Java Coding Challenge

Given the user story below:
As a librarian
I need a REST API to manage my book library and book rentals
So that I can replace my manual system.

- Implement the solution using Java and any libraries you need.
- Document your design decisions and assumptions.
- Use Maven as a build tool for this project.
- Send us your solution as a zipped archive.

Java Coding Challenge Solution

I tried to implement java coding challenge stated above. Here is the list of main technologies used:

- Java 11
- spring-boot
- spring-rest
- spring-data
- spring-cloud

Design Decisions & Assumptions

I used micro-service architectural approach to solve the challenge. For that purpose I created 3 functional micro-service and 1 architectural micro-service (config-server). 19 rest-services are implemented.

As of now, I left the coding for load-balancing, API gateway kind of infrastructural functionality unimplemented. I also didn't implement security as it may be implemented in a variety of ways.

I didn't use Lombok as well but could be used.
I implemented only very little of unit-test but it should be implemented as per organisation's testing policy.

Here is the list of micro-services.

NAME	PORT	DESCRIPTION

library-books-ms	8080	This micro-service is for crud operations for BOOK.
		These are operations on it:
		POST: /
		`{ "id": 1, "title":"Book1 Title1", "writer":"Book1 Writer1" }`
		PUT: /
		`{ "id": 1, "title":"Book1 Title2", "writer":"Book1 Writer2" }`
		PATCH: /{id}/writer/{writer}
		DELETE: /{id}
		DELETE: /logical/{id}
		GET: /
		GET: /{id}

library-users-ms	8081	This micro-service is for crud operations for USER.

<table>
<tr><td></td><td></td><td>These are operations on it:
POST: <code>/</code>
<pre>{
   "name":"Aydin",
   "surname":"Karaman",
   "birthdate": "1900-01-01"
}</pre>
PUT: <code>/</code>
<pre>{
   "name":"Aydin",
   "surname":"Karaman_CHANGE",
   "birthdate": "1980-01-01"
}</pre>
DELETE: <code>/{id}</code>
DELETE: <code>/logical/{id}</code>
GET: <code>/</code>
GET: <code>/{id}</code></td></tr>
<tr><td><code>library-book-rental-ms</code></td><td><code>8082</code></td><td>This micro-service is for all rental operations.
DESC: rentBook
POST: <code>/book/{bookId}/user/{userId}</code>
DESC: returnBook
PUT:<code>/book/{bookId}</code>
DESC: findRentalHistoryOfBook
GET: <code>/{id}</code>
DESC: findRentalHistoryOfUser
GET: <code>/logical/{id}</code>
GET: <code>/</code>
GET: <code>/{id}</code></td></tr>
<tr><td><code>library-config-server</code></td><td><code>8010</code></td><td>This is for central configuration management.

In this challenge there was no need for this micro-service but I just created it and configured it so that it can be used when needed.</td></tr>
</table>

Micro-service dependency should be as little as it can be.

Library-book-rental-ms is utilising other 2 micro-services but only in one functionality: rentBook.
All information that consists of a book-rental (including, book and user details) are persisted in this micro-service so that it can be viewed without the need of other micro-services.