1/12/2019 Sam Judkis

Edmund Dea

**CS 340 Final Project Specs**

1. Project Theme
   1. Stock website
      1. Front-End Design Inspiration - <http://www.tradingview.com>
      2. Real-Time Stock Quotes using API
         1. <https://github.com/iexg/IEX-API/issues>
         2. <https://www.alphavantage.co/>
         3. <http://finance.google.com>
         4. <http://finance.yahoo.com>
         5. <https://stackoverflow.com/questions/10040954/alternative-to-google-finance-api>
2. Querying Stock Data
   1. Use API to query stock data via JSON/etc for most recent price
   2. INSERT INTO <table> VALUES <data>
   3. Then the current logged in user can query the primary stock data table and display stock data.
3. Watchlist
   1. Display a list of stocks in a portfolio.
   2. User can
      1. ADD various stocks to add to their portfolio
      2. DELETE stocks from their portfolio
      3. SORT stock data by ticker/percentage change
      4. UPDATE the portfolio’s stock tickers
      5. SEARCH for stocks within the portfolio by ticker ???
         1. Piazza -
            1. “Every table should be used in at least one SELECT query. For the SELECT queries, it is fine to just display the content of the tables, but your website needs to also have the ability to search using text or filter using a dynamically populated list of properties. This search/filter functionality should be present for at least one entity. It is generally not appropriate to have only a single query that joins all tables and displays them.”
            2. Do they want us to implement a SELECT statement for each entity and/or do they want us to to add search/filter functionality associated with each table?
   3. If a stock is removed from the stock exchange, then handle the case of removing the many-to-one relationship by settings all user portfolios containing that stock to NULL. Then, delete that stock from the appropriate table.
   4. In a many-to-many relationship, to remove a relationship one would need to delete a row from a table.
   5. Displays Data
      1. Stock ticker
      2. Current Price
      3. Percentage Change relative to the day’s opening price
4. Users
   1. User can login
   2. User’s portfolio loads after logging in
   3. (optional) Implement SSL connection
5. Optional Features
   1. Displays stock data in a chart
      1. Shows OHLC (Open-High-Low-Close)
   2. User can add purchase price and # of shared purchased. Then user can UPDATE their purchase price and # of shared purchased.