"BookShelf" Web App Report

Author: Haiqiang Zhang & Jinhua Fan

We design the "BookShelf" Web App to show books information, manage user's reading data, and provide personal recommendations to specific users. We hope it will develop into the best **digital reading management tool!**

User can use the web app to realize the entire process of reading management: choosing book → manage favourite book list → add book to read list → compare with other user to stimulate the reading passion → review books.

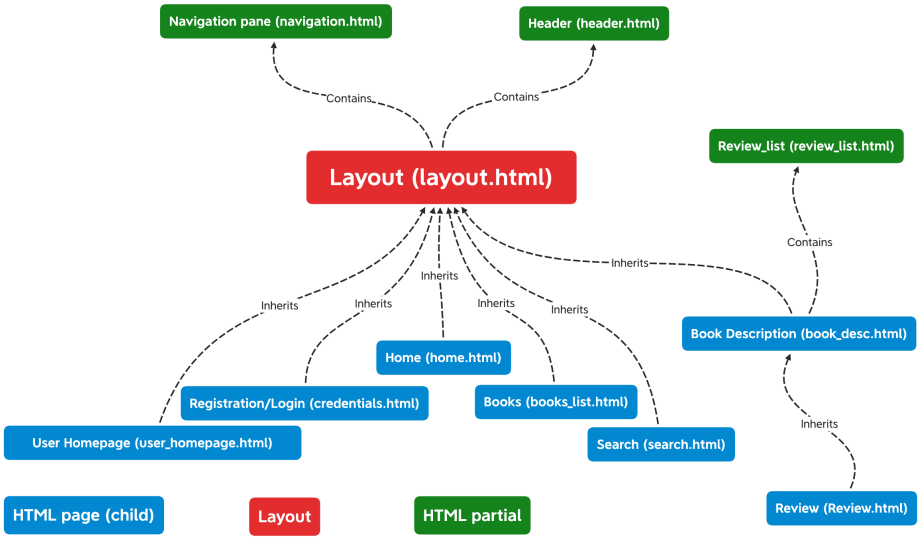
We will continuous develop the web app in the future.

I will cover the features of each unit separately below:

- Front-end structure
- Web-Application architecture
- Search
  - Local search function
  - Categorical search
  - Code implementation
- Book exhibit
  - Change pages
  - Tag page
  - Book description
    - Review
    - Rating
  - Favourite Books
  - Read Books
  - Code implementation
- User Homepage
  - Reading Statistics and comparison
  - Show and setting user's tag
  - Read books and favourite books entrance
  - Code implementation
- Recommendation & Tag System
  - Recommendation
    - Recommendation Mechanism
  - Tag System
    - Backstage management tag
    - See all tags
    - Category books by tag
    - Show each book's tags
    - Showing and setting user's tags
    - Auto add tag for user
- Test
  - Pytest
  - Test blueprint

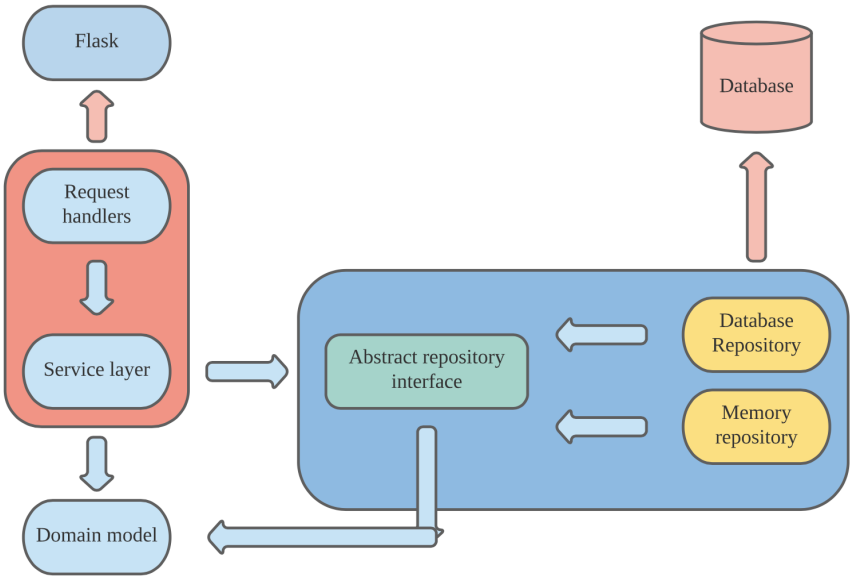
Front-end structure

This diagram below shows the front-end structure of the "bookshelf" web app:



Web-Application architecture

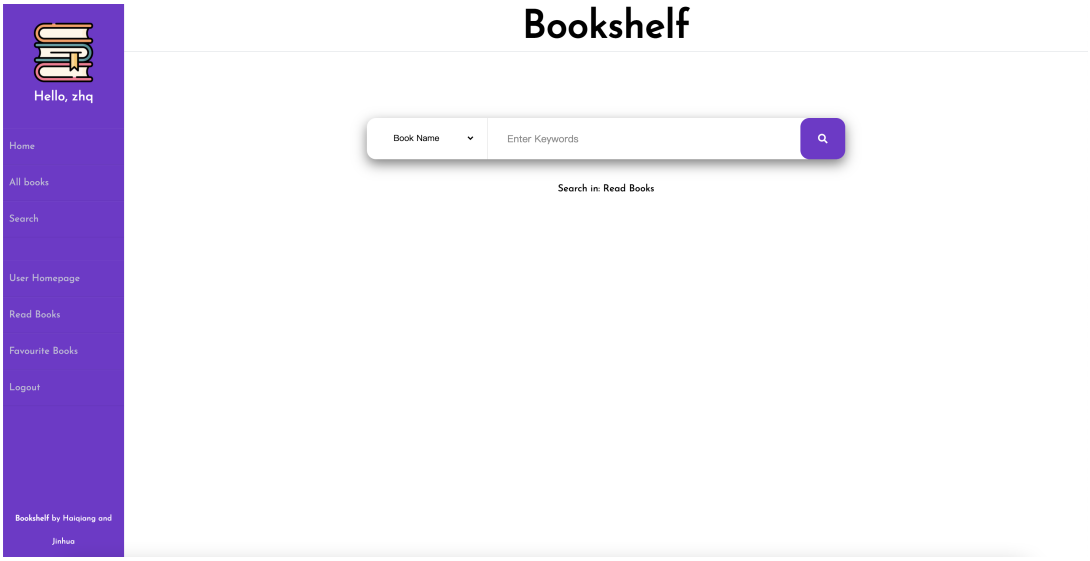
The diagram below show the back-end architecture of "bookshelf" web app:



- We follow the architecture in the development.
- we follow Single responsibility principle.

Search

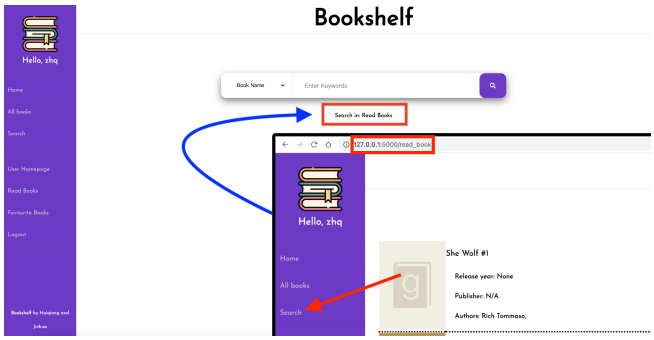
Search page has a simple and beautiful UI with Full functionality.



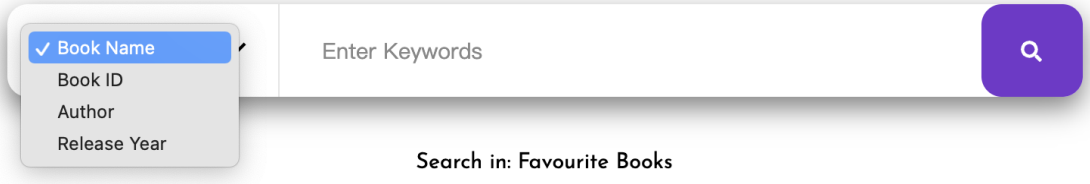
Local search function

We can click search button in any pages.

When we click search button in All books, read books, favourite books' list, the page only search this part of books. And the Search scope show in the page.



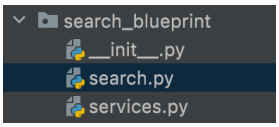
Categorical search



- You can choose four search types to search.
- Book Name and Author support Fuzzy search technology.

Code implementation

- I make a new blueprint to implement search function with services layer.




```
search_blueprint = Blueprint(
    'search_bp', __name__
)
```

- I make two views in search blueprint.
- I use two method in services to connect repository and finish search algorithm.

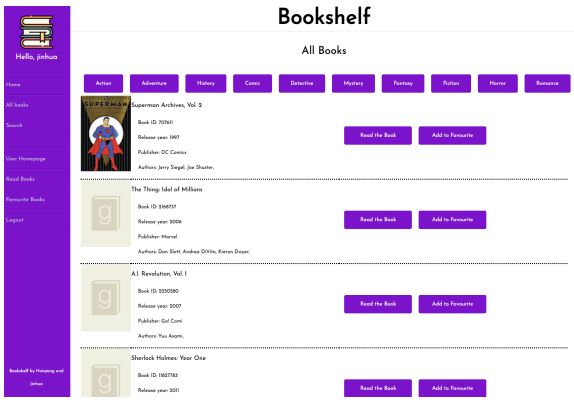
```
def get_search_books(request_search_scope:str, repo_instance:AbstractRepository)
def search_book(repo_instance:AbstractRepository, select_items, search_content, list_book)
```

- Because you need to search for books in different ranges, you don't search in the all books list every time, so you can't call repo_instance directly. So I added several methods to the memory repository specifically to search for a given list.

```
def get_books_by_title_and_given_list(list_book: List[Book], title: str)
def get_book_by_id_and_given_list(list_book: List[Book], id: int)
def get_books_by_author_and_given_list(list_book: List[Book], author_input: str)
def get_books_by_year_and_given_list(list_book: List[Book], release_year: int)
```

Book exhibit

On all books page, you may find all the books that we have on our website. We have ten tags that you can choose to see the books with the specific tag. If you click on the Action button on the top, you will be redirected to the web page containing all the books with this tag.

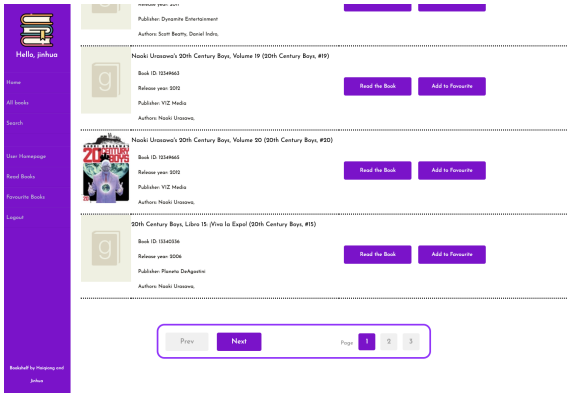


You can add this book to your Read Books shelf or Favourite Books shelf by click on the two buttons on the right of different books. Clicking on the book title and book

image, you will see the full details of this book.

Change pages

At the bottom, we also implement a page select bar. You can go to these pages by click on the different numbers or click on the Next button to get to the next page, as shown below.



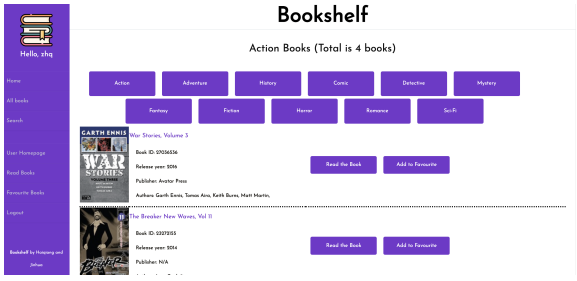
I use `form_book_list()` in book_blueprint → services to abstract change pages function to general the function.

```
def form_book_list(target_page, books_list, url, tag_str = None)
```

Tag page

This is the Action Books page. On this page, you will see all the books related to the tag of Action, and on the top, you will see the title of Action Books and the number of books.

You can read and add to favourite in the page directly.

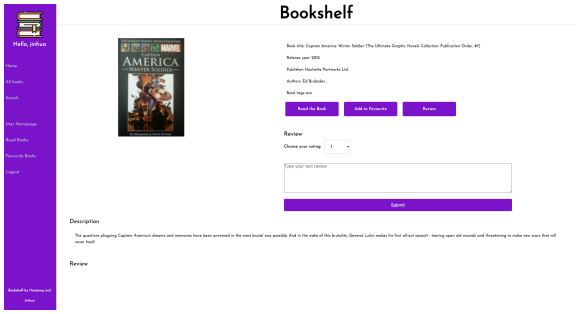


Book description

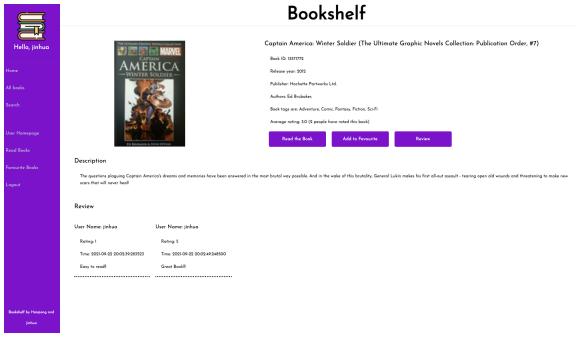
On the book description page, we can see this book's full details and all the reviews for this book. If we click on the Review Button, an adding review window will show up, as shown below.

Review

On the review window, we can comment this book and set our rating for this book as well. If we click on the Submit Button, this book's review will automatically display the reviews at the bottom of the book description page.

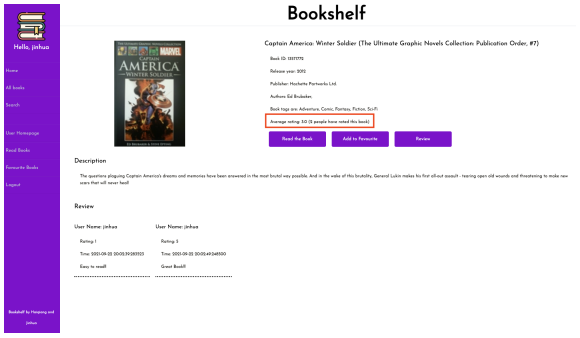


Afterwards, you will see the new review of this book is added. Each book review contains the user name, rating by the user, submit time and the comment. Recent reviews will be appended after the older reviews.



Rating

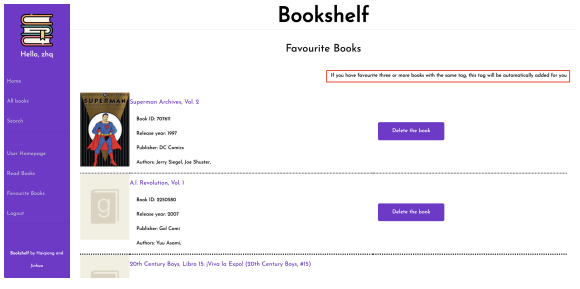
- There is a rating about this book based on the review ratings when the number of reviews is greater than zero.
- If the number of reviews is zero, average rating will not be shown.



Favourite Books

We can understand that the Favourite book list provides users with the function of marking important or very like books, or the user’s todo list

We have two ways to get to Favourite Books and Read Books. We can access



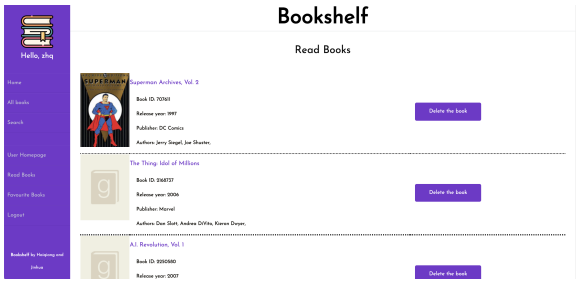
from the left navigation bar and user homepage shown above to get to these two web pages when the user has logged in.

If you have favourite three or more books with the same tag, this tag will be automatically added for you.

You can delete book in the page.

Read Books

On this page, you can see all the books you have read, and we cannot add the book to read books or favourite books on this page. You can click on the image or the book title to access the full description of this book shown in the following screenshot.



You can delete book in the page.

Code implementation

These functions are all about book exhibitions. So I set these functions(views) to one blueprint.

```
book_blueprint = Blueprint(
    'book_bp', __name__
)
```

User Homepage

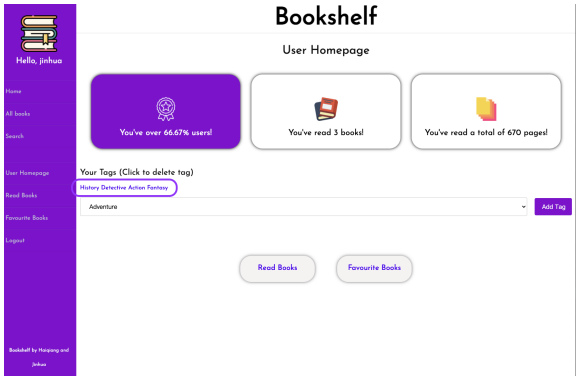
Reading Statistics and comparison

On the first row, you can find how many books and pages you have read on the user homepage, and your total number of reading books have defeated how many users.

- It is clear
- A fusion of Skeuomorph and Flat design

Show and setting user's tag

Then, in the middle of the page, there are your tags. You can add specific tags, which the web page will recommend the books with your favourite tags to you. You can easily add tags by choose your favourite tags on the combo box and click on add tag. Then, your tags will show up. You may click on these tags if you don't want these tags anymore. This page is only shown when the user has logged in, and your user name will appear on the top left.



Read books and favourite books entrance

You may also access your read books and favourite books by clicking on the Read Books button and Favourite Books button in the middle of the user homepage. Then, it will redirect you to the new web page that contains all the books you read or favourited.

Code implementation

I set a separate blueprint to implement user homepage

```
user_homepage_blueprint = Blueprint(
    'user_homepage_bp', __name__
)
```

Recommendation & Tag System

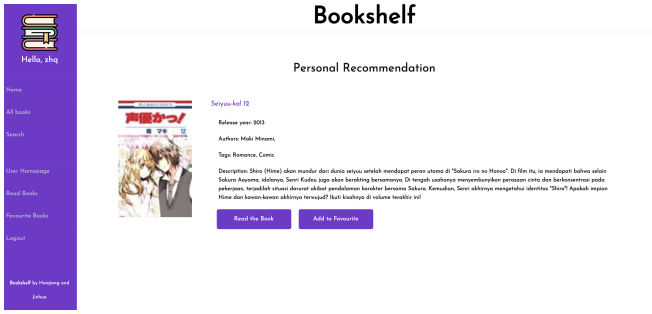
Recommendation

We will recommend books on the

homepage, and we can also read and favourite the recommended books directly.

When we click on the picture or title, we will enter the book details interface.

Let me introduce our recommendation mechanism below.



Recommendation Mechanism

- For users who are not logged in, we will randomly recommend books.
- For users who have logged in, we will recommend books based on the user's tag.
- We will not recommend books that users have read or are in the favorite list.
- We will automatically add tags to users based on their favorite book list.
- But when a user deletes a book in the favorite book list, the system will not automatically delete the user's tag, because this is the user's historical preference.

Tag System

Backstage management tag

We only need to add tags to ***tags.csv*** to manage the total tag types of this web app

- The ***tags.csv*** path: `library/adapters/data/tags.csv`
- Add one new row for new tags

```
tag_name
Action
Adventure
History
Comic
Detective
Mystery
Fantasy
Fiction
Horror
Romance
Sci-Fi
```

See all tags

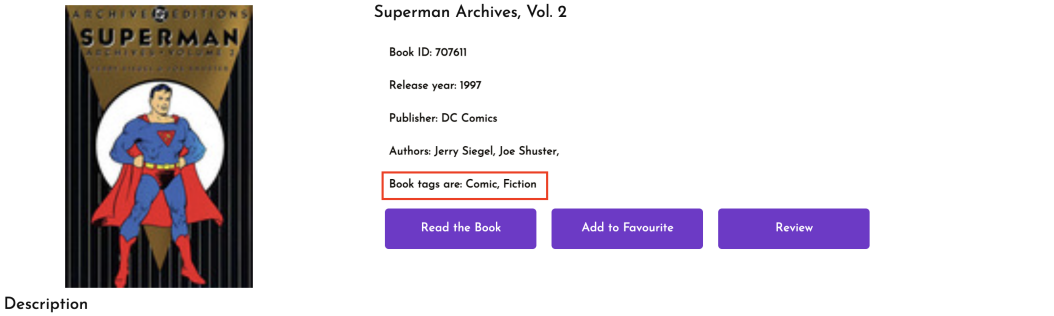
We can see all tags in all books list page.

Category books by tag

I have introduced this functions in ***Book exhibit → Tag page***

Show each book's tags

we can see each book's tags in book description page.



Showing and setting user's tags

I have introduced this functions in ***User Homepage → Show and setting user's tag***

Auto add tag for user

If you have favourite three or more books with the same tag, this tag will be automatically added for you.

We can see user tag in user homepage page.

Test

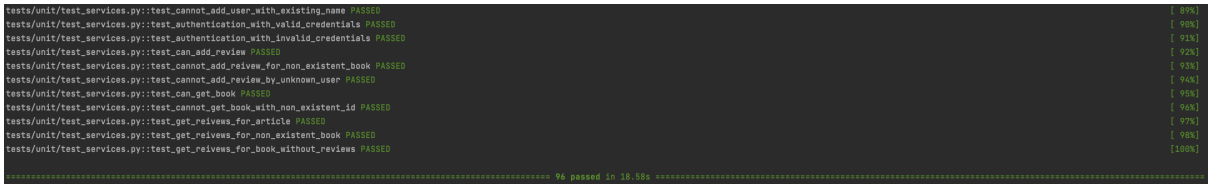
Pytest

I create pytest to test the web app.

Using the command below to run pytest:

```
$ python -m pytest -v tests
```

All tests are passed.



Test blueprint

I use the test_blueprint to debug. It can show specific variables' content to debug.

> 📌 The content will be shown in http://127.0.0.1:5000/test

We can call `get_test_content(content)` to transfer parameters