

STOCKVIEW USE CASE



“ For Day Traders to Retrieve:

- Real-time low-latency stock price information
- Current-day time-series candlestick price chart
 - in last 15m, 45m, 1h, 3h viewing
- Recent relevant news

“ StockView Functions:

- A day trader search engine that takes in user input of company ticker
- Secure user Sign-Ups and Log-Ins
- User profile storage



DATA SOURCE SPECIFICATION AND PROCUREMENT DETAILS



{ <https://www.yahoofinanceapi.com/> }

- time-series stock price data (open, high, low, close)
- `finance.download(tickers)` to retrieve stock prices in dataframe



{ <https://newsapi.org/> }

- relevant news articles (topic, sources, article title, url, description)
- `newapi.geteverything()` to retrieve all the news that matches the topic, sources and language in dict. json

PROPOSED DESIGN CHOICES AND THE RATIONALE FOR USING THE SELECTED TECHNOLOGIES

- **Flask API & html** / for interface display and interaction

Accepts user inputs, allows user interaction, displays desired output for relevant news articles and stock prices in interactive candlestick charts

- **MongoDB database applications** / for data storage and querying/retrieval

Creates collections, stores, retrieves, and performs queries on data in json format

- 1) Relevant info from newsapi
- 2) User profile data from account registration

- **Information retrieval API services using Python** / newsapi.org & Yahoofinance API

- 1) Newsapi: aggregates news data from various sources and accepts search by topic
- 2) YahooFinance API: reliable, real-time low-latency, accepts search by ticker symbol



APPLICABLE DATA GOVERNANCE POLICIES AND COST IMPLICATIONS

(CAN BE PRELIMINARY)

“ POLICIES

- The data displayed on and analyzed for the final product are all publicly available information, of publicly traded stocks and published news articles.
- User profile data are obtained solely from users' voluntary registration, limited to name, email and password, while fill-ins for other sensitive information like gender are optional. Storage of data is only used for secure log-in purposes, and no other usage will be allowed. We also do not record users' search history or any interaction on the platform other than signups and logins for confidentiality.

“ COSTS

- At the current developer stage of the project, there are no cost implications as both News API and Yahoofinance API are free of charge, each ≤ 100 requests/day.
 - However, if we do decide to scale it up, for production and published commercial projects ($\leq 250,000$ requests/month) , News API would cost **\$449 per month**, while Yahoofinance API would cost **\$259 per month** for unlimited requests (still under 300 requests/min).
-



EVALUATION CRITERIA

Since we are building a search application, our **primary success criteria are retrieval time and relevance of searches, and user satisfaction**. Possible detailed metrics (some are qualitatively, and some are quantitatively) are listed as follow:


Mean Satisfaction Rate

- By using pop up window asking whether the users are satisfied with this search experience/ result (1-satisfy, 0-not satisfy), or distributing short questionnaire asking about users' searching experience verbally
- Calculated by **mean satisfaction rate = sum of satisfaction score (1-satisfy, 0-not satisfy) of all participants / count of participants**
- Significance: this metric measure the how our search engine can contribute to the platform in long term

Mean Relevance Rate of Searches

- Calculated by **mean relevance rate of searches = sum of relevance score of each search action (1-relevant, 0-irrelevant) / count of search actions** (whenever an search action has a relevant result, we call it relevant, else irrelevant)
- Significance: this metric **measured the product itself**, we want the relevance rate of searches (1-relevant when there is at least one matched result, 0-irrelevant) as large as possible

Mean Retrieval Time

- Calculated by code execution explained in the next slide
 - Significance: this metric measures the efficiency of our search engine and the underlying database management implementation; It can also potentially affect satisfaction rate.
- 

ASSESSMENT OF SUCCESS BASED ON THE PROPOSED METRICS

1

MEAN SATISFACTION RATE MEAN RELEVANCE RATE OF SEARCHES

- Rate will be calculated based on customer survey results
- Further testing required, the higher the better

2

MEAN RETRIEVAL TIME

Time consumed for each search will be retrieved by running the following code:

- `start = time.perf_counter()`
- `end = time.perf_counter()`
- `print("time consuming : %.2fs" % (end - start))`

```
flaskProject x
In folder /Users/james/PycharmProjects/flaskProject
/usr/bin/python3 -m flask run
* Serving Flask app 'app.py' (lazy loading)
* Environment: development
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [20/Apr/2022 14:31:00] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [20/Apr/2022 14:31:03] "GET /login HTTP/1.1" 200 -
127.0.0.1 - - [20/Apr/2022 14:31:04] "POST /login HTTP/1.1" 200 -
time consuming : 0.21s
127.0.0.1 - - [20/Apr/2022 14:31:09] "POST /result HTTP/1.1" 200 -
```