Bookstore Web App

Name: Zaharie Oana Denisa
Group: 30234

Table of Contents

Deliverable 1

Project Specification

The Bookstore web application aims to provide an online platform for users to search and purchase books. Users should be able to create accounts, add books to their cart, and complete purchases securely. Additionally, the application will allow administrators to manage the book inventory, user accounts and sales.

Functional Requirements

- User Registration: users can create accounts and log in with their credentials
- Book Searching: users can browse through available books
- Order history: users can see the orders they placed
- Cart Management: users can add books to their shopping cart and view and modify it
- Place order: users can place an order and pay for it
- Administrators can manage books by adding new ones or updating the existing books
- Administrators can view all sales

Use Case Model 1

Use Cases Identification

Use-Case: Search Books

Level: user

Primary Actor: client

Main success scenario:

- User selects filters to search for books
- System retrieves relevant books based on the search criteria
- User views the list of search results
- User selects a book from the list to view details

Extensions:

- If no matching books are found, the system displays a message

Use-Case: Add to Cart

Level: user

Primary Actor: client

Main success scenario:

- User clicks on the "Add to Cart" button
- System adds the selected book to the user's shopping cart

Extensions:

- If the selected book is out of stock, the system displays a message

Use-Case: View Sales

Level: user

Primary Actor: administrator

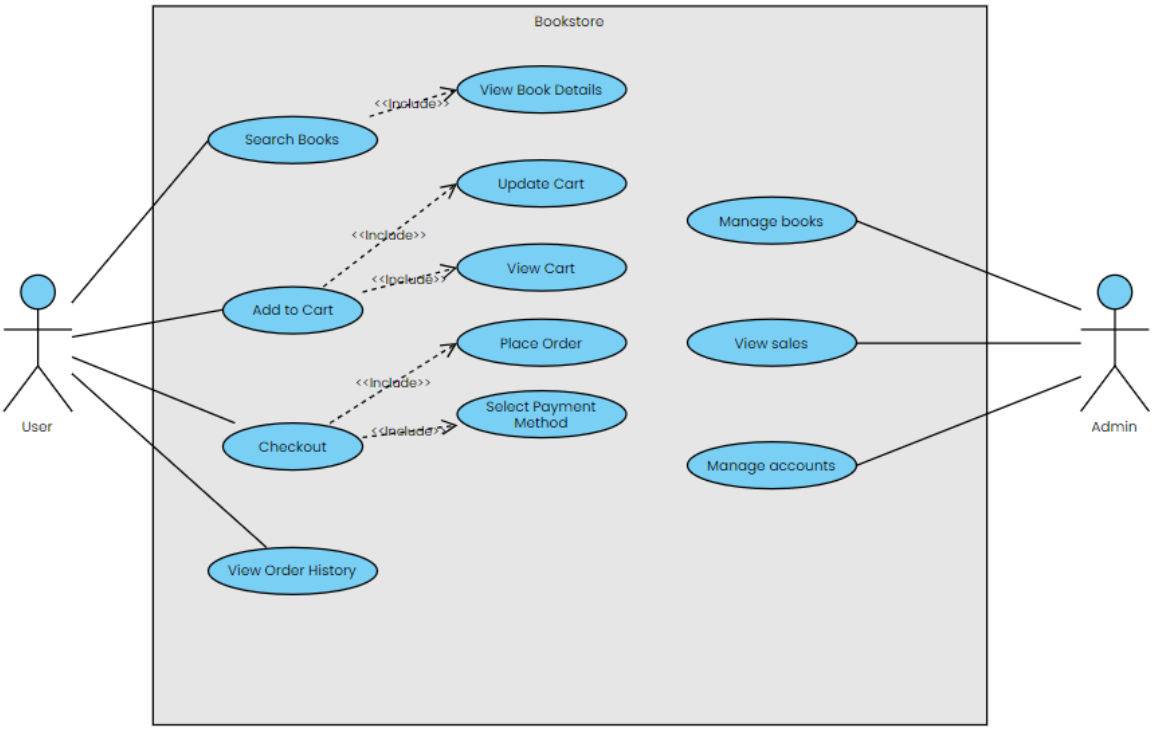
Main success scenario:

- Administrator navigates to the sales reporting section
- System retrieves sales data
- Administrator views the sales report

Extensions:

- If there are no sales data available, the system displays a message

UML Use Case Diagrams



Supplementary Specification

Non-functional Requirements

1. Performance: The system should be able to handle a large number of concurrent users without a loss in performance and the response time for user actions should be under 2 seconds. This is important so that users don't give up on using the app due to a long waiting time.
2. Security: Users data should be securely stored, to avoid personal data theft
3. Usability: the user interface should be intuitive and user-friendly, for an enjoyable experience
4. Portability: The application should be accessible across different devices and screen sizes. It should not be dependent on the operating system or browser used, so that no one is restricted from using the app

Design Constraints

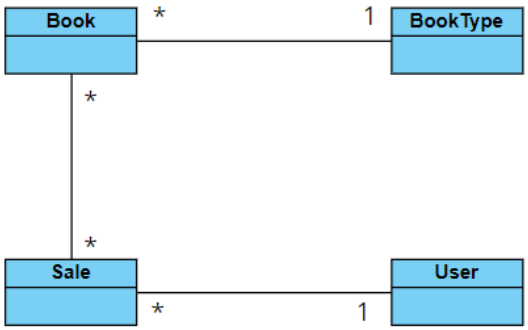
1. Technology Stack:
    - The application will be developed using Java Spring Boot framework
    - Front-end components will be implemented using React
    - Spring Security will be used for authentication and authorization purposes.
    - Data will be employed for seamless integration with the MySQL database.
2. Architectural Constraints:
    - The application will adhere to a Layered Architecture for clear separation of concerns
    - The application will utilize Hibernate ORM for database interaction, ensuring efficient data access and management

Glossary

[Present the noteworthy terms and their definition, format and validation rules if appropriate.]

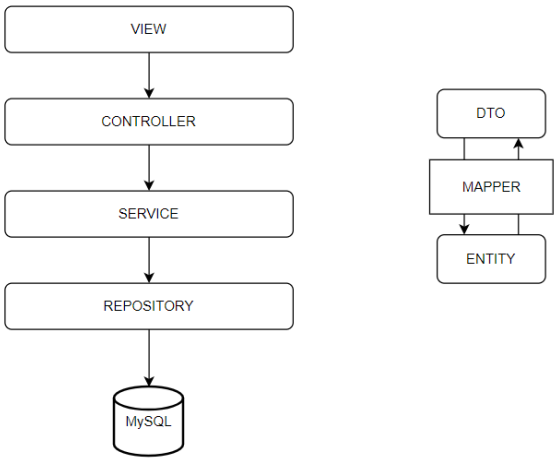
Deliverable 2

Domain Model



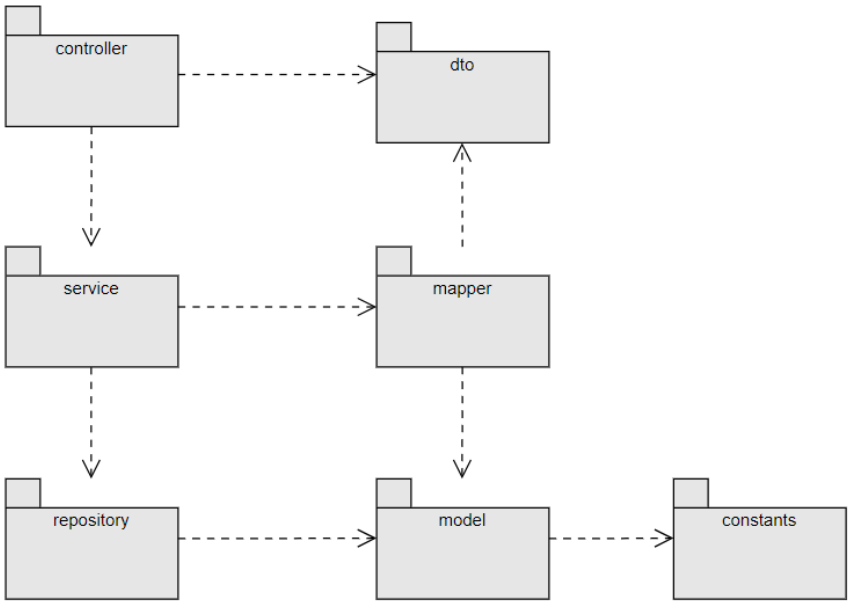
Architectural Design

Conceptual Architecture

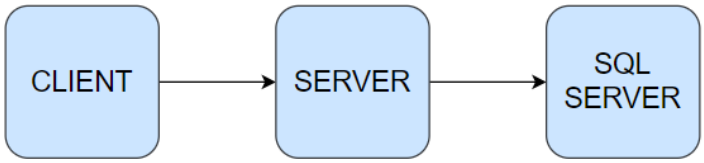


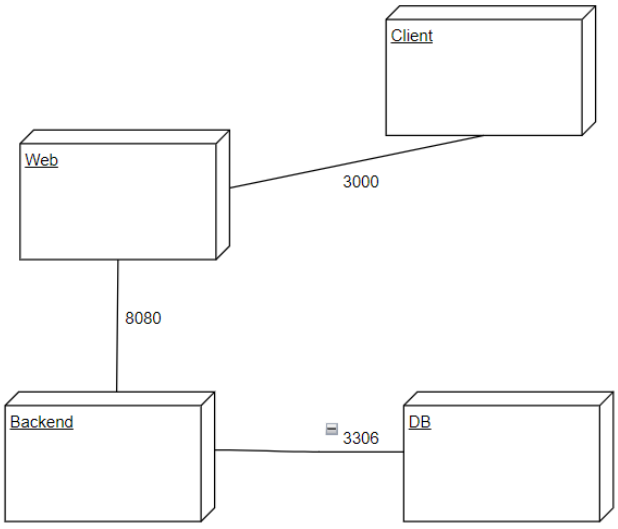
For this project, I used a layered architecture. It helps improve the maintainability, scalability and testability of the code. It also facilitates the reuse of existing components and reduces the coupling of the classes.

Package Design



Component and Deployment Diagram





Deliverable 3

Design Model

Dynamic Behavior

[Create the interaction diagrams (2 sequence) for 2 relevant scenarios]

Class Diagram

[Create the UML class diagram; apply GoF patterns and motivate your choice]

Data Model

[Create the data model for the system.]

System Testing

[Describe the testing methides and some test cases.]

Future Improvements

[Present some features that apply to the application scope.]

Conclusion

Bibliography