**RevStox**

**(Data Engineering Project)**

Center of Excellence

03-Aug-2023

Version 1.0

Copyright ©: 2023, Revature.

 Table of Contents

[Application Overview 3](#_Toc1555815592)

[Core Functional Requirements 3](#_Toc1194330873)

[Standard Functional Scope 3](#_Toc1763839213)

[Definition of Done 4](#_Toc2020375305)

[Competency-wise scoping 4](#_Toc1523109773)

[Non-Functional Expectations 6](#_Toc1436798500)

​​

​​

​

​

​

​

​

​​

​

​

​

​

​

​​

# Application Overview

The RevStox is a data engineering project aiming to provide users with valuable insights into historical stock market data for informed decision-making. The project will leverage a dataset containing various stock-related attributes, including date, symbol, open, close, high, low prices, trading volume, and more to deliver a robust, scalable, and user-friendly solution for stock analysis and decision-making.

# Core Functional Requirements

As a User, I want to:

**Data Visualization:**

1. Create historical price charts showing Open, High, Low, Close prices over time to reveal trends and patterns.
2. Calculate daily price volatility using the High and Low prices and visualize using candlestick or range bar charts. This helps plan entry/exit points.
3. Plot the Volume Weighted Average Price (VWAP) over different timeframes like 1 day, 1 week, 1 month. VWAP shows the average price weighted by volume.
4. Analyze Open and Close prices to identify candlestick patterns like Doji, Hammer, Shooting Star which indicate potential reversals.

**Data Analysis:**

1. Analyze the percentage of deliverable volume for a stock, indicating its liquidity and the level of institutional interest.
2. Study the number of trades executed for a specific stock, helping me to assess its popularity and potential for intraday trading opportunities.
3. Explore the relationship between trading volume and stock price movements, helping me to spot any potential correlations or trends.
4. Perform a comparative analysis of multiple stocks' VWAP, helping me to determine their relative value and attractiveness for inclusion in my investment portfolio.

**Data Aggregation:**

1. Calculate the daily turnover (the total value of traded shares) for a stock, helping me understand its market liquidity and overall trading activity.
2. Calculate and visualize the volume-weighted average price (VWAP) for a stock, assisting me in understanding its fair value over different time periods.

**Data Mining:**

1. Identify and analyze any gaps (differences between the previous close and the next open) in a stock's price, as they may indicate potential price movements or risks.

# Standard Functional Scope

1. Ingesting and processing various financial data feeds.
2. Transforming data into analytical models.
3. Providing interactive dashboards, queries, and reports for analysis.
4. Implementing administrative controls for user management and data governance.

This scope ensures efficient data handling, advanced analysis, and user-friendly tools, facilitating seamless transformation of raw financial data into valuable insights for informed decision-making.

# Definition of Done

1. Working application demonstration.
2. Sharing the associates’ code repo for technical review with:

* Architecture
* data models
* ETL documentation

# Competency wise scoping

|  |  |  |
| --- | --- | --- |
| **Competency** | **Project Type** | **Expectations** |
| Python, SQL  Scala, SQL | REST API service | **Framework Specific**   1. Ensure the appropriate APIs are used for any of the API calls. 2. Ensure the routing is centrally configured 3. Best practices & design patterns are to be followed.     **Validation and Error Handling:**   1. Validate the inputs for their types and format. 2. Display functional-related user messages (either for input/error/output) - no system error codes or SQL error codes. 3. Handle the exceptions and errors gracefully.     **Logging:**   1. Ensure the application is using proper logging framework and methods. 2. Ensure the application’s log level is configured using configuration files so that it can be changed without changing the code. 3. Also ensure that the application logging is configured to output to the mentioned log file.     **Testing**:   1. Ensure sufficient test cases are written using appropriate testing frameworks. 2. Ensure the code coverage closed to be 80%     **Security**:   1. Ensure the SQL injection threat is taken care. 2. Ensure the CORS restriction is applied, if applicable. 3. Ensure that the secrets are stored as environment variables using secure credential storage.   **Coding Standard:**   1. Use the industry coding standards and conventions. 2. Modular based code development for better reusability. 3. Ensure proper usage of resource objects such as database connectivity objects to avoid resource leakages. 4. Ensure proper usage of design patterns and application layering (such as Business Service, DAO Layer etc.) wherever applicable. |
| Spark  Hadoop, Hive | Data Science | **Data Preprocessing:**   1. Load the stock market dataset from a reliable source. 2. Handle missing data, outliers, and any data quality issues appropriately. 3. Perform data cleansing, normalization, and transformation to prepare it for analysis.   **Exploratory Data Analysis (EDA):**   1. Generate descriptive statistics to understand the dataset's basic characteristics. 2. Visualize historical stock price trends, trading volume, and other relevant attributes over time. 3. Identify patterns, correlations, and anomalies in the data through various charts and graphs.   **Historical Price Analysis:**   1. Calculate and plot daily price fluctuations (High-Low range) for each stock. 2. Identify and visualize gap patterns (difference between previous close and next open prices). 3. Analyze and plot moving averages to detect long-term price trends.     **Volume Analysis:**   1. Calculate and visualize the volume-weighted average price (VWAP) for each stock. 2. Explore the relationship between trading volume and price movements. 3. Analyze and plot trading volume patterns over time.   **Candlestick Pattern Analysis: -** *Nice to have.*   1. Implement algorithms to identify common candlestick patterns based on open, close, high, and low prices. 2. Present the identified patterns through visualizations to aid technical analysis.   **Institutional Interest Analysis: -** *Nice to have.*   1. Calculate and plot the percentage of deliverable volume for each stock. 2. Identify and visualize the impact of institutional interest on stock performance.   **Trading Strategies: -** *Nice to have.*   1. Develop and back test simple trading strategies (e.g., moving average crossover) using historical data. 2. Evaluate the performance of the strategies and provide insights on potential profitability. |
| Azure  AWS  GCP | ETL Pipeline | **Deployment artifacts:**   1. The deployment artifacts should be minified and obfuscated if required.   **Pipeline Orchestration**   1. Scheduled ETL jobs monitor, orchestrate and restart flows 2. Job monitoring and alerts for data quality and SLAs 3. Scalable compute resources based on workload   **Scalability and Performance:**   1. Design the ETL pipeline to handle large volumes of stock data efficiently. 2. Optimize data processing and transformation to minimize execution time.   **Monitoring and Alerts:**   1. Set up monitoring for ETL job status, performance metrics, and data quality. 2. Configure alerts to notify stakeholders about failures or anomalies in the ETL process. |

# Non-Functional Expectations

* Application development should use version control systems (e.g., Git) to manage the project codebase and facilitate collaboration.
* Application development is supposed to follow the Scrum process.

# Source Data Location

The data can be acquired from Kaggle using:

* Link to the csv files [Technology - stock-data - All Documents (sharepoint.com)](https://revature0.sharepoint.com/sites/trainers/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2Ftrainers%2FShared%20Documents%2FCenter%20of%20Excellence%2FProject%20Operations%2FData%20Engineering%2FData%20Source%2Fstock%2Ddata&viewid=d59398ae%2Dbc86%2D4d4e%2D9ff3%2Dc9cd2bfaea22)