

Universitatea din Bucureşti

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: cărţile care se află într-o bibliotecă, lista de cărţi împrumutate şi abonaţii bibliotecii. Pe parcursul desfăşurării proiectului îmi propun:

* să realizez o organizare cât mai compactă a datelor, astfel încât bibliotecarul să poată obţine uşor date despre exemplarele care se regăsesc în bibliotecă
* să ofer abonaţilor posibilitatea de a vedea lista cu cărţile împrumutate de-a lungul timpului
* să ofere posibilitatea abonaţilor de a căuta în bibliotecă toate informaţiile despre o anumită carte şi autorul acesteia
* să pastrez o evidenţă clară a tuturor abonaţilor care împrumută cărţi

Business requirements

1. În calitate de proprietar de produs, pot să comunic abonaţilor informaţii referitoare la cărţile împrumutate din bibliotecă
2. În calitate de proprietar de produs, voi oferi abonaţilor toate informaţiile 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 cărţi în funcţie de mai multe criterii
5. În calitate de proprietar de produs, voi achiziţiona noi carţi în bibliotecă pentru a avea o gamă mai largă de clienţi
6. În calitate de proprietar de produs, voi oferi o actualizare a împrumuturilor în funcţie de cererea abonatului
7. În calitate de proprietar de produs, voi oferi posibilitatea abonaţilor de a lua legătura cu biblioteca
8. În calitate de proprietar de produs, ma voi asigura ca datele personale ale clientului sunt confidenţiale
9. În calitate de proprietar de produs, pot înştiinţa abonatul despre carţile care` nu mai există, datorită cererii foarte mici de împrumut
10. În calitate de proprietar de produs, voi permite abonaţiilor să nu mai facă parte din clienţii bibliotecii





5 business requirements explicate

1. În calitate de proprietar de produs, pot să comunic abonaţilor informaţii referitoare la cărţile împrumutate din bibliotecă

În cod se vor putea selecta date referitoare la una, doua sau toate cărţile pe care clientul le-a imprumutat de-a lungul timpului. Va putea vedea titlul cărţii, data împrumutului şi returului şi îi vor fi comuncate informaţii despre ştergerea unei cărţi care nu mai este în bibliotecă.

1. În calitate de proprietar de produs, voi oferi abonaţilor toate informaţiile de care au nevoie referitoare la o carte

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

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

În cazul in care termenul de returnare al unui produs a fost depăşit, vreau să am posibilitatea de a lua legatura 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 selecţia după adresa de email. De asemenea se vor putea actualize datele de contact ale unui utilizator în cazul schimbărilor.

1. În calitate de proprietar de produs, voi putea reorganiza date despre cărţi în funcţie de mai multe criterii

Voi actualiza date în funcţie de următoarele criterii: autor, titlul cărţilor, publicaţie, categorie şi data lansării. Astfel biblioteca va fi mereu la zi cu toate detaliile despre o anumită carte. Şi astfel putem să eliminăm si erorile de sistem.

1. În calitate de proprietar de produs, voi achiziţiona noi carţi în bibliotecă pentru a avea o gamă mai largă de clienţi





Voi separa datele în funcţie de importanţă. Voi adauga date despre autor, titlul cărţilor, publicaţie, categorie şi data lansării atfel încât abonatul să primească toate informaţiile de care are nevoie pentru o anumită carte.

Endpoints

Member

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

Author

* public ResponseEntity<AuthorDTO> addAuthor -> Adaugă noi informaţii despre autorul unei cărţi
* public ResponseEntity<Optional<Author>> getOneAuthor -> Caută informaţii despre un autor după id-ul acestuia
* public ResponseEntity<List<AuthorDTO>> getByNameAndSurname -> Caută informaţii despre un autor după numele şi prenumele acestuia







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

City

* public ResponseEntity<CityDTO> addCity -> Adaugă noi informaţi despre oraşul unde se află o anumită editură
* public ResponseEntity<Optional<City>> getOneCity -> Caută informaţii despre oraşul unde se alfă editura după id-ul acestuia
* public ResponseEntity<List<CityDTO>> getByName -> Caută informaţii despre oraşul unde se alfă editura după nume
* public ResponseEntity<String> deleteCity -> Şterge un oraş în care publicaţia nu mai există

**Publishing House**

* public ResponseEntity<PublishingHouseDTO> addPublishHouse-> Adaugă noi informaţi despre editura unei anumite cărţi
* public ResponseEntity<List<PublishingHouse>> getAllPublishHouse() -> Afişază o listă a tuturor editurilor care colaborează cu biblioteca
* public ResponseEntity<Optional<PublishingHouse>> getPublishHouseById-> Caută informaţii despre o editură după id-ul acesteia
* public ResponseEntity<List<PublishingHouseDTO>>getPublishHouseByName

-> Caută informaţii despre o editură după numele acesteia

* public ResponseEntity<String> deletePublishHouse -> Şterge o publicaţie care nu mai există





Literature category

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

**Book**

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

**Borrowed Books**

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





* public ResponseEntity<Optional<BorrowedBooks>> getOneBorrowedBook -> Caută informaţii despre o carte care a fost împrumutată după id-ul acesteia
* public ResponseEntity<List<BorrowedBooksDTO>> getAllBorrowedBookByDate

-> Caută informaţii despre o carte care a fost împrumutată după data împrumutului

* public ResponseEntity<BorrowedBooksDTO> addBorrowedBook -> Adaugă informaţii despre carţile care au fost împrumutate
* public ResponseEntity<String> deleteBorrowedBook -> Şterge anumite împrumuturi
* public ResponseEntity<BorrowedBooksDTO> updateBorrowedBook -> Actualizează despre împrumutul unei anumite cărţi după oanumita dată

