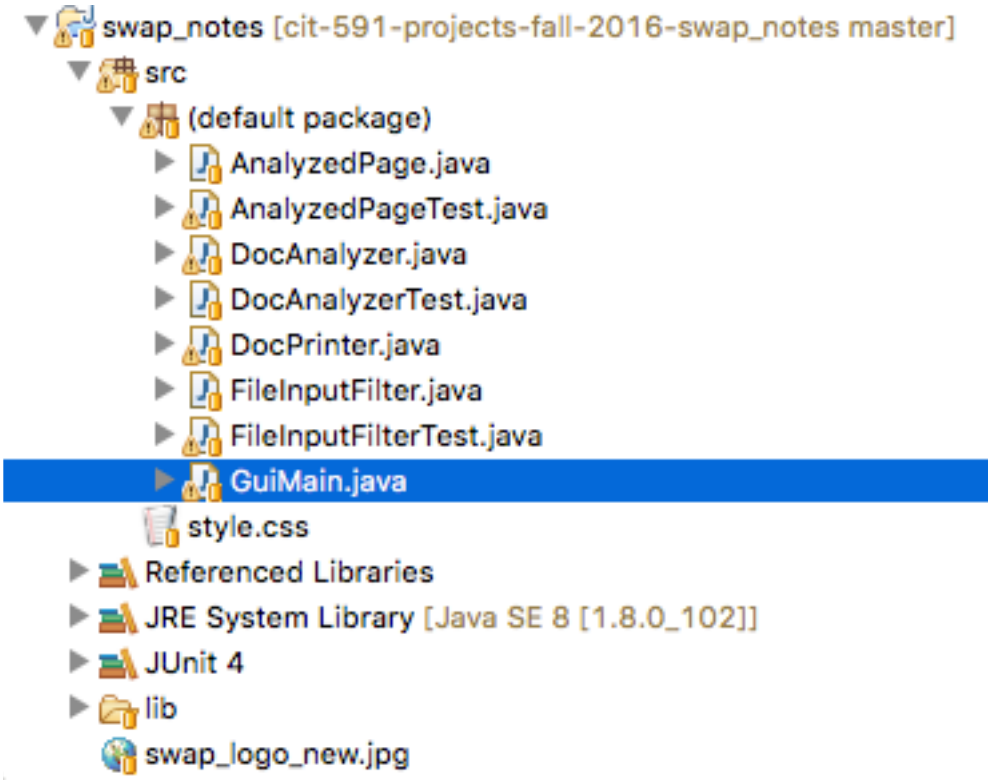
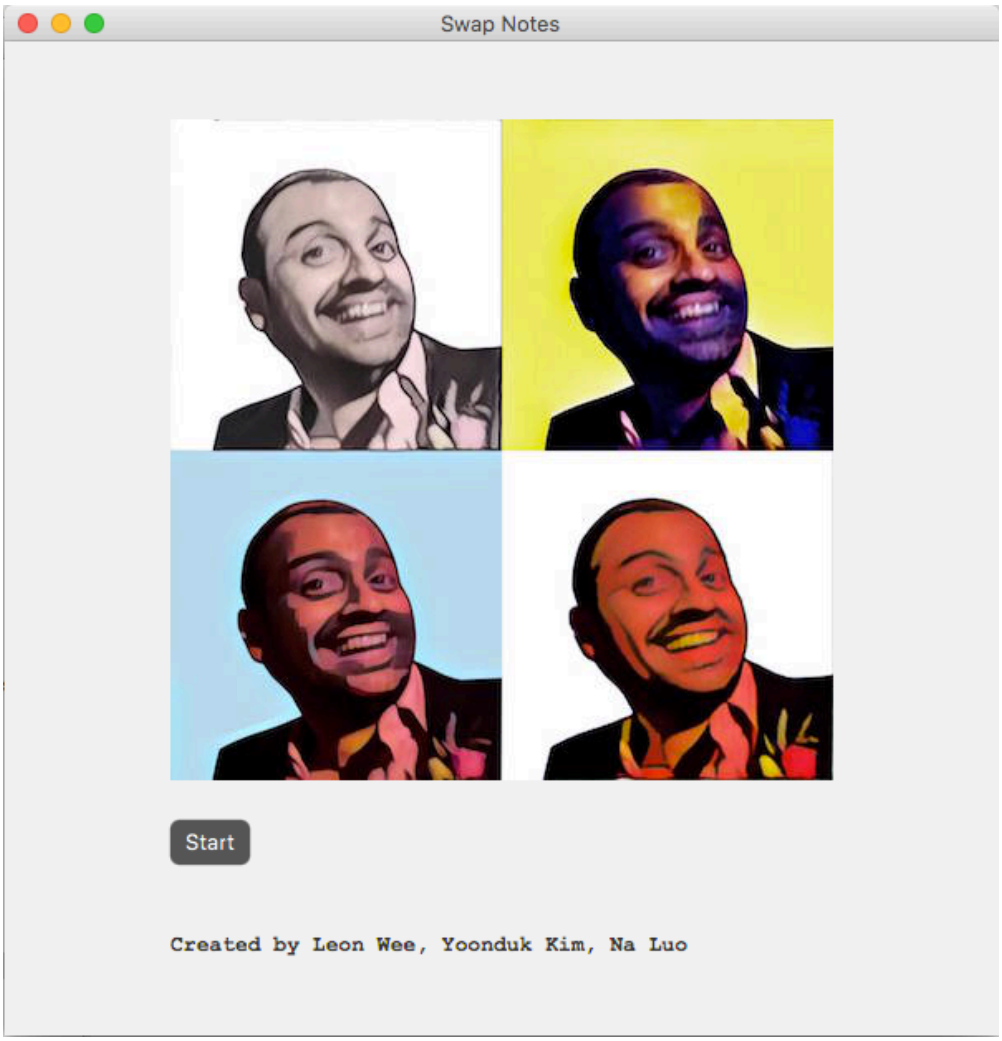
Instruction Manual for Swap Notes



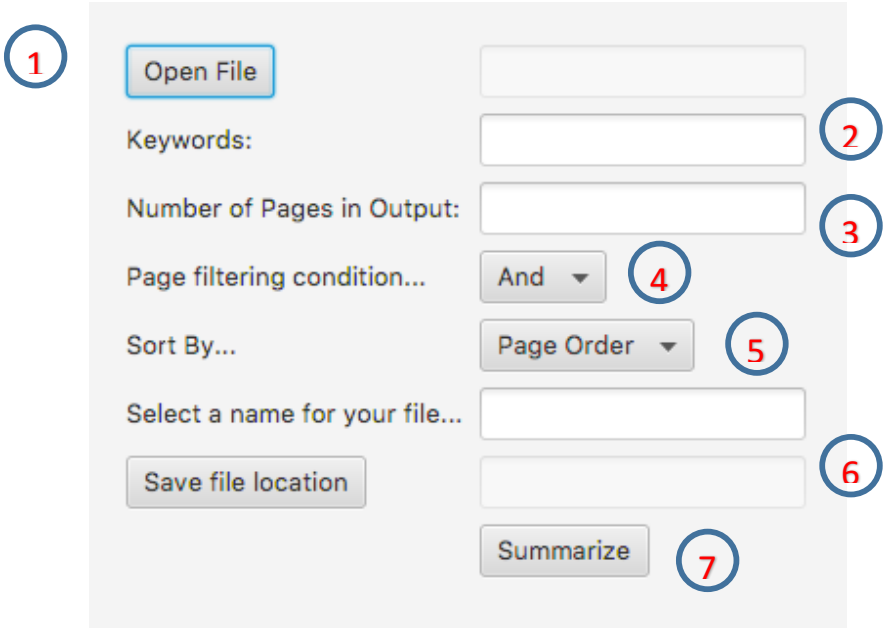
Step 1: Run GuiMain.java as a Java Application

This class contains the main method and the GUI.



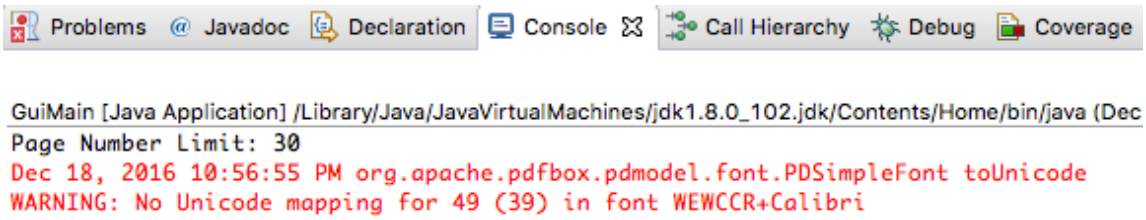
Step 2: Press the Start button on the beautiful landing page

So beautiful.



Step 3: Enter inputs and begin summarization process

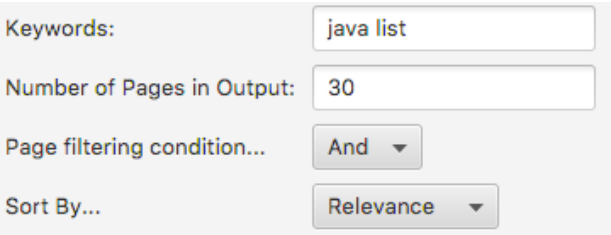
1. Open File: Opens a new File Selector window to select the pdf file.
   *PDFBox has difficulty parsing PDF documents created through OCR. If possible, use a PDF document that are well-formatted.
2. Keywords: Space separated list of keywords (Will implement comma separation for increased diversity of inputs in the future). Case insensitive.
3. Number of Pages in Output: Limits the number of pages the user wants in output file.
4. Page Filtering Condition: AND or OR option available. Determines whether the program searches pages containing all the keywords or one of the keywords.
5. Sort By: Page Order outputs documents in their original page order. Relevance sorts the results as determined by their relevance scores from the vector space algorithm.
6. Save File: Automatically appends ".pdf" at the end of the name field. Save file location button opens a Folder Selector window.
7. Summarize: Runs application.



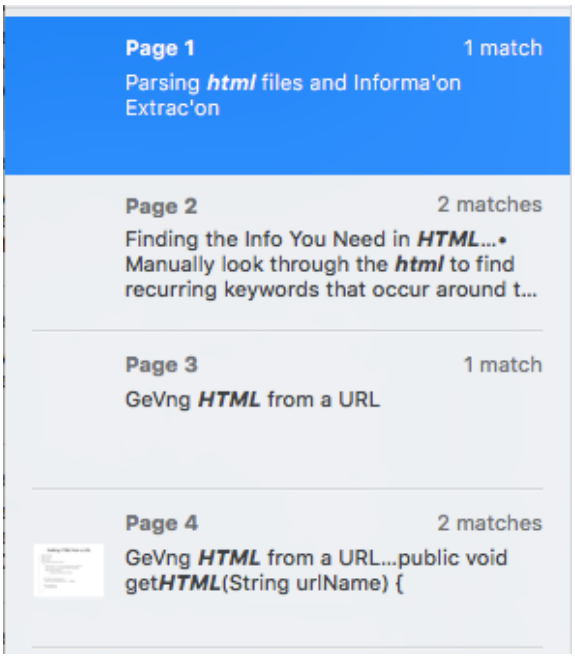
During the parsing process, this warning may pop up in the console. This just means that some of the fonts used in the PDF file are not contained in the API. As we are not using any functionality related to fonts, this message does not affect the accuracy of the application.

Result check

Above is an example of a resulting document created by running the following input.



We can see that the document correctly sorts the pages in the order of their relevance by the frequency of the two keywords “java” and “list”.



However, in some cases we see that pages are not sorted by the absolute frequency of keyword matches. This is because our algorithm ranks the pages based on the keyword's importance relative to the length of the page. For the example above, we can see that in page1, "html" has a greater weight than page2, although the latter has more matches.