**Documentația aplicației de Book Management**

**1. Prezentare generală**

Aplicația **Book Management** este concepută pentru a facilita gestionarea informațiilor despre cărți, autori, genuri literare, recenzii și utilizatori. Utilizatorii pot adăuga, vizualiza și șterge cărți, pot oferi recenzii, și pot crea liste personale de cărți. Aplicația este potrivită pentru biblioteci, librării sau pasionați de lectură care doresc să gestioneze eficient informațiile despre cărți.

**2. Tehnologii folosite**

* **Java 17**
* **Spring Boot 3.x** (Spring MVC, Spring Data JPA)
* **MySQL** pentru gestionarea bazei de date
* **Hibernate** pentru maparea obiect-relațională
* **Swagger 1.0** pentru documentarea API-urilor
* **Maven** pentru gestionarea dependințelor
* **JUnit 5** pentru teste unitare

**3. Cerințe funcționale**

1. Utilizatorii se pot inregistra in aplicatie.

2. Utilizatorii pot vizualiza detaliile si informatiile cartilor.

3. Utilizatorii pot vizualiza informatii despre autori.

3. Utilizatorii pot adauga carti la diferite liste personale (to-read, reading, completed).

4. Utilizatorii pot scrie recenzii pentru carti.

5. Utilizatorii pot evalua carti (de la 1 la 5).

6. Administratorii pot adauga si sterge carti.

7. Administratorii pot adauga si sterge autori.

8. Administratorii pot adauga genuri de carti.

9. Cartile pot fi asociate cu utilizatorii prin entitatea UserBook.

10. Cartile pot fi asociate cu unul sau mai multe genuri prin entitatea BookGenre.

**4. Cerințe nonfuncționale**

1. Performanță: Aplicația trebuie să răspundă la solicitări API în mai puțin de 1 secundă.

2. Scalabilitate: Structura aplicației permite adăugarea ulterioară de funcționalități, cum ar fi autentificarea.

3. Ușurință în utilizare: API-ul este documentat prin Swagger.

4. Persistență sigură a datelor folosind MySQL.

**5. Arhitectura aplicației**

Aplicația este organizată în module separate pentru a respecta principiul separării responsabilităților:

* **Controllers**: Răspund de manipularea cererilor HTTP.
* **Services**: Includ logica aplicației.
* **Repositories**: Gestionează interacțiunea cu baza de date.
* **Configurație Swagger**: Documentația API-ului este disponibilă la /swagger-ui.html, utilizând versiunea 1.0 a Swagger.
* **Testare**: Testele unitare sunt implementate pentru endpoint-uri și servicii folosind JUnit 5.

**6. Instrucțiuni de configurare**

Este nevoie de:

- Java 21 (sau mai recent).

- Maven (pentru gestionarea dependențelor).

- Baza de date MySQL.

**Pașii pentru rularea aplicației:**

1. Descarcă codul sursă sau clonează repository-ul.

2. Configurează conexiunea la baza de date în fișierul application.properties:

spring.datasource.url=jdbc:mysql://localhost:3306/book\_management

spring.datasource.username=root

spring.datasource.password=\*\*\*\*

3. Construiește proiectul folosind Maven:

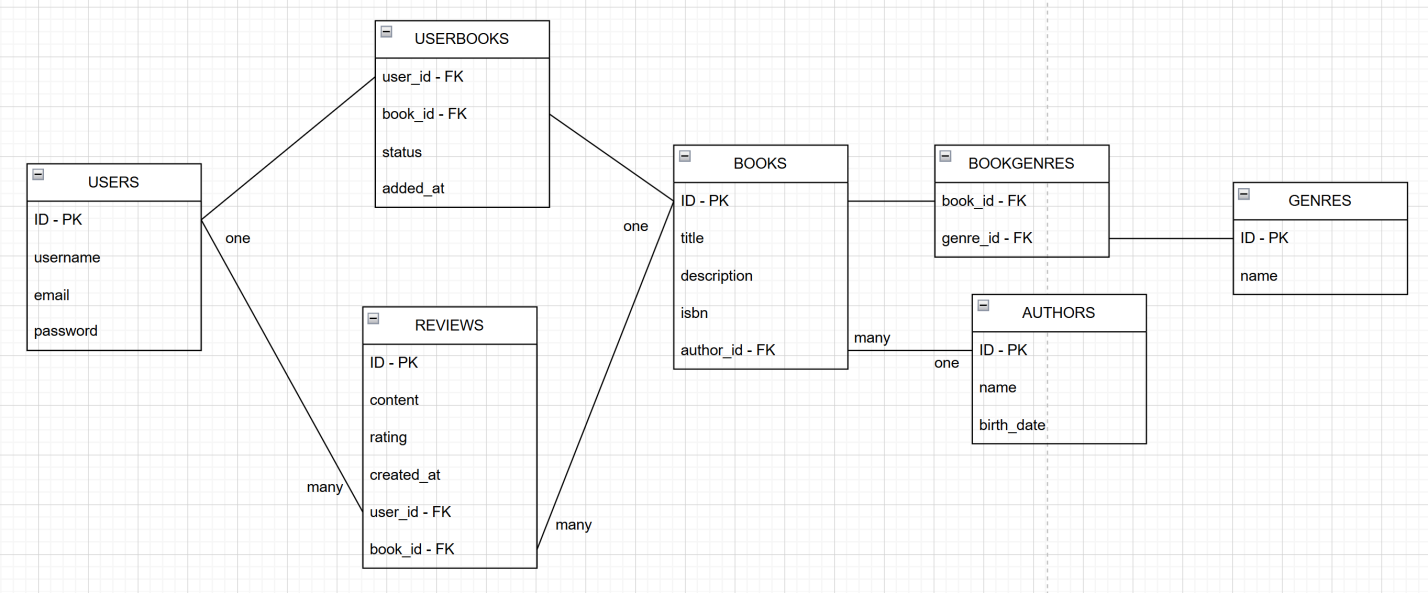
mvn clean install

4. Rulează aplicația:

mvn spring-boot:run

5. Accesează Swagger UI la <http://localhost:8080/swagger-ui.html>.

**7. Persistența datelor**



Aplicația folosește MySQL ca sistem de gestiune a bazei de date. Tabelele și relațiile dintre ele sunt:

* **USERS**: Se leagă cu tabelele REVIEWS (one-to-many) și USERBOOKS (one-to-many).
* **BOOKS**: Se leagă cu tabelele USERBOOKS (one-to-many), BOOKGENRES (one-to-many), REVIEWS (one-to-many) și AUTHORS (one-to-many).
* **GENRES**: Se leagă cu tabela BOOKGENRES (one-to-many).
* **AUTHORS**: Se leagă cu tabela BOOKS (one-to-many).
* **REVIEWS**: Legată de USERS prin relația one-to-many și de BOOKS prin relația one-to-many.
* **USERBOOKS**: Tabel asociativ pentru relația USERS-BOOKS.
* **BOOKGENRES**: Tabel asociativ pentru relația BOOKS-GENRE.

**8. Concluzii**

Aplicația Book Management oferă un cadru robust pentru gestionarea informațiilor despre cărți, autori, genuri literare, recenzii și utilizatori. Arhitectura modulară și integrarea cu Swagger fac ca aplicația să fie ușor de extins și de utilizat.