Library Management System

Design Document

Author Philipp Dreyfuss

Date: 25.07.2022

Table of Contents

Overview

Library Management System Project is a SaaS program which manages the resources of a library. It includes multiple REST API's to serve client apps/third parties. The main users of this program are Librarians and end-users.

Functional Description

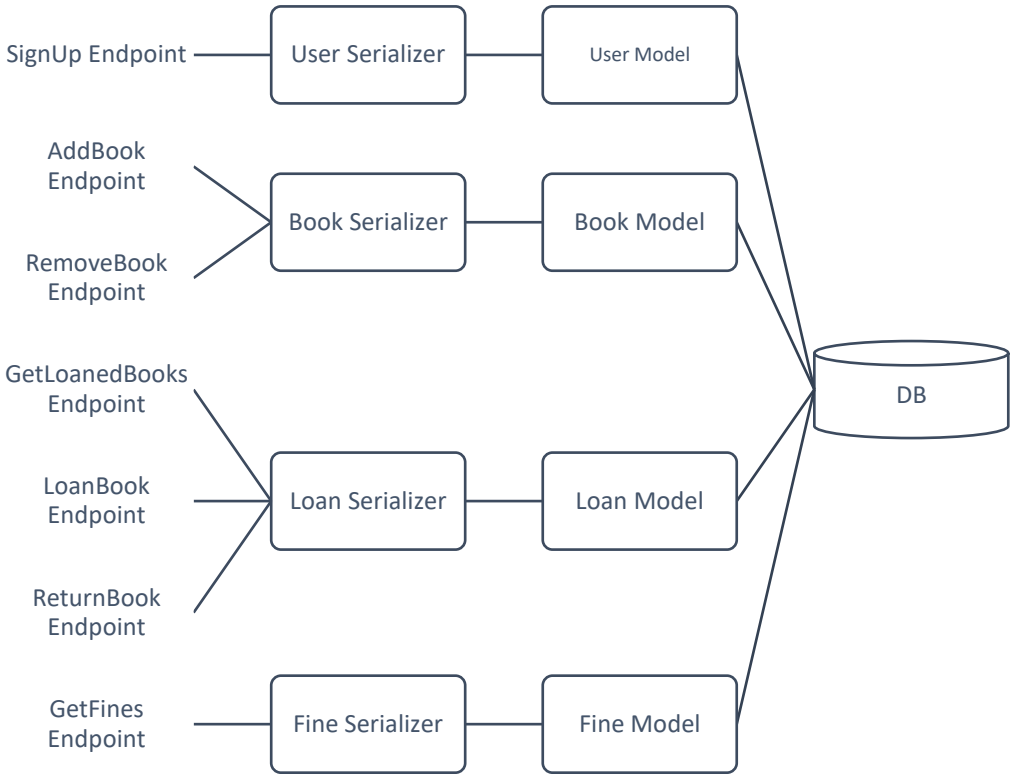
- First we need to create new users. They can be basic users or staff members.
- A staff member can add books and remove them from the system.
- Once the catalog is well defined, a user (basic or staff) can loan (limited to ten books) and return a book.
- A basic user can see all loaned books and a staff member can see the loaned books of all users.
- A due-date is defined as 14 days from the loan date. If a book was returned after its due date, a fine is placed. For each day the book is overdue, the fine increases by 10 cents. The user can see his non-paid fines.

Detailed Design

Our Database will be of type SQL, since we are going to use Django and it works hand in hand with SQLite which is of type SQL.

Data Tables:

Books	Barcode	String
	Author	String
	Title	String
	Isbn	String
	In place	Boolean
User	Username	String
	email	String
	password	String
	Is staff	Boolean
Loan	Loan date	date
	Due date	date
	Book	Book
	User	User
	Return Date	date
	Is Loaned	Boolean
Fine	Is Fine Payed	Boolean
	Paid date	date
	Loan	Loan



Models:

The models will contain the data structures.

Serializers:

The serializers are in charge of serializing and deserializing the data.

Views:

SignUp Endpoint:

- Validate input for new user.
- Create new User

AddBook Endpoint:

- Check if authenticated with oauth.
- Check permission if allowed to add books.
- Validate input for new Book
- Create new Book

RemoveBook Endpoint:

- Check if authenticated with oauth.
- Check permission if allowed to add books.
- Check If barcode is valid and exists
- Delete Book

GetLoanedBooks Endpoint

- Check if authenticated with oauth.
- If user is staff return all loaned items with user.
- Else return all loaned items of specific user.

LoanBook Endpoint

- Check if authenticated with oauth.
- Check if has less then 10 books loaned
- Check if item is for loan.
- Set book not In place.
- Create Loan with due date in 14 days.

ReturnBook Endpoint

- Check if authenticated with oauth.
- Get book by its barcode
- Get Loan by the loaned barcode
- Set Book as in place
- Set Loan as not loaned and set return date.
- If loan over due then create a Fine

GetFines Endpoint

- Check if authenticated with oauth.
- Get loans with non paid fines
- Calculate fines for over due days
- Return all non paid fines of specific user

Security/Concern

In order to provide secure API's we will add oauth authentication and authorization.

Testing Plan

We will create Postman Collections to test the different scenarios.