

Software Requirements Specification

for

Book Exchange Software

Version 6.0 approved

**Prepared by Michael Ramos, Hunter Williamson, Peter Uy, Alejandro Perez,
Weihao Liu**

**California State University, Los Angeles
CS3337 Software Engineering**

December 4th, 2024

Table of Contents

Table of Contents.....	<pg 2>
Revision History.....	<pg 3>
1. Introduction.....	<pg 4>
1.1. Purpose.....	<pg 4>
1.2. Intended Audience and Reading Suggestions.....	<pg 4>
1.3. Product Scope.....	<pg 5>
1.4. Definitions, Acronyms, and Abbreviations	<pg 5>
1.5. References.....	<pg 6>
2. Overall Description.....	<pg 7>
2.1. System Analysis.....	<pg 7>
2.2. Product Perspective.....	<pg 7>
2.3. Product Functions.....	<pg 8>
2.4. User Classes and Characteristics.....	<pg 8>
2.5. Operating Environment.....	<pg 8>
2.6. Design and Implementation Constraints.....	<pg 8>
2.7. User Documentation.....	<pg 9>
2.8. Assumptions and Dependencies.....	<pg 9>
2.9. Apportioning of Requirements.....	<pg 9>
3. External Interface Requirements.....	<pg 10>
3.1. User Interfaces.....	<pg 10>
3.2. Hardware Interfaces.....	<pg 10>
3.3. Software Interfaces.....	<pg 10>
3.4. Communications Interfaces.....	<pg 11>
4. Requirements Specification.....	<pg 12>
4.1. Functional Requirements.....	<pg 12>
4.2. External Interface Requirements.....	<pg 14>
4.3. Logical Database Requirements.....	<pg 15>
4.4. Design Constraints.....	<pg 16>
5. Other Nonfunctional Requirements.....	<pg 17>
5.1. Performance Requirements.....	<pg 17>
5.2. Safety Requirements.....	<pg 17>
5.3. Security Requirements.....	<pg 17>
5.4. Software Quality Attributes.....	<pg 17>
5.5. Business Rules.....	<pg 18>
6. Other Requirements.....	<pg 19>
Appendix A: Glossary.....	<pg 20>
Appendix B: Analysis Models.....	<pg 21>
Appendix C: To Be Determined List.....	<pg 22>

Revision History

Name	Date	Reason For Changes	Version
First Draft	11/20/24	Initial Creation of document	1.0
Second Draft	11/25/24	Added Sections 1 and 2	2.0
Third Draft	11/29/24	Added Sections 3 and 4	3.0
Fourth Draft	12/1/24	Added Sections 5 and 6	4.0
Fifth Draft	12/3/24	Added Appendixes and final formatting details	5.0
Final Draft	12/4/24	Finalized Document and added final details	6.0

1. Introduction

A web-based platform called **Book Exchange Software** was created to bring book lovers together by allowing them to upload, search, and trade books in a user-friendly, community-driven setting. The software seeks to decrease book waste, encourage literacy, and offer a cost-effective substitute for buying books. The program facilitates resource sharing and gives users access to a wide variety of books by establishing a platform for easy book exchanges.

This document is organized to guarantee accuracy and clarity about the expected functions of the program, in line with user expectations and corporate objectives. It offers the framework for productive collaboration and goal alignment among software engineers, testers, and project stakeholders.

1.1 Purpose

This paper aims to give a thorough description of the **Book Exchange Software**, covering its goals, requirements—both functional and non-functional—and implementation specifics. By enabling users to post, search, and trade books, the software seeks to unite book enthusiasts and provide a community-driven platform for resource sharing.

1.2 Intended Audience and Reading Suggestions

This document is intended for the following audiences:

1.2.1 Developers

Focus on the specific functional requirements and dependencies listed in Section 3 to understand implementation details.

1.2.2 Project Managers

Use Sections 1 and 2 to understand the high-level goals and overall scope of the project.

1.2.3 Documentation Writers

Use this document as a reference for creating user manuals or technical documentation.

This document is organized into the following major sections:

- Introduction
- Overall Description

- Specific Requirements

1.3 Product Scope

The **Book Exchange Software** is a web-based platform designed to allow users to:

1.3.1 Features

- Post books with a title, price, cover image, and link.
- View books in a comprehensive "Display Books" section.
- Bookmark books as "Favorites" for later reference.
- Manage their own book listings via a "My Books" section.
- Search books by title, rate them, and leave comments.

1.3.2 Objectives

The software seeks to encourage literacy and provide a community-driven marketplace where book lovers may easily trade resources. Users gain from:

- A streamlined process for sharing and discovering books.
- Enhanced user engagement with features like favorites, comments, and ratings.
- Easy access to their personalized book collections.

1.3.3 Non-commercial Focus

The software will concentrate on resource sharing and community engagement rather than acting as a commercial e-commerce platform.

1.4 Definitions, Acronyms, and Abbreviations

Refer to Appendix A

1.5 References

The following references were used in preparing this SRS:

1.5.1 Django Official Documentation

- **Author:** Django Software Foundation
- **Version:** 4.x
- **URL:** <https://docs.djangoproject.com>

1.5.2 Mozilla Developer Network (MDN) Documentation

- **Author:** Mozilla
- **URL:** <https://developer.mozilla.org>

1.5.3 IEEE SRS Standards

- **Author:** IEEE
- **Standard Number:** IEEE Std 830-1998
- **URL:** <https://standards.ieee.org>

1.5.4 Bootstrap Documentation

- **Author:** Bootstrap Team
- **URL:** <https://getbootstrap.com>

These references provide foundational guidance for development, design, and documentation of the Book Exchange software.

2. Overall Description

The Book Exchange platform is a Django-based web application designed to foster a community of book enthusiasts by allowing users to post, search, and exchange books. The platform aims to promote accessibility to books and create a collaborative environment where users can interact through books.

2.1 System Analysis

2.1.1 Goals of the Project:

- Create a user-friendly platform for book exchange to foster community engagement and reduce book waste.
- Provide features such as book posting, searching, rating, favoriting, and commenting to enhance user interaction.

2.1.2 Technical Hurdles:

- Managing image uploads and storage paths for book covers.
- Implementing a robust rating system to handle unique user-book relationships.

2.1.3 Proposed Solutions:

- Use Django's built-in models for managing book uploads.
- Employ Bootstrap to create a responsive design and improve the user interface across devices.

2.2 Product Perspective

2.2.1 How It Fits in a Larger System:

- It can integrate with external APIs for book data and metadata enrichment.
- Opportunities for extensions include mobile apps or integration with library systems.

2.3 Product Functions

2.3.1 The software will:

- Allow users to post books with details (name, price, web link, cover image).
- Enable search and filtering for books by title.
- Provide options to rate books and leave comments.
- Maintain a "Favorites" list for logged-in users.

2.4 User Classes and Characteristics

- **2.4.1 Admin Users:**
 - Manage the content and monitor community activities.
 - Require technical expertise in Django admin and database management.
- **2.4.2 General Users:**
 - Regular users who can post, search, rate, comment, and favorite books.
 - Varied levels of technical expertise; interface designed for ease of use.

2.5 Operating Environment

- **2.5.1 Hardware:**
 - Any device with internet access and a modern browser.
- **2.5.2 Software:**
 - Server: Deployed on a system running Python 3.12 and Django 5.1.
 - Client: Compatible with modern browsers like Chrome, Firefox, and Edge.
- **2.5.3 Additional Requirements:**
 - Bootstrap for frontend styling.

2.6 Design and Implementation Constraints

2.6.1 Constraints:

- Regulatory policies on data privacy.
- Limited storage for book images; requires optimization.
- Dependence on Django's built-in security features.

2.7 User Documentation

- 2.7.1 A user manual including:
 - How to register/login.
 - Instructions for posting, searching, and managing books.
 - Guidance on using "Favorites" and interacting with comments/ratings.
 - Online FAQs and tutorials for common tasks.

2.8 Assumptions and Dependencies

- 2.8.1 Assumes users have access to the internet and modern browsers.
- 2.8.2 Relies on Django for ORM and authentication functionalities.
- 2.8.3 Assumes Bootstrap and other dependencies are installed and up to date.

2.9 Apportioning of Requirements

- 2.9.1 Future versions might include:
 - API integration for book metadata.
 - A mobile app.
 - Advanced search filters like genres and publication years.

3. External Interface Requirements

3.1 User Interfaces

The user interface has the objective of being user friendly and intuitive

3.1.1 The user interface includes graphical elements such as buttons, icons, and menus making it extremely easy to navigate.

3.1.2 Users will have access to various features such as: posting a book, favoriting a book, rating a book, and commenting on various books

3.1.3 The user interface will provide options for administrators and users

3.1.4 Users will be able to post a book, favorite a book, rate a book, and comment on various books

3.1.5 The user interface will be accessible by desktop computers, laptops, tablets, and other various mobile devices encouraging accessibility for all

3.2 Hardware Interfaces

The system will run on any standard hardware that is commonly available today

Minimum requirements include:

3.2.1 Processor: Pendulum series or equivalent

3.2.2 RAM: 4 GB or Higher

3.2.3 Hard Disk: 200GB or Higher

3.2.4 Input Devices: Keyboard, Mouse

3.2.5 Output Devices: Monitor, Tablet screen, Phone screen, Laptop screen

3.3 Software Interfaces

3.3.1 Operating System: Compatible with Windows, Linux, or macOS

3.3.2 Developed using programming languages such as: Python, HTML, & CSS

3.3.3 Heavily relies on the framework of Django and utilizes Django

3.3.4 SQLite 3.xx

3.4 Communications Interfaces

3.4.1 This project runs efficiently on all types of web browsers, including browsers on Android/iOS.

3.4.2 The communication architecture follows the client-server model. The communication between the client and server is established over (HTTPS)

3.4.3 An email for tracking a purchase will be implemented later on

4. Requirements Specification

4.1 Functional Requirements

4.1.1 Application Requirements

4.1.1.1 The application shall run on a web browser in Windows, Mac, or Linux

4.1.1.2 The application shall process permissions upon first launch

4.1.1.2.1 The application shall connect to the Django server for data management

4.1.1.3 The application shall provide an interface for credentials

4.1.1.3.1 The application shall require user credentials, and will redirect to log in again if the credentials are incorrect

4.1.1.3 The application should provide an interface upon launching that will help select different functions of the program

4.1.1.4.1 The application shall provide a Homepage on launch

4.1.1.4.2 The application shall provide a function to register a new user

4.1.1.4.3 The application shall provide a Login/Logout function

4.1.1.4.4 The application shall provide a Display Books function

4.1.1.4.5 The application shall provide a Book Detail function

4.1.1.4.6 The application shall provide a function to rate a book

4.1.1.4.7 The application shall provide a function to add a book to favorites

4.1.1.4.8 The application shall provide a function to add a comment to a book

4.1.1.4.9 The application shall provide an About Us section

4.1.1.4.10 The application shall provide a My Books function

4.1.1.4.11 The application shall provide a Search function

4.1.1.4.12 The application shall provide a function to upload a new book

4.1.2 Homepage landing

4.1.2.1 The homepage (index) shall be the first page users access on the website

4.1.2.2 Public users will be able to view public data, but will not be able to make modifications to the data

4.1.2.3 Logged in users will have access to all the functions of the website, except for admin functions

4.1.3 User Registration

4.1.3.1 Users will be able to register a new account through the Django registration form

4.1.4 Log in/Logout Function

4.1.4.1 Users who log in should be redirected to the homepage again, while gaining access to new functions within the website

4.1.4.2 Users who are not logged in will have partial permissions to the websites function

4.1.4.3 Logout function should be available on all pages in case the user wants to switch accounts

4.1.5 Display Books Function

4.1.5.1 Users, whether logged in or not, shall be able to display all books that have been uploaded to the website thus far

4.1.5.2 Logged in users will be able to interact with the uploaded books with more functions

4.1.5.3 The Display Books function shall display the following:

- book's image

- book's name

- book's rating

- book's uploader name

4.1.6 Book Detail Function

4.1.6.1 Logged in users should be able to have functions in book detail like add to favorites, rate, and add comment

4.1.6.2 Users not logged in shall only be able to have a “read only” view of the book details

4.1.6.3 The Book Detail function shall display the following:

- book's name

- book's ratings

- book's uploader name

- book's image

- book's comments

4.1.7 Book Rate Function

4.1.7.1 The Rating function shall allow a logged in user to rate a book 1-5

4.1.7.2 The Rating function shall list the average rating of a book

4.1.7.3 The Rating function shall show how many total ratings the book has

4.1.7.4 The Rating function shall instruct a user to log in, if they have no already in order to rate a book

4.1.8 Book Favorite Function

4.1.8.1 The Favorite function shall allow logged in users to add books to a favorite list

4.1.8.2 The logged in user will have the ability to remove a book from the favorite list

4.1.8.3 If the user is not logged in, the favorite function will not be available

4.1.9 Book Comment Function

4.1.9.1 The comment function shall allow a logged in user to leave a comment in the book detail page

- 4.1.9.2 All comments will be listed on the same page, regardless of which user is logged in
- 4.1.9.3 Comments shall include the name of who posted the comment
- 4.1.9.4 Comment function will be restricted only to logged in users
- 4.1.10 About Us page
 - 4.1.10.1 The website shall provide a simple About Us page
 - 4.1.10.2 The page shall contain information of those who created the website
 - 4.1.10.3 The page shall be accessible by all users, whether logged in or not
- 4.1.11 My Books Function
 - 4.1.11.1 The My Books function shall pull up the list of books the current logged in users has uploaded
 - 4.1.11.2 This function will be limited only to those logged in
- 4.1.12 Search Books Function
 - 4.1.12.1 The search books function shall have an input box to search for books within the uploaded data base
 - 4.1.12.2 The search function shall return books with matching strings of letters just like the input
 - 4.1.12.3 The search function will be available for all users, whether logged in or not.
- 4.1.13 Upload Books Function
 - 4.1.13.1 The upload book function shall allow a logged in user to upload a new book into the database
 - 4.1.13.2 The user shall be able to upload an image with their book
 - 4.1.13.3 The user shall be able to add details like the name of the book and the price

4.2 External Interface Requirements

- 4.2.1 User Interfaces
 - 4.2.1.1 Credentials
 - 4.2.1.1.1 When a user is logging in, password shall be hidden by asterisks
 - 4.2.1.2 HTML, Bootstrap, and CSS are used to develop the Graphical User Interface
- 4.2.2 Hardware Interfaces
 - N/A
- 4.2.3 Software Interfaces
 - N/A
- 4.2.4 Communications Interfaces
 - 4.2.4.1 HTTPS GET Request
 - 4.2.4.2 Django Server Request

4.3 Logical Database Requirements

4.3.1 Types of information

4.3.1.1 User Data

- 4.3.1.1.1 User ID number
- 4.3.1.1.2 Username
- 4.3.1.1.3 Password
- 4.3.1.1.4 Account level (Admin or regular User)
- 4.3.1.1.5 Time and date of registration (for Admin)
- 4.3.1.1.6 Time and date of last login (For Admin)
- 4.3.1.1.7 Which books have been uploaded by the user
- 4.3.1.1.8 Favorites

4.3.1.2 Book Data

- 4.3.1.2.1 Book ID number
- 4.3.1.2.2 Title
- 4.3.1.2.3 Image of book
- 4.3.1.2.4 Price
- 4.3.1.2.5 Username of uploader
- 4.3.1.2.6 Rating
- 4.3.1.2.7 Comments

4.3.1.3 User Personal Data

- 4.3.1.3.1 Uploaded Books
- 4.3.1.3.2 Uploaded Comments
- 4.3.1.3.3 Favorites List
- 4.3.1.3.4 Ratings of Books

4.3.1.4 Rating Data

- 4.3.1.4.1 Username
- 4.3.1.4.2 Rating Number (1-5)
- 4.3.1.4.3 Book Name

4.3.2 Frequency of Use

4.3.2.1 User Data

- 4.3.2.1.1 Used to login and access data
- 4.3.2.1.2 Account level determines whether a user has access to certain functions
- 4.3.2.1.3 Used to verify access to database

4.2.3.1 Book Data

- 4.2.3.1.1 Used constantly throughout website whether users are logged in or not
- 4.2.3.1.2 Used to differentiate between specific book's data

4.3.3 Accessing Capabilities

4.3.3.1 HTTPS GET

4.3.3.2 HTTP GET

4.3.3.3 Django Requests

4.3.3.4 User Data

4.3.4 Data Entities and their relationships

4.3.4.1 User Data shall be accessed when needed to verify access to book data

4.4 Design Constraints

4.4.1 Having the website be fully functional on other platforms like mobile

4.4.2 How much bandwidth the website takes, if the code is not properly optimized it can lead to memory leaks

4.4.3 Time Constraints

The project gave us plenty of time to work on the website, but if we had like a year or more, I'm sure the website could be more fleshed out and have more functions

4.4.4 Backend constraints

4.4.4.1 Using Python

It would be nice to be able to use other forms of languages to help with organizing the website better

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The website runs on Django version 5.1.3. which will support the following:

- SQLite 3
- Stable internet connection
- Pages, hyperlinks, and functions that are traversable
- Server requests
- Concurrent requests
- Scalable structure

5.2 Safety Requirements

To ensure the safety of personal information that is handled, users must never share account details. Resetting your password periodically is a great way to ensure safety. We encourage a password that is more than twelve characters and contains at least one special character. A password like this lowers the possibilities of compromise. If a problem arises we are more than happy to assist you by email.

5.3 Security Requirements

Users who aren't registered will not have access to the website. There will be limited functions available to them. Users are encouraged to register on our website to gain full accessibility to our site and its features. The data that we handle is our utmost priority and we wish to protect it to the fullest extent. Sensitive information and data will not be sold or shared, encouraging the safety of our users. We will handle this data delicately and only store the data on our own servers.

5.4 Software Quality Attributes

The website maintains adaptability in which it functions efficiently on all browsers and mobile devices. By using a Model-Template-View architecture, navigating the website will be smooth. This encourages the responsiveness of our site to the user. Utilizing an object oriented structure provides us the ability to commit updates that refine our services and add new features. This makes the software have great scalability, useability, responsiveness, and maintainability.

5.5 Business Rules

If the user registers and logs in, they will have full access to the services. This will include viewing, uploading, reviewing, rating, favoriting, and commenting on the books listed on the website. Any errors or bugs of the services may be reported to our support team for review.

6. Legal and Ethical Considerations

To guarantee compliance and preserve user confidence, the Book Exchange Software must abide by moral and legal requirements. This includes the following:

6.1 Protecting intellectual property rights by preventing users from uploading or exchanging illegal or pirated book copies

6.2 Putting in place user agreement policies and content monitoring to stop these kinds of actions.

6.3 Data security and user privacy are the main ethical concerns. By only gathering the user data that is required, being transparent about data usage, and utilizing encryption to ensure safe data storage, the platform will abide by data protection laws

6.4 The platform will be built in accordance with Web Content Accessibility Guidelines (WCAG) to support users with disabilities.

6.5 Encourage diversity by guaranteeing that all users have fair access to resources and preventing prejudice in content recommendations.

6.6 Provide a reliable, equitable, and user-friendly book exchange environment, the project intends to solve these ethical and legal issues.

6.7 Unbiased sections

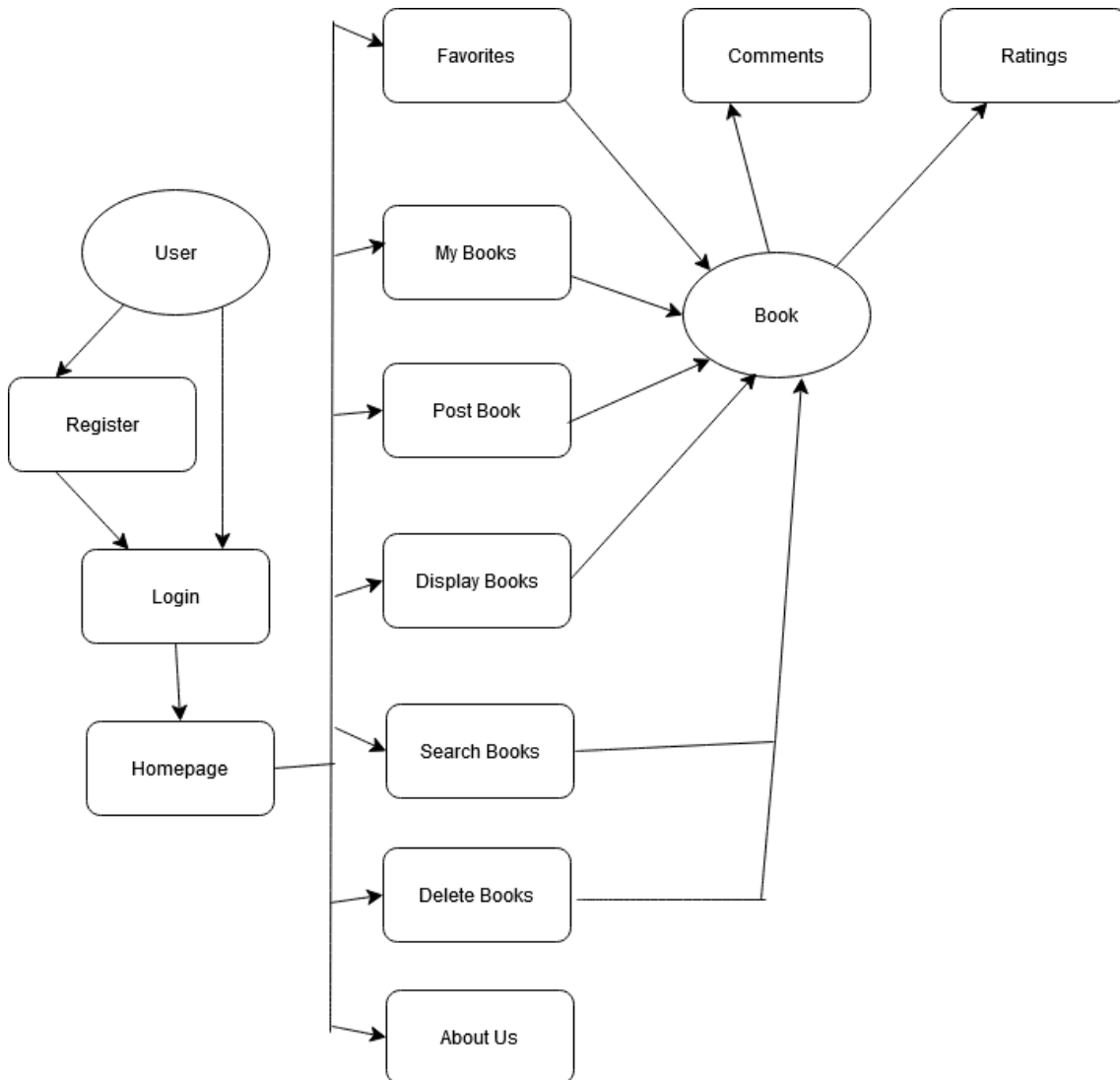
6.8 Accommodate all devices to access the site

6.9 Tame selection of colors to prevent epilepsy

Appendix A: Glossary

- **SRS:** Software Requirements Specification.
- **Book Exchange:** The name of the web-based software application.
- **Favorites:** A feature that allows users to bookmark books for later access.
- **Authenticated User:** A registered user who has logged in with valid credentials.
- **Guest User:** A user who can browse the site without logging in but cannot access personalized features.
- **Django:** A Python-based web framework used for backend development.
- **PyCharm:** an integrated development environment used for programming in Python.

Appendix B: Analysis Models



Appendix C: To Be Determined List

Account Management (4.1.3.1)

Leave User Review (4.1.9.1)

Order Tracking and Order History (4.3.1.3.1)

Review History/Management (4.1.7.3)

Contact Us Form (4.1.10.1)

Payment Processing System (3.3.3)

External Tracking Platform (3.4.2)

Email for Purchases/Tracking (3.4.3)