
Software Requirements Specification

for
<Book Nest>

Version <3.3>

Prepared by

Group Name: <SE3. Se1>

<Habiba Mamdouh>	<SE3>	<habibamamdouh91@gmail.com>
<Omar Mohamed>	<SE3>	<omarhamoda20@gmail.com>
<Bassel Adel>	<SE3>	<bassel.adel135@gmail.com>
<Youssef Mysara>	<SE1>	<e-mail>
<Abdelrahman Gamal>	<SE3>	<e-mail>

Instructor: <*Dalia Magdy*>

Course: <Software Maintenance>

Lab Section: <Se3/Se1>

Teaching Assistant: <*Hesham Ramadan*>

Date: <12/6/2025>

CONTENTS	Error! Bookmark not defined.
REVISIONS	2
1 INTRODUCTION	Error! Bookmark not defined.
1.1 DOCUMENT PURPOSE	Error! Bookmark not defined.
1.2 PRODUCT SCOPE	Error! Bookmark not defined.
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW.....	Error! Bookmark not defined.
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS.....	Error! Bookmark not defined.
1.5 DOCUMENT CONVENTIONS	2
1.6 REFERENCES AND ACKNOWLEDGMENTS.....	Error! Bookmark not defined.
2 OVERALL DESCRIPTION	Error! Bookmark not defined.
2.1 PRODUCT OVERVIEW	Error! Bookmark not defined.
2.2 PRODUCT FUNCTIONALITY	Error! Bookmark not defined.
2.3 DESIGN AND IMPLEMENTATION CONSTRAINTS	Error! Bookmark not defined.
2.4 ASSUMPTIONS AND DEPENDENCIES	Error! Bookmark not defined.
3 SPECIFIC REQUIREMENTS	Error! Bookmark not defined.
3.1 EXTERNAL INTERFACE REQUIREMENTS	6
3.2 FUNCTIONAL REQUIREMENTS	6
3.3 USE CASE MODEL.....	7
4 OTHER NON-FUNCTIONAL REQUIREMENTS.....	Error! Bookmark not defined.
4.1 PERFORMANCE REQUIREMENTS.....	13
4.2 SAFETY AND SECURITY REQUIREMENTS.....	Error! Bookmark not defined.
4.3 SOFTWARE QUALITY ATTRIBUTES	Error! Bookmark not defined.
5 OTHER REQUIREMENTS.....	Error! Bookmark not defined.
APPENDIX A – DATA DICTIONARY	19
APPENDIX B - GROUP LOG	20

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

1 Introduction

BookNest is a standalone, responsive e-commerce web application designed specifically for online book sales, providing customers with an intuitive platform to browse extensive catalogs, manage personalized shopping carts, and complete secure purchases without external API dependencies.

1.1 Document Purpose

The purpose of BookNest is to provide a complete online bookstore platform where users can browse, search, and purchase books easily from anywhere. The application connects readers with a wide variety of books while offering a smooth shopping experience similar to modern e-commerce sites.

1.2 Product Scope

BookNest provides a complete online platform for purchasing books. The system includes functionality for both customers and administrators, ensuring smooth management of book sales and bookstore operations.

1.3 Intended Audience and Document Overview

- ◊ *Intended Audience*
This document is intended for the following stakeholders:

Audience	Purpose of Use
Project Manager	Understand project scope, timeline, and deliverables.
Developers	Implement system features, architecture, and integration based on the documented requirements.
UI/UX Designers	Design user-friendly interfaces aligned with system features and user needs.
Quality Assurance (QA) Team	Create and execute test cases to ensure the system meets requirements.
System Administrators	Understand deployment, configuration, and maintenance procedures.
Client / Bookstore Owner	Validate that the system satisfies business goals and bookstore operations.

1.4 Definitions, Acronyms and Abbreviations

This section defines key terms, acronyms, and abbreviations used throughout the BookNest SRS to ensure consistent understanding among stakeholders.

- ◊ *Definitions:*
 - *BookNest: The e-commerce web application for browsing, purchasing, and managing books, hosted at <https://github.com/basseladel136/BookNest>.*

- User: Any individual interacting with the system, including customers who browse and buy books or administrators managing inventory.
 - Shopping Cart: Temporary storage for selected books before checkout completion.
 - Order: Confirmed purchase transaction containing books, customer details, and payment information.
- ◊ Acronyms and Abbreviations:
- SRS: Software Requirements Specification - Document outlining functional and non-functional requirements.
 - UI: User Interface - Visual elements for user interaction, such as product pages and login forms.
 - API: Application Programming Interface - Mechanisms for backend-frontend communication in BookNest.
 - DB: Database - Storage system for books, users, and orders.

1.5 Document Conventions

This section describes the typographical and structural conventions used in the BookNest SRS for clarity and consistency across all sections.

- ◊ Typographical Conventions:
- **Bold text:** Emphasizes key terms like "BookNest" or critical requirements (e.g., must support 1000 concurrent users).
 - *Italic text:* Denotes user actions or system responses (e.g., User clicks "Add to Cart").
 - Monospace text: Represents code snippets, menu options, or data fields (e.g., `user_id`, `POST /api/orders`).
 - Bulleted lists (-) for features; numbered lists (1., 2.) for sequences like checkout steps.
- ◊ Structural Conventions:
- Section numbering: Uses 1., 1.1, 1.1.1 format (e.g., 3.2.1 User Registration).
 - Tables: Present comparisons or specifications with cited data in cells (e.g., feature matrices).
 - Use Cases: Follow Actor-Preconditions-Steps-Postconditions format for functional requirements.
 - All requirements are traceable with unique IDs (e.g., FR-001 for Functional Requirement 1).

1.6 References and Acknowledgments

This section lists all documents, standards, and sources referenced in the BookNest SRS, including the source code repository and e-commerce templates.

◊ References

ID	Document Title	Source/Location
R1	BookNest GitHub Repository	https://github.com/basseladel136/BookNest_github
R2	IEEE Std 830-1998 Software Requirements Specifications	IEEE Standards Association dal
R3	E-Commerce SRS Template	Studocu E-Commerce Examples studocu

◊ **Acknowledgments**

Recognition goes to the BookNest development team at basseladel136 for the open-source codebase, IEEE for SRS standards, and e-commerce reference templates that informed this specification. Stakeholder contributions from project owners and reviewers ensured comprehensive coverage of book sales functionality.

2 Overall Description

2.1 Product Overview

BookNest represents a new, self-contained e-commerce web application specifically designed for online book sales to serve book enthusiasts. Unlike traditional brick-and-mortar bookstores or generic e-commerce sites, BookNest focuses exclusively on books, providing a specialized environment for browsing extensive catalogs, managing personalized shopping carts, and completing secure purchases tailored to readers' preferences for genres, authors, and formats. As a complete product rather than a component of a larger system, it integrates all necessary frontend user interfaces, backend logic, and database operations without external subsystem dependencies, though it interfaces with payment gateways and email services for transactions and notifications.

2.2 Product Functionality

BookNest provides core e-commerce capabilities focused on book sales, enabling seamless user interactions from discovery to purchase completion.

Major functions include:

- *User Authentication: Registration, login, logout, and profile management for personalized shopping experiences.*
- *Product Catalog Management: Browse books by category/author, advanced search with filters, and detailed product views with descriptions and pricing.*
- *Shopping Cart Operations: Add/remove books, adjust quantities, view cart summary, and persist across sessions.*
- *Checkout and Payment Processing: Secure order placement, address entry, payment gateway integration, and order confirmation.*
- *Order Management: Track order status, view purchase history, and receive delivery notifications.*
- *Admin Functions: Manage book inventory, process orders, oversee users, and generate sales reports.*
- *Search and Recommendations: Keyword/genre search and basic personalized book suggestions.*

2.3 Design and Implementation Constraints

BookNest development must adhere to specific technical and architectural constraints to ensure compatibility with the existing GitHub repository codebase and maintainable e-commerce standards.

◊ *Technology Stack Constraints:*

- *Frontend: Blade template engine (HTML), Tailwind (CSS), JavaScript frameworks evident in repository assets, ensuring responsive design across browsers.*
- *Backend: Laravel. PHP.*
- *Database: Relational DB (MySQL) for books, users, and transactions, with schema matching repository data models.*

2.4 Assumptions and Dependencies

BookNest development relies on several assumptions about external factors and infrastructure that, if invalidated, could impact functionality, timeline, or costs.

◊ *Key Assumptions:*

- *Target browsers (Chrome, Firefox, Safari latest versions) support required JavaScript/CSS features without polyfills.*

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

BookNest features a responsive web-based user interface optimized for desktop, tablet, and mobile devices, delivering intuitive navigation for book discovery and purchases through standard browser interactions.

◊ *Primary User Interface Characteristics:*

- *Homepage: Hero banner showcasing featured books, category grids with genre icons, search bar, and newsletter signup form for immediate engagement.*
- *Navigation: Persistent top menu with links to Home, Categories, Bestsellers, Cart, Profile; hamburger menu on mobile for compact access.*
- *Product Pages: Book covers, titles, author details, prices, "Add to Cart" buttons, reviews section, and related recommendations.*
- *Shopping Cart/Checkout: Summary table with quantity controls, subtotal calculations, shipping form, and secure payment options.*

3.1.2 Hardware Interfaces

BookNest operates as a cloud-based web application with no direct hardware dependencies, relying instead on standard client-server interactions through internet-connected devices.

◊ *Supported Hardware Platforms:*

- *Client Devices: Desktop computers (Windows, macOS, Linux), laptops, tablets (iPad, Android), smartphones (iOS 15+, Android 10+) with modern browsers (Chrome 100+, Firefox 100+, Safari 15+).*
- *Servers: Cloud virtual machines (AWS EC2, Google Cloud Compute, Azure VMs) with x86_64 architecture, minimum 4GB RAM, SSD storage for database and assets.*
- *Network Hardware: Standard routers, firewalls, load balancers supporting HTTPS/TCP port 443; CDN edge servers for static content delivery.*

3.1.3 Software Interfaces

BookNest operates as a standalone web application, relying entirely on internal components for all functionality including payments, notifications, and data management.

◊ *Internal Software Interfaces Only:*

- *Frontend-Backend: Direct server-side processing via form submissions (POST/GET requests) without REST APIs; session-based state management using cookies/server sessions.*
- *Database Connectivity: Direct SQL queries or embedded database access (e.g., SQLite/MySQL) from server-side scripts for books, users, orders without external services.*
- *Payment Processing: Simulated/internal mock payment system or basic form validation without third-party gateways.*
- *Email Notifications: Server-side SMTP using built-in mail libraries without external services like SendGrid.*

3.2 Functional Requirements

BookNest functional requirements detail specific system behaviors for user authentication, book management, shopping, and administration, traceable to Section

1.1 Product Functionality.

1.1.1 F1: User Authentication

The system shall allow users to register with email, password, and name, storing credentials securely in the database without external APIs.

The system shall authenticate users via email/password login, maintaining sessions with cookies for 30 minutes of inactivity.

The system shall enable password reset via email link sent through server SMTP.

1.1.2 F2: Product Catalog

The system shall display all books from database with title, author, price, cover image, and category filters (fiction, non-fiction, etc.).

The system shall provide keyword search across book titles, authors, and descriptions with exact/partial matching.

The system shall show individual book details including description, ISBN, stock availability, and reviews section.

1.1.3 F3: Shopping Cart

The system shall allow authenticated users to add books to cart with quantity selection, persisting via session storage.

The system shall enable cart modification (add/remove items, update quantities) with real-time subtotal calculation.

The system shall display cart summary with total price, tax calculation, and "Proceed to Checkout" button.

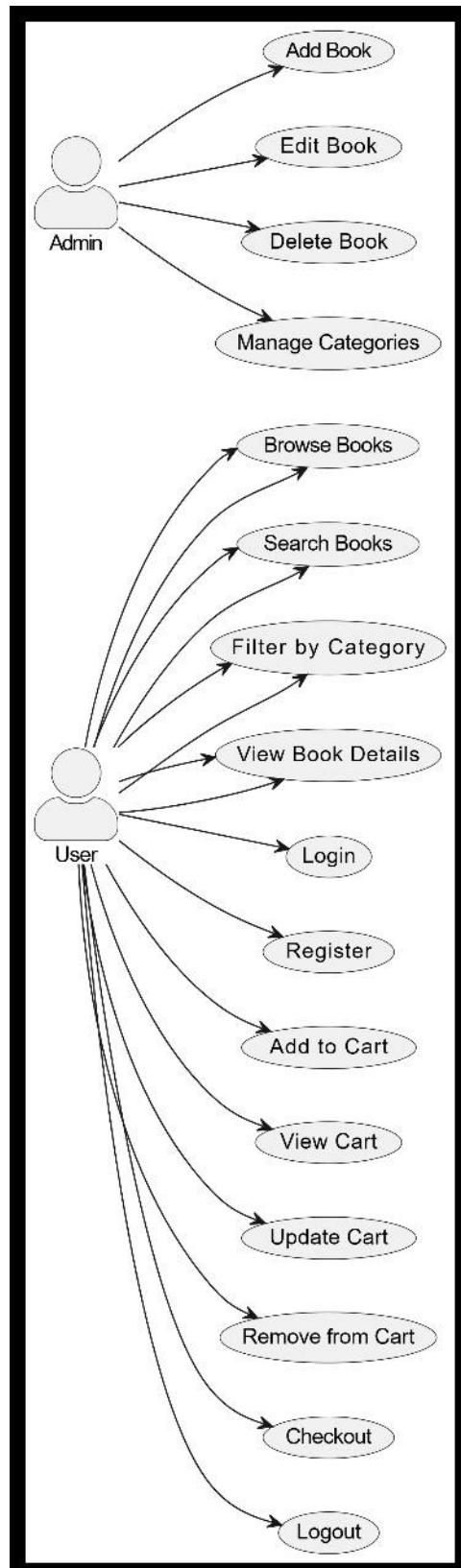
1.1.4 F4: Checkout & Orders

The system shall collect shipping/billing address during checkout and generate unique order ID upon submission.

The system shall process orders internally (no external APIs) marking status as "Pending", "Shipped", "Delivered".

The system shall provide order history view for users showing status, date, and items purchased.

3.3 Use Case Model



3.3.1 Use Case #1 (use case name and unique identifier – e.g. U1)

Use Case Specifications for Bookstore System

Below are the specifications for all use cases shown in the diagram. Each use case includes: Author, Purpose, Requirements Traceability, Priority, Preconditions, Postconditions, Actors, Extends, Basic Flow, Alternative Flow, Exceptions, Includes, and Notes/Issues.

1. Add Book

Author: Bassel adel

Purpose: Allow admin to add new books to the system.

Requirements Traceability: R1, R2

Priority: High

Preconditions: Admin must be logged in.

Postconditions: New book is stored in the database.

Actors: Admin

Extends: None

Basic Flow:

1. Admin selects “Add Book”.

2. System displays form.

3. Admin enters details.

4. System saves book.

Alternative Flow: Missing optional fields → system requests required data.

Exceptions: Database error.

Includes: Manage Categories

Notes: Validate duplicate ISBN.

2. Edit Book

Author: Bassel adel

Purpose: Admin updates existing book details.

Requirements Traceability: R3

Priority: High

Preconditions: Admin logged in; book exists.

Postconditions: Book data updated.

Actors: Admin

Extends: None

Basic Flow:

1. Admin chooses a book.

2. Edits fields.

3. Saves changes.

Alternative Flow: None

Exceptions: Invalid book ID.

Includes: None

Notes: Track edit history.

3. Delete Book

Author: Bassel adel

Purpose: Remove book from catalog.

Requirements Traceability: R4

Priority: Medium

Preconditions: Admin logged in.

Postconditions: Book removed.

Actors: Admin

Extends: None

Basic Flow:

1. Select book → delete.

2. System confirms.

3. System deletes.

Alternative Flow: Cancel.

Exceptions: Book linked to active orders.

Includes: None

Notes: Soft delete recommended.

4. Manage Categories

Author: Bassel adel

Purpose: Manage book categories.

Requirements Traceability: R5

Priority: Medium

Preconditions: Admin logged in.

Postconditions: Category list updated.

Actors: Admin

Extends: None

Basic Flow: Add/Edit/Delete categories.

Exceptions: Category in use.

5. Browse Books

Author: Habiba mamdouh

Purpose: User views all books.

Requirements Traceability: R6

Priority: High

Preconditions: None

Postconditions: None

Actors: User

Basic Flow: System displays list of books.

6. Search Books

*Author: Omar mohamed
Purpose: User searches by keyword.
Requirements Traceability: R7
Priority: High
Preconditions: None
Postconditions: Results shown.
Exceptions: No results.*

7. Filter by Category

*Author: Youssef maysara
Purpose: Filter books.
Requirements Traceability: R8
Priority: Medium
Preconditions: Categories defined.
Postconditions: Filtered list.*

8. View Book Details

*Author: Abdelrahman gamal
Purpose: Display detailed info.
Requirements Traceability: R9
Priority: High
Actors: User
Postconditions: None.*

9. Login

*Author: Habiba mamdouh
Purpose: Authenticate user.
Requirements Traceability: R10
Priority: High
Preconditions: User registered.
Postconditions: User logged in.
Exceptions: Wrong password.*

10. Register

*Author: Habiba mAMDouH
Purpose: Create account.
Requirements Traceability: R11
Priority: High
Postconditions: New user created.
Exceptions: Duplicate email.*

11. Add to Cart

*Author: Habiba mAMDouH
Purpose: Add selected book to cart.
Requirements Traceability: R12
Priority: High
Preconditions: User logged in.
Postconditions: Book added.
Exceptions: Out of stock.*

12. View Cart

*Author: Habiba mAMDouH
Purpose: Show cart contents.
Requirements Traceability: R13
Priority: High*

13. Update Cart

*Author: Habiba mAMDouH
Purpose: Modify quantities.
Requirements Traceability: R14
Priority: Medium
Exceptions: Quantity > stock.*

14. Remove from Cart

*Author: Habiba mAMDouH
Purpose: Remove items.
Requirements Traceability: R15
Priority: Medium*

15. Checkout

*Author: Habiba mamdouh
Purpose: Complete purchase.
Requirements Traceability: R16
Priority: High
Preconditions: Cart not empty.
Postconditions: Order created.
Exceptions: Payment failure.*

16. Logout

*Author: Habiba mamdouh
Purpose: End session.
Requirements Traceability: R17
Priority: Medium...*

4 Other Non-functional Requirements

4.1 Performance Requirements

< BookNest must deliver responsive e-commerce performance to ensure positive user experience during browsing and purchasing, with specific metrics tied to functional requirements.

P1:PageLoadPerformance

All pages (homepage, product detail, cart) shall load in ≤3 seconds for 95% of users under normal conditions (100 concurrent users) to maintain engagement and reduce cart abandonment.

P2:SearchResponseTime

Book search queries shall return results in ≤1 second for catalogs up to 10,000 books, supporting real-time filtering by category/price.

P3:CheckoutProcessing

Order submission shall complete in ≤5 seconds (including validation and database writes) without external APIs, displaying confirmation immediately.

P4:ConcurrentUserCapacity

System shall support 500 concurrent users with <5% error rate during peak hours (book launches/sales), using server-side session management.

P5:CartOperations

Add/remove/update cart items shall reflect changes in ≤500ms via session storage for seamless shopping experience.

4.2 Safety and Security Requirements

BookNest implements security measures to protect user data, prevent unauthorized access, and ensure safe e-commerce transactions without external API dependencies.

S1:AuthenticationSecurity

The system shall hash all passwords using bcrypt (minimum 12 rounds) before database storage and enforce login session timeouts after 30 minutes of inactivity. The system shall implement CSRF protection tokens on all forms (registration, login, checkout) to prevent cross-site request forgery attacks.

S2:DataProtection

The system shall encrypt all sensitive data (user PII, order details) in transit using HTTPS/TLS 1.3 and store session data server-side only.

The system shall sanitize all user inputs (search, comments, addresses) using prepared statements to prevent SQL injection and XSS attacks.

S3:TransactionSecurity

The system shall validate all checkout inputs (quantities, addresses) server-side and log all order attempts for audit trails.

The system shall restrict admin functions to authenticated admin accounts with separate privilege levels.

S4:PrivacyCompliance

The system shall provide user data export/deletion capabilities per GDPR/CCPA requirements and display clear privacy policy linking to data practices.

S5:SessionManagement

The system shall invalidate sessions on logout/password change and limit login attempts (5 failures = 15-minute lockout) to prevent brute force attacks.

4.3 Software Quality Attributes

BookNest prioritizes usability and maintainability to ensure intuitive user experiences and sustainable development, with specific measurable criteria for verification.

1.1.1 Usability

U1: System shall achieve 90% task completion rate for first-time users on core flows (registration → search → cart → checkout) within 3 attempts, measured via usability testing with 20 participants.

U2: All interactive elements shall follow WCAG 2.1 AA standards with ≥95% keyboard navigation coverage and color contrast ratios ≥4.5:1 for readability.

Achieved via: Consistent navigation patterns, form validation feedback, and progressive disclosure of checkout steps.

1.1.2 Maintainability

M1: Codebase shall maintain ≤10% cyclomatic complexity per function and ≥80% test coverage for business logic using unit/integration tests.

M2: Database schema changes shall support zero-downtime migrations via versioning and backward-compatible ALTER statements.

Achieved via: Modular MVC architecture, COMET design patterns, and UML documentation for all major components.

1.1.3 Reliability

R1: System shall achieve 99.9% uptime over 30-day periods, handling server restarts without data loss through database transaction logging.

R2: All user sessions shall persist cart data across server restarts using database-backed session storage instead of memory-only sessions.

Achieved via: ACID-compliant transactions and automated daily database backups.

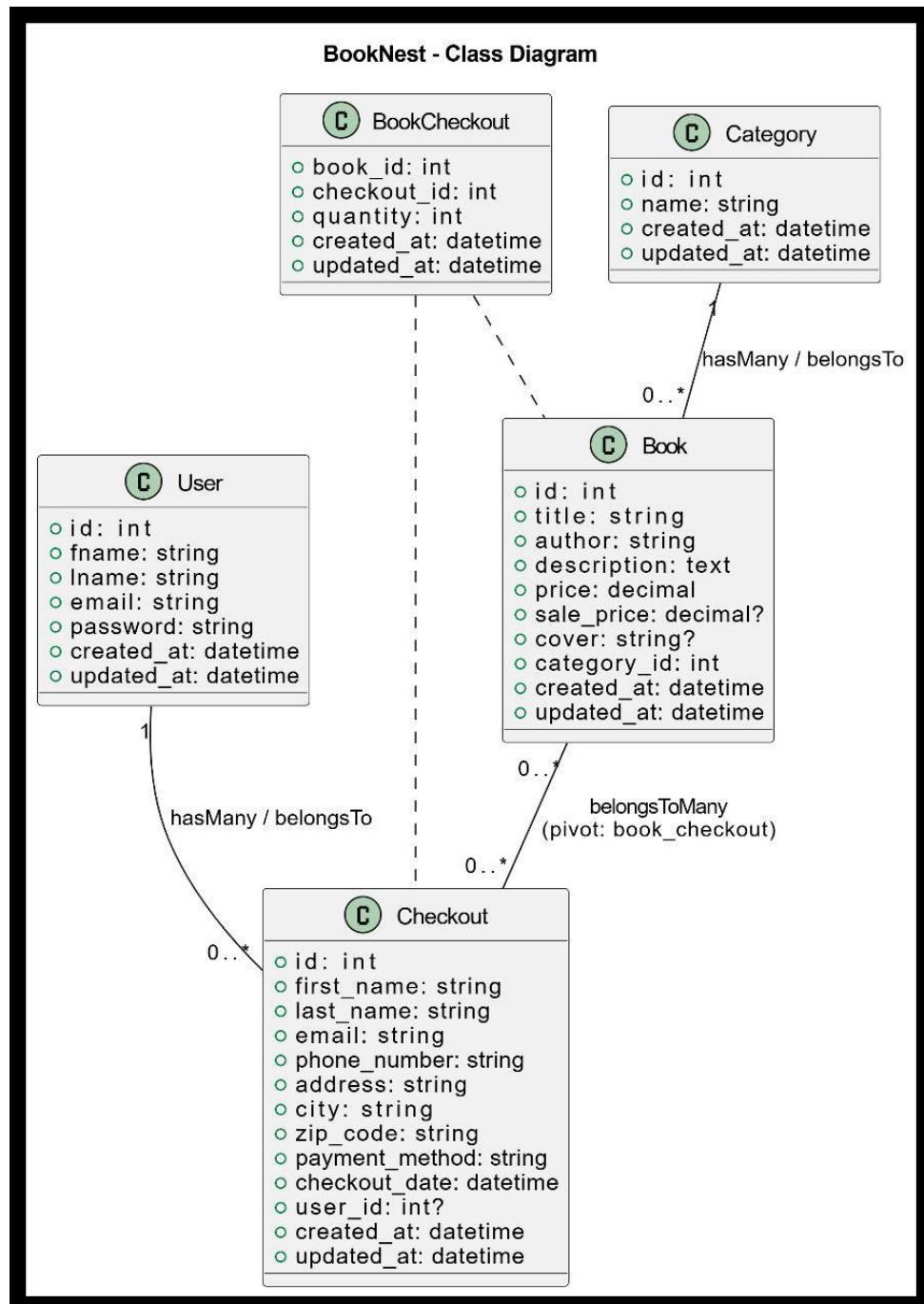
1.1.4 Portability

P1: Application shall deploy identically across Linux distributions (Ubuntu 20.04+, CentOS 8+) and Node.js versions 18.x-20.x without code modification.

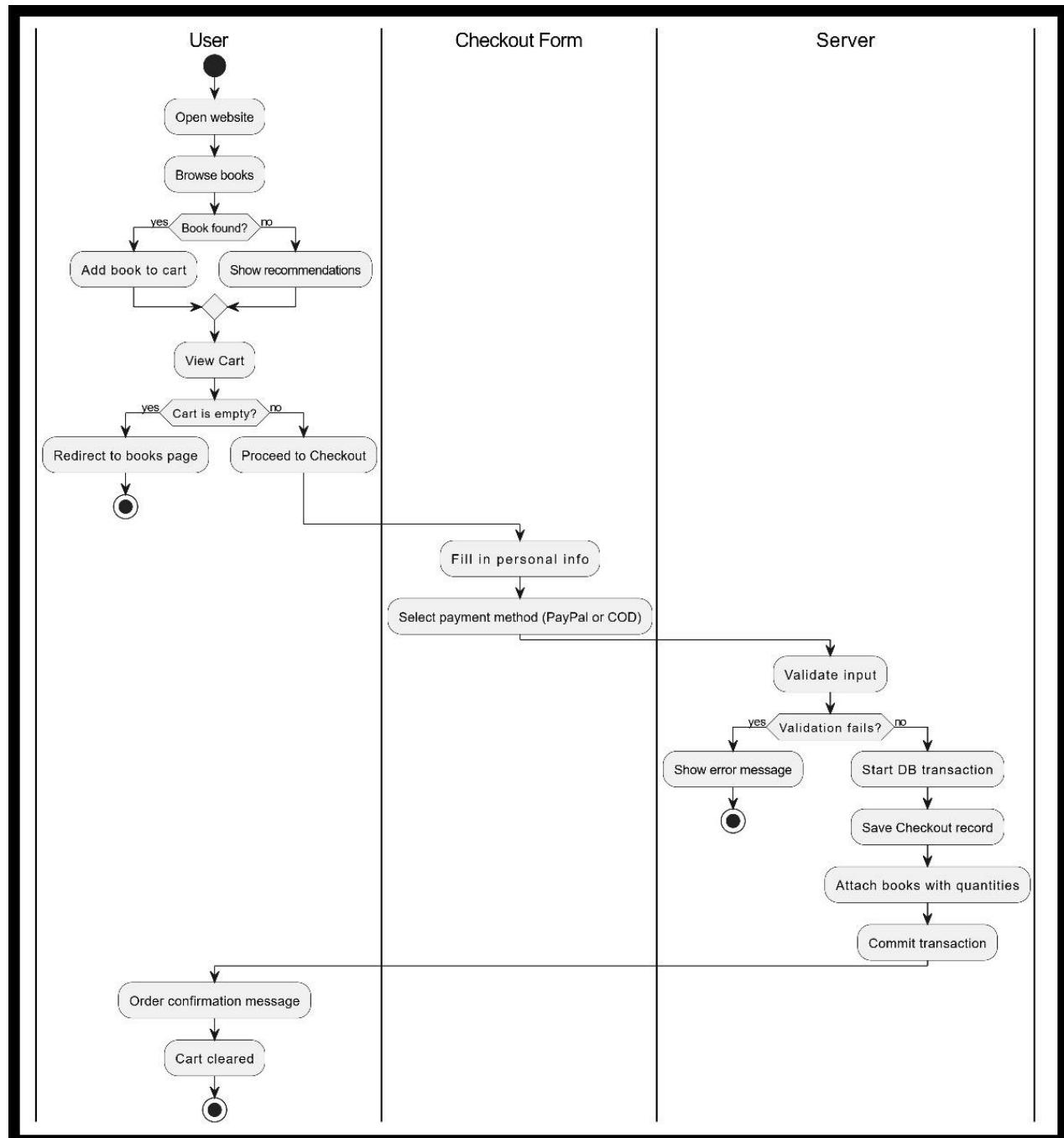
Achieved via: Docker containerization with multi-stage builds and environment-agnostic configuration files.

5 Other Requirements

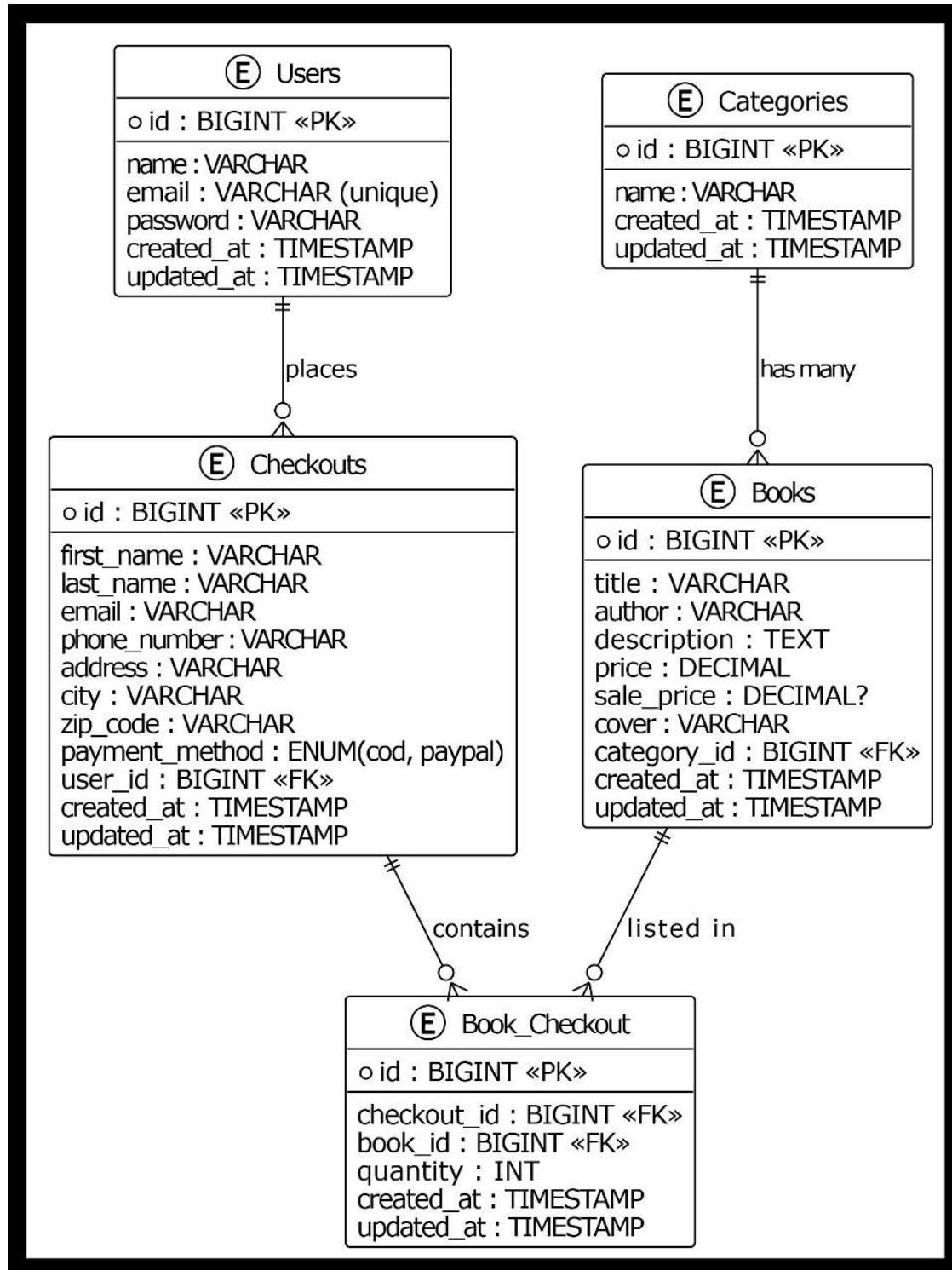
Class Diagram:



Activity Diagram:



Entity Relationship Diagram(ERD):



Appendix A – Data Dictionary

Data Element	Type	Description	Range/Form at	Related Requirements	Operations
user_id	Integer	Unique system-generated user identifier	1 - 999999	F1.1, S1	PK, Auto-increment github
email	VARCHAR(255)	User registration/login email	Valid email format	F1.1, S2	UNIQUE, Indexed
password_hash	VARCHAR(255)	bcrypt-hashed password	bcrypt \$2b\$12\$...	S1	Read-only
user_role	ENUM	User type	'customer', 'admin'	F5, S3	Default: 'customer'
book_id	Integer	Unique book identifier	1 - 999999	F2, DB1	PK
title	VARCHAR(255)	Book title	1-255 chars	F2.2	Indexed
price	DECIMAL(10, 2)	Book price in USD	0.01 - 999.99	F2.1, F3	Indexed
quantity	Integer	Available inventory	0 - 99999	F2.3, F5	Update atomic
checkout_id	Integer	Unique order identifier	1 - 999999	F4, P3	PK
session_token	VARCHAR(128)	User session identifier	SHA-256 hash	S5	Expires 30min
search_query	VARCHAR(255)	User search input	1-255 chars	F2.2, P2	Full-text indexed scribd

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to produce this document>