**Software Requirements Specification**

**for**

**The Book Exchange**

**Version 2.1 approved**

**Prepared by Dylan Huang, Thomas Ogden, Cindy Huang, Alexander Ramirez, Isaiah Villalobos, Aaran Li**

**Section 1, Group 5**

**November 19, 2024**

# Table of Contents

Table of Contents................................................................................................................... <pg 2>

Revision History..................................................................................................................... <pg 3>

1. Introduction................................................................................................................ <pg 4>
   1. Purpose........................................................................................................... <pg 4>
   2. Intended Audience and Reading Suggestions................................................ <pg 4>
   3. Product Scope................................................................................................ <pg 4>
   4. Definitions, Acronyms, and Abbreviations .................................................. <pg 5>
   5. References......................................................................................................<pg 5>
2. Overall Description.................................................................................................... <pg 6>
   1. System Analysis........................................................................................ <pg 6>
   2. Product Perspective........................................................................................ <pg 6>
   3. Product Functions...........................................................................................<pg 7>
   4. User Classes and Characteristics....................................................................<pg 7>
   5. Operating Environment.................................................................................. <pg 7>
   6. Design and Implementation Constraints........................................................ <pg 8>
   7. User Documentation...................................................................................... <pg 8>
   8. Assumptions and Dependencies.................................................................... <pg 8>
   9. Apportioning of Requirements...................................................................... <pg 9>
3. External Interface Requirements..............................................................................<pg 10>
   1. User Interfaces..............................................................................................<pg 10>
   2. Hardware Interfaces.....................................................................................<pg 10>
   3. Software Interfaces.......................................................................................<pg 11>
   4. Communications Interfaces..........................................................................<pg 11>
4. Requirements Specification......................................................................................<pg 12>
   1. Functional Requirements..............................................................................<pg 12>
   2. External Interface Requirements..................................................................<pg 12>
   3. Logical Database Requirements...................................................................<pg 12>
   4. Design Constraints.......................................................................................<pg 12>
5. Other Nonfunctional Requirements..........................................................................<pg 13>
   1. Performance Requirements..........................................................................<pg 13>
   2. Safety Requirements.....................................................................................<pg 15>
   3. Security Requirements.................................................................................<pg 15>
   4. Software Quality Attributes..........................................................................<pg 15>
   5. Business Rules..............................................................................................<pg 16>
6. Other Requirements..................................................................................................<pg 17>

Appendix A: Glossary..........................................................................................................<pg 18>

Appendix B: Analysis Models.............................................................................................<pg 19>

Appendix C: To Be Determined List....................................................................................<pg 21>

# 

# 

# Revision History

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| Alexander Ramirez | 11/13 | Implemented the Shopping Cart feature | 1.1 |
| Isaiah Villalobos | 11/14 | Implemented Comments feature | 1.2 |
| Cindy Huang | 11/15 | Implemented and formatted navbar along with Home page | 1.3 |
| Aaran Li | 11/15 | Reformatted the Login page and the Registration page | 1.4 |
| Thomas Odgen | 11/17 | Updated About Us page and implemented Search function | 1.5 |
| Dylan Huang | 11/17 | Implemented a Favorites page | 1.6 |
| Team | 11/18 | Compiled all updates into a single project file | 2.0 |
| Team | 11/18 | Debugged and finalized project | 2.1 |
|  |  |  |  |
|  |  |  |  |

# 

# 1. Introduction

The Book Exchange Web Application is designed to streamline the process of posting, searching, and managing books for exchange or sale. Key functionalities include adding books to a searchable catalog, favoriting items, and managing a shopping cart for purchases. This document outlines the requirements of the application, focusing on what the software should do while leaving implementation specifics to the Software Design Document (SDD).

## 

## 1.1 Purpose

The purpose of this document is to define the software requirements for the Book Exchange Web Application, developed using Python 3.8 or later and Django 5.1. This document serves as a comprehensive guide, detailing all functional requirements and user interactions for the application. It provides software engineers with a clear understanding of what the application should accomplish, including features such as book posting, catalog searching, favorites management, and shopping cart functionality. This document covers all aspects of the software, ensuring a complete specification of its capabilities.

## 1.2 Intended Audience and Reading Suggestions

This document is intended for developers, project managers, testers, marketing staff, end users, and documentation writers. Developers and testers should focus on the functional requirements, database requirements, and design constraints. Project managers and marketing staff may prioritize the introduction and product scope for high-level insights. End users and documentation writers can reference this document to understand the application’s features and guide future materials. Each audience is encouraged to focus on the sections most relevant to their roles.

## 1.3 Product Scope

The software product, Book Exchange Web Application, is designed to facilitate the process of managing books for exchange or sale. The application allows users to post books, search and browse a catalog, favorite books, and manage a shopping cart for purchases. Additionally, users can register accounts to track their own books and interactions.

The software is not designed to handle payment processing or integrate with external payment gateways in its current version. All transactions and exchanges are managed outside the platform.

Once released, the application will be used by students, educators, and readers to simplify book exchanges, reducing the need for physical meetings or lengthy search processes. The primary benefits include ease of use, a centralized platform for managing books, and improved efficiency in finding and exchanging books. The goal is to create an intuitive, user-friendly platform that meets the needs of its target audience.

## 1.4 Definitions, Acronyms, and Abbreviations

This information can be found in Appendix A: Glossary.

## 1.5 References

The following references were used in the creation of this Software Requirements Specification (SRS):

1. WCAG Guidelines
   * Title: "Web Content Accessibility Guidelines (WCAG) 2.1"
   * Author: World Wide Web Consortium (W3C)
   * Version: 2.1
   * Date: June 2018
   * Source:<https://www.w3.org/TR/WCAG21/>
2. General Data Protection Regulation (GDPR)
   * Title: "EU General Data Protection Regulation"
   * Author: European Union
   * Version: Regulation (EU) 2016/679
   * Date: April 27, 2016
   * Source:<https://gdpr-info.eu/>
3. California Consumer Privacy Act (CCPA)
   * Title: "California Consumer Privacy Act of 2018"
   * Author: California State Legislature
   * Version: Assembly Bill No. 375
   * Date: June 28, 2018
   * Source: <https://oag.ca.gov/privacy/ccpa>
4. Americans with Disabilities Act (ADA)
   * Title: "ADA Standards for Accessible Design"
   * Author: United States Department of Justice
   * Version: 2010 Standards
   * Date: 2010
   * Source:<https://www.ada.gov/>
5. Python Documentation
   * Title: "Python 3.8 Documentation"
   * Author: Python Software Foundation
   * Version: 3.8
   * Date: 2024
   * Source:<https://docs.python.org/3.8/>

These references provide the foundational standards, frameworks, and regulations that guided the development of the Book Exchange Web Application.

# 2. Overall Description

The Book Exchange Web Application is a platform designed to streamline the process of managing books for exchange or sale. It provides users with a range of features, including posting books for others to view, browsing and searching through a catalog of available books, favoriting items for later, and managing a shopping cart for potential purchases. The application also supports user accounts, enabling users to track their own books and activities.

The system operates within a browser-based environment, using Django 5.1 and Python 3.8 or later, and adheres to modern web development practices, including compliance with accessibility and privacy standards. It does not handle payment processing but facilitates the organization and management of book listings. This document provides a detailed overview of the software’s purpose, functionality, and constraints, offering a clear framework for developers and stakeholders to understand what the software is designed to accomplish.

## 2.1 System Analysis

* 1. Goals of the Project:  
     The Book Exchange Web Application aims to provide a centralized, user-friendly platform for managing books for exchange or sale, featuring book posting, search functionality, and personalized tools like favorites and a shopping cart.

Major Technical Hurdles and Solutions:

* **Database Management**: Ensured with Django ORM for efficient data handling.
* **User Security**: Implemented with Django’s authentication framework and HTTPS.
* **Cross-Platform Compatibility**: Achieved with responsive design using Bootstrap.

## 2.2 Product Perspective

The Book Exchange Web Application is an independent system, designed specifically for managing books for exchange or sale. It does not rely on external systems or integrate with other products, ensuring simplicity and self-containment in its operations.

Comparison to Similar Systems:

The Book Exchange application is comparable to existing online marketplaces or library management systems but focuses specifically on book exchanges for individual users, such as students or readers. Unlike platforms like Craigslist or Amazon, it does not handle payment processing or third-party logistics. This specificity reduces complexity and enhances ease of use for its target audience.

Motivations for Creation:

Existing systems often cater to broader markets, making them less user-friendly for niche use cases like peer-to-peer book exchanges. The motivation for this software lies in addressing that gap, providing a streamlined and focused platform for this purpose.

## 2.3 Product Functions

The Book Exchange Web Application provides the following major functions:

* **Book Posting**: Users can post books by entering details such as title, price, publisher, and an optional image. These books are added to the shared catalog for others to view.
* **Search and Browse**: Users can search the book catalog using keywords and browse through the available listings. Results are displayed in a clean, organized format.
* **Favorites Management**: Users can favorite books for easy access later. Favorited books are saved in a personalized list.
* **Shopping Cart**: Users can add books to a shopping cart to organize potential purchases. Items in the cart can be removed or finalized for review.
* **User Account Management**: Users can register and log in to manage their personal profiles, view their posted books, and interact with features like favorites and the shopping cart.
* **About Us Page**: Provides information about the development team and the purpose of the platform.

These functions are designed to create a seamless user experience, focusing on ease of use and the specific needs of book exchanges. The relationships between these features ensure users can efficiently manage and interact with the system.

## 2.4 User Classes and Characteristics

**General Users**: Visitors browsing the platform without accounts, primarily using search and catalog functions.

**Registered Users**: Account holders who post books, manage favorites, and use the shopping cart.

**Administrators**: Users with elevated privileges to manage content and ensure system integrity.

**Educators/Students**: Frequent users seeking academic materials, leveraging all platform features.

**Priority**: Registered users and educators/students are the primary focus, as they drive platform engagement and content creation.

## 2.5 Operating Environment

The Book Exchange Web Application is designed to operate in the following environment:

* Hardware Platform: Any device capable of running a modern web browser, such as desktops, laptops, tablets, and smartphones.
* Operating Systems: Cross-platform compatibility with Windows, macOS, Linux, iOS, and Android.
* Browser Requirements: Modern web browsers (e.g., Google Chrome, Mozilla Firefox, Microsoft Edge) with JavaScript enabled.
* Server Environment: Deployed on a server running Python 3.8 or later, with Django 5.1 framework.
* Network Requirements: Internet connectivity to access the platform and ensure seamless interaction with the server.

## 2.6 Design and Implementation Constraints

The following constraints influence the development of the application:

* Regulatory Policies: Compliance with GDPR, CCPA, and accessibility standards (WCAG 2.1).
* Hardware Limitations: The application assumes users have devices capable of running modern browsers.
* Language Requirements: The system must use Python 3.8 or later to support Django 5.1.
* Safety and Security: All user data must be encrypted in transit using HTTPS, and sensitive data must follow encryption protocols.
* Memory Constraints: Optimized queries and database interactions to handle large datasets without excessive resource use.

## 2.7 User Documentation

This document will be provided as well as the accompanying Software Design Specification (SDS)

## 2.8 Assumptions and Dependencies

The development of the Book Exchange Web Application is based on the following assumptions and dependencies:

Assumptions:

Users have access to devices with modern browsers and stable internet connectivity.

The server environment will meet hardware and software requirements (e.g., adequate storage and a supported operating system).

Dependencies:

Django 5.1 framework and its associated libraries.

Bootstrap for responsive front-end design.

Python 3.8 or later as the primary programming language.

A PostgreSQL or equivalent database for data storage and retrieval.

## 2.9 Apportioning of Requirements

None for now.

# 

# 3. External Interface Requirements

## 3.1 User Interfaces

GUI Constraints:

* On every interface page, there is a navbar allowing users to quickly navigate to main screens
* All interface screens follow the stylus sheet which specifies various fixed values
  + Font family for all “body” text: Arial, sans-serif
  + Custom padding and display characteristics, which vary depending on element families and classes
* Actions are toggled via regular links, with the exceptions of Login, Logout and Add Comment functionality
* Screen layout is handled via Django block elements
* Interfaces are used for the following software components:
  + Post Book
  + Display Book
  + About Us
  + My Books
  + Shopping Cart
  + Favorites
  + Login
  + Register
  + Search Results
  + Favorites
  + Book Detail

The program shall use simple language and clear, legible text styling to ensure all users can use the program seamlessly.

## 3.2 Hardware Interfaces

The program is developed with the built-in Django development server, so it is cross-platform. The data exchanged on the server consists of text and numbers, although custom objects are created with this data for better data association. The only communication protocol used in this program is HTTPS.

The program does not have specific hardware interface requirements, although the server should be deployed on a Linux or Windows device.

## 3.3 Software Interfaces

The program is self-contained and only relies on the Django framework (version 5.1). It does not require any external software or APIs beyond a web browser for user access.

## 3.4 Communications Interfaces

## Issues such as encryption, communication protocol selection, synchronization, and data transfer are handled by Django’s built-in development server. Messages, such as comments on a particular book, are kept as-is with no additional constraints imposed on them.

# 

# 4. Requirements Specification

## 4.1 Functional Requirements

The system shall allow users to view all books, their own books, search results for a custom input, the about us page, specific book details and their comments, their favorites list, and their shopping list. Users shall be able to post new books, remove books, post comments on books, and add/remove from their favorites list and shopping cart. Finally, the system shall support user registration and login functionality.

## 4.2 External Interface Requirements

No external interfaces shall be required. Users will only need a browser for access, and the rest of the system will be self-reliant.

## 4.3 Logical Database Requirements

Types of information: Each book shall have a name, price, publisher, book cover, and comment section. The book name and publisher will be simple text, the price a float value, and the book cover an image, displayed by referencing the link during runtime. The comment section shall be displayed by iterating a list of text values associated with the corresponding book.

Frequency of use: The data will be used every time the user views a book.

Accessing capabilities: Users will be able to view books even when they are not logged in, but they will not be shown as the publisher and will not see any books under “My Books” until they login or register.

Data entities/ relationships: Data shall be associated via Book objects, which will contain data fields each of basic types (strings, floats)

Integrity constraints: Users will be given freedom to name their books and choose usernames freely, while book prices shall strictly use numbers.

Data retention: User accounts shall be kept indefinitely

## 4.4 Design Constraints

* The system shall rely on Python 3.8 or later, since Django 5.1 requires this version or better to operate
* Text on every GUI will be clear and large enough to conform to the Americans with Disabilities Act
* All passwords will be encrypted by Django’s default one-way hashing algorithm

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

**Number of Terminals to be Supported**

- The program is made to operate on a single server and can handle requests coming from several client devices at once.

- Under the current test settings, the logs shall show no delays or bottlenecks that would restrict the number of terminals.

**- Dynamic Numerical Requirements**

- Response Times:

- Most of the site functions of the observed requests are to be completed within 0.02 seconds, which demonstrates good performance for typical web applications.

- The longest request duration (POST /login/) shall take around 0.485 seconds, reflecting additional processing required for authentication and session handling.

**Size of Data:**

- Response sizes will vary from 2820 bytes (bigger HTML pages) to 0 bytes (redirects). These numbers show that there are no excessive data payloads because they are within an acceptable range for normal web apps.

**Login Functionality**

* There are two steps in the login process:
* POST request: After a successful login, it redirects.
* Size of response: 0 bytes (redirect).
* It lasts for approximately 0.485 seconds.
* Six database queries are to be run, mostly to initialize a session and verify user credentials.

**POST request: (Redirects after successful login).**

* Size of response: 1304 bytes.
* It lasts for approximately 0.004 seconds.
* Three database queries are run in order to retrieve the material for the homepage.
* The entire login procedure (POST + GET) takes about 0.489 seconds.

**Homepage Performance**

* A lightweight page is the homepage (/).
* Size of response: 1304 bytes.
* It lasts for 0.004 seconds.
* Three database queries were run to retrieve the bare minimum of information needed to render the page.

For frequently visited pages, the homepage displays strong performance numbers and loads quickly.

**Post a Book Page**

* Compared to the homepage, the "Post a Book" page needs more data.
* Size of response: 2368 bytes.
* It lasts for 0.005 seconds.
* Three database requests were run, most likely to load the form structure and set any default values.

Observations: This page handles queries and loads faster while having a somewhat bigger response size.

**Display Books Functionality**

* Several entries are retrieved from the database and shown on the "Display Books" page.
* Size of response: 2312 bytes.
* It lasted for 0.018 seconds.
* Six database queries were run, most likely to get book details, user data, and any relevant metadata.

Observations: The page runs well with response times of less than 20 ms, even with a greater query count.

**About Us Page**

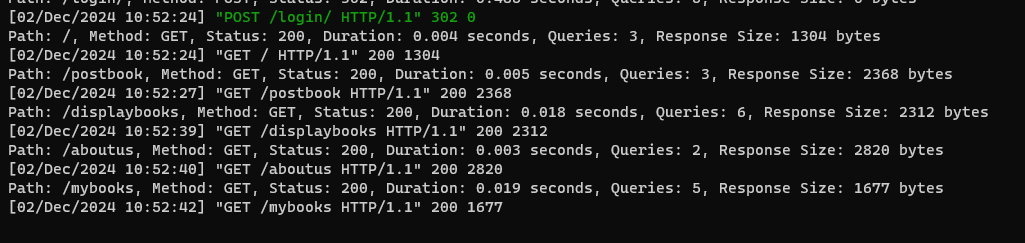
* Most of the content on the "About Us" page is static.
* The largest of the examined pages, the response size is 2820 bytes.
* Duration (fastest reaction time): 0.003 seconds.
* Two database requests were run, maybe to get page metadata or minimum configuration.

Observations: This page performs at its best, illustrating how well static or sparsely dynamic material may be served.

**My Books Page**

* The "My Books" page requires retrieving information unique to each user.
* Size of response: 1677 bytes.
* It lasted for 0.019 seconds.
* Five database requests were run, most likely to obtain a user's book list and related metadata.

Observations: Because it is dynamic, there is a somewhat higher query count, but the speed is still good.



## 5.2 Safety Requirements

The system must guarantee data integrity and prevent any loss, damage, or injury while it is in use.

* Among the safeguards are:
  + Frequent backups of important data, like book records and user credentials.
  + User input is validated to guard against data corruption or unintentional deletion.
  + Preventing risky actions, such as erasing data without asking for confirmation.
  + Limiting authorized personnel's access to sensitive operations (such database updates) is one preventive strategy.

To protect user data, external regulations like the General Data Protection Regulation (GDPR) must be adhered to.

## 5.3 Security Requirements

Authentication methods must be used to safeguard user identity and data.

* Every user needs to use legitimate login credentials.
* Sensitive user information, like email addresses and past book purchases, must be shielded against unwanted access by the system.
* All communications must encrypt data in transit using HTTPS.
* Based on user responsibilities, access control should limit sensitive operations (such seeing or changing book data).
* To ensure legal compliance, the system should adhere to external standards like the California Consumer Privacy Act (CCPA).

## 5.4 Software Quality Attributes

**Adaptability:** Other modules, such payment gateways or sophisticated search algorithms, should be simple to integrate with the application.

**Availability:** To guarantee users' continuous access, the system must maintain 99.9% uptime.

**Correctness:** Information that is shown, including book details or user-specific data, should always be up to date.

**Flexibility**: Features such as search filters or book categories ought to be adaptable to user preferences.

**Interoperability:** The program must function flawlessly on various platforms (e.g., desktop, mobile) and browsers (e.g., Chrome, Firefox, Edge).

**Maintainability:** With the help of modular architecture and good documentation, developers should be able to make changes to the software without causing regressions.

**Portability:** The site ought to function on multiple operating systems, including Linux, macOS, and Windows.

**Reliability:** The system must be able to manage multiple users at once without experiencing any issues.

**Reusability:** Common parts, such as authentication modules, can be used again in other applications.

**Robustness:** The system should log errors for debugging and gracefully handle incorrect inputs.

The codebase must provide simple unit and integration testing.

**Usability:** An intuitive user interface guarantees reduced learning curves for novice users and ease of navigation.

## 5.5 Business Rules

* Books can only be bought or posted by registered users.
* Admin users are the only ones with the power to change or remove private information, like user accounts or book records.
* Only the books that users have posted are editable or deletable.
* PCI DSS (Payment Card Industry Data Security Standard) must be followed when handling payment-related data.

# 

# 

# 6. Legal and Ethical Considerations

* To guarantee that user’s data rights are upheld, the system must comply with privacy regulations such as the CCPA or GDPR.
* All copyrighted content, including book descriptions and images, must be used with appropriate licenses or permissions.
* Ethical data use guarantees that user data isn’t sold to outside parties without express consent.
* Ensuring inclusivity for users with disabilities requires adherence to accessibility standards, such as WCAG 2.1
* Before users can access certain features, they must accept the application's clear terms of service and privacy policies.
* As required by law, any user data breaches should be promptly reported to the impacted users and regulatory agencies.

# 

# Appendix A: Glossary

1. **API**: Stands for Application Programming Interface. It’s the set of protocols for building and interacting the software.
2. **Bootstrap**: A free, open-source framework that helps web developers create responsive websites and web apps.
3. **Database**: An organized collection of structured information, or data, typically stored electronically in a computer system.
4. **Django**: Python based web framework that runs on a web server. Follows the model-template-views architectural pattern.
5. **Front-end**: A software interface (such as a graphical user interface) designed to enable user-friendly interaction with a computer by a user.
6. **Functional Requirements**: Statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations.
7. **HTTP**: Stands for Hypertext Transfer Protocol. It's used to transfer data over the web.
8. **Interface**: A defined set of rules and methods that describe how different software components or systems can interact and exchange data with each other.
9. **Non-functional Requirements**: Constraints on the services/functions offered by the system.
10. **ORM**: Stands for object relational mapping. It’s used by django to interact with the databases in an object-oriented way.
11. **UI**: Stands for user interface. It’s the point of contact between the user and the computer. This represents the front-end of the application.

# Appendix B: Analysis Models

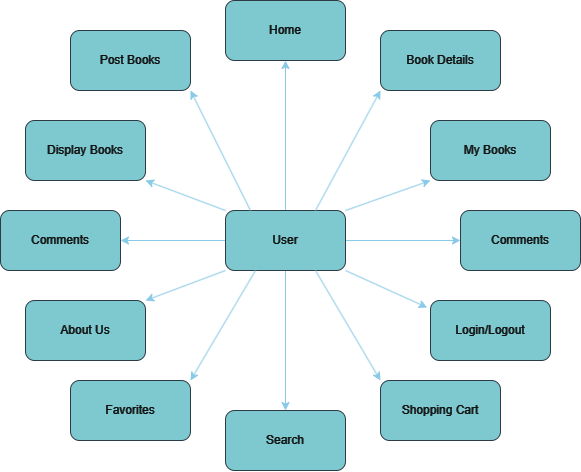


Figure A. Project Architecture Diagram

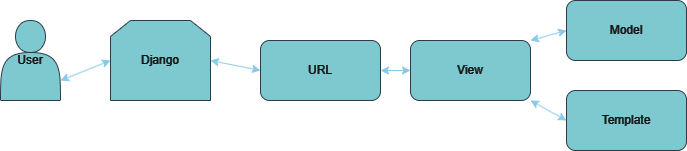


Figure B. Model-View Template (MVT) Structure

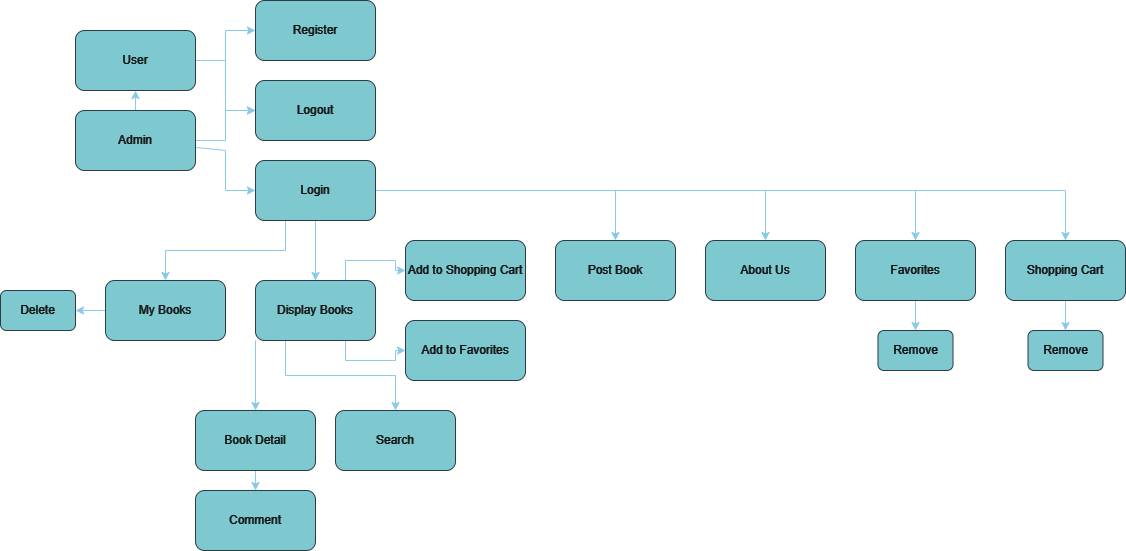


Figure C. Overall System Diagram and Data Flow

# 

# Appendix C: To Be Determined List

* Payment gateway to make purchases from the shopping cart.
* Favorites page should only be made available to registered users.
* Implement “Guest Checkout” option for unregistered users.
* Implement a profile page for customization and to manage settings.
* Default logout duration for user inactivity.
* Enhanced encryption algorithm for secure data transmission.
* 2-step authentication for extra security.
* Option to change password or to update password if forgotten.