Criterion B: Solution Overview

Designing a solution for a problem of this scope can be a very entertaining process. I decided to create it from the outside in – first starting with the GUI (Graphical User Interface), and then moving down into the “brain” of the program. I paid very close attention to detail and spent probably around 16 hours designing the GUI to look and feel very professional. After I had finished the bulk of the GUI, I transitioned to working on the search algorithm and interfacing that with each of the classes.

Class Overviews

There are 9 classes in StoryGrab, together totaling around 2700 lines of code.

**LoginWindow.java**

LoginWindow is the main class of the solution. It contains the code for the GUI as well as functions for the button actions. This class serves as the front-end and interface for the main algorithms.

**BackgroundRunner.java**

BackgroundRunner is the class that, using SwingWorker, allows the GUI to respond while the search algorithm is running. It is also responsible for controlling the multithreading and collecting all the data into a final list.

**Extractor.java**

The Extractor class contains the main searching algorithm. This method, which will be explained in more detail later, is a simple recursive bit of code.

**Scroller.java**

Scroller contains methods that work together to allow the user to control the lines that show up in a GUI panel with the mouse wheel. This is implemented in LoginWindow so that the user can see all the sources that are scanned and their corresponding statuses.

**AutomaticScrollUpdater.java**

To make the program more efficient, Scroller does not update every iteration of the main loop. Instead, it updates every 200 milliseconds. This class runs in the background and uses SwingWorker to be non-blocking.

**Tools.java**

Tools is the only abstract class in this project. It contains three functions: saveToCSV, saveToHTML, and saveToDB. They are used to save the output of the program to certain files to be viewed elsewhere.

**Link.java**

Link contains all the attributes of a Link object, such as title, hyperlink, and body. Implemented correctly, it is an easy way to collect all the data from one webpage and keep it all together.

**HelpMenu.java**

HelpMenu is a GUI form that gives the user help when they ask for it.

**User.java**

User collects the data of a user, just username and password, and stores it in an object.