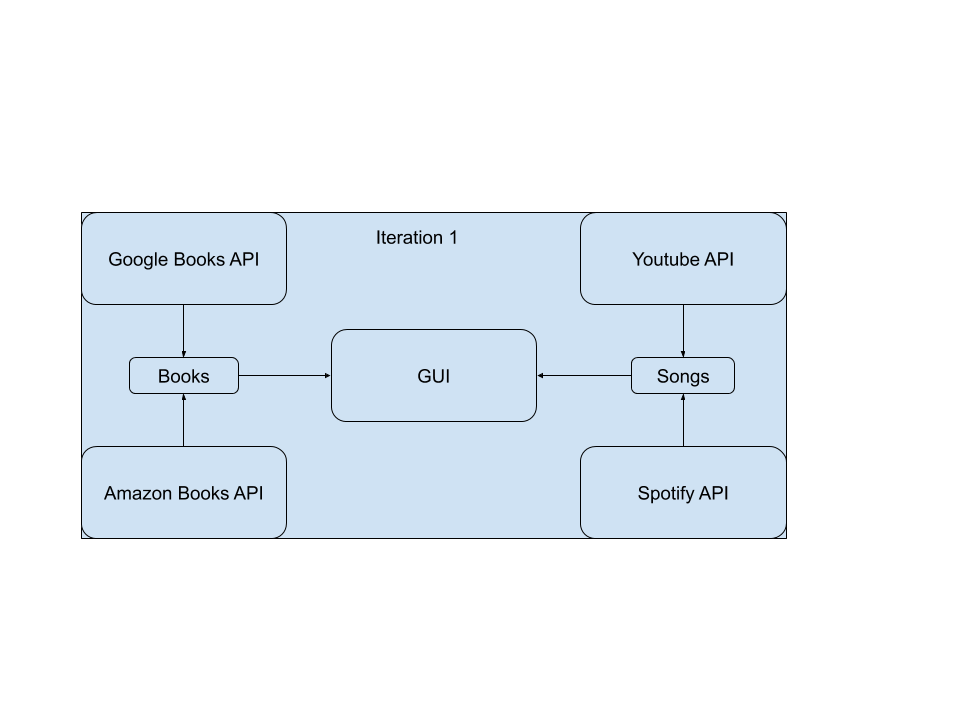
# Iteration 3 Report

Group RightAfterDeadline

04/17/2020

Design Diagram:



Cannot implement Barnes & Nobles API because it’s a private API (details below)

|  |  |
| --- | --- |
| Sicong Huang | SCRUM master, product owner, code review and merge |
| Aditya Atul Vijayvergia | Amazon Books API |
| Chen Liang | User Interface and File I/O |
| David Qin | YouTube API |
| Joseph Cisneros | Spotify API |
| Fengqiao Wang | Google Books API |

Pivotal Tracker: <https://www.pivotaltracker.com/n/projects/2437225>

GitHub repo: <https://github.com/Innoversa/Law-School-Copyright>

Meeting with Client: 4/10/2020

Discussed the details of how some APIs being implemented, also demoed the GUI to the client. The client gave supportive feedback and was satisfied with the current build

## Amazon Books API

Changes from using Amazon API to making independent Amazon book scrapper to allow quick and free access to books information. Implemented spider scrapper module in python as it allows access to Amazon website from code (direct scrapping using beautifulsoup4 redirects search to "I am not a Robot" page). Bypassed this using spider scrapper module.

Current version of code can only search for a single book that is manually entered in code. Change this to read multiple book names form an Excel sheet.

Spider scrapper can only access books if we already have Amazon product links for them. Implement an interface to get Amazon product links from book names.

Repeated redirecting to "I am not a robot" page when trying to access Amazon book information form python code.

## User Interface and File I/O

For this week, one task is to modify the user interface based on customer’s feedback. Additional buttons and textbox were added to achieve the functionality that users can specify the output of the results. Also, the function of reading xlsx and csv files is added as a module to the program. The code will read each sheet of the input spreadsheet as a dataframe, and then return the result to the scraping functions.

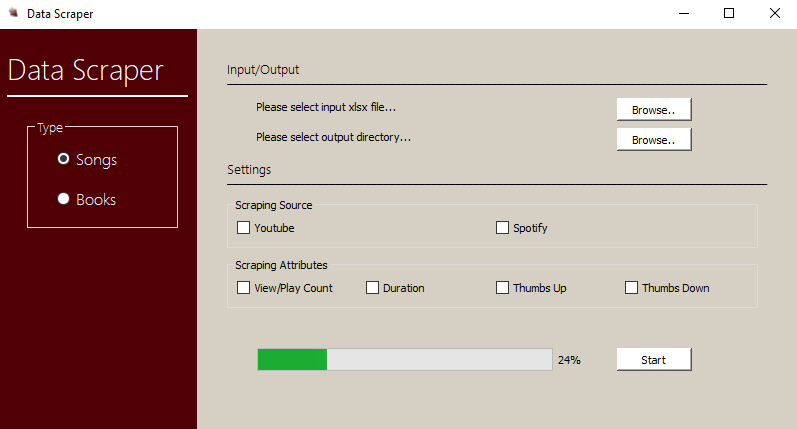


Image of New User Interface.

The task for next week is to make the controller running in parallel, so that scrapers for both songs and books can run in background at the same time.

## YouTube API

Moved YouTube query to work with ya music rather than just a top video

Best start integrating everything together

YouTube api does have severe limits on number of search queries per day, will try to take care of this

## Spotify API

Started trying to scrape from Spotify

Finish scraping from Spotify and verify that authentication stays valid

No technical issues, just illness and energy

## Google Book API

Accomplishment for this week was minor since I was busy with midterm. I tested the function of checking book prices with the book *Hamlet.* And it returned info on many different versions. And according to our client. These different versions should be seen as different books.

Goal for next week: I will make sure I can put in a book name and return the price. Also I might need help from my teammates on reading mass input to test the api. Because I am a little rightafterdeadline.