ECE 421 Project 1
Stock Project Monitor

Arun Woosaree
Alexander Rostron
Jacob Reckhard

February 3, 2020

1 Rationale

Since this software interacts with an external entity with API limitations, we had to make some adjustments to the specifications. Since the API limits the number of calls per minute, thus, the software is unable to make enough API calls in under a minute to gather the required information on 10 stocks. So, we thought it more correct to stall the program until it received information on all 10 stocks instead of proceeding with information on a limited number of stocks. The runtime of this software is primarily bounded by the API limitations, but does produce the correct result (assuming stock prices haven't changed during the runtime of the program).

In the event that there are two stocks with the highest price we arbitrarily pick one of the stocks and return that.

The functional features of Java we used were effectivley, maps, reduces, filters, and lambdas. Thus the pickShareFunctional method ends up effectively being a one-liner with no local variables. had to make some adjustments to the specifications. Since the API limits the number of calls per minute, thus, the software is unable to make enough API calls in under a minute to gather the required information on 10 stocks. So, we thought it more correct to stall the program until it reviced information on all 10 stocks instead of proceeding with information on a limited number of stocks. The runtime of this software is primarily bounded by the API limitations, but does produce the correct result (assuming stock prices havent changed during the runtime of the program).

In the event that there are two stocks with the highest price under $500 we arbitrarily pick one of the stocks and return that.

In the event that the api does not give a response with the price, it is assumed that the price of the stock is 0. This can happen when an invalid stock is passed to the function, and can also happen in the event that the API is down or returns an error response.

2 Testing

Unit tests were created to verify two simple scenarios. The first test was for the normal case, where all of the stock prices returned are different, and we expect the function to return the stock with the highest price under $500. For the second test, we assert that in the case where there there is more than one stock under $500 with the maximum value, any one of those stocks is returned.

3 Defects

The biggest fault of the program is that there is an API key limitation in that it can only get 5 stock prices every minute. This is problematic, because there is 10 stocks the program is to check. That means it will take at least a minute to run regardless of how the code is designed. This delay greatly overshadows any other aspect of the code.