

Dvanced Java and Angular Project Specification

Peter Fodor

November 10, 2024

1 Description

The goal of this project is to implement a simple **library management** system, where users can do the following activities:

- browse through the online catalog based on a variety of filtering and sorting options,
- borrow books,
- list the books they have borrowed so far, also based on several filtering and sorting options,
- Users with **admin** permissions can modify the catalog based on whether a lending service has been requested by a customer
- ...

Although people can borrow material other than books such as CDs and audio-books, let's assume the library only has a collection of printed books that must be borrowed in person.

2 Data Domains

This section introduces the data that will be used for the management system. Although most of the field have been defined, it is possible that new field will be added if it is necessary.

1. **User:** Basic information regarding the registered users, and admins of the system. As there might be services that require someone to be registered (e.g. borrowing a book), the following user categories should be defined: **User** and **Admin**
2. **Book:** the user can browse through the collection of books, search based on several parameters, such as title, author, year of publication and page number.

3. **Author:** basic pieces of information about the author of a book, for example the name of the author, year of birth, place of birth, and so on. Normally, books could have more than one author, but in this system, let's assume that **each book only has one author**.
4. **Item:** Since there can be several instances of the same book, they should be handled as separate items. This entity could also contain information regarding the lending services, such as the availability of the item, the condition of item, and so on. Additionally, whenever admins process a request for a lending service, they should browse through the collection of Items to check if the book in question is available for the user, and modify the collection, for example when a new item is given to the library, an item is given away from the library, and so on.
5. **Rent:** Regular users can borrow books from the library. In order to keep track of the books that have been lent to users, a collection of **Rents** must be used, which includes information about the requested book, the user who borrowed a book, the starting date and deadline of the Rent, and possible extensions.

Figure 1. shows the **Entity-Relationship Diagram** of the data used for the management system.

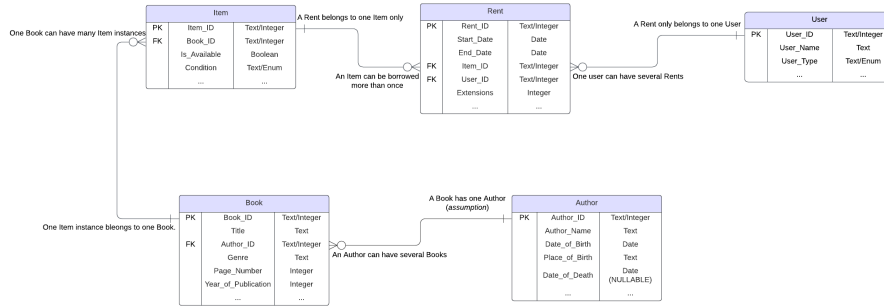


Figure 1: Entity-Relationship Diagram of the library management system

3 Functionalities

3.1 API

There is a wide variety of endpoints that can be used in a library management system, depending on what the user and the admin would like to achieve. Table 1. contains a summary of most API request that should be implemented in the system.

Name of endpoint	Type of endpoint	Description	Special Properties
GetBooks	GET	List all Book instances	Filtering, Sorting
GetBookByID	GET	Get a Book instance based on its ID	-
GetAbstract	GET	Get the abstract of a Book	non-JSON resource
GetCover	GET	Get the cover image of a Book	non-JSON resource
GetItems	GET	List all Item instances	Filtering, Sorting
GetItemByID	GET	Get an Item instance based on its ID	-
GetAuthors	GET	List all Author instances	Filtering, Sorting
GetAuthorByID	GET	Get an Author instance based on its ID	-
GetUsers	GET	List all Users	Filtering, Sorting
GetUserByID	GET	Get a User based on its ID	-
GetRents	GET	List all processed Rents	Filtering, Sorting
GetRentByID	GET	Get a Rent instance based on its ID	-
PostBook	POST	Create a new Book instance	-
PostAuthor	POST	Create a new Author instance	-
PostItem	POST	Create a new Item instance	-
PostRent	POST	Create a new Rent request	-
SendRequest	POST	Send a request for e.g. borrowing a book	-
RegisterUser	POST	Add a new User to the system	-
ChangeEntity	PUT	Change any properties on an instance	For all entities
BorrowBook	PUT	Borrow a book from the library	-
ExtendDeadline	PUT	Extend a deadline of a Rent instance	-
DeleteItem	DELETE	Delete an Item instance	-
DeleteBook	DELETE	Delete a Book instance	-
DeleteAuthor	DELETE	Delete an Author instance	-
DeleteUser	DELETE	Delete a User	-
DeleteRent	DELETE	Delete a Rent instance	-

Table 1: Table of most API requests that should be implemented

3.2 Security and user management

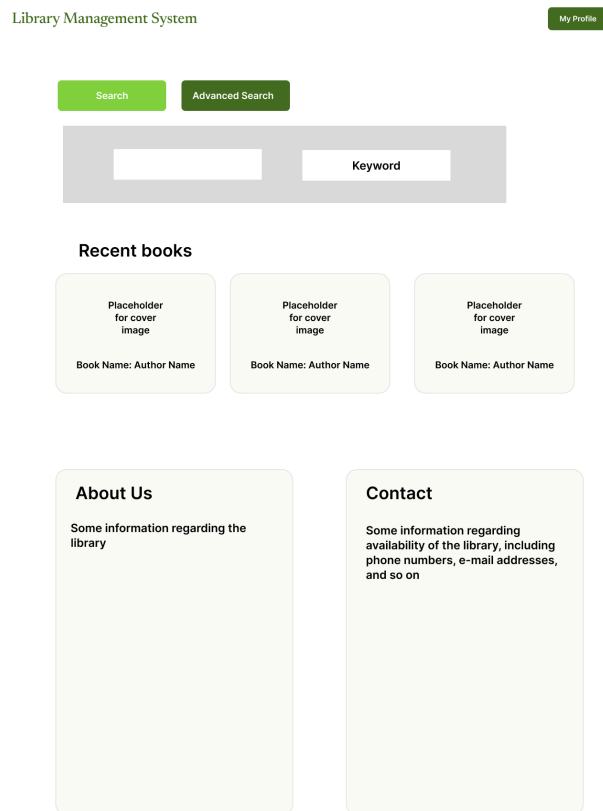
As I have mentioned before, there are 2 user categories:

1. (Regular) **User**: A regular user can borrow books, request an extension of deadline on ongoing rents, and so on.
2. **Admin**: An admin has the ability to process the requests sent by regular users, and has access to requests that can modify the existing data, e.g. the DELETE requests,

4 Views

This section contains a sample of some pages that is included in the project. The style present on the views, and some of the layouts are subject to change.

Additionally, forms will also be included, for instance for creating new items in the catalog.



The mockup for the Library Management System homepage features a clean, modern design. At the top, the title "Library Management System" is on the left, and a "My Profile" button is on the right. Below this, there are two search buttons: "Search" and "Advanced Search". A search bar with a "Keyword" label is positioned below the buttons. The "Recent books" section displays three book cards, each with a placeholder for a cover image and the text "Book Name: Author Name". At the bottom, there are two columns: "About Us" with the text "Some information regarding the library" and "Contact" with the text "Some information regarding availability of the library, including phone numbers, e-mail addresses, and so on".

Library Management System

My Profile

Search Advanced Search

Keyword

Recent books

Placeholder for cover image
Book Name: Author Name

Placeholder for cover image
Book Name: Author Name

Placeholder for cover image
Book Name: Author Name

About Us
Some information regarding the library

Contact
Some information regarding availability of the library, including phone numbers, e-mail addresses, and so on

Figure 2: Homepage

Register

First Name:

Surname:

Username

Password

Other Pieces of information should also be included here

[Have you already registered? Log in here.](#)

Figure 3: Page for registration

Login

Username

Password

[Are you new? Register here.](#)

Figure 4: Page for logging in

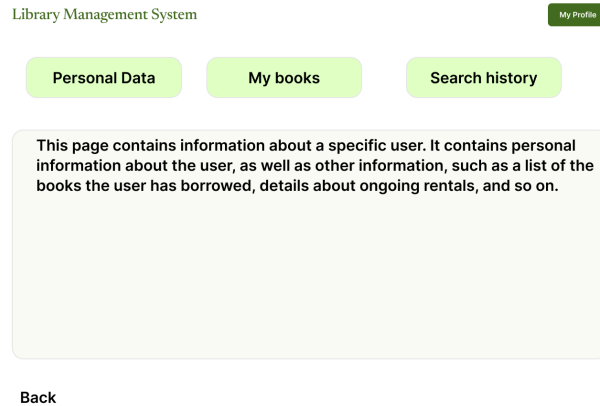


Figure 5: User Page

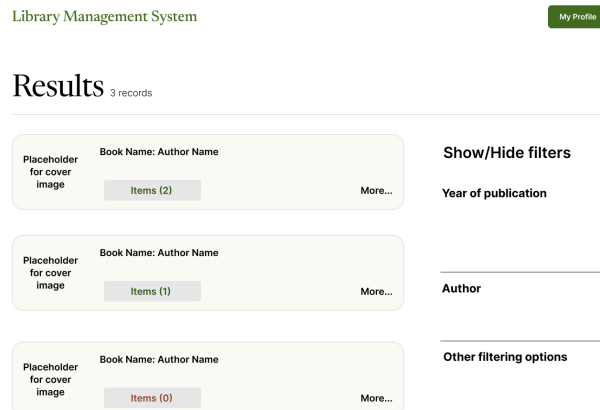


Figure 6: Page of search results

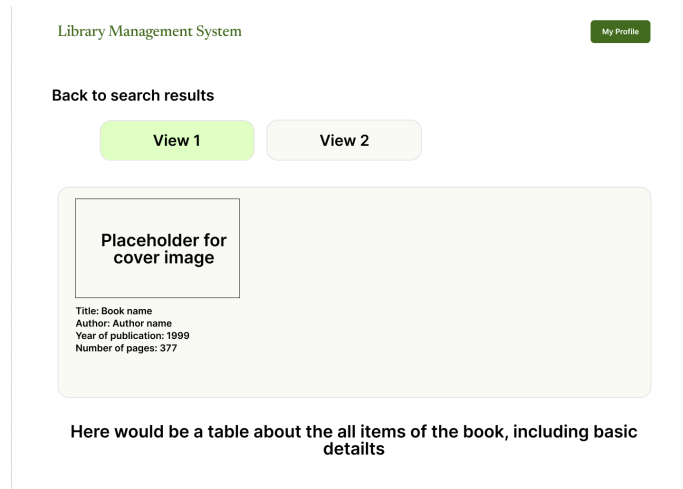


Figure 7: Page where details of a book are listed

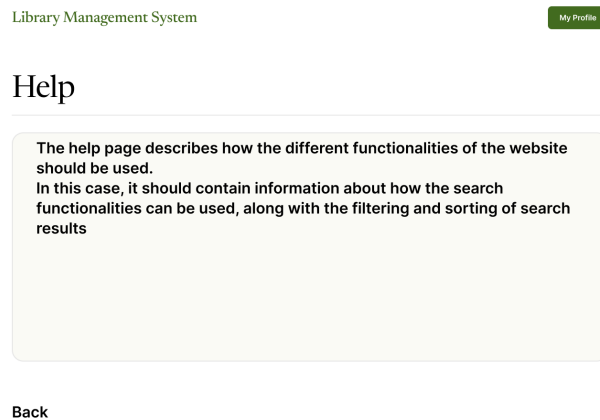


Figure 8: Help Page