# Phase 1

# 2/5/18-2/11/18

Art Fogiel – Create solution, leverage MVVM Lite to create layers, add initial UI although not completely working.

Michael Harrison – Fill out View Model implementations for LoginViewModel.

Kathleen Connell – Begin working on the parsing and requesting of the Stocks from Alpha Vantage.

Status: Initial UI complete, LoginViewModel nearly complete. Alpha vantage successfully queried but did not complete parsing of the stock.

Decision changes on design due to implementation issues:

1. Decided to simply use the text in the login text box when the user clicks create new and if is a unique name, creates and logs in. If not, posts an error message and takes back to login page. This is instead of making a completely new UI screen for creating a new user since we only need the text of the new user.
2. Alpha Vantage does not return the market cap of a stock. Instead we were able to download a master list of stock tickers and their market caps in csv form from Nasdaq. We use this list to initialize our list of stocks if the first time ever running the app. From then on we use the serialized list. We decided to use the market cap from there to populate the market cap value, but it will never update from day to day due to inaccessibility.

# Phase 2

# 2/12/18-2/18/18

Art Fogiel – Work more on UI and add test code to temporarily show created stocks until communication to alpha vantage is complete. Add serialization for users to disk.

Michael Harrison – Finish logging in implementation and start on applying filtering

Kathleen Connell – Continue working on the StockService model and acquire the stock from alpha vantage API.

Status: Logging in, saving users to disk, and loading from disk completed. UI to show settings and filtered stocks completed. Still need to do the watched stocks and applying the filter.

Decision changes on design due to implementation issues:

1. Decided to serialize in xml. Xml allows easily opening and viewing the file then modifying it in a text editor for testing. It did add slightly more complexity due to using interfaces and serializing multiple classes that are properties of a user. Serializing in binary would have been easier but then they would not be readable to humans just looking at the file.

Outside issues on progress:

1. One of our members was hospitalized with the Flu. We spread out their work between other members to keep up to date.

Future work:

Still need to parse an acquired stock from alpha vantage, serializing a stock, and implement the watching feature. We are still on schedule to complete implementation by next week but will be close.

# Phase 3

# 2/19/18-2/25/18

Art Fogiel – Cleanup communications with stocks. Cleanup UI

Michael Harrison – Finish applying the filter in the view model. Begin testing

Kathleen Connell – Parse Alpha Vantage output. Update stocks from alpha vantage.

Status: We ran into many issues this week but in the end, all implementation is complete. We still need to perform final testing and debugging as well as document finalizing.

Decision changes on design due to implementation issues:

1. We successfully parsed the stocks from Alpha Vantage. We ran into a big issue though…. Alpha Vantage was not reliable at all. According to their documentation, we should be able to request once a second. We were requesting a batch of 100 stocks per request, once a second. We only ever received the first batch; all next requests were returned with an error saying we were asking too often. We tried scaling down to once every 5 seconds but got the same message after the first. While searching online it appears their documentation is inconsistent. One user found that if you made more than 30 requests a minute it would reject you for a minute. We are requesting one every 5 seconds (12 times a minute) but still got rejected after one request. I think this was due to using the batch request. It would be unreasonable to only query 30 stocks per minute when we could have 6000 + stocks. Also, we could not query alpha to get a list of supported stocks. If we requested one that they did not support, the call would take several seconds before returning an error.
2. Due to the above problem, Michael found another free API which seemed to work much better. IEXtrading.com. We originally chose Alpha Vantage due to a large amount of people saying it worked well after Yahoo closed their API. IEXTrading appears to have much higher reliability and speed. It also supported asking for the list of supported stocks. We changed to request the list from them instead of the csv’s available on Nasdaq.com. We also changed our queries to use them to update our stock values. It required creating a new class called Quote since the query returned a JSON object we needed to deserialize. They also only allow batch quotes on up to 100 stocks, so we divided the master list up into groups of 100 and iterated over them to request stock prices.
3. IEXtrading reserves the right to block you based on IP if you overuse their API. To avoid this, we query all the stocks on startup, but then only query once a minute if the current time is M-F from 9:30 am to 4pm eastern. It still to be determined if this is sufficient as we need more testing during stock market hours.

Future work:

We still need to finish testing and debugging as well as negative path testing on network issues. We also need to update our user guide with final screens as well pull all our documents together.

# Final Submission Week

2/26 to 3/2

Art Fogiel, Michael Harrison – Final cleanup and ensure all test cases pass.

Kathleen Connell- Gather all documents together for final report.

Status: Complete!