Software Implementation and Testing Document

For

Group 5

Version 2.0

Authors:
Darbi Bauer
Jacob Dienger
Aidan Ryan
Katie Skipper
Sarah Rosenfeld

1. Programming Languages

- Python - using for backend, handles interfacing with Yahoo price data, Google NLP, and Twitter API. Chosen because was one of the easier languages to use for this purpose. Used for simple version of price prediction.
- Java - using for app interface development. Is the language that Android Studio uses.
- Cuda C++ - GPU acceleration of price prediction and scoring of a particular company. Language was chosen because it handles the math better.

2. Platforms, APIs, Databases, and other technologies used

- Android Studio - using to create the app interface
- Google Natural Language API - using in the server side to perform sentiment analysis
- Twitter API (Twitter for Python) - using in the server side to collect specific tweets
- Firebase - using to store stock, user, and twitter watchlist information

3. Execution-based Functional Testing

- Tested twitter API using different known Twitter usernames as the searched user, and matched the output file with their most recent tweets by looking at their profiles in the mobile twitter application.
- Google and Twitter API combination script was executed many times while being written; each time, the sentiment score and test file which contains the tweets was examined to ensure that the script was correctly assessing sentiment
- Tested Yahoo minute-to-minute API V8, script checks for null input to eliminate errors further down the price prediction pipeline
- Whenever I added a function or screen to the app, I would install the new version on my phone and run through each screen and testing each button.
- When adding the login functionality I tested after implementing each part - user creation, user login, and user information change. I also entered different values in for the email and password in user creation to detect errors. One such error occured when both fields were left blank resulting in a quick patch to fix it.
- Tested number of tweets output by Twitter API by counting them and ensuring that it was the number we wanted.

4. Execution-based Non-Functional Testing

Execution-based non-functional testing has not really been performed yet. This will be carried out as we get closer to finalizing the application.

5. Non-Execution-based Testing

- Read through code of Twitter API script and compared to the python-twitter API documentation to ensure the correct functionality was happening
- Code review of the Google and Twitter API combination script was performed with a few members of the group in order to ensure the correct results were being outputted by the code
- For coding the app I was able to see any errors while coding because of the editor in Android Studio. This allowed me to quickly see if anything wouldn't compile before having to wait for a compilation to occur.