Create a trading application in Java

Trading is an activity in which more and more individuals are interesting and that has truly democratized.

Today, brokers offer very objectively reasonable transaction costs. It allows investors to aspire to make their first steps in trading by starting with a investment capital much more modest than before. For example, with € 4,000 and a clear strategy, one can start trading (obviously on positions of size accordingly).

In addition, the tools and information available to carry out this activity are now numerous.
If trading platforms such as MetaTrader, ProRealTime or simply the platform provided by your broker allow you to operate in the financial markets efficiently and nothing seems to miss you for trading, keep this recipe that works.

Now, if you feel in the tools provided, you are missing some features as you consider essential to trade and backtest and therefore infine be more serene in your investment decisions, using a custom tool can bring a great advantage to you.
A customized and well done trading application will significantly outperform any well done generic trading application .

Why ?

The reason is simple: trading is neither an exact science nor a closed concept. Therefore, generic trading platforms are obliged to provide an impressive number of opportunities to stick on the needs for the largest number. Conversely, a personal trading tool will only focus on the points that are important to the user of the tool. What allows to optimize strongly these points in terms of contents, added value, user experience and exexcution time.

The financial institutions (banks, investment banks, investment fund, etc …) involved in the financial markets know it well.

Now let's be clear about the basic trading application that will be developed in this course.

The application that will be presented is not:
– A complete trading Java framework for developers
– A complete trading application for end users.

The application that will be presented is :
– A course to understand how to start the development of a trading application
– Clear explanations of each stage of development
– A bootstrap for building trading applications running under Java
– The implementation of an iterative development approach. This course is somehow the iteration one, which covers the basic features essential to a trading application.

The following iterations, under your responsibility, could provide places for enriching the functionality of iteration one, as well as adding new ones.
– A course to understand the use of some design patterns (Builder, wrapper, defensive copy, singleton, facade, etc …)

I speak about trading application but reality is multi-shaped.

According to the density and the possibilities offered by the application, we can distinguish :

- trading box tools offering some features for trading
- complete trading platforms offering a wide range of features for trading

Building a trading application covers an impressive spectrum of possibilities. Simple and common features such as:

- portfolio monitoring
- graphic visualization of the price of securities (share, index, commodity, etc …)
- display of technical indicators for securities
- security quotations download

And more complex features such as:

- signals detection
- dividends management
- strategies backtesting
- looking for opportunity in real time
- performing automated orders (simple algorithmic, high frequency, etc …).
- financial monitoring of opened positions
- strategic monitoring of opened position

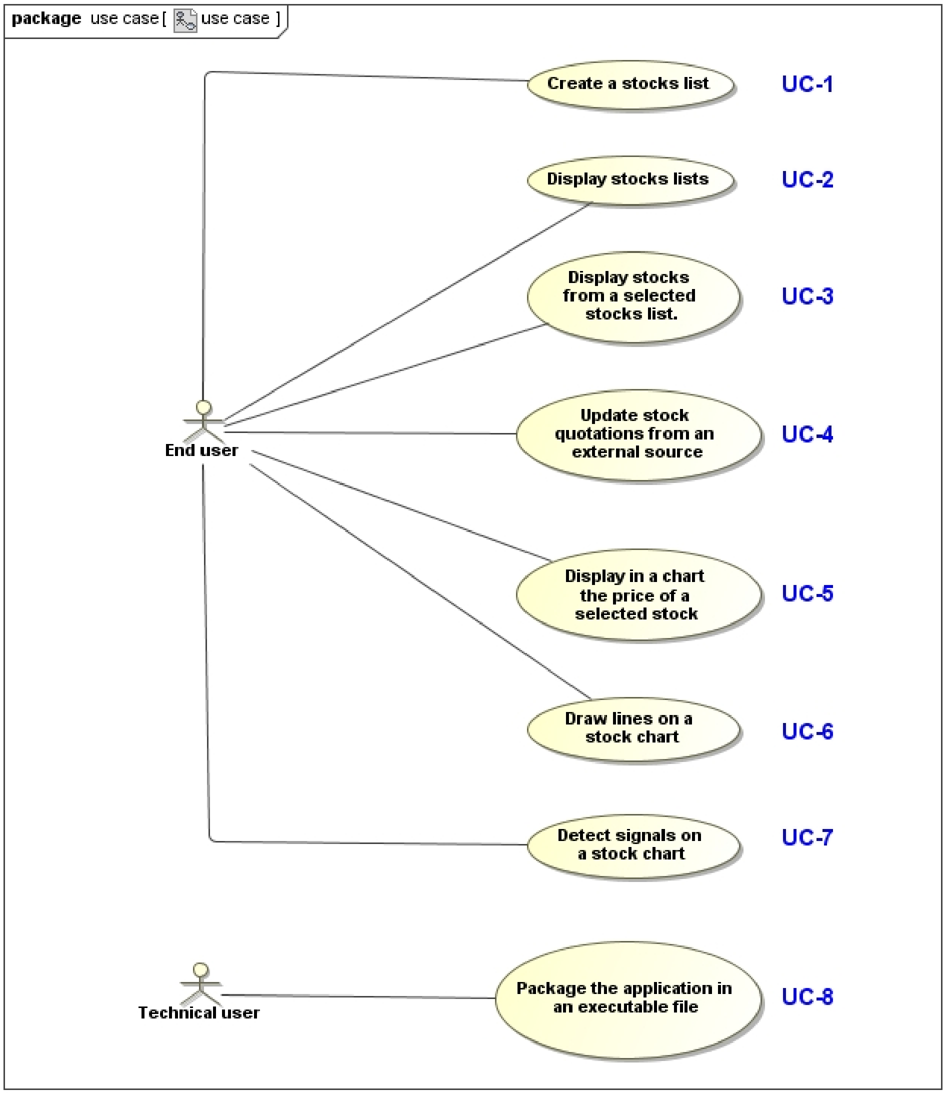
This course is not intended to teach you how to trade.
It focuses on the application part in terms of components and their interaction with each other.

Regarding the nature of financial values that we deal, we limit ourselves to shares (stocks).
Of course, there are many other financial instruments: bonds, indices, warrants, cfd, forex, etc …
Each of these instruments presents specificities, so we can not decently treat them completely uniformly in an application.
And the goal of this course is primarily educational, I have to make a choice on the instrument to be used for the demonstration.
The scope of the course being fixed, I can present use cases that we are going to study and to implement:

- UC-1 Create a stocks list
- UC-2 Display stocks lists
- UC-3 Display stocks from a selected stocks list.
- UC-4 Update stock quotations from an external source
- UC-5 Display in a chart the price of a selected stock

- UC-6 Draw lines on a stock chart
- UC-7 Detect signals on a stock chart
- UC-8 Package the application in an executable file

Below is a visual representation in UML:


package use case [use case]
Create a stocks list
UC-1
Display stocks lists
UC-2
Display stocks from a selected stocks list.
UC-3
End user
Update stock quotations from an external source
UC-4
Display in a chart the price of a selected stock
UC-5
Draw lines on a stock chart
UC-6
Detect signals on a stock chart
UC-7
Technical user
Package the application in an executable file
UC-8

This course segmented into 15 parts, will allow you to see the path leading to the construction of a very basic trading application. A minimalist toolbox.
It's up to you to enrich it to meet best your needs. And that's the the biggest of the work both in terms of the financial strategy and software design.
But remember one thing: making a custom application is not to resume all the features of existing trading platforms.
You will spend many years and that has, moreover, no interest as an application already offers this service and like most users, you will only use a tiny part of this application.
Enriching your platform must always be based on the following rule: build what is really necessary and create the added value where you want it.

The course is divided into several sub-parts:

Do the functional and technical study of the bootstrap application

- Technical Choices (1/15)
- Lay out the main GUI containing required functionalities (2/15)
- Design the conceptual model (3/15)
- Define applicative layers (4/15)

Build the bootstrap application

- Define the developpement methodology : iterative development and TDD
- Go from model domain to entity model design
- Define where stored data
- Create a stocks list (UC-1)
- Display stocks lists (UC2)
- Display stocks from a selected stocks list (UC3)
- Update stock quotations from an external source (UC4) (in progress)

Go further

- Extension of the bootstrap application