Software Engineering 5324 OL1 Group Project Review

Members:

Preston Kelly Subash Kharel Scott Willis

What was the Group Project

- What is the goal?
  - A team of 2-3 members develop a web application using a database
    - For example: eShop, reservation system, rental cars, etc.
  - The teams must use Java, git, Maven, and a Test Framework.
    - Each team will also need to establish a git repository and a form of issue or ticket tracking such as GitHub
  - The data model for the project must have at least 7 entitites
    - At least 1 one-to-many and 1 many-to-many entities

BearBooks

- What is it?
  - A web application that would allow for a staff to monitor and update a database that contains all the books owned by the buisness
- What was our Goal?
  - A working web application prototype that would allow for unique users to register, log in, log out, checkout, update, add, and delete books form the database.

What will we Cover?

- The Design
  - Both Front End and Backend
- The Case Study
  - What Use Cases did we test?
- Conclusion
  - Our thoughts and Opinions
- Demonstration
- References

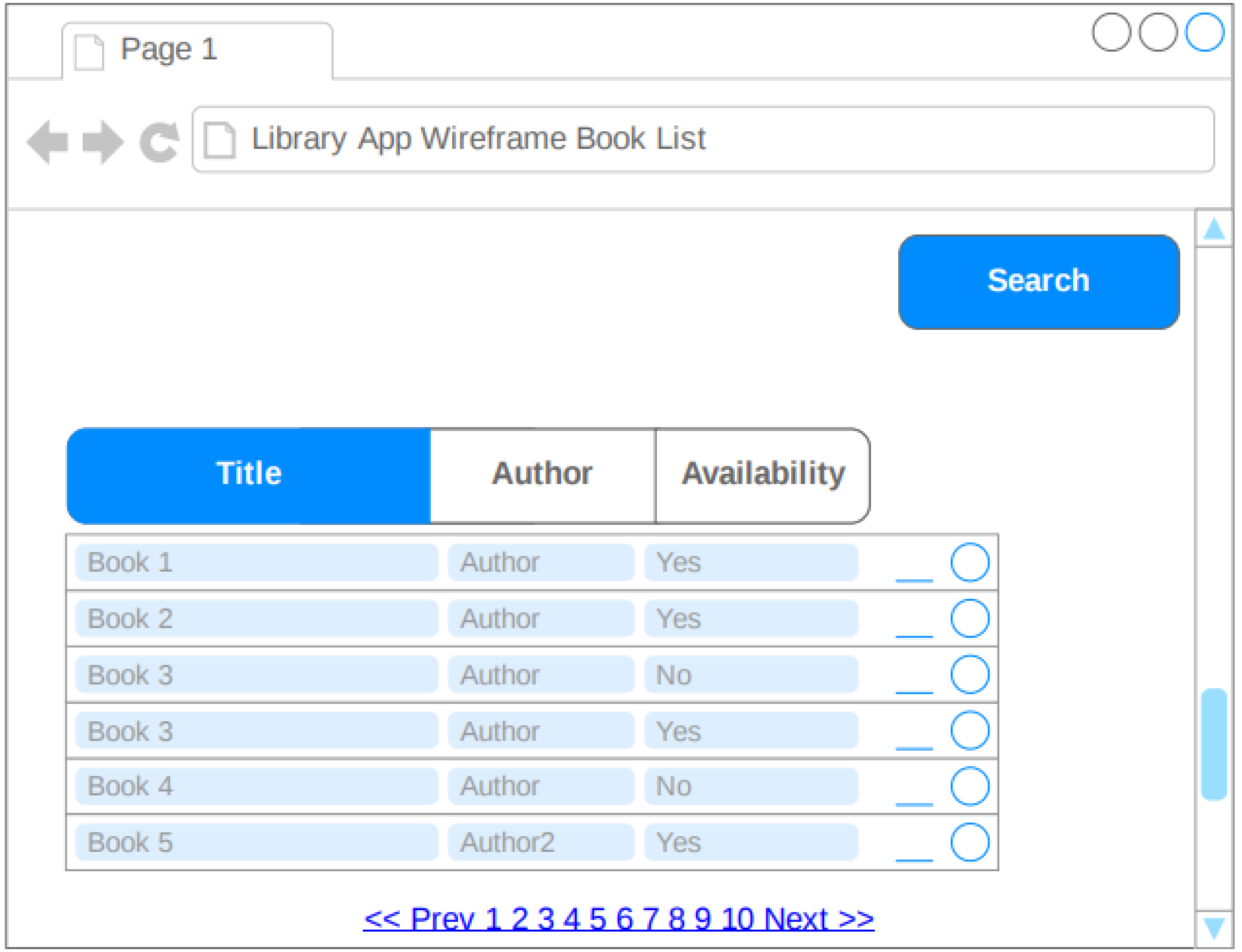
Design

- Front End
  - Built Upon the Angular Framework
    - Allowed for a quick and efficient frontend to be developed to allow for sufficient time to integrate the Back end of this project

- Back End
  - We have a fully functioning Create, Read, Update, Delete (CRUD) function service
  - All Data Model have validation Checks
  - Swagger integration
  - Currently running with a in-memory database (H2)

Front End Concept to …

The wireframe design was create to allow for an easy to create, more importantly an easily user friendly design.

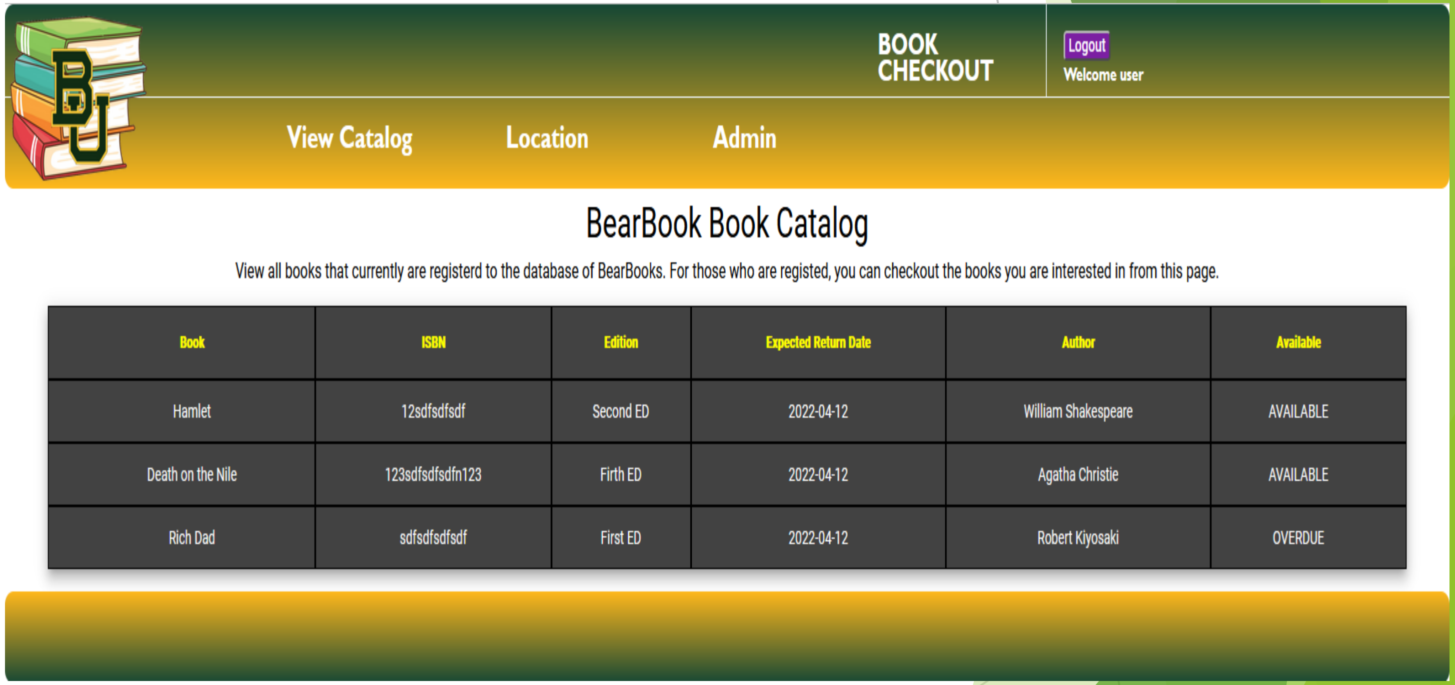


Title	Author	Availability	
Book 1	Author	Yes	__ ○
Book 2	Author	Yes	__ ○
Book 3	Author	No	__ ○
Book 3	Author	Yes	__ ○
Book 4	Author	No	__ ○
Book 5	Author2	Yes	__ ○

<< Prev 1 2 3 4 5 6 7 8 9 10 Next >>

… To Front End Reality

Ultimately, once we finished the front end design and integrated the backend functionality, we had a user friendly application.



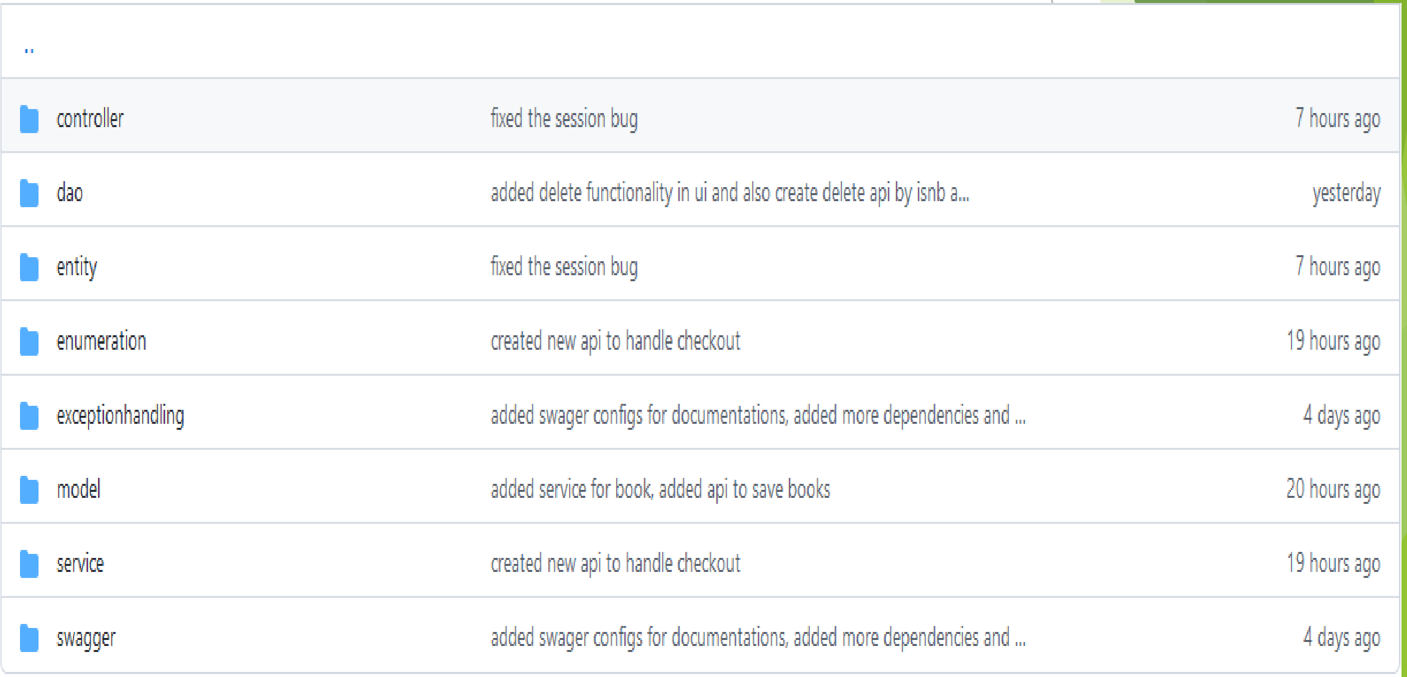
BearBook Book Catalog

View all books that currently are registerd to the database of BearBooks. For those who are registed, you can checkout the books you are interested in from this page.

Book	ISBN	Edition	Expected Return Date	Author	Available
Hamlet	12sdfsdfsdf	Second ED	2022-04-12	William Shakespeare	AVAILABLE
Death on the Nile	123sdfsdfsdfn123	Firth ED	2022-04-12	Agatha Christie	AVAILABLE
Rich Dad	sdfsdfsdfsdf	First ED	2022-04-12	Robert Kiyosaki	OVERDUE

The Back End Basics …

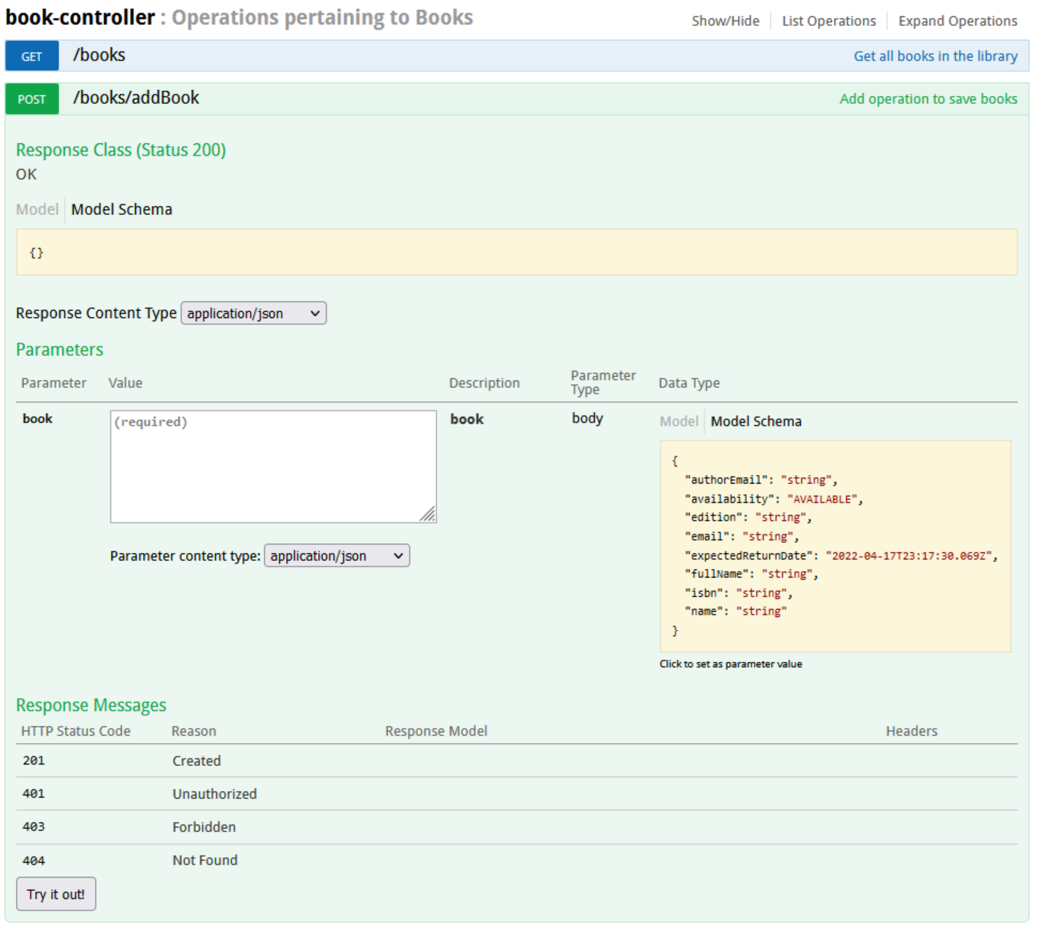
We implemented CRUD functionality into are application to allow us to have the required functionality for our application.



..		
controller	fixed the session bug	7 hours ago
dao	added delete functionality in ui and also create delete api by isnb a...	yesterday
entity	fixed the session bug	7 hours ago
enumeration	created new api to handle checkout	19 hours ago
exceptionhandling	added swager configs for documentations, added more dependencies and ...	4 days ago
model	added service for book, added api to save books	20 hours ago
service	created new api to handle checkout	19 hours ago
swagger	added swager configs for documentations, added more dependencies and ...	4 days ago

… and Standardizing the Back End

We also implemented Swagger into our Back End. This allows for easier standardization and testing for our development team as we worked on this application.



book-controller : Operations pertaining to Books

GET /books — Get all books in the library

POST /books/addBook — Add operation to save books

Response Class (Status 200)

OK

```
{}
```

Response Content Type: application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
book	(required) Parameter content type: application/json	book	body	Model Schema (see below)

```
{
  "authorEmail": "string",
  "availability": "AVAILABLE",
  "edition": "string",
  "email": "string",
  "expectedReturnDate": "2022-04-17T23:17:30.069Z",
  "fullName": "string",
  "isbn": "string",
  "name": "string"
}
```

Click to set as parameter value

Response Messages

HTTP Status Code	Reason	Response Model	Headers
201	Created		
401	Unauthorized		
403	Forbidden		
404	Not Found		

Use Cases

- What we tested:
  - Registration
  - Login
  - Book Checkout
  - Adding a Book
  - Updating a Book
  - And Deleting a Book
- Video Demonstration
  - Can be found within our repository under the Use Cases Directory

Conclusion

- Successfully created and ran our Application
  - Dedicated Front and Back End to allow for easier upgrades in the future
  - Complete the milestones the team set out to attain

Video Demonstration

- Video Demonstration can be found under the Project Demonstration folder of the GitHub Repository
- This url will be posted in the Resources Slide

Resources

- Final Repository:
  - https://github.com/willissa2121/library_checkout_proj_3
- Angular Information:
  - https://angular.io/
- Swagger Information:
  - https://swagger.io/solutions/api-design/

Resources Cont.

- 2nd Repository:
  - https://github.com/willissa2121/library_checkout_proj_2
- 1st Repository:
  - https://github.com/willissa2121/library_checkout_proj