

# Software Engineering Project

*Project short name: Lib++ Digital Library*

## Software Requirements Specification

20210808051	Burak Dere
20220808040	Deniz Eren Arıcı
20230808619	Enes Burak Atay

**Team Leader:** Enes Burak Atay  
**Product Owner:** Deniz Eren Arıcı

**Instructor:** Prof. Ümit Deniz ULUŞAR

19.03.2025

This report is submitted to the Department of Computer Engineering of Akdeniz University of the Software Engineering course CSE332.

Abbreviations	
IP	Internet Protocol
LMS	Library Management System
API	Application Programming Interface
ISBN	International Standard Book Number
SPV	Supervisor
UI	User Interface

# Contents

<b>1</b>	<b><i>Introduction.....</i></b>	<b><i>1</i></b>
1.1	Purpose .....	1
1.2	Product Scope.....	1
1.3	References.....	1
<b>2</b>	<b><i>Overall Description .....</i></b>	<b><i>1</i></b>
2.1	Product Perspective .....	1
2.2	Product Functions.....	2
2.3	User Types and Characteristics .....	2
2.4	Operating Environment .....	3
2.5	Design and Implementation Constraints .....	3
2.6	User Documentation.....	3
2.7	Assumptions and Dependencies .....	3
<b>3</b>	<b><i>External Interface Requirements .....</i></b>	<b><i>4</i></b>
3.1	User Interfaces.....	4
3.2	Hardware Interfaces .....	5
3.3	Software Interfaces .....	5
3.4	Communications Interfaces .....	5
<b>4</b>	<b><i>System Features / Requirements .....</i></b>	<b><i>5</i></b>
4.1	User Class 1 Librarian.....	5
4.1.1	System Feature 1 Manage Library System .....	5
4.2	User Class 2: Supervisor .....	5
4.2.1	System Feature: Manage Book Catalog .....	5
<b>5</b>	<b><i>Use Cases.....</i></b>	<b><i>6</i></b>
5.1	Creating a new account .....	6
5.2	Deleting an account .....	6
<b>6</b>	<b><i>Nonfunctional System Requirements .....</i></b>	<b><i>6</i></b>
6.1	Performance Requirements .....	6
6.2	Safety Requirements .....	7
6.3	Security Requirements.....	7
6.4	Software Quality Attributes .....	7
6.5	Business Rules .....	8
<b>7</b>	<b><i>Other Requirements.....</i></b>	<b><i>8</i></b>
<b>8</b>	<b><i>References.....</i></b>	<b><i>8</i></b>
<b>9</b>	<b><i>Appendix A: Glossary .....</i></b>	<b><i>8</i></b>

# 1 Introduction

## 1.1 Purpose

The purpose of this document is to provide a comprehensive description of the software requirements for the **Library Management System (LMS)**. This system will help both individuals and enterprises manage their library collections efficiently. The document outlines the functional and non-functional requirements, system constraints, interfaces, and interactions between the mobile app, web app, and backend system. The document will serve as a guide for the development team to create the first version of the system, and it is intended for approval by stakeholders.

The **LMS** includes a mobile app version tailored to personal users, which enables them to scan barcodes, search for book data (ISBN-based via Google Books API), and manage book categories with backup functionality. The **web app version** is designed for enterprise users, helping manage multiple library branches, borrowers, and book inventories.

This document is crucial for understanding how the system works across both the mobile and web versions and will ensure that the system meets both technical and business requirements for a successful deployment.

## 1.2 Product Scope

The **Library Management System (LMS)** is a software solution aimed at managing books, borrowers, staff, and library branches. The mobile version of the application is designed for personal use, providing features such as barcode scanning, ISBN lookups from the Google Books API, local data storage, offline access, and the ability to back up data to a server.

For enterprise-level use, the web version enables libraries or businesses to manage collections, track borrowed books, organize multiple branches, and monitor borrower activities. Both versions synchronize data with a centralized server to maintain consistency across platforms.

Key benefits of the system include:

- **Personal Users (Mobile App):** Simplified book management, the ability to scan and retrieve book details, offline functionality, and easy backup to a cloud server.
- **Enterprise Users (Web App):** Centralized management of multiple branches, tracking of borrower activity, and improved operational efficiency in handling library collections.

By leveraging modern mobile and web technologies, this system aims to improve the ease of managing library resources and provide a seamless user experience for both individual and organizational users.

## 1.3 References

It is planned to use the frameworks and applications stated below:

- |                                   |   |
|-----------------------------------|---|
| ➤ Google Books API documentation: | : <a href="https://developers.google.com/books">https://developers.google.com/books</a> |
| ➤ Firebase Documentation:         | : <a href="https://firebase.google.com/docs">https://firebase.google.com/docs</a>       |
| ➤ React Framework Documentation   | : <a href="https://reactjs.org/docs">https://reactjs.org/docs</a>                       |
| ➤ PostgreSQL                      | : <a href="https://www.postgresql.org/">https://www.postgresql.org/</a>                 |
| ➤ Pgadmin                         | : <a href="https://www.pgadmin.org/">https://www.pgadmin.org/</a>                       |
| ➤ JWT Authentication              | : <a href="https://jwt.io/introduction">https://jwt.io/introduction</a>                 |

# 2 Overall Description

## 2.1 Product Perspective

The **Library Management System (LMS)** consists of two primary components: a **mobile application** and a **web application**. The mobile app is designed for personal users to manage their own book collections, scan barcodes to retrieve book information, categorize books, and back up data to a remote server. The web app, on the other hand, is intended for enterprise-level library management, where administrators and librarians can manage multiple branches, oversee borrower activities, and track inventory across different locations.

Both the mobile and web applications will interact with a centralized **database** that stores information on books, borrowers, staff, and branches. The mobile app will primarily read data from the database, while the web app will allow users to perform CRUD (Create, Read, Update, Delete) operations on the data, including adding new books, managing borrowers, and adjusting library settings.

An important feature is the **synchronization** of data between mobile and web applications through a central server, ensuring that both systems remain up-to-date and consistent across platforms. The mobile app will operate offline with the ability to sync data when connected to the internet, while the web app requires a constant internet connection for real-time management and updates. The system must also interact with the **Google Books API** for fetching additional book information based on ISBN, and it must ensure compatibility with both **iOS** and **Android** platforms for the mobile app.

## 2.2 Product Functions

The **Library Management System** will provide the following functions for both mobile and web applications:

### Mobile Application:

With the mobile application, the users will be able to keep track of their books and manage their library. They will have the ability to add books in ways like scanning barcodes of books, searching from web with dedicated ISBN, adding manually and therefore managing their system with their own like.

- **Barcode Scanning:** Users can scan ISBN barcodes to fetch book information from the **Google Books API**.
- **Book Search:** Users can search for books by title, author, or ISBN and view detailed information.
- **Book Categorization:** Personal users can categorize their books into different sections (e.g., fiction, non-fiction, history, etc.).
- **Local Storage and Backup:** Books and borrower data are stored locally on the device, with the ability to back up and sync with the cloud server.
- **Offline Mode:** The mobile app can function without an internet connection and will synchronize data with the server when connectivity is restored.

### Web Application:

For the web application side, it will be provided for librarians and supervisors to manage the library with the abilities like borrowing books, adding/deleting them from system, and managing library branches.

- **Library Management:** Admins and librarians can manage library collections, including adding new books, removing outdated books, and categorizing books.
- **Borrower Management:** Web administrators can manage borrower records, track borrowed books, and generate overdue reports.
- **Branch Management:** Enterprises can manage multiple library branches, view inventory, and allocate books across locations.
- **Reporting:** Generate reports on book usage, borrower statistics, overdue books, and inventory management.
- **User Role Management:** Supervisors and admins can manage user roles, such as assigning librarian, borrower, or administrator permissions.

## 2.3 User Types and Characteristics

The **Library Management System** will support several different user types with distinct responsibilities and requirements:

1. **Personal Users (Mobile App):**
  - **Primary Functions:** Scan books, categorizing collections, back up data, and track borrowed books.
  - **Technical Expertise:** Minimal. The mobile app should be user-friendly with a focus on ease of use.
  - **Usage Frequency:** Moderate, depending on the user's interest in managing their personal library.
2. **Enterprise Users (Web App):**
  - **Primary Functions:** Manage library inventory, track borrowers, manage multiple branches, and generate reports.
  - **Technical Expertise:** Moderate to high. Web users need to understand library management processes and should be comfortable with system administration tasks.
  - **Usage Frequency:** High, as the web app is designed for active use by library staff and administrators.
3. **Supervisors/Administrators (Web App):**

- **Primary Functions:** Oversee the overall system, manage users, manage library branches, and monitor system health.
- **Technical Expertise:** High. Supervisors/administrators need to have good technical knowledge to oversee the system's proper functioning.
- **Usage Frequency:** High, as they are responsible for managing day-to-day library operations.

## 2.4 Operating Environment

The **Library Management System** will operate in the following environments:

- **Mobile Application:**
  - **Platform:** Android (version 7.0 and above)
  - **Required Hardware:** Devices with a minimum of 2GB of RAM and 100MB of free storage space.
  - **Required Software:** The mobile app will be built using **React Native** or **Flutter** for cross-platform compatibility.
- **Web Application:**
  - **Platform:** Desktop or laptop computers with access to modern web browsers (Chrome, Firefox etc.).
  - **Required Software:** **Node.js** and **React** for the front end; **PostgreSQL** for the database.
  - **Operating System:** The web app will function on well-known operating systems, including **Windows**, **macOS**, and **Linux**.
- **Server Environment:**
  - **Hosting:** AWS or any cloud-based infrastructure for hosting the system backend and database.
  - **Backend Technology:** **Node.js** with **Express.js** for handling API requests.

## 2.5 Design and Implementation Constraints

Several constraints must be considered during the development of the **Library Management System**:

- **Internet Connection:** The web application and the mobile app will require an internet connection to fetch and synchronize data with the centralized database. The mobile app should support offline functionality, allowing users to continue accessing their library information even when not connected to the internet. Once an internet connection is available, the app will automatically sync.
- **Database Scalability:** Both the mobile and web applications will access the same centralized database. The database should be able to handle concurrent read and write requests from both platforms. There may be performance bottlenecks if the system is not optimized for scaling.
- **Memory and Storage Constraints (Mobile):** The mobile app must be optimized to minimize memory usage, as devices with limited resources may struggle with performance. The app will only use up to **100MB** of storage, with efficient data management practices in place.

## 2.6 User Documentation

The following user documentation will be provided:

- **User Manual:** A comprehensive guide detailing how to use both mobile and web apps, including installation, setup, and usage instructions.
- **Tutorials:** Step-by-step tutorials for common tasks, such as scanning books, adding borrowers, and managing library branches.
- **API Documentation:** For any third-party integration, such as the Google Books API, detailed API documentation will be provided.

## 2.7 Assumptions and Dependencies

**Mobile Device Performance:** It is assumed that personal users will have mobile devices with sufficient processing power and memory (minimum of 2GB RAM) to run the app smoothly. If the user's device does not meet these specifications, the app may perform sub optimally or fail to function.

**Third-Party Services:** The system will rely on third-party APIs, particularly the **Google Books API** for fetching book information. Any changes or disruptions to these services could affect the functionality of the system.

**Internet Connectivity:** The system assumes that users have access to a stable internet connection for data synchronization between the mobile app and web app. In cases of poor internet connectivity, the mobile app should provide offline functionality, though syncing might be delayed.

### 3 External Interface Requirements

#### 3.1 User Interfaces

The system will have two primary interfaces based on the platform: mobile and web apps.

##### Mobile App User Interface:

1. **Login Page (First-time User):** When a user opens the mobile application for the first time, they will be directed to the login page. They can either log in or register a new account if they haven't already.
2. **Search Page (Returning User):** If the user is a returning user, they are directed to the search page. On this page, the user can choose from various search options (e.g., ISBN, author, book title, etc.).
3. **Profile Page:** Every user will have a profile page where they can edit their email address, phone number, and password.

##### Web App User Interface:

1. **Login/Registration:** Similar to the mobile app, users can log in or register.
2. **Admin Dashboard:** Administrators (library managers) will have access to the dashboard, where they can manage books, users, and branches.
3. **Search Functionality:** The web app will allow users to search for books, manage borrowed books, and track branch inventories. (Figure 1)
4. **Branch and Borrower Management:** The web version will also allow the management of borrowers and store/branch information. (Figure 2)

The mobile and web apps will both adhere to basic GUI standards, with a consistent layout, standard buttons (e.g., Submit, Cancel), and help functions available across all screens.

ISBN	Title	Author	TotalCopies	Left	BranchName
9789750802942	Harry Potter ve Felsefe Taşı	J. K. Rowling	1	0	Lara
9789750802942	Harry Potter ve Felsefe Taşı	J. K. Rowling	1	0	Meltem
9789750802959	Harry Potter ve Sırlar Odası	J. K. Rowling	1	0	Lara
9789750803116	Harry Potter ve Azkaban Tutsağı	J. K. Rowling	1	1	Uncalı
9789750803314	Harry Potter ve Ateş Kadehi	J. K. Rowling	1	1	Lara
9789750806452	Harry Potter ve Zümrüdüanka Yoldaşlığı	J. K. Rowling	1	1	Meltem
9789750809958	Harry Potter ve Melez Prens	J. K. Rowling	1	1	Uncalı
9789750812989	Harry Potter ve Ölüm Yadigarları	J. K. Rowling	1	1	Meltem
9789750837609	Harry Potter ve Lanetli Çocuk	J. K. Rowling	1	1	Meltem

Figure 1: Search

Search for books or articles...

### Show a borrower

ID: 128220  
Name: Abigail Lopez  
Phone: 905575264397

ISBN	Title	BorrowDate	DueDate	BranchID	DaysPassed
9786052993019	MOMO	2024-2-12	2024-3-15	1	32
9786053759584	Ben Kirke	2024-4-22	2024-4-26	4	4
9789752750715	İnanmanın Gücü	2023-4-16	2023-4-20	3	4

Lend a book

Figure 2: Borrower

## 3.2 Hardware Interfaces

The system is designed to work on the following hardware interfaces:

- **Mobile Devices:** The mobile app will be compatible with smartphones running Android. It will use the device's camera for barcode scanning and interacting with the mobile OS for authentication and notifications.
- **Web Servers:** The web version will run on enterprise-level servers with the necessary capacity to manage large volumes of books, users, and transactions.
- **Barcode Scanners:** In addition to the mobile app's built-in barcode scanning capability, external barcode scanners may be used for both mobile and web applications.

## 3.3 Software Interfaces

**Mobile Application:** It is planned to use React Native for the initial plan, compatible with Android. The app will interact with Google's Library API to fetch book information based on ISBNs.

**Web Application:** The web app will be built using .NET environment. The backend will be connected to a relational database (PostgreSQL) to manage book data, user profiles, borrowings, etc.

**Google Library API:** The system will use Google's Library API to fetch book information via ISBN lookup. This API will be utilized by mobile apps and web apps to fetch accurate data for book records.

## 3.4 Communications Interfaces

**Mobile App:** The mobile app will use HTTP/HTTPS protocols to interact with the backend servers and Google Library API. For barcode scanning, the mobile device's camera will be used, with scanned data transmitted to the system for lookups.

**Web App:** The web app will use HTTP/HTTPS for communication with the server. It will also communicate with the Google Library API for ISBN-based lookups and manage book records and user transactions.

**Backup Services:** The mobile app will support data backup to a remote server. Communication between the mobile app and the server will be encrypted using SSL/TLS to ensure data privacy and security.

# 4 System Features / Requirements

## 4.1 User Class 1 Librarian

### 4.1.1 System Feature 1 Manage Library System

#### 4.1.1.1. Description and Priority

The administrator can manage all aspects of the library, including adding and removing books, managing users, and monitoring the inventory across branches. This feature is of **High Priority**.

#### 4.1.1.2. Stimulus/Response Sequences

- Stimulus: The administrator logs into the web app.
- Response: The system displays the library management dashboard with options to manage books, users, and branches.

#### 4.1.1.3. Functional Requirements

- FR1: The system should allow an administrator to log in and access a dashboard.
- FR2: The system should allow the administrator to add, update, or delete books from the library.
- FR3: The system should allow the administrator to add/remove branches and assign books to those branches.
- FR4: The system should allow the administrator to view and manage users, including borrowers.

## 4.2 User Class 2: Supervisor

### 4.2.1 System Feature: Manage Book Catalog

#### 4.2.1.1 Description and Priority

The SPV manages the day-to-day operations of the library, including checking in and out books, updating inventory, and assisting with borrower queries. This feature is of High Priority.



#### 4.2.1.2 Stimulus/Response Sequences

- Stimulus: The SPV logs into the system and accesses the book catalog.
- Response: The system allows the SPV to update book details or mark a book as borrowed/returned.

#### 4.2.1.3 Functional Requirements

- FR1: The SPV should be able to mark books as borrowed and returned.
- FR2: The system should allow the librarian to update book details (title, author, category).
- FR3: The SPV can check the availability of books across branches.

## 5 Use Cases

### 5.1 Creating a new account

The purpose of this use-case is to describe the procedure for creating an account in the system.

**Pre-conditions:**

- The user has access to the system via mobile or web app.
- The user is not currently logged in.

**Post-conditions:**

- An account is created for the user, and they are logged in.

**Basic Flow:**

1. The user is on the homepage.
2. The user clicks on the "new account" link and is directed to the account creation page.
3. The user enters all required details (username, password, email, and contact details).
4. The user clicks the "Create" button.
5. If the username already exists, an error message appears, prompting the user to choose another username.
6. If the username is available, the system creates the account and presents a confirmation message, notifying the user that the account has been successfully created.

### 5.2 Deleting an account

The purpose of this use case is to describe the procedure for deleting an account from the system.

**Pre-conditions:**

- *The user is logged into their account.*
- *The user has administrative or personal permission to delete the account.*

**Post-conditions:**

- *The account is deleted from the system.*

**Basic Flow:**

1. *The user is on the account settings page.*
2. *The user clicks the "Delete Account" button.*
3. *The system prompts the user with a confirmation dialog to ensure they intend to delete the account.*
4. *If confirmed, the system deletes the account and removes all associated data (including any local backups or server-side storage).*
5. *The user is logged out, and they are redirected to the homepage.*

## 6 Nonfunctional System Requirements

### 6.1 Performance Requirements

**ID:** NF1

**TAG:** Response Time

**GIST:** The fastness of the search (Mobile app - ISBN search)

**SCALE:** The response time for searching ISBN

**METER:** Measurements obtained from 1000 searches during testing.

**MUST:** No more than 2 seconds 100% of the time.

**WISH:** No more than 1 second 100% of the time.

**ID:** NF2

**TAG:** Response Time for User Log-in

**GIST:** The fastness of the log-in process (Mobile app)

**SCALE:** The response time for log-in

**METER:** Measurements obtained from 1000 log-in attempts during testing.

**MUST:** No more than 2 seconds 100% of the time.

**WISH:** No more than 1 second 100% of the time.

**ID:** NF3

**TAG:** Response time for Barcode scan

**GIST:** The fastness of scanning barcode

**SCALE:** The response time for barcode scan

**METER:** Measurements obtained from 1000 scan attempts during testing.

**MUST:** No more than 2 seconds 100% of the time.

**WISH:** No more than 1 second 100% of the time.

**ID:** NF4

**TAG:** Response time for barrowing

**GIST:** The fastness of barrowing time

**SCALE:** The response time for barrow a book

**METER:** Measurements obtained from 20 barrow attempts during testing.

**MUST:** No more than 60 seconds 100% of the time.

**WISH:** No more than 45 second 100% of the time.

**ID:** NF5

**TAG:** Response time for backup

**GIST:** the speed of backup time

**SCALE:** the response time for backup

**METER:** Measurements obtained from 10 backup attempts during testing.

**MUST:** No more than 180 seconds 100% of the time.

**WISH:** No more than 120 second 100% of the time.

## 6.2 Safety Requirements

- The system must ensure that no critical data is lost during an unexpected shutdown or failure.
- Data backups for both the mobile and web app versions should be performed automatically every 24 hours on the server to prevent data loss.
- The system should provide an option for users to download their data backup manually at any time.

## 6.3 Security Requirements

- The system must support secure login for both mobile and web applications.
- Access control mechanisms must ensure that only authorized individuals (e.g., librarians, supervisors) can modify critical data such as book inventory and branch information.
- All APIs (including ISBN look-up through Google Library API) must be secured to prevent unauthorized access.

## 6.4 Software Quality Attributes

- **Adaptability:** The system must be designed to handle both small-scale libraries (personal use) and enterprise-scale libraries (web app for organizations).
- **Maintainability:** The codebase must be well-documented, and the system should support easy updates and bug fixes.
- **Interoperability:** The system should work seamlessly across mobile and web platforms with low functionality loss.
- **Usability:** The mobile app should be intuitive and easy to use, especially for non-technical users. The web app should be designed for enterprise environments with the necessary administrative features.
- **Scalability:** The system should be able to scale to accommodate libraries of various sizes, from personal collections to enterprise-level libraries with multiple branches.

## 6.5 Business Rules

- Only supervisors or authorized users can manage user accounts and permissions.
- Librarians have the ability to add, edit, or remove books from the system but cannot modify user permissions or system settings.
- Librarians are allowed to lend books to borrowers.
- The web app version must allow multiple library branches or stores to be managed under a single enterprise account.
- The system must support the categorization of books into various genres, topics, or types.

## 7 Other Requirements

- **Database Requirements:** The system should use a relational database (PostgreSQL) for storing user accounts, book information, borrowing records, and other system data.
- **Accessibility Requirements:** The system should allow the user to magnify the UI.
- **Backup and Disaster Recovery:** The system should provide options for cloud backups with redundancy, ensuring that user data is safe and recoverable in case of a system failure.

## 8 References

*Google Books API Documentation* for ISBN-based book lookup:  
<https://developers.google.com/books>

## 9 Appendix A: Glossary

- **ISBN (International Standard Book Number):** A unique identifier for books, used to retrieve detailed book information from external sources like Google Books API.
- **Librarian:** A system role responsible for managing books, users, and general library operations.
- **Supervisor:** A higher-level system role that has full access to user accounts, permissions, and library settings.
- **Borrower:** A person who can check out books but cannot modify the system's internal data or structure.
- **Branch:** A branch is the physical location that a company has for their library.