Team WINY: Black Box Unit Testing

Irwin Li (izl2000)
Wanlin Xie (wx2161)
Yiming Sun (ys2832)
Nancy Xu (nx2131)

Responsibilities
Irwin and Wanlin: Login/Logout + testing, create account + testing, formatting HTML, added continuous integration, Google Cloud, Black Box Unit Test Cases
Nancy and Yiming: Add stock + testing, search + testing, Class Diagrams, added graph, retrieved stock data from API

Git repository
https://github.com/wanlinxie/ASE-marketgame
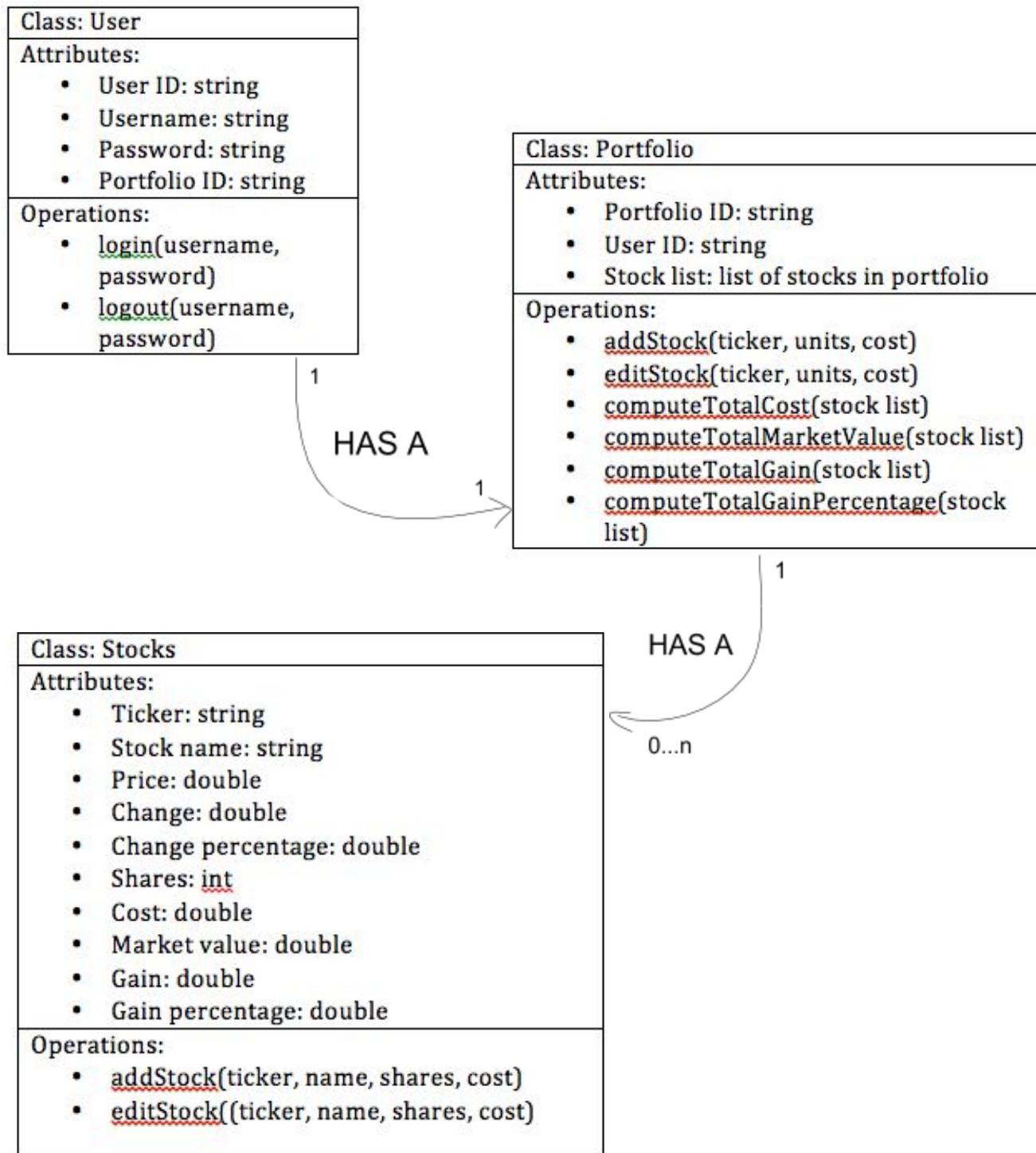
Testing Tool
We have all our tests in our test.py file that can be invoked using pytest.

Black Box Unit Test Cases
1. <u>User can login successfully</u>: Test both valid email + password combinations (already created accounts) and invalid email + password combinations (not created accounts). Email must always be in the "_@_" format or system should tell user to enter a real email address. Invalid email examples would be "hi", "1234" and boundary conditions would be: "hi@","@12","@". Any invalid email + password should show user a red error message saying "Invalid username or password".

2. <u>User can logout successfully</u>: The logout button should be displayed on every page that the user sees after they login so that they always have the option to logout. Once they click the button, they should be redirected to the login page. Check to see that this happens.

3. <u>User can create account</u>: User should not be able to sign up with an account with an email address that is not in a "_@_" format. Cases such as "hi", "1234", hi@", "@12", and "@" should not work and the user should be told to enter a real email address. If the passwords that the user enters do not match, the user should see the error on the page. Otherwise, if the user enters an email in a valid form and enters two passwords that match, they will create an account successfully.

4. <u>User can search for stock by ticker</u>: If user searches for a stock with an invalid ticker, they will see an error on the page. If the stock ticker is valid, they will be able to see the stock page with it's name, current price, graph, and be able to add shares of that stock to their portfolio.

5. <u>User can add stock to the portfolio</u>: Once the user specifies the number of shares and cost per share, they will see the newly added stock to their portfolio. If they click "Submit" without entered the necessary fields, they will be shown an error.

Class Diagrams

Class: User
Attributes:
- User ID: string
- Username: string
- Password: string
- Portfolio ID: string

Operations:
- login(username, password)
- logout(username, password)

1

HAS A

1

Class: Portfolio
Attributes:
- Portfolio ID: string
- User ID: string
- Stock list: list of stocks in portfolio

Operations:
- addStock(ticker, units, cost)
- editStock(ticker, units, cost)
- computeTotalCost(stock list)
- computeTotalMarketValue(stock list)
- computeTotalGain(stock list)
- computeTotalGainPercentage(stock list)

1

HAS A

0...n

Class: Stocks
Attributes:
- Ticker: string
- Stock name: string
- Price: double
- Change: double
- Change percentage: double
- Shares: int
- Cost: double
- Market value: double
- Gain: double
- Gain percentage: double

Operations:
- addStock(ticker, name, shares, cost)
- editStock((ticker, name, shares, cost)

Demo
In our demo, we first introduced the purpose of our web application before going through the typical work-flow of a general user. We first created an account for a new user, then showed the initial views of the pages for the user's stock portfolio. Afterwards, we searched up stocks by specific stock tickers to show the stock page. The search input bar has AJAX autocomplete for stocks. Each stock page contains information about the closing price, opening price, volume, and historical data. From the stock page, a user can purchase and sell shares of stocks. Purchasing and selling shares will update the view on the portfolio page. Once the number of shares reaches zero, the stock will be erased from the portfolio. We concluded the demo by logging out the user and showing our unit tests.

There were no problems that arose during the demo. We were recommended to use the coveralls module for testing to show testing coverage and to show a list of top performing stocks.