API Description

2.1 Web Advanced

Sandra Krevová 516044
DHI1V.So

Table of Contents

1. Class diagram

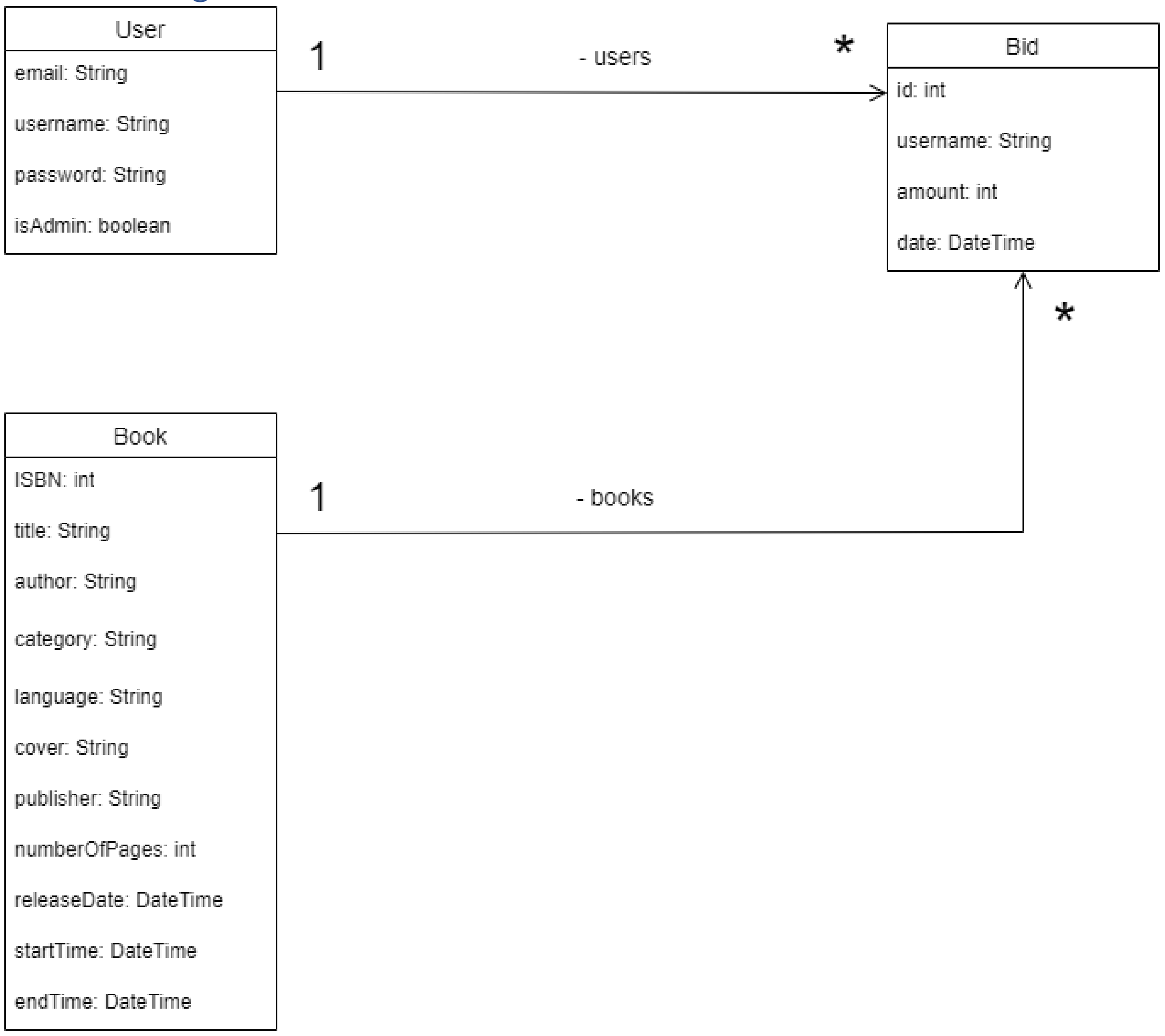

Figure 1 The relationships indicate that a user can bid on multiple books.

2. GET requests

GET	**/api/v1/books**		
Gets all books. Can be filtered on category, language, and cover.			
Parameters:	**Name**	**Type**	**Description**
** required*	category	query	Filter on category (String).
	cover	query	Filter on cover (String).
	language	query	Filter on language (String).
Responses:	**Code**	**Description / example if successful**	
	200	List of books. Can be empty.	

GET	**/api/v1/books/{isbn}**		
Gets specific book based on given isbn.			
Parameters:	**Name**	**Type**	**Description**
** required*	isbn *	path	ISBN of a book to find.
Responses:	**Code**	**Description / example if successful**	
	200	One book with provided ISBN.	
	404	Thrown when there is no book with the provided ISBN.	

GET	**/api/v1/books/{isbn}/bids**		
Gets all the bids for one book with the provided ISBN.			
Parameters:	**Name**	**Type**	**Description**
** required*	isbn *	path	ISBN of a book to find the bids for.
Responses:	**Code**	**Description / example if successful**	
	200	List of bids for a specific book. Can be empty.	
	404	Thrown when there is no book with the provided ISBN.	

GET	**/api/v1/users/**
Gets all the users.	

Responses:	**Code**	**Description / example if successful**
	200	List of users. Can be empty.
	401	Not logged in or invalid token.
	403	Not admin.

GET	**/api/v1/users/me/won**
Gets all the auctions the logged in user won.	

Responses:	**Code**	**Description / example if successful**
	200	List of books. Can be empty.
	401	Not logged in or invalid token.

3. POST requests

POST	**/api/v1/books**		
Adds a book.			
Parameters:	**Name**	**Type**	**Description**
* required	book *	body	The book to add. Example of JSON body: { "isbn": 9780451203581, "title": "To Kill a Mockingbird", "author": "Harper Lee", "category": "Classic Literature", "language": "English", "cover": "Hardcover", "publisher": "J.D. Lippincott", "numberOfPages": 196, "releaseDate": "1960", "startTime": "2023-10-05T10:30:00", "endTime": "2023-12-06T10:30:00", "images": ["https://upload.wikimedia.org/wikipedia/commons /4/4f/To_Kill_a_Mockingbird_%28first_edition_cover%29.jpg", "https://i0.wp.com/www.raptisrarebooks.com/wp-content/uploads/2017/08/mockingbird-e1504020435651.jpg?ssl=1"] }
Responses:	**Code**	**Description / example if successful**	
	201	If created successfully. Body will contain the newly created book. Either plain or JSON, based on whatever form was used in the request.	
	400	When data cannot be parsed correctly.	
	401	Not logged in or invalid token.	
	403	Not admin.	

POST	**/api/v1/books/{isbn}/bids**		
Adds a bid.			
Parameters:	**Name**	**Type**	**Description**
** required*	isbn *	path	ISBN of the book for what the bid is created.

** required*	bid *	body	The bid to add.
			Example of JSON body:
			{
			    "id": 1,
			    "username": "usertest",
			    "amount": 30,
			    "date": "01.11.2023 07:51"
			}

Responses:	**Code**	**Description / example if successful**	
	201	If created successfully. Body will contain the newly created bid.	
	400	When data cannot be parsed correctly.	
	401	Not logged in or invalid token.	

POST	**/api/v1/users**		
Adds a user.			
Parameters:	**Name**	**Type**	**Description**

** required*	user *	body	The user to add.
			Example of JSON body:
			{
			  "id": 2,
			  "username": "admintest",
			  "email": "admintest@example.com",
			  "password": "$2b$12$R2f8D/LJZ0jGGNTIrDi7Qepnp
			   fYifAl4S56vjr2NkZdcyeu6hT29a",
			  "isAdmin": true
			 }

Responses:	**Code**	**Description / example if successful**	
	201	If created successfully. Body will contain the newly created user.	
	400	When data cannot be parsed correctly.	
	401	Not logged in or invalid token.	
	403	Not admin.	

POST	**/api/v1/tokens**		
Adds a token.			
Parameters:	**Name**	**Type**	**Description**
** required*	password *	body	User password
	email *	body	User email
Responses:	**Code**	**Description / example if successful**	
	201	If created successfully.	

	400	When data cannot be parsed correctly.
	401	User not found / invalid password

4. PUT requests

PUT	/api/v1/books/{isbn}		
Modifies a book, based on path isbn.			
Parameters:	**Name**	**Type**	**Description**
** required*	isbn *	path	The isbn of the book to be modified
	book *	body	The book to modify. The isbn in the body must match with the path isbn
Responses:	**Code**	**Description / example if successful**	
	201	If successful. Body contains the newly modified book.	
	400	If body isbn does not match path isbn	
	401	Not logged in or invalid token.	
	403	Not admin.	

5. DELETE requests

DELETE	/api/v1/books/{isbn}		
Deletes specific book based on provided isbn			
Parameters:	**Name**	**Type**	**Description**
** required*	isbn *	path	isbn of book to delete
Responses:	**Code**	**Description / example if successful**	
	200	When a book has been successfully deleted	
	404	Thrown when there is no book with the provided isbn.	
	401	Not logged in or invalid token.	
	403	Not admin.	

DELETE	/api/v1/users/{id}		
Deletes specific user based on provided id			
Parameters:	**Name**	**Type**	**Description**
** required*	id *	path	id of user to delete
Responses:	**Code**	**Description / example if successful**	
	200	When a user has been successfully deleted	
	404	Thrown when there is no user with the provided id.	
	401	Not logged in or invalid token.	
	403	Not admin.	