



Universitatea din București
Facultatea de matematică-informatică

Proiect bibliotecă

- Programare web utilizând tehnologii Java -

Student:
Cătănoiu Gabriela-Anamaria 405

Cuprins

1. Introducere
2. Business requirements
3. 5 business requirements explicate
4. Endpoints

Introducere

Proiectul constă în implementarea unei baze de date care să gestioneze: cărțile care se află într-o bibliotecă, lista de cărți împrumutate și abonații bibliotecii. Pe parcursul desfășurării proiectului îmi propun:

- să realizez o organizare cât mai compactă a datelor, astfel încât bibliotecarul să poată obține ușor date despre exemplarele care se regăsesc în bibliotecă
- să ofer abonaților posibilitatea de a vedea lista cu cărțile împrumutate de-a lungul timpului
- să ofere posibilitatea abonaților de a căuta în bibliotecă toate informațiile despre o anumită carte și autorul acesteia
- să pastrez o evidență clară a tuturor abonaților care împrumută cărți

Business requirements

1. În calitate de proprietar de produs, pot să comunic abonaților informații referitoare la cărțile împrumutate din bibliotecă
2. În calitate de proprietar de produs, voi oferi abonaților toate informațiile de care au nevoie referitoare la o carte
3. În calitate de proprietar de produs, vreau să pot contacta clientul folosind datele sale personale
4. În calitate de proprietar de produs, voi putea reorganiza date despre cărți în funcție de mai multe criterii
5. În calitate de proprietar de produs, voi achiziționa noi cărți în bibliotecă pentru a avea o gamă mai largă de clienți
6. În calitate de proprietar de produs, voi oferi o actualizare a împrumuturilor în funcție de cererea abonatului
7. În calitate de proprietar de produs, voi oferi posibilitatea abonaților de a lua legătura cu biblioteca
8. În calitate de proprietar de produs, ma voi asigura ca datele personale ale clientului sunt confidențiale
9. În calitate de proprietar de produs, pot înștiința abonatul despre cărțile care nu mai există, datorită cererii foarte mici de împrumut
10. În calitate de proprietar de produs, voi permite abonaților să nu mai facă parte din clienții bibliotecii

5 business requirements explicate

1. În calitate de proprietar de produs, pot să comunic abonaților informații referitoare la cărțile împrumutate din bibliotecă

În cod se vor putea selecta date referitoare la una, doua sau toate cărțile pe care clientul le-a împrumutat de-a lungul timpului. Va putea vedea titlul cărții, data împrumutului și returului și îi vor fi comunicate informații despre ștergerea unei cărți care nu mai este în bibliotecă.

2. În calitate de proprietar de produs, voi oferi abonaților toate informațiile de care au nevoie referitoare la o carte

În cod voi putea să selectez datele despre o anumită carte, sau să selectez anumite informații referitoare la autor sau publicație. De asemenea va fi la curent cu noile informații adăugate despre o nouă carte sau dacă cartea nu mai există în biblioteca mea.

3. În calitate de proprietar de produs, vreau să pot contacta clientul folosind datele sale personale

În cazul în care termenul de returnare al unui produs a fost depășit, vreau să am posibilitatea de a lua legătura cu clientul prin utilizarea datelor sale personale. Voi putea să selectez numărul de telefon din baza de date pentru a putea fi contactat. În cazul unei modificări a numărului de telefon, aş vrea să pot face selecția după adresa de email. De asemenea se vor putea actualiza datele de contact ale unui utilizator în cazul schimbărilor.

4. În calitate de proprietar de produs, voi putea reorganiza date despre cărți în funcție de mai multe criterii

Voi actualiza date în funcție de următoarele criterii: autor, titlul cărților, publicație, categorie și data lansării. Astfel biblioteca va fi mereu la zi cu toate detaliile despre o anumită carte. Și astfel putem să eliminăm și erorile de sistem.

5. În calitate de proprietar de produs, voi achiziționa noi cărți în bibliotecă pentru a avea o gamă mai largă de clienți

Voi separa datele în funcție de importanță. Voi adauga date despre autor, titlul cărților, publicație, categorie și data lansării astfel încât abonatul să primească toate informațiile de care are nevoie pentru o anumită carte.

Endpoints

Member

- `ResponseEntity<MemberDTO>` **addMember** -> Adaugă noi informații despre un membru în bibliotecă
- `public ResponseEntity<List<Member>>` **getMembers()** -> Afișază o listă cu toți membrii din bibliotecă
- `public ResponseEntity<List<MemberDTO>>` **getMemberByEmail** -> Caută informații despre un membru după email-ul acestuia
- `public ResponseEntity<Optional<Member>>` **getMember** -> Caută informații despre un membru după id-ul acestuia
- `public ResponseEntity<String>` **deleteMember** -> Eliminarea unui membru care nu a mai împrumutat cărți de o perioadă mare de timp
- `public ResponseEntity<MemberDTO>` **updateEmailMember** -> Actualizează email-ul unui utilizator

Author

- `public ResponseEntity<AuthorDTO>` **addAuthor** -> Adaugă noi informații despre autorul unei cărți
- `public ResponseEntity<Optional<Author>>` **getOneAuthor** -> Caută informații despre un autor după id-ul acestuia
- `public ResponseEntity<List<AuthorDTO>>` **getByNameAndSurname** -> Caută informații despre un autor după numele și prenumele acestuia

- public ResponseEntity<List<Author>> **getAllAuthors()** -> Afișază o listă cu toți autorii din bibliotecă
- public ResponseEntity<String> **deleteAuthor** -> Eliminarea unui autor din bibliotecă

City

- public ResponseEntity<CityDTO> **addCity** -> Adaugă noi informații despre orașul unde se află o anumită editură
- public ResponseEntity<Optional<City>> **getOneCity** -> Caută informații despre orașul unde se află editura după id-ul acestuia
- public ResponseEntity<List<CityDTO>> **getByName** -> Caută informații despre orașul unde se află editura după nume
- public ResponseEntity<String> **deleteCity** -> Șterge un oraș în care publicația nu mai există

Publishing House

- public ResponseEntity<PublishingHouseDTO> **addPublishHouse** -> Adaugă noi informații despre editura unei anumite cărți
- public ResponseEntity<List<PublishingHouse>> **getAllPublishHouse()** -> Afișază o listă a tuturor editurilor care colaborează cu biblioteca
- public ResponseEntity<Optional<PublishingHouse>> **getPublishHouseById** -> Caută informații despre o editură după id-ul acesteia
- public ResponseEntity<List<PublishingHouseDTO>> **getPublishHouseByName** -> Caută informații despre o editură după numele acesteia
- public ResponseEntity<String> **deletePublishHouse** -> Șterge o publicație care nu mai există

Literature category

- public ResponseEntity<CategoryDTO> **addNewCategory** -> Adaugă o nouă categorie de literature
- public ResponseEntity<Optional<Category>> **getOneCategory** -> Caută informații despre categoria unei cărți după id-ul acesteia
- private ResponseEntity<List<CategoryDTO>> **getCategoryByName** -> Caută informații despre categoria unei cărți după numele acesteia

Book

- public ResponseEntity<BookDTO> **addBooks** -> Adaugă informații despre o nouă carte achiziționată de bibliotecă
- public ResponseEntity<List<Book>> **getAllBooks()** -> Afișază o listă cu toate cărțile disponibile în bibliotecă
- public ResponseEntity<Optional<Book>> **getBookById** -> Caută informații despre o carte după id-ul acesteia
- public ResponseEntity<List<BookDTO>> **getBookByTitle** -> Caută informații despre o carte după titlu
- public ResponseEntity<String> **deleteBookById** -> Șterge o carte care nu mai există în bibliotecă
- public ResponseEntity<BookDTO> **updateBooks** -> Actualizează informații despre o anumită carte

Borrowed Books

- public ResponseEntity<List<BorrowedBooks>> **getAllBorrowedBooks()** -> Afișază toate împrumuturile care s-au efectuat până în momentul actual

- `public ResponseEntity<Optional<BorrowedBooks>> getOneBorrowedBook ->`
Caută informații despre o carte care a fost împrumutată după id-ul acesteia
- `public ResponseEntity<List<BorrowedBooksDTO>> getAllBorrowedBookByDate`
-> Caută informații despre o carte care a fost împrumutată după data împrumutului
- `public ResponseEntity<BorrowedBooksDTO> addBorrowedBook ->` Adaugă informații despre cărțile care au fost împrumutate
- `public ResponseEntity<String> deleteBorrowedBook ->` Șterge anumite împrumuturi
- `public ResponseEntity<BorrowedBooksDTO> updateBorrowedBook ->`
Actualizează despre împrumutul unei anumite cărți după o anumită dată