

## Criterion B: Design

To start the design process, I sketched out what I wanted the final product to look like on paper. This is to provide a goal to work towards. The sketches are provided in this folder as sketch1.JPG and sketch2.jpg (I will put them in the final version but I can't right now because RTF)

I want the program to be as modular as possible, so I plan to use Python's module import extensively by creating my own library. The library will contain several files, to start it will have a file that contains the graphics classes and methods to handle the graphical user interface. Each of the API services that my program supports will be in separate files, and have similar layouts. Another file will act as a manager, handling the output from the API files and providing them to the graphics module. Additionally, methods used by more than one module will be kept in a common tools file, so that code is kept as simple and easy to maintain as possible. In my initial testing, I noticed that making the API calls blocked the GUI from responding, so I will need to find a way to multithread the program so that the API calls do not affect the performance of the GUI. I want the program to be set up in this way because it will make it the easiest to maintain if I need to improve upon it in the future.

The path I wish to take while programming my product is one of a feature by feature basis. I plan to start by creating the GUI first as a framework to work around, then slowly add in the features I want to program. Programming the product this way will help to prevent bugs because I will only attempt one issue at a time.

In order to test the product, I will ensure that each feature works in a variety of cases. I have determined the cases to be as follows: The happy path, the slow path, the impatient path. The happy path is testing the program with nothing wrong, perfect circumstances. I go through all the features, testing each sorting method and each combination of filters on and off. The slow path is the same as the happy path, but I test the program in an environment with a very bad internet connection. This includes slow speeds and an inconsistent connection. The final path is the impatient path, to test this I use the program spamming as many of the buttons as possible, this is so that the program is responsive enough to handle all input that is given to it.