Software Requirements Specification

for

<usedbooks.com>

Version 2.0 approved

Prepared by <雷艺湧>

<软工中外1班>

<7.1>

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Project Scope 1

1.4 References 1

2. Overall Description 1

2.1 Product Perspective 1

2.2 User Classes and Characteristics 2

2.3 Operating Environment 2

2.4 Design and Implementation Constraints 2

2.5 Assumptions and Dependencies 2

3. System Features 2

3.1 System Feature 1 2

3.2 System Feature 2 (and so on) 3

4. Data Requirements 3

4.1 Logical Data Model 3

4.2 Data Dictionary 3

4.3 Reports 3

4.4 Data Acquisition, Integrity, Retention, and Disposal 3

5. External Interface Requirements 4

5.1 User Interfaces 4

5.2 Software Interfaces 4

5.3 Hardware Interfaces 4

5.4 Communications Interfaces 4

6. Quality Attributes 5

6.1 Usability 5

6.2 Performance 5

6.3 Security 5

6.4 Safety 5

6.5 [Others as relevant] 5

7. Internationalization and Localization Requirements 5

8. Other Requirements 6

Appendix A: Glossary 6

Appendix B: Analysis Models 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| 雷艺湧 | 6.30 | 完成了所有内容 | 1.0 |
| 雷艺湧 | 7.1 | 优化了各个板块加上了对应的图表 | 2.0 |

# Introduction

## Purpose

This SRS document specifies the software requirements for the online second-hand bookstore system, including the initial release (version 1.0). The document is intended for a variety of readers, including:

Users: Understand the functionality and usage of the system.

Developers: Get detailed system requirements for development.

Project Manager: Manage project progress and resource allocation.

Database Administrator: Understand the data structure and database requirements.

Sales & Marketing Staff: Develop marketing strategies and understand user needs.

Tester: Develop a test plan and conduct system testing.

Potential Investors: Assess the feasibility and potential benefits of the project.

## Document Conventions

This document follows the following typographical conventions:

Bold: Emphasize important terms.

Italics: Used for document and chapter headings.

Code: Represents a snippet of code, command, or system response.

Requirements are uniquely identified using a format such as "REQ-001", where "REQ" represents the requirement and a number indicates its serial number.

## Project Scope

The online second-hand bookstore system is designed to facilitate the buying and selling of second-hand books among students. Its main goal is to provide an easy-to-use platform for listing books for sale, searching for desired books, and securely managing transactions. The software is designed to support university students by providing a cost-effective way to access textbooks and other reading materials, while also promoting circularity and sustainability. The detailed scope of the project can be found in the "General Description" section.

## References

This SRS references the following documents and resources:

Vision & Scope Document (version 1.0) - Provides an overview of the project's goals and objectives.

User Interface Style Guide - Detailing the criteria for UI design.

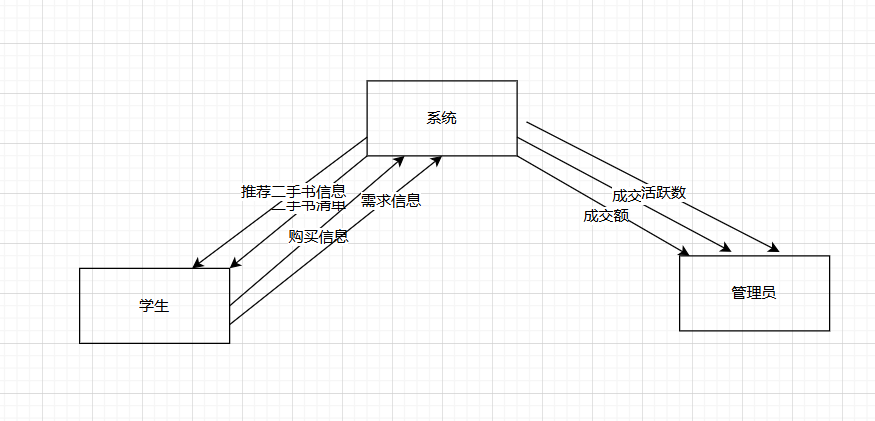
API Documentation - Describes the external and internal interfaces.

ISO/IEC 25010:2011 - Standard for quality models for systems and software.

# Overall Description

## Product Perspective

This product is a new online second-hand bookstore system, designed to replace the traditional offline second-hand book trading method. The system aims to realize the convenient trading of teaching materials and the recycling of resources among students through electronic and digital means. The system will interface with existing campus management systems and payment gateways to ensure that users can smoothly complete operations such as publishing, searching, purchasing, and trading books.



## User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product and describe their pertinent characteristics. Some requirements might pertain only to certain user classes. Identify the favored user classes. User classes represent a subset of the stakeholders described in the vision and scope document. User class descriptions are a reusable resource. If available, you can incorporate user class descriptions by simply pointing to them in a master user class catalog instead of duplicating information here.>

The main categories of users expected by the system and their characteristics are as follows:

**Student Users:**

Characteristics: University undergraduates, mainly using second-hand bookstores to buy and sell textbooks and extracurricular books.

Requirements:

1. Convenient book search, purchase and publishing functions.
2. Secure payment channels.
3. Better meet study and revision needs: such as getting notes or guidance

**Administrator:**

Features: Technicians responsible for system management and maintenance.

Requirements:

1. back-office management functions, user management, transaction review and data statistics.

## Operating Environment

The system will operate in the following environments:

1. Hardware platform: server and user terminals, including desktops, laptops, tablets and smartphone apps (mainly).
2. Operating system: Linux runs on the server side, and Windows, macOS, iOS, and Android are supported on the user side.
3. Location: Users are mainly distributed on campus, and servers and databases are hosted by technicians.
4. Related software: It needs to coexist peacefully with campus management systems, payment gateways (such as Alipay, WeChat Pay) and other software.

## Design and Implementation Constraints

The following factors will limit a developer's options:

1. Corporate or regulatory policies: Subject to compliance with relevant university and national regulations on information security and data privacy.
2. Hardware limitations: The server needs to have enough processing power and storage space to support concurrent access by a large number of users.
3. Interface: It needs to be seamlessly connected with the existing campus management system and payment gateway.
4. Specific technologies: Vue for front-end development, Java and MySQL databases for back-end development.

## Assumptions and Dependencies

This project assumes that the following factors, rather than known facts, may affect the needs stated in the SRS:

1. Third-party components: It is assumed that the interface between the payment gateway and the campus management system is stable and reliable.
2. Reuse expectations: The system design should consider possible future function expansion and module reuse.
3. Development and operation environment: It is assumed that the campus network environment is stable and can support a large number of users online at the same time.
4. Dependency: The project relies on the normal operation of external payment gateways and campus management systems, and if there is a problem with these systems, it will affect the normal operation of the online second-hand bookstore.

# System Features

## Post Textbooks

### Description

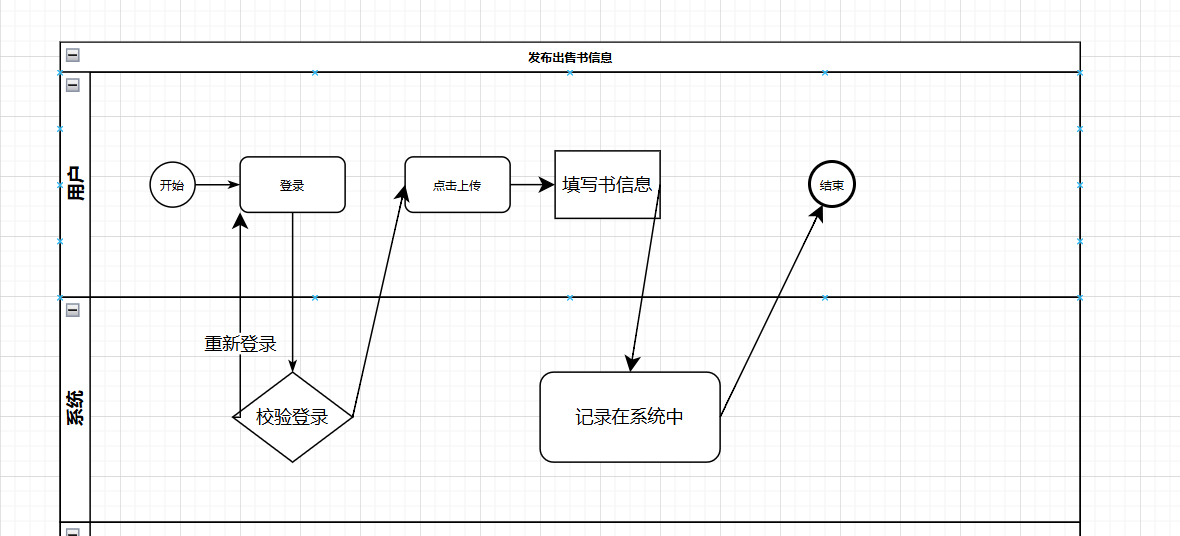
Users can publish their unwanted books on the system for other users to purchase. Priority: High.

### Stimulus/Response Sequences

1. The user logs in to the system.
2. The user selects the "Publish Used Books" option.
3. The system displays a form for publishing the book.
4. The user fills in the book information (title, author, description, price, pictures, etc.).
5. A user submits a request for book publishing.
6. The system verifies the information and saves the book data.
7. The system displays a confirmation message that the publication was successful.

### Functional Requirements

1. The system must provide a book publishing form with fields such as title, author, description, price, and image upload.
2. The system must verify the book information filled in by the user, including the required items check and the data format check. If the validation fails, the system should display an appropriate error message.
3. The system must save the information of the book submitted by the user to the database. If the save fails, the system should log an error and prompt the user to try again later.
4. The system must display a confirmation message to the user after the book has been successfully published.
5. The system must process the book images uploaded by the user and store them in the specified storage location. If the image upload fails, the user should be prompted to re-upload the image.



## Buy Textbooks

### Description

Users can browse and buy second-hand books published by other users. Priority: High.

### Stimulus/Response Sequences

1. The user logs in to the system.
2. The user browses the books that have been published in the system.
3. The user chooses to purchase a book.
4. Book details and purchase options are displayed.
5. The user confirms the purchase and submits the payment information.
6. The system processes the payment request.
7. After the payment is successful, the system updates the status of the book and notifies the seller.

### Functional Requirements

1. The system must provide a book browsing feature that allows users to filter books by different criteria such as category, price, author, etc.
2. The system must display the book's details, including the title, author, description, price, and seller information.
3. The system must provide a purchase option that allows the user to submit payment information.
4. The system must integrate with the payment gateway to process the user's payment request and return the payment status. If the payment fails, the user should be prompted and allowed to resubmit the payment information.
5. The system must update the status of the book and send a notification to the seller after the payment is successful. If updating the book status fails, the system should log an error and prompt the user to contact support.

### 

## Post textbooks in need

### Description

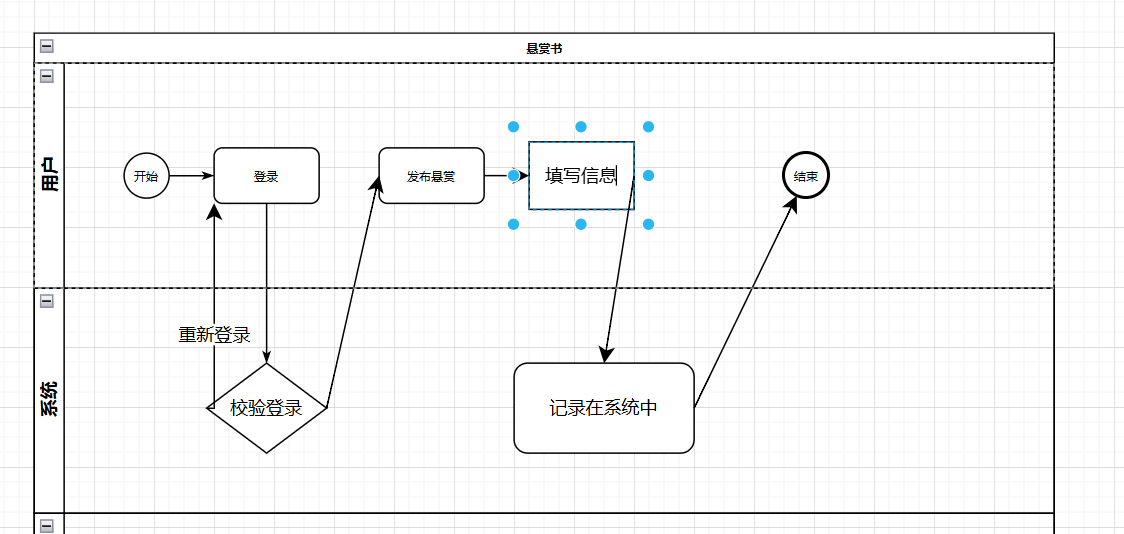
Users can post their own book requests for purchase for other users to view and respond to. Priority: Medium.

### Stimulus/Response Sequences

1. The user logs in to the system.
2. The user selects the "Post purchase demand" option.
3. The system displays the requirements release form.
4. The user fills in the required information (title, author, expected price, description, etc.).
5. The user submits a request for the release of requirements.
6. The system validates the information and saves the requirements data.
7. The system displays a confirmation message that the publication was successful.

### Functional Requirements

1. The system must provide a purchase request release form with fields such as title, author, expected price, and description.
2. The system must validate the requirements information filled in by the user, including the required items check and the data format check. If the validation fails, the system should display an appropriate error message.
3. The system must save the requirements submitted by the user to the database. If the save fails, the system should log an error and prompt the user to try again later.
4. The system must display a confirmation message to the user after the requirement is successfully published.
5. The system must allow other users to browse and search for posted purchase requirements and provide the option to respond. If browsing or searching fails, the system should log the error and prompt the user to try again later.



# Data Requirements

## Logical Data Model

**Students**

Attributes:

name (String) password (String) phone (String) account (decimal) major (String) grade (int)

Relationships: A student can sell multiple books and have multiple sales records

**Book**

Attributes:

title (String) seller (String) author (String) description (String) price (decimal) imageUrl (String)

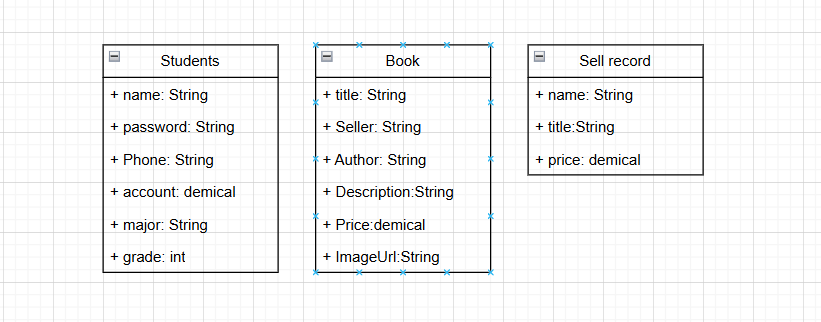
Relationship: A book can only be sold 1 time, and there is only one sales record

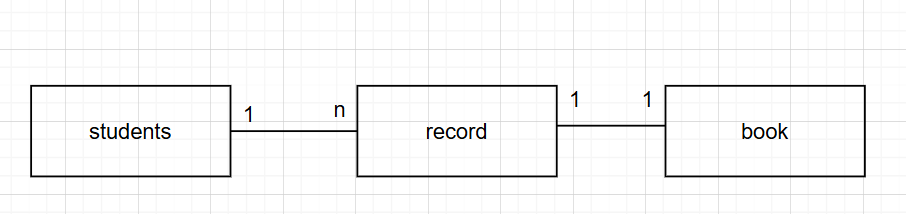
**SellRecord**

Attributes:

name (String), title (String), price (decimal)

Releationship:Keep a record of each book and the person who sold it





## Data Dictionary

**Students**

1. name: Student's name, string type, maximum length of 50 characters
2. password: Password, string type, maximum length 50 characters
3. phone: phone, string type, maximum length 20 characters
4. account: Account balance, decimal type, in the format of 0.00
5. major: Professional, string type, maximum length 100 characters
6. grade: grade, integer type

**Book**

1. title: Title: Title of the book, string type, maximum length 255 characters
2. seller: Seller, string type, maximum length 50 characters
3. author: Author, string type, maximum length 255 characters
4. description: Description, string type, maximum length of 1000 characters
5. price: price, decimal type, format 0.00
6. imageUrl: image address, string type, maximum length of 500 characters

**SellRecord**

1. name: Seller's name, string type, maximum length 50 characters
2. title: Title: Title of the book, string type, maximum length 255 characters
3. price: price, decimal type, format 0.00

## Reports

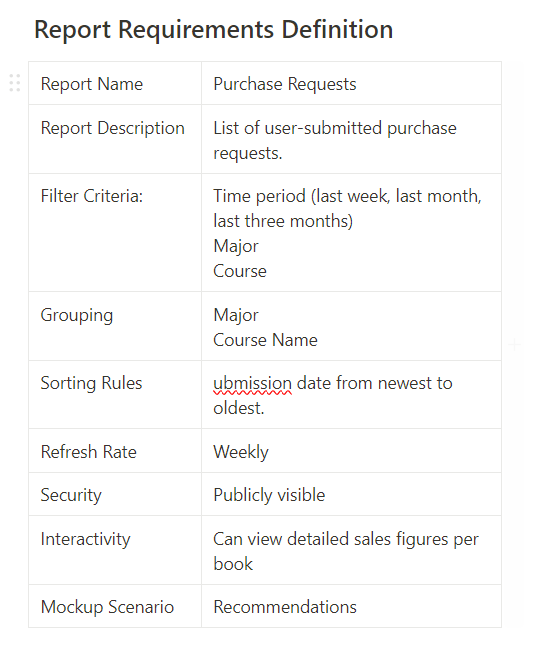
**Sales reports**

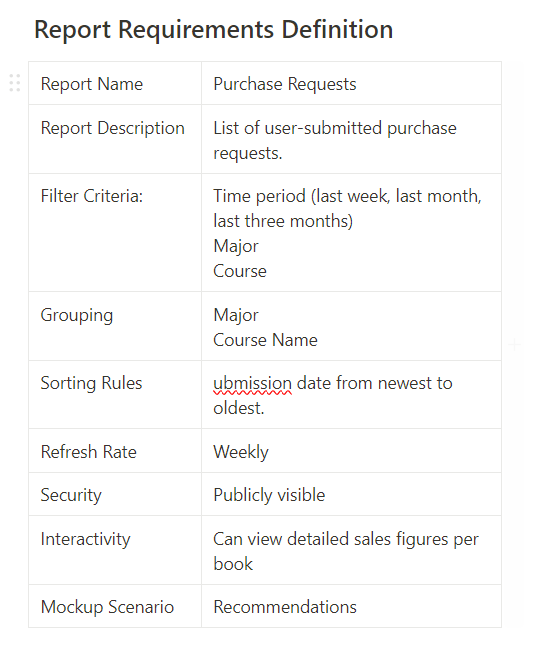
1. Description: Displays book sales within a specified time frame.
2. Content: Book title, author, number of sales, total revenue.
3. Sort: Sort by number of sales in descending order.



**Purchase Demand Report**

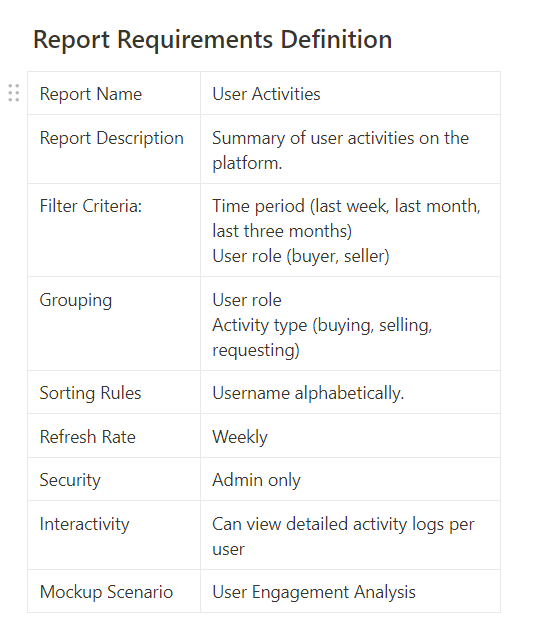
1. Description: Displays the user's published purchase requirements.
2. Content: Requirement ID, Book Title, Author, Expected Price, Major, Course Name, Release Date.
3. Sort: Sort by publication time in descending order.





**User activity reports**

1. Description: Displays the user's activities in the system.
2. Content: User ID, username, number of books published, number of books purchased,
3. number of requirements published.
4. Sort: Sort by username in ascending order.



## Data Acquisition, Integrity, Retention, and Disposal

**Data acquisition**

1. The data is obtained through the user's actions in the system, such as publishing a book,
2. publishing a purchase request, purchasing a book.
3. The system should provide an interface to import book information from external data sources.

**Data Integrity**

1. The system must verify that the data is in the correct format and that all required fields are filled in before the data is stored.
2. The system should back up the database regularly to prevent data loss.
3. Consistency and integrity should be considered when data is stored, and transactions should be used to ensure data consistency.

**Data Retention**

1. Data should be retained in accordance with legal and business needs. User accounts and transaction records are retained for at least five years.
2. For deleted data, the system should permanently delete it after a certain period of time, such as 30 days, and clean up the storage space.

**Data Disposal**

1. The system must provide a secure method of data deletion to ensure that deleted data cannot be recovered.
2. Users can request the deletion of their personal data, and the system shall handle these requests in accordance with relevant laws and regulations.

# External Interface Requirements

## User Interfaces

**Required Software Components:**

· **Login screen**

* **Description:** Interface for user authentication and access control.
* **Features:** Username and password fields, login button, forgot password link, and registration link.
* **Standards:** Follow company GUI standards for consistency.
* **Error Handling:** Display error messages for invalid credentials.

· **Main interface**

* **Description:** Central interface displaying the system's main functions and navigation options.
* **Features:** Navigation menu, dashboard overview, quick access to major features, and notifications.
* **Standards:** Consistent layout with a standard header and footer.
* **Error Handling:** Provide feedback for navigation errors.

· **Book search and browse interface**

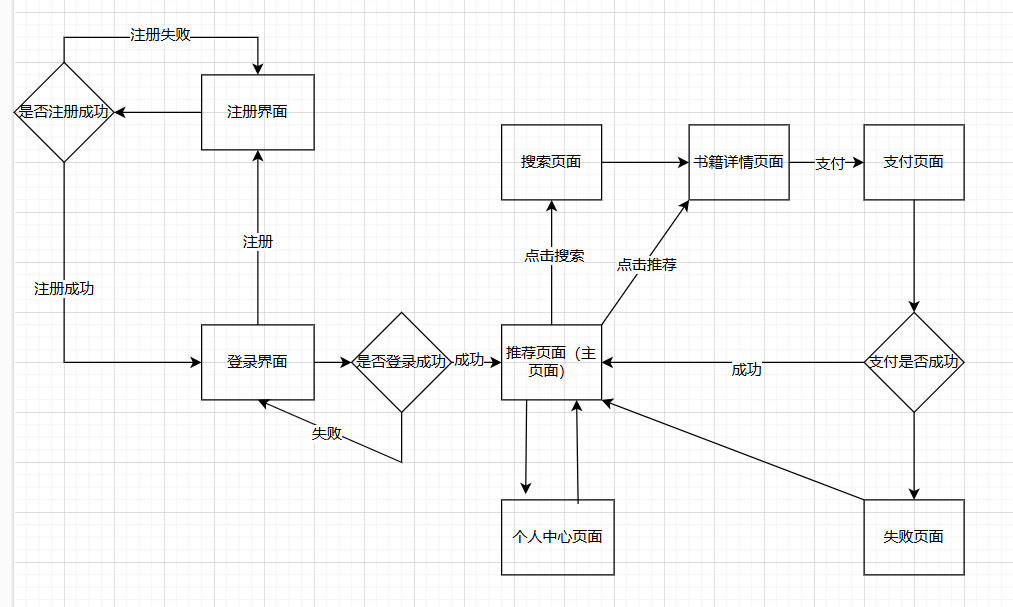
* **Description:** Allows users to search and browse book information.
* **Features:** Search bar, filters (by title, author, genre, etc.), book list with sorting options, and detailed book view.
* **Standards:** Intuitive and responsive design.
* **Error Handling:** Display appropriate messages for no results or search errors.

· **Purchase request submission form**

* **Description:** Interface for users to submit purchase requests for books.
* **Features:** Form fields for book title, author, expected price, description, major, class name, and image upload.
* **Standards:** Form validation and user-friendly layout.
* **Error Handling:** Highlight and describe errors in form submission.

· **User profile interface**

* **Description:** Allows users to view and edit their personal profile information.
* **Features:** Fields for username, password, email, grade, major, and account balance.
* **Standards:** Secure and private data handling.
* **Error Handling:** Provide clear messages for update success or failure.



## Software Interfaces

· **Database management system**

* **Description:** Processes and stores user information, book information, and purchase records.
* **Purpose:** Centralized data storage and retrieval.
* **Connections:** SQL queries for data operations.
* **Security:** Ensure secure access and data encryption.

· **Payment gateways**

* **Description:** Process payment transactions and verification.
* **Purpose:** Handle financial transactions securely.
* **Connections:** API calls for payment processing.
* **Security:** Use HTTPS and token-based authentication.

· **Mail server**

* **Description:** Sends notifications and confirmation emails.
* **Purpose:** Communicate with users via email.
* **Connections:** SMTP protocol for sending emails.
* **Security:** SSL/TLS encryption for email transmission.

· **External API**

* **Description:** Get and update book information.
* **Purpose:** Integrate with external book information services.
* **Connections:** RESTful API endpoints.
* **Security:** API key or OAuth for secure access.

## Hardware Interfaces

· **Input:**

* **Description:** User enters data via keyboard, mouse, and scanning device.
* **Supported Devices:** Standard keyboards, mice, and barcode scanners.
* **Data Interaction:** Real-time data input and validation.

· **Output:**

* **Description:** Data output to screen displays, printing devices, and storage devices.
* **Supported Devices:** Monitors, printers, and external storage (e.g., USB drives).
* **Data Interaction:** Display information in real-time, print reports, and store data securely.

## Communications Interfaces

Email attachments: Attachments must not exceed 10MB in size, and executable or compressed files are not allowed.

Network protocols: All API calls must follow a RESTful style and use standard HTTP methods (GET, POST, PUT, DELETE).

# Quality Attributes

## Usability

**Ease of Use:**

* The system should be intuitive, with a clean and straightforward user interface. Users should be able to navigate through the system with minimal guidance.
* The design should adhere to modern UI/UX principles to enhance user experience.

**Ease of Learning:**

* Provide tutorials and help documentation to assist new users in learning how to use the system effectively.
* Include tooltips and contextual help within the application to guide users.

**Memorability:**

* Design the interface in a way that users can easily remember how to perform tasks after not using the system for a while.
* Consistent design elements and workflows should be maintained throughout the application.

**Error Avoidance, Handling, and Recovery:**

* Implement form validations and input constraints to minimize user errors.
* Provide clear and descriptive error messages when something goes wrong and offer suggestions for recovery.
* Include an undo feature for reversible actions to allow users to recover from mistakes easily.

**Efficiency of Interactions:**

* Optimize workflows to minimize the number of steps required to complete tasks.
* Ensure the system responds quickly to user inputs to provide a smooth experience.

**Accessibility:**

* Follow accessibility guidelines (such as WCAG) to ensure the system is usable by people with disabilities.
* Provide keyboard navigation and screen reader support.

**Ergonomics:**

* Design the interface to reduce the physical strain on users, including appropriate font sizes, contrast, and spacing.

**Standards:**

* The application should conform to the company’s established user interface design standards and guidelines.

## Performance

**Response Time:**

* The system should respond to user actions within 2 seconds for most operations.
* More complex operations should not exceed a response time of 5 seconds.

**Throughput:**

* The system should handle up to 1000 concurrent users without significant performance degradation.

**Scalability:**

* The system should be designed to scale horizontally to handle increased loads as the user base grows.

**Efficiency:**

* Optimize backend processes to minimize server load and ensure efficient use of resources.

**Load Time:**

* Initial load times for the application should not exceed 3 seconds.

## Security

**Authentication and Authorization:**

* Implement strong authentication mechanisms (e.g., multi-factor authentication) to ensure that only authorized users can access the system.
* Use role-based access control (RBAC) to restrict access to sensitive features and data.

**Data Encryption:**

* Encrypt sensitive data both in transit (using HTTPS) and at rest (using AES-256 or similar encryption standards).

**Data Privacy:**

* Adhere to relevant data protection regulations (e.g., GDPR, CCPA) to ensure user data is handled appropriately.
* Implement data anonymization techniques where necessary.

**Security Policies:**

* Follow the company’s security policies and guidelines.
* Regularly update and patch the system to protect against vulnerabilities.

**Audit Logging:**

* Maintain comprehensive logs of all user actions for auditing purposes.
* Ensure that logs are protected from unauthorized access and tampering.

## Safety

**Data Loss Prevention:**

* Implement regular data backup processes to prevent data loss.
* Ensure that backups are stored securely and can be restored quickly in case of failure.

**Fail-Safe Mechanisms:**

* Design the system to enter a safe state in case of critical failures.
* Prevent actions that could lead to data corruption or loss.

**Safety Regulations:**

* Conform to any relevant safety standards and certifications required for the system’s operation.

## Reliability

**Uptime:**

* The system should have an uptime of 99.9%, ensuring high availability for users.

**Fault Tolerance:**

* Implement redundancy and failover mechanisms to ensure continuous operation in case of component failures.

**Error Handling:**

* Ensure robust error handling to gracefully manage unexpected conditions without crashing.

**Data Integrity:**

* Implement checks and balances to ensure data integrity is maintained throughout all operations.

**Recovery:**

* Develop a disaster recovery plan to restore operations quickly in the event of a major failure or disaster.

# Internationalization and Localization Requirements

To ensure that the product is suitable for use in various nations, cultures, and geographic locations, the following internationalization and localization requirements must be met:

#### Currency

* Support multiple currencies and allow users to select their preferred currency for transactions.
* Implement automatic currency conversion based on real-time exchange rates.

#### Date and Time Formatting

* Provide date and time formats that adhere to local conventions (e.g., MM/DD/YYYY for the US, DD/MM/YYYY for the UK).
* Allow users to select their time zone and display all times according to the selected time zone.

#### Number Formatting

* Format numbers based on local conventions, including the use of commas, periods, and spacing (e.g., 1,000.00 in the US, 1.000,00 in Germany).

#### Language

* Support multiple languages, including regional variations (e.g., American English vs. British English).
* Provide a mechanism for translating and displaying text in the user's preferred language.
* Ensure that the application supports Unicode to handle various character sets.

#### Names and Addresses

* Allow flexible input formats for names and addresses to accommodate different cultural conventions (e.g., family name first in East Asia).
* Provide address formatting that adheres to local postal standards.

#### Symbols and Units

* Use appropriate symbols and units for measurements, weights, and dimensions based on the user's locale (e.g., metric vs. imperial system).

#### Legal and Regulatory Compliance

* Adhere to international laws and regulations relevant to data privacy, e-commerce, and consumer protection.
* Ensure that the product complies with local regulations and standards, such as GDPR in the European Union.

#### Paper Sizes

* Support different paper sizes for printing, such as A4 in Europe and Letter in North America.

#### Electrical Specifications

* If applicable, ensure compatibility with local electrical standards, including voltages and plug shapes.

# Other Requirements

<Examples are: legal, regulatory or financial compliance, and standards requirements; requirements for product installation, configuration, startup, and shutdown; and logging, monitoring and audit trail requirements. Instead of just combining these all under "Other," add any new sections to the template that are pertinent to your project. Omit this section if all your requirements are accommodated in other sections. >

Appendix A: Glossary

<Define any specialized terms that a reader needs to know to understand the SRS, including acronyms and abbreviations. Spell out each acronym and provide its definition. Consider building a reusable enterprise-level glossary that spans multiple projects and incorporating by reference any terms that pertain to this project.>

Appendix B: Analysis Models

<This optional section includes or points to pertinent analysis models such as data flow diagrams, feature trees, state-transition diagrams, or entity-relationship diagrams. You might prefer to insert certain models into the relevant sections of the specification instead of collecting them at the end.>