Functional Requirements

Data Ingestion Service

FR1.1: The system shall retrieve financial data from the Alpha Vantage API as an API endpoint, ensuring secure authentication and efficient data retrieval.

FR1.2: The system shall support web scraping from multiple stock exchange websites, including Yahoo Finance, utilizing scraping frameworks such as BeautifulSoup or Scrapy to extract relevant data.

FR1.3: The system shall update stock market data at predefined time intervals, configurable by the user, with options for real-time updates, hourly, daily, or weekly intervals.

FR1.4: The system shall retry failed API calls up to 3 times before logging an error, implementing exponential backoff and error handling mechanisms to prevent excessive requests.

FR1.5: The system shall provide logging and monitoring capabilities for data ingestion operations, recording timestamps, sources, and any encountered issues.

FR1.6: The system shall validate and clean incoming data, removing duplicates and ensuring format consistency before further processing.

FR1.7: The system shall support multiple data formats, including JSON and CSV, for ease of integration with other services.

FR1.8: The system shall provide alerts or notifications in case of persistent data ingestion failures, allowing administrators to take corrective action.

Real-Time Processing Service

FR2.1: The system shall use Apache Kafka for real-time data transfer between services, ensuring efficient, distributed, and fault-tolerant messaging.

FR2.2: The system shall use Apache Spark Streaming to process incoming stock market data in real time, enabling low-latency analytics and transformation of streaming data.

FR2.3: The system shall filter, clean, and normalize real-time stock data before storage, handling missing values, duplicates, and erroneous records to ensure consistency.

FR2.4: The system shall provide real-time notifications for significant market movements based on predefined thresholds, allowing users to take immediate action.

FR2.5: The system shall support event-driven processing, triggering automated workflows when specific stock patterns or anomalies are detected.

FR2.6: The system shall allow configurable processing pipelines, enabling users to adjust processing logic, data filtering rules, and aggregation methods dynamically.

FR2.7: The system shall support distributed processing, allowing scaling across multiple nodes to handle large volumes of streaming data efficiently.

FR2.8: The system shall log all processing activities, including errors, transformations, and alerts, for auditing and troubleshooting purposes.

Data Storage Service

FR3.1: The system shall store structured stock data in a PostgreSQL database, ensuring data integrity and efficient retrieval.

FR3.2: The database shall be indexed by stock symbol and timestamp to ensure efficient data retrieval, optimizing query performance for large datasets.

FR3.3: The system shall validate incoming data before inserting it into the database, enforcing schema constraints and detecting anomalies.

FR3.4: The system shall support data backup and recovery procedures to prevent data loss, implementing automated backups and failover mechanisms.

FR3.5: The system shall ensure ACID (Atomicity, Consistency, Isolation, Durability) compliance to maintain database reliability and prevent data corruption.

FR3.6: The system shall allow historical data archiving, enabling long-term storage and retrieval of older stock records without affecting performance.

FR3.7: The system shall support role-based access control (RBAC) to restrict unauthorized modifications to stored data.

FR3.8: The system shall provide logging and audit trails for all database transactions to enhance security and facilitate debugging.

Data Analysis Service

FR4.1: The system shall perform statistical analysis on historical stock data to identify trends, using techniques such as moving averages, volatility measurements, and correlation analysis.

FR4.2: The system shall use machine learning algorithms to generate stock price predictions, leveraging models such as regression analysis, neural networks, and time-series forecasting.

FR4.3: The system shall generate analytical reports on stock performance and market trends, providing users with insights through charts, summaries, and key performance indicators.

FR4.4: The system shall allow users to define custom analysis parameters, such as time range, stock symbols, risk factors, and preferred analytical methods.

FR4.5: The system shall support real-time data analytics, allowing users to assess market trends and predictions based on the most recent stock data.

FR4.6: The system shall provide anomaly detection capabilities, alerting users to unusual stock movements or patterns that may indicate significant market shifts.

FR4.7: The system shall allow integration with external analytical tools and libraries to enhance flexibility in data interpretation and visualization.

FR4.8: The system shall log all performed analyses and user-defined parameters for auditing and reproducibility of results.

Data Visualization Service

FR5.1: The system shall visualize stock data using graphs, bar charts, and line charts, offering multiple visualization formats to accommodate different user preferences.

FR5.2: The system shall use Plotly or similar libraries to generate interactive visualizations, allowing users to interact with and explore stock data effectively.

FR5.3: The system shall allow users to select specific data points and zoom in on trends, enabling detailed examination of stock performance over various timeframes.

FR5.4: The system shall support exporting visualizations in image and PDF formats, providing users with the ability to save and share their analysis easily.

FR5.5: The system shall provide customizable chart settings, including color schemes, axis labels, and data point annotations, to enhance the clarity and presentation of data.

FR5.6: The system shall support real-time updates in visualizations, ensuring that graphs and charts reflect the latest stock data without requiring manual refreshes.

FR5.7: The system shall allow users to compare multiple stocks or indices in a single visualization, facilitating side-by-side performance analysis.

FR5.8: The system shall integrate with dashboarding tools to provide a centralized view of stock trends and analytics, improving accessibility and usability.

Non-Functional Requirements

Performance

NFR1.1: The system shall process real-time stock data with a maximum latency of 2 seconds, ensuring timely analysis and decision-making.

NFR1.2: The system shall support concurrent data processing tasks without significant performance degradation, optimizing resource utilization and system efficiency.

NFR1.3: The system shall handle at least 10,000 API requests per hour without failure, ensuring scalability and robustness under heavy load.

NFR1.4: The system shall optimize query performance for large datasets by using indexing, caching, and parallel processing techniques.

NFR1.5: The system shall ensure efficient memory management, preventing excessive consumption of system resources during intensive processing.

NFR1.6: The system shall implement load balancing techniques to distribute incoming requests evenly across multiple servers.

NFR1.7: The system shall maintain a high availability rate of 99.9%, minimizing downtime and ensuring continuous service.

NFR1.8: The system shall provide real-time monitoring and alerting mechanisms to detect performance bottlenecks and failures proactively.

Scalability

NFR2.1: The system shall be designed to scale horizontally to support increasing data volumes, allowing additional servers or processing nodes to be added as needed.

NFR2.2: The architecture shall support the addition of new data sources with minimal code changes, ensuring flexibility in integrating multiple stock exchanges and APIs.

NFR2.3: The system shall use distributed computing techniques to efficiently handle large datasets and concurrent processing tasks.

NFR2.4: The system shall automatically adjust resource allocation based on workload demands, optimizing processing power and storage utilization.

NFR2.5: The system shall ensure seamless scaling of data ingestion, processing, storage, and visualization components without significant downtime or reconfiguration.

NFR2.6: The system shall provide load balancing mechanisms to evenly distribute requests across multiple servers, preventing bottlenecks and ensuring consistent performance.

NFR2.7: The system shall support multi-region deployment options to improve data availability and reduce latency for global users.

NFR2.8: The system shall allow for modular expansion, enabling new analytics, visualization, or processing features to be integrated without major architectural changes.

Reliability

NFR3.1: The system shall achieve a data accuracy rate of 99.9% for all processed stock data.

NFR3.2: The system should implement robust validation checks to prevent incorrect data from being entered or processed.

NFR3.3: The system shall implement automatic retries in case of temporary network failures or server errors.

NFR3.4: The system shall include alternative data sources or backup systems to maintain data access during primary system downtime.

NFR3.5: The system should provide real-time status updates to users during any disruptions or maintenance to ensure transparency and user awareness.

Security

NFR4.1: All data transmitted between system components shall be encrypted using HTTPS to prevent unauthorized access during transmission.

NFR4.2: The system shall implement role-based access control (RBAC) to restrict unauthorized access based on user roles and permissions.

NFR4.3: The system shall log all user activities for audit and security monitoring purposes, ensuring traceability and accountability.

Usability

NFR5.1: The user interface shall provide clear and interactive elements for data visualization and analysis to enhance user experience.

NFR5.2: The system shall provide detailed error messages and troubleshooting logs to assist users and developers in debugging issues efficiently.

Maintainability

NFR6.1: The system shall use a modular code structure to facilitate maintenance, scalability, and future enhancements by isolating functionalities into independent components.

NFR6.2: The system shall include inline comments in the code and an external developer guide to ensure clarity and ease of understanding for developers.

NFR6.3: Proper dependency management practices shall be implemented to ensure external libraries or components can be updated or replaced without major disruptions.

NFR6.4: Robust error handling and logging mechanisms shall be in place to facilitate debugging, including meaningful error messages and comprehensive logs of system events.

Portability

NFR7.1: The application shall be compatible with Windows, macOS, and Linux to ensure accessibility across different operating systems.

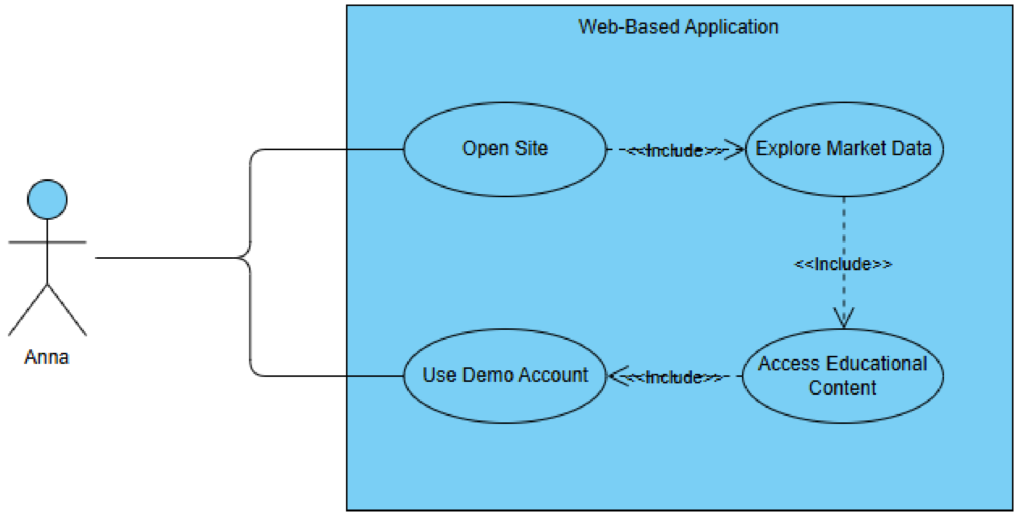
NFR7.2: The system shall require only Python and PostgreSQL as core dependencies, reducing complexity in deployment across platforms.

Compliance

NFR8.1: The system shall comply with all relevant financial data privacy regulations to protect sensitive user information.

NFR8.2: The application shall maintain audit logs that meet industry standards for compliance, ensuring transparency and accountability in operations.

User Scenario



User Persona

Example persona
Anna is a 28-year-old marketing professional with no prior experience in stock trading. She recently became interested in investing to grow her savings and achieve financial independence. Anna values simplicity and education, as she is unfamiliar with market concepts and trading strategies. She uses the application to learn about stock trading, explore market trends, and practice with demo accounts before making real investments.