O imagine care conține text, Font, Grafică, siglă

Descriere generată automat

**Documentație Proiect - E-Biblio**

**Proiect realizat de**

**Mesaroș Antonio**

**Cuprins**

1. **Titlul proiectului**
2. **Scopul proiectului**
3. **Funcționalitățile proiectului**
   * **3.1 Funcționalități pentru utilizatori normali**
   * **3.2 Funcționalități pentru administratori**
4. **Tehnologii utilizate**
5. **Diagrame UML**
   * **5.1 Diagramă Use Case**
   * **5.2 Diagrama de activitate: Împrumutul unei cărți**
   * **5.3 Diagrama de activitate: Editarea unei cărți**
6. **Design Pattern utilizat**
7. **Link-ul către repository-ul GitHub**
8. **Structura aplicației**
9. **Testarea**

**10. Observații**

1. Titlul proiectului

**E-Biblio**: Sistem de gestionare a bibliotecii

1. **Scopul proiectului**

Proiectul *E-Biblio* este o aplicație web care permite utilizatorilor să interacționeze eficient cu o bibliotecă virtuală. Sistemul oferă funcționalități pentru utilizatori normali și administratori, precum împrumutul, rezervarea sau gestionarea cărților.

1. **Funcționalitățile proiectului**

* Funcționalități pentru utilizatori normali:
  + Vizualizarea listei de cărți disponibile.
  + Căutare avansată după titlu, autor, gen etc.
  + Vizualizarea detaliilor unei cărți.
  + Împrumutarea unei cărți (dacă este disponibilă).
  + Rezervarea unei cărți (în cazul stocului indisponibil).
  + Returnarea unei cărți împrumutate.
* Funcționalități pentru administratori:
  + Adăugarea unei cărți noi.
  + Editarea informațiilor despre o carte existentă.
  + Ștergerea unei cărți din sistem.
  + Suplimentarea stocului unei cărți.

Exemplu pentru funcționalitatea "Rezervarea unei cărți":

* Introducere  
  "Funcționalitatea de rezervare a unei cărți permite utilizatorilor să rezerve o carte pe o perioadă de o zi. Aceasta este utilă în cazul în care utilizatorul are nevoie urgentă de carte și dorește să fie sigur că o va putea obține în ziua următoare. Rezervarea este posibilă doar dacă cartea este disponibilă în stoc."
* Imagine
* Descrierea procesului
  1. Utilizatorul accesează pagina "Rezervă o carte" din meniul aplicației.
  2. Introduce în câmpul de căutare titlul sau o parte din titlul cărții pe care dorește să o rezerve.
  3. Apasă pe butonul "Caută" pentru a vizualiza lista cărților disponibile.
  4. Selectează cartea dorită din lista afișată.
  5. Apasă butonul "Rezervă" pentru a confirma rezervarea.
  6. Sistemul verifică dacă utilizatorul nu are deja o altă rezervare activă și dacă există exemplare disponibile ale cărții selectate.
  7. Dacă rezervarea este validă, apare un mesaj de succes care confirmă procesul. În caz de eroare (de exemplu, cartea nu este disponibilă sau utilizatorul are deja o rezervare activă), se afișează un mesaj corespunzător.

Exemplu pentru funcționalitatea "Adăugarea unei cărți noi":

* Introducere  
  "Funcționalitatea de adăugare a unei cărți noi permite administratorului să introducă o carte nouă în sistem. Aceasta include completarea informațiilor despre carte, cum ar fi titlul, autorul, descrierea, genul și stocul total. După salvare, cartea devine disponibilă pentru utilizatori."
* O imagine care conține text, captură de ecran, Font, număr

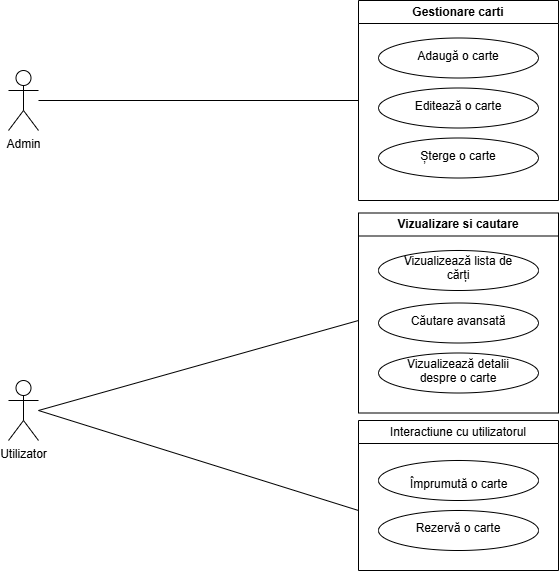
  Descriere generată automatImagine
* Descrierea procesului
  1. Administratorul accesează pagina de gestionare a cărților și selectează opțiunea "Adaugă o carte".
  2. Completează câmpurile necesare:
     + Titlu: Numele cărții.
     + Autor: Autorul cărții.
     + Descriere: Un rezumat al conținutului cărții.
     + Gen: Categoria literară a cărții (ex.: roman, poezie, științific).
     + Stoc total: Numărul total de exemplare disponibile.
  3. Apasă butonul "Adaugă".
  4. Dacă informațiile sunt corecte, apare un mesaj de succes care confirmă adăugarea cărții. Dacă există o eroare (de exemplu, cartea deja există), se afișează un mesaj corespunzător.

1. **Tehnologii utilizate**

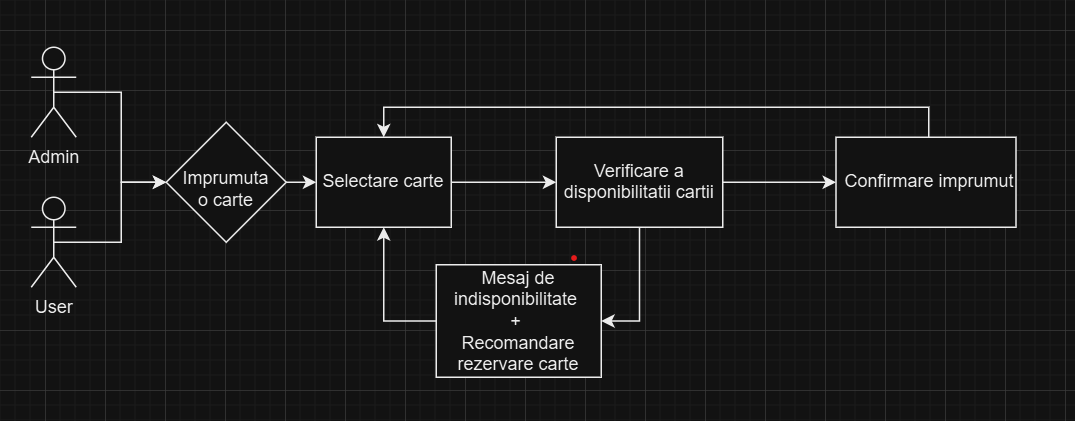
* **Frontend:**
  + Limbaj: *JavaScript*.
  + Framework: *React.js*.
  + Biblioteci suplimentare: *Bootstrap* pentru design și stilizare.
* **Backend:**
  + Limbaj: *Java*.
  + Framework: *Spring Boot*.
  + Arhitectură: *RESTful API*.
* **Bază de date:**
  + SGBD: *MySQL*.
  + Conexiune: *Hibernate JPA* pentru *ORM*.
* **Altele:**
  + *GitHub* pentru versionare și colaborare.
  + *Postman* pentru testarea endpoint-urilor API.
* **5. Diagrame UML**

**5. Diagrame**

**5.1. Diagramă de interacțiune**

* ****Descrie interacțiunea principală dintre utilizator și sistem.

**5.2. Diagrama de activitate: Împrumutul unei cărți**

* ****Explică pașii necesari pentru ca un utilizator să împrumute o carte.

**5.3. Diagrama de activitate: Editarea unei cărți**

* **O imagine care conține captură de ecran, diagramă, text, Font

  Descriere generată automat**Pașii necesari pentru administratori să editeze o carte existentă.

**6. Design Pattern utilizat**

* **Pattern-ul folosit**: Repository Pattern
* **Scenariu**: Acest pattern a fost implementat în backend pentru a gestiona accesul la baza de date prin intermediul claselor repository (ex. CarteRepository, UserRepository).
* **Cod relevant**: java

*@Repository*

*public interface UserRepository extends JpaRepository<User, Long> {  
 Optional<User> findByUsername(String username);  
}*

**Avantaje:**

* + Separarea logicii de acces la date de celelalte componente ale aplicației.
  + Creșterea modularității și testabilității aplicației.

**7. Link-ul către repository-ul GitHub**

Link către proiectul GitHub unde se poate verifica codul sursă, commit-urile și colaborarea: [GitHub Repository - E-Biblio](https://github.com/repo_example)

**8. Structura aplicației**

* **Frontend:** Directorul src conține componentele React corespunzătoare fiecărei funcționalități (e.g., EditeazaCarte.js, ImprumutaCarte.js).
* **Backend:** Structurat pe pachete: controller, service, repository și model.

**9. Testarea**

* **Frontend:**
  + Testarea interfeței utilizator prin interacțiunea cu butoanele și verificarea alertelor.
* **Backend:**
  + Testarea endpoint-urilor API folosind Postman.
  + Exemple:
    - **GET:** /api/carti/cautare?titlu=Test.
    - **POST:** /api/imprumuturi/creeaza/{userId}/{carteId}