Big Data - System Design Document

DataFinance Analytics - Milestone 3

Alvin Isaac

Gerald Fattah

Max Savasta

Saif Alzaabi

CSCI 4907- Introduction to Big Data and Analytics

Dr. Roozbeh Haghnazar

Purpose

This project aims to use the power of big data analytics in order to gain deeper insights into stock valuation and market sentiment. The primary objective is to construct a comprehensive data processing pipeline that combines quantitative data from established financial institution APIs with qualitative data derived from news articles and social media platforms like Twitter. Utilizing AWS tools, the project aims to efficiently handle and store large volumes of financial data, ensuring data quality and consistency across diverse sources. Ultimately, this project seeks to contribute valuable insights into the intricacies of financial markets, empowering decision-makers with a holistic understanding of stock dynamics and sentiment trends.

Scope

This project encompasses the creation of an extensive big data system that incorporates Google Cloud Platform (GCP) Big Data tools, utilizes Jupyter Notebook for visualization, employs Google Cloud Functions for data processing, and Lambda for data integration. It addresses various stages of the data lifecycle, including collection, storage, analysis, visualization, and forecasting.

Project Design

Data Ingestion, Data Transformation, and Data Visualization form the core components of our pipeline. These integral stages define the architectural framework and each of the softwares utilized in our project can be placed into one of these categories. Below is a detailed description of each of these components.

Data Ingestion:

In the data integration phase, Lambda served as a pivotal component in seamlessly pulling stock information from the NASDAQ API and efficiently inserting it into an S3 bucket. Leveraging the serverless computing capabilities of Lambda, the integration process was streamlined, allowing for automated and scalable data transfer. Lambda's event-driven architecture enabled the immediate response to triggers, ensuring that stock information was regularly updated in the S3 bucket. This dynamic integration not only facilitated real-time data retrieval but also optimized storage and accessibility within the AWS ecosystem. The serverless approach offered by Lambda enhanced the efficiency of data integration, contributing to the overall effectiveness of the project's architecture. Next in our data integration process, we transferred the information stored in the S3 bucket to Google Cloud. This cross-cloud data transfer was executed to leverage the unique benefits offered by Google Cloud Platform (GCP). The integration between S3 and GCP facilitated a more comprehensive and flexible data ecosystem. By harnessing the strengths of both AWS and GCP, we ensured a robust data flow, allowing for improved analytics and visualization capabilities. This dual-cloud approach not only maximized the advantages of each platform but also enhanced data redundancy and accessibility.

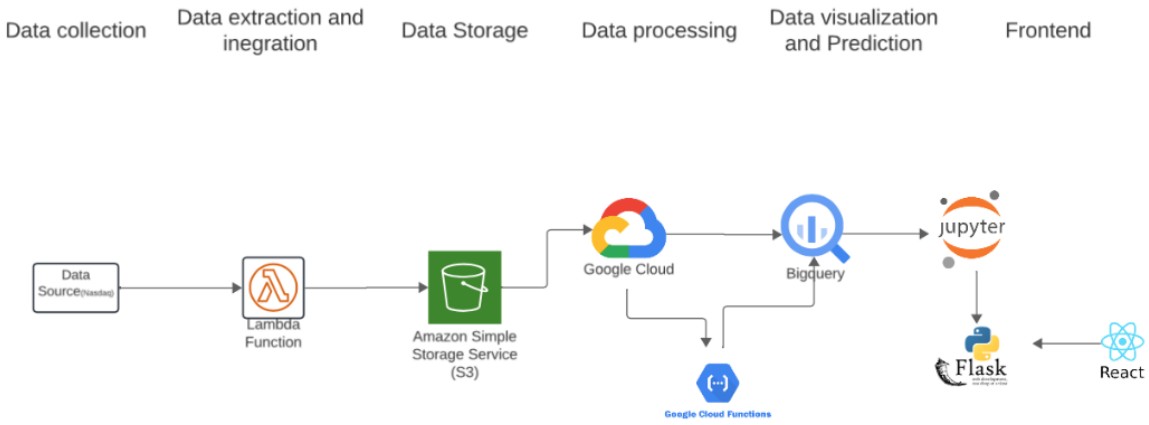
Data Transformation:

In the data processing phase, Google Cloud Functions played a pivotal role as we created functions tailored to the requirements of our project. These functions were specifically designed to refine and enhance the incoming data within Google Cloud, eliminating redundancy and ensuring proper formatting. Leveraging the serverless capabilities of Google Cloud Functions, we achieved an agile and scalable data cleaning process. The software processed the data before sending it to be stored in tables within BigQuery. This approach not only streamlined the data processing pipeline but also allowed us to use the analytical power of BigQuery for insightful queries and data analysis.

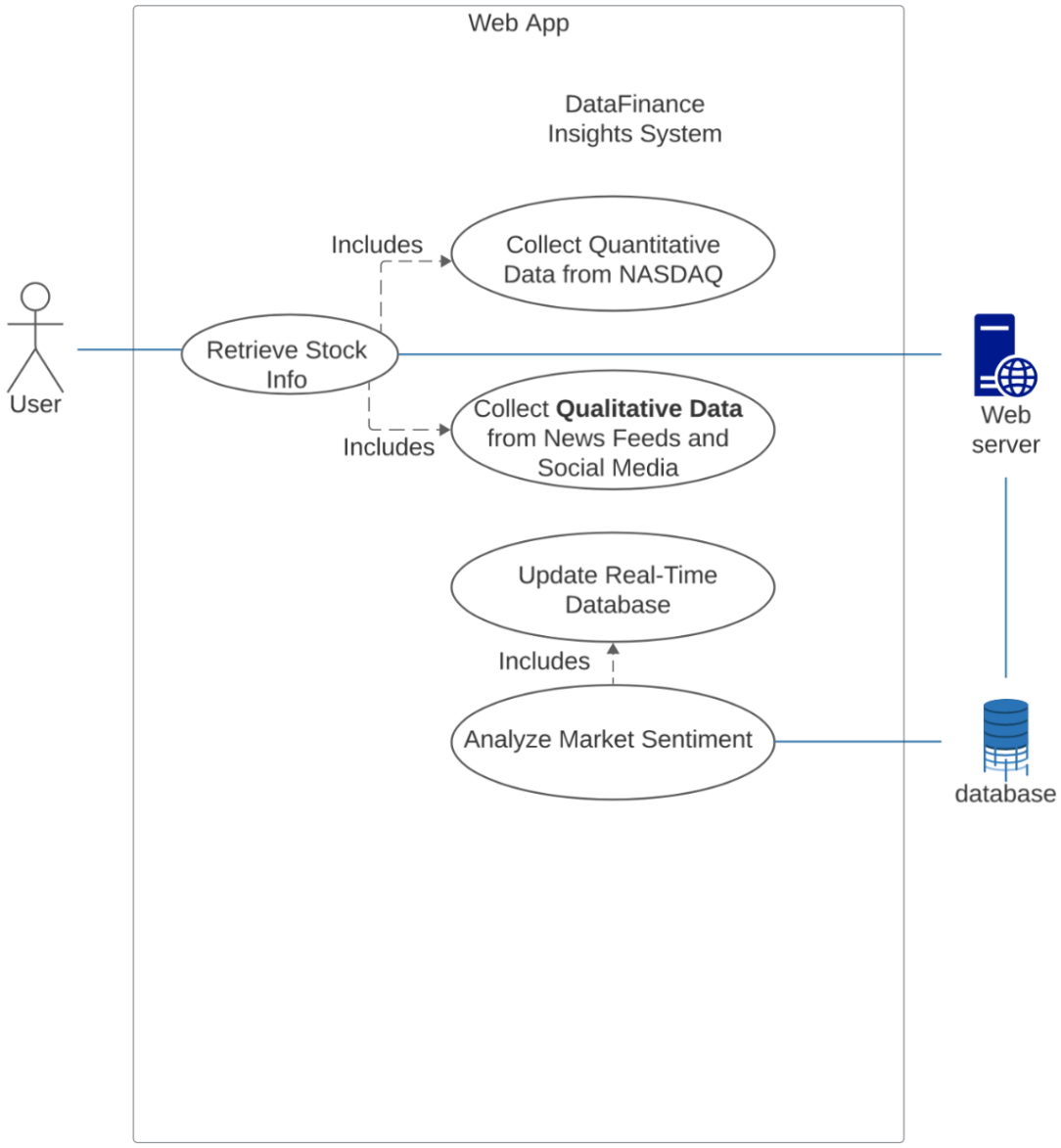
Data Visualization:

In the data visualization phase, we employed Python scripts to create dynamic and insightful visual representations of the collected data. These scripts served as a crucial tool in transforming our BigQuery data into meaningful graphs and charts. By leveraging popular visualization libraries, the scripts brought the data to life, offering users a clear and intuitive understanding of complex patterns and trends. To enhance user accessibility, the visualizations were seamlessly integrated into a frontend user interface. This interactive interface provides users with a user-friendly platform to explore and interpret the data comprehensively. The Python scripts, acting as the backbone of this visualization process, empower users to extract valuable insights from the data collected.

Architecture Diagram:



Use Case Diagram:



Component Diagram (Data Retrieval):