Criterion A: Planning

The problem I am tackling for my IA is for the ComOps department at AEP. ComOps deals with the buying and selling of power. They needed a way to find news about the price of power and really all things that would interest that department on consumer sites. The goal of my program will be to get a list of webpages that could be potentially relevant to a certain keyword. Google does not do this for them because Google a) does not do the best job of sorting by relevance in this instance, and b) does not write to a database. The client’s name is Joe Sheridan and he is in the ComOps department. A family member of mine has worked with him on a few occasions and that’s how I got in contact with him.

Rationale for Solution

My Java program will search a list of sources for a given keyword and write the output to a database sorted by relevance. The sources will include most consumer news sites and will also include a Twitter API as well as the option to add more sources. Pre-installed sources will include

* <http://bbc.com/>
* <http://foxnews.com/>
* <http://www.huffingtonpost.com/>
* <http://www.nbcnews.com/>
* <http://www.nytimes.com/>
* <http://www.wsj.com/>
* <http://www.usatoday.com/>
* <http://news.google.com/>
* <http://www.rollcall.com>
* <http://www.latimes.com>
* <http://www.wired.com>
* <http://www.cnn.com>
* <http://www.npr.org>
* <http://abcnews.go.com/>
* <https://www.usnews.com/news>
* <https://www.yahoo.com/news/>
* <https://www.ap.org/en-us/>

In order to read these sources, I’ll need to use a third-party library to connect to the webpage and save the data to a field in the program. I’ll also need to design a searching algorithm to write to some sort of List storing the increasing number of “hits.” I plan on making the GUI fully functioning at all times, so I will need to use Thread to stop the program from hanging while the sources are being searched. Finally, the user will need to possess some way of reading the output, possibly through a web browser or, as Mr. Sheridan would also like, a database.

I decided to implement the solution in Java 8 because:

* Fairly easy to create GUI with
* Object-oriented language so my multi-class design fits well
* Easy-to-follow syntax
* Free to download and distribute

Success Criteria

* Must search for the given keyword
* Results must be at least reasonable
* Execution time should be low
* Program must be efficient (multithreading)
* Interface must be intuitive and easy to use
* Output must be easy to interpret and easy to find
* Code must be readable with comments
* Interface should look modern