# Iteration 2 Report

Group RightAfterDeadline

04/03/2020

## Design Diagram

The design remains unchanged

|  |  |
| --- | --- |
| Sicong Huang | SCRUM master, product owner, code review and merge |
| Aditya Atul Vijayvergia | Amazon Books API |
| Chen Liang | GUI |
| David Qin | YouTube API |
| Joseph Cisneros | Spotify API / Barnes Noble API |
| Fengqiao Wang | Google Books API |

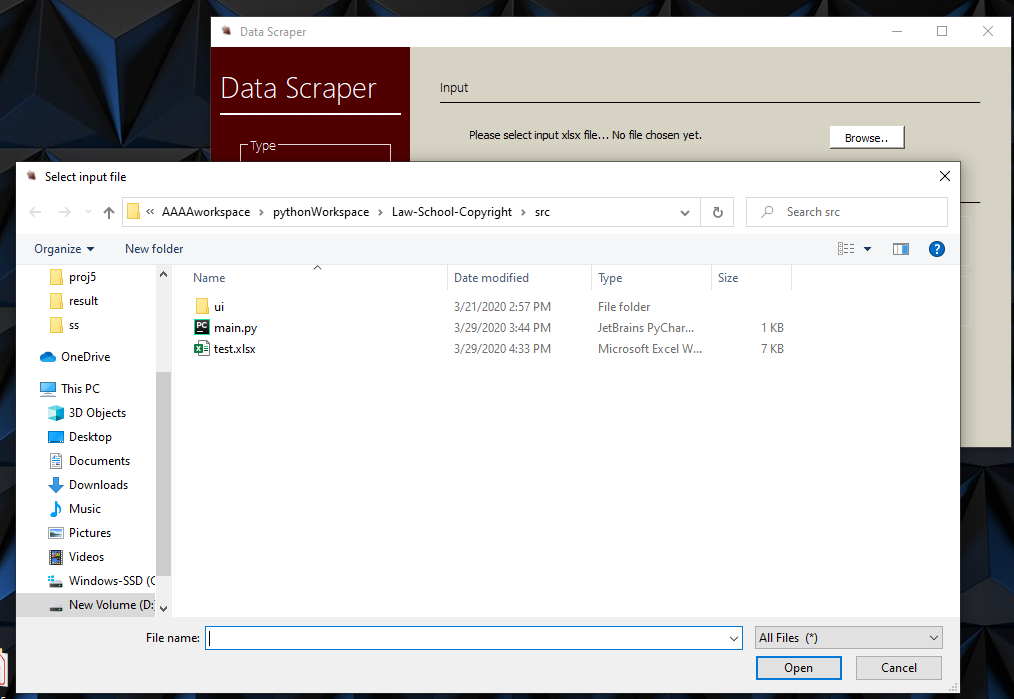
## Question to the Client

For our sponsor, we must ask for some clarifications. **What should we do with duplicate results?** Every book/song will almost always have multiple uploaders/sellers, currently our portions will usually display information for the first result queried. Is this acceptable, or should

we do more?

## Graphical User Interface (GUI)

First version of Graphical User Interface is finished. The corresponding tasks on Pivotal Tracker are also marked as complete and were approved by scrum master. In this iteration, the view model of GUI is complete. All UI widgets and interaction should be functioning now, and information entered on UI will also be recorded for scraping programs to use. Whenever user clicks the type of information they want to retrieve (i.e. books or songs), the UI will update widgets accordingly to show relevant information. Also, users now can select the input file from a pop up file explorer window to make the process easier, as shown below:



In addition, in this iteration, program logic is separated from view. Now a controller class is created to handle signal and slots, events, and other functionality of the program. Other classes can call a controller function to get all current information on UI to finish their tasks. This can make the code clearer and more readable.

For the following iterations, the task is to work with other teammates to get their function connected to the UI. Currently those functions are marked as ‘not implemented’ and will be updated once scraping functions are complete.

## Amazon Books API

Finished with testing multiple APIs for getting books information other than Amazon books API. These APIs include goodsread API, bookshare API, bookworm e-pub, catalogWS, New York Times bestseller API and some of the lesser-known ones.

A major problem with most APIs is that they don’t have all the books and the information is also limited for a large portion of books. Amazon API has information on most books among all the above options and it follows a standard format with the proper information for every book.

Amazon API needs Amazon associate sign-up. An easier way is to simply scrape amazon.com for book results when the book is searched by ISBN. Implementation of Amazon Web scraping tested for basic information like book title and price.

Some issues were faced during this iteration progress.

One book has multiple versions - hardcover, paperback, audiobook, and ebook. Information like ratings and prices vary depending on the type of book. We need to discuss with the customer on which type of book to use for Web scraping.

None of the tested APIs provide the number of books sold. Needs further discussion with the customer.

## Google Books API

Almost done with testing the API. Chance is that this api will work well to fulfill the requirements of our client. The price checking functionality has been tested along with some other functionalities that can help with this project,

A limit for google books api is the quota limitation. I am not sure how much search our client will want to make per day. The default limitation is 1000 per day. So if the search includes many books and many versions. We might need to consider applying for a professional license。

Therefore we need a little more testing on the Google books api. And ask our client about the search per day that he wants to make.

## YouTube API

Currently meets the goals of being able to gather information off of a video given search queries such as the title and authors/artists. Is able to read in the file and print out information that’s desired aside from video length.

Next goals will be to handle how the google API works. Currently, the program forces you to authenticate yourself manually and has security issues involved from not setting up properly. Otherwise, the next goal will be to integrate this with the GUI.

## Spotify API / Barnes Noble API

Accomplished: Read documentation, registered for Spotify developer account, created Spotify client id and client secret

Plan: Run initial queries and verify the Spotify authentication pipeline works as expected

Obstacle: Weird Spotify api authentication requirements

## Wrap-up

Will be demoing to the client once he replies the email. We have finished individual parts and now ready to start the first big merge. The project proceeding successful. We need to wait to meet our client to get details on how he wants the data be presented.