



# BOOKBAZAAR

## Requirements Document

Viktor Pavlov  
v.pavlov@student.fontys.nl

## Table of Contents

<b>1. Introduction .....</b>	<b>2</b>
Document Purpose.....	2
MoSCoW Definition.....	2
<b>2. Functional Requirements .....</b>	<b>3</b>
Functional Requirements Definition .....	3
Functional Requirements List.....	3
Functional Requirements Prioritization .....	6
<b>3. Non-Functional Requirements .....</b>	<b>7</b>
Non-Functional Requirements Definition .....	7
Non-Functional Requirements List.....	7
Non-Functional Requirements Prioritization .....	10

# 1. Introduction

## Document Purpose

This document serves as requirements specification for BookBazaar, an e-commerce web application designed to enable users to buy books online. The platform functions as a digital marketplace for buyers, providing seamless transaction experience through modern web technologies and industry best practices.

This requirements document establishes the foundation for the BookBazaar development project by defining both functional and non-functional requirements that must be implemented to ensure successful delivery of a robust, scalable, and secure e-commerce platform.

## MoSCoW Definition

The requirements in this document are prioritized using the MoSCoW prioritization method, for categorizing requirements based on their importance to project success. MoSCoW is an acronym that represents four priority categories:

- **Must Have (M):** Critical requirements that are non-negotiable and essential for project success. The system cannot be considered functional or launched without these features.
- **Should Have (S):** Important requirements that add significant value to the system but are not critical for initial launch. These features enhance user experience and system capabilities.
- **Could Have (C):** Desirable features that would be beneficial if time and resources permit. These requirements represent nice-to-have functionality that can improve the system but are not essential.
- **Won't Have (W):** Requirements that are explicitly acknowledged but will not be implemented in the current project phase. These may be considered for future development cycles.

## 2. Functional Requirements

### Functional Requirements Definition

Functional requirements are the essential characteristics that the system should deliver that the end-user specifically demands. These requirements define what the system should do by specifying specific behaviors, functions, and services that the BookBazaar platform must provide to satisfy business objectives and user needs.

### Functional Requirements List

#### FR-001: User Registration and Authentication

**Description:** The system must provide secure user registration and login functionality supporting both customer and administrator accounts. Users shall be able to create accounts with email verification, securely authenticate using username/password combinations, and maintain persistent sessions across browser sessions.

##### *Acceptance Criteria:*

- Users can register with email, username, and password
- Email verification required for account activation
- Secure login with BCrypt password hashing
- JWT-based session management
- Password reset functionality via email
- Role-based access control distinguishing customers from administrators

#### FR-002: Book Catalog Management

**Description:** The system must provide comprehensive book catalog functionality enabling administrators to manage book inventory and users to browse and search the available collection. The catalog shall support detailed book metadata, inventory tracking, and advanced search capabilities.

##### *Acceptance Criteria:*

- Display books with title, author, description, price, and availability
- Search functionality by title, author, category, and keywords
- Pagination for large result sets
- Category-based browsing
- Inventory level tracking and display
- Book metadata management (ISBN, publication date, genre)

### FR-003: Shopping Cart and Checkout

**Description:** The system must provide shopping cart functionality allowing users to select books, manage quantities, and complete purchases through a secure checkout process integrated with payment processing services.

#### *Acceptance Criteria:*

- Add books to the shopping cart with quantity selection
- Modify cart contents (update quantities, remove items)
- Persistent cart storage across user sessions
- Order total calculation including taxes and shipping
- Secure checkout workflow with payment processing
- Order confirmation and receipt generation

### FR-004: Order Processing and Management

**Description:** The system must handle the complete order lifecycle including order creation, payment processing, inventory updates, and order status tracking for both customers and administrators.

#### *Acceptance Criteria:*

- Order creation upon successful checkout
- Integration with payment gateway for transaction processing
- Automatic inventory reduction upon successful payment
- Order status tracking (pending, confirmed, shipped, delivered)
- Order history access for customers
- Administrative order management interface

### FR-005: Notification System

**Description:** The system should provide automated notification capabilities to inform users of important events including order confirmations, shipping updates, and back-in-stock alerts for requested books.

#### *Acceptance Criteria:*

- Email notifications for order confirmations
- Back-in-stock alert subscriptions
- Automated email delivery for order status changes
- User preference management for notification types
- Integration with third-party email service provider

#### **FR-006: Book Request Feature**

**Description:** The system shall allow users to submit requests for books not currently available in the catalog, enabling administrators to consider inventory expansion based on user demand.

##### ***Acceptance Criteria:***

- Users can submit book requests with title, author, and justification
- Administrative interface to review and manage book requests
- Status tracking for submitted requests
- Notification to users when requested books are added to catalog

#### **FR-007: Administrative Dashboard**

**Description:** The system shall provide administrators with a comprehensive dashboard interface for managing platform operations including inventory management, order processing, user management, and system analytics.

##### ***Acceptance Criteria:***

- Book inventory management (add, edit, delete books)
- Order management and processing tools
- User account management and permissions
- Sales reporting and analytics visualization
- System performance monitoring metrics

## Functional Requirements Prioritization

### Must Have Requirements

- **FR-001: User Registration and Authentication** - Essential security foundation
- **FR-002: Book Catalog Management** - Core e-commerce functionality
- **FR-003: Shopping Cart and Checkout** - Primary business transaction capability
- **FR-004: Order Processing and Management** - Critical for order fulfillment

### Should Have Requirements

- **FR-005: Notification System** - Important for user engagement and communication
- **FR-006: Book Request Feature** - Valuable for inventory expansion and user satisfaction

### Could Have Requirements

- **FR-007: Administrative Dashboard** - Enhances operational efficiency but not critical for launch

### 3. Non-Functional Requirements

#### Non-Functional Requirements Definition

Non-functional requirements address system properties such as performance, security, scalability, usability, reliability, and maintainability rather than specific behaviors or functions. Non-functional requirements are equally important to ensure software quality and product business case delivery.

#### Non-Functional Requirements List

##### NFR-001: Performance Requirements

**Description:** The system must provide responsive user experience with acceptable loading times and transaction processing speeds under normal and peak usage conditions.

##### *Specifications:*

- Page load times shall not exceed 3 seconds under normal load
- Search results shall be returned within 2 seconds
- Checkout process completion within 30 seconds
- Database queries optimized with proper indexing
- Support for concurrent user sessions without performance degradation

##### NFR-002: Security Requirements

**Description:** The system must implement comprehensive security measures to protect user data, financial information, and system integrity against common security threats and vulnerabilities.

##### *Specifications:*

- JWT-based authentication with secure token management
- BCrypt password hashing with appropriate salt rounds
- HTTPS encryption for all data transmission
- Input validation and sanitization to prevent injection attacks
- PCI DSS compliance for payment processing
- Role-based access control with principle of least privilege



### **NFR-003: Scalability Requirements**

**Description:** The system must be designed to accommodate growth in user base, transaction volume, and data storage without requiring fundamental architectural changes.

#### ***Specifications:***

- Horizontal scaling capability through stateless component design
- Database performance optimization for growing data sets
- Load balancing support for multiple application instances
- Efficient caching strategies to reduce server load
- Modular architecture supporting independent component scaling

### **NFR-004: Usability Requirements**

**Description:** The system should provide an intuitive and accessible user interface that enables users to accomplish their goals efficiently regardless of their technical expertise level.

#### ***Specifications:***

- Responsive design supporting desktop and mobile browsers
- Intuitive navigation with clear information architecture
- Accessibility compliance with WCAG 2.1 guidelines
- Consistent user interface design patterns
- Error messages that provide clear guidance for resolution

### **NFR-005: Reliability Requirements**

**Description:** The system should maintain high availability and data integrity while providing consistent functionality under various operational conditions.

#### ***Specifications:***

- System uptime of 99.5% excluding scheduled maintenance
- Graceful error handling with appropriate user feedback
- Data backup and recovery procedures
- Transaction consistency through ACID compliance
- Fault tolerance for external service dependencies

#### **NFR-006: Maintainability Requirements**

**Description:** The system shall be designed and implemented to facilitate ongoing maintenance, updates, and feature enhancements throughout its operational lifecycle.

##### ***Specifications:***

- Clean code architecture following SOLID principles
- Comprehensive documentation for system components
- Automated testing suite with adequate code coverage
- Version control integration with branching strategies
- Continuous integration and deployment pipeline support

#### **NFR-007: Compatibility Requirements**

**Description:** The system shall operate effectively across different browsers, devices, and integration environments to ensure broad accessibility and functionality.

##### ***Specifications:***

- Cross-browser compatibility (Chrome, Firefox, Safari, Edge)
- Mobile-responsive design for various screen sizes
- RESTful API design for third-party integrations
- Database compatibility with MySQL
- Docker containerization for deployment flexibility

## Non-Functional Requirements Prioritization

### Must Have Requirements

- **NFR-002: Security Requirements** - Critical for protecting user data and maintaining trust
- **NFR-001: Performance Requirements** - Essential for acceptable user experience
- **NFR-005: Reliability Requirements** - Fundamental for business operations

### Should Have Requirements

- **NFR-004: Usability Requirements** - Important for user adoption and satisfaction
- **NFR-006: Maintainability Requirements** - Critical for long-term system sustainability
- **NFR-007: Compatibility Requirements** - Important for broad user accessibility

### Could Have Requirements

- **NFR-003: Scalability Requirements** - Valuable for future growth but not critical for initial launch