E-BOOK STORE

Project Documentation

A Seamless Platform
for Buying and Selling
Digital Books

2025

Project Planning & Management

Project Overview

We are developing a specialized eBook selling platform with interactive features that enhance the reading and learning experience. The platform will support two types of accounts:

1- Sellers (Authors/Publishers) – They can upload and manage their eBooks.
2- Buyers (Readers) – They can purchase and interact with the eBooks.

Unlike traditional eBook stores, our platform will enhance the reading experience by allowing books to include:

- Audio narration for listening to books.
- Embedded videos inside eBooks.

Project Plan

2-Month Execution Timeline

Phase 1: Planning & Design (Week 2-1)

- Define project requirements and features.
- Design the database structure.
- Create the UI/UX design (using Figma or similar tools).

Phase 2: Backend Development (Week 4-3)

- Develop the backend using ASP.NET Core.
- Implement user authentication (JWT) for security.
- Build the eBook management system (upload, edit, delete books).

Phase 3 : Testing & Deployment (Week 8-7)

- Perform QA testing (security, performance, and usability).
- Fix bugs and optimize performance.

Roles & Responsibilities

- Full-Stack Developer (.NET & Blazor) – Develop backend & frontend.
- UI/UX Designer – Create an intuitive and user-friendly experience.
- QA Tester – Ensure quality, fix bugs, and optimize performance.
- Marketing & Sales Team – Promote the platform to authors and readers.

Risk Assessment & Mitigation Plan

1- Development Delays
- Risk: Unexpected technical issues or slow progress in development.
- Mitigation: Implement Agile methodology with bi-weekly sprints to track progress and adjust timelines accordingly.

2- Low User Adoption
- Risk: Authors and buyers may not engage with the platform as expected.
- Mitigation: Launch a targeted marketing campaign, offer early-bird incentives for sellers, and provide interactive demos to attract users.

3- Legal & Copyright Issues
- Risk: Potential conflicts over eBook ownership and content rights.
- Mitigation: Implement digital contracts that clearly define ownership rights and integrate copyright protection tools (such as watermarking or DRM).

4- Security Vulnerabilities
- Risk: Cybersecurity threats like unauthorized access or data breaches.
- Mitigation: Use secure authentication (JWT, OAuth), encrypt sensitive data, and conduct regular security audits.

5- Scalability Challenges
- Risk: The platform might struggle to handle a growing number of users and eBooks.
- Mitigation: Develop a cloud-based scalable infrastructure (Azure/AWS) and optimize database performance from the start.

Key Performance Indicators (KPIs)

Response Time – Ensure fast platform performance.
User Adoption Rate – Track how many sellers & buyers register.
Average Time Spent on Site – Measure engagement with interactive eBooks.
Monthly Book Sales – Evaluate platform success in attracting buyers.

Literature Review

Feedback & Evaluation

Our lecturer provided an assessment of our project, acknowledging the strength of our business logic and confirming that no changes were required in this aspect. However, feedback was given regarding the Entity-Relationship Diagram (ERD), which required certain modifications to better represent data relationships within our system. Based on this feedback, we updated the ERD and attached the revised version to our documentation.

Suggested Improvements

While the core business logic was validated as sound, several improvements were suggested to enhance the system's usability and functionality:

Database Enhancements
Refining the ERD structure to ensure better data representation and integrity.

AI-Powered Book Design
Adding a built-in editor that allows sellers to design and format their eBooks using AI-driven templates and tools, making the publishing process more intuitive and efficient.

Final Grading Criteria

The project's evaluation is based on multiple aspects, including:

- Documentation – Completeness and clarity of system analysis and design.
- Implementation – Functionality of core features and adherence to requirements.
- Testing – Ensuring reliability and bug-free performance.
- Presentation – Clear demonstration of the system's features and benefits.

Requirements Gathering

Stackholder Analysis

1. Authors

- They write eBooks and provide content for the platform.
- They can upload their books, set prices (in some models), and interact with readers.

2. Publishers

- They supply the platform with professional content, such as textbooks or popular novels.
- They may have licensing agreements with the platform to sell eBooks.

3. Readers/Users

- The end-users who buy, rent, or read eBooks through the platform.
- They may have features like bookmarks, comments, and recommendations.

4. Platform Administrators

- They oversee platform operations, including content management, technical support, and issue resolution.
- They ensure compliance with copyright and data protection regulations.

5. Developers

- They design and develop the eBook platform, whether as a website or mobile app.
- They are responsible for features such as search, online payment, and interactive reading experiences.

6. Marketers

- They promote the platform through social media, SEO, and advertisements.
- They work on attracting readers and increasing subscriptions or sales.

7. Payment Providers

- They facilitate online payment transactions, such as PayPal, Visa, or subscription services.
- They ensure the security of financial transactions on the platform.

8. Digital Distributors

- They help distribute books to other platforms (such as Amazon Kindle or Apple Books).
- They work on expanding the reach of books to a broader audience.

9. Regulators & Legal Advisors

- They ensure the platform complies with digital publishing laws and intellectual property rights.
- They help resolve legal disputes between authors, users, or the platform.

User Stories & Use Cases

1. User Stories

User stories describe the needs and expectations of different stakeholders in the eBook platform.

- As a reader, I want to browse and search for eBooks so that I can find books that match my interests.
- As a reader, I want to purchase or rent eBooks securely so that I can access them anytime from my library.
- As a reader, I want to highlight text and add notes so that I can remember key points.
- As a reader, I want to leave reviews and rate books so that I can help other readers make informed choices.
- As an author, I want to upload and publish my eBooks so that I can reach a wider audience.
- As an author, I want to track my sales and earnings so that I can evaluate my book's performance.
- As an author, I want to receive feedback and respond to readers so that I can engage with my audience.
- As a platform administrator, I want to review and approve new books before publishing so that I can maintain quality standards.
- As a platform administrator, I want to manage users and their roles so that I can prevent misuse of the platform.
- As a platform administrator, I want to generate reports on book sales and user activity so that I can analyze platform performance.
- As a marketer, I want to send promotional emails and recommendations so that I can increase book sales.
- As a marketer, I want to track user behavior and engagement so that I can optimize marketing campaigns.

2. Use Case Scenarios

Use cases describe how users interact with the system to achieve specific goals.

Use Case 1: Buying an eBook

- Actor: Reader
- Steps:

1. The reader searches for an eBook using keywords or categories.
2. The system displays relevant search results.
3. The reader selects a book and views its details.
4. The reader clicks "Buy Now" and proceeds to checkout.
5. The system processes the payment through a secure gateway.
6. The book is added to the reader's digital library.
7. The reader can now access and read the book anytime.

Use Case 2: Uploading an eBook as an Author

- Actor: Author
- Steps:

1. The author navigates to the "Upload eBook" section.
2. The author fills in the book title, description, and category.
3. The author uploads the eBook file and sets the price.
4. The system validates the file format and content.
5. The book is submitted for review by an administrator.
6. Upon approval, the book is published on the platform.

Use Case 3: Leaving a Book Review

- Actor: Reader
- Steps:

1. The reader navigates to the book's page.
2. The reader clicks on "Write a Review."
3. The reader rates the book and writes a review.
4. The system verifies the review and publishes it.
5. Other users can see and interact with the review.

Use Case 4 : Admin Management

- Actor: Admin
- Steps:

1. The admin logs into the system and accesses the admin dashboard.
2. The admin selects the desired management option from the following:
   - Manage Books:
     - Review and approve/reject newly uploaded books.
     - Edit or delete existing books.
     - Feature or promote specific books on the homepage.
   - Manage Users:
     - Add, edit, or delete user accounts (readers and authors).
     - Ban users for policy violations.
     - Assign or change user roles (e.g., promote a user to admin).
   - Manage Reviews:
     - Approve, reject, or delete user reviews.
     - Edit review content if necessary.
     - Respond to user reviews (e.g., reply to complaints or feedback).
   - Manage Payments:
     - View all payment transactions.
     - Refund payments in case of disputes.
     - Resolve payment gateway issues.

Functional Requirements

Functional Requirements – List of Features and Functionalities
Functional requirements define the features and functionalities that the eBook platform must provide to ensure a seamless user experience. Below is a list of key features with a brief explanation of each:

1. User Management

- Account Creation & Login: Users can register using email or social media accounts.
- Profile Management: Users can update personal information such as name and password.
- Role Management: Each user has a specific role (Reader, Author, Publisher, Admin).

2. eBook Management

- Book Uploading: Authors and publishers can upload their books in formats such as PDF and EPUB.
- Metadata Management: Titles, descriptions, categories, and pricing (if applicable) can be set.
- Book Approval: Admins review books before publishing.
- Edit & Delete Books: Authors can update or remove their books from the store.

3. Search & Navigation

- Advanced Search: Users can search for books by title, author, category, or keywords.
- Category Browsing: Books are categorized into sections like Fiction, Science, Technology, etc.
- Smart Recommendations: The system suggests books based on user interests and reading history.

4. Reading Experience

- Built-in eBook Reader: Users can read books directly within the platform without downloading them.
- Bookmarks: Users can save pages for future reference.
- Highlighting & Notes: Users can highlight text and add personal notes.
- Dark Mode: Night mode options for a comfortable reading experience.

5. Ratings & Reviews

- Book Ratings: Users can rate books from 1 to 5 stars.
- Write Reviews: Users can leave comments about books.
- Display Ratings: Books show an average rating based on user reviews.

6. Payments & Transactions

- Secure Purchases: Support for various payment methods such as credit cards, PayPal, and e-wallets.
- Subscription Management: Some books may be available only through a monthly subscription.
- Invoice Generation: Users receive electronic receipts after purchases.
- Revenue Sharing for Authors: Authors receive earnings based on book sales.

7. Security & Digital Rights Management (DRM)

- Anti-Piracy Protection: DRM (Digital Rights Management) technology prevents unauthorized book sharing.
- Two-Factor Authentication: Adds an extra layer of security for user accounts.
- Data Encryption: Protects user and financial information.

8. Notifications & Alerts

- Promotional Alerts: Users get notified about book discounts and offers.
- Activity Notifications: Authors receive alerts when readers leave reviews.
- Payment Notifications: Users receive confirmations for purchases and subscriptions.

9. Admin Dashboard

- User Management: Admins can modify or disable user accounts when necessary.
- Content Management: Admins review books before publishing and remove inappropriate content.
- Reports & Analytics: Sales tracking, user engagement, and activity analysis.

Non-functional Requirements

Non-Functional Requirements - Performance, Security, Usability, and Reliability Criteria

Non-functional requirements define the quality attributes of the **eBook platform** that ensure a smooth, secure, and reliable user experience. These requirements focus on how the system performs rather than what it does.

1. Performance

- Fast Response Time: The platform should load pages and process user requests within 2 seconds under normal conditions.
- Scalability: The system should handle thousands of concurrent users without performance degradation.
- Efficient Book Search: Search queries should return results in less than 1 second for a seamless experience.
- Optimized Book Downloads & Streaming: eBooks should open instantly without long loading times.

2. Security

- User Authentication & Authorization: Secure login with two-factor authentication (2FA) to prevent unauthorized access.
- Data Encryption: All sensitive data (e.g., passwords, payment details) should be encrypted using AES-256 encryption.
- Digital Rights Management (DRM): Prevent illegal sharing and piracy of eBooks through DRM protection.
- Secure Payment Processing: Transactions should comply with PCI-DSS standards for online payments.
- Regular Security Audits: The platform should undergo periodic security testing to identify and fix vulnerabilities.

3. Usability

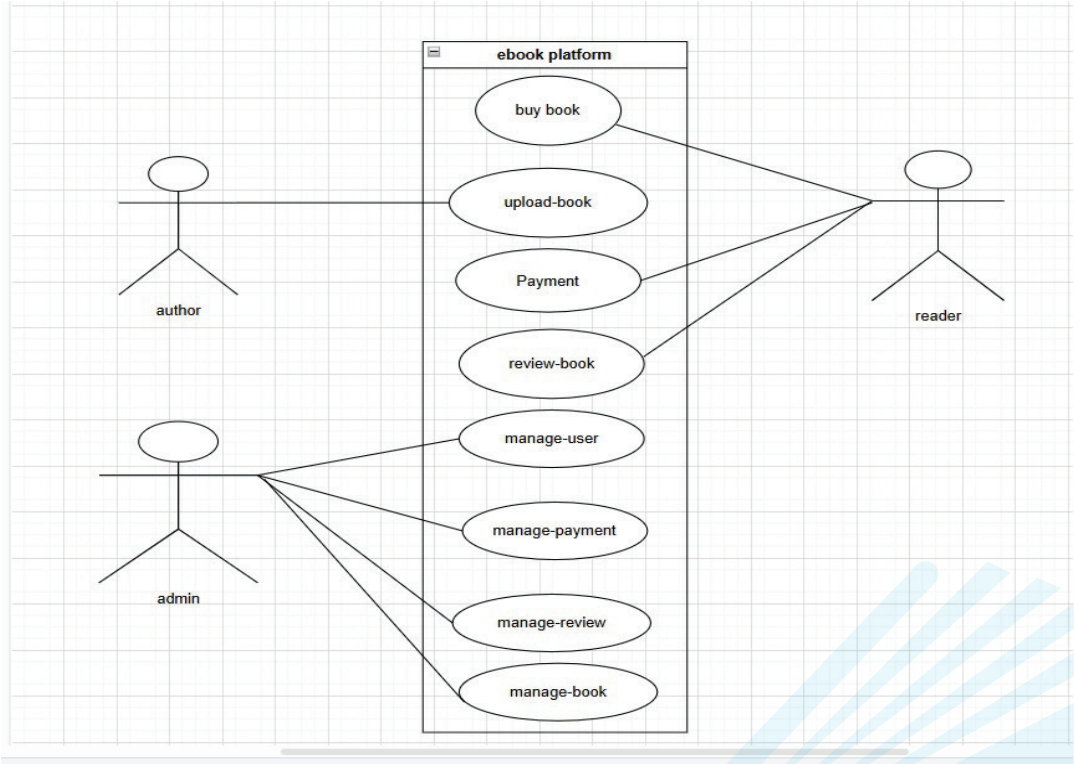
- User-Friendly Interface: The design should be intuitive and easy to navigate, even for first-time users.
- Accessible Design: The platform should follow WCAG (Web Content Accessibility Guidelines) to support users with disabilities.
- Multi-Device Compatibility: Users should be able to access the platform on desktops, tablets, and mobile devices.
- Customizable Reading Experience: Users should have options like font size adjustment, dark mode, and text-to-speech.

4. Reliability & Availability

- Uptime Guarantee: The system should have %99.9 uptime, ensuring continuous availability.
- Automatic Backups: Regular daily backups to prevent data loss.
- Disaster Recovery Plan: In case of a failure, the system should be restored within 30 minutes.
- Load Balancing: The system should distribute traffic across multiple servers to prevent downtime.

System Analysis & Design

Problem Statement & Objectives



1. Use Case Diagram & Descriptions

The eBook Store system involves three main actors:

- Reader: A user who purchases books, makes payments, and reviews books.
- Author (Seller): A user who uploads books to the platform for sale.
- Admin: A system administrator who manages books, users, payments, and reviews.

Use Cases:

- Buy Book: A reader purchases a book through the platform.
- Upload Book: An author uploads a book to the store.
- Payment: A reader completes a transaction to buy a book.
- Review Book: A reader can leave a review for a purchased book.
- Manage User: The admin manages user accounts.
- Manage Payment: The admin oversees and controls payment transactions.
- Manage Review: The admin monitors and moderates book reviews.
- Manage Book: The admin has control over books available on the platform.

2. Functional & Non-Functional Requirements

Functional Requirements:

- Users should be able to create accounts and log in securely.
- Readers should be able to browse, search, and buy books.
- Authors should be able to upload books for sale.
- Readers should be able to make payments through a secure gateway.
- Readers should be able to review purchased books.
- Admins should have control over users, payments, reviews, and books.

Non-Functional Requirements:

- The system should ensure data security and encryption for transactions.
- The platform should be responsive and accessible across devices.
- The system should handle concurrent user transactions efficiently.
- The application should maintain high availability and performance.
- The platform should be scalable to accommodate future growth.

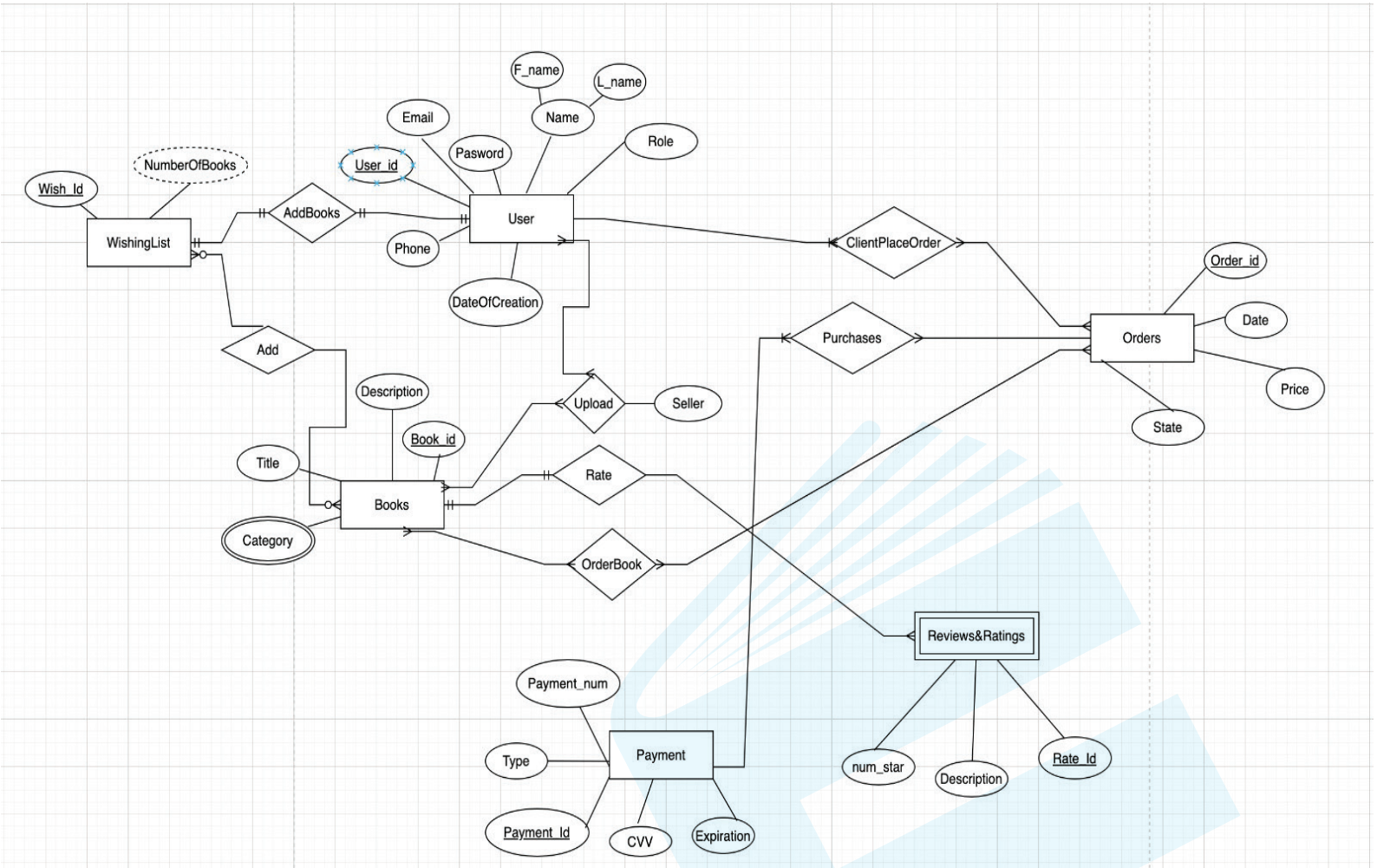
3. Software Architecture

The eBook Store is developed as a full-stack .NET project, incorporating a structured architecture:

- Architecture Style: Follows the MVC (Model-View-Controller) pattern for clear separation of concerns.
- Frontend: Built with Blazor, offering a dynamic user experience.
- Backend: Developed using .NET Core API, handling business logic and data processing.
- Database: Utilizes MS SQL Server for managing books, users, payments, and reviews.
- Authentication & Authorization: Implemented using role-based access control to differentiate between readers, authors, and admins.
- Payment Integration: Secure payment processing to facilitate book purchases.

Database Design & Data Modeling

ER Diagram (Entity-Relationship Diagram)



Logical & Physical Schema

Users Table

Field	Description
UserId	Primary Key, Auto-increment
FirstName	String, 50 characters
LastName	String, 50 characters
Email	String, 100 characters, Unique, Not Null
Password	String, Encrypted, Not Null
Phone	String, 20 characters
Role	Enum: Seller, Buyer, Not Null
DateOfCreation	Timestamp, Default: Current Timestamp

Books Table

Field	Description
BookId	Primary Key, Auto-increment
Title	String, 255 characters, Not Null
Description	Text
Category	String, 100 characters
SellerId	Foreign Key References Users(UserId), On Delete Cascade

WishingList Table

Field	Description
WishId	Primary Key, Auto-increment
UserId	Foreign Key References Users(UserId), On Delete Cascade
NumberOfBooks	Integer, Default: 0

Orders Table

Field	Description
OrderId	Primary Key, Auto-increment
UserId	Foreign Key References Users(UserId), On Delete Cascade
Date	Timestamp, Default: Current Timestamp
Price	Decimal, 10,2
State	Enum: Pending, Completed, Cancelled

OrderBook Table (Mapping Orders to Books)

Field	Description
OrderId	Foreign Key References Orders(OrderId), On Delete Cascade
BookId	Foreign Key References Books(BookId), On Delete Cascade
Primary Key	OrderId, BookId

Payments Table

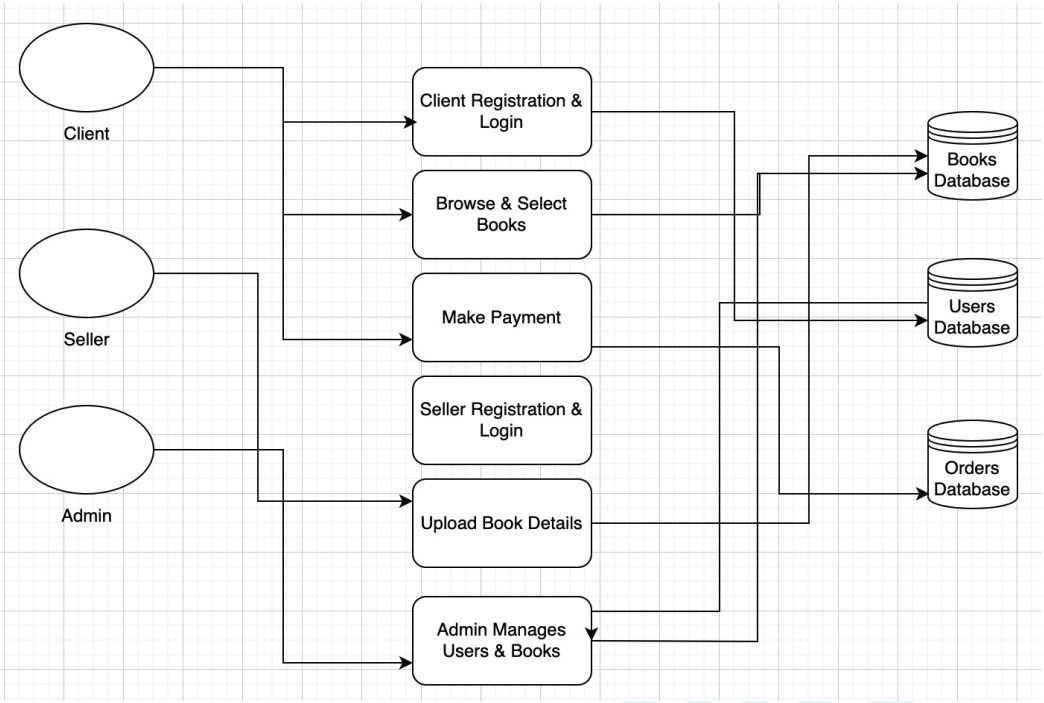
Field	Description
PaymentId	Primary Key, Auto-increment
OrderId	Foreign Key References Orders(OrderId), On Delete Cascade
PaymentNum	String, 50 characters, Unique
Type	String, 50 characters
CVV	String, 10 characters
Expiration	Date

Reviews & Ratings Table

Field	Description
RateId	Primary Key, Auto-increment
UserId	Foreign Key References Users(UserId), On Delete Cascade
BookId	Foreign Key References Books(BookId), On Delete Cascade
NumStars	Integer, 1-5, Constraint: Between 1 and 5
Description	Text

Data Flow & System Behavior

Data Flow Diagram (DFD) for eBook Store



Entities:

- Client – Registers, browses, buys books.
- Seller – Uploads books.
- Admin – Manages users & books.
- Payment Gateway – Handles payments.

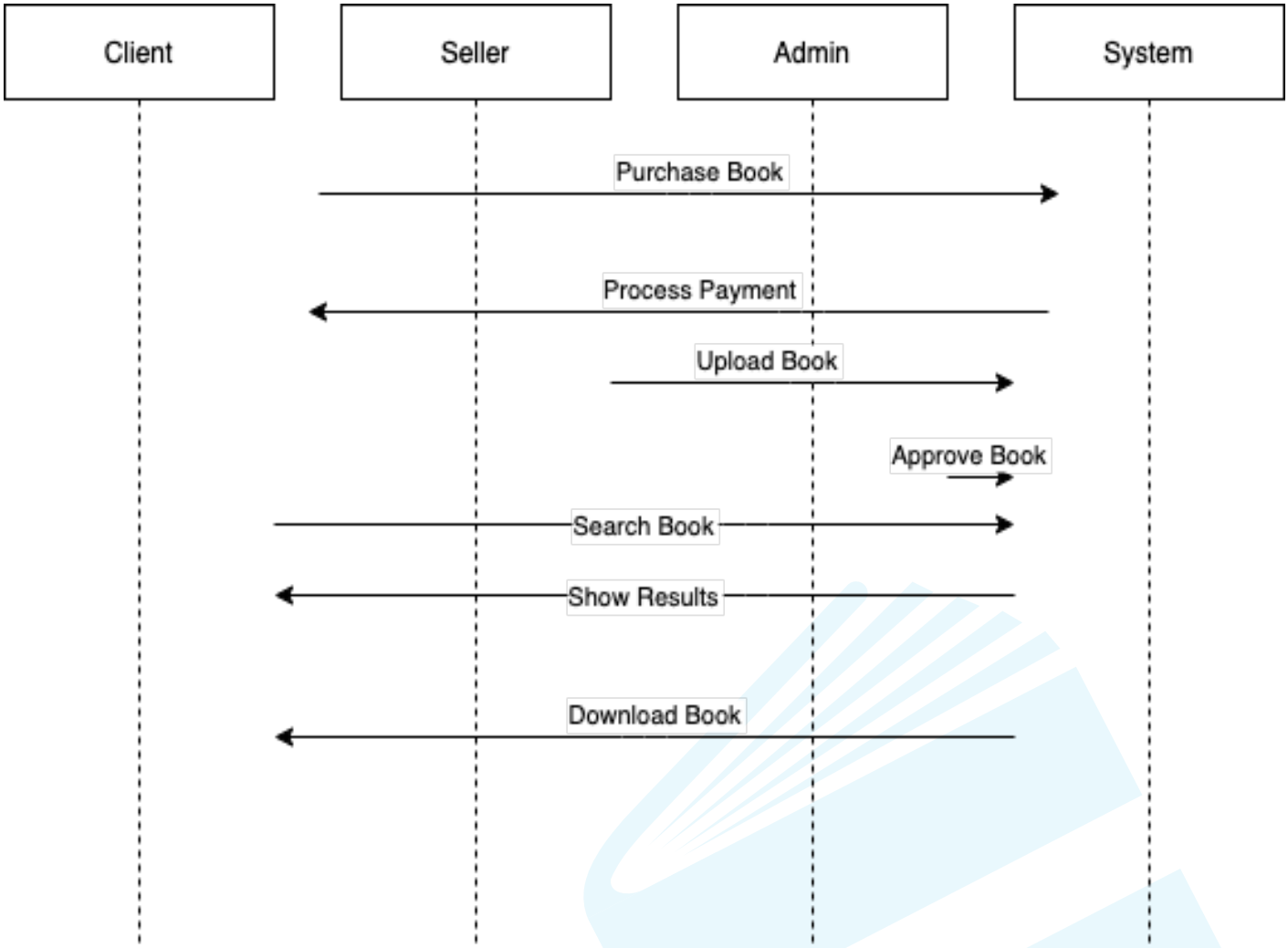
Processes:

- Client Registration & Login ->Updates Users Databas
- Browse & Select Books -> Retrieves from Books Database
- Make Payment Updates Orders Database
- Seller Registration & Login -> Updates Users Database
- Upload Book Details -> Updates Books Database
- Admin Management -> Manages users & books

Data Stores:

- Users Database – Stores user data.
- Books Database – Stores books.
- Orders Database – Stores purchases.

Sequence Diagram



1. User Registration & Login

- User visits the website.
- User selects Sign Up (if new) or Login (if existing).
- System verifies credentials (if login) or creates an account (if signup).
- System redirects to the homepage upon success.

2. Browsing & Searching Books

- User navigates to the Book Catalog.
- User searches for a book (optional).
- System fetches matching books from the Books Database.
- User views book details.

3. Adding Books to Cart & Checkout

- User clicks Add to Cart on a book.
- System updates the Cart with the selected book.
- User proceeds to Checkout.
- System displays the total amount and payment options.

4. Payment Processing

- User selects Payment Method (Credit Card, PayPal, etc.).
- System sends payment request to Payment Gateway.
- Payment Gateway processes transaction.
- System updates Order Status and notifies the user.

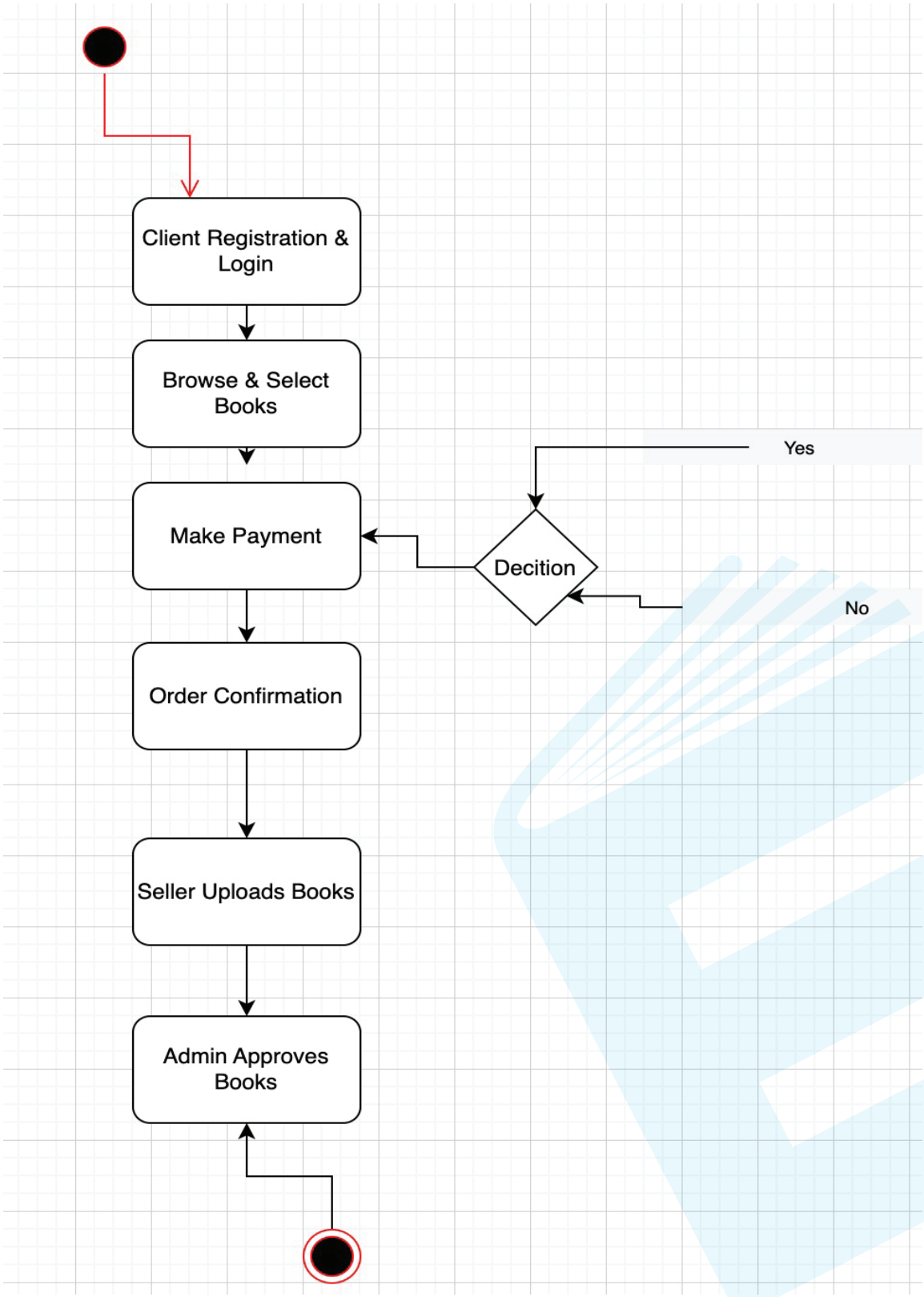
5. Downloading Purchased eBooks

- After a successful payment, the system provides a Download Link.
- User clicks on the link and downloads the eBook.
- System records the download in the Order History.

6. Reviewing & Rating a Book

- User navigates to Purchased Books.
- User selects Write a Review.

Activity Diagram



Activity Diagram Explaination

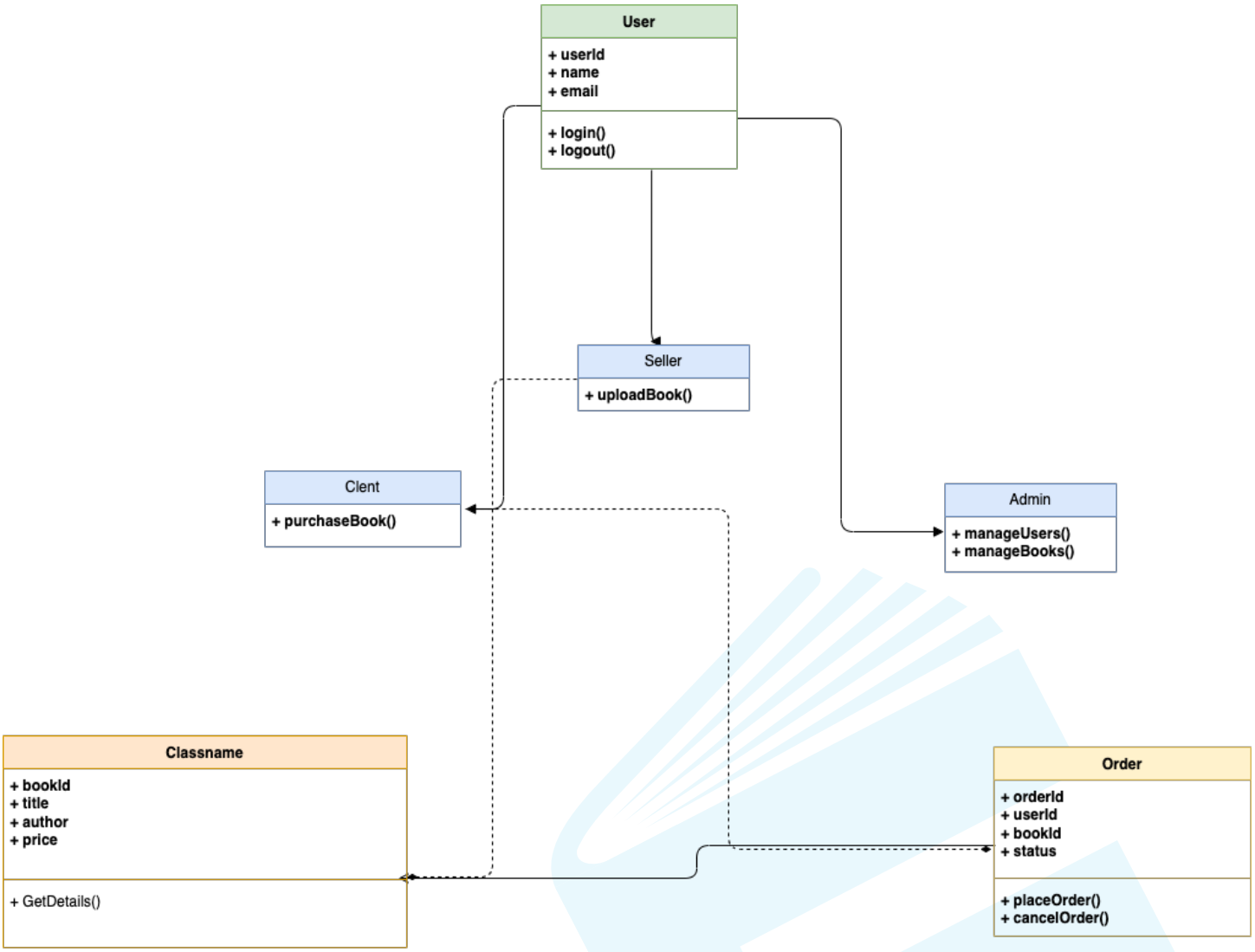
The Activity Diagram represents the workflow of user interactions in the eBook Store system, detailing the sequence of processes for clients, sellers, and administrators. The main actors include the Client (Buyer) who registers, browses books, and makes purchases, the Seller who registers and uploads books, and the Admin who manages users and book approvals.

The Client Workflow starts with registration or login. If successful, the client browses and selects books, proceeding to payment. If the payment is successful, the order is confirmed; otherwise, the client is prompted to retry. The process concludes with order confirmation and system updates.

The Seller Workflow begins with registration and login, followed by uploading book details such as title, description, and price. The Admin Workflow involves reviewing and approving or rejecting uploaded books. Approved books are listed for sale, while rejected books trigger a notification to the seller.

The Admin is responsible for managing users and books, ensuring smooth system operations. Key decision points in the process include checking login success, payment completion, and book approval. The Activity Diagram provides a structured overview of these interactions, ensuring a seamless workflow for all users in the eBook Store system.

Class Digram



Class Diagram Explanation for eBook Store

The Class Diagram represents the structure of the eBook Store system, defining its main classes, attributes, methods, and relationships.

1- User Class (Parent Class)
- Attributes: UserId, Name, Email, Password
- Methods: Register(), Login(), Logout()
- This class serves as a base for different user types (Client, Seller, Admin).

2- Client Class (Inherits from User)
- Attributes: OrderHistory, Wishlist
- Methods: BrowseBooks(), PlaceOrder(), MakePayment()
- Represents customers who can browse books, make purchases, and manage their wishlist.

3- Seller Class (Inherits from User)
- Attributes: BookList
- Methods: UploadBook(), ManageBooks()
- Represents users who can upload and manage books for sale.

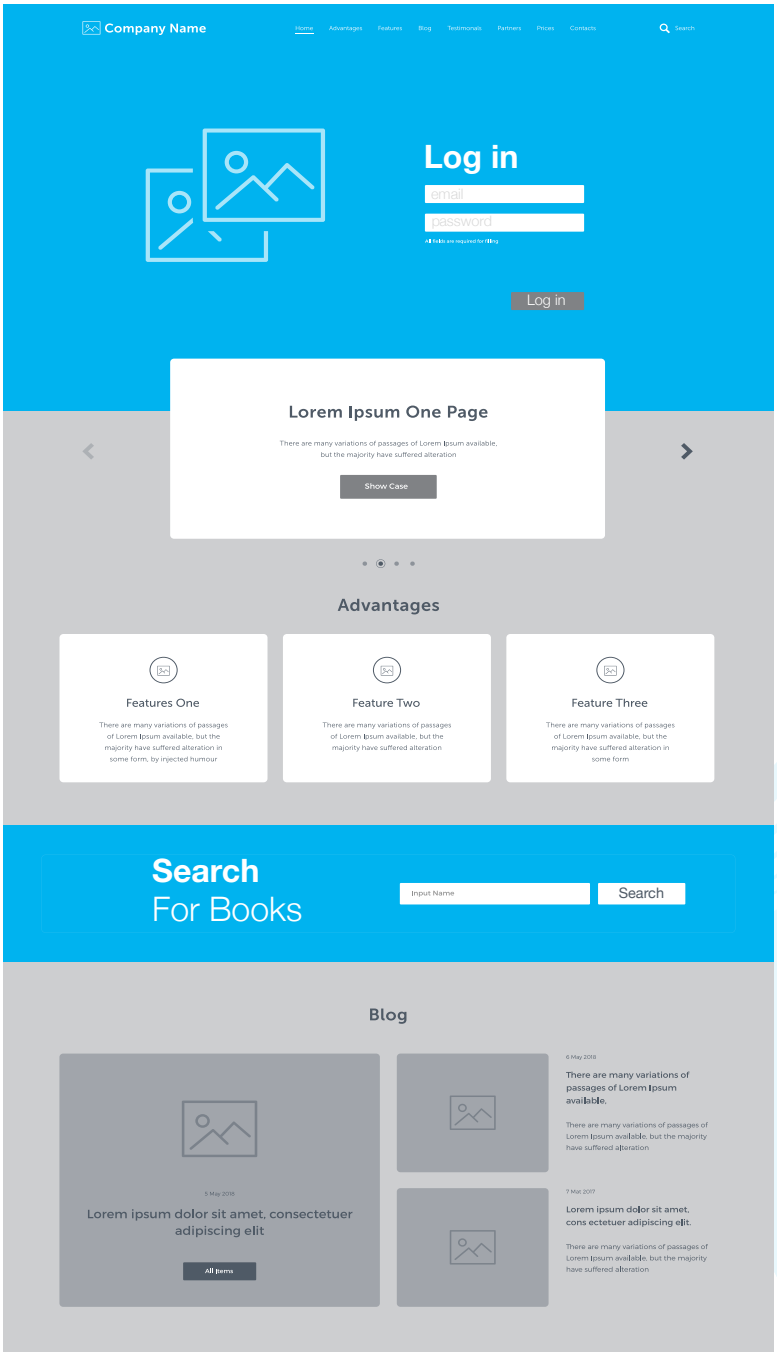
4- Admin Class (Inherits from User)
- Methods: ManageUsers(), ApproveBooks()
- Responsible for managing users and approving books for sale.

5- Order Class
- Attributes: OrderId, Date, Price, Status
- Methods: ProcessOrder(), CancelOrder()
- Represents a transaction made by a client, storing order details.

Relationships:
- User is a parent class for Client, Seller, and Admin.
- Client places an order, establishing a relationship with the Order class.
- Seller uploads books, linking it to book management functions.
- Admin manages users and approves books, ensuring smooth system operations.

UI/UX Design & Prototyping

Wireframes & Mockups



Mockups

UI/UX Guidelines

1. Login Page
- Simple Layout – Minimal design, store logo, and theme colors.
- Input Fields – Email, password, show/hide option.
- CTA Buttons – "Login", "Forgot Password?", "Register".
- UX: Autofocus on email, error messages, quick navigation.

2. Client Page (Purchased Books)
- Dashboard – Welcome message, book grid/list view.
- Features – Search, filter, view details, download option.
- Navigation – "Home", "My Books", "Profile", "Logout".
- UX: Fast loading, responsive design, easy access.

3. Seller Page (Upload Books)
- Upload Form – Title, description, category, price, file upload.
- Features – Drag & drop support, book management (edit/delete).
- Navigation – "Dashboard", "Upload Book", "Sales Reports", "Logout".
- UX: Progress bar, validation, success/error feedback.

4. Admin Page (User & Book Management)
- Dashboard – Overview of users, books, sales.
- User Management – Suspend/delete accounts.
- Book Approval – Approve/reject books, search & filter.
- Navigation – "Dashboard", "Users", "Books", "Reports", "Logout".
- UX: Bulk approvals, confirmation dialogs, restricted access.

System Deployment & Integration

1- Technology Stack

The eBook store system is developed using the following technologies:

Frontend:

- Blazor WebAssembly (WASM) – A modern web framework for building interactive UIs using C# and .NET.
- Razor Components – To create reusable UI components.
- Bootstrap/Tailwind CSS – For styling and responsiveness.

Backend:

- .NET (ASP.NET Core Web API) – To handle business logic and API communication.
- Entity Framework Core – ORM for database interactions.

Database:

- SQL Server – Relational database for storing user, book, and order data.

Authentication & Security:

- JWT Authentication – Secure login and access control.
- Role-Based Authorization – For different user roles (Client, Seller, Admin).

2- Deployment Diagram

The deployment diagram shows how software components are distributed across servers.

Deployment Overview:

Client-Side (Blazor WASM)

Runs in the user's browser, communicating with the backend via API calls.

Server-Side (.NET Backend)

Hosts the business logic and processes client requests.

Database (SQL Server)

Stores all book, user, and order data.

3- Component Diagram

The component diagram represents major system components and their dependencies.

Main Components:

Frontend (Blazor WASM)

Connects with the backend via API.

Backend (.NET API)

Handles authentication, book management, and order processing.

Database (SQL Server)

Stores data for books, users, and orders.

E-BOOK STORE

A Seamless Platform for Buying and Selling Digital Books

TEAMWORK

E-Book Store (CAI2_SWD5_S8)

Abdelrahman Ayman ELSherif

Nour Safwat Ataalla

Mohamed Ali Abo Helal

Adham Elkammar

GitHubLink

A Seamless Platform for Buying
and Selling Digital Books