

OUSL ONLINE BOOK STORE

2024/11/09



Contents

1.Introduction

1.1Background of the OUSLBookStore

1.2Objectives of the System

2. System Requirements

2.1 Functional Requirements

2.2Non-Functional Requirements

3. System Design

3.1 UML Diagram

4. Implementation Plan

4.1 Front-End Development

4.2 Back-End Development

4.3 Payment Gateway Integration

4.4 Security Implementation

4.5 Testing

4.6 Deployment

5. Gathering Results

6. Conclusion

1.Introduction

1.1Background of the OUSLBookStore

OUSLBookStore, a small online bookstore, aims to modernize its operations by shifting from manual or basic digital processes to a comprehensive web-based management system. This new platform will address existing inefficiencies in handling customer orders, managing inventory, and processing payments, all of which are currently limited by manual workflows.

With this digital transition, OUSLBookStore can offer customers an improved experience by enabling online browsing, purchasing, and payment options, making it easier for users to engage with the bookstore. Additionally, the platform will enhance internal processes for tracking inventory and monitoring sales, supporting scalability to accommodate multiple users and provide a seamless interface across both desktop and mobile devices. This upgrade will help OUSLBookStore reach a broader, more diverse customer base, ensuring competitiveness in the online retail space.

1.2Objectives of the System

The OUSLBookStore project aims to build a web-based platform that streamlines the entire online book sales process. The platform will offer a user-friendly experience, allowing customers to:

- Create accounts
- Browse and search for books
- Add books to a shopping cart
- Complete secure online payments

For bookstore staff, the system will provide an administrative interface to:

- Manage book inventory (add, update, and remove books)
- Generate sales reports

for order and revenue tracking

Key design priorities include security, scalability, and responsiveness, ensuring smooth performance for up to 1,000 users and accessibility on both desktop and mobile devices. This report will cover the system's development process, from requirements gathering and design to implementation, testing, and deployment, aligned with OUSLBookStore's business goals.

2. System Requirements

2.1 Functional Requirements

Must Have

1. Customer registration and login functionality.
2. Book browsing by title, author, and genre.
3. Shopping cart to add/remove books before purchase.
4. Secure online payment gateway for checkout.
5. Admin interface for adding, updating, or removing books from inventory.
6. Admin functionality to generate sales reports to track orders and revenue.

Should Have

1. Search functionality with filters like genre, author, and rating.
2. Mobile and desktop responsive design.

Could Have

1. Book rating and review system by customers.
2. Recommendation system based on previous purchases or preferences.

Won't Have

1. Social media integration for now.
2. Advanced analytics beyond basic sales reports.

2.2 Non-Functional Requirements

Must Have

1. Secure transactions with encryption (e.g., HTTPS, SSL).
2. Handle up to 1,000 concurrent users.
3. Ensure availability across both mobile and desktop devices.
4. User-friendly design for all features.

Should Have

1. System scalability to handle future traffic growth.
2. 24/7 availability and minimal downtime.

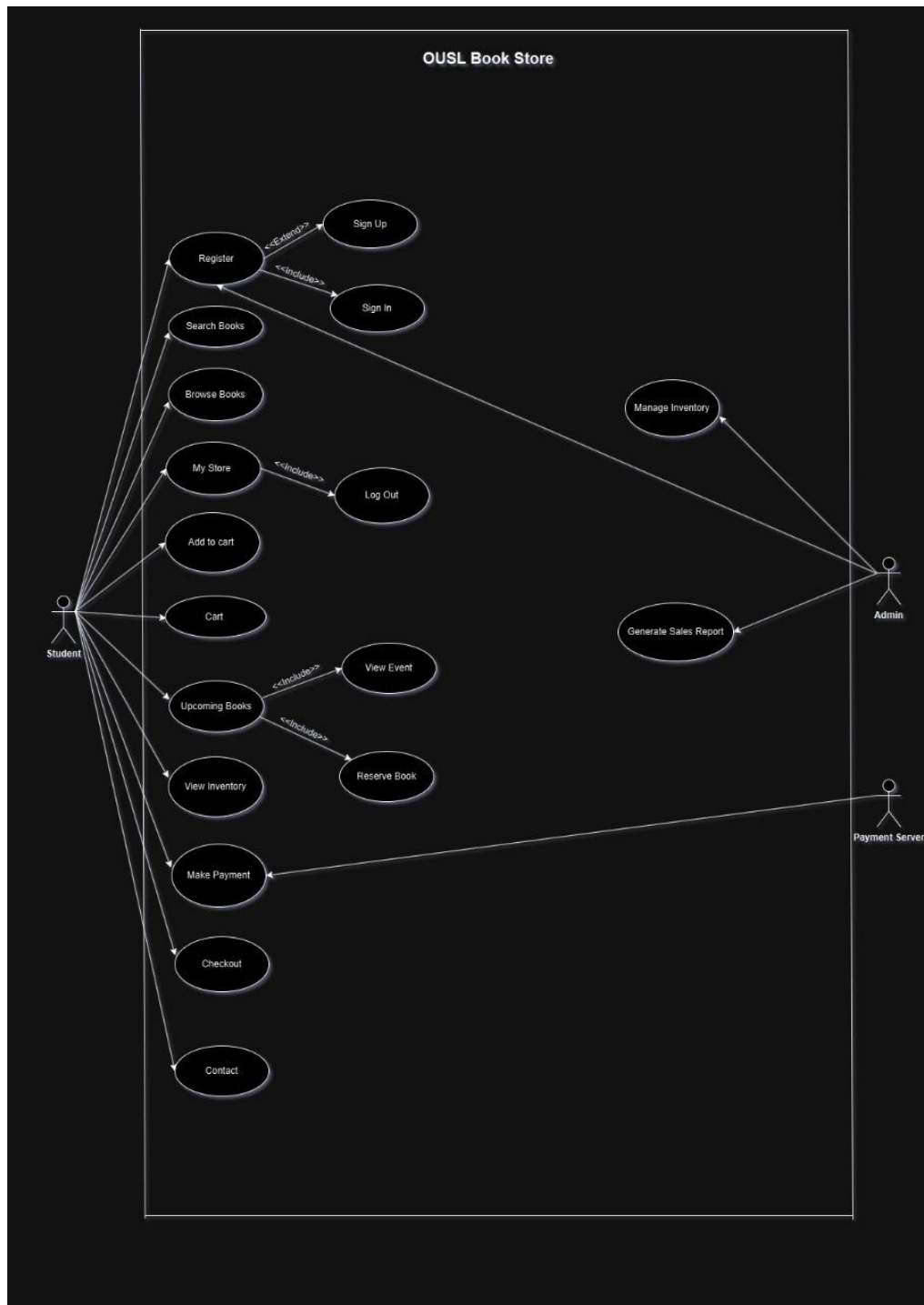
Could Have

1. Multi-language support for users from different regions.

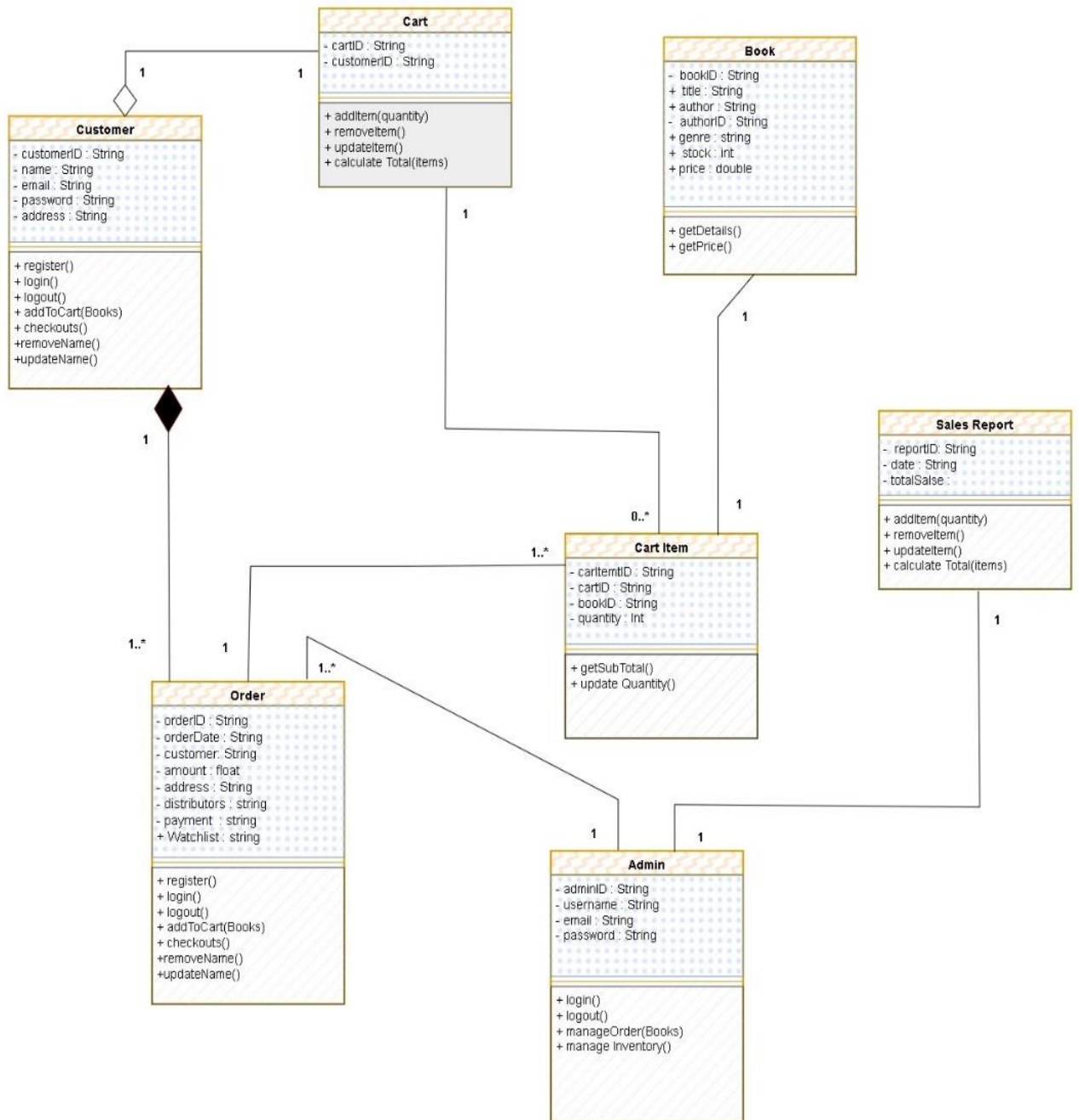
3. System Design

3.1 UML Diagram

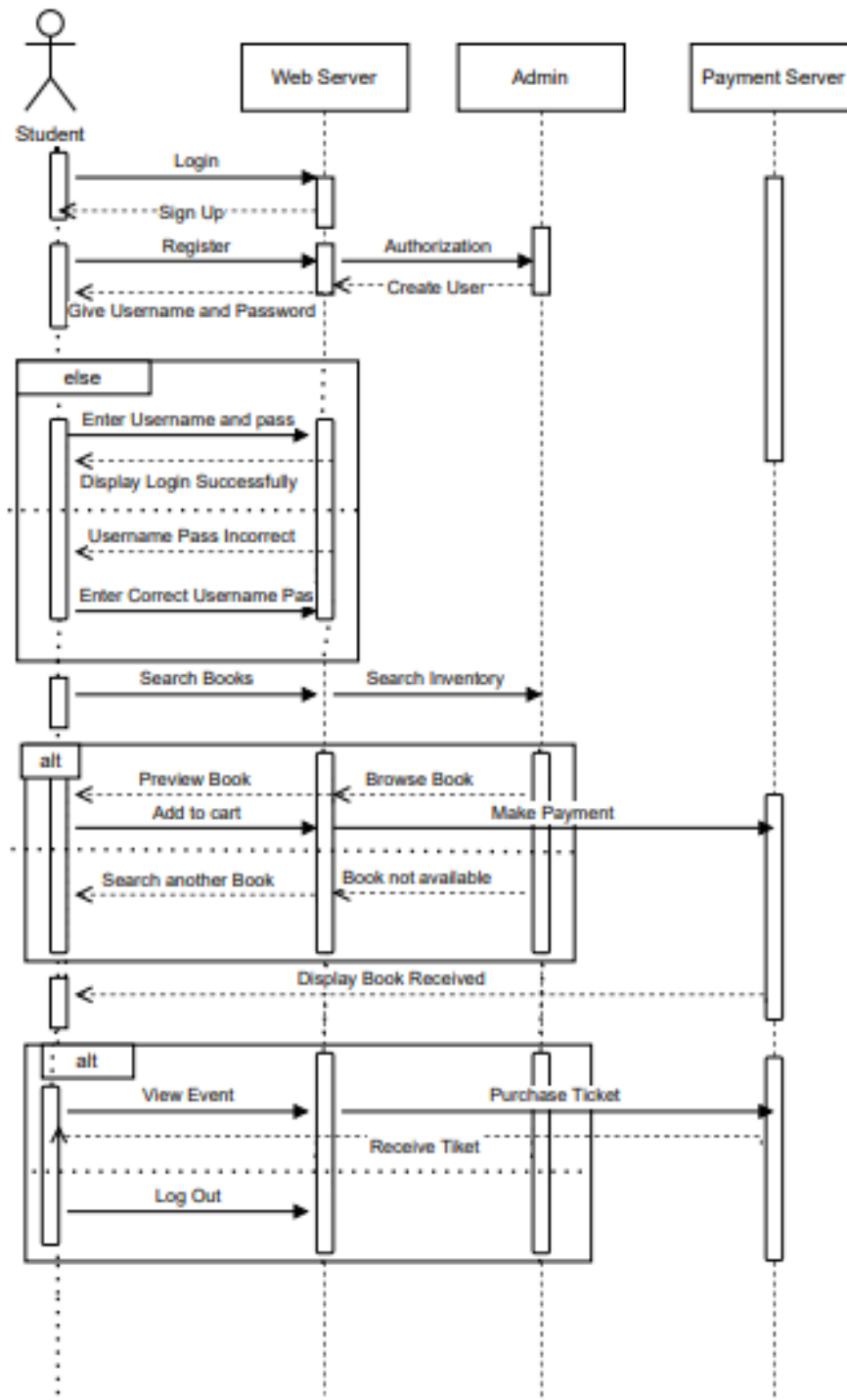
3.1.1 Use Case Diagram



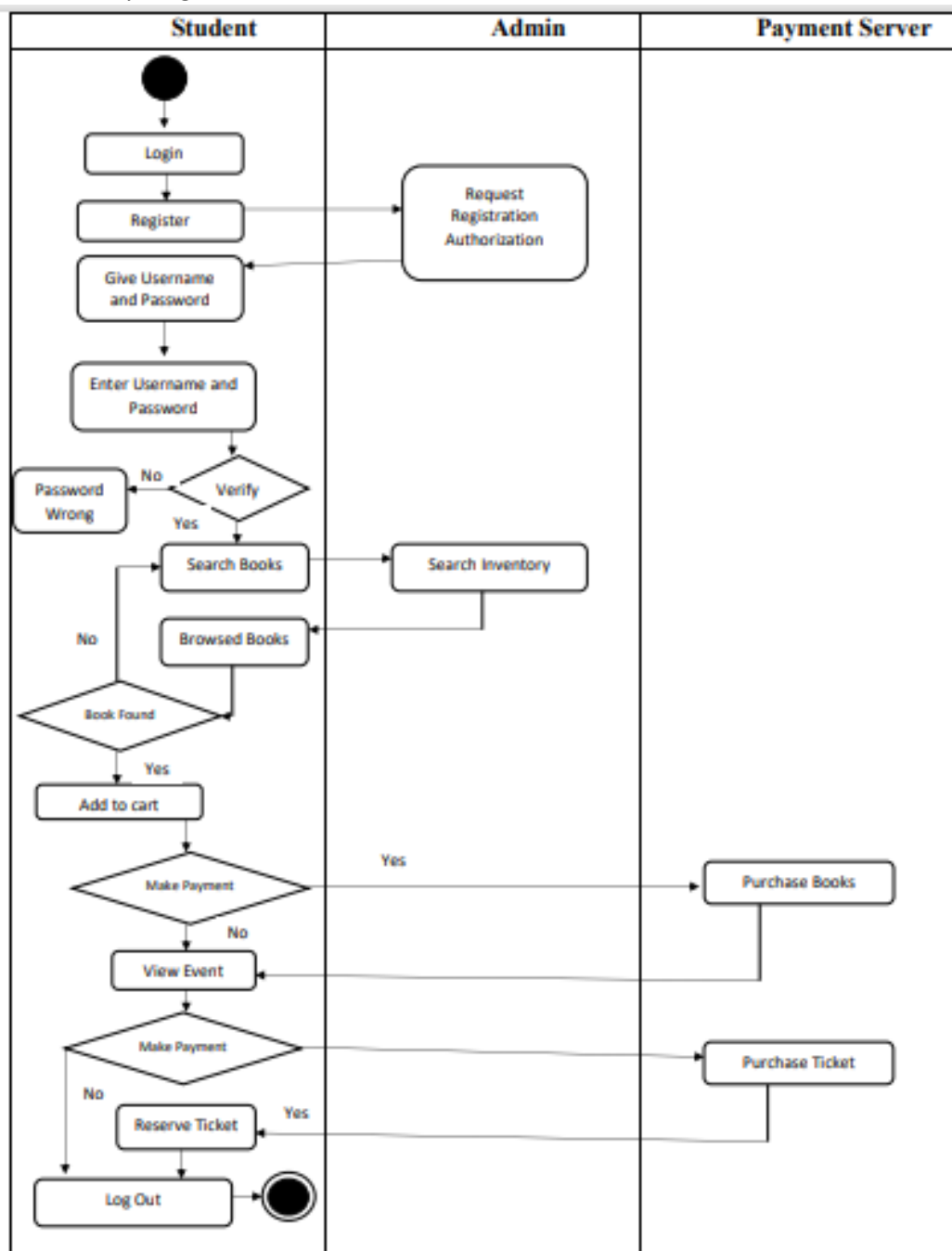
3.1.2 Class Diagram



3.1.3 Sequence Diagram



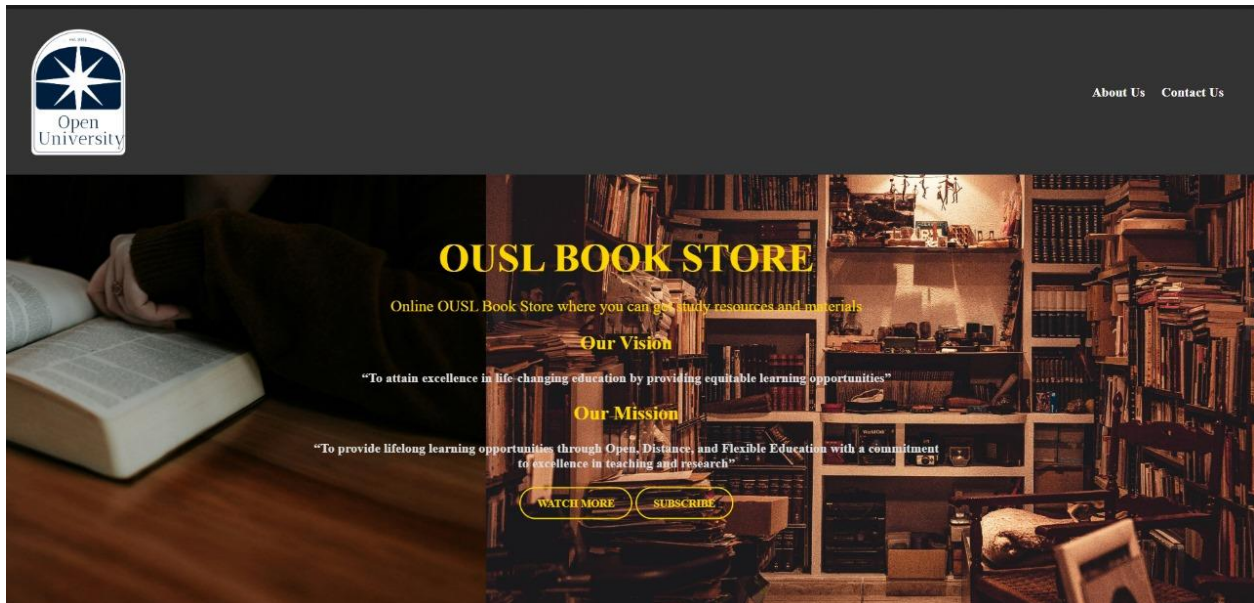
3.1.4 Activity Diagram



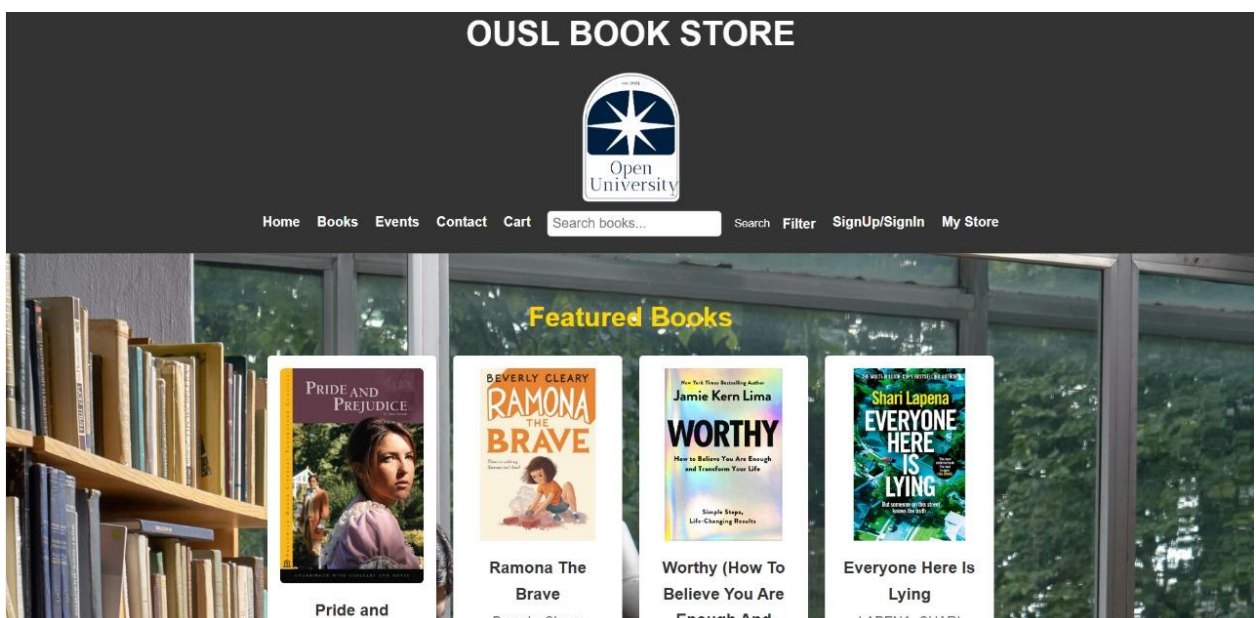
4. Implementation Plan

4.1 Front-End Development

- Welcome page



- Home page
 - Implement browsing and search features for books.



- account registration and sign in

OUSL Book Store

[Home](#) [Books](#) [Events](#) [Cart](#) [Sign In](#)

Full Name

Email

Password

Confirm Password

Sign Up

Already have an account? [Sign In here.](#)

© 2024 OUSL BookStore. All Rights Reserved.

OUSL Book Store

[Home](#) [Books](#) [Events](#) [Cart](#) [Sign Up](#)

Email

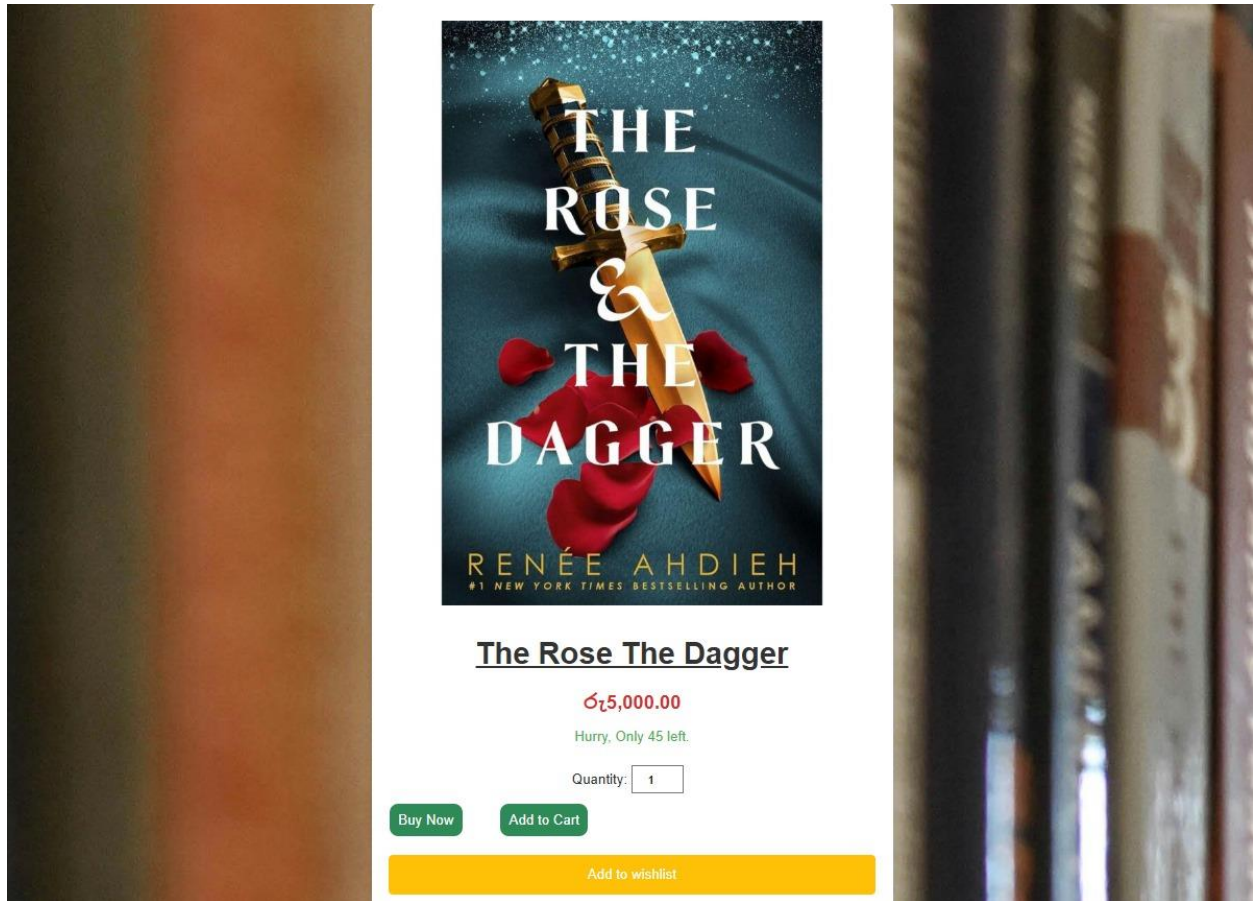
Password

Sign In

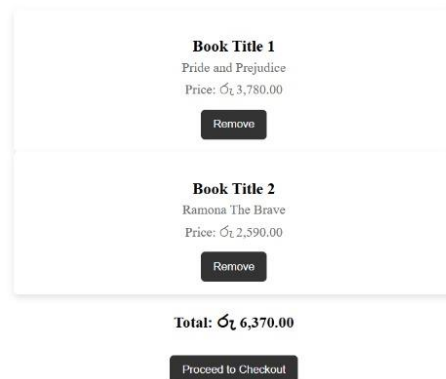
Don't have an account? [Sign Up here.](#)

© 2024 OUSL BookStore. All Rights Reserved.

o Create a shopping cart interface to add/remove books and checkout.



Your Cart



- Set up payment integration for the checkout process.

OUSL Book Store

Checkout

Order Summary

Book Title 1 - € 3,780.00

Book Title 2 - € 2,590.00

Total: € 6,370.00

Shipping Information

Full Name

Address

City

ZIP Code

Country

Payment Information

Card Number

Expiration Date

CVV

Confirm Order

- **Admin Interface:**
 - o Build a dashboard for administrators to add, update, and remove books.

Admin Dashboard

Logout

Manage Book Inventory

Add New Book

Book ID	Title	Author	Price	Quantity	Actions
1	Pride and Prejudice	Jane Austen	₺ 3,780.00	1	<div>EditDelete</div>
1	Ramona The Brave	Beverly, Cleary	₺ 2,590.00	1	<div>EditDelete</div>

Generate Sales Report

Generate Report

Admin Dashboard - Sales Report

HomeManage InventoryLogout

Sales Report

Start Date:

mm/dd/yyyy

End Date:

mm/dd/yyyy

Filter

Order ID	Customer Name	Book Title	Quantity	Price	Date	Total
102	Dilshan	Pride and Prejudice	1	₺ 3,780.00	2024-11-01	₺ 3,780.00
101	Perera	Ramona The Brave	1	₺ 2,590.00	2024-11-02	₺ 2,590.00

Export as CSV

© 2024 OUSL Book Store Admin. All Rights Reserved.

4.2 Back-End Development

Task: Create the backend API to manage business logic and database operations.

- **Customer Endpoints:**
 - ✓ Account Registration
 - ✓ User SignIn
 - ✓ Book Browsing
 - ✓ Shopping Cart Management
 - ✓ Checkout Process
- **Admin Endpoints:**
 - ✓ Manage Book Inventory
 - ✓ Generate Sales Reports
- **Security:** Protect API endpoints with JSON Web Tokens (JWT) for authentication.
- **Tech Stack:** Options include Python (Django/Flask), Node.js (Express), or Java (Spring Boot).

4.3 Payment Gateway Integration

Task: Implement a secure third-party payment gateway for transaction processing.

- **Steps:**
 - a. Create an account with either Stripe or PayPal.
 - b. Integrate the payment gateway's API to process transactions in the checkout route.
 - c. Manage success and failure callbacks from the payment provider.
 - d. Record successful transactions in the Orders table
- **Tools:** Use the official Stripe or PayPal SDK compatible with the back-end technology in use.

4.4 Security Implementation

Task: Ensure system security for safe transactions and data protection.

- **Steps:**
 - a. Enable HTTPS to secure all communications.
 - b. Follow the payment gateway's security guidelines, including PCI compliance, to protect the payment process.

- c. Encrypt sensitive data, such as customer passwords, with bcrypt or equivalent encryption
- d. Apply input validation and guard against SQL injection and cross-site scripting (XSS) vulnerabilities.

4.5 Testing

Task: Conduct thorough testing of the system.

Unit Testing: Validate back-end business logic, such as cart calculations and checkout functionality.

Integration Testing: Verify seamless interaction between front-end and back-end components.

Load Testing: Emulate up to 1,000 simultaneous users to assess system scalability.

Payment Gateway Testing: Use sandbox environments provided by payment gateways to test transaction flows.

4.6 Deployment

Task: Launch the system on a cloud platform for production use.

Steps:

- a. Provision a server on AWS, Heroku, or DigitalOcean.
- b. Deploy both the back-end API and front-end application.
- c. Use a managed database service (e.g., AWS RDS, Heroku Postgres) for database setup.
- d. Configure domain settings and install SSL certificates to enable HTTPS.

5. Gathering Results

User Feedback: Collect input on the ease of use for both customer and admin interfaces.

Performance Monitoring: Track system performance during peak traffic to ensure scalability.

Sales Monitoring: Utilize admin reports to verify accurate capture of sales data.

6. Conclusion

The OUSLBookStore web platform addresses current operational challenges by providing a streamlined solution for online book sales. It enables customers to create accounts, browse, add items to a cart, and make secure payments, while an administrative interface allows bookstore staff to manage inventory and track sales. With a user-friendly design, secure payment integration, and scalability for up to 1,000 users, this system is built for performance and growth. Key focuses include security, mobile responsiveness, and scalability to ensure readiness for production and alignment with business goals.

Group members:-

M.K.G Chithrasena -223586544

G.W.L.D.Bandara-623574615

P G B D D Gunawardhana-223577588

Chamod gamage - 623574097