Team Verbose Spoon — Design Doc

Rodda R. John, Jack Schluger, Bayle Smith-Salzberg, Jessica Titensky, Jordan Yaqoob

Introduction:

For our project, we will create a stock market simulator. Users will be able to log on, and buy and sell stocks. We will incorporate stock api's, and news api's to show headlines relevant to user's stocks.

Task Delegation:

Project Manager: Jack
Database Interaction: Bayle
API Interaction: Rodda
Flask: Jordan
Bootstrap: Jessica

Front End:

- Show graphs of stocks
- Show history of stock if click
  - Have ability to buy/sell stock
- Show profile with how much money and shares of stocks they have

Site Map:

- Home Page: @app.route("/")
  - One form for login, one form for register.
  - This is where all pages will redirect to if they are accessed by someone who is not logged in.
- Feed: @app.route("/feed")
  - Show basic information about all stocks.
  - Show more detailed information about stocks that the user has shares of, or has marked as a favorite.
  - Uses "markitondemand" API
- Stock Page: @app.route("/stock")
  - Show all information about one selected stock
  - Relevant news related to stock
  - Uses nasdaqdod and ft APIs

- ○ Get info about if stock share was bought/sold
- User Page: @app.route("/me")
  - ○ Show all stocks that a user owns, and some information about them.
  - ○ Show other funds that a user has.
  - ○ Show gains/losses and portfolio info

Back End:

- app.py
  - ○ Contains all the routes
- Utils:
  - ○ Initialize.py:
    - ■ Initialize the database and create all the tables
    - ■ Write secret key to utils/key
    - ■ Should be run ONCE, the first time one wants to run the flask app after cloning the repo
  - ○ api.py
    - ■ Contains functions for getting information from each api
  - ○ dbmanager.py
    - ■ Contains functions for all interactions with database.db
  - ○ accountManager.py
    - ■ Handles user creation, and logging in and out.

- Templates:
  - ○ login/register
  - ○ Feed
  - ○ Specific info on one stock
  - ○ profile

Database Schema:

In database.db, we will have 2 tables:
1. users

username	password	Display name/id/something like that?	Available Funds	User ID
TEXT	TEXT	TEXT	FLOAT	INTEGER

2. Owned stocks

username	stock name	shares

TEXT	TEXT (stock abbr)	INTEGER

API's we will use:

http://dev.markitondemand.com/MODApis/#doc_lookup   —confirmed USE
- Includes API for making cool charts
  - Has historical data for each stock which can be graphed

https://developer.ft.com/docs -- financial times   —confirmed
> https://developer.ft.com/docs/api_v1_reference/content_items/ -- dict items inside ft

http://usfundamentals.com/ -- free for stocks   —confirmed

https://www.quandl.com/ -- real time stocks   —confirmed