|  |
| --- |
| Bookstore project In java programming language  Student name: Aydin Najdat Abduljabbar |



Universitatea din Bucureşti

Facultatea de matematică-informatică

Introduce

The project consists of implementing a database to manage: The books that are in a library, the list of category books and the author of the book. During the project, I intend to:

- to achieve a more compact organization of data, so that the bookstore can easily obtain data about the copies that are found in the bookstore

-to give subscribers the opportunity to see the list of books in the bookstore

-to provide information the opportunity to search in the bookstore for all information about a particular book and its author.

Business requirements

1.as a product owner, I can communicate to categories informations about new books from the bookstore

In the code you can select data relating to one, two or all the cards that the books has . He will be able to see the title of the book.

2.as a product owner, I will provide all the information they need about a book

In the code I will be able to select the data about a particular book, or select certain information about the author or publication. He will also be up to date with the new information added about a new book or if the book no longer exists in my bookstore.

3.as a product owner, I want to be able to add the new books to bookstore

4.as a product owner, I will be able to reorganize data about books according to several criteria

I will update data according to the following criteria: Author, book title, publication, category and release date. This way the bookstore will always be up to date with all the details of a particular book. We can also eliminate system errors.

1. as a product owner, I will offer new books in the bookstore.

Endpoints

1. Author

public ResponseEntity<AuthorResponseDto> addAuthor

public ResponseEntity<AuthorResponseDto> getAuthor

public ResponseEntity<AuthorResponseDto> deleteAuthor

private ResponseEntity<AuthorResponseDto> editAuthor

public ResponseEntity<List<AuthorResponseDto>> getAuthors()

1. Book

public ResponseEntity<BookResponseDto> addBook

public ResponseEntity<BookResponseDto> getBook

public ResponseEntity<BookResponseDto> deleteBook

public ResponseEntity<BookResponseDto> editBook

public ResponseEntity<List<BookResponseDto>> getBooks()

1. Category

public ResponseEntity<CategoryResponseDto> addCategory

public ResponseEntity<CategoryResponseDto> getCategory

public ResponseEntity<CategoryResponseDto> deleteCategory

public ResponseEntity<CategoryResponseDto> editCategory

public ResponseEntity<List<CategoryResponseDto>> getCategories()

1. City

public ResponseEntity<City> addCity

public ResponseEntity<City> getCityById

public ResponseEntity<City> deleteCity

public ResponseEntity<City> editCity

public ResponseEntity<List<City>> getCities()

1. Zipcode

public ResponseEntity<Zipcode> addZipcode

public ResponseEntity<Zipcode> getZipcode

public ResponseEntity<Zipcode> deleteZipcode

public ResponseEntity<Zipcode> editZipcode

public ResponseEntity<List<Zipcode>> getZipcodes()