University of Pisa
School of Engineering

Large Scale and Multi-Structured Databases

StockSim: Stock Portfolio Simulator

Supervisor
Prof. Pietro Ducange

Students
Marco Pinna
Rambod Rahmani
Yuri Mazzuoli

January 2, 2021

Contents

Part I

Documentation

Chapter 1

Introduction

StockSim is a Java application which, as main feature, allows users to simulate stock market portfolios. The StockSim application is composed by two main programs:

- **StockSim Server**: supposed to be running 24/7 to ensure historical data is always up-to-date;
- **StockSim Client**: can be launched in either `admin` or `user` mode.

The StockSim Server is not thought to be distributed to end users, whereas the StockSim Client can be used by both administrators and normal users. The choice was made to provide the same program to both administrators and normal users with two different running modes. Administrators can add new ticker symbols, new administrator accounts, delete both administrator and normal user accounts. Normal users have access to stocks and ETFs historical data, day by day, starting from 2010. They can create their own stock portfolios, run simulations a visualize the resulting statistics.

Before continuing with what follows, the following terms should be clarified:

- the **stock market** is any exchange that allows people to buy and sell stocks and companies to issue stocks; a stock represents the company's equity, and shares are pieces of the company;
- a collection of investments owned by an investor makes up his or her **portfolio**; you can have as few as one stock in a portfolio, but you can also own an infinite amount of stocks or other securities;
- a **stock symbol** is a one- to four-character alphabetic root symbol that represents a publicly traded company on a stock exchange; Apple's stock symbol is AAPL, while Walmart's is WMT;
- the NYSE and Nasdaq are open from Monday through Friday 9:30 A.M. to 4:00 P.M. (eastern time);

- the NYSE and Nasdaq close at 4 P.M., with after-hours trading continuing until 8 P.M.; the close simply refers to the time at which a stock exchange closes to trading;
- trading stocks after normal market hours through an electronic market, typically between 4:05 and 8:00 P.M., is **after-hours trading**;
- the **high** is the highest price at which a stock traded during a period;
- the **low** is the lowest price of the period;
- **open** means the price at which a stock started trading when the opening bell rang; it can be the same as where the stock closed the night before, but not always; sometimes events such as company earnings reports that happen in after-hours trading can alter a stock's price overnight;
- **close** refers to the price of an individual stock when the stock exchange closed shop for the day; it represents the last buy-sell order executed between two traders; in many cases, this occurs in the final seconds of the trading day;
- the **adjusted closing price** amends a stock's closing price to reflect that stock's value after accounting for any corporate actions; a stock's price is typically affected by some corporate actions, such as stock splits, dividends, and rights offerings; adjustments allow investors to obtain an accurate record of the stock's performance;
- **volume** is the total number of shares traded in a security over a period; every time buyers and sellers exchange shares, the amount gets added to the period's total volume.

Chapter 2

Actors and requirements

Blablabla.

2.1 Actors

Something about requirements in general.

2.2 Requirements

2.2.1 Functional requirements

Something about functional requirements.

2.2.2 Non-functional requirements

Something about non-functional requirements.

Chapter 3

UML diagrams

Blablabla.

3.1 Use Case diagram

Something about functional requirements.

3.2 Class diagram

Something about functional requirements.

Chapter 4

Database

Something in general about the dataset and then about the choice of the DB architectures we choose (e.g. "We decided to use a documentDB because blablabla and a column DB because financial analytics on these volumes of data are performed

4.1 Dataset

4.1.1 Yahoo! Finance

4.1.2 NasdaqTrader

4.2 MongoDB

4.2.1 Aggregations

4.2.2 Indexes

4.3 Apache Cassandra

4.3.1 Aggregations

4.3.2 Indexes

4.4 Sharding and Replicas

4.5 Apache Cassandra vs MongoDB

Chapter 5

Software architecture

5.1 Maven Multi-Module Structure

5.1.1 Library

5.1.2 Server

5.1.3 Client

5.2 Apache Maven Assembly Plugin

The Assembly Plugin for Maven enables developers to combine project output into a single distributable archive that also contains dependencies, modules, site documentation, and other files.

Chapter 6

Conclusions

Content.

Part II

User Manual

Chapter 7

StockSim Server Manual

For an application dealing with the stock market, it is essential to always provide up-to-date, consistent and reliable data. This is the purpose of the **StockSim Server** program. It is not intended to be distributed to end users. It is thought to be running 24/7.

The StockSim Server has two different startup modes: the first one

```
$ java -jar Server.jar

Welcome to the StockSim Server.

DATA CONSISTENCY CHECK SUCCESS. PROCEEDING WITH UPDATE.

Updating historical data for: IRBT.
Last update date: 2020-12-30.
Days since last update: 2.
Historical data updated for IRBT. Moving on.

...

Updating historical data for: EWU.
Last update date: 2020-12-28.
Days since last update: 4.
Historical data updated for EWU. Moving on.

...
```

which executes the `historical data update` procedure right after startup. Whereas, the second startup mode can be triggered using the `--no-update` command line argument:

```
$ java -jar Server.jar --no-update

Welcome to the StockSim Server.

Available Commands:
```

```
6  status        check databases status.
7  update        update databases historical data.
8  quit          quit Stocksim server.
9  >
10 [UPDATER THREAD] Current New York time: 2021-01-02T06:29-[America/New_York]
11 [UPDATER THREAD] Going to sleep for 14 hours before next update.
12 > update
13
14 DATA CONSISTENCY CHECK SUCCESS. PROCEEDING WITH UPDATE.
15
16 Updating historical data for: IRBT.
17 Last update date: 2020-12-30.
18 Days since last update: 2.
19 Historical data updated for IRBT. Moving on.
20
21 ...
22
23 Updating historical data for: EWU.
24 Last update date: 2020-12-28.
25 Days since last update: 4.
26 Historical data updated for EWU. Moving on.
27
28 ...
```

in this mode, no update is executed right after startup, the main menu is shown and the user can decide the action to be performed.

In the previously shown examples, something we should pay attention to, are the following

- `DATA CONSISTENCY CHECK SUCCESS. PROCEEDING WITH UPDATE.`
- `the sotcksim server automatically detects the last update date, the number of days since the last update and fetches the required data using Yahoo Finance for EACH and EVERY ticker symbol.`
- `Historical Data Update logs.`
- `[UPDATER THREAD]`

Chapter 8

StockSim Client Manual

Content.