|  |
| --- |
| **SOFTWARE REQUIREMENTS SPECIFICATION**  **REPORT**  **CMSE 322**  **Computer Engineering Department**  **Eastern Mediterranean University**  **PROJECT NAME:** University Second Hand Book Sales and Exchange Platform  **PROJECT START DATE: 05/03/2025**  **PROJECT END DATE: 21/05/2025**  **SUPERVISOR:** Prof. Dr. Duygu Çelik Ertuğrul  **SEMESTER TERM:** Spring 2024/2025  **PROJECT GR NO: 11** |

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

This SRS document was created primarily to inform the development team and end users about the project's goals and scope, as well as the system requirements, including both functional and non-functional needs. It will give them an idea of how our system will be. Before beginning the design phase, this paper will help designers correctly understand the requirements. Additionally, since all of the features will be stated, there will be less chance of future redesigns. We will focus on all the essential features: handling book availability, user transactions, and basic notifications.

## Document Conventions

This document follows the IEEE Standards for software requirements specification.

## Intended Audience and Reading Suggestions

This SRS is intended for multiple audiences, including developers, project managers, testers, and research writers who will do research about this type of application. It will begin with an overview of the system and its objectives, then we will dive into detailed functional and non-functional requirements. Readers who need quick orientation should start with the introduction section, which will give the reader context and a summary of the scope. Developers and testers may proceed to the technical requirements section.

## Product Scope

This project is based on designing and implementing an online platform for secondhand book sales and exchanges tailored for university students and also for anybody who wants to exchange or sell their book. The main objective of the platform is to make textbooks and educational resources more affordable and accessible, eliminating constraints like the high costs of new books, the limited availability of specific editions, and the lack of a centralized system for buying, selling, or exchanging books. Our goal is to promote sustainability by reusing books, enhance convenience for students through an easy-to-use interface, and provide a secure, efficient, and maintainable system that addresses the daily challenges faced by students in acquiring educational materials.

## References

IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.

IEEE Std 730-1998, IEEE Standard for Software Quality Assurance Plans.

IEEE Std 2089-2021, IEEE Standard for an Age-Appropriate Digital Services Framework Based on the 5 Rights Principles for Children

# Overall Description

## Product Perspective

The purpose of the University Second-Hand Book Sales and Exchange Platform is to create an efficient and secure online marketplace where students can buy, sell, and exchange used books. Reducing the cost of expensive books also encourages student book reuse & sustainability. In addition to this, we want to create a system that is more than sufficient and extremely beneficial to users. Although there are already several competitive systems on the software market, we are confident that our system will go one step further and present fresh viewpoints

## Product Functions

Although there exist similar systems in the software market that serve a lot of people, most second-hand book exchange platforms today have significant limitations, lacking advanced features that could improve user experience and efficiency. Most platforms only allow users to buy and sell books but do not support book exchanges, limiting affordability and flexibility. Our system supports these features that enable users to swap books based on mutual interests, making it a more cost-effective solution for students. Our system will have real-time messaging, where they can use our device to message each other. We will have a location-based search function to make it easier for the user to find books available nearby. The main functions are listed below:

1. User Management

* Account Registration and login
* Profile editing (name, email, phone, etc....)

1. User-Centric Marketplace:

* List and search for books
* Filters for price, location, and academic relevance (e.g., course codes, departments).

1. Listing Management

* Creating a new listing for books
* Editing or deleting existing listings

1. Location-Based Search:

* Help the user to find nearby books

1. Exchange book Requests

* Able to offer a listing in exchange for another user’s listing
* Accepting or rejecting offers

1. Notifications

* Sending confirmations or alerts

1. In-App Communication:

* User-Centric Marketplace-time chat for buyers and sellers to negotiate

1. Admin

* A web-based dashboard for admins to manage users, approve/reject book listings, resolve disputes, and display transaction analytics.

1. Free book

* The system will provide an option for the user if they want to give the book for free as a PDF or EPUB.

## User Classes and Characteristics

The system supports two main actors: User (student or any Interested Individual), and Admin.

**User (student or any interested individual):** Our system is basically for everyone; it can be a student or any individual that needs a book. The user can exchange/sell or buy the book they want

**Admin:** The system Admin is the backbone of our platform. Using a web-based dashboard, they keep everything running smoothly. From monitoring user accounts and book listings to tracking transactions and analyzing data and giving reports of book exchanges, etc., the admin has a clear view of the entire system. They also have the authority to approve or reject book listings and remove any account that does fraudulent things in our system, ensuring that every exchange on the platform meets our quality and safety standards.

## Operating Environment

The requirement for this type of software is that the user of the system must have an internet connection. Our system will be a mobile application that will run on iOS version 14.0+ and Android version 8.0+. It is developed using a cross-platform framework, React. The application will communicate with the backend server for data retrieval, user authentication, and transaction processing.

## Design and Implementation Constraints

All of the development must follow security policy for our system, including encrypted data transmission and single sign-in integration for user authentication. Users are required to have an internet connection to use the website.

Mobile Device Constraint: The minimum hardware requirement for the phone is 4GB of RAM and 150MB of storage.

## User Documentation

The application will come with a hands-on tutorial for first-time users. An FAQ section will be included in the mobile application.

## Assumptions and Dependencies

The application will rely on single sign-in and stable network connectivity. Whether you are using a mobile device, it is anticipated to have an internet connection.

# External Interface Requirements

## User Interfaces

1. **Mobile App (React Native):**

* **GUI Standards:** Material Design guidelines for consistent UI elements and user experience.
* **Screen Layout Constraints: Responsive** grid layout for varying device sizes (iOS/Android).
* **Common Elements:**
  + Navigation bar with icons for Home, Search, Messages, and Profile.
  + Standard buttons such as "Post Book Listing," "Search," "Add to Wishlist," and "Chat."
  + Error messages are displayed as toast notifications or popups with emphasized text.
  + Keyboard shortcuts (e.g., "Enter" to submit forms).
* **Sample Screens:**
  + Login: App logo, email/password fields, "Forgot Password" option.
  + Book Listing: Book pictures, price, distance, "Contact Seller," and “Details” option.
  + Seller Chat: Bubbles for messages, text input field, "Send" button, and “Seller info” button.

1. **Admin Panel (ASP.NET):**

* Dashboard with views for users, books, transactions, and analytics.
* Moderation tools to approve new book listings and flag and remove inappropriate content, fake book listings, etc.

Note: UI/UX mockups are documented in the UI Specification.

## Hardware Interfaces

* **Mobile Devices:**
  + **Supported Hardware:** Smartphone**s** running iOS version 14.0+ or Android version 8.0+.
  + **Hardware Requirements:**
* GPS to enhance book search using nearby book listings.
* Camera to take and upload book images.
* Internet connectivity (Wi-Fi/4G/5G).
* Touchscreen to navigate and input data.
* Speaker to notify users about new messages, exchange requests etc.
* **Server-Side:**
  + Cloud hosting with a minimum of 4GB RAM and 2 vCPUs.
  + 50GB SSD storage for the PostgreSQL database.

## Software Interfaces

* **Backend Services:**
  + Node.js v18 + Express.js (RESTful API) for user authentication and book listing operations.
  + PostgreSQL v15 databaseto store user data, book details, and transactions.
* **Third-Party APIs:**
  + Firebase Authentication API (OAuth, email/password) and Firebase Cloud Messaging to implement push notifications for messages and exchange requests.
  + Google Maps API for location-based search.
* **Shared Data:**
  + User profiles (ID, email, location) shared across auth and book modules.

## Communications Interfaces

* **Protocols:**
  + Use HTTPS (TLS 1.3) for secure unit integration and API calls.
  + WebSocket for real-time chat feature.
* **Data Formats:**
  + JSON format for API requests and responses.
  + Images (JPEG/PNG, max 5MB size).
* **Standards:**
  + RESTful API design (OpenAPI 3.0 specification).
  + GDPR compliance for EU user data encryption.

# System Features

## User Registration & Authentication

**4.1.1 Description and Priority**

* Let users create accounts, log in, and reset their account password.
* Priority: High (critical for security and user management).

**4.1.2 Stimulus/Response Sequences**

1. The user enters an email and password for account registration. The system validates user input, checks password strength, and creates the account.
2. The user enters an email and password to log in. The system authenticates the user and creates a temporary JSON Web Token (session) for the user and grants access to the application.
3. The user forgets the password. The system sends a reset link to the user's registered email address.

**4.1.3 Functional Requirements**

* **REQ-1:** The system shall validate email format (e.g., user@domain.com).
* **REQ-2:** Passwords must be 8+ characters with uppercase, lowercase, and numbers.
* **REQ-3:** JSON Web Tokens expire after 24 hours for security, prompting the user to log in again.

## Book Listing & Search

**4.2.1 Description and Priority**

* Users can list books for sale/exchange and search using filters.
* Priority: High (core functionality).

**4.2.2 Stimulus/Response Sequences**

1. The seller uploads details of the book he wants to list for sale/exchange, like book title, author, and picture of the book, along with any relevant tags like course codes or genre. The system saves details along with the GPS location of the seller in the database.
2. The prospective buyer searches for the book he wishes to exchange, e.g., "Calculus Textbook." The system presents matches within a particular distance, e.g., 10km, in the search listing.

**4.2.3 Functional Requirements**

* **REQ-4:** Location-based search uses device GPS (accuracy ±100m).
* **REQ-5:** Filters include price range and genre or course code (e.g., "MATH101") for course books.

## Book Exchange System

**4.3.1 Description and Priority**

* Users propose exchanges; the system matches based on book value and preferences.
* Priority: Medium (innovative feature).

**4.3.2 Stimulus/Response Sequences**

1. A user initiates a book exchange proposal by selecting a book to exchange from the search listing.

The system notifies the seller of the book with a push notification.

1. The system opens a chat interface for the buyer and seller to negotiate the price or any specifics.

The discussion occurs in real time as the application supports real-time messaging.

**4.3.3 Functional Requirements**

* **REQ-6:** Users can initiate, accept, or decline a book exchange proposal
* **REQ-7:** Users can talk with each other in real-time using the in-app messaging feature.
* **REQ-8:** Unsuccessful exchange proposals expire after 7 days.
* **REQ-9:** Users can rate their book exchange from 1 to 5 stars after the transaction.

## In-App Messaging

**4.4.1 Description and Priority**

* Real-time chat between buyers/sellers using Firebase.
* Priority: High (critical for user interaction).

**4.4.2 Stimulus/Response Sequences**

1. The buyer and seller negotiate the terms of the exchange. If the buyer agrees to the exchange, he selects the Exchange button and confirms his purchase.
2. If the exchange is successful, the system prompts the buyer to provide his address or meeting place to receive the item.
3. If the exchange is not successful, the system closes the book exchange proposal.

**4.4.3 Functional Requirements**

* **REQ-10:** Ensure user authentication (OTP verification) before letting users access real-time chat.
* **REQ-11:** The application should handle issues like weak Wi-Fi or cellular signals, and potential loss of internet connectivity. It should attempt automatic reconnection if the connection is dropped.
* **REQ-12:** Maintain a history of exchanged messages for each user conversation for at least 30 days.
* **REQ-13**: Typing indicators show in real-time when a user is composing a message.

## Admin Panel

**4.5.1 Description and Priority**

* Dashboard for managing users, books, and analytics.
* Priority: High (essential for moderation).

**4.5.2 Stimulus/Response Sequences**

1. The admin deletes the book listing with inappropriate content; the system removes it from the listing.
2. The admin bans user accounts for repeated infringement of terms of service.

**4.5.3 Functional Requirements**

* **REQ-14:** Analytics show daily active users, anonymized locations of users, and transaction volume.
* **REQ-15:** Admins can ban users and delete book listings that violate terms of service.

## Wishlist and Favorites

**4.6.1 Description and Priority**

* Let users save books they are interested in for future reference.
* Priority:  Medium (Enhances user experience but is not critical.).

**4.6.2 Stimulus/Response Sequences**

1. A user views a book listing and clicks the "Add to Wishlist" button. The system saves the book to the users’ Wishlist.
2. A user opens their Wishlist to see saved books. The system retrieves and displays the list.
3. A user removes a book from their Wishlist. The system updates the list.

**4.6.3 Functional Requirements**

* **REQ-16:** Users can add/remove books from their Wishlist.
* **REQ-17:** Wishlist items persist in user accounts even after logging out.

## Location-Based Search

**4.7.1 Description and Priority**

* Let users find books based on their current or specified location.
* Priority: High (essential for exchanges).

**4.7.2 Stimulus/Response Sequences**

1. A user searches for "math textbook" and sets a 5km range. The system displays books within 5km.
2. A user manually inputs a location (e.g., campus name). The system filters the search results.

**4.7.3 Functional Requirements**

* **REQ-18:** Users can adjust the search radius (within 1km, 5km, 10km, etc.).
* **REQ-29:** If GPS is not available, the system prompts the user to enter or mark his location on map.
* **REQ-20:** Search results are sorted by distance (nearest first).

## Book Ratings and Reviews

**4.8.1 Description and Priority**

* Let users rate and review books after successful trades to help other users make informed decisions.
* Priority: Medium (value-added feature, not necessary for book exchange).

**4.8.2 Stimulus/Response Sequences**

1. After a successful exchange, the system prompts the buyer to rate the book and leave a user review.
2. A user views a book listing and checks the average rating and reviews from past buyers.

**4.8.3 Functional Requirements**

* **REQ-21:** Ratings are anonymous but verifiable (only users who completed a transaction can rate).
* **REQ-22:** Reviews can include text (max 500 characters) and ratings.
* **REQ-23:** The system flags inappropriate reviews for admin moderation.

## Book Recommendations

**4.9.1 Description and Priority**

* Suggests relevant books based on user activity, wish lists, and past exchanges.
* Priority:   Low (Nice to have, not mandatory for functionality)

**4.9.2 Stimulus/Response Sequences**

1. A user frequently searches for "computer science" books. The system recommends similar listings.
2. A user adds books to the Wishlist. The system suggests related books (same author, course, or genre).

**4.9.3 Functional Requirements**

* **REQ-24:** Recommendations are based on search history, Wishlist, and completed exchanges.
* **REQ-25:** Users can dismiss recommendations they’re not interested in.
* **REQ-26:** Recommendations update in real-time as user preferences change.

# Other Nonfunctional Requirements

## Performance Requirements

Performance requirements for the application are defined to ensure efficient functioning under various conditions. The requirements involve response time, efficiency, execution time, and storage capacity. Efficient response time. The execution time for all user interface use cases should be around 2 seconds. Other performance requirements, including memory, storage, and processing, must follow the industry-recommended practices to ensure everything is working efficiently. The application should be able to support about 1000 users concurrently without any decrease in the application's performance. Background processes will be optimized to run efficiently without affecting real-time user usage and interactions. With a maximum round-trip time of 100 milliseconds, the system must maintain minimal network latency. The application will be designed to manage and store data efficiently, ensuring minimal database fragmentation.

## Safety Requirements

The purpose of safety requirements is to ensure that the system operates safely under differen conditions and situations. The system will implement regular and automated data checks and backup procedures for user data. The system should not cause any harm to end-users’ hardware or software components and should be free of any viruses, unnecessary ads, etc. Users should not be exposed to any security risks or malware through the website. It will ensure a quick and reliable recovery process in case of system failure and data loss. The system will have regular database optimization processes so that system performance is maintained and it does not crash. The application will be implemented with performance monitoring tools to track system performance, identify bottlenecks, and optimize as necessary. Log and monitor user activities for troubleshooting and for protecting the system. To achieve these aims, we are planning to use anti-virus software and run several testing methodologies using testing tools and perform these tests frequently.

## Security Requirements

* Security requirements are meant to ensure that the system is secure and protected against unauthorized access, data theft, and other security threats.
* The applicant should comply with all security and data protection regulations and standards.
* Data is encrypted when it is stored or when it is in use, using SSL/TLS protocols.
* The application will use privacy and cookies.
* The application will be integrated with user authentication and authorization mechanisms, which are the best in the industry and are robust.
* The system will not accept fewer than 8 characters without a special key (@$#) for the password.
* Verification will be done for every transaction to prevent fraud.
* ReCAPTCHA enables web hosts to distinguish between human and automated access to websites by giving some tasks, like unclear letters, and asking the user to write them.
* These kinds of tasks are easy for humans and difficult for bots.

## Software Quality Attributes

By focusing on software quality attributes, developers can ensure that their software meets certain standards of quality and usability. And can provide a better user experience.

### Usability:

The system should be easy to use and meet the user's needs. In particular:

* The system must be easy to learn for users and should have a similar user experience to other marketplace applications.
* A usable system should be intuitive and provide clear feedback to users.T
* The system should be user-friendly, well graphically designed, and easy to use.
* The app must comply with WCAG 2.1 accessibility standards, including support for screen readers and high-contrast modes.
* The system should have a tutorial for users to go over every function of the application.
* The application should have a system uptime of up to 99.9% to ensure reliability.
* A reliable backup and recovery system should be in place.
* The application is designed to accommodate any increase in the number of users.
* It should be capable of scaling horizontally to accommodate a growing number of data records.

**5.4.2 Supportability:**   
Code should be well-written and follow all coding standards.  
Updates and patches should be regular to maintain the application without disrupting normal applications.  
Customer support channels cater to their problems.  
Maintenance of the system should be cost-efficient.   
The application should be compatible with all types of mobile platforms (iOS, Android).   
  
**5.4.3 Testability:**   
This refers to how easily the system can be tested for defects and bugs. A testable system should have a clear and consistent architecture, with well-defined interfaces and testing procedures. We will test our website using different testing tools available, such as Postman and W3C, to check that there are no broken links or missing tags in the HTML code.   
  
**5.4.4 Maintainability:**   
This refers to how easy it is to make changes to the system and fix defects. A maintainable system should be modular and well-documented, with clear separation of concerns.   
Since our system is intended to scale up in the future, the website shall remove all the back-end complexities for in-house engineers to make changes to the system in the future.   
  
**5.4.5 Compatibility:**   
The software system should be able to work effectively and correctly with other systems, software, hardware, and environments. It involves ensuring that the software can operate without conflicts, errors, or failures with different operating systems, hardware, browsers, other software, and internationalization factors.

## Business Rules

**5.5.1 User Authentication:**

Only university students and staff are able to register an account by only using their university email for verification. Only admins who are authorized can see the user data. This will help protect user privacy and will ensure compliance with data security standards in the industry.

**5.5.2 Privacy/Data Accuracy:**

Users’ data (email, contact, and payment info) should remain confidential. Users can add data and delete accounts at any time. User Information stays safe, not saleable to third parties

**5.5.3 Book Listing Rules**

Users can list only academic-related books (textbooks, reference books, research materials). Each book listing must include title, author, edition, condition, price, and a clear image. A user cannot list duplicate books within a 24-hour window. Books should be categorized properly (e.g., Science, Engineering, Business, Arts). Listings expire after 90 days unless renewed by the user. Prices should be reasonable according to market value. Exchanges can be made through direct messages between users.

# Other Requirements

## Resource Utilization:

Applications should be optimized to use system resources efficiently, minimizing effects on device battery and memory.

## System Integration Requirements:

System should be linked with university authentication systems for sign in to make sure Student ID is verified. For transactions system should be integrated with main payment gateways in the world. For deliverables, the system should be linked with main postal services in the world like Amazon and DHL).

**Appendix A:**

CSS - Cascading Style Sheets

HTML— Hyper-Text Markup Language

SRS- —Software Requirements Specification

DB – Database

SQL— Structured Query Language

RAM— Random Access Memory

HTTP - Hyper Transfer Protocol

TRNC - Turkish Republic of Northern Cyprus

JavaScript— Programming language

**Appendix B: Analysis Models**

**Appendix C: To Be Determined List**

Source: <http://www.frontiernet.net/~kwiegers/process_assets/srs_template.doc>