

Thalia Project Report

Team Charlie

**Martti Aukia, Albert Boehm, Arthur-Louis Heath,
George Stoian, Daniel Joffe, Weronika Kakavou,
Marcell Veiner**



University of Aberdeen
Sunday 26th January, 2020

Acknowledgement

TO DO: Thank stackoverflow and everyone else here.

Contents

1	Introduction	3
1.1	Project Overview	3
1.2	Motivation/Rationale	3
1.3	Project management strategy	3
2	Background and Competitors	4
2.1	Portfolio Visualizer	4
2.2	Other Competing Software	5
2.3	Summary	7
3	Requirements	8
3.1	Functional Requirements	8
3.2	Non-functional Requirements	10
4	Design	12
5	Coding and Integration	13
5.1	Web Framework	13
6	Testing and Evaluation	14
7	Conclusions and further work	15
8	Appendices	16
8.1	Appendix A - User Manual	17
8.2	Appendix B - Maintenance Manual	18
8.3	Appendix C - Glossary	19
	References	21

1 Introduction

1.1 Project Overview

This project is to create a portfolio backtesting software, which enables creating custom portfolios and measuring their performance with different backtesting functions. The user can pick from a variety of assets, some of which are Equities, Fixed Income, Currencies, Commodities and Cryptocurrencies. Then a quick visualization is performed for the allocated assets with risk and performance metrics.

1.2 Motivation/Rationale

Since retail investing is a growing market, our target audience consists of individual investors, who instead of seeking the full package that comes with financial advising, would like to take over the wheel and assess the viability of their investment strategies themselves. As retail investors are non-professionals and invest comparatively small amounts, with financial advice services being non affordable for those individual clients, we wanted to create a product that would not only be more affordable for small retail investors, but would also include a variety of international assets, which currently most backtesting software doesn't include.

1.3 Project management strategy

Team members are:

- Martti Aukia
- Arthur-Louis Heath
- Albert Boehm
- Marcell Veiner
- George Stoian
- Daniel Joffe
- Weronika Kakavou

The team held regular meetings, which consist of weekly meetings between the whole team and the project guide Dr Nigel Beachem, between the project leader - Martti Aukia and Dr Ernesto Compatangelo. During the analysis stage meetings took place weekly and were aimed to discuss ideas and requirements. During the implementation stage the team held meeting at least once per week, some of which were aimed to discuss the design for the tool and others were used as coding sessions.

2 Background and Competitors

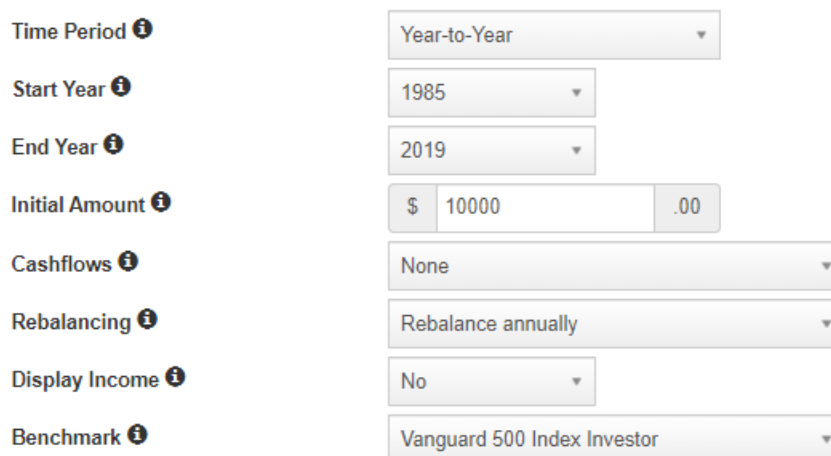
As a result of our initial competitor analysis, we may group the competing software into two categories. For a comprehensive list of available backtesters please see [1]. The first consists various third-party trading software, such as Fidelity, MetaTrader and NinjaTrader that offer backtesting features as well. Although it may be beneficial to consider some trading features provided by the members of this group, these would only be part of a future development process, and not in the scope of this course.

The second category consists pieces of software that do not offer trading as a service, and solely focus on backtesting. Thus, for now, we shall only consider the feature set provided by the second category. In the following paragraphs we will consider some of the design decisions made by competing software, together with the brief analysis and conclusion on each feature.

2.1 Portfolio Visualizer

Our main competitor of the second category is an online backtesting tool named Portfolio Visualizer[2]. During the inception phase of development we heavily relied on this website for writing the requirement analysis. Let us now briefly dissect what it has to offer.

Upon opening the website we are greeted with a brief description of the domain, together with the following input form:



Time Period ⓘ	Year-to-Year ▼
Start Year ⓘ	1985 ▼
End Year ⓘ	2019 ▼
Initial Amount ⓘ	\$ 10000 .00
Cashflows ⓘ	None ▼
Rebalancing ⓘ	Rebalance annually ▼
Display Income ⓘ	No ▼
Benchmark ⓘ	Vanguard 500 Index Investor ▼

Figure 1: Portfolio Visualizer - Input

As we can already see, the key elements of portfolio analysis are provided. Considering the functionality of our prototype as a baseline (that is testing an allocation of assets with a fixed initial investment on a fixed time period), we see that in addition users are able to do the following set of features:

- Set the endpoints of the investment period, up to months.
- Set the initial investment.
- Specify a regular cashflow and its frequency.
- Select a rebalancing strategy.
- Select a benchmark strategy for comparison.
- Compare multiple strategies at the same time.
- Adjust to inflation.
- Select from a set of lazy portfolios.

- Calculate additional metrics.
- Export the results to PDF, Excel, or save link.

At first it may seem as if Portfolio Visualizer meets all the requirements needed for a financial backtester, and indeed our main criticism is regarding the UI, responsiveness and user experience, and is a result of the initial user testing.

The UI design is simplistic and has a non-commercial, bare-bones look, and although this was appreciated while testing the system, it certainly does not improve the user experience. The input, mostly using dropdown menus is straightforward to use, except for the selection of Assets, which we will discuss briefly at the end of this section.

Moreover, users will want to fine-tune their investment strategies, by changing their allocations frequently. The only means to do this using Portfolio Visualizer is to scroll to the top, change the input and send it again. Our goal is to design a more responsive and dynamic system, to ease this procedure.

The website also has user accounts, these accounts however offer little to no extra features, which further decreases the overall user experience. Finally, as a last remark, which holds for most of the backtesting tools we have tried, the website is heavily US biased, and so the selection of asset classes is limited. Furthermore, we have no means of changing the currency, which is needless to say, suboptimal.

2.2 Other Competing Software

Fortunately the rest of the alternatives are of a significantly lower quality. In the rest of this section we will briefly consider a few design decisions made by these websites.

Tree-like Asset Selection

One of the tricky design decision in building the system, is the following:

What is the most intuitive way for selecting a portfolio item?

Where a portfolio item could be: an equity, ETF index, a commodity, bond or stock. Within these classes we have more options to choose from. Many backtesters opted for a search bar, which is a sensible approach, but poor in practice. Many portfolio items are named similarly, and for a new user it is a barrier, as they might not know what is available.

One of the better option is what ETFReplay [3] has implemented, which is a tree-like selection form, shown below.



Figure 2: ETFReplay - Tree-like Structure

We feel this was the most intuitive to use, and will thus chase a similar approach. Our only criticism was that it opens in a new window, which we would like to avoid.

Tiles

It may be worth briefly discussing an example of a backtester with a good layout design. We found the simplistic and tiled design of Backtest Curvo [4] a good choice as it gives the website an overall modern and fresh look, whereas most backtesters looked old and out of date. Our goal is to achieve a layout similar to this.

Simple Logic

In our whitepaper, we briefly discussed implementing a simple scripting language allowing users to simulate dynamic trading strategies. As mentioned, this would not be in the scope of the course, but for inspiration we can have a look at the approach of Stockbacktest [5].

Signals (Need at least one buy or short signal)
 Buy When:

Signal 1:	Upper Bollinger Band (days devs) ▼	Crosses Below ▼	Percentage % ▼	AND ▼
	20 2	none	70	
Signal 2:	Close Price ▼	Is At Least % Above (%) ▼	Percentage % ▼	
	none	15	10	

Figure 3: Stockbacktest - Simple Logic

Although not intuitive at all, given its presentation, it shows this feature is also possible. We believe that after nailing down how exactly the user would create the rules, this would be a straightforward task, as the calculations involved are not more complicated than in the static case. As mentioned above, this is something we would like to implement in the future.

2.3 Summary

As we initially tested the competing software during the inception phase of development, we were left with three key observations. These were the following:

- Transparency: Many of the testers we have found did not look transparent or trustworthy, which is something we should avoid.
- Learning Curve: After getting to know our target audience we concluded that most seem willing to learn how to use a complex system if it is worth it.
- Design of the UI: It was easy to tell which backtesters are still getting updated, by looking at their design. We should aim at looking fresh. In addition, some backtesters offered so many features that every corner of their UI was packed with information, this is something we should also avoid doing.

Having analysed the competing software, we are now better suited to determine the requirements of our system.

3 Requirements

We classify our requirements using the established FURPS+ model [1]. Below you will see our main functional and non-functional requirements. The items outlined below focus on some of the key requirements we have so far identified as features necessary to provide a compelling product for paying customers.

3.1 Functional Requirements

Portfolio Configuration	
Allocate fixed amount/proportions of the portfolio to given assets	Choose how much each asset contributes to the portfolio's total value using either percentages or raw monetary amounts
Find assets quickly by category or name	When adding an asset the user can search a category for assets or search for a specific asset by its name
Share portfolio	Portfolios can be shared between people using a URL
Edit portfolio	Change asset allocation and their distributions in a portfolio

Portfolio analysis	
Compare portfolios	Use multiple portfolios in a single analysis to see differences in their performance
Use a selection of lazy portfolios	Select an existing common portfolio to compare against, such as common index funds (e.g. Vanguard 500 Index Investor or SPY)
Plot portfolio as a time-series	View portfolio performance as a line graph for a quick overview
Specify a time frame for the analysis	Select start and end dates for portfolio analysis
Choose rebalancing strategy	Optionally choose a strategy for buying and selling assets to meet your strategy e.g. buying and selling stocks each year to ensure the value of portfolio stays at 60% stocks and 40% bonds (i.e. maintain the initial allocation)
Change the distribution of assets in a portfolio using a slider	A slider for each asset to quickly increase or decrease its proportion of the total value
Edit portfolio analysis	Change parameters for portfolio's analysis after running it (e.g. date range or rebalancing strategy)

View results	
See key numerical figures	Show important numerical metrics for a portfolio's performance such as Initial Balance, Standard Deviation, Worst Year, Sharpe Ratio, and Sortino Ratio
See both real and nominal values	See portfolio's value as both adjusted and not adjusted for inflation
A breakdown of portfolio value at specific points of time	See what the value of the portfolio is at some point in time (e.g. January 3rd 1997)
Export result of analysis	Exports results to PDF for sharing and offline reading

User accounts	
Combine portfolios	Combine two portfolios' assets into one single portfolio
Save portfolio analysis for later	Save portfolio analysis parameters to the account so you can rerun it with a single click
Delete saved portfolio analysis	Remove a stored portfolio analysis from your account
Manage portfolio analyses	Edit saved portfolio analysis with different assets, distributions or other parameters
Sign-up, log in and log out	Basic authentication

Assets	
Choose assets from European market	Data for European assets were found to be lacking in competing products
Choose assets from Equities, Fixed Income, Currencies, Commodities, and Cryptocurrencies	Coverage of some of the largest asset classes

3.2 Non-functional Requirements

1. Usability:

- The product must be easily usable for users who already have some financial investment experience.
- The basic backtesting interface needs to look familiar to people already experienced with it.
- The product must have detailed instructions on how to use its advertised functions.
- All major functions must be visible from the initial landing page.
- Must work in both desktop and mobile browsers.
- The results page should scale with mobile.

2. Reliability:

- The product must have a greater than 99% uptime.
- All our assets need to have up to date daily data where the asset is still publicly tradeable.
- All assets supported by the system must provide all publicly available historical data.

3. Performance:

- The website should load within 3 seconds on mobile [2].
- Large portfolios must be supported - up to 300 different assets.

4. Implementation:

- The system needs to work on a cloud hosting provider.

5. Interfacing:

- The Data Gathering Module must never use APIs stated to-be-deprecated within a month.
- The Data Gathering Module must not exceed its contractual usage limits.

6. Operations:

- An administrator on-call will be necessary for unexpected issues.

7. Packaging:

- The product needs to work inside a Linux container (e.g. Docker).
- All dependencies need to be installable with a single command.

8. Legal:

- All user testing must be done with ethical approval from the University.
- UI must display a clear legal disclaimer about the service not providing financial advice.
- All third-party code should allow for commercial use without requiring source disclosure (e.g. no GPL-3).
- User data handling should comply with GDPR.

4 Design

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

5 Coding and Integration

5.1 Web Framework

One of the first decisions we had to make was which web framework to use. The two major options for Python are Django and Flask. We used Django for our MVP in the first semester and didn't have a great experience with it, so we had to decide whether we were going to stick with it or learn Flask. In the end, we opted to go with Flask for the following reasons:

- Django has one architecture that all projects must share, and we have designed the architecture for our project ourselves. While neither architecture is wrong, the two are not compatible. Flask, on the other hand, is structure-agnostic, so we can lay out the code as we see fit.
- Flask comes with the bare minimum for web-development, which means that we don't need to manage the complexity of any feature we're not using. Django has a more complete feature-set from the beginning. This would be desirable in a large web application, but introduces significant overhead in our case, where the website has only a handful of pages.
- Django all but insists on using its ORM for all database interaction, while we plan to have a more manual approach.
- We spoke to more senior students with more experience in web-development, and their concerns with Django echoed our own.

6 Testing and Evaluation

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

7 Conclusions and further work

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

8 Appendices

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

8.1 Appendix A - User Manual

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

8.2 Appendix B - Maintenance Manual

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

8.3 Appendix C - Glossary

- **Portfolio** - A portfolio is a grouping of financial assets such as stocks, bonds, commodities, currencies and cash equivalents, as well as their fund counterparts, including mutual, exchange-traded and closed funds. An investor's portfolio is the group of assets they have currently invested in. [<https://www.investopedia.com/terms/p/portfolio.asp>]
- **Asset** - Generally an asset that gets its value from being owned; can be traded on financial markets. Stocks, bonds, commodities, (crypto-)currencies are all types of financial assets. [<https://www.investopedia.com/terms/a/asset.asp>]
- **Asset Class** - An asset class is a grouping of investments that exhibit similar characteristics and are subject to the same laws and regulations. Asset classes are made up of instruments which often behave similarly to one another in the marketplace. [<https://www.investopedia.com/terms/a/assetclasses.asp>]
- **Backtesting** - Backtesting is the general method for seeing how well a strategy or model would have done ex-post. Backtesting assesses the viability of a trading strategy by discovering how it would play out using historical data. [<https://www.investopedia.com/terms/b/backtesting.asp>]
- **Standard Strategy / Lazy Portfolios** - A lazy portfolio is a collection of investments that require very little maintenance. [<https://www.thebalance.com/how-to-build-the-best-lazy-portfolio-2466533>]
- **Rebalancing** - Rebalancing is the process of realigning the weightings of a portfolio of assets. Rebalancing involves periodically buying or selling assets in a portfolio to maintain an original or desired level of asset allocation or risk. [<https://www.investopedia.com/terms/r/rebalancing.asp>]
- **Key metrics** - Performance measures of a portfolio that are of high interest to the majority of investors.
- **Standard Deviation** - The standard deviation is a statistic that measures the dispersion of a dataset relative to its mean. [<https://www.investopedia.com/terms/s/standarddeviation.asp>]
- **Worst Year** - The worst performance over any given 365 day period starting from January 1st of some year.
- **Sharpe Ratio** - The Sharpe ratio was developed by Nobel laureate William F. Sharpe and is used to help investors understand the return of an investment compared to its risk. [<https://www.investopedia.com/terms/s/sharperatio.asp>]
- **Sortino Ratio** - The Sortino ratio is a variation of the Sharpe ratio that differentiates harmful volatility from total overall volatility by using the asset's standard deviation of negative portfolio returns, called downside deviation, instead of the total standard deviation of portfolio returns. [<https://www.investopedia.com/terms/s/sortinoratio.asp>]
- **Inflation** - Inflation is a quantitative measure of the rate at which the average price level of a basket of selected goods and services in an economy increases over a period of time. [<https://www.investopedia.com/terms/i/inflation.asp>]
- **Nominal Values** - A value that is unadjusted for inflation.
- **Real Values** - A value that is adjusted for inflation.
- **Equity** - Equity is typically referred to as shareholder equity (also known as shareholders' equity) which represents the amount of money that would be returned to a company's shareholders if all of the assets were liquidated and all of the company's debt was paid off.

- Fixed Income - Fixed income is a type of investment security that pays investors fixed interest payments until its maturity date.
[<https://www.investopedia.com/terms/f/fixedincome.asp>]
- Commodity - A commodity is a basic good used in commerce that is interchangeable with other commodities of the same type. Commodities are most often used as inputs in the production of other goods or services. The quality of a given commodity may differ slightly, but it is essentially uniform across producers.
[<https://www.investopedia.com/terms/c/commodity.asp>]
- FOREX / FX - Forex (FX) is the marketplace where various national currencies are traded. The forex market is the largest, most liquid market in the world, with trillions of dollars changing hands every day.
[<https://www.investopedia.com/terms/f/forex.asp>]
- Overfitting - Overfitting is a modeling error that occurs when a function is too closely fit for a limited set of data points.
[<https://www.investopedia.com/terms/o/overfitting.asp>]
- Leverage - Leverage results from using borrowed capital as a funding source when investing to expand the firm's asset base and generate returns on risk capital.
[<https://www.investopedia.com/terms/l/leverage.asp>]
- Technical Analysis - Technical analysis is a trading discipline employed to evaluate investments and identify trading opportunities by analyzing statistical trends gathered from trading activity, such as price movement and volume.
[<https://www.investopedia.com/terms/t/technicalanalysis.asp>]

References

- [1] Quantpedia. A comprehensive list of tools for quantitative traders. <https://quantpedia.com/links-tools/?category=historical-data>.
- [2] Portfolio Visualizer. <https://www.portfoliovisualizer.com/backtest-portfolio>.
- [3] ETFReplayr. <https://www.etfreplay.com/combine.aspx>.
- [4] Backtest Curvo. <https://backtest.curvo.eu/>.
- [5] Stockbacktest. <http://www.stockbacktest.com/>.