I am a wealth manager, working for a wealth management company incorporated in Switzerland. I am creating a service that is at the frontier of direct wealth management and family office services. This service incorporates the following:

* A comprehensive investment profile, based on behavorial finance
* A cost analysis of the portfolio
* A correlation analysis, to optimize the number of instruments and get rid of redundant vehicles
* An optimizer to calculate the weight of the instruments
* A consolidation tool that also performs performance review and risk management

A substantial part of these services have been developed with python code and incorporated into a front end platform called streamlit (before going more ambitious and try dash or even Django)

I am now working on the optimizer. I want it to be classical at first, a MPT optimizer that maximizes the sharpe ratio, minimizes the volatility or maximize the potential returns (user chooses what he prefers). I will then integrate more advanced optimization (like black letterman, equally risk contribution, …).

For now, the data will be loaded from an excel file that has the following structure:

2 sheets: streamlit, Histo\_Price

* In the streamlit sheet, the first row contains the following data:
  + #ID, the ticker of the instruments
  + #Name, the name of the insrtuments
  + #Asset, the asset class of the instruments
  + #Quantity, the quantity
  + #Last\_Price, the latest price available
* Histo\_Price
  + The first row contains the tickers of the instruments
  + The first column contains the dates. They start in row 3
  + The prices start also in row 3

In Streamlit, we should add the constraints:

* + - * Constraints on max and min weight for asset class
        + We could provide the user with many options, like matching the current asset allocation in terms of weight per asset class, give some flexibility (-/+10% with respect to the current asset allocation)
      * Constraints on max and min weight per instruments

From a visual perspective, it would be important to draw the efficient frontier and plot the current portfolio and the optimized one, to see how far we are currently from the optimal wealth line. We should also have a backtest showing the evolution of the current and optimized portfolio since the beginning of the dataset. We should have a table with metrics lòike total return, annual return, volatility, sharpe, var, excess var, …

In addition, we should have the possibility to add the portfolio that comes from the correlation analyzer. It is the same portfolio but with less instruments (no new instruments). We should be able to compare the old portfolio, the new one and the optimized correlation one

The application should be very simple (even though we have behind the curtails advanced algortihms) to be able to take fast decisions. The idea is to provide the client with full control and clarity over his portfolio

Before starting coding, ask me question you think important to realize the project. It is important to note that we need to create module to have a scalable code that we could reuse for more advanced front end platform (dash, Django, …). In my view, the structure of the project is:  
  
- creation of a new environment in conda, with all the necessary requirements and libraries

* Creation of the main module and the standalone modules for functions etc…