Portfolio Rebalancing App Documentation

Cheng Ji
c7ji@uwaterloo.ca

Summary

The goal for this Undergraduate Research Assistantship was to implement new features on the Android App. During my URA, I implemented portfolio rebalancing strategies such as CPPI, Covered Call Writing, and Stop Loss strategy into the Android App, as well as some user interface improvements. The final product can run both real-time and historical simulations with different kinds of rebalancing strategies using the closing price of each day.

App Design and Stock Data

Please refer to the documentation written by Yuwei Xu.

Calculations

Please refer to the *Portfolio Rebalancing Strategy.pdf.*
Sample calculations of every day balances are shown at the last few pages of this documentation.

User Interface

The following refers to the code in the Android App

i) **MainActivity**
This is the Activity that is started when the app first opens. There is a ViewPager that contains both the StockActivityFragment and the SimulationFragment.

ii) **StockActivityFragment**
This Fragment is under the “Portfolio” tab and refers to the main UI which allows the user to add stocks to the portfolio and create new simulations. When the “Add Stock” button is pressed, the contexts of the EditText are first verified to be a legit stock symbol. Then, the UI is populated with the stock. When the “Add Simulation” button is pressed, a new SimulationActivity opens.

iii) **SimulationFragment**

This Fragment is under the “Simulations” tab and refers to the simulations that the user has run. Clicking on one of the simulations starts a SimulationActivity and displays the appropriate data.

iv) **SimulationActivity**

This refers to the separate Activity that displays the status of the portfolio. This Activity is created when a user creates a new simulation or views an old simulation. In this Activity, two charts are populated, one displaying the progress of the portfolio stocks according to market data and another displaying the progress of the portfolio balance.

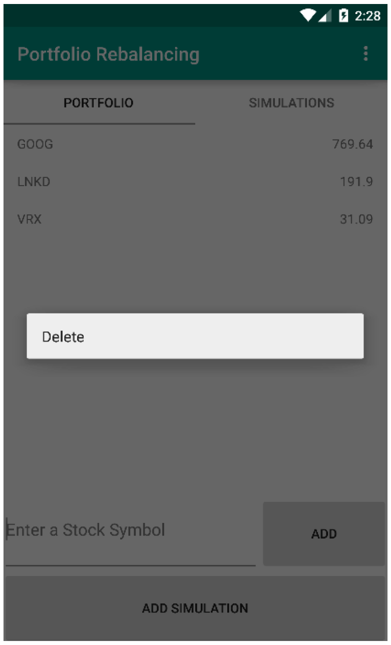
The charts are generated within separate Fragments using the MPAndroidChart library (https://github.com/PhilJay/MPAndroidChart).

v) **PortfolioInfoActivity**

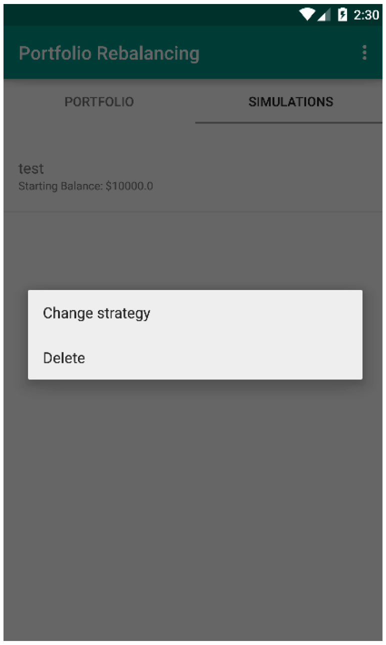
This refers to a child activity of SimulationActivity. This activity is created when user long-clicks on the second graph (portfolio balance chart) in SimulationActivity. In this activity, the portfolio balance chart is displayed, followed by some important information for that simulation (e.g. simulation type, CPPI floor/multiplier, option price, etc).

New UI features

- Ability for users to delete stocks from the current list of stocks on long click
- Ability for users to delete simulations from the current list of simulations on long click

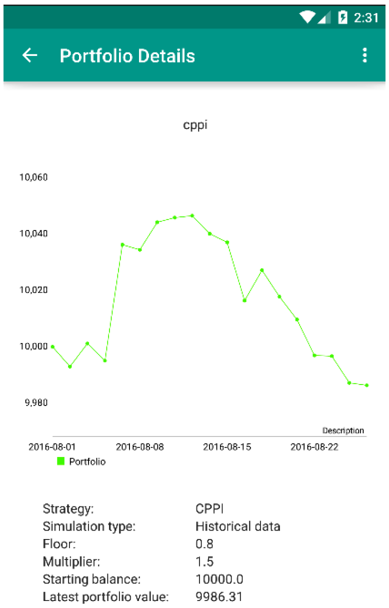


GOOG	769.64
LNKD	191.9
VRX	31.09



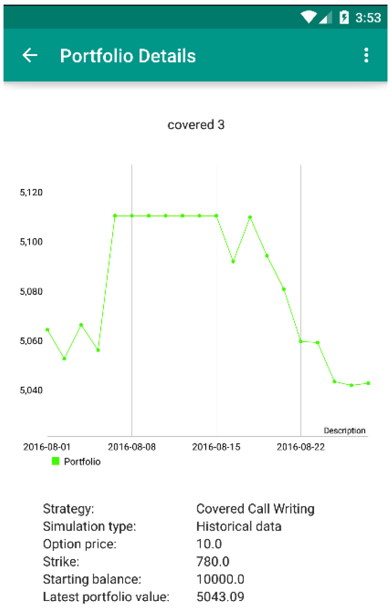
test
Starting Balance: $10000.0

- Ability to see detailed portfolio settings on long lick on the simulation graph



cppi

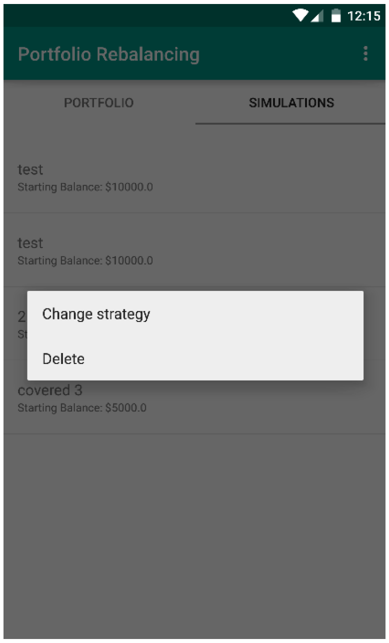
Strategy:	CPPI
Simulation type:	Historical data
Floor:	0.8
Multiplier:	1.5
Starting balance:	10000.0
Latest portfolio value:	9986.31



covered 3

Strategy:	Covered Call Writing
Simulation type:	Historical data
Option price:	10.0
Strike:	780.0
Starting balance:	10000.0
Latest portfolio value:	5043.09

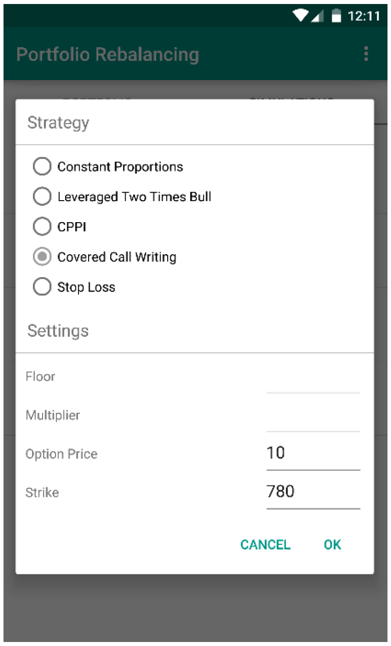
- Ability to change strategy in an on-going simulation



test
Starting Balance: $10000.0

test
Starting Balance: $10000.0

covered 3
Starting Balance: $5000.0



Strategy

○ Constant Proportions
○ Leveraged Two Times Bull
○ CPPI
● Covered Call Writing
○ Stop Loss

Settings

Floor	
Multiplier	
Option Price	10
Strike	780

Future Work

i) Handle more user input errors and ensure there is a user-friendly response

ii) Make an impressive looking UI

iii) Pull daily option chain from some finance API to set the option price automatically instead of letting user set the option price

   - Here’s the document of accessing and parsing the data using Google Stock Options API: http://www.jarloo.com/google-stock-options-api/ and the associated Github Repo: https://github.com/kelias/GoogleOptionsApi
   - This is implemented in C#. See if we can implement it using Java and add it to the Portfolio Rebalancing App.