Criterion A Planning

**Client: Jerry Cai, an A Level student and vice leader of an investment team**

**Problem: lack of stock data collection tool**

**Solution: stock data collection and calculation Python module**

**Defining the problem**

There is a growing popularity of business competition among high school students in international schools in Beijing. Some of my peers are participating in one of them, the KWHS Investment Competition, in which they compete their stock investment capabilities. However, they can’t find an easy-to-use platform or a free toolkit for them to massively collect stock data for their quantitative analysis. In response to their demand, I decide to design a Python module aiming to automatically collect and process massive stock data.

According to my consultation with my client (refer to Appendix A), Jerry Cai, vice team leader of one investment team and a current A-level computer science student, the module should be in Python and should have several sub-modules. He suggested that one sub-module should be designated for collecting data from website. The module should possess some operations for him to read, add, delete stock data. Apart from using standard Web Spyder program with saving and opening data sheets, the module should also have some calculations, such as quantitative analysis index, factor ranking, and so on. In addition, the module should update data automatically, and the syntax should be easy to use. He gave me a list of companies and calculations he needed for this product.

My computer science teacher, Ms. Wu Di, approved it and will be my advisor.

**Rationale for proposed solution**

A Python module is suitable for Jerry’s need, since he can easily access my new Python module using “pip install” without installing any other specialized processes. Jerry can apply the module in his quantitative analysis algorithms written in Python because of his long coding experience. He also has a working Python IDE to support my module. In addition, python has simple syntax and is powerful because of the availability of abundant open-source modules, such as pandas (for sheet operation), numpy (for calculations), and requests (for internet request).

I have the skills and project experiences on Web Spyder and data analysis so I can develop this product. I choose Yahoo Finance as data source, as financial data there are open and free to download and use, so there’s no privacy issue and I can access data easily.

I decide not to use any external database, such as MySQL or MongoDB, because they usually are big in size and require lots of setup works. Financial data aren’t very large in size. Online database is not desirable because it creates unnecessary requirement of internet connectivity when using the module. Python has powerful internal modules for file operations, including access, storage, creation, and files modification.

Moreover, Python can run on multiple OS, supporting Windows and Linux/Unix based OS, so my product can be used regardless of my client’s OS, which decreases the maintenance burden because there is no need to rewrite the source code.

**Success criteria**

After consultation with my client, we agreed that the product should possess following functionalities:

For installation and setup process:

* Installation using “pip install” is supported and no other steps are needed;
* The **setup** function can convert setup data and initialize database;

For collecting data:

* An **update** function to update data of companies in database, but this should concatenate on original data rather rewriting original ones;
* A **get** function that can automatically collect ALL the factors and ALL financial data of companies provided by Jerry from Yahoo Finance;
* Progress of functions above should be reported.

For calculations:

* Some basic calculations should be available: **factors, ranking**, etc.;
* Calculations don’t take longer than 1 second;

For data operations:

* **Open** can handle possible errors and open sheets in database;
* **Save** can save data sheets to database as well as to a chosen path;
* **Translate** can translate description of companies into Chinese;
* **Folder operations** can create and check existence of sheets;
* **Path control** can locate factors in database and can output correct path of each module, sub module, and function;
* **Tools and reports** can report database status (data sheet and company counts),;
* **Format control** can clean non-number characters in numbers (comma, money sign, etc.) and format output of **Get data**;
* **Get data** provides functions to retrieve data in database and give formatted results. This include getting data sheets, factors, and so on;