Software Requirements Specification

for

Bookstore Web-App

Version 1.0, Approved by Tran Dinh Phu

Prepared by Nguyen Tien Dat, Doan Minh Quan,
Bo Quoc Trung

Group 13, 2324II INT2208E 23, VNU-UET

March 14, 2024

Table of Contents

Revision History

Name	**Date**	**Reason For Changes**	**Version**
Draft #1	2024-03-14	Initial Version	0.1
Draft #2	2024-03-21	Added requirements for missing sections	0.2
Version 1.0	2024-03-28	First complete version of the SRS	1.0

1. Introduction

The introduction of the Software Requirements Specification (SRS) offers a comprehensive overview of the entire SRS, including its purpose, scope, definitions, acronyms, abbreviations, references. The primary goal of this document is to thoroughly examine and provide detailed insights into the complete **Bookstore Web-App software system** by defining the problem statement in detail. Additionally, it focuses on identifying the capabilities desired by stakeholders and their respective needs while outlining high-level product features. Within this document, detailed requirements for the **Bookstore Web-App software system** are provided.

1.1 Document Purpose

This document aims to present a detailed description of the **Bookstore Web-App software system.** It will describe the purpose and features of the system, its target audience, what it may do, its user interfaces and the constraint under which it may operate. It also defines how our client, team and audience see the product and use its functionality.

1.2 Document Conventions

1.2.1 Abbreviations

Acronyms and abbreviations related to Bookstore Web-App are spelled out upon first use, followed by the abbreviated form in parentheses for subsequent mentions.

1.2.2 Formatting

- Headings: Times New Roman/18 font size/Bold
- Sub Headings: Times New Roman/14-16 font size/Bold

1.3 Project Scope

Bookstore Web-App is an online platform that facilitates the purchase of books over the Internet. Its primary purpose is to provide a convenient, easy-to-understand, user-friendly marketplace where users can easily browse, search and purchase books from a vast catalog spanning various genres and categories

The key goals of **Bookstore Web-App** are :

- Provide an efficient shopping experience
- Enable access to a wide variety of books from various publishers and authors

The essential key features of the E-Commerce include

- CRUD functionality for admin (Create, Read, Update, Delete), view and confirm book order
- Users can views book details and add, delete books from their shopping cart, checkout and the web-app will generate a confirmation order for them
- Guest users can view the details of books (Authors, Summary, Price, etc…)

1.4 References

This Software Requirements Specification (SRS) does not currently reference any external documents or resources. As the project progresses, additional resources may be identified and documented in this section.

2. Overall Description

The Bookstore Web-App Software is a platform designed to facilitate the online buying and selling of books. It provides customers with an intuitive interface to effortlessly browse, search, and purchase books, while offering administrators powerful tools for inventory management, order tracking, and customer engagement. Anticipated users include readers, publishers, and bookstore owners seeking to enhance their online presence. Operating within known constraints and dependencies, the software prioritizes security, privacy, and accessibility, ensuring a seamless and efficient experience for all stakeholders.

2.1 Product Perspective

Bookstore Web-App is an entirely new product. This SRS defines the functionalities and features for the initial launch, and is not intended as an update or replacement for any existing application.

2.2 User Classes and Characteristics

2.2.1 Customers

- Primary, favored user class of the application
- Customers can browse the book catalog, search for specific titles or authors, add book to shopping cart, and complete checkout process
- Can create and manage their account, view order history, leave review on books

2.2.2 Administrator

- Managing book catalog, including adding, editing, and removing book listings
- Access to administrative dashboard to manage order
- Oversee user accounts, handling customers inquiries or issues

2.2.3 Guest Users

- Users that don't have any accounts
- Will be provided with limited functionality such as browse book catalogs view books details, reviews from customers
- Cannot add book to shopping cart, turn to checkout process
- Encourage users to create account and required to create account when they access limited functionalities

2.3 Operating Environment

2.3.1 Hardware Platform

There are no specific hardware requirements. The application is web-based and intended to run on various user devices with a modern (actively maintained and updated as of March 2024) Web Browser.

2.3.2 Operating Systems and Versions

The application should function properly on commonly used operating systems, including:

- Windows, MacOS (latest major versions)
- Linux (latest stable/supported kernel versions)

2.3.3 Co-existing Software

The web application should be designed to coexist peacefully with other applications and services typically used on user devices (with Web Browser(s) installed) and web servers. Standard internet protocols and web technologies will be employed to ensure compatibility.

2.3.4 Infrastructure Requirements

Due to the web-based nature and the expected development timeframe of a few weeks, extensive technical infrastructure work is not anticipated.

Standard web server and database configurations will suffice for the initial deployment:

● Disk capacity	>=64GB
● Memory (RAM)	>=2GB
● Front-end	HTML, CSS with Django Templates
● Back-end	Django, PostgreSQL (latest versions as of March 2024)

Should the application evolve, become production-ready and its user-base expands, additional infrastructure requirements might be identified and be updated in this section.

2.4 Design and Implementation Constraints

- This system is developed using Python Django framework and PostgreSQL database.
- Compliance with corporate and regulatory policies, particularly regarding data privacy and security, such as PCI DSS standards for payment card data.

2.5 Assumptions and Dependencies

- Users are assumed to have reading ability.
- Users are assumed to have stable internet connectivity while accessing the Bookstore Web App.
- The app may utilize browser cookies and local storage mechanisms for storing user preferences, session data, and shopping cart information. It is assumed that users' browsers support and allow the use of cookies and local storage for these purposes.

3. System Features

Features of Bookstore Web-App will be categorized by Use-cases, with each use-case expanded with a description, actors involved, trigger(s) initiating the interaction, success conditions, and the main and secondary events that make up the user journey.

No.	**Use-case**	**Actor**
01	Account login	User
02	Account registration	User, Guest
03	Add to wishlist	User
04	See book information	User, Guest
05	Checkout	User
06	Order processing	Administrator
07	Add/remove books currently for sale	Administrator
08	Account management	Administrator
09	See statistics	Administrator
10	Change book’s information	Administrator
11	Change user’s information	User
12	Find books’ names and information	User

Specification:

Use case	Content
Use case name	Login
Description	Allow guest users to login to their already-existing account
Actor	Guest user
Trigger	Click on Login button
Condition	Access the website
Result	Login into accounts successfully
Main events	1. A login page appear 2. Guest users input account username and password 3. System check if account username exists and if password is correspond to password saved in Database 4. If yes, login to user account, and move them to main page 5. End user case
Additional events	1. If username does not exist, display error and return to login page 2. If password doesn't match username, display error and return to login page

Use case	Content
Use case name	Register
Description	Allow guest user/user to create login account and receive rights of user
Actor	Guest user, user
Trigger	Click on Register button
Condition	Access to website
Result	Register an account successfully
Main events	1. A register page appear 2. Guest user put necessary information

	3. System verify information input 4. If successful, access the main page and the system automatically login for user 5. End use-case
Additional events	1. If input information has error, display errors 2. Refresh the register page

User case	Content
User case name	Add to wishlist
Description	User add their desired books into a list to monitor Books availability and easier find
Actor	User
Trigger	Click on Star button
Condition	The book hasn't on wishlist yet
Result	Books add to users' wishlists
Main events	1. The wishlist button (Star button) appear and initially look grey 2. User click on the button 3. System add the books to their wishlist 4. The star turned to bright yellow color
Additional events	1. If books already exist in wishlist, remove it from wishlist and 2. If users haven't had account yet, require them to register new account or login into already-exist account

Use case	Content
Use case name	View books' informations
Description	Users and Guest Users can view book's information (Authors, Price, Publishers, etc…)
Actor	User, Guest user
Trigger	Click on a books' corresponding name or image or button

Condition	Access to the website
Result	Redirect to the page displaying books informations
Main events	1. User and Guest Users can click on links of books 2. A page appear and display images of books uploaded and informations
Additional events	

Use case	Content
Use case name	Checkout
Description	Users buys books in their carts and process to payments
Actor	Users
Trigger	Users click on the "Process to Checkout" button
Condition	Access to the website
Result	A page dedicated to process orders appear in user browser
Main events	1. A pages for processing orders appear 2. Users can add and remove books they want from the carts 3. Users press the button "Pay" 4. System update the Database and confirm the user
Additional events	1. Systems display errors when a books is out of stock 2. Systems required Guest User to register new account or login to their accounts to use this features

Use case	Content
Use case name	Order processing
Description	Administrators view orders and confirm order
Actor	Administrator

Trigger	Administrator click on name or button corresponding to each orders
Condition	Having admin roles
Result	A pop-up of orders' information appears on admin's browser
Main events	1. Orders information appear 2. Admin click "Confirm" or "Decline" buttons 3. Systems display result and send result to users corresponding to orders
Additional events	

Use case	Content
Use case name	Add/remove books from catalog
Description	Add new books to catalog and delete existing books from catalog
Actor	Administrator
Trigger	Administrator right-click to the books' buttons or name or images to delete or the "Add" button to add
Condition	User have admin roles
Result	Small popup for delete and add appear
Main events	1. Administrator add book image and information or select delete books 2. System update the Database
Additional events	

Use case	Content
Use case name	Account Management
Description	Managers accounts in information, assurance for accounts for security, private information.
Actor	Administrators

Trigger	Access to Database
Condition	Need Admin roles
Result	A page dedicated to process orders appear in user browser
Main events	
Additional events	

4. Data Requirements

The data handled by the Bookstore Web-App includes information entered by users, such as book titles, author names (for searching), user information (for account registration/login), as well as any preloaded data like book descriptions and inventory information. The app will process this data to display books, manage user accounts (if applicable), and facilitate functionalities like search and purchase.

4.1 Data Acquisition, Integrity, Retention, and Disposal

4.1.1 Acquisition

Bookstore Web-App will acquire data through two primary methods:

1. **User Input:** Users will create data by adding books (titles, authors), potentially creating user accounts (usernames, passwords), and interacting with the app's features (searches, reviews).
2. **Pre-loaded Data:** Certain data, like book descriptions and initial inventory levels, may be pre-loaded into the system during initial setup. This data can be provided manually or through a data import process.

4.1.2 Integrity

The system enforces data integrity through the use of various techniques:

- **Input Validation:** User input should be validated to prevent nonsensical entries (for example, negative inventory levels).
- **Regular Backups:** Backups of the entire database should be performed periodically to allow for recovery in case of data loss.

4.1.3 Retention and Disposal

- **User Data:** User accounts and associated data (purchase history, reviews) should be retained as long as the user account is active. Users should be able to request data deletion upon account closure, with a defined timeframe for complete removal (for instance, anonymization after 30 days).
- **Temporary Data:** Temporary data generated during app operation (cache, search results) can be disposed of once its purpose is served.
- **Log Files:** Server logs containing information about user activity and system events may be retained for a specific period for troubleshooting purposes, then deleted.
- **Backups:** Backups should be retained for a designated period based on disaster recovery needs, then overwritten or deleted.

5. External Interface Requirements

This section will define how Bookstore Web-App interacts with users and any external systems. User communication will likely focus on web interfaces, providing clear and informative messages, prompts, and error handling. If external systems are involved (e.g., payment gateways for purchases), API specifications will be outlined, ensuring smooth data exchange and functionality. The goal is to establish clear communication protocols for a seamless user experience and efficient system operation.

5.1 User Interfaces

5.1.1 General UI Characteristics

- **Standard Elements:** Common UI elements like buttons, menus, and search bars will be implemented consistently across the app (in terms of color, size, and so on).
- **Error Handling:** Informative error messages will be displayed in a user-friendly manner (on a toast notification or an automatic redirect to an error webpage, with clear indication that an error was encountered), guiding users towards resolving any issues encountered.

5.1.2 UI Components

- **Homepage:** The landing page will showcase featured books. Prominent search and browsing options (for example, categories,...) will be readily available at the top of the page.
- **Book Details Page:** Individual book pages will display key information like title, author, description, product image (with minimum resolution of 300x200px and at most 50% JPEG compression (if applicable) for adequate clarity), and a clear call to action (for instance, "Add to Cart").
- **Browse/Search:** Users will have intuitive options to browse by category or use a search bar to find specific books. Facet filters (for example, genre, price range,...) can be considered for future enhancements, but will be introduced only if they don't add complexity.
- **User Accounts (if applicable):** Registered users may have a dedicated account area to manage profile details, purchase history, and potentially create wishlists.

5.2 Software Interfaces

5.2.1 Interactions with External Software Components

- **Django Framework (>=5.0.0):** This Python web framework will provide the foundation for building the web application. Django will handle user requests, interact with the database, and generate dynamic web pages.
- **PostgreSQL Database (>=16.0):** This relational database management system will store all application data, including book information, user accounts, and potentially order history.
- **Operating System:** The web application will run on and potentially interact with any of the designated operating system environments, for instance, via system-provided software libraries.

5.2.2 Data Communication and Exchange

- **Django and PostgreSQL:** With the help of framework-provided software libraries, Django will translate user interactions and requests into appropriate database queries, and vice versa.
- **Data Formats:** Data will primarily be exchanged in structured text formats, allowing efficient data transfer between the application and database.

5.2.3 Inter-component Communication Specifics

- **Django to PostgreSQL:**
    - **Data:** Book details, user information (if applicable), purchase details (if applicable).
    - **Services Needed:** Create, read, update, and delete (CRUD) operations on the database.
    - **Non-Functional Requirements:**
        - Response times for database queries should be acceptable (<=2 seconds) for a smooth user experience.
        - It is preferable that secure communication protocols (like HTTP over SSL/TLS) be used to protect sensitive data during transfer.
- **PostgreSQL to Django:**
    - **Data:** Query results containing book information, user data (if applicable), and potentially order history.
    - **Services Provided:** Responds to Django's database queries by delivering requested data.
    - **Non-Functional Requirements:**
        - The database should be highly available to ensure minimal downtime for the web application.
        - Security measures should be in place to prevent unauthorized access to the database.

5.3 Hardware Interfaces

Bookstore Web-App is a web-based application and has minimal direct interaction with hardware components. Users will access the application through standard web browsers on various devices (desktops, laptops, tablets, smartphones).

- **Inputs:** User interactions like mouse clicks, keystrokes, and touch inputs (on touch-enabled devices) will be received through the web browser.
- **Outputs:** The web browser will display the application's user interface, including book listings, product details, and potentially user account information (if applicable) on the device’s screen/monitor.
- **Timing Considerations:** Optimization techniques can be employed to minimize the amount of data transferred/processed and improve perceived performance.

5.4 Communications Interfaces

- **Web Browser Communication:**
  - **Protocol:** HTTP and potentially HTTPS for secure communication.
  - **Data Formats:** HTML, CSS, and JavaScript (if applicable) for user interface rendering and interaction.
  - **Security:** Implement HTTPS for secure user logins (if applicable) and any transmission of sensitive data (for example, user information during account creation).
  - **Data Transfer Rates:** Techniques like image optimization and code minification can be employed to improve loading times for users on slower connections.
- **Email (Optional):**
  - May be used for sending order confirmations or password reset requests (if applicable).
- **Electronic Forms (Optional):**
  - If user accounts are implemented, online forms will be used for registration and potentially reviews (if applicable).
  - **Data Formats:** Will be handled by the server-side code (for example, Django forms).
  - **Security:** Validate user input to prevent malicious code injection attempts and sanitize data before storing it in the database.

6. Quality Attributes

6.1 Usability

- The Web-App’s user interface should be intuitive, visually appealing, and easy to navigate, allowing users to find and potentially buy books effortlessly.
- Accessibility features should be implemented such as keyboard shortcuts for navigation.

6.2 Performance

- The Web-App should deliver content (icons, UI elements) with acceptable loading times and responsive user interfaces to ensure a smooth browsing and shopping experience.
- The Web-App should have high availability, ensuring that users can access manga content 24/7 with minimal downtime for maintenance or updates.

6.3 Security

- User authentication and authorization mechanisms should be implemented to protect user accounts and prevent malicious actors from accessing sensitive data.
- Data hashing/encryption should be used to obfuscate/secure sensitive user data, such as login credentials and payment information.

6.4 Safety

- Special measures should be taken to ensure the safety of underage users, including parental controls, age-appropriate content filters, and educational resources on online safety.
- User registration should require age verification (in the form of a Date of Birth (DoB) form) to prevent minors from accessing age-inappropriate content.
- The website should employ content filtering mechanisms (automatic or manual approval of new products) to prevent the distribution of inappropriate or offensive material, such as explicit content, hate speech, or graphic violence.

7. Internationalization and Localization Requirements

The initial rollout of the Bookstore Web-App will target a specific region (Vietnam), and currently there are no plans to include internationalization and localization (i18n/l10n) for other languages.

- Language Support:
  - User Interface (UI): Text displayed in the UI (menus, labels, buttons) should all be in Vietnamese.
  - Content: Book descriptions and other textual content may also be localized for Vietnamese. The book products themselves do not have to be Vietnamese, though having more of those can help boost the popularity of the web-application.
- Formatting:
  - Dates: The app should display dates according to Vietnamese locale (“DD/MM/YYYY”, “DD tháng MM năm YYYY).

- Numbers: The decimal separator should be a comma and currency symbols should be “VND” or "đ", placed after the numeric part of prices.

8. Glossary

Database	Collection of all the information monitored by this system.
Stakeholder	Any person with an interest in the project who is not a developer.
Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.