

Congenial Broccolis: Wen Hao Dong (Jal Hordan), Austin Ngan (Gerald), Liesel Wong (King Hagrid), Rachel Xiao (Mooana)  
SoftDev  
P01 -- ArRESTed Development  
2021-12-09

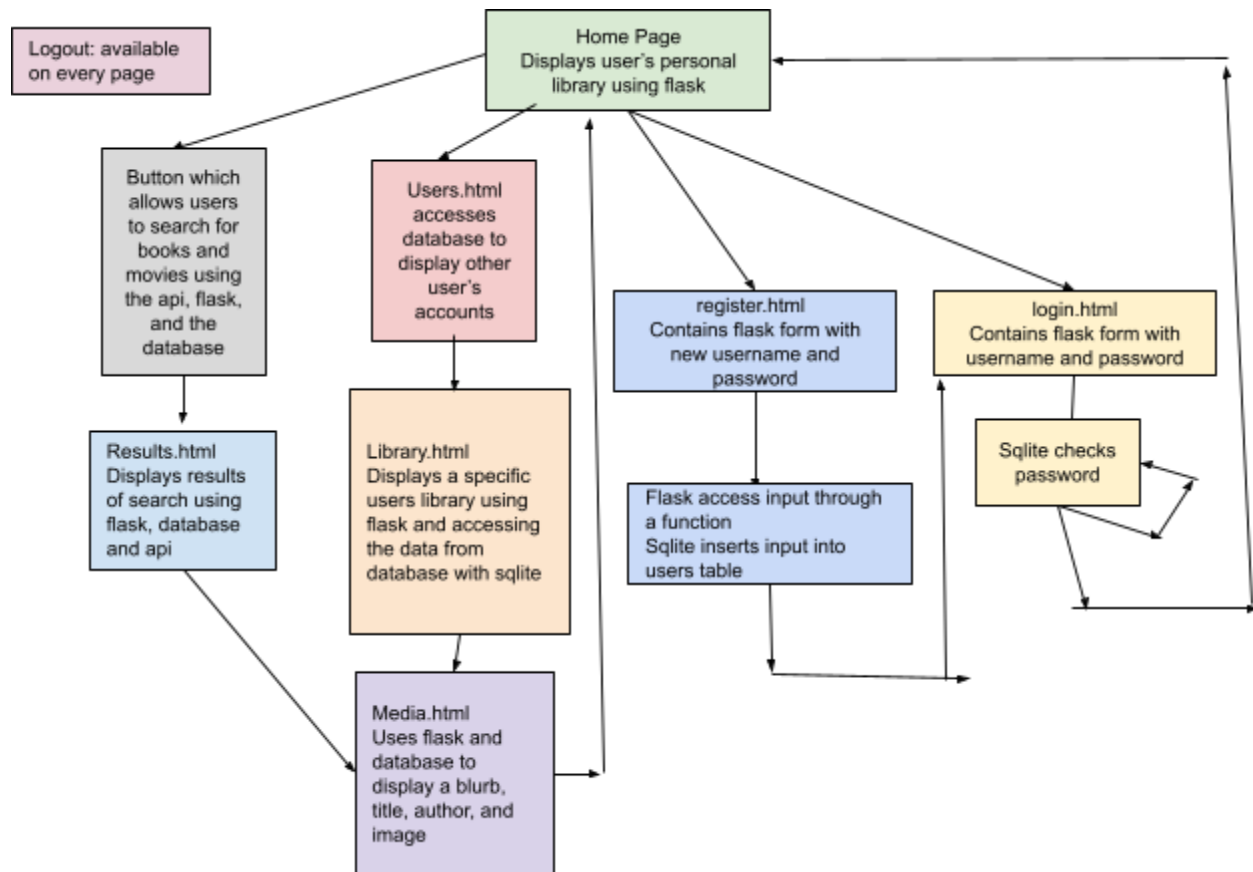
### Description:

An online library containing a personalized selection list of books and movies. Each user has their own personal library, which has *only* the books and movies they added to. They can add these via a search or by looking through other users' libraries. The user can mark a book or movie as finished or not finished. Users can also remove books or movies from their library.

### Components and Purpose:

- SQL database:
  - Keep track of login user information
  - Stores information for the book and movie libraries
  - Stores user entries when they add a new entry to their library
  - Obtains user inputs from Flask
- Bootstrap:
  - Builds a responsive website with CSS and JS templates that makes designing the website easy
- Jinja2:
  - Create a layout for web pages the user will interact with
  - Renders templates to populate the website with some user specific information
- Flask:
  - Uses Jinja2 and HTML templates to create the web pages that the user will see and interact with
  - Data is pulled from the database in order to populate each webpage
  - Will get username and passwords from the front end when a user logs in or registers
  - Will create sessions for each user when they are logged in
  - NOTE: login/logout is subject to removal if deemed impractical
- Main Page:
  - User's personal library containing all the books/movies they have consumed
  - Contains an add book and add movie button to add to their respective database
- Add Book/Movie:
  - Search button (searching through IMDB or OpenLibrary)
  - Returns entries that have been found through the API
  - User can then choose which entry they want to add to their personal library

## Component Map:



## Database Organization:

users:

|                          |                 |                 |
|--------------------------|-----------------|-----------------|
| id (integer primary key) | username (text) | password (text) |
|--------------------------|-----------------|-----------------|

movies (for caching movie information):

|  |              |                  |                |             |
|--|--------------|------------------|----------------|-------------|
| id (integer primary key)<br>the same "tt" id imdb uses | title (text) | cover_url (text) | summary (text) | link (text) |
|--|--------------|------------------|----------------|-------------|

books (for caching book information):

|   |                 |                     |                  |                   |                 |
|---|-----------------|---------------------|------------------|-------------------|-----------------|
| id (integer<br>primary<br>key)<br>OpenLibrary<br>'s work id | title<br>(text) | cover_url<br>(text) | author<br>(text) | summary<br>(text) | pages<br>(text) |
|---|-----------------|---------------------|------------------|-------------------|-----------------|

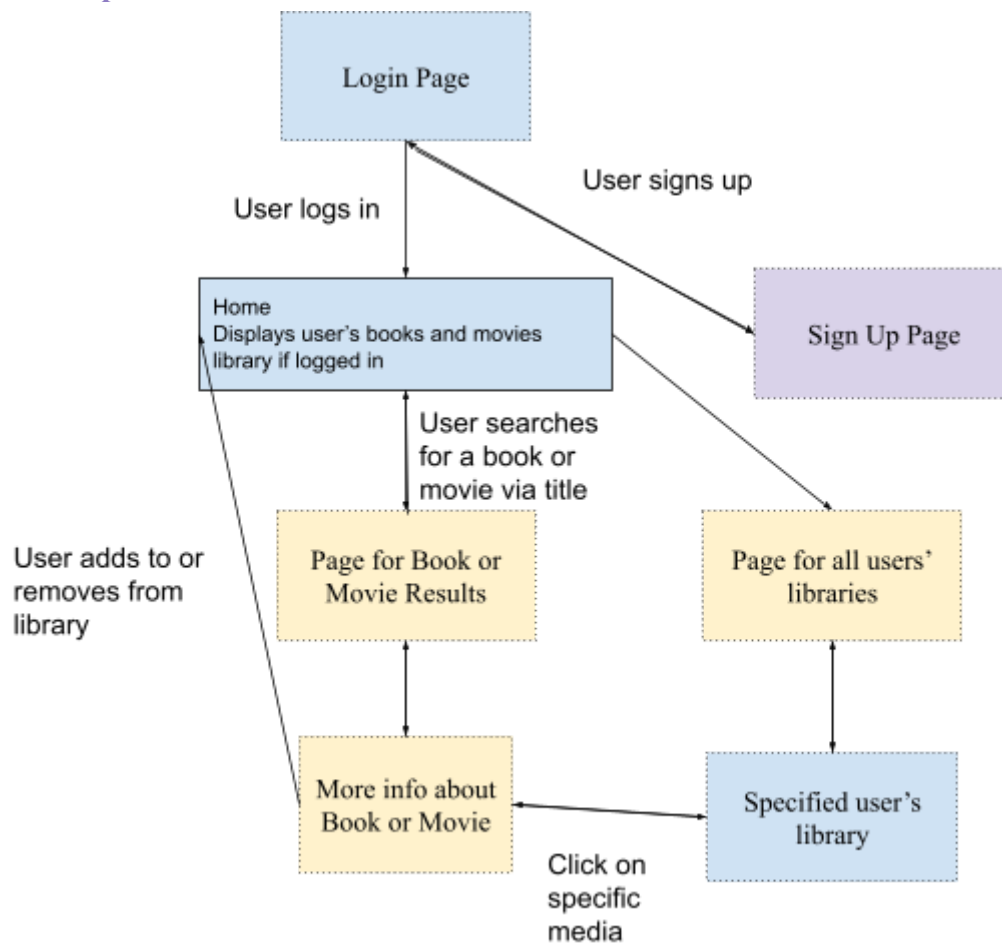
entries: whenever a user adds a book or movie to their library, a new entry is added with their user id and the id of the thing they added

|  |                      |  |                       |
|--|----------------------|--|-----------------------|
| type (text)<br>either "book"<br>or "movie" | user_id<br>(integer) | media_id<br>(integer) the<br>id of the<br>book/movie | complete<br>(boolean) |
|--|----------------------|--|-----------------------|

#### APIs used:

- IMDB API
  - Provides information (cast, ratings, trailers, posters) about movies with many ways of searching for them
- OpenLibrary API
  - Retrieves information about books using many book APIs
- NYT Movie and Books Reviews API
  - Provides subjective information about movies or books
- Google Books API
  - Provides information about a specific book

## Site Map:



## Front End Framework (Bootstrap):

We're using Bootstrap because we like the CSS styling they have. The default styling is cleaner and easier to implement, thus, easily giving our website more of a professional feel to it. We believe it has more features than Foundation. It also has a lot more documentation than Foundation because it is a very popular framework. Compared to Foundation, which for the most part requires a download, Bootstrap allows you to use the CSS through 1 to 2 external stylesheets. Lastly, it is more difficult to find ways to use Javascript in Foundation than in Bootstrap.

### Template Representations:

NOTE: design and color are subject to change (in particular the library “finished” switches, cause it may need JS to work properly)

Login:

## Login

Login

Register:

## Register

Login

Library and Home:

## User's Library

Title

Cover Image

Finished ☒

Title

Cover Image

Finished ☐

Title

Cover Image

Finished ☐

Title

Cover Image

Finished ☒

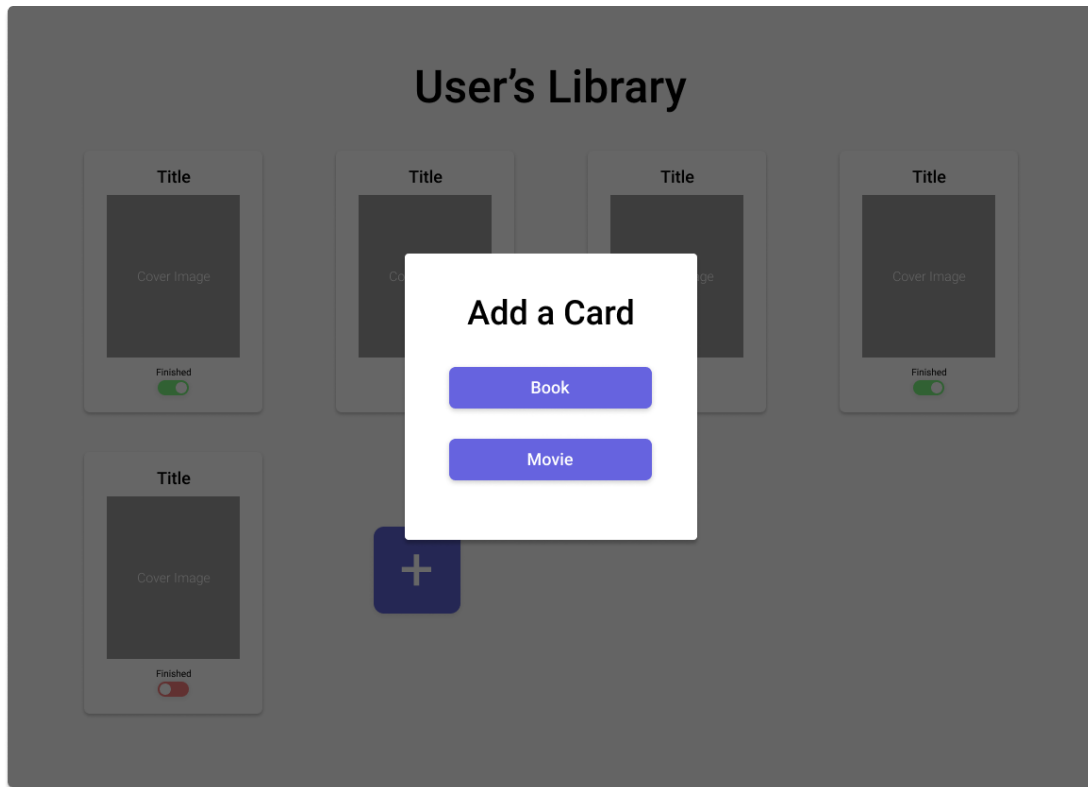
Title

Cover Image

Finished ☐

+

Select card type modal (pop up)



Search:

| Search for Book                       | Search for Movie                      |
|---------------------------------------|---------------------------------------|
| <input type="text" value="Title"/>    | <input type="text" value="Title"/>    |
| <input type="button" value="Search"/> | <input type="button" value="Search"/> |

Results:

## Results for "Title"

Title

Cover Image

Add to Library

Title

Cover Image

Add to Library

Title

Cover Image

Add to Library

Title

Cover Image

Add to Library

Title

Cover Image

Add to Library

Title

Cover Image

Add to Library

Title

Cover Image

Add to Library

Users:

## Other Libraries

user1

user2

Book/Movie:



Pages/Link

Summary

**Division of Tasks:**

Wen Hao: Project Manager and Flask

Austin: Flask

Liesel: Bootstrap / Templates

Rachel: Database