**Booked**

**By Scarlet Warner**

Contents

[Problem Statement: 2](#_Toc185667451)

[Description 2](#_Toc185667452)

[Objectives of the system 3](#_Toc185667453)

[System requirements 3](#_Toc185667454)

[Typical Customers 4](#_Toc185667455)

[Project planning 4](#_Toc185667456)

[Development approach 5](#_Toc185667457)

[Development Plan 5](#_Toc185667458)

[Traceability Matrix 5](#_Toc185667459)

[System Architecture 6](#_Toc185667460)

[System Design 6](#_Toc185667461)

## Problem Statement:

As the popularity of recreational reading grows, many readers seek affordable ways to access a wide range of books. However, the cost of purchasing new recreational books can be prohibitive, especially for avid readers or those with limited budgets. Libraries are unable to provide for the demand, resulting in month long wait times. While there are second-hand marketplaces, they often cater more towards textbooks or specialized genres, leaving recreational readers with limited options. These platforms frequently lack a streamlined system for browsing, purchasing, and selling used recreational books, resulting in poor user experiences, unverified book conditions, and unreliable transactions. They also promote price gouging, allowing sellers to list new books for exorbitant prices, capitalizing on perceived scarcity.

This project aims to develop a dedicated second-hand book resale system specifically for recreational readers. The platform will provide a convenient, user-friendly space for readers to buy and sell pre-owned novels, non-fiction, and other recreational titles. By offering clear listings, book condition transparency, and a secure transaction process, this system will foster a vibrant community of book lovers while making recreational reading more accessible and affordable.

## Description

The platform is designed for avid readers who seek affordable recreational books. The system will connect individuals who want to sell their used books with those looking to purchase them at a lower cost. The platform will emphasize simplicity, transparency, and community, making it easier for recreational readers to buy and sell novels, non-fiction, and other literary genres in a secure environment. By creating a dedicated marketplace for second-hand recreational books, the system will foster a community of readers while promoting sustainable book consumption and reuse.

## Objectives of the system

* **Provide an Accessible Marketplace:** Develop an intuitive platform where users can easily list, browse, and purchase recreational books.
* **Facilitate Safe and Secure Transactions:** Offer integrated payment solutions and a rating/review system to ensure safe, reliable exchanges between buyers and sellers.
* **Promote Environmental Sustainability:** Encourage the reuse of books by making second-hand sales convenient and accessible, and supporting eco-friendly consumption habits.

## System requirements

**Functional Requirements:**

1. User Account Management:
   * Users can create accounts, update profiles, and manage personal information.
2. Book Listings:
   * Users can upload books based on ISBN, ensuring accurate edition awareness.
   * Photos of the specific edition are uploaded automatically.
3. Search and Filtering:
   * Buyers can search for books by title, author, genre, or price.
   * Filtering options for book condition and price range.
4. Transaction Management:
   * Integrated payment gateway for secure transactions (e.g., Stripe, PayPal).
   * Order tracking for both buyers and sellers.
5. Notifications:
   * Email or in-app notifications for new messages, book listings, or status updates on orders.
6. Recommendation System:
   * A system that suggests books.

**Non-Functional Requirements:**

1. Security:
   * Secure payment transactions using encryption and fraud protection.
   * User data protection and privacy compliance (GDPR, etc.).
2. Scalability:
   * The platform should be able to handle a growing user base and large volumes of book listings.
3. Usability:
   * A user-friendly interface accessible on both desktop and mobile devices, with clear navigation and minimal complexity.
4. Performance:
   * Fast load times and smooth browsing even with a large database of books.
5. Support and Maintenance:
   * Regular updates and customer support to resolve any issues users may encounter.

## Typical Customers

1. Avid Readers:

* Individuals who read frequently for leisure and are looking for more affordable ways to access books.

1. Casual Readers:

* Readers who pick up a few books a year but want to find deals on popular titles without paying full price.

1. Book Collectors:

* People interested in collecting rare or out-of-print books at lower prices.

1. Students:

* Though the platform is focused on recreational books, students looking for affordable literature or leisure reading material could be potential customers.

1. Eco-Conscious Consumers:

* Individuals who are environmentally conscious and prefer buying second-hand to reduce waste and support sustainability.

## Project planning

Web browser, GitHub, Visual Studio Code

## Development approach

HTML, CSS, C#, Identity framework, Entity framework

## Development Plan

Week 1-2: Initial planning & requirements gathering

Week 3-8: System Design

Week 5-8: Development

Week 9-12: Testing and debugging

Week 13: Deployment and final testing

## Traceability Matrix

| Requirement ID | Requirement Description | Functional Module | Verification Method |
| --- | --- | --- | --- |
| FR-001 | Users can create accounts, update profiles, and manage personal information. | User Account Management | Unit Testing, Integration Testing |
| FR-002 | Users can upload books | Book Listings | Unit Testing, UI Testing |
| FR-003 | Photos of the book’s condition must be sourced | Book Listings | Unit Testing, UI Testing |
| FR-004 | Buyers can search for books by title, author, genre, or price. | Search and Filtering | Unit Testing, UI Testing |
| FR-005 | Filtering options for book condition and price range. | Search and Filtering | Unit Testing, UI Testing |
| FR-010 | Book recommendations | Recommendation System | Unit Testing |
| NFR-002 | Compliance with data protection laws (e.g., GDPR). | Security and User Data Management | Security Testing |
| NFR-003 | Scalable to handle growing users and listings. | System Architecture | Load Testing |
| NFR-004 | User-friendly interface on desktop and mobile. | User Interface Specification | Usability Testing |
| NFR-005 | Fast load times and smooth browsing. | Performance Optimization | Performance Testing |

## System Architecture

The Booked application leverages the Blazor Server framework to provide a robust and scalable architecture for a web-based second-hand book marketplace. The architecture consists of the following components:

* 1. Frontend (User Interface):
* Built using Blazor Server, providing a rich interactive experience.
* Renders server-side Razor components for dynamic, fast-loading pages.
* Compatible with desktop and mobile devices, ensuring accessibility and usability.
  1. Backend (Server):
* Implemented using .NET Core for high performance and reliability.
* Handles business logic, data processing, and interaction with the database.
* Supports authentication and authorization using ASP.NET Identity.
  1. Database Layer:
* Uses Entity Framework Core for object-relational mapping.
* Relational databasestores user data, book listings, transactions, and reviews.
  1. Notification System:
* Sends real-time email or in-app notifications for new messages, listings, and order status updates.
  1. API Integration:
* Enables interaction with third-party APIs for external book databases for enhanced metadata.

## System Design

1. Modular Design:

* The system is divided into distinct, reusable modules:
* User Module: Manages user profiles, authentication, and roles.
* Book Listings Module: Handles adding, updating, and displaying book data.
* Notification Module: Manages alerts and updates for users.

1. Database Design:

* Tables:
  + Users: UserID, Name, Email, PasswordHash, Role, CreatedAt.
  + Books: BookID, Title, Author, Price, SellerID, ListingDate.
  + Notifications: NotificationID, UserID, Type, Content, Status, CreatedAt.
* Relationships:
* One-to-many: Users to Books (one user can list multiple books).
* Data Flow:
* User requests (e.g., searching for books) are processed by the frontend and relayed to the backend.
* The backend retrieves or updates data from the database using Entity Framework Core.
* Results are sent back to the frontend for display to the user.
* Asynchronous notifications are triggered for relevant events.

1. Security Measures:

* Password hashing and storage using ASP.NET Identity.
* HTTPS for all communications to ensure data encryption.
* Role-based access control for administrative actions.
* Compliance with GDPR for data privacy and protection.

1. Scalability:

* Horizontal scaling by adding more application servers behind a load balancer.
* Database indexing and caching strategies to improve performance.

1. Performance Optimization:

* Lazy loading and eager fetching in Entity Framework to manage database queries efficiently.
* Frontend optimizations, such as pre-rendering components and using lightweight assets.