

StoryGrab

IB COMPUTER SCIENCE HL

INTERNAL ASSESSMENT

By Calvin Kinateder, 2019

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[[1]](#footnote-1)

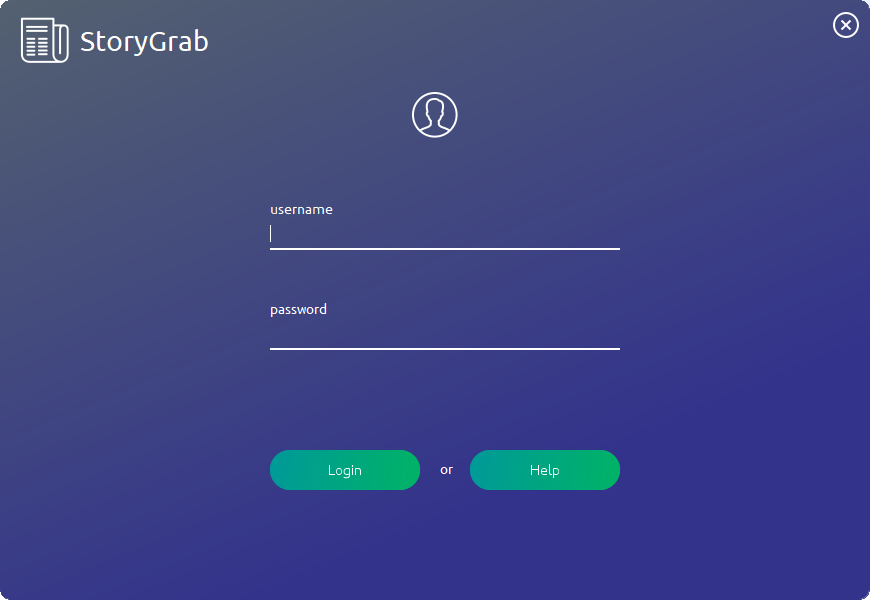
Criterion B: Solution Overview

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.

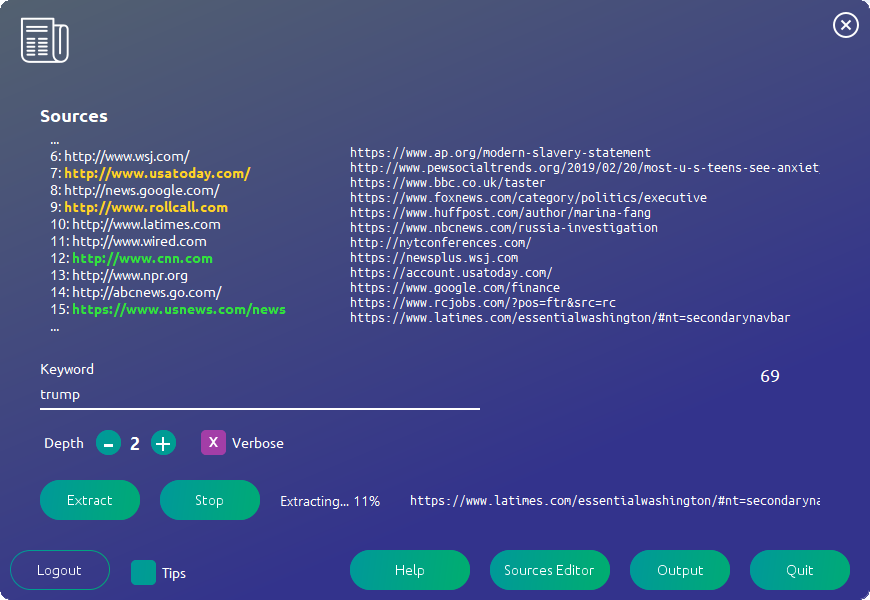
At first, I had designed part of the program to use Vector. There was only one final link set that every instance of Extractor wrote to. This was a problem because a) Vector has been obsolete since Java 2[[2]](#footnote-2), and b) this slowed down the overall execution time. Vector is synchronized each iteration. This is more convenient, but it’s slower. So, I replaced it with ArrayList and it became hitLinks.

Object Design

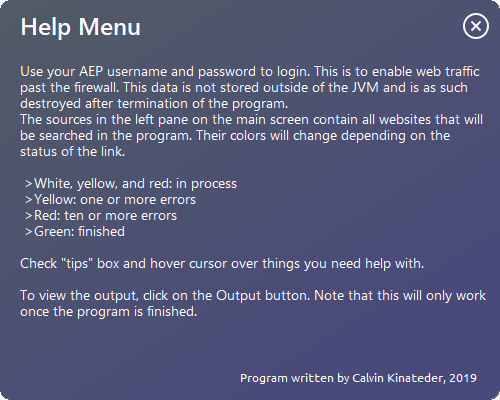
The program will have three forms: a login form (to use with the firewall), a main screen (to operate the extractors from), and a help menu. The color scheme for all three contains blue, grey, and green as the primary colors, and the font family is Ubuntu. The login screen is as follows:



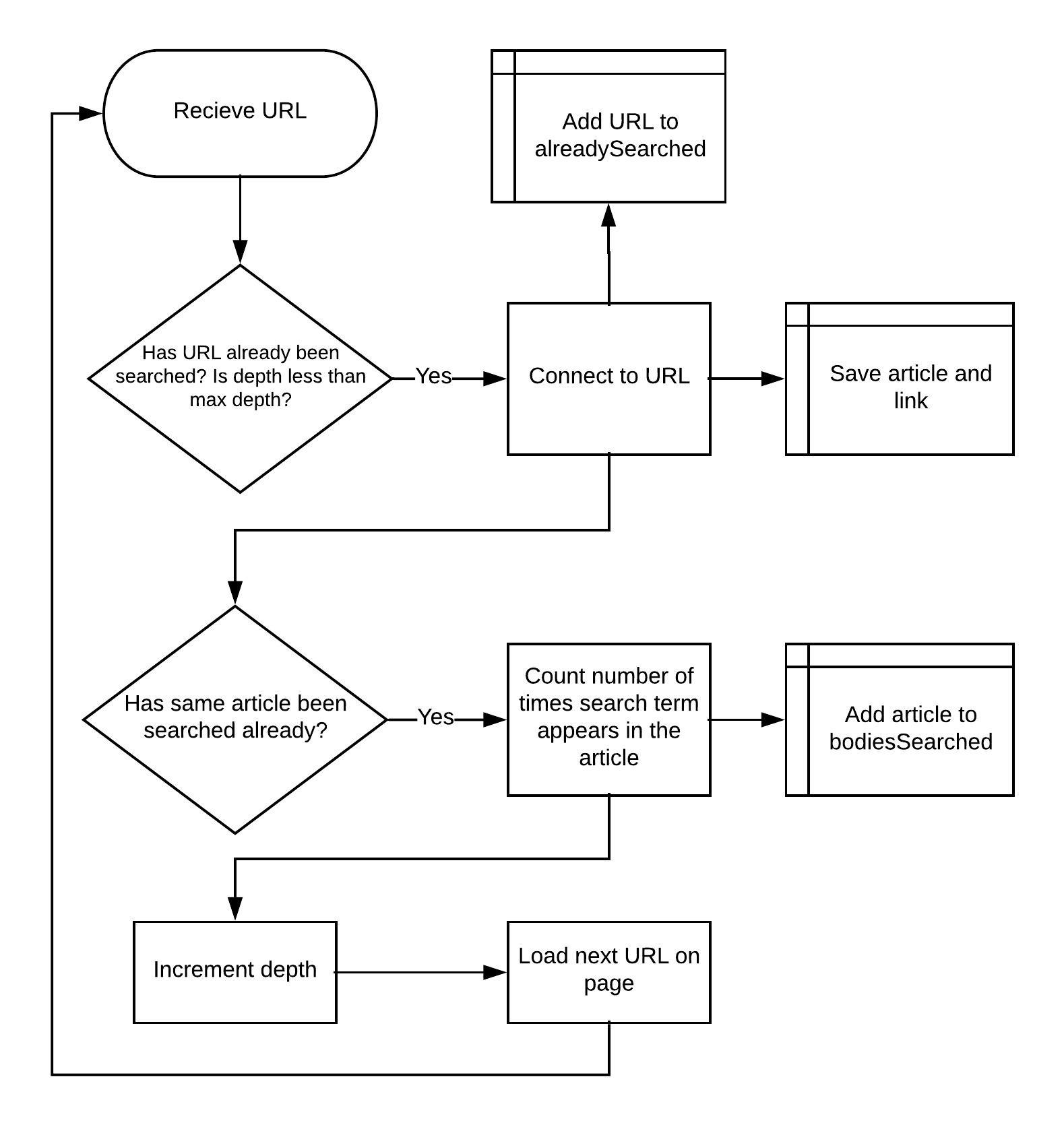
The interface for the main screen is designed for maximum information. The more the user knows about what is going on and how far the program has progressed, the better.

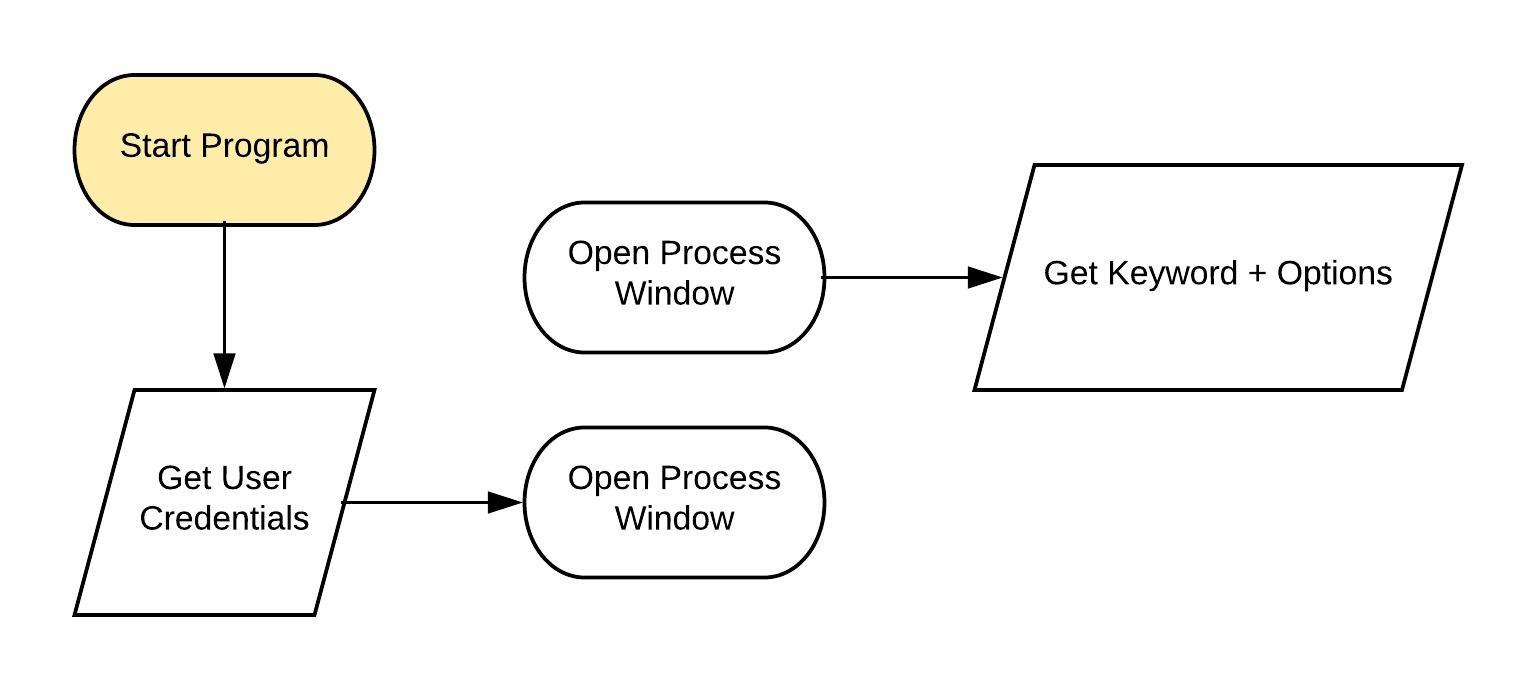
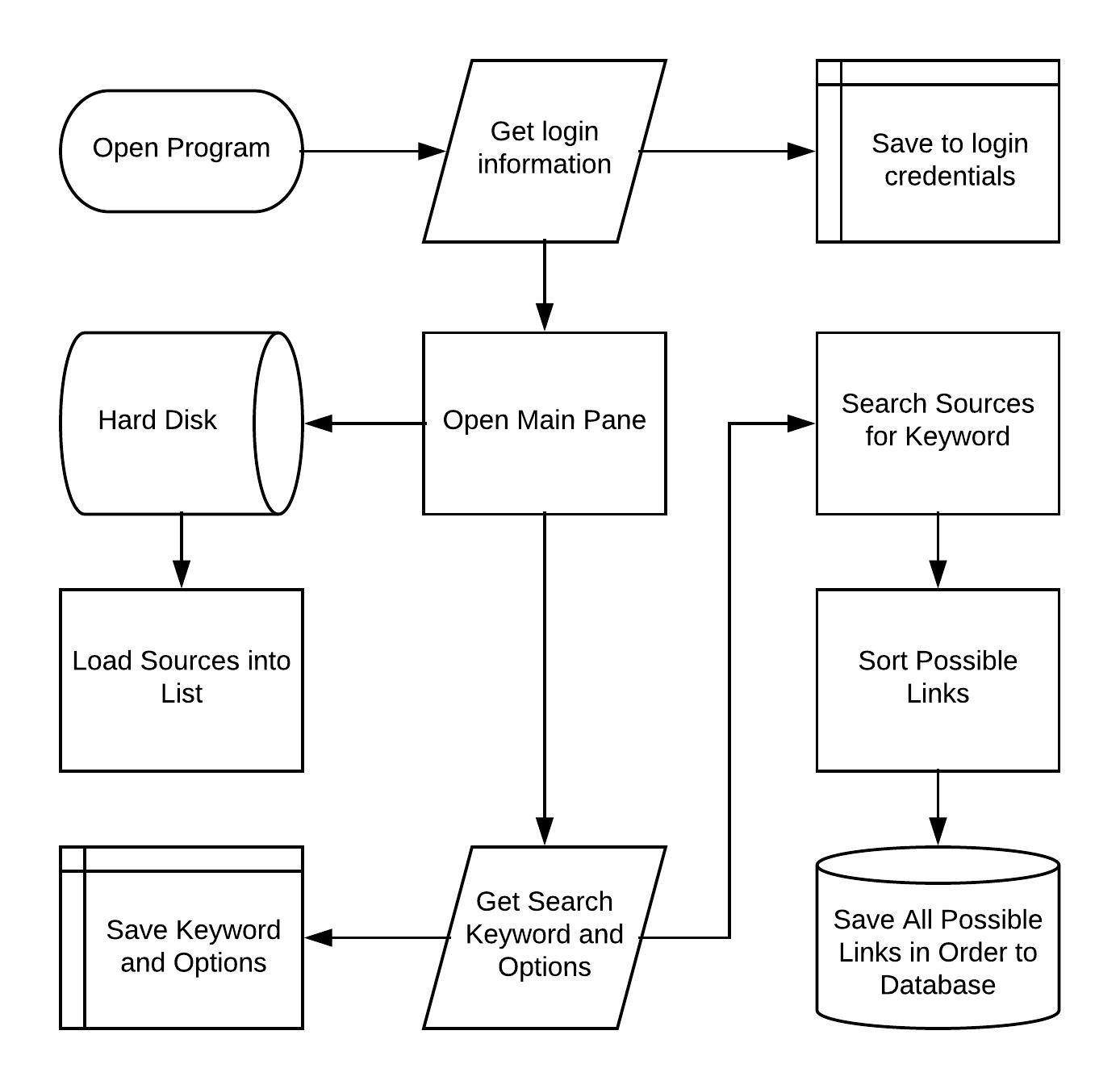


The help menu aims to provide clarity to any possibly vague instructions on the main program.

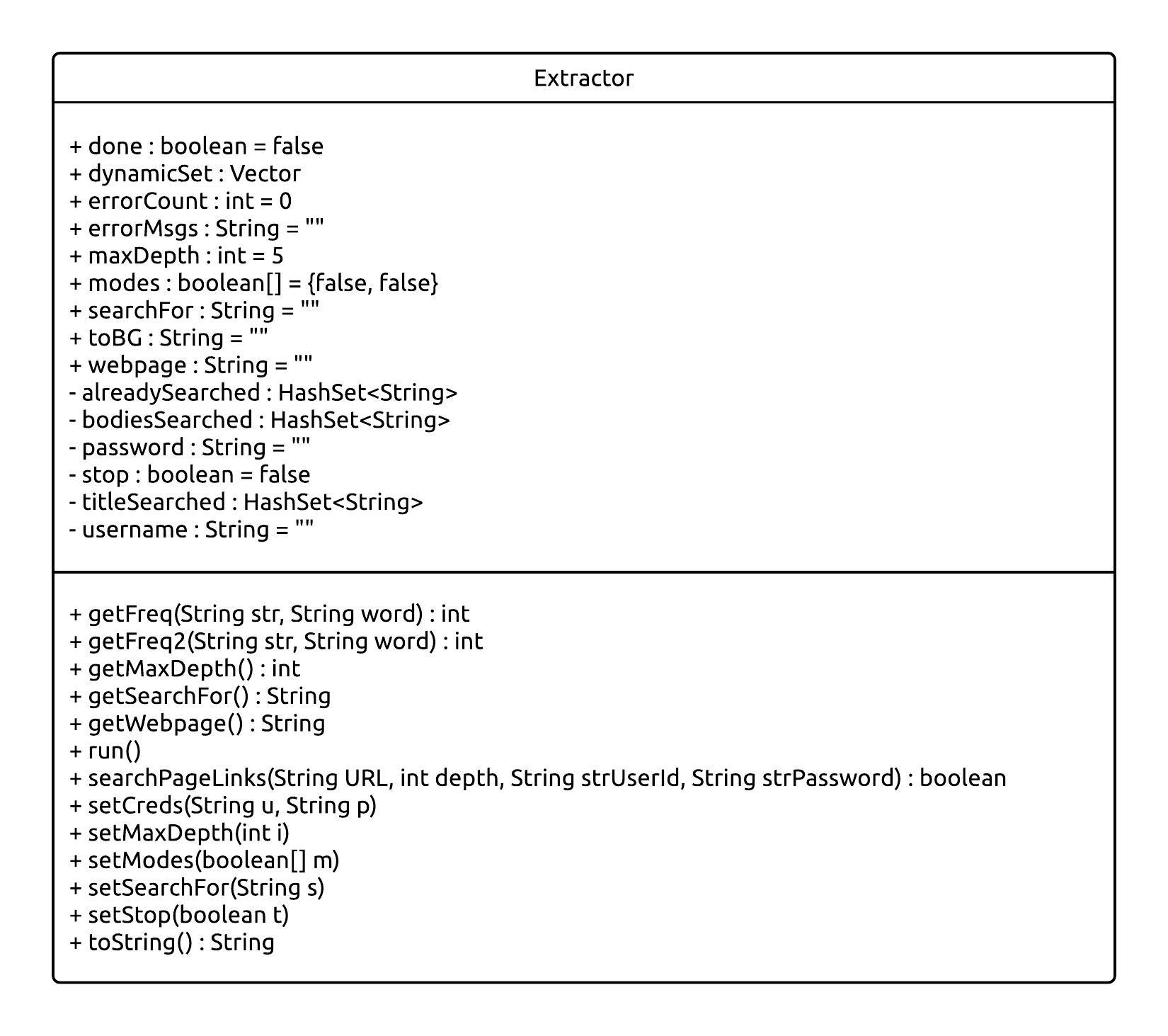


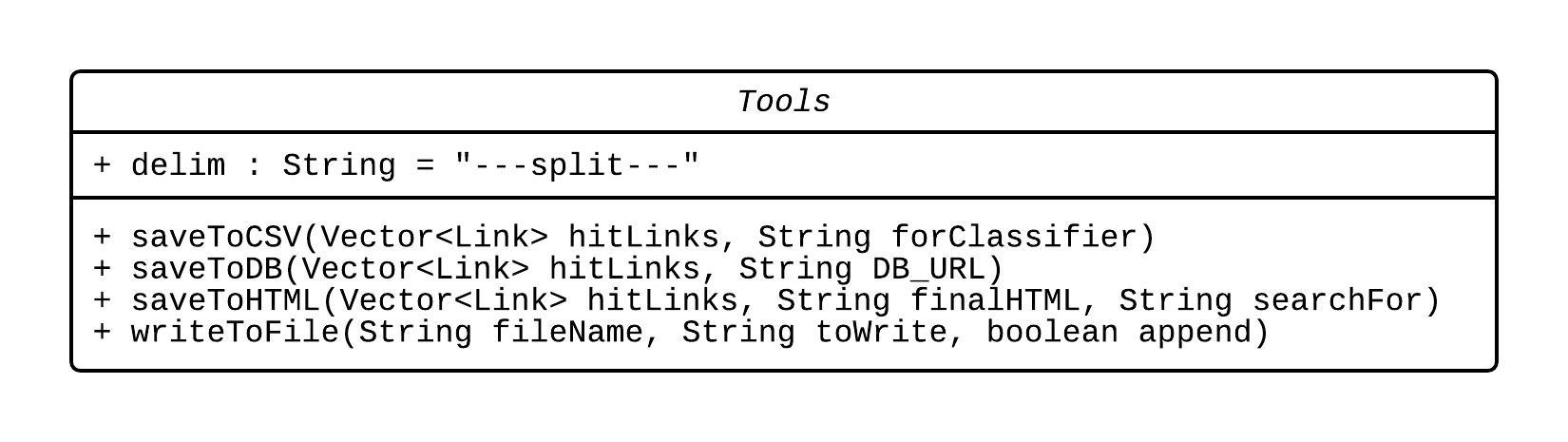
Flowcharts

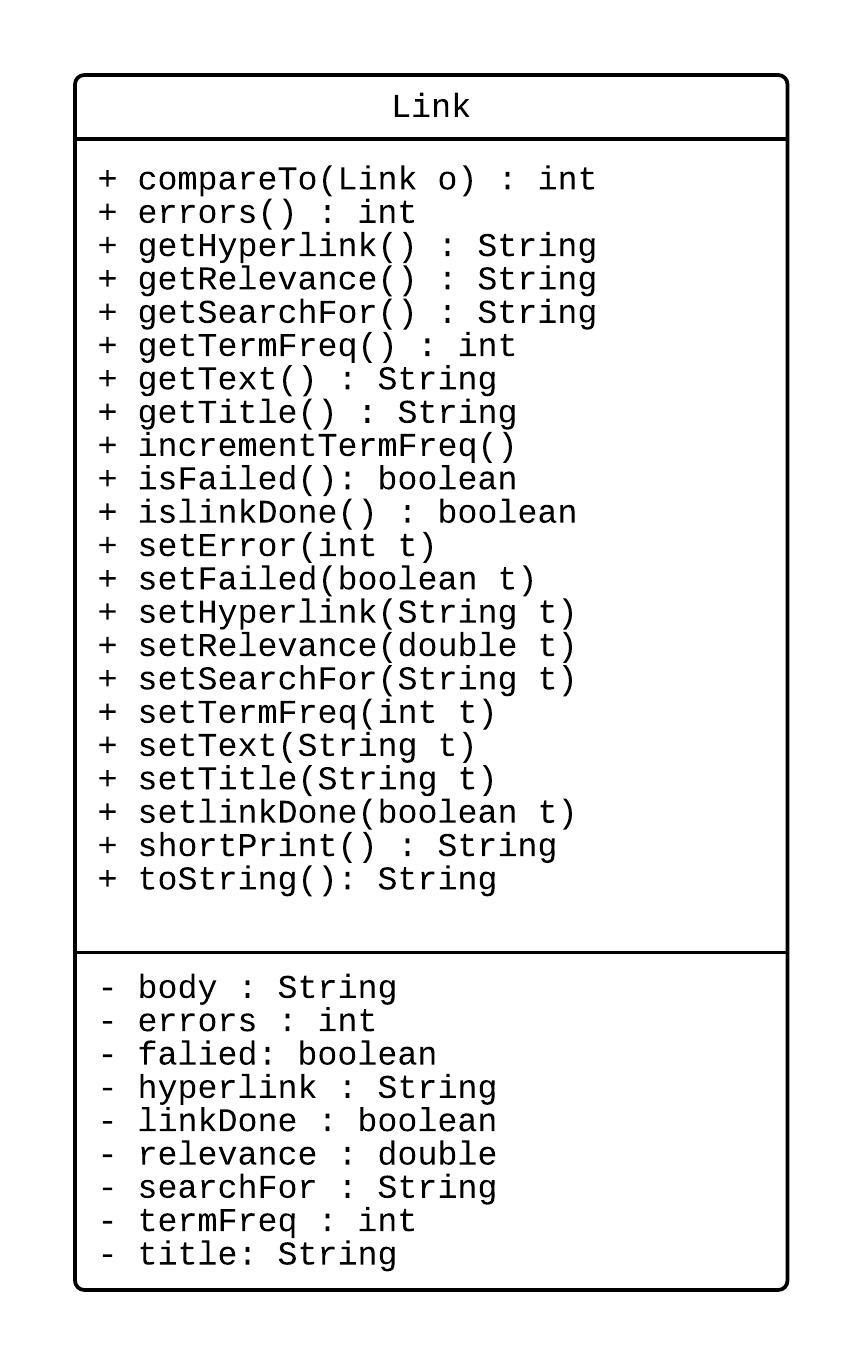
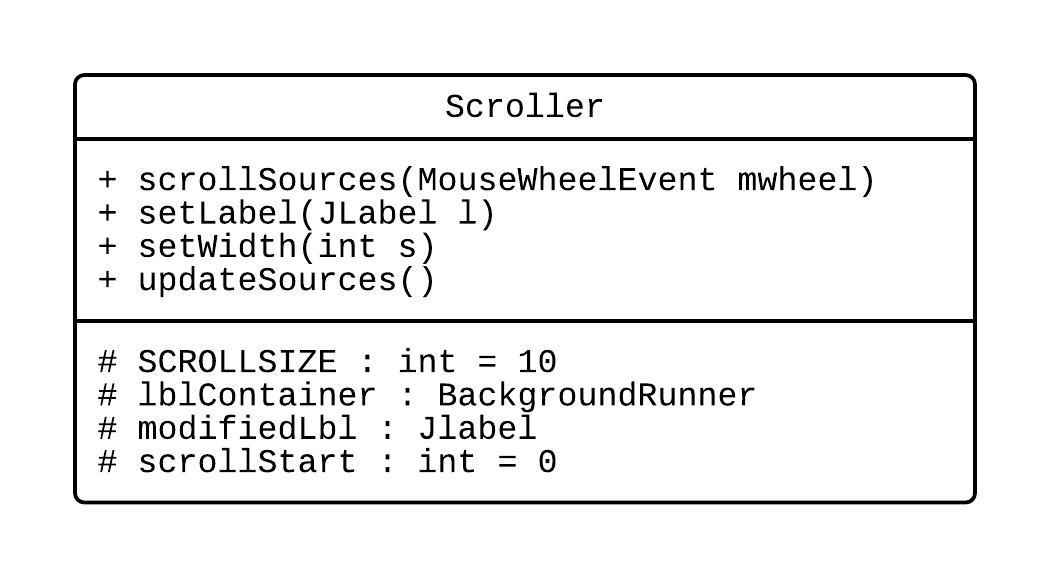
****



Class Diagrams

Using object-oriented programming, I designed this project with 9 classes. They all interact with each other and make use of polymorphism, encapsulation, and inheritance. 





Development Pseudocode

These give a general overview of what the most critical methods do.

***Login:***

*String username*

*String password*

*username and password =respective text fields on the login screen*

*switch to main screen*

***Start extraction (from LoginWindow):***

*if not running:*

*reset all variables*

*pass output labels, datasets to loader //loader is BackgroundRunner object*

*updateSources*

*start extraction in loader*

*endif*

***Start extraction (from BackgroundRunner):***

*while not done:*

*for every source:*

*if source isn’t searching yet:*

*start it searching*

*endif*

*publish output to screen*

*endfor*

*endwhile*

***Search webpage:***

*initialize authentication string to username+password*

*initialize error messages to a blank string*

*if link hasn’t already been searched and depth is less than max depth and no stop flags have been thrown*

*mark URL as already searched*

*connect to the URL and create a document*

*if search term appears more than once in the document*

*Count the number and add it to the list of matches*

*endif*

*for every link on page*

*re-run this method with depth incremented by one*

*endfor*

*endif*

Test Plan

These tests are here to ensure the solution meets the requirements

|  |  |
| --- | --- |
| Action to test | method of testing/expected result |
| Login screen is effective | Try to login to the firewall and access the internet from the program |
| Main screen is fluid and easy to use | Try to move the screen around and play with it to make things crash |
| All button clicks work | Test clicking each button |
| Extracting algorithm is accurate | Test searches with different depths and keywords and see if the output is reasonable |
| Scroll method works | Try scrolling in the source pane and make sure it works right |
| Help menu is helpful | Give the program to a new user and see if they can operate it without help |
| Source editor does not cause problems | Try to add a source from the source.txt file and make sure it does not fail to the backup |
| Output is readable and accurate | Run the program and make sure the output makes sense |
| Stop button saves correctly | Try terminating the program during execution and make sure the data is not corrupted in any way |
| Quit button does not crash but exits cleanly | Try terminating the program with the “quit” or “cancel” buttons |
| Colors in the scroll pane make sense | Run the program and watch the colors in the scroll pane and make sure nothing unexpected happens |

Record of Tasks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TASK # | PLANNED ACTION | PLANNED OUTCOME | TIME ESTIMATED | TARGET COMPLETION DATE | CRITERION |
| 1 | Meet with Joe | Get project idea.  Stimulate thought | 30 minutes | Completed May 2018 | A |
| 2 | Meet with Joe | Get plan  Get enough to work on  for a while | 1 hour | 8/2018 | A |
| 3 | Work on project | get work done  No goal, just as much  as possible | 15 hours | 12/2018 | C |
| 4 | Make flow charts | Finished flow charts | 4 hours | 1/2019 | B |
| 5 | Make UML diagrams | Finished UML diagrams | 3 hours | 1/2019 | B |
| 6 | Make test cases | Finished test cases | 2 hours | 2/2019 | B |
| 7 | Solution tested by client | Updating code | 30 minutes | 2/2019 | B/C |
| 8 | Finalize code | Finished code | 10 hours | 2/2019 | C |
| 9 | Finish writeup and video | Project completed | 12 hours | 2/2019 | A, B, C, D, E |

[[3]](#footnote-3)

Criterion C: Development

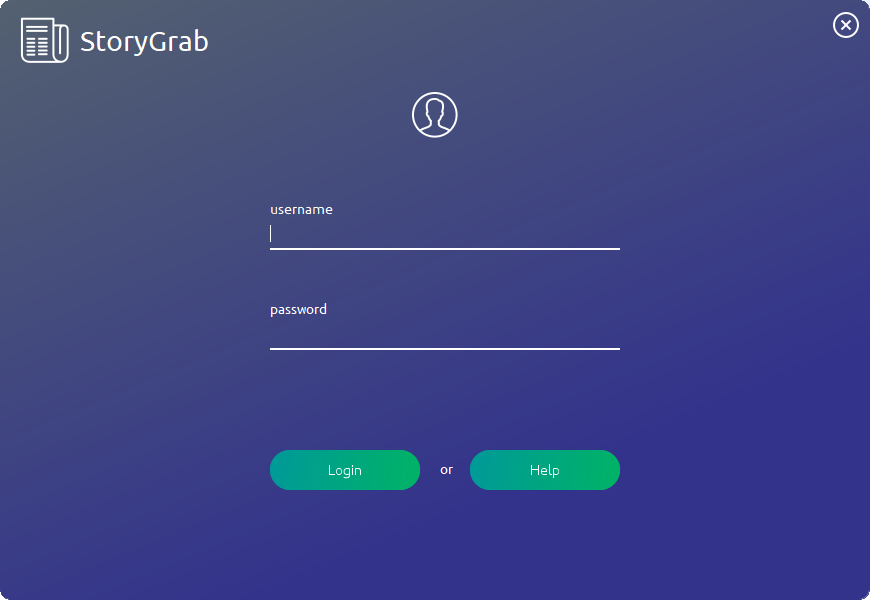
See Appendix 0 for the solution overview.

List of Techniques

This program uses all the following techniques:

* Graphical user interface
* Global and local variables
* Recursion
* Search algorithms
* Merging data structures
* Polymorphism
* Inheritance
* Encapsulation
* Parsing a text file
* Use of additional libraries
* Inserting data into an ordered sequential file
* Abstract data type

Graphical User Interface

The GUI, as requested by the client, shows a visual representation of what is going on in the program.

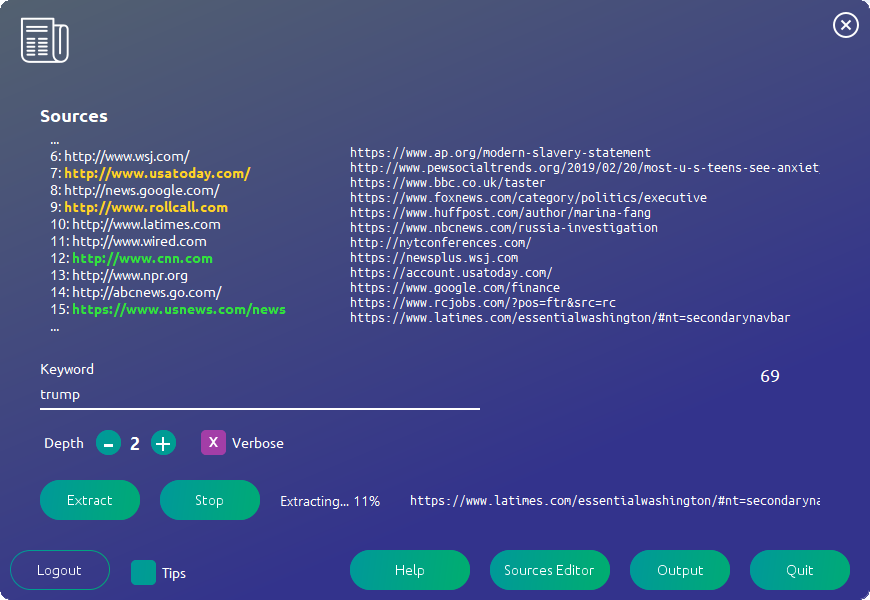
Close button

Help button

Login button

Password field

Username field



Progress label

Tips button

Logout button

Stop button

Extract button

Quit button

Output button

Source button

Help button

Depth controls

# of hits

Output line

Search field

Close button

Output pane

Sources pane

The first code piece is from the method doInBackground() in the class BackgroundRunner. It is placed here to illustrate the use of publish(List<String> chunks) to update the output panel. publish(List<String> chunks) sends chunks to process(List<String> chunks), which is reviewed in the second piece of code. This is because SwingWorker is a special class with reserved methods to be implemented. chunks is parsed and formatted to be more readable and then sent to the 3 output labels, referenced as longout, outputlbl, and hitslbl. hitslbl contains the number of search results currently found. outputlbl contains one line, the most recent message, and longout contains the 12 most recent messages.

Consult appendix for full code.

Search Algorithm

**public** boolean searchPageLinks(String URL, int depth, String strUserId,

String strPassword) {*//returns true when done*

errorMsgs = "";

String authString = strUserId + ":" + strPassword;

String encodedString = *//need to authenticate for firewall*

Base64.getEncoder().encodeToString(authString.getBytes());

**if** ((!alreadySearched.contains(URL) && (depth < maxDepth))&&!stop) {

toBG=""+URL+"\n";*//add to toBG*

**try** {

alreadySearched.add(URL); *//add link to the hashset*

Document = Jsoup.connect(URL) *//connect to doc*

.header("Authorization", "Basic " + encodedString)

.get();

Elements alreadySearchedOnPage = document.select("a[href]");

Elements txt = document.select("p");

String article = txt.text();

String title = document.title();

**if**(getFreq(article, searchFor)>1

&&!bodiesSearched.contains(article)

&&!titlesSearched.contains(title)){

int f = getFreq2(article, searchFor);

hits.add(**new** Link(URL, article,searchFor,title,f));

}

bodiesSearched.add(article);

titlesSearched.add(title);

depth++;

**for** (Element page : alreadySearchedOnPage) {

searchPageLinks(page.attr("abs:href"),

depth,strUserId,strPassword);

}

} **catch** (IOException | IllegalArgumentException e) {

System.err.println("For '" + URL + "': " + e.getMessage());

errorMsgs = "E:"+URL;

errorCount++;

}

}

**return** **true**; *//reached finish*

}

Recursion

searchPageLinks() calls itself in the for loop of this method. The line

**if** ((!alreadySearched.contains(URL) && (depth < maxDepth))&&!stop) {...}

ensures that the method will not advance unless depth < maxDepth, saving the method from an overflow error.

Use of Additional Libraries

This program could not be possible without JSoup.[[4]](#footnote-4) JSoup is an external library that provides a framework to analyze webpages with.

Document = Jsoup.connect(URL) *//connect to doc*

.header("Authorization", "Basic " + encodedString)

.get();

Elements alreadySearchedOnPage = document.select("a[href]");

Elements txt = document.select("p");

Document and Elements are data structures created by JSoup.

Advanced Data Types

Extractor uses HashSet to prevent duplicate links from being saved. HashSet is generally faster than other Collection-type data structures.[[5]](#footnote-5)

**private** HashSet<String> alreadySearched; *//no double searching*

Use case:

**if** ((!alreadySearched.contains(URL) && (depth < maxDepth))&&!stop) {

...

alreadySearched.add(URL); *//add link to the hashset*

...

}

Parsing a Text File

StoryGrab’s sources are stored in a text file outside of Java. FileReader and BufferedReader allow for this data to be read from the Background class.

**public** void updateSrc(){

**if**(!isRunning){

sources.clear();

**try** {

*// FileReader reads text files in the default encoding.*

FileReader =

**new** FileReader(sourcesFile);

BufferedReader =

**new** BufferedReader(fileReader);

String line;

**while** ((line = bufferedReader.readLine()) != **null**) {

sources.add(**new** Link(line));

}

*// Always close files.*

bufferedReader.close();

}

**catch**(FileNotFoundException ex) {

System.out.println(

"Unable to open file '" +

sourcesFile + "'");

backupSrcs();

}

**catch**(IOException ex) {

System.out.println(

"Error reading file '"

+ sourcesFile + "'");

backupSrcs();

}

}

}

A new Link object for each line in the file is created and added to sources. Note that sources is

ArrayList<Link> sources = **new** ArrayList<>();

Multithreading

Multithreading is implemented by creating an ArrayList of extractors and iterating through it whenever they all need to be acted on in the same way. This is all done in a non-blocking way.

**for**(Extractor e : extractors){

e.setModes(**new** boolean[] {**true**, **true**});

e.start();*//make new thread for each extractor*

}

However, the join method,

**for**(Thread tt : extractors){

tt.join();

}

is technically blocking – but as this is within a SwingWorker, it is non-blocking to the rest of the program.

Merging Data Structures

Merging multiple data structures in a program is actually quite simple given that they are all already in an ArrayList.

**public** void combine(){ *//only call when joined*

**for**(Extractor e : extractors){

hitLinks.addAll(e.hits);

}

}

This list is then sorted in cleanup().

Inserting Data into an Ordered Sequential File

This is not needed to be done until the end, when everything is processed and ready for output.

**public** String cleanup(){

combine();

shouldStop = **true**;

String out = "";

Collections.sort(hitLinks);

out+=Tools.saveToCSV(hitLinks, forClassifier)+"<br>";

out+=Tools.saveToHTML(hitLinks, finalHTML, searchFor)+"<br>";

out+=Tools.saveToDB(hitLinks, DB\_URL);

**return** out;

}

cleanup() is called as the program finishes up execution, to save the data that is gathered.

**public** **static** String saveToCSV(List<Link> hitLinks, String forClassifier){

FileWriter =

**null**;

**try** {

fileWriter = **new** FileWriter(forClassifier,**false**); *//add true to append*

BufferedWriter =

**new** BufferedWriter(fileWriter);

**for**(Link l : hitLinks){

bufferedWriter.write(Tools.delim+l.getRelevance()+Tools.delim); *//delim -s ','*

bufferedWriter.write(l.getHyperlink()+Tools.delim);

bufferedWriter.write(l.getText()+"\n");

}

bufferedWriter.close();

fileWriter.close();

**return** "Successfully wrote to CSV";

}

**catch** (IOException ex) {

Logger.getLogger(BackgroundRunner.class.getName()).log(Level.SEVERE, **null**, ex);

**return** "Could not write to CSV";

}

}

Encapsulation

About half of the variables in the project are used privately. For these variables, the only way to access them is with get and set methods. The set methods are void and only set the variable, and the get methods return the variable type and set no variables. Both are essential in accessing their respective variables within the project.

Definitions:

**private** String searchFor;

**private** double relevance;

**private** boolean linkDone;

Accessors/Modifiers:

**public** String getSearchFor(){

**return** searchFor;

}

**public** double getRelevance(){

**return** relevance;

}

**public** boolean islinkDone(){

**return** linkDone;

}

**public** void setSearchFor(String t){

**this**.searchFor = t;

}

**public** void setRelevance(double t){

**this**.relevance = t;

}

**public** void setlinkDone(boolean t){

**this**.linkDone = t;

}

Inheritance

The Extractor class inherits (extends) Thread. Use of the Thread class allows for multithreading within the program.

**public** **class** **Extractor** **extends** Thread {

...

}

This way, Extractor can use the methods and properties of Thread in the project without re-writing that whole class.

Polymorphism

By the principle of polymorphism, inherited classes can change an inherited method from the superclass. This allows for subclasses to be further customized to better fit their needs. For example, the Extractor class extends Thread. Extractor gains properties of the superclass but is able to change some of the methods, such as run() and toString().

run() runs the whole class:

**public** void run() {

Thread.currentThread().setName(webpage);

System.out.println("Extractor running on "+webpage+" at max depth "

+maxDepth);

done = searchPageLinks(webpage, 0, username, password);

**if**(stop){

System.out.println("Extractor prematurely finished on "+webpage);

}

**else**{

System.out.println("Extractor client successful on "+webpage);

}

}

toString() defines what will be returned when the initialized object is printed:

**public** String toString(){

**return** "Page: "+webpage+", Depth: "+maxDepth;

}

*(Full source code and evidence of development are in Appendix 4)[[6]](#footnote-6)*

Criterion E: Evaluation

Meeting the Success Criteria

My initial test criteria stayed around the same throughout the entire project.

* Use a Graphical User Interface (GUI) as the controls for the program – *works well as my GUI has good, readable screens and an informative interface.*
* Take a set of sources to search on – *the “sources editor” button takes users to a text file containing the sources that can be edited.*
* Take a search term and search the given sources for it – *there’s a text field for this and it works.*
* Reasonable speed for program execution – *this works well with the multithreading that I implemented, otherwise it would be 4-8 times slower.*
* View output in a well-readable form – *this works as the output is viewable in a browser.*
* Provide framework for further expansion if necessary – *achieved with comments in the code and the* saveToCSV() *method.* saveToCSV() *creates a dataset to be used with the Python side which I may implement later.*

Recommendations for Further Improvement

Based on what the client decided, there isn’t much to improve on for his purposes. However, we did talk about adding a neural network classifier using Latent-Direchlet Allocation. This would improve the accuracy but would be incredibly difficult, if not impossible without years of study. So, the client and I decided not to take this approach.

Overall, the client and all parties involved were satisfied with the program. I think I was successful in designing and coding an efficient and useful program. Using Java was a good choice due to its understandable syntax and multi-platform support, and its simplicity in designing a GUI. I was also able to implement polymorphism, inheritance, encapsulation, and multithreading, as well as many other computer science principles.[[7]](#footnote-7)

Appendix 0

Solution Overview

StoryGrab uses many classes working tightly together to achieve the fluidity that it does. LoginWindow acts a net for every input, taking user feedback and distributing it to where it needs to go. Recall that StoryGrab takes an almost infinite number of sources to search for. Also familiarize yourself to the concept of an Extractor object: a shell for a recursive search algorithm that contains all methods needed to effectively search a webpage. This class saves the hits to an ArrayList called hits. Note that Extractor extends the Thread class. Extractor objects are not meant to be operated directly from LoginWindow. Instead, they are run from BackgroundRunner, which acts as a container and runner for each instance of the Extractor class (one for each source). BackgroundRunner also implements the SwingWorker class. This allows BackgroundRunner to process in the background and to be non-blocking. Now, running each extractor sequentially would be a mistake. This would make execution take much longer than it really has the potential to do with multithreading.

Because Extractor can be created as a Thread object, it retains Thread methods such as start() and join(). Extractor objects are all started individually on the search term defined in BackgroundRunner. Until they have all finished executing, the program (in BackgroundRunner) continuously checks if they are done. When they are all finished, they are joined together using Thread.join() and each respective hits is combined into a sorted final set called hitLinks. Using the static methods of abstract class Tools, this set is then saved to an HTML file to be viewed in a browser.

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 important 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.

Appendix 1: Works Cited

Java Swing UI - SignUp Example 2018 -KControls. Screenplay by KeepToo, YouTube, 2018.

"Recursion Definition." Merriam-Webster, www.merriam-webster.com/dictionary/recursion. Accessed 28 Feb. 2019.

Leadership, The Berlin School Of Creative. "10 Journalism Brands Where You Find Real Facts Rather Than Alternative Facts." Forbes, 10 Feb. 2017,

"Creating a thread: Extending thread vs implementing runnable - JAXenter." JAXenter, 10 Mar. 2016, jaxenter.com/learn-from-brilliant-java-programmers-to-create-a-thread-with-implements- runnable-vs-extends-thread-124627.html.

"How to write good pseudocode." Gist, 28 Feb. 2019, gist.github.com/camilstaps/59c4574ab8131fb83612a446606cbcba.

"Java Best Practices – Vector vs ArrayList vs HashSet." Java Code Geeks, 28 Feb. 2019, www.javacodegeeks.com/2010/08/java-best-practices-vector-arraylist.html.

Janssen, Thorben. "OOP Concept for Beginners: What is Encapsulation." Stackify, 1 Dec. 2017, stackify.com/oop-concept-for-beginners-what-is-encapsulation.

Appendix 2: Interview Transcript

[Greetings]

Joe: Alright, so what is this school project.

Calvin: Well I’m taking IB computer science and we do a project with a client so I was wondering if you could maybe help me

Joe: Yes I probably could.

Calvin: Great! So do you have any problems you would need me to code a solution for?

Joe: Well we do a lot with news. We have commercial news sites that we use for getting things like the price of coal and stuff like that, but we don’t have anything to do that for consumer news sites.

Calvin: So what would that entail?

Joe: Well you would need to handle a lot of data and parse into links sorted by relevance. It would need to be stored in a database for the output. People on my team would use it so it would need to be easy to use. It would probably be a good idea to multithread because it’s a lot of data.

Calvin: Ok cool I’ll update you with my progress.

Joe: Sounds good.

[Calvin leaves]

Hi Calvin!

I hope you had a good summer! I spoke with my management earlier and we can only use data which is also available to the public, but that doesn’t mean we can’t still do something cool and useful to AEP. I had a couple of ideas, but I’m open to anything.

Com Ops uses a tremendous amount of information (for example news stories) from a large number of disparate sources. There is usually a lot of data presented that is of no use, but we still have to sift through it to find the few useful items. One idea I had was to write a program that could analyze various sites and produce a list of things we might be interested in. The most fun option might be to use the free (standard version) Twitter API (<https://developer.twitter.com/en/docs.html>), but you could always just pull in a couple of news sources instead. You could use a string compare or regex to find specific strings that would be of interest. Regular expressions are a little scary at first, but they are useful for a lot of different situations and worth learning. You mentioned that you were interested in machine learning, if you want to take that on you could use automatic classification tools (that will be a lot of work though). If you wanted to build a GUI, the user could input keywords to look for. You could write the results to an HTML file (maybe just links and a subject line) or even a database if you want to get ambitious. Just one idea, and if you like it we can flesh out the requirements.

Another option I thought of is to use the publicly available Data Miner 2 from the PJM website (<https://www.pjm.com/markets-and-operations/etools/data-miner-2.aspx>). PJM is a bulk electricity market that AEP sells our generation (energy created from our power plants) into and buys our load (the electricity our customers actually use). We can go over what the market is and isn’t if you want to go this route. Data Miner 2 has a web service API (<https://www.pjm.com/-/media/etools/data-miner-2/data-miner-2-api-guide.ashx?la=en>), though I’ve never used it so I’m not sure how complicated it is. Pretty much anything publicly available on Data Miner 2 would be useful. If you wanted you could store it into a database like derby (<https://db.apache.org/derby/>) or something.

Let me know what you think. I’m open to just about anything.

Thanks!

-Joe

Thanks for working to let me help you with some programs. I'm not really sure what the second option's goal would be, do you think you could elaborate a little on it? The first option looks like it would work well for me though, and I've done some regex stuff before in linux although I'll need to brush up a little on using that with Java. I also did some GUI work last year for a project so I could make a front end for it too. I'm interested in the ML automatic classification feature for that as well, and it would be really great to pipe the output to a database. What are the requirements for that project?

Also, when would you need me to come down to AEP next?

Thanks,

Calvin Kinateder

Appendix 3: Git Log

9da0ba4 - (HEAD -> master, origin/master) 10th complexity - could be done (24 hours ago - Fri, 22 Feb 2019 09:35:34 -0500) <Calvin>

5d9ce7f - code almost done (2 days ago - Thu, 21 Feb 2019 20:44:11 -0500) <calvin kinateder>

d74f7a3 - fix spelling errors lol (7 days ago - Sat, 16 Feb 2019 18:19:16 -0500) <calvin kinateder>

9fae94d - fix style (8 days ago - Fri, 15 Feb 2019 18:49:52 -0500) <calvin kinateder>

4c0f5b4 - updated class overview (10 days ago - Wed, 13 Feb 2019 06:52:11 -0500) <calvin kinateder>

2083f22 - added more comments (11 days ago - Mon, 11 Feb 2019 22:12:34 -0500) <calvin kinateder>

ba1489b - vector successfully eliminated (13 days ago - Sun, 10 Feb 2019 18:44:05 -0500) <calvin kinateder>

479708a - about to try to eliminate Vector :/ (2 weeks ago - Wed, 6 Feb 2019 17:47:23 -0500) <calvin kinateder>

ae4940f - fixed csv problem (3 weeks ago - Tue, 5 Feb 2019 20:07:26 -0500) <calvin kinateder>

65a3f02 - Update README.md (3 weeks ago - Sun, 3 Feb 2019 14:37:51 -0500) <Calvin Kinateder>

0631782 - Update README.md (3 weeks ago - Sun, 3 Feb 2019 14:36:47 -0500) <Calvin Kinateder>

9d169c5 - Update README.md (3 weeks ago - Sun, 3 Feb 2019 14:36:33 -0500) <Calvin Kinateder>

855a8b1 - Update README.md (3 weeks ago - Sun, 3 Feb 2019 14:35:51 -0500) <Calvin Kinateder>

4268efe - Update README.md (3 weeks ago - Sun, 3 Feb 2019 14:33:29 -0500) <Calvin Kinateder>

c2b2fb9 - Update README.md (3 weeks ago - Sun, 3 Feb 2019 14:33:09 -0500) <Calvin Kinateder>

17460de - added backup sources hardcoded (3 weeks ago - Sat, 2 Feb 2019 15:57:40 -0500) <calvin kinateder>

18c795a - Merge origin/master (3 weeks ago - Sat, 2 Feb 2019 15:38:45 -0500) <calvin kinateder>

3c2dc7a - changed icon on main screen and move and rename like everything (3 weeks ago - Sat, 2 Feb 2019 15:38:30 -0500) <calvin kinateder>

0b3b8e8 - Update README.md (3 weeks ago - Sat, 2 Feb 2019 15:16:45 -0500) <Calvin Kinateder>

52b7c3f - Update README.md (3 weeks ago - Sat, 2 Feb 2019 15:16:24 -0500) <Calvin Kinateder>

6048ed6 - Update README.md (3 weeks ago - Sat, 2 Feb 2019 15:16:10 -0500) <Calvin Kinateder>

e36b9cf - moved some files around (3 weeks ago - Sat, 2 Feb 2019 15:10:24 -0500) <calvin kinateder>

b4c6bb2 - aDDED a comment (3 weeks ago - Thu, 31 Jan 2019 09:26:43 -0500) <calvin kinateder>

4816f57 - removed unused imports (3 weeks ago - Wed, 30 Jan 2019 15:03:18 -0500) <calvin kinateder>

190493d - more javadoc updating again (3 weeks ago - Wed, 30 Jan 2019 14:08:30 -0500) <calvin kinateder>

5e25c9d - changed icons, added quit button (4 weeks ago - Tue, 29 Jan 2019 13:56:59 -0500) <Calvin>

5ce4669 - deleted unused files and removed license headers (4 weeks ago - Tue, 29 Jan 2019 12:21:10 -0500) <Calvin>

2b88d2d - twitter gone (4 weeks ago - Tue, 29 Jan 2019 12:06:52 -0500) <Calvin>

b368db1 - more documentation (4 weeks ago - Tue, 29 Jan 2019 11:52:28 -0500) <Calvin>

a5d7f97 - more javadoc updating again (4 weeks ago - Mon, 28 Jan 2019 21:16:56 -0500) <calvin kinateder>

a296733 - more javadoc updating (4 weeks ago - Mon, 28 Jan 2019 18:24:27 -0500) <calvin kinateder>

1c46150 - added more javadoc (4 weeks ago - Mon, 28 Jan 2019 15:41:17 -0500) <Calvin>

9324749 - possibly fixed java.util.ConcurrentModificationException (4 weeks ago - Mon, 28 Jan 2019 10:28:54 -0500) <Calvin>

fb2495b - added error catching and reporting for Tools (4 weeks ago - Sun, 27 Jan 2019 22:01:06 -0500) <calvin kinateder>

212e56b - can restart! (4 weeks ago - Sun, 27 Jan 2019 21:49:42 -0500) <calvin kinateder>

21b8f7b - added different extractor finish message (4 weeks ago - Sun, 27 Jan 2019 21:39:16 -0500) <calvin kinateder>

75418ba - removed more unused functions (4 weeks ago - Sun, 27 Jan 2019 15:39:11 -0500) <Calvin>

41f14fd - removed more unused functions (4 weeks ago - Sun, 27 Jan 2019 15:19:16 -0500) <Calvin>

72fbe07 - took out some unnessecary code (4 weeks ago - Sun, 27 Jan 2019 13:29:17 -0500) <calvin kinateder>

85fe79f - removed unused file variable (3 months ago - Mon, 3 Dec 2018 10:14:12 -0500) <calvin kinateder>

7e5b41f - cleaned up (3 months ago - Mon, 12 Nov 2018 08:32:40 -0500) <Calvin>

4144fc6 - added op.gif + deleted some duplicate files (3 months ago - Mon, 12 Nov 2018 08:27:51 -0500) <Calvin>

b38b175 - sync (3 months ago - Sun, 11 Nov 2018 17:42:30 -0500) <Calvin>

2309db1 - fixed output not opening (4 months ago - Wed, 7 Nov 2018 22:22:28 -0500) <calvin kinateder>

41072a9 - removed 1 thing (4 months ago - Wed, 7 Nov 2018 22:05:07 -0500) <calvin kinateder>

734dbc9 - added more umls (4 months ago - Wed, 7 Nov 2018 21:58:33 -0500) <Calvin>

9ce9c2b - sync (4 months ago - Wed, 7 Nov 2018 18:24:13 -0500) <Calvin>

59fbce3 - fixed width issues (4 months ago - Tue, 6 Nov 2018 21:59:01 -0500) <calvin kinateder>

e3c7cd5 - output looks nicer also fixed the sorting issue by making a temp copy of the hitLinks (4 months ago - Tue, 6 Nov 2018 21:54:10 -0500) <calvin kinateder>

c18dea2 - changed to LDA multicore (4 months ago - Mon, 5 Nov 2018 12:04:24 -0500) <Calvin>

33a2253 - syncing classifier2 still crashes on CoherenceModel (4 months ago - Mon, 5 Nov 2018 07:02:45 -0500) <calvin kinateder>

e651149 - added mallet and trump3078 dataset (4 months ago - Fri, 2 Nov 2018 14:24:44 -0400) <Calvin>

6123ab1 - biggest dataset (4 months ago - Thu, 1 Nov 2018 22:36:57 -0400) <calvin kinateder>

91e44f2 - html works well (4 months ago - Thu, 1 Nov 2018 14:44:42 -0400) <Calvin>

22b658b - updated links.csv (4 months ago - Wed, 31 Oct 2018 22:10:40 -0400) <calvin kinateder>

4ecd6de - started saveToHTML() (4 months ago - Wed, 31 Oct 2018 15:01:44 -0400) <Calvin>

b6cbb72 - added getFreq2() (4 months ago - Wed, 31 Oct 2018 14:57:02 -0400) <Calvin>

ed6e59c - updated classifier.py (4 months ago - Tue, 30 Oct 2018 11:09:26 -0400) <calvin kinateder>

cd79d34 - classifier updated and moved some stuff around (4 months ago - Mon, 29 Oct 2018 15:01:37 -0400) <Calvin>

45ec391 - updated classifier.py (4 months ago - Mon, 29 Oct 2018 11:25:06 -0400) <calvin kinateder>

51a1805 - topic modeling ACTUALLY KIND OF WORKS!!! (4 months ago - Mon, 29 Oct 2018 01:29:36 -0400) <calvin kinateder>

341f3ff - no duplicate articles (4 months ago - Sun, 28 Oct 2018 19:34:32 -0400) <calvin kinateder>

20f172a - changed screenshots (4 months ago - Sun, 28 Oct 2018 17:36:10 -0400) <calvin kinateder>

0d42728 - changed a lot - changed datastructure and started on python (4 months ago - Sun, 28 Oct 2018 17:33:55 -0400) <calvin kinateder>

ec2f079 - sorting algorithm updated with multiple delimiters (4 months ago - Sat, 27 Oct 2018 22:30:23 -0400) <calvin kinateder>

cd31b0b - database works, added compareTo to Link, but sorting algorithm isn't good (4 months ago - Fri, 26 Oct 2018 11:42:00 -0400) <Calvin>

a98ea26 - added FinalLinks Derby database (4 months ago - Fri, 26 Oct 2018 09:13:33 -0400) <Calvin>

0c716be - tried to fix continuous errors on output (4 months ago - Tue, 16 Oct 2018 14:52:36 -0400) <Calvin>

0546791 - started migrating to ranking output by termFreq (4 months ago - Mon, 15 Oct 2018 19:33:27 -0400) <calvin kinateder>

e9d0b32 - update ia (4 months ago - Fri, 12 Oct 2018 00:53:45 -0400) <calvin kinateder>

51bc9bc - started framework for modes array (5 months ago - Thu, 11 Oct 2018 08:48:09 -0400) <Calvin>

530d7c6 - update ia (5 months ago - Thu, 11 Oct 2018 01:51:34 -0400) <calvin kinateder>

0f9e671 - hit count works! (5 months ago - Thu, 11 Oct 2018 01:27:02 -0400) <calvin kinateder>

5db03fb - added error color codes (5 months ago - Wed, 10 Oct 2018 19:56:55 -0400) <calvin kinateder>

962c7dd - added search settings and error color flashing (5 months ago - Tue, 9 Oct 2018 21:36:32 -0400) <calvin kinateder>

15562d5 - CHANGED SOURCES TO VECTOR (5 months ago - Tue, 9 Oct 2018 14:31:49 -0400) <Calvin>

47d0522 - Merge origin/master (5 months ago - Tue, 9 Oct 2018 08:06:32 -0400) <Calvin>

d7de0f5 - added link and stuff (5 months ago - Mon, 8 Oct 2018 21:35:08 -0400) <calvin kinateder>

a1e415c - delimit in comma csv (5 months ago - Mon, 8 Oct 2018 20:55:08 -0400) <calvin kinateder>

49944bb - Delete Mon Oct 8 13:49:14 EDT 2018.txt (5 months ago - Mon, 8 Oct 2018 16:43:28 -0400) <Calvin Kinateder>

6d92b7d - added purposes to each file, 10 (5 months ago - Mon, 8 Oct 2018 14:44:34 -0400) <Calvin>

f37ad39 - added log\_MAC.command (5 months ago - Mon, 8 Oct 2018 13:50:10 -0400) <Calvin>

b783566 - fixed scroller not updating and added catch to prohibit source modification while running (5 months ago - Mon, 8 Oct 2018 13:23:16 -0400) <Calvin>

0a75026 - updated working tree (5 months ago - Mon, 8 Oct 2018 00:30:39 -0400) <calvin kinateder>

39f540e - updated IA.docx (5 months ago - Sat, 6 Oct 2018 19:09:10 -0400) <calvin kinateder>

55dd6fd - fixed some formatting (5 months ago - Sat, 6 Oct 2018 16:23:17 -0400) <calvin kinateder>

30c34e2 - fixed some formatting (5 months ago - Sat, 6 Oct 2018 16:22:39 -0400) <calvin kinateder>

cfaec09 - about to migrate to full on LINK (5 months ago - Sat, 6 Oct 2018 02:32:57 -0400) <calvin kinateder>

2de8e05 - open out opens links.txt (5 months ago - Fri, 5 Oct 2018 21:51:26 -0400) <calvin kinateder>

bac4759 - ICONS WORK!!! (5 months ago - Fri, 5 Oct 2018 20:59:10 -0400) <calvin kinateder>

2637835 - mainly just updated working tree (5 months ago - Fri, 5 Oct 2018 14:56:28 -0400) <Calvin>

0d9b3d1 - added tooltips (5 months ago - Thu, 4 Oct 2018 13:03:31 -0400) <Calvin>

f90af2b - changed screenshot (5 months ago - Thu, 4 Oct 2018 10:55:29 -0400) <Calvin>

b7409a5 - made some things more consistent (5 months ago - Thu, 4 Oct 2018 09:37:18 -0400) <Calvin>

3b0a398 - renamed variables and took out the trash (5 months ago - Thu, 4 Oct 2018 09:23:44 -0400) <Calvin>

3959af7 - update (5 months ago - Wed, 3 Oct 2018 07:26:48 -0400) <calvin>

388ebbd - improved quit time by 99% (5 months ago - Tue, 2 Oct 2018 14:51:46 -0400) <Calvin>

60db300 - source editor doesnt work but ok to revert to (5 months ago - Tue, 2 Oct 2018 12:02:07 -0400) <Calvin>

b45ce35 - addasource sucks so were gonna fix that next (5 months ago - Mon, 1 Oct 2018 20:30:13 -0400) <calvin>

002011f - fixed scrolling stuff (5 months ago - Mon, 1 Oct 2018 17:48:13 -0400) <calvin>

7e16521 - more cosmetic (5 months ago - Mon, 1 Oct 2018 14:04:57 -0400) <Calvin>

b9d0c56 - added constant for output size and formatted scroller a little (5 months ago - Mon, 1 Oct 2018 12:24:05 -0400) <Calvin>

37f1337 - Added automatic refresh plus complete notification (5 months ago - Sat, 29 Sep 2018 12:42:35 -0400) <calvin>

dd443f3 - minor (5 months ago - Fri, 28 Sep 2018 22:13:54 -0400) <calvin>

2881453 - BIG ONE : THREAD STOP WORKS (5 months ago - Fri, 28 Sep 2018 16:25:35 -0400) <calvin>

d6c031f - modified some names (5 months ago - Fri, 28 Sep 2018 14:50:45 -0400) <Calvin>

b5d8ef1 - added uml project (5 months ago - Fri, 28 Sep 2018 11:22:07 -0400) <Calvin>

1448d59 - updated writeup (5 months ago - Thu, 27 Sep 2018 13:29:10 -0400) <Calvin>

b296504 - added reuters (5 months ago - Wed, 26 Sep 2018 14:07:23 -0400) <Calvin>

a5be823 - Added apache common codec (5 months ago - Wed, 26 Sep 2018 10:10:55 -0400) <Calvin>

607ff35 - change to vector (5 months ago - Tue, 25 Sep 2018 21:46:36 -0400) <calvin kinateder>

420dd88 - continuing linkset (5 months ago - Tue, 25 Sep 2018 16:59:47 -0400) <Calvin>

2031f8e - started linkset (5 months ago - Tue, 25 Sep 2018 13:56:05 -0400) <Calvin>

c808764 - Added Link Class (5 months ago - Tue, 25 Sep 2018 10:35:54 -0400) <Calvin>

a659979 - ADDED ia (5 months ago - Mon, 24 Sep 2018 11:45:51 -0400) <Calvin>

476a949 - replace datasets (5 months ago - Wed, 19 Sep 2018 21:04:18 -0400) <calvin kinateder>

ffde864 - added files (5 months ago - Wed, 19 Sep 2018 17:01:07 -0400) <Calvin>

8745104 - fixed source updating problem and added datasets (5 months ago - Tue, 18 Sep 2018 09:35:29 -0400) <Calvin>

471f20b - update sc (5 months ago - Mon, 17 Sep 2018 13:52:27 -0400) <Calvin>

84b7289 - added verbose option (5 months ago - Mon, 17 Sep 2018 13:50:25 -0400) <Calvin>

9e66540 - updated scroller (5 months ago - Sun, 16 Sep 2018 19:36:32 -0400) <calvin kinateder>

8608e6f - added long out (5 months ago - Fri, 14 Sep 2018 18:54:01 -0400) <calvin kinateder>

b44c9e6 - update (5 months ago - Fri, 14 Sep 2018 15:02:52 -0400) <Calvin>

1802279 - moved scroll to scroller (5 months ago - Fri, 14 Sep 2018 14:13:55 -0400) <Calvin>

2d68635 - disregard source changes, but scroll works (5 months ago - Fri, 14 Sep 2018 13:34:31 -0400) <Calvin>

435771e - finished message is good (5 months ago - Fri, 14 Sep 2018 12:56:28 -0400) <Calvin>

58bbc80 - fixed concurrency error when clicking extract when running (5 months ago - Thu, 13 Sep 2018 20:54:38 -0400) <Calvin>

37bea48 - MAJOR MAJOR MAJOR - STOP FINALLY WORKS OKAY - some sync probs tho (5 months ago - Thu, 13 Sep 2018 18:09:31 -0400) <calvin kinateder>

7dd70ea - update (5 months ago - Thu, 13 Sep 2018 17:13:00 -0400) <Calvin>

dfaec7b - progress output works well (5 months ago - Thu, 13 Sep 2018 16:22:56 -0400) <Calvin>

76a7beb - Updated formatting (5 months ago - Thu, 13 Sep 2018 08:26:16 -0400) <Calvin>

4c942a6 - Extends thread not implements runnable. now can access extractor vars so cool (5 months ago - Wed, 12 Sep 2018 20:53:19 -0400) <Calvin>

d467dbf - OK to revert - progress bar works great (6 months ago - Tue, 11 Sep 2018 08:06:09 -0400) <Calvin>

ea2f66e - dont revert just updating progress (6 months ago - Mon, 10 Sep 2018 17:01:37 -0400) <Calvin>

f48b8d5 - MAJOR. REVERT TO HERE. Progress works! (6 months ago - Mon, 10 Sep 2018 15:11:48 -0400) <Calvin>

3e489c3 - background runner works well (6 months ago - Mon, 10 Sep 2018 14:40:00 -0400) <Calvin>

51cf4cf - extractorcontainer is replaced by background runner (6 months ago - Mon, 10 Sep 2018 09:06:57 -0400) <Calvin>

49961df - just backing up work DO NOT REVERT (6 months ago - Mon, 10 Sep 2018 08:43:23 -0400) <Calvin>

7f73529 - update working tree dont revert to here (6 months ago - Mon, 10 Sep 2018 07:26:13 -0400) <calvin kinateder>

644b6e7 - wouldn't necessarily revert to here, just updating tree for desktop (6 months ago - Sat, 8 Sep 2018 17:30:14 -0400) <Calvin>

30c971f - modified props and little ui change (6 months ago - Fri, 7 Sep 2018 09:17:17 -0400) <Calvin>

3daad9b - added sources and authentication (6 months ago - Wed, 5 Sep 2018 09:50:20 -0400) <Calvin>

97c3158 - added refresh src button (6 months ago - Mon, 3 Sep 2018 15:28:48 -0400) <calvin kinateder>

d2dd75a - fixed readme.md (6 months ago - Mon, 3 Sep 2018 15:23:55 -0400) <calvin kinateder>

299b280 - Delete sgmain.PNG (6 months ago - Mon, 3 Sep 2018 15:22:15 -0400) <Calvin Kinateder>

30caa0f - Update README.md (6 months ago - Mon, 3 Sep 2018 15:21:43 -0400) <Calvin Kinateder>

9d361cb - fixed readme.md (6 months ago - Mon, 3 Sep 2018 15:20:59 -0400) <calvin kinateder>

98e4982 - fixed readme.md (6 months ago - Mon, 3 Sep 2018 15:18:54 -0400) <calvin kinateder>

bf4ef6f - fixed readme.md (6 months ago - Mon, 3 Sep 2018 15:17:31 -0400) <calvin kinateder>

13214a7 - revisited main layout (6 months ago - Mon, 3 Sep 2018 15:14:40 -0400) <calvin kinateder>

de6b00d - works about to change executor to larger scope ok to revert to (6 months ago - Mon, 3 Sep 2018 14:47:34 -0400) <calvin kinateder>

5c88b89 - about to change to executor.submit - OK TO REVERT (6 months ago - Sun, 2 Sep 2018 07:57:57 -0400) <calvin kinateder>

0a3ba6e - updated ok to revert (6 months ago - Sat, 1 Sep 2018 17:44:33 -0400) <calvin kinateder>

d5edead - source editor works ok to revert still needs work though (6 months ago - Sat, 1 Sep 2018 17:34:47 -0400) <calvin kinateder>

fc762cd - executerservice works just like thread rn. but added sources. multithreaded speed confirmed. (6 months ago - Sat, 1 Sep 2018 15:14:23 -0400) <calvin kinateder>

d212677 - Added some functions for joining threads, still stuck. beginning port to executor service. ok to revert to (6 months ago - Sat, 1 Sep 2018 10:57:40 -0400) <calvin kinateder>

d4fd45c - search works well now extract with threads still a little glitchy ok to revert to i guess (6 months ago - Thu, 30 Aug 2018 20:20:23 -0400) <calvin kinateder>

d6ffb98 - multithread works but still dont revert yet (6 months ago - Thu, 30 Aug 2018 16:45:08 -0400) <calvin kinateder>

4ff8515 - DO NOT REVERT TO THIS COMMIT -- BETA -- ADDED EXTRACTOR CONTAINER (6 months ago - Thu, 30 Aug 2018 15:47:03 -0400) <Calvin>

c6578f6 - single threaded works (6 months ago - Thu, 30 Aug 2018 08:32:00 -0400) <Calvin>

daa5669 - added twitter custom checkbox (6 months ago - Wed, 29 Aug 2018 20:09:42 -0400) <calvin kinateder>

e9d3cc9 - better code (6 months ago - Wed, 29 Aug 2018 14:50:13 -0400) <Calvin>

083a00c - attached remote javadoc to jtwitter: https://www.winterwell.com/software/jtwitter/javadoc/ (6 months ago - Wed, 29 Aug 2018 14:30:14 -0400) <Calvin>

f98f03b - added libs (6 months ago - Wed, 29 Aug 2018 14:11:31 -0400) <Calvin>

fd8ae4b - writes as it goes (6 months ago - Tue, 28 Aug 2018 20:36:58 -0400) <calvin kinateder>

e365048 - writes as it goes (6 months ago - Tue, 28 Aug 2018 20:23:03 -0400) <calvin kinateder>

a75edc0 - Changed main ui (6 months ago - Tue, 28 Aug 2018 15:25:26 -0400) <Calvin>

0b2ffd9 - Fixed broken project (6 months ago - Tue, 28 Aug 2018 14:37:35 -0400) <Calvin>

bc24ed3 - Fixed broken project (6 months ago - Tue, 28 Aug 2018 14:32:05 -0400) <Calvin>

ea2d69a - Delete project.xml (6 months ago - Tue, 28 Aug 2018 14:21:17 -0400) <Calvin Kinateder>

b96d615 - Delete project.properties (6 months ago - Tue, 28 Aug 2018 14:21:11 -0400) <Calvin Kinateder>

e53896a - Delete genfiles.properties (6 months ago - Tue, 28 Aug 2018 14:21:04 -0400) <Calvin Kinateder>

c5ee986 - Delete build-impl.xml (6 months ago - Tue, 28 Aug 2018 14:20:52 -0400) <Calvin Kinateder>

7ce2fd6 - Delete manifest.mf (6 months ago - Tue, 28 Aug 2018 14:20:39 -0400) <Calvin Kinateder>

4ac773d - Delete links.txt (6 months ago - Tue, 28 Aug 2018 14:20:31 -0400) <Calvin Kinateder>

b83606c - Delete build.xml (6 months ago - Tue, 28 Aug 2018 14:20:18 -0400) <Calvin Kinateder>

abbc215 - Delete acountss.txt (6 months ago - Tue, 28 Aug 2018 14:20:08 -0400) <Calvin Kinateder>

81ed1c3 - Delete Java 8 Articles (6 months ago - Tue, 28 Aug 2018 14:20:01 -0400) <Calvin Kinateder>

cd7cd4d - remove storygrab folder (6 months ago - Tue, 28 Aug 2018 14:18:33 -0400) <Calvin>

d2c516c - tst (6 months ago - Tue, 28 Aug 2018 14:16:27 -0400) <Calvin>

6509ccd - extractor works recursively (6 months ago - Tue, 28 Aug 2018 14:11:32 -0400) <Calvin>

32f04c5 - changed extractor (6 months ago - Tue, 28 Aug 2018 13:26:22 -0400) <Calvin>

8b9560b - added links (6 months ago - Tue, 28 Aug 2018 08:25:48 -0400) <Calvin>

32928d7 - added nonfunct xtractor (6 months ago - Mon, 27 Aug 2018 21:23:03 -0400) <calvin kinateder>

713fceb - added period (6 months ago - Mon, 27 Aug 2018 20:37:33 -0400) <calvin kinateder>

d0def41 - added help menu (6 months ago - Mon, 27 Aug 2018 17:00:19 -0400) <calvin kinateder>

bae6355 - fixed last commit (6 months ago - Mon, 27 Aug 2018 09:16:20 -0400) <Calvin>

1a828b3 - Update README.md (6 months ago - Mon, 27 Aug 2018 09:15:05 -0400) <Calvin Kinateder>

dd77a32 - Update README.md (6 months ago - Mon, 27 Aug 2018 09:14:28 -0400) <Calvin Kinateder>

ca3f5e8 - Update README.md (6 months ago - Mon, 27 Aug 2018 09:14:16 -0400) <Calvin Kinateder>

ed28358 - Merge origin/master (6 months ago - Mon, 27 Aug 2018 09:11:24 -0400) <Calvin>

bdc93de - BIG ONE: Removed access to create account panel-it was not necessary and it was insecure (6 months ago - Mon, 27 Aug 2018 09:11:00 -0400) <Calvin>

cb4d40b - Update README.md (6 months ago - Mon, 27 Aug 2018 08:55:14 -0400) <Calvin Kinateder>

0946599 - Update README.md (6 months ago - Mon, 27 Aug 2018 08:54:33 -0400) <Calvin Kinateder>

c58167b - Update README.md (6 months ago - Sun, 26 Aug 2018 20:52:02 -0400) <Calvin Kinateder>

627a131 - added screenshots (6 months ago - Sun, 26 Aug 2018 20:47:04 -0400) <calvin kinateder>

8eefafd - added two icons (6 months ago - Sun, 26 Aug 2018 20:44:35 -0400) <calvin kinateder>

d0aee4f - fixed corners by changing JFrame shape (6 months ago - Sat, 25 Aug 2018 20:35:12 -0400) <calvin kinateder>

0663cf6 - fixed mac issue with spawning pwdfield in create account (6 months ago - Sat, 25 Aug 2018 19:11:21 -0400) <Calvin>

2bb0652 - add logo (6 months ago - Sat, 25 Aug 2018 23:07:07 -0400) <calvin kinateder>

7e02be6 - change author from calvi to my actual name (6 months ago - Sat, 25 Aug 2018 14:04:12 -0400) <calvin kinateder>

3b0c9e6 - Cleaned up functions and improved create accounts screen (6 months ago - Sat, 25 Aug 2018 14:02:18 -0400) <calvi>

37f5a7e - Buttons and labels now work for release too (6 months ago - Sat, 25 Aug 2018 13:20:11 -0400) <calvi>

2a7cff4 - Big one: fixed buttons toggling and create account only works if it fields are filled all the way To do: pressed not clicked for buttons (6 months ago - Sat, 25 Aug 2018 12:15:02 -0400) <calvi>

a3ecd69 - added writing accounts to file and loading (6 months ago - Fri, 24 Aug 2018 22:22:40 -0400) <calvi>

694b1f3 - added create account (6 months ago - Fri, 24 Aug 2018 16:51:48 -0400) <calvi>

b159377 - update readme (6 months ago - Fri, 24 Aug 2018 16:12:39 -0400) <calvi>

81733bc - update readme (6 months ago - Fri, 24 Aug 2018 16:12:01 -0400) <calvi>

ce0b7fb - Added Accountmanager and user classes. (6 months ago - Fri, 24 Aug 2018 15:36:11 -0400) <Calvin>

bd1a92c - added checked and unchecked icons - may not be used (6 months ago - Fri, 24 Aug 2018 13:52:04 -0400) <Calvin>

94521e8 - login creates new screen (6 months ago - Fri, 24 Aug 2018 13:51:20 -0400) <Calvin>

3b0946e - Made it look nicer and added icons (6 months ago - Fri, 24 Aug 2018 09:14:14 -0400) <Calvin>

b55b0c3 - add lib (6 months ago - Fri, 24 Aug 2018 07:29:15 -0400) <calvi>

3030fbf - update readme (6 months ago - Fri, 24 Aug 2018 07:26:50 -0400) <calvi>

2e14007 - update readme (6 months ago - Fri, 24 Aug 2018 07:26:30 -0400) <calvi>

350a022 - Create README.md (6 months ago - Thu, 23 Aug 2018 20:33:16 -0400) <Calvin Kinateder>

abb3afc - Fixed opacity problem and added X (6 months ago - Thu, 23 Aug 2018 20:31:44 -0400) <calvi>

29493ca - First commit (6 months ago - Thu, 23 Aug 2018 19:48:02 -0400) <calvi>

Appendix 4: Code

LoginWindow.java

**package** **main**;

**import** **java.awt.Color**;

**import** **java.awt.Desktop**;

**import** **java.awt.Toolkit**;

**import** **java.awt.geom.RoundRectangle2D**;

**import** **java.io.File**;

**import** **java.io.IOException**;

**import** **java.util.ArrayList**;

**import** **java.util.Vector**;

**import** **java.util.logging.Level**;

**import** **java.util.logging.Logger**;

**import** **javax.swing.SwingWorker**;

**import** **keeptoo.Drag**;

*/\*\**

*\* Main class*

*\* @author Calvin Kinateder*

*\* git log --date=short --pretty=format:"Work on code,%s, ,%ad %ar"*

*\* https://youtu.be/TOgTPm\_AU\_c*

*\**

*\*/*

**public** **class** **LoginWindow** **extends** javax.swing.JFrame {

*/\*\**

*\* Creates new form*

*\*/*

**public** LoginWindow() {

initComponents();

verbosebtn.setSelected(**true**);

verbosebtn.setText("X");

setVerbose(**true**);

setIcon();

panelSwitcher(1);

updateSources();

refreshChild.execute();

}

*/\*\**

*\* This method is called from within the constructor to initialize the form.*

*\*/*

@SuppressWarnings("unchecked")

*// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents*

**private** void initComponents() {

LoginPanel = **new** keeptoo.KGradientPanel();

userfield = **new** javax.swing.JTextField();

pwdfield = **new** javax.swing.JPasswordField();

orlabel = **new** javax.swing.JLabel();

closelbl = **new** javax.swing.JLabel();

helpbtn1 = **new** keeptoo.KButton();

loginbtn = **new** keeptoo.KButton();

profilelblicon = **new** javax.swing.JLabel();

nouserwithname = **new** javax.swing.JLabel();

pwdlbl = **new** javax.swing.JLabel();

mainlogowithicon = **new** javax.swing.JLabel();

userlbl1 = **new** javax.swing.JLabel();

MainPanel = **new** keeptoo.KGradientPanel();

closelbl1 = **new** javax.swing.JLabel();

profileiconlbl = **new** javax.swing.JLabel();

usertitlelbl = **new** javax.swing.JLabel();

logoutbtn1 = **new** keeptoo.KButton();

helpbtn = **new** keeptoo.KButton();

extractorpanel = **new** javax.swing.JPanel();

sourceslist = **new** javax.swing.JLabel();

largeoutput = **new** javax.swing.JLabel();

maincontrolpanel = **new** javax.swing.JPanel();

extractbtn = **new** keeptoo.KButton();

depthtogglepanel = **new** javax.swing.JPanel();

depthlbl = **new** javax.swing.JLabel();

plusdepthbtn = **new** keeptoo.KButton();

dynamicdepthlbl = **new** javax.swing.JLabel();

minusdepthbtn = **new** keeptoo.KButton();

keywordfield = **new** javax.swing.JTextField();

stopbtn = **new** keeptoo.KButton();

statuslbl = **new** javax.swing.JLabel();

htslbl = **new** javax.swing.JLabel();

verbosepanel = **new** javax.swing.JPanel();

cblbl2 = **new** javax.swing.JLabel();

verbosebtn = **new** keeptoo.KButton();

onelineout = **new** javax.swing.JLabel();

websitelbl8 = **new** javax.swing.JLabel();

sourceslbl = **new** javax.swing.JLabel();

addsourcebtn2 = **new** keeptoo.KButton();

addsourcebtn = **new** keeptoo.KButton();

tipsbtnpanel = **new** javax.swing.JPanel();

cblbl3 = **new** javax.swing.JLabel();

tipsbtn = **new** keeptoo.KButton();

stopbtn1 = **new** keeptoo.KButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

setTitle("StoryGrab");

setLocation(**new** java.awt.Point(0, 0));

setLocationByPlatform(**true**);

setUndecorated(**true**);

setShape(**new** RoundRectangle2D.Double(0, 0, 870, 600, 20, 20));

setSize(**new** java.awt.Dimension(800, 600));

getContentPane().setLayout(**new** org.netbeans.lib.awtextra.AbsoluteLayout());

LoginPanel.setBackground(**new** Color(0,0,0,0));

LoginPanel.setBorder(javax.swing.BorderFactory.createEmptyBorder(1, 1, 1, 1));

LoginPanel.setDebugGraphicsOptions(javax.swing.DebugGraphics.NONE\_OPTION);

LoginPanel.setkBorderRadius(20);

LoginPanel.setkEndColor(**new** java.awt.Color(51, 51, 140));

LoginPanel.setkGradientFocus(250);

LoginPanel.setkStartColor(**new** java.awt.Color(83, 97, 112));

LoginPanel.addMouseMotionListener(**new** java.awt.event.MouseMotionAdapter() {

**public** void mouseDragged(java.awt.event.MouseEvent evt) {

LoginPanelMouseDragged(evt);

}

});

LoginPanel.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** void mousePressed(java.awt.event.MouseEvent evt) {

LoginPanelMousePressed(evt);

}

**public** void mouseReleased(java.awt.event.MouseEvent evt) {

LoginPanelMouseReleased(evt);

}

});

LoginPanel.addKeyListener(**new** java.awt.event.KeyAdapter() {

**public** void keyPressed(java.awt.event.KeyEvent evt) {

LoginPanelKeyPressed(evt);

}

});

LoginPanel.setLayout(**new** org.netbeans.lib.awtextra.AbsoluteLayout());

userfield.setBackground(**new** Color(0,0,0,0));

userfield.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

userfield.setForeground(**new** java.awt.Color(255, 255, 255));

userfield.setBorder(javax.swing.BorderFactory.createMatteBorder(0, 0, 2, 0, **new** java.awt.Color(255, 255, 255)));

userfield.setCaretColor(**new** java.awt.Color(255, 255, 255));

userfield.setCursor(**new** java.awt.Cursor(java.awt.Cursor.TEXT\_CURSOR));

userfield.addActionListener(**new** java.awt.event.ActionListener() {

**public** void actionPerformed(java.awt.event.ActionEvent evt) {

userfieldActionPerformed(evt);

}

});

LoginPanel.add(userfield, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(270, 220, 350, 30));

userfield.getAccessibleContext().setAccessibleName("");

pwdfield.setBackground(**new** Color(0,0,0,0));

pwdfield.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

pwdfield.setForeground(**new** java.awt.Color(255, 255, 255));

pwdfield.setBorder(javax.swing.BorderFactory.createMatteBorder(0, 0, 2, 0, **new** java.awt.Color(255, 255, 255)));

pwdfield.setCaretColor(**new** java.awt.Color(255, 255, 255));

LoginPanel.add(pwdfield, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(270, 320, 350, 30));

orlabel.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

orlabel.setForeground(**new** java.awt.Color(255, 255, 255));

orlabel.setText("or");

LoginPanel.add(orlabel, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(440, 460, -1, -1));

closelbl.setFont(**new** java.awt.Font("Segoe UI", 1, 18)); *// NOI18N*

closelbl.setForeground(**new** java.awt.Color(255, 255, 255));

closelbl.setIcon(**new** javax.swing.ImageIcon(getClass().getResource("/icons/icons8-cancel-32.png"))); *// NOI18N*

closelbl.addMouseMotionListener(**new** java.awt.event.MouseMotionAdapter() {

**public** void mouseDragged(java.awt.event.MouseEvent evt) {

closelblMouseDragged(evt);

}

});

closelbl.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** void mousePressed(java.awt.event.MouseEvent evt) {

closelblMousePressed(evt);

}

**public** void mouseReleased(java.awt.event.MouseEvent evt) {

closelblMouseReleased(evt);

}

**public** void mouseClicked(java.awt.event.MouseEvent evt) {

closelblMouseClicked(evt);

}

**public** void mouseEntered(java.awt.event.MouseEvent evt) {

closelblMouseEntered(evt);

}

});

LoginPanel.add(closelbl, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(830, 10, 30, 30));

helpbtn1.setText("Help");

helpbtn1.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

helpbtn1.setkAllowTab(**false**);

helpbtn1.setkBackGroundColor(**new** java.awt.Color(255, 255, 255));

helpbtn1.setkBorderRadius(40);

helpbtn1.setkEndColor(**new** java.awt.Color(0, 204, 51));

helpbtn1.setkHoverEndColor(**new** java.awt.Color(78, 160, 143));

helpbtn1.setkHoverForeGround(**new** java.awt.Color(255, 255, 255));

helpbtn1.setkHoverStartColor(**new** java.awt.Color(63, 167, 89));

helpbtn1.setkIndicatorThickness(0);

helpbtn1.setkPressedColor(**new** java.awt.Color(167, 69, 199));

helpbtn1.setkSelectedColor(**new** java.awt.Color(163, 62, 167));

helpbtn1.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** void mouseClicked(java.awt.event.MouseEvent evt) {

helpbtn1MouseClicked(evt);

}

**public** void mousePressed(java.awt.event.MouseEvent evt) {

helpbtn1MousePressed(evt);

}

**public** void mouseReleased(java.awt.event.MouseEvent evt) {

helpbtn1MouseReleased(evt);

}

});

helpbtn1.addActionListener(**new** java.awt.event.ActionListener() {

**public** void actionPerformed(java.awt.event.ActionEvent evt) {

helpbtn1ActionPerformed(evt);

}

});

helpbtn1.addKeyListener(**new** java.awt.event.KeyAdapter() {

**public** void keyPressed(java.awt.event.KeyEvent evt) {

helpbtn1KeyPressed(evt);

}

});

LoginPanel.add(helpbtn1, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(470, 450, 150, 40));

loginbtn.setText("Login");

loginbtn.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

loginbtn.setkAllowTab(**false**);

loginbtn.setkBorderRadius(40);

loginbtn.setkEndColor(**new** java.awt.Color(0, 204, 51));

loginbtn.setkHoverEndColor(**new** java.awt.Color(78, 160, 143));

loginbtn.setkHoverForeGround(**new** java.awt.Color(255, 255, 255));

loginbtn.setkHoverStartColor(**new** java.awt.Color(63, 167, 89));

loginbtn.setkIndicatorThickness(0);

loginbtn.setkPressedColor(**new** java.awt.Color(167, 69, 199));

loginbtn.setkSelectedColor(**new** java.awt.Color(163, 62, 167));

loginbtn.setName(""); *// NOI18N*

loginbtn.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** void mousePressed(java.awt.event.MouseEvent evt) {

loginbtnMousePressed(evt);

}

**public** void mouseReleased(java.awt.event.MouseEvent evt) {

loginbtnMouseReleased(evt);

}

**public** void mouseClicked(java.awt.event.MouseEvent evt) {

loginbtnMouseClicked(evt);

}

});

loginbtn.addActionListener(**new** java.awt.event.ActionListener() {

**public** void actionPerformed(java.awt.event.ActionEvent evt) {

loginbtnActionPerformed(evt);

}

});

LoginPanel.add(loginbtn, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(270, 450, 150, 40));

profilelblicon.setIcon(**new** javax.swing.ImageIcon(getClass().getResource("/icons/icons8-male-user-50.png"))); *// NOI18N*

LoginPanel.add(profilelblicon, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(410, 90, -1, -1));

nouserwithname.setFont(**new** java.awt.Font("Segoe UI", 0, 12)); *// NOI18N*

nouserwithname.setForeground(**new** java.awt.Color(255, 255, 255));

LoginPanel.add(nouserwithname, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(270, 360, 400, 20));

pwdlbl.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

pwdlbl.setForeground(**new** java.awt.Color(255, 255, 255));

pwdlbl.setText("password");

LoginPanel.add(pwdlbl, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(270, 300, 70, -1));

mainlogowithicon.setFont(**new** java.awt.Font("Ubuntu", 0, 28)); *// NOI18N*

mainlogowithicon.setForeground(**new** java.awt.Color(255, 255, 255));

mainlogowithicon.setIcon(**new** javax.swing.ImageIcon(getClass().getResource("/icons/icons8-news-50.png"))); *// NOI18N*

mainlogowithicon.setText(" StoryGrab");

LoginPanel.add(mainlogowithicon, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(20, 10, 230, 60));

userlbl1.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

userlbl1.setForeground(**new** java.awt.Color(255, 255, 255));

userlbl1.setText("username");

LoginPanel.add(userlbl1, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(270, 200, 70, -1));

getContentPane().add(LoginPanel, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 870, 600));

MainPanel.setBackground(**new** Color(0,0,0,0));

MainPanel.setBorder(javax.swing.BorderFactory.createEmptyBorder(0, 0, 0, 0));

MainPanel.setDebugGraphicsOptions(javax.swing.DebugGraphics.NONE\_OPTION);

MainPanel.setkBorderRadius(20);

MainPanel.setkEndColor(**new** java.awt.Color(51, 51, 140));

MainPanel.setkGradientFocus(250);

MainPanel.setkStartColor(**new** java.awt.Color(83, 97, 112));

MainPanel.setOpaque(**false**);

MainPanel.addMouseMotionListener(**new** java.awt.event.MouseMotionAdapter() {

**public** void mouseDragged(java.awt.event.MouseEvent evt) {

MainPanelMouseDragged(evt);

}

});

MainPanel.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** void mousePressed(java.awt.event.MouseEvent evt) {

MainPanelMousePressed(evt);

}

});

MainPanel.addKeyListener(**new** java.awt.event.KeyAdapter() {

**public** void keyPressed(java.awt.event.KeyEvent evt) {

MainPanelKeyPressed(evt);

}

});

MainPanel.setLayout(**new** org.netbeans.lib.awtextra.AbsoluteLayout());

closelbl1.setFont(**new** java.awt.Font("Segoe UI", 1, 18)); *// NOI18N*

closelbl1.setForeground(**new** java.awt.Color(255, 255, 255));

closelbl1.setIcon(**new** javax.swing.ImageIcon(getClass().getResource("/icons/icons8-cancel-32.png"))); *// NOI18N*

closelbl1.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** void mouseReleased(java.awt.event.MouseEvent evt) {

closelbl1MouseReleased(evt);

}

**public** void mouseClicked(java.awt.event.MouseEvent evt) {

closelbl1MouseClicked(evt);

}

**public** void mouseEntered(java.awt.event.MouseEvent evt) {

closelbl1MouseEntered(evt);

}

});

MainPanel.add(closelbl1, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(830, 10, 30, 30));

profileiconlbl.setIcon(**new** javax.swing.ImageIcon(getClass().getResource("/icons/icons8-news-50.png"))); *// NOI18N*

MainPanel.add(profileiconlbl, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(20, 10, -1, 60));

usertitlelbl.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

usertitlelbl.setForeground(**new** java.awt.Color(255, 255, 255));

usertitlelbl.setText("ckinateder");

MainPanel.add(usertitlelbl, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(80, 30, 320, 30));

logoutbtn1.setText("Logout");

logoutbtn1.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

logoutbtn1.setkAllowTab(**false**);

logoutbtn1.setkBorderRadius(40);

logoutbtn1.setkEndColor(**new** java.awt.Color(0, 204, 51));

logoutbtn1.setkFillButton(**false**);

logoutbtn1.setkHoverColor(**new** java.awt.Color(255, 255, 0));

logoutbtn1.setkHoverEndColor(**new** java.awt.Color(78, 160, 143));

logoutbtn1.setkHoverForeGround(**new** java.awt.Color(255, 255, 255));

logoutbtn1.setkHoverStartColor(**new** java.awt.Color(63, 167, 89));

logoutbtn1.setkIndicatorThickness(0);

logoutbtn1.setkPressedColor(**new** java.awt.Color(167, 69, 199));

logoutbtn1.setkSelectedColor(**new** java.awt.Color(163, 62, 167));

logoutbtn1.setName(""); *// NOI18N*

logoutbtn1.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** void mousePressed(java.awt.event.MouseEvent evt) {

logoutbtn1MousePressed(evt);

}

**public** void mouseReleased(java.awt.event.MouseEvent evt) {

logoutbtn1MouseReleased(evt);

}

**public** void mouseClicked(java.awt.event.MouseEvent evt) {

logoutbtn1MouseClicked(evt);

}

});

logoutbtn1.addActionListener(**new** java.awt.event.ActionListener() {

**public** void actionPerformed(java.awt.event.ActionEvent evt) {

logoutbtn1ActionPerformed(evt);

}

});

MainPanel.add(logoutbtn1, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(10, 550, 100, 40));

helpbtn.setText("Help");

helpbtn.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

helpbtn.setkAllowTab(**false**);

helpbtn.setkBackGroundColor(**new** java.awt.Color(255, 255, 255));

helpbtn.setkBorderRadius(40);

helpbtn.setkEndColor(**new** java.awt.Color(0, 204, 51));

helpbtn.setkHoverEndColor(**new** java.awt.Color(78, 160, 143));

helpbtn.setkHoverForeGround(**new** java.awt.Color(255, 255, 255));

helpbtn.setkHoverStartColor(**new** java.awt.Color(63, 167, 89));

helpbtn.setkIndicatorThickness(0);

helpbtn.setkPressedColor(**new** java.awt.Color(167, 69, 199));

helpbtn.setkSelectedColor(**new** java.awt.Color(163, 62, 167));

helpbtn.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** void mouseClicked(java.awt.event.MouseEvent evt) {

helpbtnMouseClicked(evt);

}

**public** void mousePressed(java.awt.event.MouseEvent evt) {

helpbtnMousePressed(evt);

}

**public** void mouseReleased(java.awt.event.MouseEvent evt) {

helpbtnMouseReleased(evt);

}

});

helpbtn.addActionListener(**new** java.awt.event.ActionListener() {

**public** void actionPerformed(java.awt.event.ActionEvent evt) {

helpbtnActionPerformed(evt);

}

});

helpbtn.addKeyListener(**new** java.awt.event.KeyAdapter() {

**public** void keyPressed(java.awt.event.KeyEvent evt) {

helpbtnKeyPressed(evt);

}

});

MainPanel.add(helpbtn, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(350, 550, 120, 40));

extractorpanel.setOpaque(**false**);

extractorpanel.setLayout(**new** org.netbeans.lib.awtextra.AbsoluteLayout());

sourceslist.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

sourceslist.setForeground(**new** java.awt.Color(255, 255, 255));

sourceslist.setText("Add a source");

sourceslist.setVerticalAlignment(javax.swing.SwingConstants.TOP);

sourceslist.setBorder(javax.swing.BorderFactory.createMatteBorder(0, 0, 0, 0, **new** java.awt.Color(255, 255, 255)));

sourceslist.addMouseWheelListener(**new** java.awt.event.MouseWheelListener() {

**public** void mouseWheelMoved(java.awt.event.MouseWheelEvent evt) {

sourceslistMouseWheelMoved(evt);

}

});

extractorpanel.add(sourceslist, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(30, 30, 300, 210));

largeoutput.setFont(**new** java.awt.Font("Ubuntu Mono", 0, 14)); *// NOI18N*

largeoutput.setForeground(**new** java.awt.Color(255, 255, 255));

largeoutput.setVerticalAlignment(javax.swing.SwingConstants.TOP);

largeoutput.setBorder(javax.swing.BorderFactory.createMatteBorder(0, 0, 0, 0, **new** java.awt.Color(255, 255, 255)));

extractorpanel.add(largeoutput, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(330, 30, 470, 210));

maincontrolpanel.setOpaque(**false**);

maincontrolpanel.setLayout(**new** org.netbeans.lib.awtextra.AbsoluteLayout());

extractbtn.setText("Extract");

extractbtn.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

extractbtn.setkAllowTab(**false**);

extractbtn.setkBorderRadius(40);

extractbtn.setkEndColor(**new** java.awt.Color(0, 204, 51));

extractbtn.setkHoverColor(**new** java.awt.Color(255, 102, 204));

extractbtn.setkHoverEndColor(**new** java.awt.Color(78, 160, 143));

extractbtn.setkHoverForeGround(**new** java.awt.Color(255, 255, 255));

extractbtn.setkHoverStartColor(**new** java.awt.Color(63, 167, 89));

extractbtn.setkIndicatorThickness(0);

extractbtn.setkPressedColor(**new** java.awt.Color(167, 69, 199));

extractbtn.setkSelectedColor(**new** java.awt.Color(163, 62, 167));

extractbtn.setName(""); *// NOI18N*

extractbtn.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** void mousePressed(java.awt.event.MouseEvent evt) {

extractbtnMousePressed(evt);

}

**public** void mouseReleased(java.awt.event.MouseEvent evt) {

extractbtnMouseReleased(evt);

}

**public** void mouseClicked(java.awt.event.MouseEvent evt) {

extractbtnMouseClicked(evt);

}

});

extractbtn.addActionListener(**new** java.awt.event.ActionListener() {

**public** void actionPerformed(java.awt.event.ActionEvent evt) {

extractbtnActionPerformed(evt);

}

});

maincontrolpanel.add(extractbtn, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(0, 120, 100, 40));

depthtogglepanel.setOpaque(**false**);

depthlbl.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

depthlbl.setForeground(**new** java.awt.Color(255, 255, 255));

depthlbl.setText("Depth");

plusdepthbtn.setText("+");

plusdepthbtn.setFont(**new** java.awt.Font("Ubuntu", 0, 30)); *// NOI18N*

plusdepthbtn.setkAllowTab(**false**);

plusdepthbtn.setkBorderRadius(40);

plusdepthbtn.setkEndColor(**new** java.awt.Color(0, 204, 51));

plusdepthbtn.setkHoverEndColor(**new** java.awt.Color(78, 160, 143));

plusdepthbtn.setkHoverForeGround(**new** java.awt.Color(255, 255, 255));

plusdepthbtn.setkHoverStartColor(**new** java.awt.Color(63, 167, 89));

plusdepthbtn.setkIndicatorThickness(0);

plusdepthbtn.setkPressedColor(**new** java.awt.Color(167, 69, 199));

plusdepthbtn.setkSelectedColor(**new** java.awt.Color(163, 62, 167));

plusdepthbtn.setName(""); *// NOI18N*

plusdepthbtn.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** void mouseClicked(java.awt.event.MouseEvent evt) {

plusdepthbtnMouseClicked(evt);

}

**public** void mousePressed(java.awt.event.MouseEvent evt) {

plusdepthbtnMousePressed(evt);

}

**public** void mouseReleased(java.awt.event.MouseEvent evt) {

plusdepthbtnMouseReleased(evt);

}

});

plusdepthbtn.addActionListener(**new** java.awt.event.ActionListener() {

**public** void actionPerformed(java.awt.event.ActionEvent evt) {

plusdepthbtnActionPerformed(evt);

}

});

dynamicdepthlbl.setFont(**new** java.awt.Font("Segoe UI", 1, 18)); *// NOI18N*

dynamicdepthlbl.setForeground(**new** java.awt.Color(255, 255, 255));

dynamicdepthlbl.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

dynamicdepthlbl.setText("0");

minusdepthbtn.setText("-");

minusdepthbtn.setToolTipText("");

minusdepthbtn.setFont(**new** java.awt.Font("Ubuntu", 0, 36)); *// NOI18N*

minusdepthbtn.setkAllowTab(**false**);

minusdepthbtn.setkBorderRadius(40);

minusdepthbtn.setkEndColor(**new** java.awt.Color(0, 204, 51));

minusdepthbtn.setkHoverEndColor(**new** java.awt.Color(78, 160, 143));

minusdepthbtn.setkHoverForeGround(**new** java.awt.Color(255, 255, 255));

minusdepthbtn.setkHoverStartColor(**new** java.awt.Color(63, 167, 89));

minusdepthbtn.setkIndicatorThickness(0);

minusdepthbtn.setkPressedColor(**new** java.awt.Color(167, 69, 199));

minusdepthbtn.setkSelectedColor(**new** java.awt.Color(163, 62, 167));

minusdepthbtn.setName(""); *// NOI18N*

minusdepthbtn.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** void mouseClicked(java.awt.event.MouseEvent evt) {

minusdepthbtnMouseClicked(evt);

}

**public** void mousePressed(java.awt.event.MouseEvent evt) {

minusdepthbtnMousePressed(evt);

}

**public** void mouseReleased(java.awt.event.MouseEvent evt) {

minusdepthbtnMouseReleased(evt);

}

});

minusdepthbtn.addActionListener(**new** java.awt.event.ActionListener() {

**public** void actionPerformed(java.awt.event.ActionEvent evt) {

minusdepthbtnActionPerformed(evt);

}

});

javax.swing.GroupLayout depthtogglepanelLayout = **new** javax.swing.GroupLayout(depthtogglepanel);

depthtogglepanel.setLayout(depthtogglepanelLayout);

depthtogglepanelLayout.setHorizontalGroup(

depthtogglepanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(depthtogglepanelLayout.createSequentialGroup()

.addGap(4, 4, 4)

.addComponent(depthlbl)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(minusdepthbtn, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(5, 5, 5)

.addComponent(dynamicdepthlbl, javax.swing.GroupLayout.PREFERRED\_SIZE, 19, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(plusdepthbtn, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 10, Short.MAX\_VALUE))

);

depthtogglepanelLayout.setVerticalGroup(

depthtogglepanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(depthtogglepanelLayout.createSequentialGroup()

.addGroup(depthtogglepanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, **false**)

.addComponent(plusdepthbtn, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(dynamicdepthlbl, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(minusdepthbtn, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(depthlbl, javax.swing.GroupLayout.Alignment.LEADING, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

.addGap(0, 5, Short.MAX\_VALUE))

);

maincontrolpanel.add(depthtogglepanel, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(0, 70, -1, -1));

keywordfield.setBackground(**new** Color(0,0,0,0));

keywordfield.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

keywordfield.setForeground(**new** java.awt.Color(255, 255, 255));

keywordfield.setText("trump");

keywordfield.setBorder(javax.swing.BorderFactory.createMatteBorder(0, 0, 2, 0, **new** java.awt.Color(255, 255, 255)));

keywordfield.setCaretColor(**new** java.awt.Color(255, 255, 255));

keywordfield.setCursor(**new** java.awt.Cursor(java.awt.Cursor.TEXT\_CURSOR));

keywordfield.addActionListener(**new** java.awt.event.ActionListener() {

**public** void actionPerformed(java.awt.event.ActionEvent evt) {

keywordfieldActionPerformed(evt);

}

});

maincontrolpanel.add(keywordfield, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(0, 20, 440, 30));

stopbtn.setText("Stop");

stopbtn.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

stopbtn.setkAllowTab(**false**);

stopbtn.setkBorderRadius(40);

stopbtn.setkEndColor(**new** java.awt.Color(0, 204, 51));

stopbtn.setkHoverColor(**new** java.awt.Color(255, 102, 204));

stopbtn.setkHoverEndColor(**new** java.awt.Color(78, 160, 143));

stopbtn.setkHoverForeGround(**new** java.awt.Color(255, 255, 255));

stopbtn.setkHoverStartColor(**new** java.awt.Color(63, 167, 89));

stopbtn.setkIndicatorThickness(0);

stopbtn.setkPressedColor(**new** java.awt.Color(167, 69, 199));

stopbtn.setkSelectedColor(**new** java.awt.Color(163, 62, 167));

stopbtn.setName(""); *// NOI18N*

stopbtn.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** void mousePressed(java.awt.event.MouseEvent evt) {

stopbtnMousePressed(evt);

}

**public** void mouseReleased(java.awt.event.MouseEvent evt) {

stopbtnMouseReleased(evt);

}

**public** void mouseClicked(java.awt.event.MouseEvent evt) {

stopbtnMouseClicked(evt);

}

});

stopbtn.addActionListener(**new** java.awt.event.ActionListener() {

**public** void actionPerformed(java.awt.event.ActionEvent evt) {

stopbtnActionPerformed(evt);

}

});

maincontrolpanel.add(stopbtn, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(120, 120, 100, 40));

statuslbl.setFont(**new** java.awt.Font("Segoe UI", 0, 14)); *// NOI18N*

statuslbl.setForeground(**new** java.awt.Color(255, 255, 255));

maincontrolpanel.add(statuslbl, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(240, 130, 130, 20));

htslbl.setFont(**new** java.awt.Font("Ubuntu", 0, 18)); *// NOI18N*

htslbl.setForeground(**new** java.awt.Color(255, 255, 255));

maincontrolpanel.add(htslbl, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(720, 0, 60, 30));

verbosepanel.setOpaque(**false**);

cblbl2.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

cblbl2.setForeground(**new** java.awt.Color(255, 255, 255));

cblbl2.setText("Verbose");

verbosebtn.setFont(**new** java.awt.Font("Ubuntu", 1, 14)); *// NOI18N*

verbosebtn.setIconTextGap(0);

verbosebtn.setkAllowTab(**false**);

verbosebtn.setkEndColor(**new** java.awt.Color(0, 204, 51));

verbosebtn.setkHoverColor(**new** java.awt.Color(249, 153, 217));

verbosebtn.setkHoverEndColor(**new** java.awt.Color(78, 160, 143));

verbosebtn.setkHoverForeGround(**new** java.awt.Color(255, 255, 255));

verbosebtn.setkHoverStartColor(**new** java.awt.Color(63, 167, 89));

verbosebtn.setkIndicatorThickness(0);

verbosebtn.setkPressedColor(**new** java.awt.Color(163, 62, 167));

verbosebtn.setkSelectedColor(**new** java.awt.Color(163, 62, 167));

verbosebtn.setMargin(**new** java.awt.Insets(2, 2, 2, 2));

verbosebtn.setName(""); *// NOI18N*

verbosebtn.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** void mouseClicked(java.awt.event.MouseEvent evt) {

verbosebtnMouseClicked(evt);

}

**public** void mousePressed(java.awt.event.MouseEvent evt) {

verbosebtnMousePressed(evt);

}

**public** void mouseReleased(java.awt.event.MouseEvent evt) {

verbosebtnMouseReleased(evt);

}

});

verbosebtn.addActionListener(**new** java.awt.event.ActionListener() {

**public** void actionPerformed(java.awt.event.ActionEvent evt) {

verbosebtnActionPerformed(evt);

}

});

javax.swing.GroupLayout verbosepanelLayout = **new** javax.swing.GroupLayout(verbosepanel);

verbosepanel.setLayout(verbosepanelLayout);

verbosepanelLayout.setHorizontalGroup(

verbosepanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(verbosepanelLayout.createSequentialGroup()

.addGap(1, 1, 1)

.addComponent(verbosebtn, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(cblbl2)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

verbosepanelLayout.setVerticalGroup(

verbosepanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(verbosepanelLayout.createSequentialGroup()

.addGroup(verbosepanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, **false**)

.addComponent(verbosebtn, javax.swing.GroupLayout.PREFERRED\_SIZE, 0, Short.MAX\_VALUE)

.addComponent(cblbl2, javax.swing.GroupLayout.DEFAULT\_SIZE, 25, Short.MAX\_VALUE))

.addGap(0, 5, Short.MAX\_VALUE))

);

maincontrolpanel.add(verbosepanel, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(160, 70, 90, -1));

onelineout.setFont(**new** java.awt.Font("Ubuntu Mono", 0, 14)); *// NOI18N*

onelineout.setForeground(**new** java.awt.Color(255, 255, 255));

maincontrolpanel.add(onelineout, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(370, 120, 410, 40));

websitelbl8.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

websitelbl8.setForeground(**new** java.awt.Color(255, 255, 255));

websitelbl8.setText("Keyword");

maincontrolpanel.add(websitelbl8, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 90, -1));

extractorpanel.add(maincontrolpanel, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(20, 260, 780, 180));

sourceslbl.setFont(**new** java.awt.Font("Ubuntu", 1, 18)); *// NOI18N*

sourceslbl.setForeground(**new** java.awt.Color(255, 255, 255));

sourceslbl.setText("Sources");

extractorpanel.add(sourceslbl, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(20, 0, 100, 30));

MainPanel.add(extractorpanel, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(20, 100, 810, 440));

addsourcebtn2.setText("Output");

addsourcebtn2.setEnabled(**false**);

addsourcebtn2.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

addsourcebtn2.setkAllowTab(**false**);

addsourcebtn2.setkBorderRadius(40);

addsourcebtn2.setkEndColor(**new** java.awt.Color(0, 204, 51));

addsourcebtn2.setkHoverColor(**new** java.awt.Color(255, 102, 204));

addsourcebtn2.setkHoverEndColor(**new** java.awt.Color(78, 160, 143));

addsourcebtn2.setkHoverForeGround(**new** java.awt.Color(255, 255, 255));

addsourcebtn2.setkHoverStartColor(**new** java.awt.Color(63, 167, 89));

addsourcebtn2.setkIndicatorThickness(0);

addsourcebtn2.setkPressedColor(**new** java.awt.Color(167, 69, 199));

addsourcebtn2.setkSelectedColor(**new** java.awt.Color(163, 62, 167));

addsourcebtn2.setName(""); *// NOI18N*

addsourcebtn2.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** void mousePressed(java.awt.event.MouseEvent evt) {

addsourcebtn2MousePressed(evt);

}

**public** void mouseReleased(java.awt.event.MouseEvent evt) {

addsourcebtn2MouseReleased(evt);

}

**public** void mouseClicked(java.awt.event.MouseEvent evt) {

addsourcebtn2MouseClicked(evt);

}

});

addsourcebtn2.addActionListener(**new** java.awt.event.ActionListener() {

**public** void actionPerformed(java.awt.event.ActionEvent evt) {

addsourcebtn2ActionPerformed(evt);

}

});

MainPanel.add(addsourcebtn2, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(630, 550, 100, 40));

addsourcebtn.setText("Sources Editor");

addsourcebtn.setEnabled(**false**);

addsourcebtn.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

addsourcebtn.setkAllowTab(**false**);

addsourcebtn.setkBorderRadius(40);

addsourcebtn.setkEndColor(**new** java.awt.Color(0, 204, 51));

addsourcebtn.setkHoverColor(**new** java.awt.Color(255, 102, 204));

addsourcebtn.setkHoverEndColor(**new** java.awt.Color(78, 160, 143));

addsourcebtn.setkHoverForeGround(**new** java.awt.Color(255, 255, 255));

addsourcebtn.setkHoverStartColor(**new** java.awt.Color(63, 167, 89));

addsourcebtn.setkIndicatorThickness(0);

addsourcebtn.setkPressedColor(**new** java.awt.Color(167, 69, 199));

addsourcebtn.setkSelectedColor(**new** java.awt.Color(163, 62, 167));

addsourcebtn.setName(""); *// NOI18N*

addsourcebtn.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** void mouseClicked(java.awt.event.MouseEvent evt) {

addsourcebtnMouseClicked(evt);

}

**public** void mousePressed(java.awt.event.MouseEvent evt) {

addsourcebtnMousePressed(evt);

}

**public** void mouseReleased(java.awt.event.MouseEvent evt) {

addsourcebtnMouseReleased(evt);

}

});

addsourcebtn.addActionListener(**new** java.awt.event.ActionListener() {

**public** void actionPerformed(java.awt.event.ActionEvent evt) {

addsourcebtnActionPerformed(evt);

}

});

MainPanel.add(addsourcebtn, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(490, 550, 120, 40));

tipsbtnpanel.setOpaque(**false**);

cblbl3.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

cblbl3.setForeground(**new** java.awt.Color(255, 255, 255));

cblbl3.setText("Tips");

tipsbtn.setFont(**new** java.awt.Font("Ubuntu", 1, 14)); *// NOI18N*

tipsbtn.setIconTextGap(0);

tipsbtn.setkAllowTab(**false**);

tipsbtn.setkEndColor(**new** java.awt.Color(0, 204, 51));

tipsbtn.setkHoverColor(**new** java.awt.Color(249, 153, 217));

tipsbtn.setkHoverEndColor(**new** java.awt.Color(78, 160, 143));

tipsbtn.setkHoverForeGround(**new** java.awt.Color(255, 255, 255));

tipsbtn.setkHoverStartColor(**new** java.awt.Color(63, 167, 89));

tipsbtn.setkIndicatorThickness(0);

tipsbtn.setkPressedColor(**new** java.awt.Color(163, 62, 167));

tipsbtn.setkSelectedColor(**new** java.awt.Color(163, 62, 167));

tipsbtn.setMargin(**new** java.awt.Insets(2, 2, 2, 2));

tipsbtn.setName(""); *// NOI18N*

tipsbtn.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** void mouseClicked(java.awt.event.MouseEvent evt) {

tipsbtnMouseClicked(evt);

}

**public** void mousePressed(java.awt.event.MouseEvent evt) {

tipsbtnMousePressed(evt);

}

**public** void mouseReleased(java.awt.event.MouseEvent evt) {

tipsbtnMouseReleased(evt);

}

});

tipsbtn.addActionListener(**new** java.awt.event.ActionListener() {

**public** void actionPerformed(java.awt.event.ActionEvent evt) {

tipsbtnActionPerformed(evt);

}

});

javax.swing.GroupLayout tipsbtnpanelLayout = **new** javax.swing.GroupLayout(tipsbtnpanel);

tipsbtnpanel.setLayout(tipsbtnpanelLayout);

tipsbtnpanelLayout.setHorizontalGroup(

tipsbtnpanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(tipsbtnpanelLayout.createSequentialGroup()

.addGap(1, 1, 1)

.addComponent(tipsbtn, javax.swing.GroupLayout.PREFERRED\_SIZE, 25, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(cblbl3)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

tipsbtnpanelLayout.setVerticalGroup(

tipsbtnpanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(tipsbtnpanelLayout.createSequentialGroup()

.addGroup(tipsbtnpanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, **false**)

.addComponent(tipsbtn, javax.swing.GroupLayout.PREFERRED\_SIZE, 0, Short.MAX\_VALUE)

.addComponent(cblbl3, javax.swing.GroupLayout.DEFAULT\_SIZE, 25, Short.MAX\_VALUE))

.addGap(0, 5, Short.MAX\_VALUE))

);

MainPanel.add(tipsbtnpanel, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(130, 560, -1, -1));

stopbtn1.setText("Quit");

stopbtn1.setFont(**new** java.awt.Font("Ubuntu", 0, 14)); *// NOI18N*

stopbtn1.setkAllowTab(**false**);

stopbtn1.setkBorderRadius(40);

stopbtn1.setkEndColor(**new** java.awt.Color(0, 204, 51));

stopbtn1.setkHoverColor(**new** java.awt.Color(255, 102, 204));

stopbtn1.setkHoverEndColor(**new** java.awt.Color(78, 160, 143));

stopbtn1.setkHoverForeGround(**new** java.awt.Color(255, 255, 255));

stopbtn1.setkHoverStartColor(**new** java.awt.Color(63, 167, 89));

stopbtn1.setkIndicatorThickness(0);

stopbtn1.setkPressedColor(**new** java.awt.Color(167, 69, 199));

stopbtn1.setkSelectedColor(**new** java.awt.Color(163, 62, 167));

stopbtn1.setName(""); *// NOI18N*

stopbtn1.addMouseListener(**new** java.awt.event.MouseAdapter() {

**public** void mousePressed(java.awt.event.MouseEvent evt) {

stopbtn1MousePressed(evt);

}

**public** void mouseReleased(java.awt.event.MouseEvent evt) {

stopbtn1MouseReleased(evt);

}

**public** void mouseClicked(java.awt.event.MouseEvent evt) {

stopbtn1MouseClicked(evt);

}

});

stopbtn1.addActionListener(**new** java.awt.event.ActionListener() {

**public** void actionPerformed(java.awt.event.ActionEvent evt) {

stopbtn1ActionPerformed(evt);

}

});

MainPanel.add(stopbtn1, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(750, 550, 100, 40));

getContentPane().add(MainPanel, **new** org.netbeans.lib.awtextra.AbsoluteConstraints(0, 0, 870, 600));

pack();

}*// </editor-fold>//GEN-END:initComponents*

*// <editor-fold defaultstate="collapsed" desc="Label and Button Functions">*

**private** void userfieldActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_userfieldActionPerformed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_userfieldActionPerformed*

**private** void LoginPanelKeyPressed(java.awt.event.KeyEvent evt) {*//GEN-FIRST:event\_LoginPanelKeyPressed*

*// TODO add your handling code here:*

*//*

}*//GEN-LAST:event\_LoginPanelKeyPressed*

**private** void LoginPanelMouseDragged(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_LoginPanelMouseDragged*

*// TODO add your handling code here:*

*//drag here*

**new** Drag(LoginPanel).moveWindow(evt);

}*//GEN-LAST:event\_LoginPanelMouseDragged*

**private** void LoginPanelMousePressed(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_LoginPanelMousePressed*

*// TODO add your handling code here:*

**new** Drag(LoginPanel).onPress(evt);

*//use mouse always*

}*//GEN-LAST:event\_LoginPanelMousePressed*

**private** void closelblMouseClicked(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_closelblMouseClicked*

*// TODO add your handling code here:*

quit();

}*//GEN-LAST:event\_closelblMouseClicked*

**private** void closelblMouseEntered(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_closelblMouseEntered*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_closelblMouseEntered*

**private** void helpbtn1ActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_helpbtn1ActionPerformed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_helpbtn1ActionPerformed*

**private** void loginbtnActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_loginbtnActionPerformed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_loginbtnActionPerformed*

**private** void loginbtnMouseClicked(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_loginbtnMouseClicked*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_loginbtnMouseClicked*

**private** void closelbl1MouseClicked(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_closelbl1MouseClicked*

*// TODO add your handling code here:*

quit();

}*//GEN-LAST:event\_closelbl1MouseClicked*

**private** void closelbl1MouseEntered(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_closelbl1MouseEntered*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_closelbl1MouseEntered*

**private** void MainPanelMouseDragged(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_MainPanelMouseDragged*

*// TODO add your handling code here:*

**new** Drag(LoginPanel).moveWindow(evt);

}*//GEN-LAST:event\_MainPanelMouseDragged*

**private** void MainPanelMousePressed(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_MainPanelMousePressed*

*// TODO add your handling code here:*

**new** Drag(LoginPanel).onPress(evt);

}*//GEN-LAST:event\_MainPanelMousePressed*

**private** void MainPanelKeyPressed(java.awt.event.KeyEvent evt) {*//GEN-FIRST:event\_MainPanelKeyPressed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_MainPanelKeyPressed*

**private** void helpbtn1KeyPressed(java.awt.event.KeyEvent evt) {*//GEN-FIRST:event\_helpbtn1KeyPressed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_helpbtn1KeyPressed*

**private** void helpbtn1MouseClicked(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_helpbtn1MouseClicked*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_helpbtn1MouseClicked*

**private** void loginbtnMousePressed(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_loginbtnMousePressed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_loginbtnMousePressed*

**private** void closelblMousePressed(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_closelblMousePressed*

*// TODO add your handling code here:*

*//quit();*

}*//GEN-LAST:event\_closelblMousePressed*

**private** void closelblMouseDragged(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_closelblMouseDragged*

*// TODO add your handling code here:*

quit();

}*//GEN-LAST:event\_closelblMouseDragged*

**private** void closelblMouseReleased(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_closelblMouseReleased*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_closelblMouseReleased*

**private** void helpbtn1MousePressed(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_helpbtn1MousePressed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_helpbtn1MousePressed*

**private** void helpbtn1MouseReleased(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_helpbtn1MouseReleased*

*// TODO add your handling code here:*

*//resetAllFields();*

*//panelSwitcher(3);*

help();

}*//GEN-LAST:event\_helpbtn1MouseReleased*

**private** void loginbtnMouseReleased(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_loginbtnMouseReleased*

*// TODO add your handling code here:*

*//CreateAccountPanel.setVisible(false);*

*//change to ref not val*

login();

}*//GEN-LAST:event\_loginbtnMouseReleased*

**private** void closelbl1MouseReleased(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_closelbl1MouseReleased*

*// TODO add your handling code here:*

quit();

}*//GEN-LAST:event\_closelbl1MouseReleased*

**private** void LoginPanelMouseReleased(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_LoginPanelMouseReleased*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_LoginPanelMouseReleased*

**private** void logoutbtn1MouseClicked(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_logoutbtn1MouseClicked*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_logoutbtn1MouseClicked*

**private** void logoutbtn1MousePressed(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_logoutbtn1MousePressed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_logoutbtn1MousePressed*

**private** void logoutbtn1MouseReleased(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_logoutbtn1MouseReleased*

*// TODO add your handling code here:*

logout();

}*//GEN-LAST:event\_logoutbtn1MouseReleased*

**private** void logoutbtn1ActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_logoutbtn1ActionPerformed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_logoutbtn1ActionPerformed*

**private** void minusdepthbtnMousePressed(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_minusdepthbtnMousePressed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_minusdepthbtnMousePressed*

**private** void minusdepthbtnMouseReleased(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_minusdepthbtnMouseReleased*

*// TODO add your handling code here:*

changeDepth(-1);

dynamicdepthlbl.setText(""+loader.getMaxDepth());

}*//GEN-LAST:event\_minusdepthbtnMouseReleased*

**private** void minusdepthbtnMouseClicked(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_minusdepthbtnMouseClicked*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_minusdepthbtnMouseClicked*

**private** void minusdepthbtnActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_minusdepthbtnActionPerformed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_minusdepthbtnActionPerformed*

**private** void extractbtnMousePressed(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_extractbtnMousePressed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_extractbtnMousePressed*

**private** void extractbtnMouseReleased(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_extractbtnMouseReleased*

**try** {

*// TODO add your handling code here:*

extract();

} **catch** (InterruptedException ex) {

Logger.getLogger(LoginWindow.class.getName()).log(Level.SEVERE, **null**, ex);

}

}*//GEN-LAST:event\_extractbtnMouseReleased*

**private** void extractbtnMouseClicked(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_extractbtnMouseClicked*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_extractbtnMouseClicked*

**private** void extractbtnActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_extractbtnActionPerformed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_extractbtnActionPerformed*

**private** void plusdepthbtnMousePressed(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_plusdepthbtnMousePressed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_plusdepthbtnMousePressed*

**private** void plusdepthbtnMouseReleased(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_plusdepthbtnMouseReleased*

*// TODO add your handling code here:*

changeDepth(1);

dynamicdepthlbl.setText(""+loader.getMaxDepth());

}*//GEN-LAST:event\_plusdepthbtnMouseReleased*

**private** void plusdepthbtnMouseClicked(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_plusdepthbtnMouseClicked*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_plusdepthbtnMouseClicked*

**private** void plusdepthbtnActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_plusdepthbtnActionPerformed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_plusdepthbtnActionPerformed*

**private** void keywordfieldActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_keywordfieldActionPerformed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_keywordfieldActionPerformed*

**private** void stopbtnMouseClicked(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_stopbtnMouseClicked*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_stopbtnMouseClicked*

**private** void stopbtnMousePressed(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_stopbtnMousePressed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_stopbtnMousePressed*

**private** void stopbtnMouseReleased(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_stopbtnMouseReleased*

**try** {

*// TODO add your handling code here:*

stopExtract();

} **catch** (InterruptedException ex) {

Logger.getLogger(LoginWindow.class.getName()).log(Level.SEVERE, **null**, ex);

}

}*//GEN-LAST:event\_stopbtnMouseReleased*

**private** void stopbtnActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_stopbtnActionPerformed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_stopbtnActionPerformed*

**private** void addsourcebtn2MouseClicked(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_addsourcebtn2MouseClicked*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_addsourcebtn2MouseClicked*

**private** void addsourcebtn2MousePressed(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_addsourcebtn2MousePressed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_addsourcebtn2MousePressed*

**private** void addsourcebtn2MouseReleased(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_addsourcebtn2MouseReleased*

**try** {

*// TODO add your handling code here:*

openOutput();

} **catch** (IOException ex) {

Logger.getLogger(LoginWindow.class.getName()).log(Level.SEVERE, **null**, ex);

}

}*//GEN-LAST:event\_addsourcebtn2MouseReleased*

**private** void addsourcebtn2ActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_addsourcebtn2ActionPerformed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_addsourcebtn2ActionPerformed*

**private** void sourceslistMouseWheelMoved(java.awt.event.MouseWheelEvent evt) {*//GEN-FIRST:event\_sourceslistMouseWheelMoved*

*// TODO add your handling code here:*

sourceScroller.scrollSources(evt);

}*//GEN-LAST:event\_sourceslistMouseWheelMoved*

**private** void verbosebtnMouseClicked(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_verbosebtnMouseClicked*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_verbosebtnMouseClicked*

**private** void verbosebtnMousePressed(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_verbosebtnMousePressed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_verbosebtnMousePressed*

**private** void verbosebtnMouseReleased(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_verbosebtnMouseReleased*

*// TODO add your handling code here:*

**if**(verbosebtn.isSelected()){

verbosebtn.setSelected(**false**);

verbosebtn.setText("");

setVerbose(**false**);

}

**else**{

verbosebtn.setSelected(**true**);

verbosebtn.setText("X");

setVerbose(**true**);

}

}*//GEN-LAST:event\_verbosebtnMouseReleased*

**private** void verbosebtnActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_verbosebtnActionPerformed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_verbosebtnActionPerformed*

**private** void tipsbtnMousePressed(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_tipsbtnMousePressed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_tipsbtnMousePressed*

**private** void tipsbtnMouseReleased(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_tipsbtnMouseReleased*

*// TODO add your handling code here:*

**if**(tipsbtn.isSelected()){

tipsbtn.setSelected(**false**);

tipsbtn.setText("");

setTips(**false**);

}

**else**{

tipsbtn.setSelected(**true**);

tipsbtn.setText("X");

setTips(**true**);

}

}*//GEN-LAST:event\_tipsbtnMouseReleased*

**private** void tipsbtnMouseClicked(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_tipsbtnMouseClicked*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_tipsbtnMouseClicked*

**private** void tipsbtnActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_tipsbtnActionPerformed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_tipsbtnActionPerformed*

**private** void addsourcebtnActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_addsourcebtnActionPerformed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_addsourcebtnActionPerformed*

**private** void addsourcebtnMouseReleased(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_addsourcebtnMouseReleased*

**try** {

*// TODO add your handling code here:*

*//sourcesEditor();*

openSources();

} **catch** (IOException ex) {

Logger.getLogger(LoginWindow.class.getName()).log(Level.SEVERE, **null**, ex);

}

}*//GEN-LAST:event\_addsourcebtnMouseReleased*

**private** void addsourcebtnMousePressed(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_addsourcebtnMousePressed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_addsourcebtnMousePressed*

**private** void addsourcebtnMouseClicked(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_addsourcebtnMouseClicked*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_addsourcebtnMouseClicked*

**private** void stopbtn1ActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_stopbtn1ActionPerformed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_stopbtn1ActionPerformed*

**private** void stopbtn1MouseClicked(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_stopbtn1MouseClicked*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_stopbtn1MouseClicked*

**private** void stopbtn1MouseReleased(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_stopbtn1MouseReleased*

*// TODO add your handling code here:*

quit();

}*//GEN-LAST:event\_stopbtn1MouseReleased*

**private** void stopbtn1MousePressed(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_stopbtn1MousePressed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_stopbtn1MousePressed*

**private** void helpbtnMousePressed(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_helpbtnMousePressed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_helpbtnMousePressed*

**private** void helpbtnMouseReleased(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_helpbtnMouseReleased*

*// TODO add your handling code here:*

help();

}*//GEN-LAST:event\_helpbtnMouseReleased*

**private** void helpbtnMouseClicked(java.awt.event.MouseEvent evt) {*//GEN-FIRST:event\_helpbtnMouseClicked*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_helpbtnMouseClicked*

**private** void helpbtnActionPerformed(java.awt.event.ActionEvent evt) {*//GEN-FIRST:event\_helpbtnActionPerformed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_helpbtnActionPerformed*

**private** void helpbtnKeyPressed(java.awt.event.KeyEvent evt) {*//GEN-FIRST:event\_helpbtnKeyPressed*

*// TODO add your handling code here:*

}*//GEN-LAST:event\_helpbtnKeyPressed*

*// </editor-fold>*

String iconFile = "/icons/icons8-news-50.png";

String sourceFile = "sources.txt";

String outputFile = "out/storygrab.html";

User currentusr;

BackgroundRunner loader = **new** BackgroundRunner();

SwingWorker backburner = loader.createWorker();

Scroller sourceScroller = **new** Scroller(loader);

ArrayList<Link> linkset = **new** ArrayList<>(); *//fill with links*

AutomaticScrollUpdater refresher =

**new** AutomaticScrollUpdater(sourceScroller);

SwingWorker refreshChild = refresher.createWorker();

*/\*\**

*\* Resets all text fields.*

*\*/*

**public** void resetAllFields(){

pwdfield.setText("");

userfield.setText("");

nouserwithname.setText("");

usertitlelbl.setText("");

dynamicdepthlbl.setText(""+loader.getMaxDepth());

statuslbl.setText("");

}

*/\*\**

*\* Changes the BackgroundRunner object depth.*

*\* @param t depth to change to*

*\*/*

**public** void changeDepth(int t){

loader.setMaxDepth(loader.getMaxDepth()+t);

}

*/\*\**

*\* Sets whether output is verbose or not.*

*\* @param t boolean to set to*

*\*/*

**public** void setVerbose(boolean t){

loader.setVerbose(t);

}

*/\*\**

*\* Opens text file containing the sources to search.*

*\* @throws IOException if desktop not supported*

*\*/*

**public** void openSources() **throws** IOException{

openFile(sourceFile);

}

**public** void openFile(String f) **throws** IOException {

**if**(!loader.isRunning()){

File file = **new** File(f);

*//first check if Desktop is supported by Platform or not*

**if**(!Desktop.isDesktopSupported()){

System.out.println("Desktop is not supported");

**return**;

}

Desktop desktop = Desktop.getDesktop();

**if**(file.exists()) desktop.open(file);

}

**else**{

*//send messsage*

}

}

*/\*\**

*\* Sets the toolTipText for everything.*

*\* @param t boolean whether it should be on or off*

*\*/*

**public** void setTips(boolean t){

**if**(t){

*//add all the tooltiptexts*

userfield.setToolTipText("Use AEP username");

pwdfield.setToolTipText("Use AEP password\nNone of this is stored"

+ " outside of the JVM");

loginbtn.setToolTipText("Open the operations window");

sourceslist.setToolTipText("All sources");

keywordfield.setToolTipText("Keyword to search for");

depthtogglepanel.setToolTipText("Set how deep the extractor will "

+ "recursively explore");

verbosepanel.setToolTipText("Verbose output");

extractbtn.setToolTipText("Start extracting");

stopbtn.setToolTipText("Stop extracting");

tipsbtnpanel.setToolTipText("Tips");

logoutbtn1.setToolTipText("Logout");

addsourcebtn.setToolTipText("Edit sources");

}

**else**{

userfield.setToolTipText(**null**);

pwdfield.setToolTipText(**null**);

loginbtn.setToolTipText(**null**);

sourceslist.setToolTipText(**null**);

keywordfield.setToolTipText(**null**);

depthtogglepanel.setToolTipText(**null**);

verbosepanel.setToolTipText(**null**);

extractbtn.setToolTipText(**null**);

stopbtn.setToolTipText(**null**);

tipsbtnpanel.setToolTipText(**null**);

logoutbtn1.setToolTipText(**null**);

addsourcebtn.setToolTipText(**null**);

}

}

*/\*\**

*\* Switches the screen from the main to login and login to main*

*\* @param sw int to switch to; 1 for login, 2 for main*

*\*/*

**public** void panelSwitcher(int sw){

resetAllFields();

sourceScroller.setLabel(sourceslist);

updateSources();

*//1 for login,2 for main*

**switch**(sw){

**case** 1:

LoginPanel.setVisible(**true**);

MainPanel.setVisible(**false**);

**break**;

**case** 2:

LoginPanel.setVisible(**false**);

MainPanel.setVisible(**true**);

usertitlelbl.setText(currentusr.screenName);

**break**;

}

}

*/\*\**

*\* Switches to login screen.*

*\*/*

**public** void login(){*//SECURE VERSION*

currentusr = **new** User(userfield.getText(),

**new** String(pwdfield.getPassword()));

panelSwitcher(2);

}

*/\*\**

*\* Logs out the user and switches to the login screen.*

*\*/*

**public** void logout(){

currentusr = **null**;

panelSwitcher(1);

}

*/\*\**

*\* Opens the help menu.*

*\*/*

**public** void help(){

*//HelpMenu hw = new HelpMenu();*

String[] sg = **new** String[0];

HelpMenu.main(sg);

}

*/\*\**

*\* Opens the output HTML file.*

*\* @throws IOException if desktop not supported*

*\*/*

**public** void openOutput() **throws** IOException{

openFile(outputFile);

}*/\*\**

*\* Starts the extraction process.*

*\* @throws InterruptedException if thread is interrupted*

*\*/*

**public** void extract() **throws** InterruptedException{

**if**(!loader.isRunning()){

loader.res();

loader.setVerbose(**true**);

backburner = loader.createWorker(); *//move these to extract.*

loader.passVec(linkset);

loader.passLbl(statuslbl);

loader.passInitializedOP(onelineout);

loader.passBigOut(largeoutput);

loader.passHitsLbl(htslbl);

updateSources();

loader.setBefore(keywordfield.getText(),currentusr);*//add depth here*

backburner.execute();

statuslbl.setText("Extracting...");

}

}

*/\*\**

*\* Stops the extraction process.*

*\* @throws InterruptedException if thread is interrupted*

*\*/*

**public** void stopExtract() **throws** InterruptedException{

loader.broStop();

}

*/\*\**

*\* Updates sources.*

*\*/*

**public** void updateSources(){

sourceScroller.updateSources();

}

*/\*\**

*\* Sets window icon.*

*\*/*

**private** void setIcon() {

setIconImage(Toolkit.getDefaultToolkit().getImage(getClass().getResource(iconFile)));

}

*/\*\**

*\* Quits the program.*

*\*/*

**public** void quit(){

dispose();

*//loader.cleanup();*

System.exit(0);

}

*/\*\**

*\* Main function.*

*\* @param args the command line arguments*

*\*/*

**public** **static** void main(String[] args) {

*/\* Set the Nimbus look and feel \*/*

*//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">*

*/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.*

*\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html*

*\*/*

**try** {

**for** (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

**if** ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

**break**;

}

}

} **catch** (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(LoginWindow.class.getName()).log(java.util.logging.Level.SEVERE, **null**, ex);

} **catch** (InstantiationException ex) {

java.util.logging.Logger.getLogger(LoginWindow.class.getName()).log(java.util.logging.Level.SEVERE, **null**, ex);

} **catch** (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(LoginWindow.class.getName()).log(java.util.logging.Level.SEVERE, **null**, ex);

} **catch** (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(LoginWindow.class.getName()).log(java.util.logging.Level.SEVERE, **null**, ex);

}

*//</editor-fold>*

*//</editor-fold>*

*/\* Create and display the form \*/*

java.awt.EventQueue.invokeLater(**new** Runnable() {

**public** void run() {

**new** LoginWindow().setVisible(**true**);

}

});

}

*// Variables declaration - do not modify//GEN-BEGIN:variables*

**private** keeptoo.KGradientPanel LoginPanel;

**private** keeptoo.KGradientPanel MainPanel;

**private** keeptoo.KButton addsourcebtn;

**private** keeptoo.KButton addsourcebtn2;

**private** javax.swing.JLabel cblbl2;

**private** javax.swing.JLabel cblbl3;

**private** javax.swing.JLabel closelbl;

**private** javax.swing.JLabel closelbl1;

**private** javax.swing.JLabel depthlbl;

**private** javax.swing.JPanel depthtogglepanel;

**private** javax.swing.JLabel dynamicdepthlbl;

**private** keeptoo.KButton extractbtn;

**private** javax.swing.JPanel extractorpanel;

**private** keeptoo.KButton helpbtn;

**private** keeptoo.KButton helpbtn1;

**private** javax.swing.JLabel htslbl;

**private** javax.swing.JTextField keywordfield;

**private** javax.swing.JLabel largeoutput;

**private** keeptoo.KButton loginbtn;

**private** keeptoo.KButton logoutbtn1;

**private** javax.swing.JPanel maincontrolpanel;

**private** javax.swing.JLabel mainlogowithicon;

**private** keeptoo.KButton minusdepthbtn;

**private** javax.swing.JLabel nouserwithname;

**private** javax.swing.JLabel onelineout;

**private** javax.swing.JLabel orlabel;

**private** keeptoo.KButton plusdepthbtn;

**private** javax.swing.JLabel profileiconlbl;

**private** javax.swing.JLabel profilelblicon;

**private** javax.swing.JPasswordField pwdfield;

**private** javax.swing.JLabel pwdlbl;

**private** javax.swing.JLabel sourceslbl;

**private** javax.swing.JLabel sourceslist;

**private** javax.swing.JLabel statuslbl;

**private** keeptoo.KButton stopbtn;

**private** keeptoo.KButton stopbtn1;

**private** keeptoo.KButton tipsbtn;

**private** javax.swing.JPanel tipsbtnpanel;

**private** javax.swing.JTextField userfield;

**private** javax.swing.JLabel userlbl1;

**private** javax.swing.JLabel usertitlelbl;

**private** keeptoo.KButton verbosebtn;

**private** javax.swing.JPanel verbosepanel;

**private** javax.swing.JLabel websitelbl8;

*// End of variables declaration//GEN-END:variables*

}

BackgroundRunner.java

**package** **main**;

**import** **java.io.BufferedReader**;

**import** **java.io.FileNotFoundException**;

**import** **java.io.FileReader**;

**import** **java.io.IOException**;

**import** **java.util.ArrayList**;

**import** **java.util.Collections**;

**import** **java.util.List**;

**import** **java.util.concurrent.ExecutionException**;

**import** **javax.swing.JLabel**;

**import** **javax.swing.SwingWorker**;

*/\*\**

*\* This class is what controls all the extractors running in the background.*

*\* @author Calvin Kinateder*

*\*/*

**public** **class** **BackgroundRunner** {

String searchFor;

User currentusr;

**final** int OUTSIZE = 12;

ArrayList<Extractor> extractors = **new** ArrayList<>();

String sourcesFile = "sources.txt";

String finalHTML = "out/storygrab.html";

ArrayList<Link> sources = **new** ArrayList<>();

int maxDepth = 2;

**private** boolean isRunning = **false**; *//changes frequently*

String forClassifier = "src/datasets/links.csv";

JLabel statusLblRef, outputlbl, longout, hitslbl;

boolean shouldStop = **false**;

boolean verbose = **false**;

ArrayList<Link> hitLinks;

String DB\_URL = "jdbc:derby://localhost:1527/FinalLinks";

**private** boolean dynamic = **true**;

**public** BackgroundRunner() {

searchFor="";

currentusr=**new** User();

updateSrc();

}

*/\*\**

*\* Resets all the variables in this class as an object so it can be reused*

*\* without reinitialization*

*\* Precondition: object has been initialized*

*\* Postcondition: object can be used again*

*\*/*

**public** void res(){

extractors = **new** ArrayList<>();

sourcesFile = "sources.txt";

String finalHTML = "out/storygrab.html";

ArrayList<Link> sources = **new** ArrayList<>();

int maxDepth = 2;

isRunning = **false**; *//changes frequently*

forClassifier = "src/datasets/links.csv";

shouldStop = **false**;

verbose = **false**;

DB\_URL = "jdbc:derby://localhost:1527/FinalLinks";

dynamic = **true**;

}

*/\*\**

*\* Updates the sources list from the file.*

*\* Precondition: sourcesFile exists*

*\*/*

**public** void updateSrc(){

**if**(!isRunning){

sources.clear();

**try** {

*// FileReader reads text files in the default encoding.*

FileReader fileReader =

**new** FileReader(sourcesFile);

BufferedReader bufferedReader =

**new** BufferedReader(fileReader);

String line;

**while** ((line = bufferedReader.readLine()) != **null**) {

sources.add(**new** Link(line));

}

*// Always close files.*

bufferedReader.close();

}

**catch**(FileNotFoundException ex) {

System.out.println(

"Unable to open file '" +

sourcesFile + "'");

backupSrcs();

}

**catch**(IOException ex) {

System.out.println(

"Error reading file '"

+ sourcesFile + "'");

backupSrcs();

}

}

}

*/\*\**

*\* In case sources.txt cannot be used, this holds the sources too.*

*\*/*

**private** void backupSrcs(){

String bckup = "http://bbc.com/\n" +

"http://foxnews.com/\n" +

"http://www.huffingtonpost.com/\n" +

"http://www.nbcnews.com/\n" +

"http://www.nytimes.com/\n" +

"http://www.wsj.com/\n" +

"http://www.usatoday.com/\n" +

"http://news.google.com/\n" +

"http://www.rollcall.com\n" +

"http://www.latimes.com\n" +

"http://www.wired.com\n" +

"http://www.cnn.com\n" +

"http://www.npr.org\n" +

"http://abcnews.go.com/\n" +

"https://www.usnews.com/news\n" +

"http://www.yahoo.com/news/\n" +

"http://www.ap.org/en-us/\n" +

"http://www.pewresearch.org/\n";

String[] asArr = bckup.split("\n");

**for**(String src : asArr){

sources.add(**new** Link(src));

}

}

*/\*\**

*\* Add Extractor to the extractor list.*

*\* @param e Extractor to add to the list*

*\*/*

**public** void add(Extractor e){

extractors.add(e);

}

*/\*\**

*\* Sets the recursion depth.*

*\* @param m depth value to set*

*\*/*

**public** void setMaxDepth(int m){

maxDepth = m;

**for**(int i = 0; i<extractors.size();i++){

extractors.get(i).setMaxDepth(maxDepth);

}

}

*/\*\**

*\* Sets searchFor for every Extractor in the container list.*

*\* @param s String to set*

*\*/*

**public** void setSearchFor(String s){

*//System.out.println(extractors.size());*

**for**(int i = 0; i<extractors.size();i++){

extractors.get(i).setSearchFor(s);

*//System.out.println(extractors.get(i).searchFor);*

}

}

*/\*\**

*\* Creates the container list holding all the Extractors.*

*\* @param u User to setCreds with*

*\*/*

**public** void createContainer(User u){

updateSrc();

extractors.clear();

hitLinks.clear();

**for**(Link src : sources){

extractors.add(**new** Extractor(src.getHyperlink()));*//make new extractor for each source*

}

setMaxDepth(maxDepth);

setCreds(u);

*//System.out.println(extractors.toString());*

}

*/\*\**

*\* Gets whether the object is running.*

*\* @return isRunning*

*\*/*

**public** boolean isRunning(){

**return** isRunning;

}

*/\*\**

*\* Sets all the credentials for each link.*

*\* @param u User to pass the creds on*

*\*/*

**public** void setCreds(User u){

**for**(int i = 0; i<extractors.size();i++){

extractors.get(i).setCreds(u.getUser(), u.getPassword());

}

}

*/\*\**

*\* Prints all the Extractors in the list.*

*\*/*

**public** void printExtractors(){

**for**(Extractor e : extractors){

System.out.println(e);

}

System.out.println();

}

*/\*\**

*\* Sets shouldStop to true ands updates the sources.*

*\*/*

**public** void broStop(){

shouldStop = **true**;

updateSrc();

}

*/\*\**

*\* Finds Link with attached to source l.*

*\* @param l String referencing the source to be found*

*\* @return Link with l as the source.hyperlink*

*\*/*

**public** Link findSource(String l){

**for**(int i = 0; i<sources.size(); i++){

**if**(sources.get(i).getHyperlink().equals(l)){

**return** sources.get(i);

}

}

**return** **null**;

}

*/\*\**

*\* Counts the total number of errors for all Extractor objects.*

*\* @return the sum of all errors*

*\*/*

**public** int totalErrors(){

int l = 0;

**for**(int i = 0; i<sources.size(); i++){

l+=sources.get(i).errors();

}

**return** l;

}

*/\*\**

*\* Combines all the individual extractor links into one big ArrayList*

*\*/*

**public** void combine(){ *//only call when joined*

**for**(Extractor e : extractors){

hitLinks.addAll(e.hits);

}

}

*/\*\**

*\* Counts the total number of hits for all the extractors*

*\* @return int count of all hits*

*\*/*

**public** int countAll(){

int ct = 0;

**for**(Extractor e : extractors){

ct+=e.hits.size();

}

**return** ct;

}

*/\*\**

*\* Cleans up the remaining variables and saves everything. Sorts hitLinks*

*\* too.*

*\* @return String containing any error/finishing messages*

*\*/*

**public** String cleanup(){

combine();

shouldStop = **true**;

String out = "";

Collections.sort(hitLinks);

out+=Tools.saveToCSV(hitLinks, forClassifier)+"<br>";

out+=Tools.saveToHTML(hitLinks, finalHTML, searchFor)+"<br>";

out+=Tools.saveToDB(hitLinks, DB\_URL);

**return** out;

}

**public** SwingWorker createWorker() {

**return** **new** SwingWorker<Boolean, String>() {

*/\*\**

*\* Collects all the Extractors and keeps track of them all to run*

*\* smoothly.*

*\* @return true if finished successfully, false if failed*

*\* @throws Exception for Thread.join()*

*\*/*

@Override

**protected** Boolean doInBackground() **throws** Exception {

System.out.println("Search for: "+searchFor+" User: "+currentusr);

statusLblRef.setText("Starting...");

createContainer(currentusr);

setSearchFor(searchFor);

printExtractors();*//print all extractors*

**for**(Extractor e : extractors){

e.setModes(**new** boolean[] {**true**, **true**});

e.start();*//make new thread for each extractor*

}

boolean alldone=**false**;

int leftToDo = extractors.size();

int percent = 0;

ArrayList<Extractor> used = **new** ArrayList<>();

statusLblRef.setText("Extracting... "+percent+"%");

**for**(int i = 0; i<OUTSIZE;i++){

publish("Starting..."); *//fill chunks to OUTSIZE*

}

**while**(!alldone){

isRunning = **true**;

**for**(int i = 0;i<extractors.size();i++){

Extractor t=extractors.get(i);

**if**(!t.isAlive()&&!used.contains(t)){

used.add(t);

leftToDo--;

*//set tmp vars*

double s = extractors.size();

double l = leftToDo;

double sm = s-l;

findSource(t.getWebpage()).setlinkDone(**true**);

System.out.println();

System.out.println("Done: ");

**for**(Extractor e : used){

System.out.println("\t"+e);

}

System.out.println();

percent=(int) ((sm/s)\*100);

statusLblRef.setText("Extracting... "+percent+"%"); *//set label*

}

publish(t.toBG);*//publish output to process*

**if**(!t.errorMsgs.equals("")){

publish("<font color=FFD126>"+t.errorMsgs+"</font>");

findSource(t.getWebpage()).setError(t.errorCount);

}

**if**(shouldStop){

publish("Cancelled by user - syncing threads");

**for**(Extractor ts : extractors){

publish("Extractor termination on "+ts.getWebpage());

ts.setStop(**true**);

}

**for**(Thread tt : extractors){

tt.join();

}

publish("Saving to database");

shouldStop = **false**; *//so loader can be used again*

isRunning = **false**; *//update this*

**return** **false**;

}

}

**if**(leftToDo==0){

alldone=**true**;

}

}

publish("Finished successfully - saving to database");

**for**(Thread t : extractors){

t.join();

}

publish("Completed");

isRunning = **false**;

*// Finished*

**return** **true**;

}

*/\*\**

*\* Part of formatting the output.*

*\* @param chunks List of strings to publish to the output labels*

*\*/*

**protected** void process(List<String> chunks) {

*// Get Info*

String currentOut = "";

currentOut = chunks.get(chunks.size()-1);

outputlbl.setText("<html>"+currentOut+"</html>");

int st = chunks.size()-OUTSIZE;

**if**(st<0){

st=0;

}

**if**(verbose){

String bo; *//no scroll*

bo = "<html><br>";

**for**(int i = st; i<chunks.size();i++){

String se = chunks.get(i).replaceAll("Searching ", "");

bo=bo+se+"<br>";

}

bo+="</html>";

longout.setText(bo);

}

**else**{

longout.setText("");

}

hitslbl.setText(*/\*"Hits: "+\*/*countAll()+"");

**if**(dynamic){Tools.saveToHTML(hitLinks, finalHTML, searchFor);}

}

*/\*\**

*\* What to do when the processing is finished.*

*\*/*

@Override

**protected** void done() {

boolean bStatus = **false**;

**try** {

bStatus = get();

statusLblRef.setText("");

publish(cleanup());

**if**(bStatus){

System.out.println("Done on all!");

}

**else**{

publish("<br><br>"

+ "Finished<br>");

}

} **catch** (InterruptedException | ExecutionException ex) {

ex.printStackTrace();

}

}

};

} *// End of Method: createWorker()*

*/\*\**

*\* Accessors and modifiers.*

*\*/*

**public** void passVec(ArrayList h){

hitLinks = h;

}

void setDynamic(boolean t) {

dynamic = t;

}

**public** void passLbl(JLabel l){

statusLblRef = l;

statusLblRef.setText("");

}

**public** void passInitializedOP(JLabel l){

outputlbl = l;

outputlbl.setText("");

}

void passBigOut(JLabel l) {

longout = l;

longout.setText("");

}

void passHitsLbl(JLabel l) {

hitslbl = l;

}

**public** int getMaxDepth(){

**return** maxDepth;

}

**public** void setBefore(String s, User u){*//set the variables relevant before running.*

searchFor = s; currentusr=u;

}

**public** void setVerbose(boolean j){

verbose = j;

}

}

Extractor.java

**package** **main**;

**import** **org.jsoup.Jsoup**;

**import** **org.jsoup.nodes.Document**;

**import** **org.jsoup.nodes.Element**;

**import** **org.jsoup.select.Elements**;

**import** **java.io.IOException**;

**import** **java.util.ArrayList**;

**import** **java.util.Base64**;

**import** **java.util.HashSet**;

**import** **java.util.Vector**;

*/\*\**

*\* Searches the webpage for a given keyword.*

*\* @author Calvin Kinateder*

*\*/*

**public** **class** **Extractor** **extends** Thread {

**public** int maxDepth = 5; *//initial depth*

**private** HashSet<String> alreadySearched; *//no double searching*

**private** HashSet<String> bodiesSearched = **new** HashSet<>();

**private** HashSet<String> titlesSearched = **new** HashSet<>();

**public** String searchFor = "";

**public** String webpage = "";

**public** boolean done = **false**;

**private** boolean stop = **false**;

**private** String username="";

**private** String password="";

**public** String toBG = "";*//to send to the backgroundrunner*

**public** String errorMsgs = "";*//to send to bg too*

**public** ArrayList<Link> hits = **new** ArrayList<>();

**public** int errorCount = 0;

**public** boolean[] modes = {**false**, **false**};

**public** Extractor() {

alreadySearched = **new** HashSet<>();

}

**public** Extractor(String w){

alreadySearched = **new** HashSet<>();

webpage=w;

}

*/\*\**

*\* Returns the number of times a specified word appears in a*

*\* specified String. Uses .equals() instead of .contains()*

*\* @param str the block of text needing to be searched*

*\* @param word the word being searched for*

*\* Precondition: str words are separated by " ",".", or ","*

*\* @return the number of times the specified word appears in the*

*\* specified str*

*\*/*

**public** int getFreq(String str, String word){

String[] a = str.split(" |\\.|\\,");

int count = 0;

**for** (int i = 0; i < a.length; i++){

**if** (word.toLowerCase().equals(a[i].toLowerCase()))

count++; *// if match found increase count*

}

**return** count;

}

*/\*\**

*\* Returns the number of times a specified word appears in a*

*\* specified String. Uses .contains() instead of .equals()*

*\* @param str the block of text needing to be searched*

*\* @param word the word being searched for*

*\* Precondition: str words are separated by " ",".", or ","*

*\* @return the number of times the specified word appears in the*

*\* specified str*

*\*/*

**public** int getFreq2(String str, String word){

String[] a = str.split(" |\\.|\\,");

int count = 0;

**for** (String i : a){

**if** (i.toLowerCase().contains(word.toLowerCase()))

count++; *// if match found increase count*

}

**return** count;

}

*/\*\**

*\* Loads the webpage connected to the specified URL and, using getFreq(),*

*\* recursively finds every link and webpage containing the universal search*

*\* term.*

*\* Precondition: the provided login credentials are correct and URL connects*

*\* successfully*

*\* @param URL the URL of the webpage*

*\* @param depth how many times the method will recursively search for the*

*\* search term*

*\* @param strUserId username to get past the firewall if it exists*

*\* @param strPassword password to get past the firewall if it exists*

*\* @return true if completed successfully, false if not*

*\*/*

**public** boolean searchPageLinks(String URL, int depth, String strUserId,

String strPassword) {*//returns true when done*

errorMsgs = "";

String authString = strUserId + ":" + strPassword;

String encodedString = *//need to authenticate for firewall*

Base64.getEncoder().encodeToString(authString.getBytes());

**if** ((!alreadySearched.contains(URL) && (depth < maxDepth))&&!stop) {

toBG=""+URL+"\n";*//add to toBG*

**try** {

alreadySearched.add(URL); *//add link to the hashset*

Document document = Jsoup.connect(URL) *//connect to doc*

.header("Authorization", "Basic " + encodedString)

.get();

Elements alreadySearchedOnPage = document.select("a[href]");

Elements txt = document.select("p");

String article = txt.text();

String title = document.title();

**if**(getFreq(article, searchFor)>1

&&!bodiesSearched.contains(article)

&&!titlesSearched.contains(title)){

int f = getFreq2(article, searchFor);

hits.add(**new** Link(URL, article,searchFor,title,f));

}

bodiesSearched.add(article);

titlesSearched.add(title);

depth++;

**for** (Element page : alreadySearchedOnPage) {

searchPageLinks(page.attr("abs:href"),

depth,strUserId,strPassword);

}

} **catch** (IOException | IllegalArgumentException e) {

System.err.println("For '" + URL + "': " + e.getMessage());

errorMsgs = "E:"+URL;

errorCount++;

}

}

**return** **true**; *//reached finish*

}

*/\*\**

*\* Runs the whole class.*

*\*/*

@Override

**public** void run() {

Thread.currentThread().setName(webpage);

System.out.println("Extractor running on "+webpage+" at max depth "

+maxDepth);

done = searchPageLinks(webpage, 0, username, password);

**if**(stop){

System.out.println("Extractor prematurely finished on "+webpage);

}

**else**{

System.out.println("Extractor client successful on "+webpage);

}

}

*/\*\**

*\* Returns the webpage and depth as a String.*

*\* @return String including the page and depth*

*\*/*

@Override

**public** String toString(){

**return** "Page: "+webpage+", Depth: "+maxDepth;

}

*/\**

*Accessors and modifiers*

*\*/*

**public** void setModes(boolean[] m){

modes = m;

}

**public** void setStop(boolean t) {

stop = t;

}

**public** void setWebpage(String l){

webpage = l;

}

**public** String getWebpage(){

**return** webpage;

}

**public** void setMaxDepth(int i){

maxDepth=i;

}

**public** int getMaxDepth(){

**return** maxDepth;

}

**public** void setSearchFor(String s){

searchFor = s;

}

**public** String getSearchFor(){

**return** searchFor;

}

**public** void setCreds(String u, String p){

username = u;

password = p;

}

}

Scroller.java

**package** **main**;

**import** **javax.swing.JLabel**;

*/\*\**

*\* Controls the scrolling methods*

*\* @author Calvin Kinateder*

*\*/*

**public** **class** **Scroller** {

JLabel modifiedLbl;

int scrollStart = 0;

int SCROLLSIZE = 10; *//add note at bottom that there is more*

BackgroundRunner lblContainer;

*/\*\**

*\* Constructor for Scroller*

*\* @param lblContainer the BackgroundRunner containing the sources list*

*\*/*

**public** Scroller(BackgroundRunner lblContainer){

**this**.lblContainer = lblContainer;

**this**.lblContainer.updateSrc();

**if**(lblContainer.sources.size()<=SCROLLSIZE){

SCROLLSIZE = lblContainer.sources.size();

}

}

*/\*\**

*\* Updates scrollStart and SCROLLSIZE for the finalOut pane on the GUI.*

*\* @param mwheel the mousewheel event*

*\*/*

**public** void scrollSources(java.awt.event.MouseWheelEvent mwheel){

int scAmount = mwheel.getWheelRotation();

**if**(scrollStart+scAmount>=0 && SCROLLSIZE+scAmount<lblContainer.sources.size()+1){

scrollStart+=scAmount;

SCROLLSIZE+=scAmount;

}

updateSources();

}

*/\*\**

*\* Updates the HTML String for the finalOut pane on the GUI.*

*\*/*

**public** void updateSources(){

lblContainer.updateSrc();

*//sort sources list with not done at the top*

String finalOut = "<html>";

**if**(scrollStart>0){

finalOut+="...<br>";

}

**else**{

finalOut+="<br>";

}

**for**(int i = scrollStart; i<SCROLLSIZE;i++){

**if**(i<lblContainer.sources.size()){

String se = "";

Link tmpLink = lblContainer.sources.get(i);

**if**(tmpLink.islinkDone()){

se = (i+1)+": <b><font color=31E13C>"+tmpLink.getHyperlink()+"</font></b>";

}

**else** **if**(!tmpLink.islinkDone()&&tmpLink.errors()>10){

se = (i+1)+": <b><font color=FF7777>"+tmpLink.getHyperlink()+"</font></b>";

}

**else** **if**(!tmpLink.islinkDone()&&tmpLink.errors()>0){

se = (i+1)+": <b><font color=FFD126>"+tmpLink.getHyperlink()+"</font></b>";

}

**else**{

se = (i+1)+": "+tmpLink.getHyperlink();

}

finalOut=finalOut+se+"<br>";

}

}

**if**(lblContainer.sources.size()>SCROLLSIZE){

finalOut+="...<br>";

}

finalOut+="</html>";

modifiedLbl.setText(finalOut);

}

*/\**

*\* Modifiers*

*\*/*

**public** void setLabel(JLabel l){

modifiedLbl = l;

}

**public** void setWidth(int s){

SCROLLSIZE = s;

}

}

AutomaticScrollUpdater.java

**package** **main**;

**import** **javax.swing.\***;

*/\*\**

*\* Refreshes the state of Link objects once every WAIT ms*

*\* @author Calvin Kinateder*

*\*/*

**public** **class** **AutomaticScrollUpdater** {

Scroller scroller;

**final** int WAIT = 200;

*/\*\**

*\* Constructor for this class.*

*\* @param s Scroller to be manipulated*

*\*/*

**public** AutomaticScrollUpdater(Scroller s){

scroller = s;

}

*/\*\**

*\* Creates the worker for this class*

*\* @return SwingWorker*

*\*/*

**public** SwingWorker createWorker() {

**return** **new** SwingWorker<Boolean, Integer>() {

@Override

**protected** Boolean doInBackground() **throws** Exception {

*// Start Progress*

waitFor(2000);

setProgress(0);

**while**(**true**){

scroller.updateSources();

waitFor(WAIT);

}

}

@Override

**protected** void done() {

boolean bStatus = **false**;

**try** {

bStatus = get();

} **catch** (Exception ex) {

ex.printStackTrace();

}

System.out.println("Finished with status " + bStatus);

}

};

} *// End of Method: createWorker()*

*/\*\**

*\* Wait the given time in milliseconds*

*\* @param iMillis number of milliseconds to wait for*

*\*/*

**private** void waitFor (int iMillis) {

**try** {

Thread.sleep(iMillis);

}

**catch** (Exception ex) {

System.err.println(ex);

}

}

}

Tools.java

**package** **main**;

**import** **java.io.BufferedWriter**;

**import** **java.io.FileWriter**;

**import** **java.io.IOException**;

**import** **java.sql.Connection**;

**import** **java.sql.DriverManager**;

**import** **java.sql.PreparedStatement**;

**import** **java.sql.SQLException**;

**import** **java.util.ArrayList**;

**import** **java.util.Collections**;

**import** **java.util.List**;

**import** **java.util.Vector**;

**import** **java.util.logging.Level**;

**import** **java.util.logging.Logger**;

*/\*\**

*\* Contains functions needed for saving data.*

*\* @author Calvin Kinateder*

*\*/*

**public** **abstract** **class** **Tools** {

**public** **static** String delim ="---split---";

*/\*\**

*\* Writes String toWrite to a specified file.*

*\* @param fileName the name of the file to write to*

*\* @param toWrite the String to be written to the file*

*\* @param append boolean determining whether the file should be overwritten*

*\* or appended to*

*\*/*

**public** **static** void writeToFile(String fileName, String toWrite,

boolean append){

**try** {

FileWriter fileWriter =

**new** FileWriter(fileName,append);*//add true to append*

BufferedWriter bufferedWriter =

**new** BufferedWriter(fileWriter);

bufferedWriter.write(toWrite);

bufferedWriter.close();

}

**catch**(IOException ex) {

System.out.println(

"Error writing to file '"

+ fileName + "'");

}

}

*/\*\**

*\* Saves the contents of hitLinks to a database*

*\* @param hitLinks Vector containing all links that need to be written to*

*\* the database*

*\* @param DB\_URL String that holds the URL to the database*

*\* @return String noting any completion or error messages*

*\*/*

**public** **static** String saveToDB(List<Link> hitLinks, String DB\_URL) {

*//sort hitLinks*

Collections.sort(hitLinks);

**try** {

Connection conn = DriverManager.getConnection(DB\_URL, "calvin", "2255");

PreparedStatement dlt = conn.prepareStatement("DELETE FROM LINKS WHERE 1=1");

dlt.executeUpdate();

PreparedStatement ps = conn.prepareStatement("insert into LINKS(RELEVANCE,URL)"

+ "values(?,?)");

**for**(Link l : hitLinks){

ps.setDouble(1, l.getRelevance());

ps.setString(2, l.getHyperlink());

ps.executeUpdate();

}

**return** "Successfully wrote to database";

} **catch** (SQLException ex) {

*//Logger.getLogger(BackgroundRunner.class.getName()).log(Level.SEVERE, null, ex);*

**return** "Could not write to database";

}

}

*/\*\**

*\* Saves the contents of hitLinks to a CSV file.*

*\* @param hitLinks Vector containing all links that need to be written to*

*\* the database*

*\* @param forClassifier String to put at the beginning of the file for the*

*\* Python classifying algorithm- this is not in use*

*\* @return String noting any completion or error messages*

*\*/*

**public** **static** String saveToCSV(List<Link> hitLinks, String forClassifier){

FileWriter fileWriter =

**null**;

**try** {

fileWriter = **new** FileWriter(forClassifier,**false**); *//add true to append*

BufferedWriter bufferedWriter =

**new** BufferedWriter(fileWriter);

**for**(Link l : hitLinks){

bufferedWriter.write(Tools.delim+l.getRelevance()+Tools.delim); *//delim -s ','*

bufferedWriter.write(l.getHyperlink()+Tools.delim);

bufferedWriter.write(l.getText()+"\n");

}

bufferedWriter.close();

fileWriter.close();

**return** "Successfully wrote to CSV";

}

**catch** (IOException ex) {

Logger.getLogger(BackgroundRunner.class.getName()).log(Level.SEVERE, **null**, ex);

**return** "Could not write to CSV";

}

}

*/\*\**

*\* Saves the contents of hitLinks to an HTML file to be viewed in a browser.*

*\* @param hitLinks hitLinks Vector containing all links that need to be written to*

*\* the database*

*\* @param finalHTML the final HTML file containing all the links sorted by*

*\* relevance for easy viewing*

*\* @param searchFor the term that the links were searched for*

*\* @return String noting any completion or error messages*

*\*/*

**public** **static** String saveToHTML(List<Link> hitLinks, String finalHTML, String searchFor){

*//Collections.sort(hitLinks);*

ArrayList<Link> copy = **new** ArrayList(hitLinks);

Collections.sort(copy);

StringBuilder sb = **new** StringBuilder();

sb.append("<!DOCTYPE html>\n" +

"<html>\n" +

" <head>\n" +

" <title>StoryGrab</title>\n" +

" <link rel=\"stylesheet\" href=\"style.css\" type=\"text/css\">\n" +

" </head>\n" +

" <body>\n" +

" <div class=\"gee\">\n" +

" <span class=\"title1\">StoryGrab</span>\n" +

"<span class=\"hits\">Hits: "+copy.size()+" - Search Term: \""+

searchFor+"\"</span>"+

" </div>\n" +

" <div class=\"heading\"> \n" +

" \n" +

" </div>\n" +

" <div class=\"out\">\n" +

" <span class=\"txt\">");

Collections.reverse(copy);

**for**(int i = 0; i<copy.size();i++){

sb.append("Frequency: "+copy.get(i).getTermFreq()+

" - <a href=\""+copy.get(i).getHyperlink()+"\">"+

copy.get(i).getTitle()+"</a><br>\n");

}

sb.append(" </span>\n" +

" </div>\n" +

" </body>\n" +

"</html>");

Collections.sort(copy);

*//sb.append("</body>\n</html>");*

FileWriter fileWriter =

**null**;

**try** {

fileWriter = **new** FileWriter(finalHTML,**false**); *//add true to append*

BufferedWriter bufferedWriter =

**new** BufferedWriter(fileWriter);

bufferedWriter.write(sb.toString());

bufferedWriter.close();

fileWriter.close();

**return** "Successfully wrote to HTML";

}

**catch** (IOException ex){

**return** "Could not write to HTML";

}

}

}

Link.java

**package** **main**;

*/\*\**

*\* The link object stores the link, title, relevance constant*

*\* @author Calvin Kinateder*

*\*/*

**public** **class** **Link** **implements** Comparable<Link>{

**private** String title;

**private** String hyperlink;

**private** String body;

**private** String searchFor;

**private** double relevance;

**private** boolean linkDone;

**private** boolean failed = **false**;

**private** int errors = 0;

**private** int termFreq;

**public** Link(){

title = ""; *// avoid NullPointerException*

hyperlink = ""; *// avoid NullPointerException*

relevance = -1; *// not calculated*

linkDone = **false**;

}

**public** Link(String hyperlink){

**this**.title = "no title ";

**this**.hyperlink = hyperlink;

relevance = -1; *//not calculated*

linkDone = **false**;

}

**public** Link(String hyperlink, String body){

**this**.body = body;

**this**.hyperlink = hyperlink;

relevance = -1; *//not calculated*

linkDone = **false**;

}

**public** Link(String hyperlink, String body, String sf, String title, int f){

**this**.body = body;

**this**.hyperlink = hyperlink;

relevance = -1; *//not calculated*

linkDone = **false**;

searchFor = sf;

**this**.title = title;

termFreq = f; relevance = f;

}

**public** Link(String title, String hyperlink, double relevance){

**this**.title = title;

**this**.hyperlink = hyperlink;

**this**.relevance = relevance;

}

*// <editor-fold defaultstate="collapsed" desc="Accessors/Modifiers">*

*/\*\**

*\* Accessors and Modifiers*

*\*/*

**public** String getTitle(){

**return** title;

}

**public** String getHyperlink(){

**return** hyperlink;

}

**public** String getText(){

**return** body;

}

**public** String getSearchFor(){

**return** searchFor;

}

**public** double getRelevance(){

**return** relevance;

}

**public** boolean islinkDone(){

**return** linkDone;

}

**public** boolean isFailed(){

**return** failed;

}

**public** int errors() {

**return** errors;

}

**public** int getTermFreq() {

**return** termFreq;

}

**public** void setTitle(String t){

**this**.title = t;

}

**public** void setText(String t){

**this**.body = t;

}

**public** void setHyperlink(String t){

**this**.hyperlink = t;

}

**public** void setSearchFor(String t){

**this**.searchFor = t;

}

**public** void setRelevance(double t){

**this**.relevance = t;

}

**public** void setlinkDone(boolean t){

**this**.linkDone = t;

}

**public** void setFailed(boolean t){

**this**.failed = t;

}

**public** void setError(int t){

**this**.errors=t;

}

**public** void setTermFreq(int t){

**this**.termFreq=t;

}

**public** void incrementTermFreq(){

**this**.termFreq++;

}

*// </editor-fold>*

*/\*\**

*\* Returns String of information to distinguish each link object.*

*\* @return String containing hyperlink, relevance, # of errors, searchFor,*

*\* and termFreq*

*\*/*

@Override

**public** String toString(){

**return** "\nLink: "+hyperlink+""

+ "\nRelevance: "+relevance+"\nErrors: "+errors+

"\nSearchFor: '"+searchFor+"'\nFrequency in body: "+termFreq;

}

@Override

**public** int compareTo(Link o) {

**return** Double.compare(**this**.relevance, o.getRelevance());

}

}

1. Criterion A Word Count: 428 words [↑](#footnote-ref-1)
2. <https://docs.oracle.com/javase/7/docs/api/java/util/Vector.html> [↑](#footnote-ref-2)
3. Criterion B Word Count: 830 words [↑](#footnote-ref-3)
4. <https://jsoup.org/> [↑](#footnote-ref-4)
5. <https://www.javacodegeeks.com/2010/08/java-best-practices-vector-arraylist.html> [↑](#footnote-ref-5)
6. Criterion C Word Count: 651 words [↑](#footnote-ref-6)
7. Criterion E Word Count: 291 words [↑](#footnote-ref-7)