Project Activity 3

Chimzim Ogbondah & Sam Moore

Iteration 1

1. **Review of inputs**:

|  |  |
| --- | --- |
| Driver | Description |
| Use Case 1 - Scrape reddit for trending stock data | This use case will scrape subreddits to find trending stocks (happens periodically) |
| Use Case 2 - Save High and low data from trending stocks | This use case will save high and low data for trending stocks found |
| Use Case 3 - Notify user on optimal stock purchase time | This use case will allow the user to know the optimal time to purchase a stock |
| Use Case 4 - Separate data frames for stock data | This use case creates separate data frames for the trending stock data and the user’s stocks |
| QA1 - Quickly analyze periodical reddit stock data | The application will determine trending stocks periodically (every hour) in 10 minutes or less |
| QA2 - Quickly save high and low data of stocks | The application will be able to save high and low trending stock data in less than 5 seconds |
| QA3 - Quickly notify user on when to purchase stocks | The application will notify the user on the optimal time to purchase a stock in under 15 seconds |
| Constraint 1 - Compatible with yahoo API | The application must utilize yahoo finances API |
| Constraint 2 - compatible with reddit API | The application must create a bots using reddit API |
| Constraint 3 - Compatible with mobile devices and work/home computers | The application must be able to run on a different devices such as a mobile phone or laptop/desktop |
| Constraint 4 - Can have 100 concurrent users | The application must be able to work with 100 users at any given time |
| Constraint 5 - Can work with poor internet connection | The application must be able to work on low bandwidths |
| Concern 1 - Knowledge of python | Leverages the teams knowledge of python to keep prior application intact and continue forward |
| Concern 2 - Knowledge of data analysis | Leverage the team’s knowledge of data analysis tools (no machine learning) |
| Concern 3 - Quick plan for architecture | This concern addresses the need for new versions or change to occur frequently |

1. **Establish Iteration Goal:**

* The goal of this iteration is to extend the original architecture for the system to support the new use cases which can be seen in step 1 labeled Use Case 1, 2 , 3, and 4, Concerns 1 and Constraint 1 and 2.

1. **Elements to refine:**

* We will be refining the capabilities of the systems. We will be adding notifications on when to purchase a stock, about trending stocks on Reddit, and sending their high and low data.

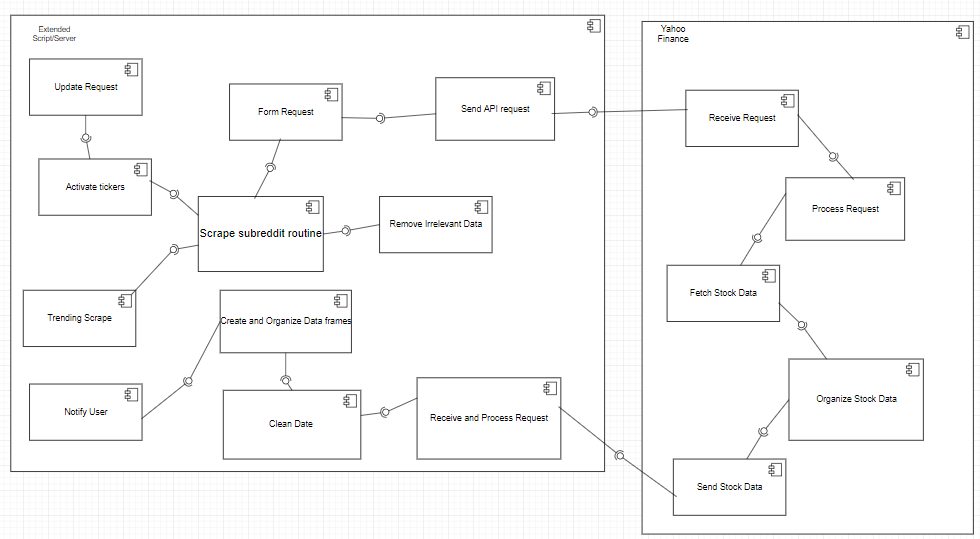
1. **Design Concepts:**

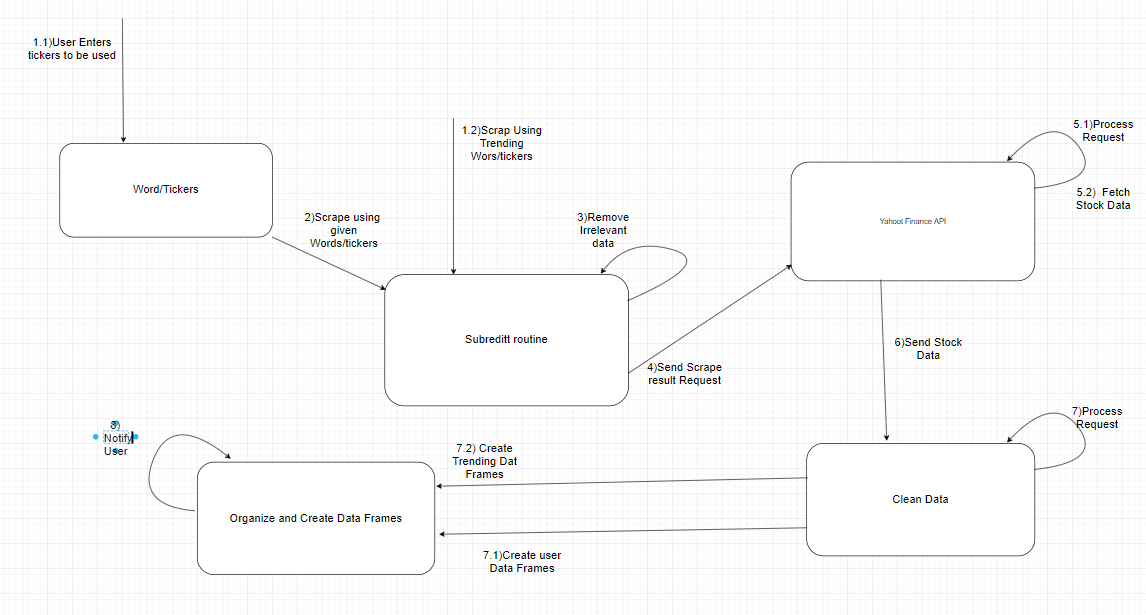
* This step involves deciding how to design a solution for the selected drivers. For this iteration, we will be looking at all the uses cases (see table in step 1) and concern 1. For concern 1, we want to leverage our knowledge of python to keep the original system intact. For this reason, we will be using a layered architecture so the new deployment will not mess with the older version. For use case 1, 2, 3, 4, we will be using the Master Slave pattern to divide up the work for reviewing data from subreddits, saving high and low data from various stocks, optimal time to purchase a stock, and creating separate data frames.

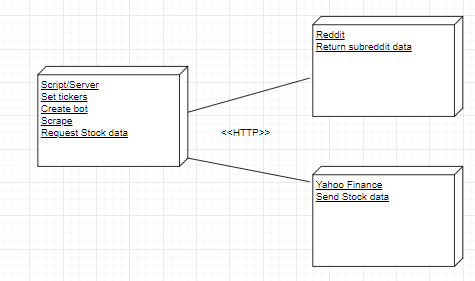
1. **Instantiate architecture elements:**

* For use case 1, we will be creating various bots using Reddit’s API to gather and review data for stocks that are currently trending. Use case 2 will require taking the trending stocks and utilizing Yahoo Finance’s API in order to gather stock data (e.g. open, high, low, close). Use case 3 will utilize Yahoo Finance’s API by looking through the history of a stock given by the user to determine optimal purchase times. Use case 4 will produce separate data frames for the stock data; additionally, we will use Master Slave to dispatch a ‘slave’ for making either data frame. Concern 1 addresses the team’s knowledge of python and not changing the original system. For this concern, layered architecture will be used in order to extend the system effectively while not corrupting the old version.

1. **Views**







1. **Analysis:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Addressed** | **Partially Address** | **Not Addressed** | **Design Decisions made** |
| Use Case 1, Use Case 2, Use Case 3, Use Case 4 |  |  | All the use cases will be implemented using Master Slave |
|  | QA1, QA2, QA3 |  | Implemented using Master Slave but not refined to meet specifications |
| Constraints 1 & 2 |  | Constraint 3, 4,5 | The architecture utilizes the reddit API to scrape data from the website and yahoo finance API to get stock information |
| Concern 1 |  | Concern 2 Concern 3 | Addressed Concern 1 through Layered architecture. |

Iteration 2

1. **Review of Inputs**:

* See table in step 1 of Iteration 1. For this iteration, we will be covering the quality attributes which are labeled QA1, QA2, and QA3 in the table. They are to quickly analyze periodical reddit stock data, to quickly save High and Low data of stocks, and to quickly notify the user on when to purchase stocks; all having to do with performance of the extension.

1. **Establish Iteration Goal**:

* The goal of this iteration is to apply the architecture design process to address and support the quality attribute mentioned above.

1. **Elements to Refine**:

* Source code: the source code must be studied, analyzed, and refined in order to optimize the speed and efficiency (changing data structures, removing unneeded logic, etc.)
* Notification settings: the user will have the ultimate control over how he or she wants to be notified, but the method and manner in which the application notifies will have to be studied and optimized in order to notify the user instantaneously.
* Database (Yahoo Finance) Server: Additionally, since this application must query this database potentially millions of times, the manner in which the application contacts the database must be refined.

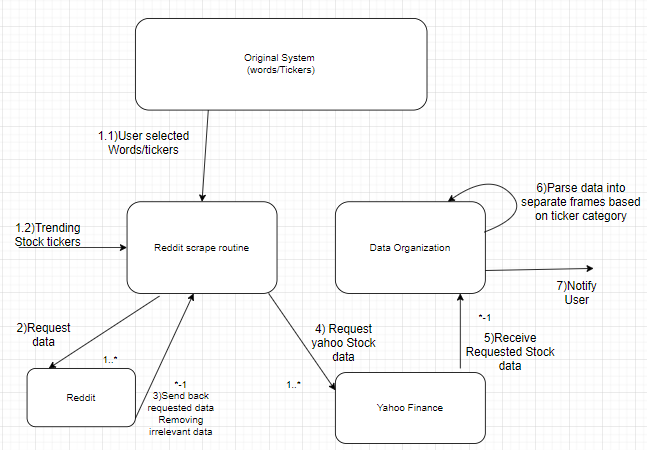
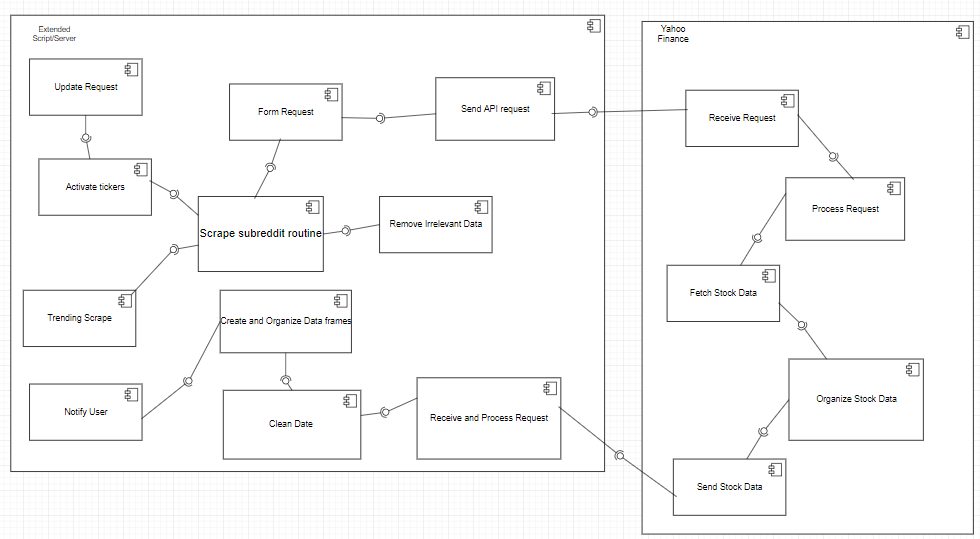
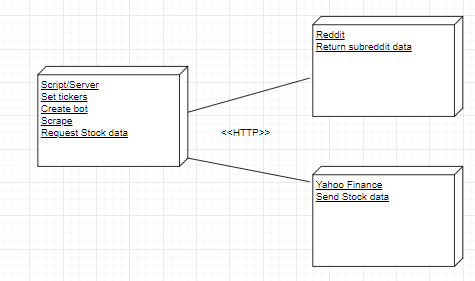
1. **Design** **Concepts**:

* To address these quality attributes, the Master Slave design will be utilized for all three of them. We chose the Master Slave approach because they all have to do with performance and how quickly we can do calculations and load data in from the Yahoo Finance API. Master Slave is the design to implement if performance needs to be improved so no single program gets overloaded with tasks; in this design approach it can disperse its tasks to “slaves” to achieve much faster execution time therefore meet our performance requirements.

1. **Instantiate** **Architecture** **Elements**:

* For QA1, we will write several different scripts that handle small parts of the problem. For example, we want to scrape subreddits periodically, so a Master Slave approach to this problem would the coding of multiple scripts, one of which would scan for bearish (negativity surrounding the stock) sentiments, another would scan for bullish (positivity surrounding the stock) sentiment, etc. This way, one single program will not have to do it all itself consequently speeding up the execution time. The same idea applies to QA2 and QA3, we will write extra scripts to avoid the overloading of a single script.

1. **Views**:



1. **Analysis**:

|  |  |  |  |
| --- | --- | --- | --- |
| **Addressed** | **Partially Address** | **Not Addressed** | **Design Decisions made** |
|  |  | Use Case 1, Use Case 2, Use Case 3, Use Case 4 |  |
| QA1, QA2, QA3 |  |  | All Quality Attributes are now fully addressed using Master Slave |
|  |  | Constraint 1, 2, 3, 4, 5 |  |
|  |  | Concern 1, Concern 2, Concern 3 |  |

Iteration 3

1. **Review of inputs**

* See table in Step 1 of iteration 1. For this iteration, we will be going covering the remaining drivers that haven’t been addressed which are Concern 2 & 3 and Constraints 3, 4 and 5.

1. **Establish iteration Goal:**

* The goal of this iteration is to address all the remaining drivers of the system that weren’t addressed in the previous two iterations.

1. **Elements to refine**

* We will be refining the capabilities and attributes of the system. We will be configuring the system to work with both mobile devices and laptops/desktops. We will allow for 100 concurrent users at any given time and structure the application to work with poor internet connection. We will also leverage the teams knowledge of data analysis when picking a configuration and architecture style.

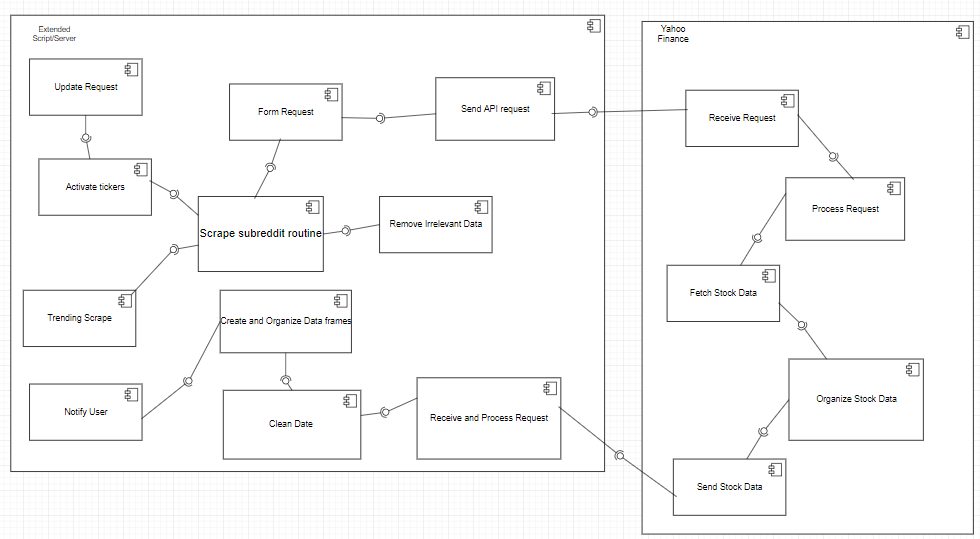
1. **Design Concepts**

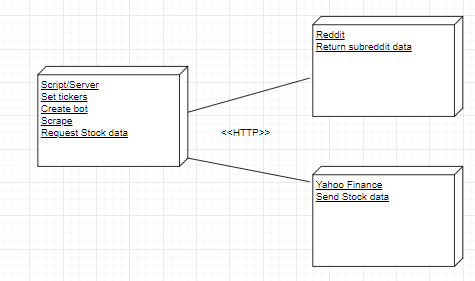
* This step includes deciding a solution for the selected drivers. For this iteration, the drivers are Constraint 3, 4, and 5 and Concern 2 and 3 (see table in step 1 of iteration 1). For Concern 2, a Master Slave pattern will be used in order to optimize the data being processed; for concern 3, layered architecture will be used overall to allow for easy modifiability that won’t affect prior deployments. Constraint 3 will use the adaptor pattern which will allow the devices to use our configured interface so it can work on various devices. Constraints 4 and 5 will use the Master Slave pattern to divide up the work so more users can interact with the script/server and can work on poor connection due to smaller data transfer sizes.

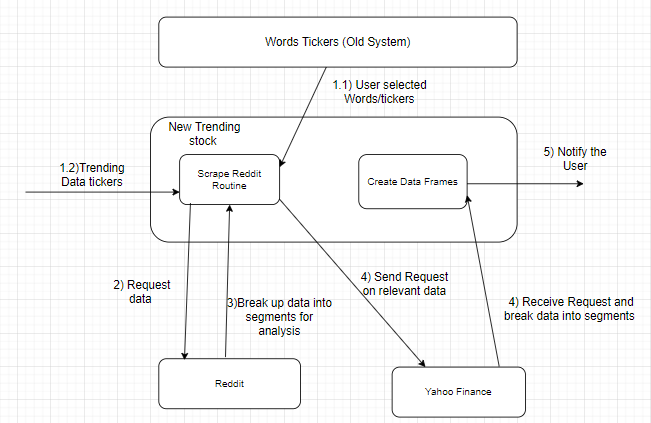
1. **Instantiate Architecture Elements**:

* For Concern 2, the data will be broken down into segments where irrelevant data will be removed and then compared to decide what data is trending. Concern 3 will create 2 layers and then a sub layer so too much time isn’t wasted during planning, implementing or revising (old system stays intact and older deployments aren’t affected). Constraint 3 will create an interface for figuring out what kind of trending data is acceptable, getting stock data and inputting user defined data stock data so it can work on any given system. Constraints 4 and 5 will divide up the work so more users can use the application and it can work in low bandwidth since the system as a whole is divided up.

1. **Views**







1. **Analysis**

|  |  |  |  |
| --- | --- | --- | --- |
| **Addressed** | **Partially Addresses** | **Not Addressed** | **Design Decision Made** |
|  |  | Use Case 1, Use Case 2, Use Case 3, Use Case 4 |  |
|  |  | QA1, QA2, QA3, QA4 |  |
| Constraint 3, Constraint 4, Constraint 5 |  | Constraint 1, Constraint 2 | The architecture utilizes and adaptor pattern for interfacing so that an device is compatible, It also address the number of concurrent users and poor internet connection due to the Master Slave pattern dividing things into segments |
| Concern 2, Concern 3 |  | Concern 1 | Concern 2 is addressed using the Master Slave pattern to find commonalities while dividing up the work and Concern 3 is addresses using Layered architecture to help with modifiability |