Progress Report

- Increment 2 -

Group #5

1) **Team Members**

   Aidan Ryan - afr17b - AidanFRyan

   Darbi Bauer - dkb17 - darbibauer

   Sarah Rosenfeld - smr17 - SarahRosenfeld

   Jacob Dienger - jwd17 - dienger

   Katie Skipper - kas18ff - KatherineSkipper

2) **Project Title and Description**

   Heimdall - Financial Recommendations

3) **Accomplishments and overall project status during this increment**

In this increment, the Google and Twitter API were combined. The issues with the Twitter API were solved by writing a script for command line rather than the IDE that was initially being used. This script utilizes Python 3, which is essential for this particular API. A Firebase database was created to store all of our application information. Stock information retrieval script was reworked to write directly to Firebase with no local storage of stock/option data. A database was created and linked to our app, and a function was created to access each individual stock information. Firebase's authentication option was also added to our app to allow for user creation, user login, and password changes.

4) **Challenges, changes in the plan and scope of the project and things that went wrong during this increment**

When trying to combine the Google and Twitter APIs, we realized that the Google implementation had accidentally been set up using Python 2.17. This proved to be a problem, as the Twitter API only runs on Python 3. The entire Google NLP had to be reconfigured, which proved to be a little difficult. Additionally, the database we originally implemented could not read in JSON files, meaning every entry would have to be put in by hand. As we have over 3,500 stock entries that need to be in the database, this was not realistic. Writing information to the database was only difficult until Aidan realized how to retrieve the correct API key. A threading issue in the information retrieval script was resolved as well, now a maximum of 16 threads pull data from Yahoo at any given time. During this iteration, we decided to implement our database in Firebase instead of MongoDB because Firebase has much easier Android interface to interact with.

5) **Team Member Contribution for this increment**

Aidan:

a) Contributed to Sections 3, 4, and 5
b) Nada.
c) Aided in sections 2 and 4
d) Rewrote data retrieval script to fix issue with multithreading. Write directly to Firebase with no local storage of data. Began testing computation within data retrieval script and writing computed values to database. This will be implemented fully in the next increment.

e) Video-taped self discussing contributions made during this iteration

Darbi:

a) Helped write Sections 3, 4, 5, and 6
b) Made use case diagrams and sequence diagrams in Sections 4 and 5, edited Section 1, wrote textual descriptions of each use case in Section 4
c) Contributed to Sections 3, 4, and 5
d) While the Twitter API implementation was being fixed, I worked with Jacob to build and work on the database that will be used in our app. Later, I realized it had been set up incorrectly, as it did not allow the importation of JSON files. I recreated the database in a way that would allow JSON files. After the Twitter API implementation had been completed, I worked with the combination of the Google API, and combined the scripts into one file that will be run on our server. This proved to be challenging, as I had accidentally set up the Google API using Python 2.17, and the Twitter API runs exclusively on Python 3 and above. After getting the two APIs to work in conjunction, I changed the source code to search Twitter for whatever term is passed in as a command line argument.
e) Edited and put together video, discussed individual contributions mentioned above, showed execution of combined Google and Twitter code, displayed contributions in database

Sarah:

a) Contributed to sections 3, 4, 5, and 6 regarding the integration of the twitter API and Google NLP, along with the fixing of the twitter API.
b) Contributed to 1, 5, 6, 7.
c) Contributed to sections 3 and 5.
d) I wrote the script for the twitter API to run the necessary commands, and output the search results into a text file - this was the format necessary for the Google NLP to use it as input. I continued to work on the twitter API in the IDE I had been using for another week before trying command line which is what ultimately worked - the API only works using Python3. Assisted Darbi with understanding and using my Twitter API code for her to use it in her Google NLP work.
e) Filmed video showing the functioning twitter script on it's own (separate from combined.py) to show functionality of my individual code.

Jacob:

a) Contributed to sections 3, 4, and 5.
b) Contributed to sections 2, 3, helped with Case Diagrams.
c) Contributed to sections 3, 4, and 5
d) Implemented authentication in the app to allow for unique user logins, as well as a user creation and password reset page. Connected app to database and created function to display all data from the 'calls' section on the home screen. Meshed Katie's screens into the Github Android Studio file so now we are able to work from one copy (it was previously set up incorrectly and we had two concurrent files with different namespaces).
e) Filmed video showing my changes to the application and their functionality.

Katie:

a) Updated accomplishments, challenges, and contributions.
b) Updated the use case diagram. Edited Functional and Non-Functional Requirements.
c) Edited Execution-based Functional Testing and Non-execution-based Testing.

d) Attempted to mesh both halves of the app created by Jacob and I. Made name changes necessary to make a smooth transition. My project was made initially with only my activities, so I attempted to copy paste Jacob's elements into my project. I changed my main activity from one of the app's sub-activities to the app's main activity (the login screen). I then added into my code some of the functions he created to reach his screens, and changed the names to work within my own framework. From there, I changed some of the added buttons I made as a temporary placeholder during the last iteration into new tabs, matching my design. I also removed some extra buttons that ended up being the same functionality as the tabs. Finally, I changed the design color from blue to green for a better look. Due to integration still not being complete, I have not been able to do much more.

e) Discuss the progress I made on trying to integrate both halves of the app. I show both how it looks, and some minor changes I made to the code.

6) Plans for the next increment

For the next increment, the Google and Twitter API combination script needs to be adapted to work with the information that will be provided from the database (e.g. each user's Twitter watchlist). The server which will host our application needs to be set up. Additionally, each of our individual components needs to be combined to work within our Android application. Finally, the stock prediction algorithm needs to be finalized and adapted to display correct information in our application. The next increment in terms of the app will include a design update to satisfy one of our non-functional requirements. It will also have each user's stock watchlist saved and displayed to their home screen, as well as functions implemented to change which stocks the user is watching. We also must create a screen that displays the long version of stock data - e.g. a graph of the past month, the calculations made on the backend server, etc. - that the user will use to get a better idea of the stock performance.

7) Link to video

https://drive.google.com/file/d/1CaNQWCACMkdx4pliH09t_cYZey_zUhgt/view?usp=sharing