CS 222 - Team 80

- Siddarth Aananth (aananth2)
- Sambuddha Biswas (sbiswas7)
- Kaushik Pulgari (kpulg2)
- Suchit Bapatala (suchit2)

IFI

(Instant-Financial-Information) Terminal

February 08, 2023

Pitch

Introducing the IFI Terminal - a financial tool that combines real-time stock information, financial news, and charting features with sentiment analysis from online discussion boards to provide a comprehensive view of both retail trader perspectives and traditional fundamental and technical information. Make smarter investment decisions with a single source of insightful knowledge.

Functionality

1. Retrieval of information on the most widely discussed stocks and securities from online forums. This will include NLP based analysis as well.
2. Integration of multiple APIs, making the terminal a one-stop aggregator for diverse financial information sources
3. Access to fundamental financial information of companies
4. Real-time display of stock prices, accompanied by UNIX timestamps
5. Provision of up-to-date market news on selected stocks
6. Visualization of stocks through interactive candlestick charts
7. Continual updates of financial information, ensuring users always have access to the latest data

8. Customizable options for receiving stock updates and searching for specific stock information, charts, and trends
9. Terminal-based implementation enables users to install a low storage software as it utilizes system graphics rather than incorporating additional graphics components, resulting in reduced storage requirements

Components

1. **Backend:** We wil be using python for the backend of our application, primarily comprised of two sections.

    a. **Utilities Folder:** We would need a set of python functions to be able to interact with the APIs and to handle HTTP requests.

    b. **Main Engine:** A critical component of the backend system, responsible for processing user inputs and generating the corresponding results. It functions as a logic unit that manipulates data to produce an intermediate version of the output before it is finally presented to the user.

2. **Frontend:** This section of the project will utilize Python to create a terminal-based user interface that integrates and presents the results from the backend. The focus is to create an intuitive and straightforward interface that provides an easily understandable representation of the information processed by the backend.

3. **Major Libraries Used:** In addition to internal libraries utilized for testing, JSON handling, basic modeling, and visualization, the only external libraries required are those necessary for API interactions such as YahooFinance, FinHub API, Reddit API, and a few supplementary charting tools.

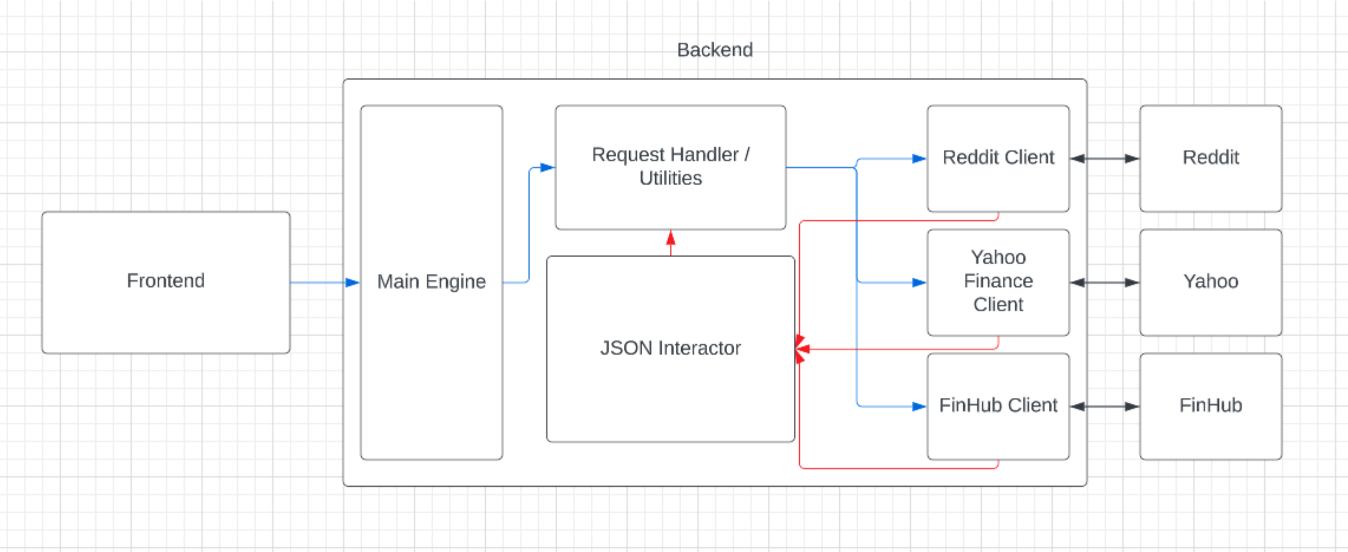
4. **Testing Methodology:** Some functionalities can be evaluated using libraries and pre-established test cases, for instance, the NLP application of online forums can be assessed by verifying the TF-IDF and tokenization calculations. However, for financial information, manual verification of the output data must be performed to ensure its accuracy, as it cannot be tested through automated means.

5. **Additional Considerations, Languages and Component Interactions:**

   a. It is important to exercise caution when handling sensitive information, such as API keys, and therefore, a secrets file should be created and kept confidential. The only other aspect to consider regarding component integration is that HTTP requests will be utilized to transmit and receive data in JSON format.

   b. Python is ideal for this project due to its simple and straightforward syntax, which makes it useful for creating large-scale applications, rendering simple terminal-based graphics, handling HTTP requests, and producing clear code. The all-in-one functionality of Python is the main motivation behind its selection as the primary language.

   c. As the project will only utilize Python and a terminal for the user interface, we decided to not separate the front-end and back-end components and instead opt for a monolithic system. (Note that the separation was mentioned above only for descriptive purposes). The whole project will be developed in a conda based virtual environment.



Continuous Integration

1. **Library:** The team will utilize the **unittest** library to perform tests on the Python code. This library was selected due to its capability to perform isolated testing and its compatibility with the command line. Additionally, it is a built-in component of Python, which eliminates the need for external installation.
2. **Style:** The team has decided to use snake_case, where variable names will be separated by underscores to clearly separate words for improved readability. Visual Studio extensions such as Case Converter can be used to convert between different styles. This tool is easily installed on our devices and can be useful for automatically checking and converting coding styles.
3. **Coverage:** Our team will evaluate the code coverage using the conda-installed coverage tool. It is a straightforward command that can be installed on a local device. In addition, manual coverage algorithms can be developed to test the coverage of test cases. The unittest library supports the use of coverage in the command line, and the addition of coverage to a test command should be sufficient to report test coverage.
4. **Pull Requests:** Weekly code reviews will be conducted through pull requests. These pull requests will be submitted before Sunday each week to allow ample time for feedback. The reviewer will typically be another team member working on a similar task (e.g. backend or scripting). In the event that a reviewer is unable to fulfill their role, alternative arrangements will be made through communication on Discord. Discord will also serve as the primary platform for scheduling review times and handling exceptions. To resolve merge conflicts, the team will either hold a discussion on Discord or provide comments on which code worked best in certain test cases. Based on the code's accuracy and readability, the team will make a decision on Discord regarding which code to merge and which to overwrite, potentially leading to a cleaner solution.

Schedule

- **Week 1:** Requirements gathering and project planning. This includes defining project goals, determining the scope of the project, and creating a high-level project plan.
- **Week 2-3:** API integration and data retrieval. Work on integrating multiple APIs to gather real-time stock information, financial news, and other data including minute details like the UNIX timestamps.
- **Week 3-4:** Development of stock price display and visualization features. This includes the implementation of real-time stock prices display with UNIX timestamps and other information, as well as the development of interactive candlestick charts.
- **Week 5-6:** Implementation of sentiment analysis and customization features. This includes integrating sentiment analysis from online discussion boards and implementing customization options for receiving stock updates and searching for specific stock information, charts, and trends.
- **Week 7-8:** Testing and debugging. Final testing of all features and fixing any bugs or issues that may arise.

Risks

- **Technical difficulties**: Our application will integrate multiple APIs, and the data obtained will depend heavily on their performance. Although API reliability is usually robust, there may be instances where the API does not return the intended data, such as in the case of the chatGPT API not recognizing a company. To mitigate these risks, we will verify the data obtained from the APIs to ensure its accuracy and relevance, and we may also explore alternative APIs that can provide the necessary information.
- **Lack of expertise**: Our team may not possess the full breadth of knowledge in the financial news industry, which could result in potential misunderstandings or oversights that affect the app's functionality. To address this, we propose to extensively utilize online resources to familiarize ourselves with the industry and its practices. Additionally, we can seek collaboration with financial experts or individuals with greater experience in the sector to further enhance our understanding and capabilities.

- **Team dynamics:** There's always a chance of encountering friction or miscommunication among team members, which could negatively impact the project's progress and outcome. To mitigate this risk, we should establish a clear project timeline with defined milestones, conduct regular code reviews, and test functionality from the start. It's also important to be flexible and adaptable as unforeseen challenges may arise and changes to the timeline may be necessary.

Teamwork

To minimize disruptions and enhance efficiency, utilizing a shared product management tool that serves as a centralized hub for all project-related information, such as schedules, assignments, and progress updates, is advisable. This approach is beneficial as it provides a clear and transparent view of the project's status, allowing all team members to remain informed and understand their responsibilities. Furthermore, it streamlines collaboration by eliminating potential communication gaps and promotes consistency in workflow. Task assignments should be aligned with each team member's strengths and availability, while establishing deadlines and a project timeline is critical to maintaining progress. Regular meetings are essential in ensuring everyone is working towards a common goal.

***