* User sign-ins
  + You should be able to download whatever you do.
  + You potentially open a pandora’s box if you’re storing user authentication data
    - That seems risky and concerning
    - Cissi’s recommendations: Don’t do user authentication and tell the user to download their data or get it emailed to themselves.
* Regarding connecting a user’s social account during the session:
  + Make sure it’s explicit that the user is enabling their account info to be used
  + Make sure the account info isn’t stored.
  + Otherwise, that sounds like a good idea for getting a corpus.
  + Make sure the program is explicit about the legal/usage requirements with the data
* Document uploading
  + User would often want to upload a directory filled with files rather than just a single file
  + Cissi thought the concept of a “project folder” from the last project where the user could see what components were in their project
  + Scope is “educational” so we can/should put an upper limit on the number of files being analyzed, in order to avoid perf problems
  + One goal here: Allowing a user to categorize the corpus data they are working with
    - Maybe tagging files to help categorize them based on user-selected parameters
      * Would be helpful to tag multiple files at one time
    - **Maybe allow a config file to define the groupings/organization of files**
    - There’s no persistence, so you need a good way to preserve the organization for the user
* Pre-curated corpora
  + Project Gutenberg
    - Recommended to get some interesting ones in a smaller set
    - We could maybe look into leveraging the Gutenberg search to allow a user to look for a specific book to use
  + Wall Street Journal - in NLTK
  + Maybe look for another news source
  + Corpus of Contemporary American Language - BYU/Mark Davies
* Document downloading and project structure
  + Downloading a config file that helps maintain the project details would be very useful as part of the project files which get downloaded.
  + Can you allow data to be compared?
    - Maybe you allow data to be shown with 2 or 3 comparisons, shown in one “case” of running the program
    - Maybe a user will want to download their analyzed data so they can compare it to other files in a similar way
  + File exporting should be very basic: CSV, XML. Split it based on the type of data being delivered.
* Can every analysis be a visualization?
  + Sure, that works and keeps things simple
* Requirements document comments
  + Don’t bother supporting .DOC or .PDF
  + Make the text cleanup options into a set of user-selectable checkbox options
  + Take abbreviations out of the text cleanup options, due to ambiguity
  + Allow a user to remove particular tokens, like numbers, items that don’t have alpha characters, etc.
  + Maintain a working copy of the raw text in case user wants to adjust their cleaning parameters
  + There may be analysis dependencies
    - Coreference might require a dependency parse or some other syntactic analysis.
  + Think about best utilizing the library tools and domain tools for taking advantage in the analysis
    - Sequence-based analyses versus holistic ones, topic modeling and syntax/tree structured ones.
* Meeting outcomes: