Architecture Style

1. Conceptual Architecture

Overview

The conceptual architecture defines the high-level components and their interactions in the stock prediction application, illustrating the data flow and relationships between key system elements.

Components and Interactions

1. UI Component

- Primary user interface for interacting with the application
- Displays stock data, predictions, and analysis
- Connects to Search Component to retrieve and display information

2. Search Component

- Manages data retrieval and search functionality
- Interfaces with UI to present user-requested information
- Receives data from Prediction Engine and Analysis Engine
- Connects to UI for displaying search results

3. Database

- Central data storage system
- Stores historical stock data
- Provides processed data to Prediction Engine
- Receives processed data from Data Processing module

4. Prediction Engine

- Generates stock price and trend predictions
- Receives data from Search Component
- Pulls processed data from Database
- Sends prediction results to Search Component

5. Data Processing

- Transforms raw data into structured, analyzable format
- Receives raw data from Data Ingestion
- Processes and stores data in Database

6. Data Ingestion

- Collects raw stock data from external sources
- Scrapes website daily for latest stock information

- Passes raw data to Data Processing module

7. Analysis Engine

- Performs complex statistical and financial analysis
- Generates insights from processed data
- Sends analysis results to Search Component
- Retrieves data from Database

2. Execution Architecture

Overview

The execution architecture focuses on runtime interactions, communication protocols, and service calls between different system components.

Components and Communication

1. GUI (Graphical User Interface)

- Initiates HTTP requests to Service layer
- Provides user interaction point
- Sends and receives data via HTTP protocol

2. Service Layer

- Manages callbacks and orchestrates component interactions
- Provides callback mechanism to:
  - Prediction Engine
  - Analysis Engine

3. Prediction Engine

- Makes synchronous calls to Data layer
- Retrieves necessary data for generating predictions
- Returns prediction results to Service layer

4. Analysis Engine

- Performs synchronous calls to Data layer
- Retrieves data for conducting analysis
- Returns analysis results to Service layer

5. Data Layer

- Handles data retrieval and storage
- Responds to synchronous calls from Prediction and Analysis Engines

3. Implementation Architecture

Overview

The implementation architecture details the specific technologies, frameworks, and modules used to build the application.

Frontend

React Components

- Web browser-based user interface
- Sends HTTP requests to Application Server
- Implements responsive and interactive UI design

Backend

Application Server (Spring Boot)

- RESTful API implementation
- Handles request routing and processing
- Manages application logic and service coordination

Application Components

1. Data Processing Module
   - Handles data ingestion and transformation
   - Implements data cleaning and preparation logic
   - Python-based data processing scripts
2. Prediction Module
   - Implements machine learning models
   - Generates stock price predictions
   - Python-based predictive algorithms
3. Analysis Module
   - Conducts statistical analysis
   - Generates insights and trends
   - Python-based analytical tools

Database

SQLite Database

- Lightweight, file-based relational database
- Stores historical stock data
- Shared resource for Java and Python components
- Enables cross-language data persistence

Technology Stack

- Frontend: React
- Backend: Java Spring Boot
- Data Processing: Python
- Database: SQLite
- Data Scraping: Python
- Prediction: Python Machine Learning Libraries