

# **Documentatie Proiect Catalog Resurse Universitare**

**Student Marcu Stefan-Cristian**

**Grupa 332AB**

## **a. Functionalitati de baza si suplimentare**

### **Functionalitati de baza**

#### **1. Gestionarea cartilor:**

- Adaugarea unei carti noi.
- Modificarea informatiilor despre o carte (titlu, autor, categorie, an publicare).
- Stergerea unei carti existente.
- Listarea tuturor cartilor disponibile.

#### **2. Gestionarea utilizatorilor:**

- Adaugarea utilizatorilor cu rolul de citator, admini sunt deja existenti in baza de date.
- Validarea autentificarii utilizatorilor.

#### **3. Gestionarea cererilor de imprumut:**

- Trimiterea unei cereri de imprumut pentru o carte de catre utilizatori.
- Modificarea statusului unei cereri ("aprobat", "respins").
- Vizualizarea cererilor de imprumut in functie de status.

### **Functionalitati suplimentare**

#### **1. Interfata de administrare:**

- Dashboard pentru admin cu acces la gestionarea cartilor si cererilor.
- Vizualizarea si filtrarea cererilor de imprumut.

#### **2. Validari avansate:**

- Prevenirea duplicatelor in baza de date pentru carti si utilizatori.
- Verificarea unicitatii pentru username si parola.

### 3. Notificari pentru utilizatori:

- Afisarea mesajelor de confirmare si eroare in timpul interactiunilor cu aplicatia.

### 4. Functii de logout:

- O functie care permite utilizatorilor sa iasa din sesiune si sa fie redirectionati la pagina principala.

---

## b. Descrierea claselor/metodelor/atributelor proiectului

### 1. Clasa Carte

- **Descriere:** Reprezinta entitatea principala pentru gestionarea informatiilor despre carti.
- **Atribute:**
  - **id (Long):** Identificator unic al cartii.
  - **titlu (String):** Titlul cartii.
  - **autor (String):** Autorul cartii.
  - **categorie (String):** Categoria cartii.
  - **anPublicare (int):** Anul in care a fost publicata cartea.

### 2. Clasa Utilizator

- **Descriere:** Reprezinta entitatea care modeleaza utilizatorii aplicatiei.
- **Atribute:**
  - **id (Long):** Identificator unic al utilizatorului.
  - **username (String):** Numele utilizatorului (unic).
  - **password (String):** Parola utilizatorului.
  - **rol (String):** Rolul utilizatorului ("admin" sau "cititor").

### 3. Clasa CerereImprumut

- **Descriere:** Reprezinta entitatea pentru gestionarea cererilor de imprumut de carti.
- **Atribute:**
  - **id (Long):** Identificator unic al cererii.
  - **titluCarte (String):** Titlul cartii solicitate.
  - **usernameCititor (String):** Utilizatorul care a trimis cererea.

- **status (String):** Statusul cererii ("in asteptare", "aprobat", "respins").

#### **4. Clasa CarteController**

- **Descriere:** Este responsabil de:
  - gestionarea cererilor si raspunsurilor HTTP legate de entitatea carte,
  - transfera aceste responsabilitati catre CarteService .
- **Metode:**
  - **adaugaCarte:** Permite adaugarea unei carti.
  - **actualizeazaCarte:** Permite modificarea informatiilor unei carti existente.
  - **stergeCarte:** Permite stergerea unei carti.
  - **getCarti:** Returneaza lista cartilor disponibile.

#### **5. Clasa UtilizatorController**

- **Descriere:** Expune API-ul pentru gestionarea utilizatorilor cu aceste scopuri:
  - **Gestionarea autentificarii:** Verifica existenta unui utilizator si valideaza parola acestuia.
  - **Gestionarea inregistrarii:** Se asigura ca utilizatorul nu exista deja si permite crearea unui nou utilizator.
- **Metode:**
  - **login:** Permite autentificarea utilizatorilor.
  - **signup:** Permite inregistrarea unui utilizator nou.

#### **6. Clasa CerereImprumutController**

- **Descriere:** Expune API-ul pentru gestionarea cererilor de imprumut cu scopul:
  - **Gestionarea cererilor noi:** Creaza cereri cu un status initial predefinit.
  - **Filtrarea cererilor:** Returneaza doar cererile cu un anumit status.
  - **Asigurarea statusului:** Asigura validarea statusului inainte de validare.
- **Metode:**
  - **adaugaCerere:** Permite adaugarea unei cereri de imprumut.
  - **modificaStatusCerere:** Permite modificarea statusului unei cereri.
  - **getCereri:** Returneaza lista tuturor cererilor de imprumut.

#### **7. Clasa CarteService**

- **Descriere:** Este o clasa ce se ocupa cu interactiunea cu repository-ul responsabil de tabela CARTE.
- **Metode:**
  - **adaugaCarte:** Valideaza si salveaza o carte in baza de date.
  - **stergeCarte:** Elimina o carte pe baza ID-ului.
  - **getToateCartile:** Returneaza toate cartile din baza de date.

#### 8. Clasa UtilizatorService

- **Descriere:** Este o clasa ce se ocupa cu interactiunea cu repository-ul responsabil de tabela UTILIZATOR.
- **Metode:**
  - **adaugaUtilizator:** Salveaza un utilizator nou in baza de date.
  - **validareLogin:** Verifica autentificarea unui utilizator.
  - **existsByUsername:** Verifica unicitatea username-ului.

#### 9. Clasa CerereImprumutService

- **Descriere:** Este o clasa ce se ocupa cu interactiunea cu repository-ul responsabil de tabela CERERE\_IMPRUMUT.
- **Metode:**
  - **adaugaCerere:** Salveaza o noua cerere de imprumut.
  - **modificaStatus:** Actualizeaza statusul unei cereri existente.
  - **getToateCereri:** Returneaza toate cererile din baza de date.

#### 10. Clasa CarteRepository

- **Descriere:** Interactiuneaza cu baza de date pentru entitatea Carte( tabelul CARTE).
- **Metode:**
  - **existsByTitluAndAutor:** Verifica daca o carte exista pe baza titlului si a autorului.
  - **deleteByTitluAndAutor:** Sterge o carte dupa titlu si autor.

#### 11. Clasa UtilizatorRepository

- **Descriere:** Interactiuneaza cu baza de date pentru entitatea Utilizator( tabelul UTILIZATOR).
- **Metode:**

- **findByUsernameAndPassword:** Cauta un utilizator dupa numele contului (username) si parola.
- **existsByUsername:** Verifica daca exista un utilizator cu un anumit username.
- **findByUsername:** Gaseste un utilizator dupa username.
- **deleteByUsername:** Sterge un utilizator dupa username.

## 12. Clasa CerereImprumutRepository

- **Descriere:** Interactioneaza cu baza de date pentru entitatea CerereImprumut( tabelul CERERE\_IMPRUMUT).
- **Metode:**
  - **findByStatus:** Gaseste cererile de imprumut dupa status.
  - **existsByUsernameCititorAndTitluCarte:** Verifica daca exista o cerere de imprumut in functie de numele cititorului si titlul cartii.
  - **deleteByUsernameCititorAndTitluCarte:** sterge o cerere.
  - **findByUsernameCititor:** Gaseste cererile de imprumut dupa username.

## c. Descrierea elementelor functionale oferite prin interfata grafica cu utilizatorul

### 1. Dashboard Admin:

- **Formular pentru adaugarea unei carti:**
  - **Campuri:** titlu, autor, categorie, an publicare.
  - **Buton:** "Adauga Carte".
- **Lista de carti:**
  - Fiecare carte are butoane pentru modificare ("Modifica") si stergere ("Sterge").

### 2. Gestionarea cererilor de imprumut:

- **Lista de cereri afisate cu utilizatorul, cartea solicitata si statusul.**
- **Butoane pentru actualizarea statusului ("Aprobat", "Respins").**

### 3. Logout:

- **Buton dedicat pentru iesirea din sesiune.**

## **d. Aspecte privind testarea aplicatiei**

### **1. Testarea unitatii:**

- Clasele de testare pentru `CarteService`, `UtilizatorService` si `CerereImprumutService`, etc. pentru a valida functionalitatile individuale.

### **2. Testarea integrarii:**

- Validarea comunicarii intre Controller si Service folosind mock-uri.

### **3. Testarea interfetei grafice:**

- Teste manuale pentru verificarea functionalitatilor din paginile HTML .

### **4. Testarea bazei de date:**

- Verifica unicitatea constrangerilor si integritatea datelor dupa operatii CRUD.
- 

## **e. 5 idei despre cum poate fi imbunatatita aplicatia:**

### **1. Autentificare avansata:**

- Implementarea unui sistem de autentificare cu parola encriptata.

### **2. Cautare si filtrare:**

- Adaugarea unui sistem de cautare si filtrare in functie de autor, categorie sau an publicare.

### **3. Notificari prin email:**

- Trimiterea notificarilor prin email catre utilizatori pentru cererile de imprumut aprobate sau respinse.

### **4. Posibilitate de a semnala eventuale probleme:**

- In interfata cu utilizatorul se poate adauga un buton de “semnaleaza problema”, care sa deschida un formular prin care poate fi semnalat admin-ul in legatura cu problema intampinata de user.

### **5. Statistici:**

- Adaugarea unui modul de statistici pentru admin, cum ar fi numarul de carti imprumutate sau statusul cererilor.