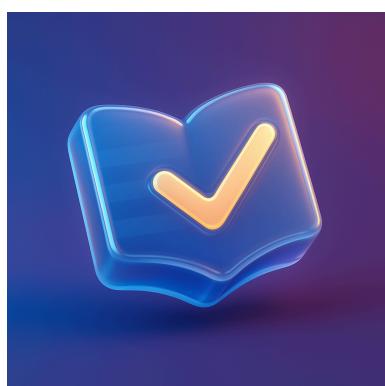


# **REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM**

**Issue No: 1.0**

**Issue Date: Nov 2025**



**Prepared by**  
**Ahmet Mücahit Gündüz**  
**Enes Korkmaz**  
**Onurhan Talan**  
**Rümeysa Kalay**  
**Veysel Cemaloğlu**

	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

<b>1. Introduction</b>	<b>3</b>
1.1 Purpose of The System	3
1.2 Scope of The System	3
1.3 Objectives and success criteria of the project	4
1.4 Definitions, acronyms, and abbreviations	6
1.5 References	7
1.6 Overview	7
<b>2. Current System</b>	<b>8</b>
<b>3. Proposed system</b>	<b>9</b>
3.1 Overview	9
3.2 Functional Requirements	9
3.3 Non-Functional Requirements	13
3.3.1 Usability	13
3.3.2 Reliability	14
3.3.3 Performance	14
3.3.4 Supportability & Maintainability	14
3.3.5 Implementation	15
3.3.6 Interface	15
3.3.7 Packaging & Deployment	16
3.3.8 Security	16
3.3.9 Data & Privacy	16
3.3.10 Legal & Compliance	16
3.4 System Models	17
3.4.1 Scenarios	17
3.4.2 Use case model	20
3.4.3 Object Model	47
3.4.4 Dynamic Model	48
3.4.5 User interfaces and Mock-ups	52
<b>4. Glossary of Terms</b>	<b>55</b>
<b>5. Traceability</b>	<b>56</b>
5.1 Traceability to Problem Statement	56
5.2 Traceability Summary	57

	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

## 1. Introduction

### 1.1 Purpose of The System

The purpose of the *Personal Library Management System (PLMS)* is to provide individuals with a centralized, intelligent platform to manage and enrich their personal collections of books and DVDs.

The system allows users to:

- Catalog physical media items efficiently.
- Integrate with online APIs (Google Books, IMDb) for detailed metadata.
- Track reading and viewing progress.
- Build customized lists (e.g., "Favorites," "Watch Later").

The PLMS aims to reduce inefficiencies such as lost items or duplicate purchases, while enhancing the enjoyment of personal media collections.

### 1.2 Scope of The System

The Personal Library Management System (PLMS) is designed to serve as a comprehensive tool for individuals to catalog, manage, and track their personal physical media collections, specifically books and DVDs.

The scope of the system includes the following core functional areas:

- **Inventory Management:** Full CRUD (Create, Read, Update, Delete) operations for media items (FR-1). This includes both manual data entry (FR-1.1) and automated entry via ISBN/barcode scanning (FR-1.2).
- **Organization:** Support for media categorization (e.g., Book, DVD) (FR-2) and flexible, user-defined tagging (FR-3).
- **List Management:** Creation and management of customized lists (e.g., "To Read," "Favorites") separate from the main library, including operations to add, remove, and prioritize items (FR-5).
- **Progress Tracking:** Granular tracking of media consumption (pages read for books, minutes watched for DVDs) with historical logging and visual indicators (FR-6).
- **Data Enrichment:** Integration with third-party APIs (such as Google Books, Open Library, and IMDb) to automatically fetch and link rich metadata like summaries(FR-8).
- **Search and Discovery:** Robust search, filter, and sort capabilities within the main library and within specific user lists (FR-4, FR-5.6-6.9).

	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

- **Personalization:** An AI-driven recommendation system that suggests new media based on the user's existing library and preferences, with explicit user consent (FR-9, NFR-DATA-1).
- **Authentication:** Secure user account management delegated to OAuth 2.0 identity providers (FR-7).

**Out of Scope:** The system's scope is explicitly limited and does not include:

- **Media Playback:** The PLMS is a *cataloging* and *tracking* system. It is not an e-reader, video player, or media streaming service.
- **Public Library Functions:** The system is designed for *personal* collections and does not support multi-user public-facing circulation, fines, or acquisitions management.
- **Password Management:** The system will not store or manage user passwords directly. All authentication and password recovery is fully delegated to external OAuth providers (FR-7.1).
- **AI Model Development:** The system will *integrate* with an external AI API for recommendations (FR-9.1). The development, training, or hosting of the AI model itself is out of scope.

### 1.3 Objectives and success criteria of the project

The primary objectives of the PLMS project are to deliver a system that is usable, reliable, secure, and maintainable. Success will be measured against the following specific, quantifiable criteria derived from the non-functional requirements.

#### **Objective 1: To provide a highly usable and responsive user experience.**

- **Success Criteria:**
  - The system shall provide a usable experience across mobile and desktop platforms without horizontal scrolling on primary flows (NFR-US-1).
  - Core tasks (search, add item, create list) shall be completable by 80% of new users within 10 minutes without documentation (NFR-US-3).
  - Median response time for common library actions (search, view details, sort) shall be  $\leq$  500 ms for libraries up to 2,000 items (NFR-PERF-1).
  - The 95th percentile response time for these same actions shall be  $\leq$  1500 ms (NFR-PERF-3).

#### **Objective 2: To ensure high availability and reliability of core services.**

- **Success Criteria:**
  - The system shall achieve  $\geq$  99.5% monthly uptime (NFR-REL-1).
  - Core CRUD and list operations shall remain fully operational even if external metadata or AI APIs are unavailable (NFR-REL-2).

	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

- Calls to external services shall implement exponential backoff and circuit breaker patterns to prevent cascading failures (NFR-IMPL-3).

**Objective 3: To securely manage user data and system communications.**

- **Success Criteria:**

- Authentication shall be handled exclusively via OAuth 2.0 (FR-7, NFR-SEC-1).
- All client-server and server-server communications shall use TLS 1.2 or later (NFR-IMPL-1).
- Sensitive user data shall be encrypted at rest (NFR-IMPL-2).
- API keys and other secrets shall be stored in a dedicated secrets manager (NFR-SEC-2).
- Security-relevant events shall be logged (NFR-SEC-4).

**Objective 4: To protect user privacy and ensure compliance.**

- **Success Criteria:**

- The system shall obtain explicit, auditable user consent before sending any identifiable data to external AI services (NFR-DATA-1, FR-9.3).
- The system shall comply with all usage and attribution terms of third-party APIs (NFR-LEG-1).

**Objective 5: To build a maintainable, scalable, and supportable architecture.**

- **Success Criteria:**

- The system shall be architected in modular components (frontend, API, connectors) to allow for future expansion (NFR-SUP-1).
- All code commits shall pass automated CI/CD pipelines, including linting and testing (NFR-SUP-3).
- The system shall be packaged as Docker containers (NFR-PKG-1).
- System performance shall degrade in a near-linear manner as library size increases (NFR-PERF-4).

	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

## 1.4 Definitions, acronyms, and abbreviations

### Acronyms and Abbreviations

- **API** – Application Programming Interface
- **CRUD** – Create, Read, Update, Delete
- **DVD** – Digital Versatile Disc
- **ISBN** – International Standard Book Number
- **OAuth 2.0** – Open standard for access delegation
- **UI** – User Interface
- **AI** – Artificial Intelligence
- **TLS** – Transport Layer Security
- **JSON** – JavaScript Object Notation
- **SRS** – Software Requirements Specification
- **PLMS** – Personal Library Management System
- **CI/CD** – Continuous Integration / Continuous Deployment

### Definitions (Short Technical)

- **Metadata** – External or descriptive data retrieved from APIs (e.g., title, summary, author).
- **Backoff Policy** – A timing strategy for retrying failed external calls.
- **Tail Latency** – 95th percentile response time measurement.
- **Circuit Breaker** – A mechanism that stops repeated calls to failing external services.

	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

## 1.5 References

**[PLMS-Problem]:** *Personal Library Management System - Problem Statement.* (Provided [Personal Library Management System.docx](#) document) .

**[Req-Writing-Guide]:** *CSE 443 - Requirements Specification Writing.* (Provided [03 - Requirements Specification Writing.pdf](#) document) .

**[Larman-OOAD]:** Larman, C. (2005). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design* (3rd ed.). Prentice-Hall.

**[API-GoogleBooks]:** Google Books API. (External API to be integrated by the system).

**[API-OpenLibrary]:** Open Library API. (External API to be integrated by the system).

**[API-IMDb]:** IMDb API. (External API to be integrated by the system).

**[IEEE-830]:** IEEE Std 830-1998 - *IEEE Recommended Practice for Software Requirements Specifications.*

**[ISO-29148]:** ISO/IEC/IEEE 29148:2018 - *Systems and software engineering — Life cycle processes — Requirements engineering.*

## 1.6 Overview

This document is the official Software Requirements Specification (SRS) defining the scope, features, and constraints for the Personal Library Management System (PLMS).

- **Section 1 (Introduction):** Introduces the purpose of the system, its scope, project objectives, and the terminology used in this document.
- **Section 2 (Current System):** Describes the problems and deficiencies of existing solutions and outlines how the PLMS will address these gaps.
- **Section 3 (Specific Requirements):** Details all requirements the system must meet. This section is the primary source for all system expectations:
  - **3.1 Use Cases:** Defines the main user scenarios and the interactions between actors (users) and the system to achieve specific goals.
  - **3.2 Functional Requirements:** Defines the specific actions the system must be able to perform (e.g., "The system shall allow the user to add a new item").
  - **3.3 Non-Functional Requirements:** Defines the quality attributes (e.g., performance, security, usability) and constraints of the system.

This document will serve as the primary reference for the design, development, and testing teams.

	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

---

## 2. Current System

Currently, individuals manage personal media collections using a fragmented set of tools, including manual methods (e.g., physical notebooks), generic office software (e.g., spreadsheets), or existing library applications. These approaches suffer from significant deficiencies that limit their effectiveness for personal media management.

The primary problems and gaps in existing systems are:

- **Lack of Personal Focus:** Most available library software is designed for large-scale institutional or public use, not for the specific needs of an individual owner. They lack features for tracking personal details like item condition, loan status, or purchase location.
- **No Progress Tracking:** Current systems do not offer functionality to track the consumption progress of media items. Users cannot log pages read for a book or minutes watched for a DVD, which is a key requirement for a personal library .
- **Limited Media Support:** Existing solutions are often rigid and focus exclusively on books. They lack the capability to manage a mixed-media collection, such as cataloging both books and DVDs within a single, integrated system.
- **No Custom Lists or Tagging:** Users are restricted by predefined categories. These systems typically lack features for flexible, user-defined tagging (e.g., "Favorites," "Genre") or the creation of dynamic, custom lists (e.g., "To Read," "Watch Later").
- **Absence of Content Enrichment:** Current solutions act as static databases. They do not automatically connect to external internet resources (like Google Books or IMDb) to enrich the collection with valuable metadata such as summaries details.
- **No Recommendation Features:** Existing tools do not analyze the user's collection to provide personalized recommendations, a feature the PLMS aims to include via AI integration.

The proposed Personal Library Management System (PLMS) is designed to solve these specific problems. It differentiates itself from current systems by providing an integrated, personally-focused platform that supports mixed media (books and DVDs), offers granular progress tracking, allows for flexible tagging and list creation, and enriches the user's collection with external metadata and personalized recommendations .

	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

### 3. Proposed system

#### 3.1 Overview

The proposed system is a **web-based media library management application** designed to allow users to organize, track, and enrich their personal media collections—including **books** and **DVDs**—through intuitive interfaces, intelligent recommendations. It provides core CRUD operations, user-defined categorization and tagging, advanced search capabilities, all while ensuring security, scalability, and maintainability.

#### 3.2 Functional Requirements

##### FR-1: Manage Media Items (CRUD Operations)

###### **FR-1.1 — Add Media Item (Manual Entry)**

The system shall allow users to add a new media item to the library by manually entering its information, so that the item is stored in the system's database.

###### **FR-1.2 — Add Media Item (ISBN/Barcode Scanning)**

The system shall allow users to add a new media item to the library by scanning its ISBN or barcode, so that the system automatically retrieves and saves the item's details.

###### **FR-1.3 — View Media Item Details**

The system shall allow users to view all stored details of a selected media item, so that users can access complete information about that item.

###### **FR-1.4 — Update Media Item Information**

The system shall allow users to update existing information of a selected media item, so that corrections or changes are reflected in the system's database.

###### **FR-1.5 — Delete Media Item (Single Item)**

The system shall allow users to delete a selected media item from the library, and upon deletion, the system shall automatically remove that item from all user lists it belongs to, so that obsolete or incorrect items are removed from both the library and any associated lists.

###### **FR-1.6 — Delete Multiple Media Items (Bulk Deletion)**

The system shall allow users to delete multiple selected media items at once, and upon deletion, the system shall automatically remove those items from all user lists they belong to, so that inventory cleanup operations are performed consistently across the library and associated lists.

---

##### FR-2: Categorization

	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

### **FR-2.1 — Categorize Media Items**

The system shall allow users to assign a category (e.g., Book or DVD) to each media item, so that items are organized based on their type.

## **FR-3: Tag Management**

### **FR-3.1 — Create Tag**

The system shall allow users to create a new tag by specifying a unique tag name, so that it can be used to categorize media items.

### **FR-3.2 — Update Tag**

The system shall allow users to modify the name of an existing tag, so that the updated tag name is reflected across all associated media items.

### **FR-3.3 — Delete Tag**

The system shall allow users to delete an existing tag, and upon deletion, the system shall automatically remove that tag from all media items to maintain data consistency.

### **FR-3.4 — Assign Tag to Item**

The system shall allow users to assign one or more existing tags to a media item, so that items can be classified using flexible labels.

### **FR-3.5 — Remove Tag from Item**

The system shall allow users to remove an assigned tag from a media item, so that the item is no longer associated with that tag.

## **FR-4: Search, Filter, and Sort (Library Context)**

### **FR-4.1 — Search Items in Library**

The system shall allow users to search for media items in the main library based on one or more properties (such as title, tag, genre, or author), so that specific items can be located efficiently.

### **FR-4.2 — Filter Items in Library**

The system shall allow users to filter media items in the main library according to selected properties (such as category, tag, or release date), so that only relevant items are displayed.

### **FR-4.3 — Sort Items in Library**

The system shall allow users to sort media items in the main library by selected properties (such as title, author name, or release date), so that the order of items can be adjusted according to preference.

	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

---

## FR-5: List Operations (User Lists Context)

### FR-5.1 — Create New List

The system shall allow users to create a new customized list, so that they can organize selected media items separately from the main library.

### FR-5.2 — Add Item to List

The system shall allow users to add a single media item to an existing list, so that individual items can be organized without affecting other items.

### FR-5.3 — Remove Item from List

The system shall allow users to remove a single media item from an existing list, and the system shall also automatically remove the item from all lists if it is deleted from the main library, so that list integrity is maintained.

### FR-5.4 — Remove Items from List (Multiple Items)

The system shall allow users to remove multiple media items at once from an existing list, and the system shall also automatically remove these items from all lists if they are deleted from the main library, so that list integrity is maintained across all user-defined lists.

### FR-5.5 — Delete List

The system shall allow users to delete an existing customized list, so that lists that are no longer needed are removed from the system.

### FR-5.6 — Sort Items in List

The system shall allow users to sort items within a list based on properties such as title, author, or release date, so that items can be ordered according to user preference.

### FR-5.7 — Prioritize Items in List

The system shall allow users to prioritize items within a list by manually adjusting their order or importance, so that high-priority items are easily accessible.

### FR-5.8 — Search Items Within a List

The system shall allow users to search for media items within a specific list using one or more properties, so that individual items can be quickly located.

### FR-5.9 — Filter Items Within a List

The system shall allow users to filter items within a list by properties such as category, tag, or genre, so that only relevant items are displayed within the list.

	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

---

## FR-6: Progress Tracking

### **FR-6.1 — Track Book Progress**

The system shall track the number of pages read and the percentage of total pages completed for each book, and calculate the number and percentage of remaining pages, so that users can monitor their reading progress.

### **FR-6.2 — Track DVD Progress**

The system shall track the number of minutes watched and the percentage of total duration completed for each DVD, and calculate the number and percentage of remaining minutes, so that users can monitor their viewing progress.

### **FR-6.3 — Progress Update Logging**

The system shall maintain historical logs of all reading or viewing progress updates, including the date and duration of each update, so that users can review their consumption history over time.

### **FR-6.4 — Visual Progress Indicators**

The system shall display visual indicators (such as progress bars or charts) for each media item to show the percentage of reading or viewing completed, so that users can quickly assess their progress at a glance.

---

## FR-7: User Account Management (OAuth 2.0)

### **FR-7.1 — User Registration / Account Linking**

The system shall allow users to register or link an existing account via an OAuth 2.0 identity provider (e.g., Google, Facebook), so that users can securely create or access an account. All account operations, including password management and recovery, are handled by the identity provider.

### **FR-7.2 — User Login via OAuth 2.0**

The system shall allow users to log in using OAuth 2.0 authentication, so that access to the system is secured through the identity provider.

---

## FR-8: External API Integration and Metadata Linking

### **FR-8.1 — Fetch Metadata for Books**

The system shall retrieve metadata for books (such as title, summary) from external APIs

	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

like Google Books API or Open Library, so that book entries contain comprehensive and up-to-date information.

#### **FR-8.2 — Fetch Metadata for DVDs**

The system shall retrieve metadata for DVDs (such as title, summary, release year) from external APIs like IMDb, so that video entries contain comprehensive and up-to-date information.

#### **FR-8.3 — Update Metadata from APIs**

The system shall refresh or update metadata for existing media items by fetching the latest data from the external APIs, so that the information remains current.

---

### FR-9: AI-Based Recommendation System

#### **FR-9.1 Recommendation Generation**

The system shall connect to an external AI API to generate personalized recommendations for books and DVDs based on existing user data, including interaction history, preferences, and library contents.

#### **FR-9.2 Recommendation Display**

The system shall display AI-generated recommendations within the user interface, grouped by media type (Books / DVDs).

#### **FR-9.3 User Consent and Data Privacy**

The system shall request user consent before sending user data to the AI API for recommendation generation and allow users to disable AI-based recommendations at any time.

## 3.3 Non-Functional Requirements

### 3.3.1 Usability

- **NFR-US-1:** The system shall follow a mobile-first responsive design. On portrait mobile ( $\leq 412\text{px}$  width) and desktop ( $\geq 1024\text{px}$  width) breakpoints, all primary UI flows (browse, search, view item, add item, access lists, view recommendations) shall be usable without horizontal scrolling and with all primary actions reachable

	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

within 3 taps/clicks.

- **NFR-US-2:** The system shall display contextual feedback for user actions: success confirmations within 1s of action completion, progress indicators for operations longer than 200 ms, and clear error messages with recovery steps when errors occur.
- **NFR-US-3:** New users shall be able to perform core tasks (search, add an item, create a list) without external documentation; this shall be measured by usability testing where  $\geq 80\%$  of participants complete each core task in  $\leq 10$  minutes with no assistance.

### 3.3.2 Reliability

- **NFR-REL-1 (Availability):** The system shall achieve an availability of  $\geq 99.5\%$  measured on a monthly basis (monthly uptime  $\geq 99.5\%$ ).
- **NFR-REL-2 (Graceful Degradation):** If an external metadata or AI API is unavailable, the system shall degrade gracefully: core CRUD and list operations remain fully operational; failed external calls shall be retried according to backoff policy (see NFR-IMPL-3) and failures surfaced to users as non-blocking warnings.

### 3.3.3 Performance

- **NFR-PERF-1 (Typical Library Response):** For libraries of  $\leq 2,000$  items, the system shall return results for common actions (search, view item details, list sort) with median response time  $\leq 500$  ms under normal network conditions.
- **NFR-PERF-2 (Concurrency):** The system shall sustain **200 concurrent active authenticated users** performing mixed read/write operations while maintaining the median response time specified in NFR-PERF-1.
- **NFR-PERF-3 (Tail Latency):** For libraries of  $\leq 2,000$  items, the 95th percentile response time for the same actions shall be  $\leq 1500$  ms.
- **NFR-PERF-4 (Scaling Behavior):** System performance shall degrade in a near-linear manner with library size; doubling the library size shall not exceed double average query latency under equivalent load.

### 3.3.4 Supportability & Maintainability

	<p style="text-align: center;">REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</p>	<p style="text-align: center;">Issue No: 1.0 Issue Date: Nov 2025</p>
---	--	---

- **NFR-SUP-1 (Modularity):** The system shall be architected in modular components (frontend, API layer, metadata connectors, AI connector, persistence) with clearly defined interfaces to allow adding new media types without major redesign.
- **NFR-SUP-2 (Test Coverage):** The codebase shall maintain automated unit and integration test coverage of  $\geq 70\%$  for backend and  $\geq 60\%$  for frontend components.
- **NFR-SUP-3 (CI/CD):** All commits to main branches shall pass automated pipelines (lint, unit tests, integration tests). Deployments to production shall be automated and require successful CI before release.
- **NFR-SUP-4 (Documentation):** Public API endpoints and internal integration points (external metadata, AI API) shall have machine-readable OpenAPI/Swagger specifications and developer README docs updated with each interface change.

### 3.3.5 Implementation

- **NFR-IMPL-1 (Transport Security):** All client-server and server-server communications shall use **TLS 1.2 or later (HTTPS)**.
- **NFR-IMPL-2 (Encryption at Rest):** Sensitive user data and personal identifiers stored in persistent storage shall be encrypted at rest using industry standard algorithms (e.g., **AES-256**).
- **NFR-IMPL-3 (External Call Policies):** Calls to external services (metadata APIs, AI API) shall implement exponential backoff with jitter, maximum 3 retries, and circuit breaker behavior to prevent cascading failures.
- **NFR-IMPL-4 (API Design):** Internal and public APIs shall use JSON over HTTPS, follow RESTful principles, and be versioned using URI versioning (`/api/v1/`) with backward compatibility guarantees for minor version increments.

### 3.3.6 Interface

- **NFR-IF-1 (External API Compatibility):** Integrations with Google Books, Open Library, and IMDb shall adhere to each provider's current authentication and response formats; integration adapters shall normalize external metadata into the internal item model.
- **NFR-IF-2 (Frontend-Backend Contract):** The frontend shall rely on a stable contract (OpenAPI) and detect contract violations at CI time; any breaking changes

	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

require a version bump and migration plan.

### 3.3.7 Packaging & Deployment

- **NFR-PKG-1 (Containerization):** The system shall be packaged as Docker containers per component and deployable to standard Linux container platforms.
- **NFR-PKG-2 (Resource Footprint):** Typical production container image sizes shall not exceed 1GB for backend components; memory and CPU requests/limits must be documented for each service.

### 3.3.8 Security

- **NFR-SEC-1 (Authentication):** The system shall use **OAuth 2.0** for authentication. Access tokens shall expire after **1 hour**, and refresh tokens shall be rotated and stored securely HTTP-only cookies.
- **NFR-SEC-2 (Secrets Management):** API keys, database credentials, and other secrets shall be stored in a secrets manager; secrets must not be present in code or public logs.
- **NFR-SEC-3 (Least Privilege):** Components shall use least-privilege credentials for external services and databases.
- **NFR-SEC-4 (Audit & Logging):** Security-relevant events (auth failures, token refresh, permission changes, external API failures) shall be logged with timestamps and user identifiers and retained. Logging must not include plaintext sensitive data (passwords, full tokens).

### 3.3.9 Data & Privacy

- **NFR-DATA-1 (User Consent):** The system shall obtain explicit user consent before sending identifiable user data to any external AI service. Consent records shall be auditable.

### 3.3.10 Legal & Compliance

	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

- **NFR-LEG-1 (Third-party Terms):** The system shall comply with the license, attribution, and usage terms of all third-party APIs and datasets. Required attributions must be displayed as specified by the provider.

## 3.4 System Models

### 3.4.1 Scenarios

#### **Scenario 1: Adding a New Book (Barcode Scan & API Integration)**

- **Actor:** User
- **Goal:** To quickly add a newly purchased book to the library by scanning its barcode.
- **Related Requirements:** FR-1.2 (Add by ISBN/Barcode) , FR-2.1 (Categorize) , FR-7.1 (Fetch Book Metadata) , FR-6.2 (Login)

#### **Steps:**

1. The User opens the PLMS application on their mobile device and logs in using their Google account (OAuth 2.0).
2. From the home screen, the User taps the "Add New Item" button.
3. The User selects the "Scan ISBN/Barcode" option.
4. The application activates the camera, and the User scans the barcode on the back of the book ("Dune").
5. The system sends a request to the Google Books API using the captured ISBN.
6. Data from the API (Title: "Dune", Author: "Frank Herbert", Summary, Page Count: 412) is automatically populated into the form fields.
7. The system automatically assigns the item to the "Book" category.
8. The User taps the "Save" button.
9. The book is added to the library, and the system displays an "Item added successfully" message.

#### **Scenario 2: Tracking Progress and Creating a Custom List**

- **Actor:** User
- **Goal:** To log reading progress on a book and add that book to a "Favorites" list.
- **Related Requirements:** FR-5.1 (Track Book Progress) , FR-5.4 (Visual Indicators) , FR-5.1 (Create List) , FR-5.2 (Add Item to List)

#### **Steps:**

1. The User logs into their library via a web browser.

	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

2. In the "My Library" view, the User finds the book "Dune" and clicks on it.
3. On the item detail page, the User sees the current progress bar (0% complete).
4. The User clicks the "Update Progress" option.
5. In the pop-up window, the User enters "120" (the number of pages read) and saves.
6. The system calculates the progress (e.g., 29% complete), displays the new percentage, and the remaining page count. This update is also saved to the progress history log.
7. The User then navigates to the "My Lists" tab.
8. The User presses the "Create New List" button and names the list "Favorites".
9. The User returns to the library, selects the "Dune" book, and uses the "Add to List" > "Favorites" option to include the book in this list.

### Scenario 3: API Failure (Graceful Degradation)

- **Actor:** User, System
- **Goal:** To successfully add a DVD manually even if the external metadata API (IMDb) is unresponsive.
- **Related Requirements:** FR-1.1 (Manual Add) , FR-7.2 (Fetch DVD Metadata) , FR-7.3 (Handle API Errors) , NFR-REL-2 (Graceful Degradation)

#### Steps:

1. The User opens the "Add Item Manually" form.
2. The User selects "DVD" as the category and types "The Matrix" as the title.
3. The User presses the "Fetch Info" button.
4. The System saves the item to the database with the basic information (Title, Category).
5. The System attempts to fetch metadata for "The Matrix" from the IMDb API in the background.
6. The IMDb API returns a server error (503) or times out.
7. The System logs the error and initiates a retry policy as per NFR-IMPL-3.
8. "The Matrix" appears in the User's library, but details like summary and director are empty.
9. The System displays a non-blocking warning (e.g., a small icon) to the User: "Could not retrieve metadata.".
10. Core CRUD operations (adding to a list, deleting, etc.) remain fully functional for this DVD item.

### Scenario 4: Filtering the Library and Assigning Tags

- **Actor:** User

	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

- **Goal:** To find all unread Science Fiction books in the library and assign a new priority tag ("Must Read") to one of them.
- **Related Requirements:** FR-4.2 (Filter Items in Library) , FR-3.1 (Create Tag) , FR-3.4 (Assign Tag to Item)

**Steps:**

1. The User opens the main library view, which contains over 150 items (books and DVDs).
2. The User opens the filtering options.
3. The User selects "Book" for the "Category" filter. The list narrows to show only books.
4. The User sets the "Genre" filter to "Science Fiction".
5. The User selects the existing "Unread" tag from the "Tag" filter.
6. The system updates the list to display only the items that match all three criteria (Book, Science Fiction, Unread).
7. The User selects the "Dune" book from the filtered list and navigates to its detail page.
8. The User clicks on the "Manage Tags" section within the item details.
9. The User types "Must Read" into the text box, noticing this tag does not exist yet.
10. The User clicks the "Create New Tag: Must Read" option.
11. The system creates the new tag and automatically assigns it to the "Dune" book. The book's tags now show "Unread" and "Must Read".

**Scenario 5: Opting-In and Viewing AI Recommendations**

- **Actor:** User
- **Goal:** To grant consent for AI-powered recommendations and view the new book suggestions.
- **Related Requirements:** FR-8.1 (Recommendation Generation) , FR-9.2 (Recommendation Display) , FR-8.3 (User Consent) , NFR-DATA-1 (User Consent)

**Steps:**

1. The User has added "Dune," "1984," and "Brave New World" to their library.
2. The User clicks on the "Recommendations" tab from the main menu for the first time.
3. As required by FR-8.3, the system displays a consent pop-up: "To provide you with personalized recommendations, your library content (e.g., genres, authors) must be sent to our external AI API for analysis. Do you approve?"
4. The User clicks the "I Approve" button. The system records this consent.
5. The system sends the User's library data (Sci-Fi, Dystopian) to the AI API as per FR-8.1.
6. The AI API returns a response.

	<b>REQUIREMENTS ANALYSIS DOCUMENT</b> for <b>PERSONAL LIBRARY MANAGEMENT</b> <b>SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	---	---------------------------------------

7. In accordance with FR-8.2, the system displays "Foundation" and "Fahrenheit 451" under the "Book Recommendations" heading.

#### **Scenario 6: Deleting an Item from the Main Library (Impact on Lists)**

- **Actor:** User
- **Goal:** To delete a book they no longer own from the main library and have it automatically removed from all associated lists.
- **Related Requirements:** FR-1.5 (Delete Media Item) , FR-5.3 / FR-5.4 (Remove Item from List - Automatic)

#### **Steps:**

1. The User has "The Hitchhiker's Guide to the Galaxy" in their library.
2. This book is also included in two of the User's custom lists: "My Favorites" and "To Re-read."
3. The User, having given the book to a friend permanently, finds the item in the main library view.
4. The User clicks the "Delete" button for that item.
5. The system asks for confirmation: "Are you sure you want to permanently delete this item? It will also be removed from all lists it belongs to."
6. The User confirms the deletion.
7. The system deletes the book from the main database (**FR-1.5**).
8. As per the rules in **FR-1.5** and **FR-5.3**, the system also automatically removes the item from the "My Favorites" and "To Re-read" lists.
9. When the User navigates to the "My Favorites" list, they observe the book is no longer there.

#### 3.4.2 Use case model

##### **1. UC-1: Add a New Item**

Use case name	Add a New Item
Participating actors	<b>User, System</b>
Brief description	Allows the user to add a new physical <b>Book</b> or <b>DVD</b> to their personal library collection either by <b>manual data entry</b> or by

	REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

	<b>scanning a barcode/ISBN.</b>
Flow of events	<ol style="list-style-type: none"> <li>1. The User initiates the "Add New Item" function.</li> <li>2. The System presents a choice: Manual Entry or Scan Barcode/ISBN.</li> <li>3. The User selects an option:           <ul style="list-style-type: none"> <li><b>A) Manual Entry:</b> The User enters details (title, category, condition, etc.) and submits.</li> <li><b>B) Scan Barcode/ISBN (UC-18):</b> The User uses the device camera to scan the ISBN/barcode.</li> </ul> </li> <li>4. <b>If Scan:</b> The System automatically attempts to retrieve metadata (UC-26). If successful, the data is pre-filled.</li> <li>5. The User reviews/completes the item details and saves.</li> <li>6. The System saves the new item record to the database and displays a success confirmation.</li> </ol>
Entry condition	* The User is authenticated (logged in).
Exit condition	<ul style="list-style-type: none"> <li>* The new item record (Book or DVD) has been successfully saved to the database.</li> <li>* The System has displayed a successful confirmation message.</li> </ul>
Quality requirements	* If using barcode scan, metadata should be fetched and pre-filled with a median response time $\leq$ 500 ms (NFR-PERF-1).

## 2. UC-2: Edit Existing Item

Use case name	Edit Existing Item
Participating actors	<b>User, System</b>
Brief description	Allows the user to modify or correct the information (e.g., title, author, condition, location) of an item already saved in their library to ensure data accuracy.
Flow of events	<ol style="list-style-type: none"> <li>1. The User views their library and selects an existing item.</li> <li>2. The User initiates the "Edit Item" function.</li> <li>3. The System displays the editable item details form.</li> <li>4. The User makes the necessary corrections or updates (e.g., updates loan status, condition).</li> </ol>

	<p align="center"><b>REQUIREMENTS ANALYSIS DOCUMENT</b> for <b>PERSONAL LIBRARY MANAGEMENT</b> <b>SYSTEM</b></p>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

	5. The User submits the changes. 6. The System validates the input and updates the corresponding record in the database. 7. The System displays a success confirmation.
Entry condition	* The User is authenticated (logged in). * The selected media item already exists in the user's library.
Exit condition	* The selected item's information has been successfully updated in the database. * The System has displayed a successful confirmation message.
Quality requirements	* The item update operation shall complete with a median response time $\leq$ 500 ms (NFR-PERF-1).

### 3. UC-3: Delete Item

Use case name	Delete Item
Participating actors	<b>User, System</b>
Brief description	Allows the user to permanently remove an item they no longer own from their library. The system must also automatically remove the item from any associated custom lists (FR-1.5, FR-5.3).
Flow of events	1. The User selects an item from the library view or item details page. 2. The User initiates the "Delete Item" function. 3. The System displays a confirmation dialog warning that the item will be permanently removed from the library <b>and all associated lists</b> . 4. The User confirms the deletion. 5. The System removes the item record from the main database. 6. The System automatically removes all links between the deleted item and any custom lists (FR-1.5, FR-5.3). 7. The System displays a success confirmation.
Entry condition	* The User is authenticated (logged in). * The selected media item exists in the user's library.
Exit condition	* The item record has been successfully deleted from the main database. * All links between the deleted item and custom lists have been severed (FR-1.5, FR-5.3). * The System has displayed a successful confirmation

	REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

	message.
Quality requirements	* The entire deletion process, including list cleanup, shall be completed with a median response time $\leq$ 500 ms (NFR-PERF-1).

#### 4. UC-4: View Item Details

Use case name	View Item Details
Participating actors	<b>User, System</b>
Brief description	Allows the user to view all stored, corrected, and enriched detailed information (title, author/director, summary, condition, progress, etc.) about a selected media item.
Flow of events	<ol style="list-style-type: none"> <li>1. The User browses or searches the library and selects a media item.</li> <li>2. The System retrieves all associated data for the item (including locally saved metadata, user-defined tags, progress logs, and ratings).</li> <li>3. The System displays the comprehensive item details page to the User.</li> <li>4. The User reviews the information.</li> </ol>
Entry condition	<ul style="list-style-type: none"> <li>* The User is authenticated (logged in).</li> <li>* The selected media item exists in the user's library.</li> </ul>
Exit condition	<ul style="list-style-type: none"> <li>* The System has successfully displayed all details of the selected item to the User</li> </ul>
Quality requirements	* Item details shall be displayed with a median response time $\leq$ 500 ms for libraries up to 2,000 items (NFR-PERF-1).

#### 5. UC-5: Create & Edit Tags

Use case name	Create & Edit Tags
Participating actors	<b>User, System</b>

	<p align="center"><b>REQUIREMENTS ANALYSIS DOCUMENT</b> for <b>PERSONAL LIBRARY MANAGEMENT</b> <b>SYSTEM</b></p>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

Brief description	Allows the user to manage user-defined tags by creating new tags or modifying the name of existing ones for flexible categorization.
Flow of events	<ol style="list-style-type: none"> <li>1. The User navigates to the Tag Management section.</li> <li>2. <b>To Create Tag (FR-3.1):</b> The User specifies a unique tag name and saves it.</li> <li>3. The System validates the name for uniqueness and saves the new tag record.</li> <li>4. <b>To Edit Tag (FR-3.2):</b> The User selects an existing tag and modifies its name.</li> <li>5. The User saves the modification.</li> <li>6. The System updates the tag record and ensures the new tag name is reflected across all associated media items.</li> <li>7. The System displays a success confirmation.</li> </ol>
Entry condition	* The User is authenticated (logged in).
Exit condition	<ul style="list-style-type: none"> <li>* A new tag has been successfully created and saved (FR-3.1) , or an existing tag name has been successfully updated (FR-3.2).</li> <li>* The System has displayed a success confirmation message.</li> </ul>
Quality requirements	* Tag creation and update operations shall complete within 1 second.

## 6. UC-6: Assign / Remove Tags on Item

Use case name	Assign / Remove Tags on Item
Participating actors	<b>User, System</b>
Brief description	Allows the user to associate an item with one or more existing tags, or to remove an existing association.
Flow of events	<ol style="list-style-type: none"> <li>1. The User views the details of a media item.</li> <li>2. The User initiates the "Manage Tags" function.</li> <li>3. The User selects one or more existing tags to <b>assign</b> to the item (FR-3.4).</li> <li>4. The User selects one or more currently assigned tags to <b>remove</b> from the item (FR-3.5).</li> <li>5. The User saves the changes.</li> </ol>

	<b>REQUIREMENTS ANALYSIS DOCUMENT</b> for <b>PERSONAL LIBRARY MANAGEMENT</b> <b>SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	---	---------------------------------------

	6. The System updates the Item-Tag relationship records (junction table) in the database. 7. The System displays a success confirmation.
Entry condition	<ul style="list-style-type: none"> <li>* The User is authenticated (logged in).</li> <li>* The selected media item exists in the user's library.</li> <li>* At least one tag exists in the system (FR-3.1).</li> </ul>
Exit condition	<ul style="list-style-type: none"> <li>* The Item-Tag relationship has been successfully created (assigned) or destroyed (removed).</li> <li>* The item's detail view reflects the new list of tags.</li> </ul>
Quality requirements	<ul style="list-style-type: none"> <li>* Tag assignment and removal should be completed within 1 second (NFR-US-2).</li> </ul>

## 7. UC-7: Filter by Tag(s)

Use case name	Filter by Tag(s)
Participating actors	<b>User, System</b>
Brief description	Allows the user to narrow down the view of the main library by selecting one or more user-defined tags.
Flow of events	<ol style="list-style-type: none"> <li>1. The User is viewing the main library collection.</li> <li>2. The User initiates the filtering options.</li> <li>3. The User selects one or more tags (e.g., "Must Read," "Sci-Fi") from the list of available tags.</li> <li>4. The User applies the filter.</li> <li>5. The System executes a query to retrieve only the media items associated with <b>all</b> selected tags.</li> <li>6. The System displays the filtered results to the User.</li> </ol>
Entry condition	<ul style="list-style-type: none"> <li>* The User is authenticated (logged in).</li> <li>* The user's library contains items.</li> </ul>
Exit condition	<ul style="list-style-type: none"> <li>* The library view is updated to display only the items matching the selected tags.</li> </ul>
Quality requirements	<ul style="list-style-type: none"> <li>* Filtered results shall be returned with a median response time <math>\leq</math> 500 ms for libraries up to 2,000 items (NFR-PERF-1).</li> </ul>

	<b>REQUIREMENTS ANALYSIS DOCUMENT</b> for <b>PERSONAL LIBRARY MANAGEMENT</b> <b>SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	---	---------------------------------------

## 8. UC-8: Create Custom List

Use case name	Create Custom List
Participating actors	<b>User, System</b>
Brief description	Allows the user to create a new customized list (e.g., "Watch Later," "Favorites") separate from the main library to organize selected items.
Flow of events	<ol style="list-style-type: none"> <li>1. The User navigates to the Lists management section.</li> <li>2. The User initiates the "Create New List" function (FR-5.1).</li> <li>3. The User enters a name for the new list (e.g., "Top 10 Dystopian Books").</li> <li>4. The User submits the list name.</li> <li>5. The System validates the input and saves the new custom list record.</li> <li>6. The System displays a success confirmation and shows the new, empty list in the Lists view.</li> </ol>
Entry condition	* The User is authenticated (logged in).
Exit condition	<ul style="list-style-type: none"> <li>* A new custom list record has been successfully created and saved (FR-5.1).</li> <li>* The System has displayed a successful confirmation message.</li> </ul>
Quality requirements	* List creation shall be completed within 1 second (NFR-US-2).

## 9. UC-9: View All Lists

Use case name	View All Lists
Participating actors	<b>User, System</b>
Brief description	Allows the user to view all their existing custom lists and see the count of items contained within each list.
Flow of events	<ol style="list-style-type: none"> <li>1. The User navigates to the Lists management section.</li> <li>2. The System queries the database to retrieve all custom list records associated with the User.</li> <li>3. For each list, the System calculates the total number of associated media items.</li> <li>4. The System displays the list of custom lists, including their names and item counts, to the User.</li> </ol>
Entry condition	<ul style="list-style-type: none"> <li>* The User is authenticated (logged in).</li> <li>* The selected media item exists in the user's library.</li> </ul>

	REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

	* At least one tag exists in the system (FR-3.1).
Exit condition	* All existing custom lists have been successfully displayed to the User.
Quality requirements	* The list view shall load with a median response time ≤ 500 ms (NFR-PERF-1).

#### 10. UC-10: Rename List

Use case name	Rename List
Participating actors	<b>User, System</b>
Brief description	Allows the user to modify the name of an existing custom list.
Flow of events	<ol style="list-style-type: none"> <li>1. The User navigates to the Lists management section.</li> <li>2. The User selects an existing custom list.</li> <li>3. The User initiates the "Rename List" function.</li> <li>4. The System displays a text field containing the current list name.</li> <li>5. The User enters the new list name and confirms.</li> <li>6. The System validates the new name and updates the list record in the database.</li> <li>7. The System displays a success confirmation.</li> </ol>
Entry condition	<ul style="list-style-type: none"> <li>* The User is authenticated (logged in).</li> <li>* At least one custom list exists.</li> </ul>
Exit condition	<ul style="list-style-type: none"> <li>* The selected list's name has been successfully updated in the database.</li> <li>* The System has displayed a successful confirmation message.</li> </ul>
Quality requirements	* List rename operation shall complete within 1 second (NFR-US-2).

#### 11. UC-11: Delete List

Use case name	Delete List
Participating actors	<b>User, System</b>
Brief description	Allows the user to delete a custom list that is no longer needed.

	<p align="center"><b>REQUIREMENTS ANALYSIS DOCUMENT</b> for <b>PERSONAL LIBRARY MANAGEMENT</b> <b>SYSTEM</b></p>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

Flow of events	<ol style="list-style-type: none"> <li>1. The User navigates to the Lists management section.</li> <li>2. The User selects an existing custom list.</li> <li>3. The User initiates the "Delete List" function (FR-5.5).</li> <li>4. The System displays a confirmation dialog.</li> <li>5. The User confirms the deletion.</li> <li>6. The System deletes the custom list record and severs all item-to-list relationships for that list.</li> </ol> <p><b>Note:</b> The media items themselves remain in the main library.</p> <ol style="list-style-type: none"> <li>7. The System displays a success confirmation.</li> </ol>
Entry condition	<ul style="list-style-type: none"> <li>* The User is authenticated (logged in).</li> <li>* At least one custom list exists.</li> </ul>
Exit condition	<ul style="list-style-type: none"> <li>* The selected custom list record has been successfully deleted (FR-5.5).</li> <li>* The System has displayed a successful confirmation message.</li> </ul>
Quality requirements	<ul style="list-style-type: none"> <li>* List deletion operation shall complete within 1 second (NFR-US-2).</li> </ul>

## 12. UC-12: Add / Remove Item from List

Use case name	Add / Remove Item from List
Participating actors	<b>User, System</b>
Brief description	Allows the user to add an item to an existing custom list or remove an item from a list.
Flow of events	<ol style="list-style-type: none"> <li>1. The User selects a media item (from the library or item details).</li> <li>2. The User initiates the "Manage Lists" function.</li> <li>3. The User selects an existing list to <b>add</b> the item to (FR-5.2). OR</li> <li>4. The User selects an existing list to <b>remove</b> the item from (FR-5.3).</li> <li>5. The User confirms the action.</li> <li>6. The System creates or deletes the Item-List relationship (junction table record) in the database.</li> <li>7. The System displays a success confirmation.</li> </ol>
Entry condition	<ul style="list-style-type: none"> <li>* The User is authenticated (logged in).</li> <li>* At least one media item exists in the library.</li> <li>* At least one custom list exists.</li> </ul>
Exit condition	<ul style="list-style-type: none"> <li>* The item has been successfully added to or removed from the selected list.</li> </ul>

	REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

	* The System has displayed a successful confirmation message.
Quality requirements	* Item manipulation within a list shall be completed within 1 second (NFR-US-2).

### 13. UC-13: Reorder Items in a List

Use case name	Reorder Items in a List
Participating actors	<b>User, System</b>
Brief description	Allows the user to manually adjust the order of items within a custom list to prioritize or arrange them according to preference (FR-5.7).
Flow of events	<ol style="list-style-type: none"> <li>1. The User selects an existing custom list.</li> <li>2. The System displays the items within the list.</li> <li>3. The User initiates the reordering function (e.g., drag-and-drop or manual priority setting) (FR-5.7).</li> <li>4. The User moves an item to a new position in the list.</li> <li>5. The User saves the new order.</li> <li>6. The System updates the internal priority/order metadata for the items within that specific list.</li> <li>7. The System displays the list in the new, customized order.</li> </ol>
Entry condition	<ul style="list-style-type: none"> <li>* The User is authenticated (logged in).</li> <li>* The selected custom list contains at least two items.</li> </ul>
Exit condition	* The order of items in the selected list has been successfully updated and saved (FR-5.7).
Quality requirements	* The list view shall refresh to reflect the new order within 500 ms (NFR-PERF-1).

### 14. UC-14: Search and Filter Within a List

Use case name	Search and Filter Within a List
Participating actors	<b>User, System</b>
Brief description	Allows the user to quickly locate items within a specific custom list using search (title, author) or filtering (tag, category) capabilities.
Flow of events	<ol style="list-style-type: none"> <li>1. The User selects an existing custom list.</li> </ol>

	<p align="center"><b>REQUIREMENTS ANALYSIS DOCUMENT</b> for <b>PERSONAL LIBRARY MANAGEMENT</b> <b>SYSTEM</b></p>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

	<p>2. The User initiates the search/filter function within that list (FR-5.8, FR-5.9).</p> <p>3. <b>Search:</b> The User enters text (e.g., part of a title or author name).  <b>Filter:</b> The User selects properties (e.g., Category: Book).</p> <p>4. The System executes the search/filter logic constrained only to the items in the selected list.</p> <p>5. The System displays the results that match the criteria.</p>
Entry condition	<ul style="list-style-type: none"> <li>* The User is authenticated (logged in).</li> <li>* The selected custom list contains items.</li> </ul>
Exit condition	<ul style="list-style-type: none"> <li>* The list view is updated to display only the items matching the search/filter criteria (FR-5.8, FR-5.9).</li> </ul>
Quality requirements	<ul style="list-style-type: none"> <li>* Search and filter results shall be returned with a median response time ≤ 500 ms (NFR-PERF-1).</li> </ul>

### 15. UC-15: Track Reading/Watching Time

Use case name	Track Reading/Watching Time
Participating actors	<b>User, System</b>
Brief description	Allows the user to log the specific time or duration they spent reading a book or watching a DVD for historical analysis (FR-6.3).
Flow of events	<p>1. The User selects an item currently "In Progress."</p> <p>2. The User initiates the "Log Session" function.</p> <p>3. The User enters the duration of the reading/viewing session (e.g., "45 minutes").</p> <p>4. The System saves a historical log entry (FR-6.3), including the date, the duration, and the item identifier.</p> <p>5. The System displays a success confirmation.</p>
Entry condition	<ul style="list-style-type: none"> <li>* The User is authenticated (logged in).</li> <li>* The selected item is marked as "In Progress" or "Not Started" (UC-16).</li> </ul>
Exit condition	<ul style="list-style-type: none"> <li>* A new log entry containing the date and duration has been successfully saved to the progress history (FR-6.3).</li> </ul>
Quality requirements	<ul style="list-style-type: none"> <li>* Progress update logging shall be completed within 1 second (NFR-US-2).</li> </ul>

	REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

### 16. UC-16: Filter by Completion Status

Use case name	Filter by Completion Status
Participating actors	<b>User, System</b>
Brief description	Allows the user to filter the main library view based on the consumption status of the media item: "Completed," "In Progress," or "Not Started".
Flow of events	<ol style="list-style-type: none"> <li>1. The User is viewing the main library collection.</li> <li>2. The User initiates the filtering options.</li> <li>3. The User selects a completion status filter (e.g., "In Progress").</li> <li>4. The System executes a query to retrieve items matching the selected status (based on pages read/minutes watched vs. total count).</li> <li>5. The System displays the filtered results to the User.</li> </ol>
Entry condition	<ul style="list-style-type: none"> <li>* The User is authenticated (logged in).</li> <li>* The user's library contains items.</li> </ul>
Exit condition	* The library view is updated to display only the items matching the selected completion status.
Quality requirements	* Filtered results shall be returned with a median response time $\leq$ 500 ms (NFR-PERF-1).

### 17. UC-17: Advanced Filtering

Use case name	Advanced Filtering
Participating actors	<b>User, System</b>
Brief description	Allows the user to apply multiple, complex filters simultaneously, such as combining genre, tag, author/director, release year, and format (Category) to refine search results.
Flow of events	<ol style="list-style-type: none"> <li>1. The User is viewing the main library collection.</li> <li>2. The User accesses the advanced filtering options.</li> <li>3. The User selects multiple criteria (e.g., Category: "Book," Genre: "Sci-Fi," Tag: "Unread").</li> <li>4. The User applies the filters.</li> <li>5. The System constructs and executes a complex query that satisfies <b>all</b> selected criteria (FR-4.2).</li> <li>6. The System displays the refined list of matching items.</li> </ol>
Entry condition	* The User is authenticated (logged in).

	REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

	* The user's library contains items.
Exit condition	* The library view is updated to display only the items matching the combination of advanced filter criteria (FR-4.2).
Quality requirements	* The 95th percentile response time for advanced filtering shall be ≤ 1500 ms (NFR-PERF-3).

#### 18. UC-18: Scan ISBN number of a Book

Use case name	Scan ISBN number of a Book
Participating actors	<b>User, System</b>
Brief description	Allows the user to use their device's camera to capture a book's ISBN/barcode, which the system uses to automatically fetch item details.
Flow of events	<ol style="list-style-type: none"> <li>1. The User initiates the "Scan ISBN/Barcode" function (FR-1.2).</li> <li>2. The System activates the device camera.</li> <li>3. The User scans the ISBN/barcode on the book.</li> <li>4. The System captures the code and deactivates the camera.</li> <li>5. The System uses the captured code to initiate the <b>Fetch Online Details (UC-26)</b> process.</li> <li>6. The System presents the pre-filled item details form to the User (Step 5 of UC-1).</li> </ol>
Entry condition	<ul style="list-style-type: none"> <li>* The User is authenticated (logged in).</li> <li>* The User's device has a functional camera.</li> </ul>
Exit condition	<ul style="list-style-type: none"> <li>* The System has successfully captured a valid ISBN/barcode.</li> <li>* The process to add the item continues with the automatic population of details (FR-1.2).</li> </ul>
Quality requirements	* The scanning and capture process should be highly usable (NFR-US-1).

#### 19. UC-19: Track Progress

Use case name	Track Progress
Participating actors	<b>User, System</b>

	<b>REQUIREMENTS ANALYSIS DOCUMENT</b> for <b>PERSONAL LIBRARY MANAGEMENT</b> <b>SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	---	---------------------------------------

Brief description	Allows the user to update their reading (pages/chapters) or viewing (minutes) progress for an item, and for the system to calculate the percentage completed.
Flow of events	<ol style="list-style-type: none"> <li>1. The User selects an item from their library.</li> <li>2. The User initiates the "Update Progress" function.</li> <li>3. The User enters the new total pages read (for Books) or minutes watched (for DVDs).</li> <li>4. The System calculates the new completion percentage and remaining amount (FR-6.1, FR-6.2).</li> <li>5. The System updates the item's progress status and saves a historical log entry (FR-6.3).</li> <li>6. The System updates the item's detail view to show the new visual progress indicator (FR-6.4).</li> </ol>
Entry condition	<ul style="list-style-type: none"> <li>* The User is authenticated (logged in).</li> <li>* The selected item exists and has total page count or duration metadata.</li> </ul>
Exit condition	<ul style="list-style-type: none"> <li>* The item's progress status has been updated in the database (FR-6.1, FR-6.2).</li> <li>* A historical progress log has been created (FR-6.3).</li> </ul>
Quality requirements	<ul style="list-style-type: none"> <li>* Progress calculation and display must complete within 1 second (NFR-US-2).</li> </ul>

## 20. UC-20: View Analytics

Use case name	View Analytics
Participating actors	<b>User, System</b>
Brief description	Allows the user to view summary statistics about their collection (e.g., number of books read, total pages, number of DVDs watched).
Flow of events	<ol style="list-style-type: none"> <li>1. The User navigates to the Analytics/Dashboard section.</li> <li>2. The System queries the database and log repository (LogRepository) to calculate summary statistics (e.g., total completed items, total time logged).</li> <li>3. The System compiles the statistics into easily digestible formats (tables, numbers).</li> <li>4. The System displays the calculated statistics to the User.</li> </ol>
Entry condition	<ul style="list-style-type: none"> <li>* The User is authenticated (logged in).</li> <li>* The user's library contains items and progress logs.</li> </ul>
Exit condition	<ul style="list-style-type: none"> <li>* The calculated summary statistics have been displayed to the User.</li> </ul>

	REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

	the User.
Quality requirements	* The dashboard view (UC-22) must load within 1500 ms (NFR-PERF-3).

## 21. UC-21: Manage User Account

Use case name	Manage User Account
Participating actors	<b>User, System, Google</b> (or other OAuth Provider)
Brief description	Allows the user to register, log in, and manage their system profile securely via an OAuth 2.0 identity provider.
Flow of events	<ol style="list-style-type: none"> <li>1. The User initiates the registration or login process.</li> <li>2. The System delegates authentication to the OAuth 2.0 identity provider (e.g., Google) (FR-7.1, FR-7.2).</li> <li>3. The User authenticates with the external provider.</li> <li>4. The OAuth provider returns an authorization/access token to the System.</li> <li>5. The System verifies the token and either creates a new user profile (Registration) or logs the existing user in (Login).</li> <li>6. The System displays the main library view.</li> </ol>
Entry condition	* The User is not currently logged in, or the User explicitly accesses profile management settings.
Exit condition	<ul style="list-style-type: none"> <li>* The User is successfully logged in and access to the system is secured (FR-7.2).</li> <li>* User authentication is handled exclusively by the OAuth provider (NFR-SEC-1).</li> </ul>
Quality requirements	<ul style="list-style-type: none"> <li>* Authentication shall be handled exclusively via OAuth 2.0 (NFR-SEC-1).</li> <li>* Access tokens shall expire after 1 hour (NFR-SEC-1).</li> </ul>

## 22. UC-22: Data Visualization Dashboard

Use case name	Data Visualization Dashboard
Participating actors	<b>User, System</b>
Brief description	Allows the user to view their reading/viewing patterns and habits over time using graphical representations (e.g., graphs by month, genre, or duration).

	<b>REQUIREMENTS ANALYSIS DOCUMENT</b> for <b>PERSONAL LIBRARY MANAGEMENT</b> <b>SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	---	---------------------------------------

Flow of events	1. The User navigates to the Dashboard/Analytics section. 2. The System collects and processes data from multiple sources (ItemRepository, LogRepository). 3. The System groups and aggregates the data (e.g., "pages read per month," "items per genre"). 4. The System renders the aggregated data into visual charts and graphs (e.g., bar charts, pie charts). 5. The System displays the dashboard to the User.
Entry condition	* The User is authenticated (logged in). * The user's library contains items and progress logs.
Exit condition	* The User has been successfully presented with the data visualization dashboard.
Quality requirements	* The dashboard view shall load with a 95th percentile response time ≤ 1500 ms (NFR-PERF-3).

### 23. UC-23: Mark Favorite Items

Use case name	Mark Favorite Items
Participating actors	<b>User, System</b>
Brief description	Allows the user to explicitly mark a media item as a "Favorite" for easier access or filtering.
Flow of events	1. The User selects a media item. 2. The User initiates the "Mark as Favorite" toggle/function. 3. The System updates a specific attribute (e.g., <code>isFavorite</code> boolean or assigns a dedicated tag) on the item's database record. 4. The System displays a visual indicator (e.g., a heart icon) next to the item. 5. The System displays a success confirmation.
Entry condition	* The User is authenticated (logged in). * The selected media item exists in the user's library.
Exit condition	* The selected item's "Favorite" status has been successfully updated in the database.
Quality requirements	* Status update operation shall complete within 1 second (NFR-US-2).

	<b>REQUIREMENTS ANALYSIS DOCUMENT</b> for <b>PERSONAL LIBRARY MANAGEMENT</b> <b>SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	---	---------------------------------------

#### 24. UC-24: Rate or Review an Item

Use case name	Rate or Review an Item
Participating actors	<b>User, System</b>
Brief description	Allows the user to assign a personal rating (e.g., 1-5 stars) and/or write a short review for a book or DVD to record their impressions.
Flow of events	<ol style="list-style-type: none"> <li>1. The User selects a media item.</li> <li>2. The User initiates the "Add Rating/Review" function.</li> <li>3. The User enters a rating (e.g., 4/5 stars) and/or a text review.</li> <li>4. The User submits the input.</li> <li>5. The System saves the personal rating and review content, linking it to the item and the User.</li> <li>6. The System displays the user's rating/review on the item details page.</li> </ol>
Entry condition	<ul style="list-style-type: none"> <li>* The User is authenticated (logged in).</li> <li>* The selected media item exists in the user's library</li> </ul>
Exit condition	* The user's personal rating and/or review has been successfully saved for the item.
Quality requirements	* Rating and review submission shall complete within 1 second (NFR-US-2).

#### 25. UC-25: Dark/Light Mode Toggle

Use case name	Dark/Light Mode Toggle
Participating actors	<b>User, System</b>
Brief description	Allows the user to switch the system's color scheme between light and dark themes for better visibility and user comfort.
Flow of events	<ol style="list-style-type: none"> <li>1. The User navigates to system settings or locates the mode toggle control.</li> <li>2. The User activates the toggle (e.g., clicks the Sun/Moon icon).</li> <li>3. The System detects the change and applies the corresponding CSS/styling rules.</li> <li>4. The System updates the preference setting in the user's profile database record.</li> <li>5. The System displays the application interface with the</li> </ol>

	REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

	newly selected theme.
Entry condition	* The User is authenticated (logged in).
Exit condition	* The system's UI successfully renders in the newly selected theme (Dark or Light).
Quality requirements	* The UI mode switch must be instantaneous, with visual changes completed in ≤ 200 ms (NFR-US-2).

## 26. UC-26: Fetch Online Details

Use case name	Fetch Online Details (Metadata)
Participating actors	<b>System, External API</b> (Google Books, IMDb, Open Library)
Brief description	Allows the System to automatically retrieve rich metadata (summary, release date, author/director) from external APIs to enrich a new or existing item entry.
Flow of events	<ol style="list-style-type: none"> <li>1. The System receives a trigger (e.g., ISBN from UC-18, manual request for an existing item).</li> <li>2. The System determines the media type (Book or DVD).</li> <li>3. <b>If Book:</b> The System calls the Google Books/Open Library API (FR-8.1).  <b>If DVD:</b> The System calls the IMDb API (FR-8.2).</li> <li>4. The External API returns the requested metadata in JSON format (NFR-IMPL-4).</li> <li>5. The System normalizes the external metadata into the internal item model (NFR-IF-1).</li> <li>6. The System saves the enriched data to the item record.</li> </ol>
Entry condition	<ul style="list-style-type: none"> <li>* A valid identifier (ISBN, title) is available for the media item.</li> <li>* The System has network connectivity to the External API.</li> </ul>
Exit condition	<ul style="list-style-type: none"> <li>* The media item record has been successfully enriched with external metadata.</li> <li>* If the API fails, the core item is still saved (NFR-REL-2).</li> </ul>
Quality requirements	<ul style="list-style-type: none"> <li>* Calls to external services shall implement exponential backoff and circuit breaker patterns (NFR-IMPL-3).</li> </ul>

	<b>REQUIREMENTS ANALYSIS DOCUMENT</b> for <b>PERSONAL LIBRARY MANAGEMENT</b> <b>SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	---	---------------------------------------

## 27. UC-27: Automatic Metadata Update

Use case name	Automatic Metadata Update
Participating actors	<b>System, External API</b> (Google Books, IMDb, Open Library)
Brief description	Allows the System to periodically check and update metadata for existing media items from external APIs to keep information current (e.g., cover art, external rating changes).
Flow of events	<ol style="list-style-type: none"> <li>1. The System initiates a scheduled background task (System Actor).</li> <li>2. The System retrieves a batch of media items that are eligible for an update check (FR-8.3).</li> <li>3. For each item, the System calls the relevant External API to fetch the latest metadata (UC-26).</li> <li>4. If new or updated metadata is found (e.g., a new professional rating), the System updates the corresponding item record.</li> <li>5. The System logs the update operation.</li> </ol>
Entry condition	<ul style="list-style-type: none"> <li>* A scheduled time/trigger for the background update has occurred.</li> <li>* The System has network connectivity to the External API.</li> </ul>
Exit condition	* All eligible item records have been checked and updated with the latest external metadata (FR-8.3).
Quality requirements	<ul style="list-style-type: none"> <li>* External calls must implement robust error handling (NFR-IMPL-3) and degrade gracefully upon failure (NFR-REL-2).</li> </ul>

## 28. UC-28: Generate Recommendations

Use case name	Generate Recommendations
Participating actors	<b>System, External AI API</b>
Brief description	Allows the System to generate personalized suggestions for books and DVDs based on the user's existing library, history, and preferences.
Flow of events	<ol style="list-style-type: none"> <li>1. The System is triggered (e.g., by user consent, scheduled task) (System Actor).</li> <li>2. The System verifies the user's explicit consent to use identifiable data (FR-9.3, NFR-DATA-1).</li> </ol>

	<p align="center"><b>REQUIREMENTS ANALYSIS DOCUMENT</b> for <b>PERSONAL LIBRARY MANAGEMENT</b> <b>SYSTEM</b></p>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

	<p>3. <b>If Consent Granted:</b> The System extracts the required user data (e.g., genres, authors).</p> <p>4. The System sends the anonymized/consented data to the External AI API (FR-9.1).</p> <p>5. The AI API processes the request and returns a list of recommended media items.</p> <p>6. The System saves the generated recommendations (e.g., to a RecommendationsTable).</p> <p>7. The System makes the recommendations available for display (UC-29).</p>
Entry condition	* The User has explicitly granted consent for AI recommendation (NFR-DATA-1).
Exit condition	* A set of personalized recommendations has been generated and saved by the System (FR-9.1).
Quality requirements	<ul style="list-style-type: none"> <li>* The System shall obtain explicit, auditable user consent (NFR-DATA-1).</li> <li>* The System shall degrade gracefully if the AI API is unavailable (NFR-REL-2).</li> </ul>

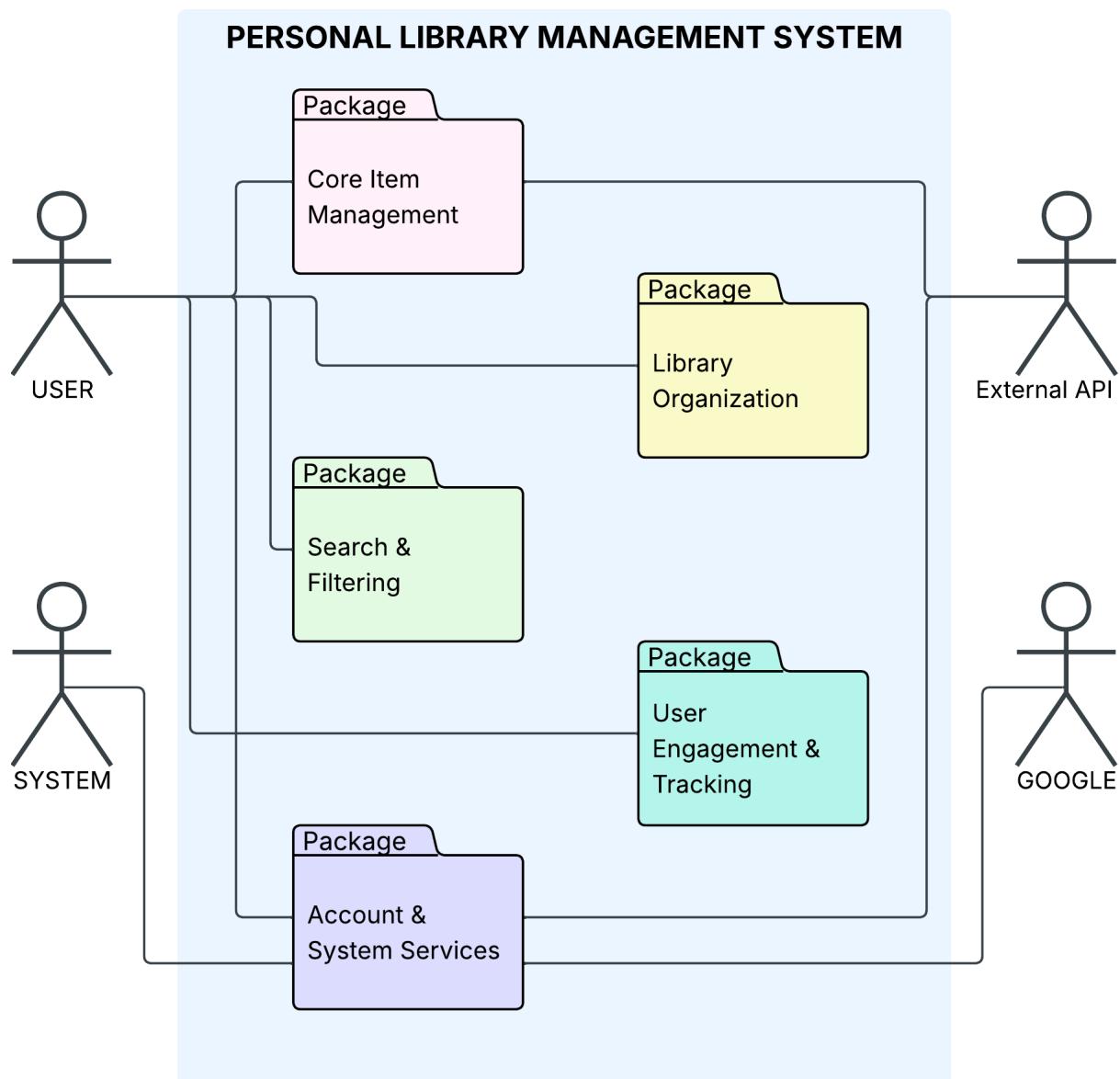
## 29. UC-29: Social Login Integration

Use case name	Social Login Integration
Participating actors	<b>User, System, Google</b> (or other OAuth Provider)
Brief description	Allows the system to integrate with external identity providers (like Google) to simplify the User registration and login process (UC-21).
Flow of events	<ol style="list-style-type: none"> <li>1. The User clicks the "Sign in with Google" button (FR-7.1).</li> <li>2. The System redirects the User to the Google authentication server.</li> <li>3. The User logs in and authorizes the PLMS application on the Google side.</li> <li>4. Google returns an authorization code to the PLMS (System).</li> <li>5. The System exchanges the code for an access token (FR-7.2).</li> <li>6. The System uses the token to retrieve basic user profile information (email, OAuth ID).</li> <li>7. The System creates or verifies the local user account and establishes the session (UC-21).</li> <li>8. The System redirects the User to the main application</li> </ol>

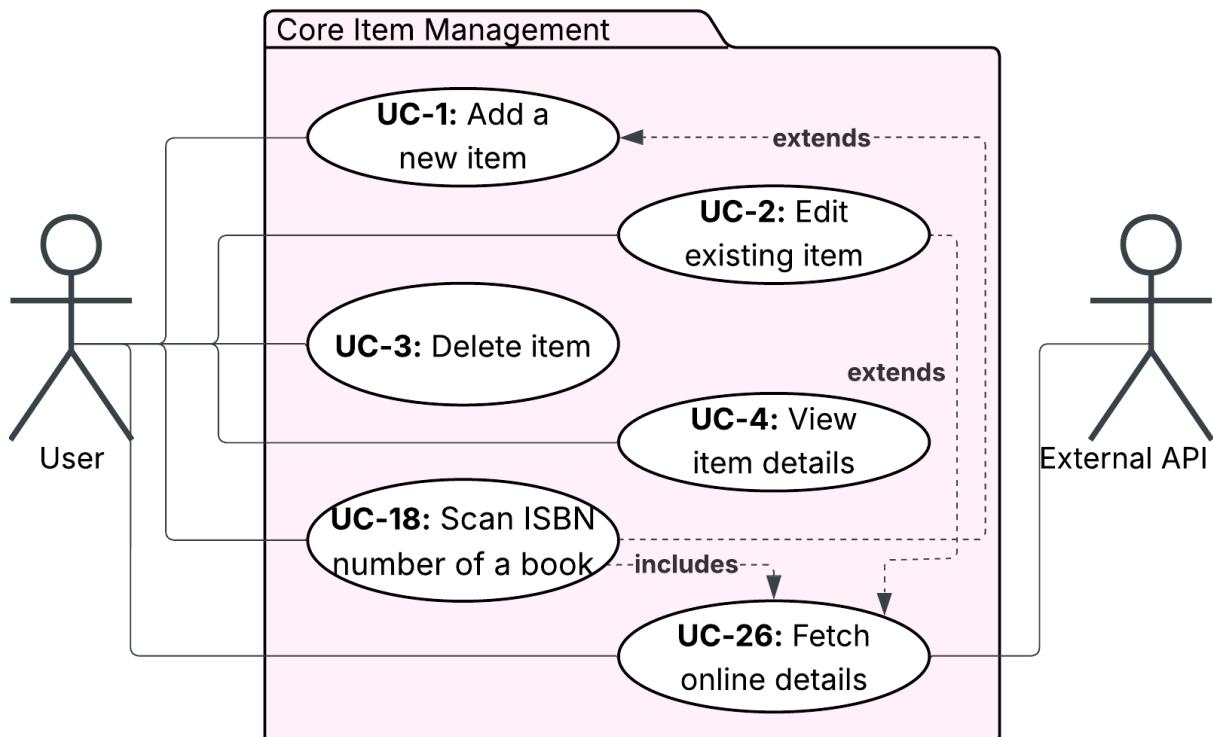
	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

	interface.
Entry condition	* The User is not currently logged in.
Exit condition	<ul style="list-style-type: none"> <li>* The User is successfully authenticated via OAuth 2.0 (FR-7.2).</li> <li>* An access token is securely stored (NFR-SEC-1).</li> </ul>
Quality requirements	<ul style="list-style-type: none"> <li>* Authentication shall be handled exclusively via OAuth 2.0 (NFR-SEC-1).</li> <li>* Refresh tokens shall be stored securely in HTTP-only cookies (NFR-SEC-1).</li> </ul>

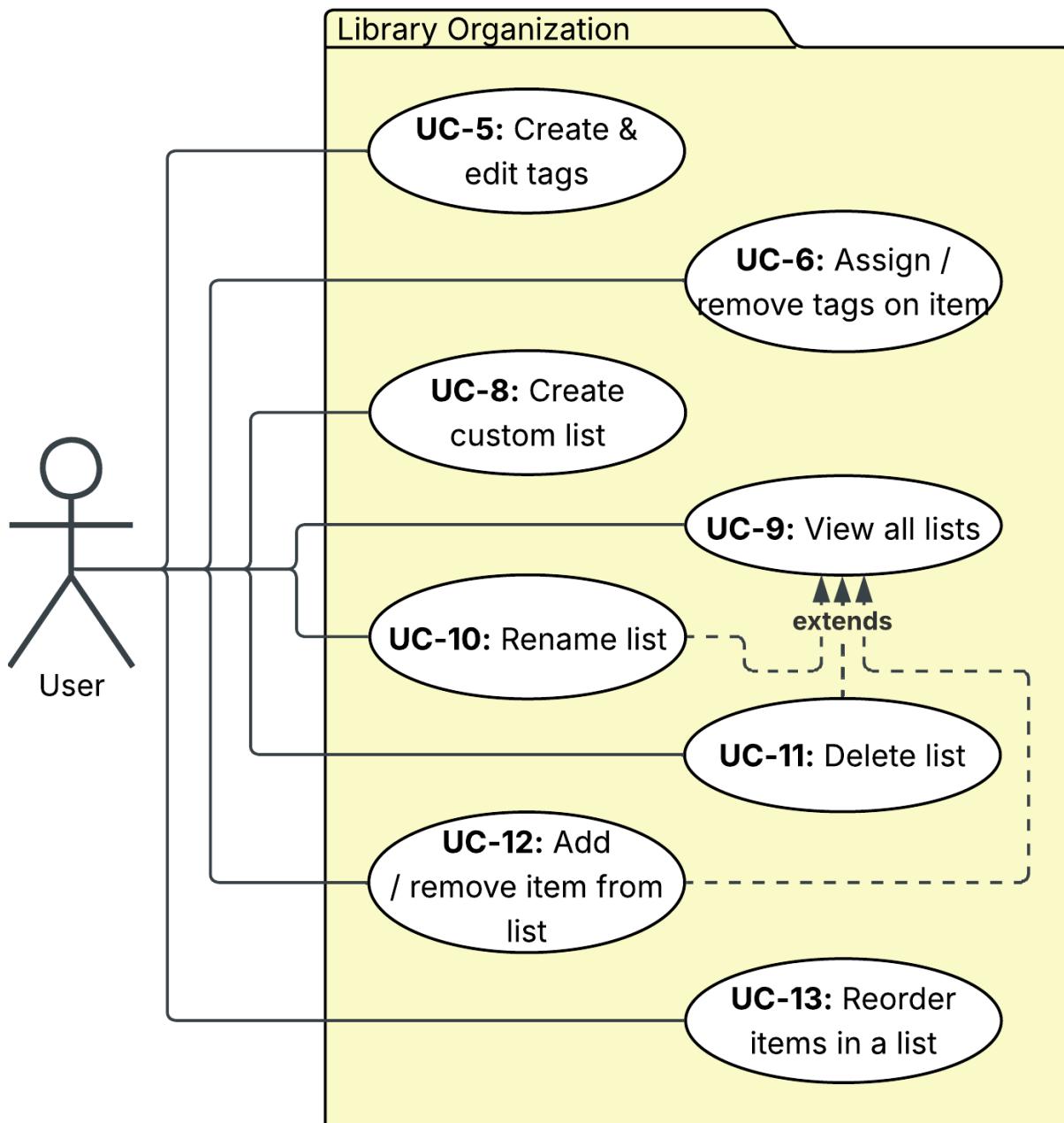
	<p align="center"><b>REQUIREMENTS ANALYSIS DOCUMENT</b> for <b>PERSONAL LIBRARY MANAGEMENT</b> <b>SYSTEM</b></p>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------



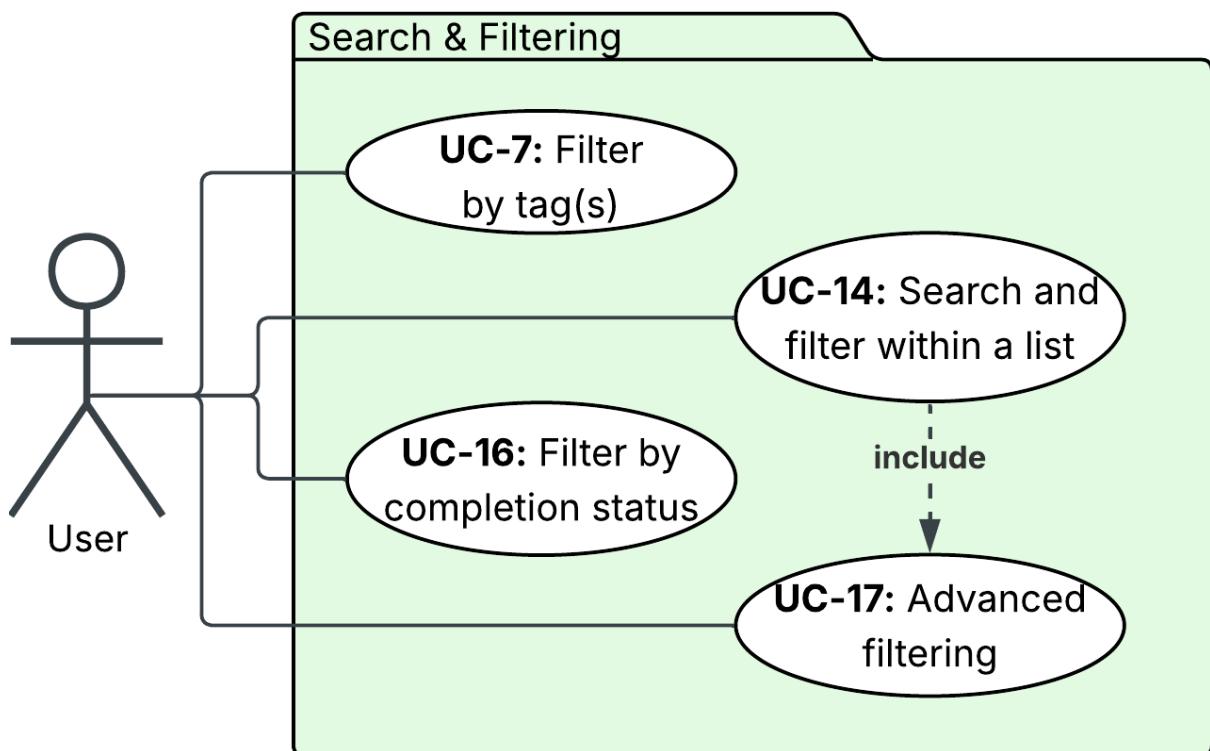
	<p align="center"><b>REQUIREMENTS ANALYSIS DOCUMENT</b> for <b>PERSONAL LIBRARY MANAGEMENT</b> <b>SYSTEM</b></p>	<p align="right">Issue No: 1.0 Issue Date: Nov 2025</p>
---	--	---



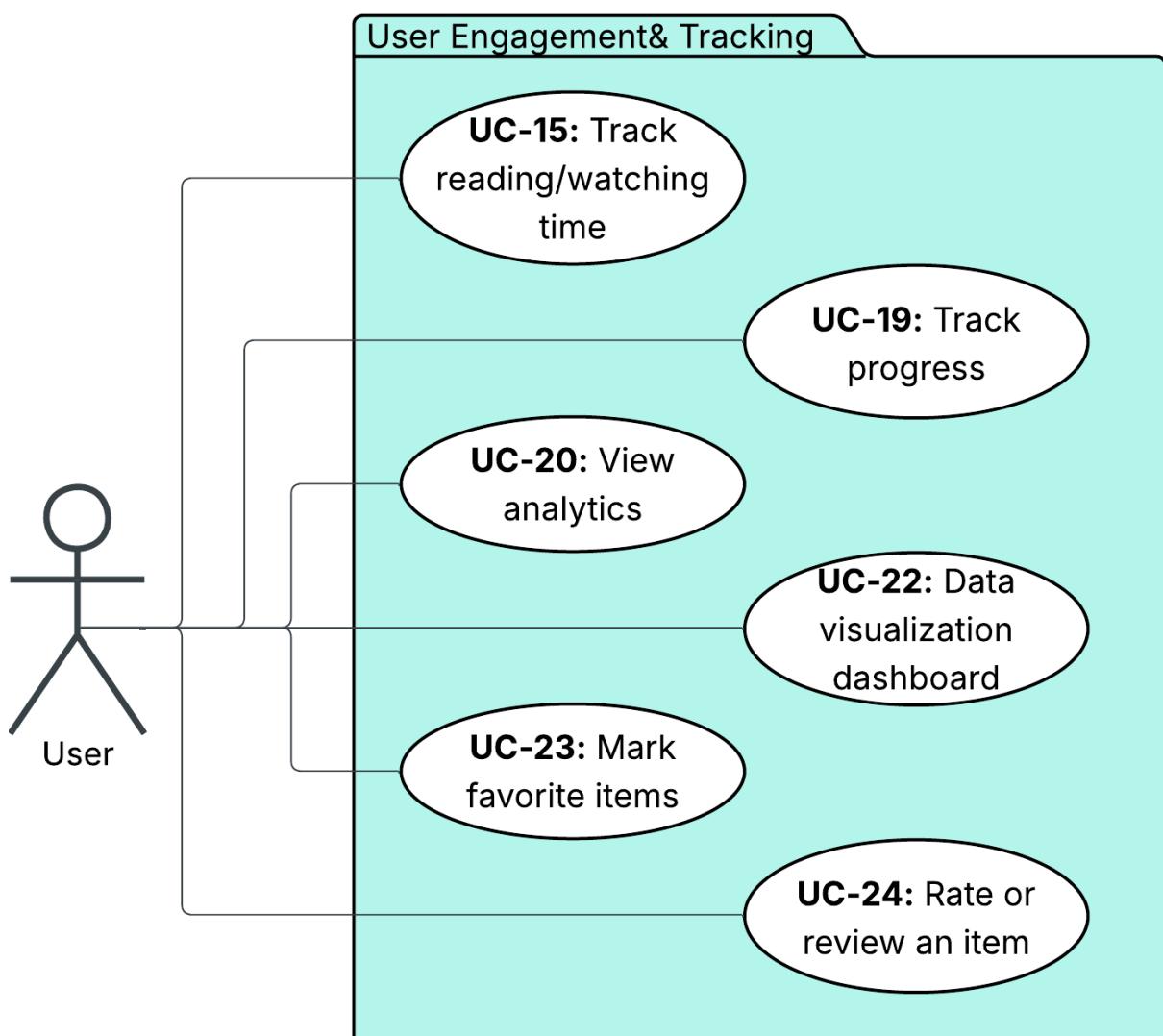
	<p align="center"><b>REQUIREMENTS ANALYSIS DOCUMENT</b> for <b>PERSONAL LIBRARY MANAGEMENT</b> <b>SYSTEM</b></p>	<p align="right">Issue No: 1.0 Issue Date: Nov 2025</p>
---	--	---



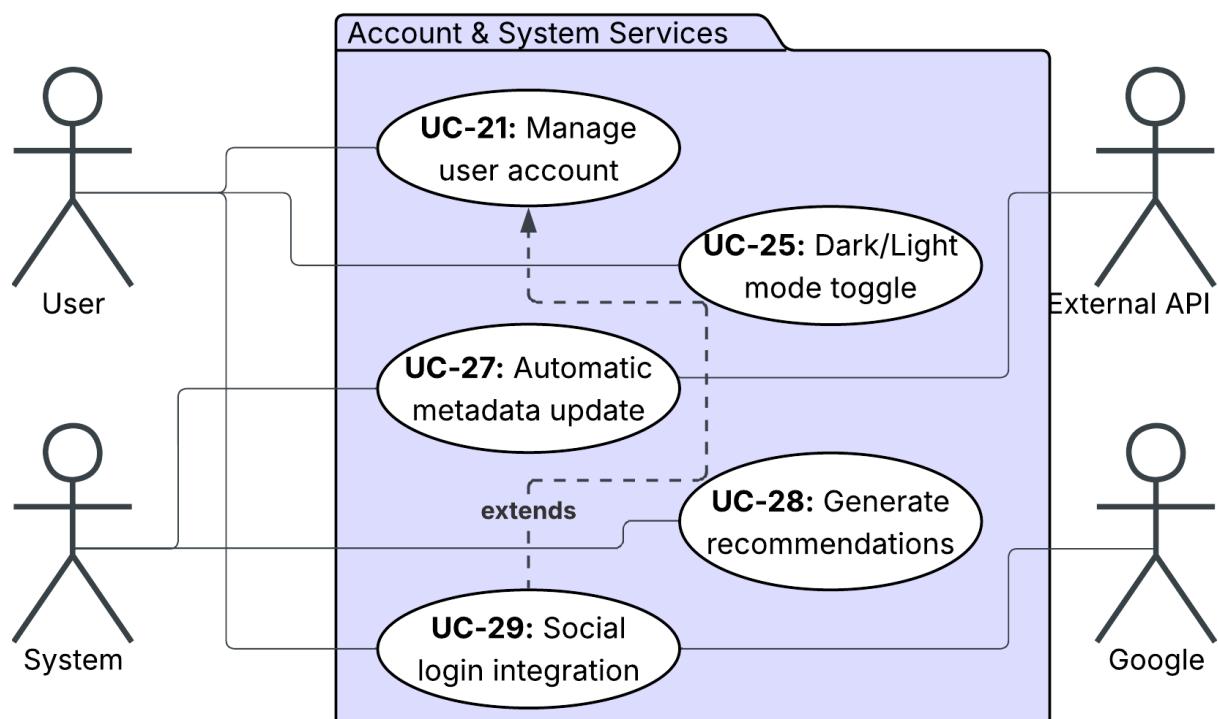
	<p>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</p>	<p>Issue No: 1.0 Issue Date: Nov 2025</p>
---	--	---

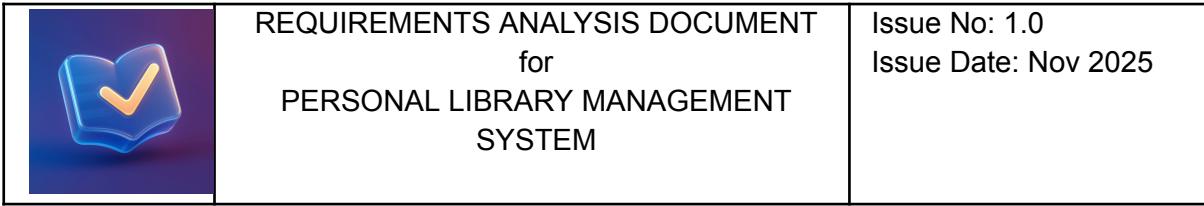


	<p align="center"><b>REQUIREMENTS ANALYSIS DOCUMENT</b> for <b>PERSONAL LIBRARY MANAGEMENT</b> <b>SYSTEM</b></p>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

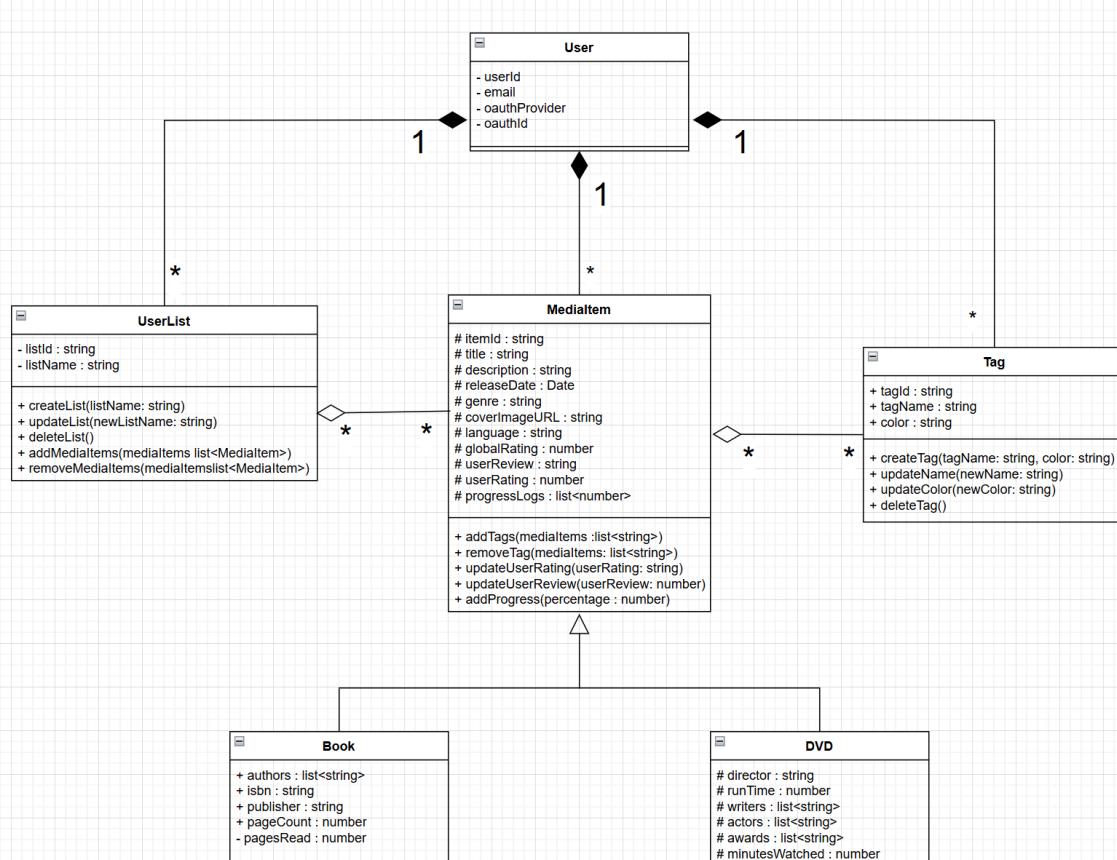


	<p align="center"><b>REQUIREMENTS ANALYSIS DOCUMENT</b> for <b>PERSONAL LIBRARY MANAGEMENT</b> <b>SYSTEM</b></p>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------





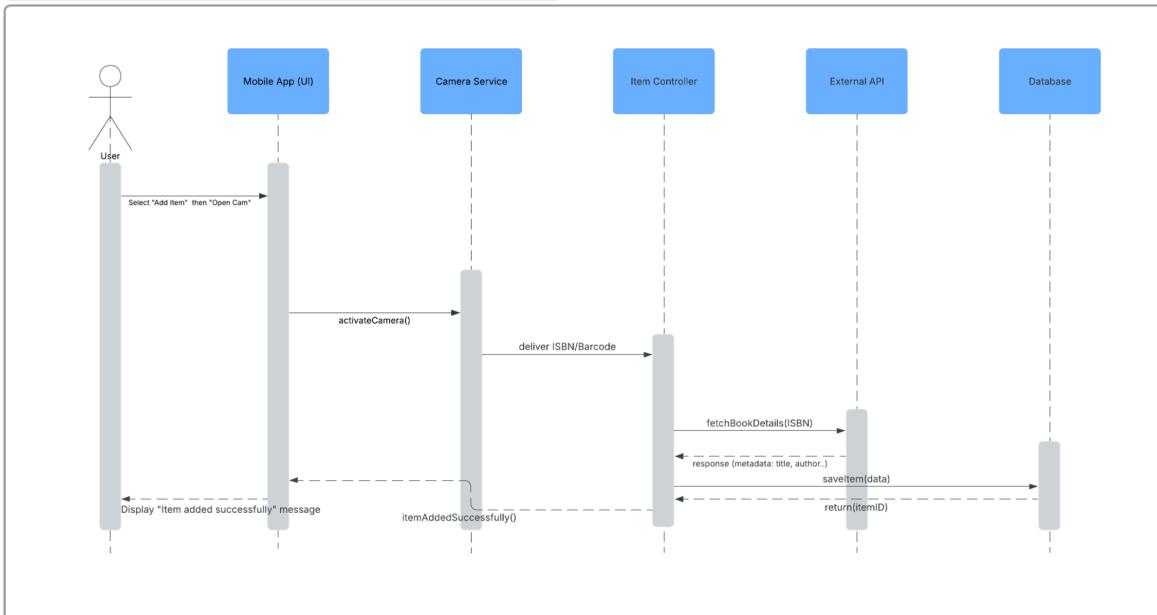
### 3.4.3 Object Model



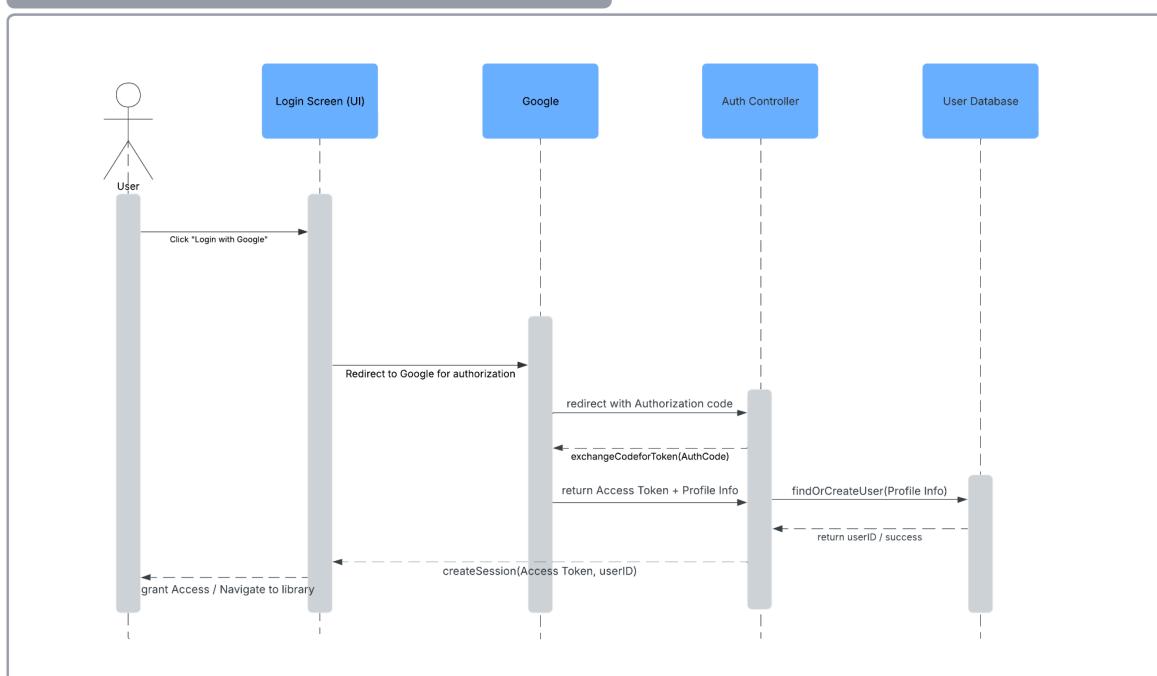
	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	<b>Issue No: 1.0 Issue Date: Nov 2025</b>
---	--	---

### 3.4.4 Dynamic Model

#### Adding a New Item via Camera/Barcode

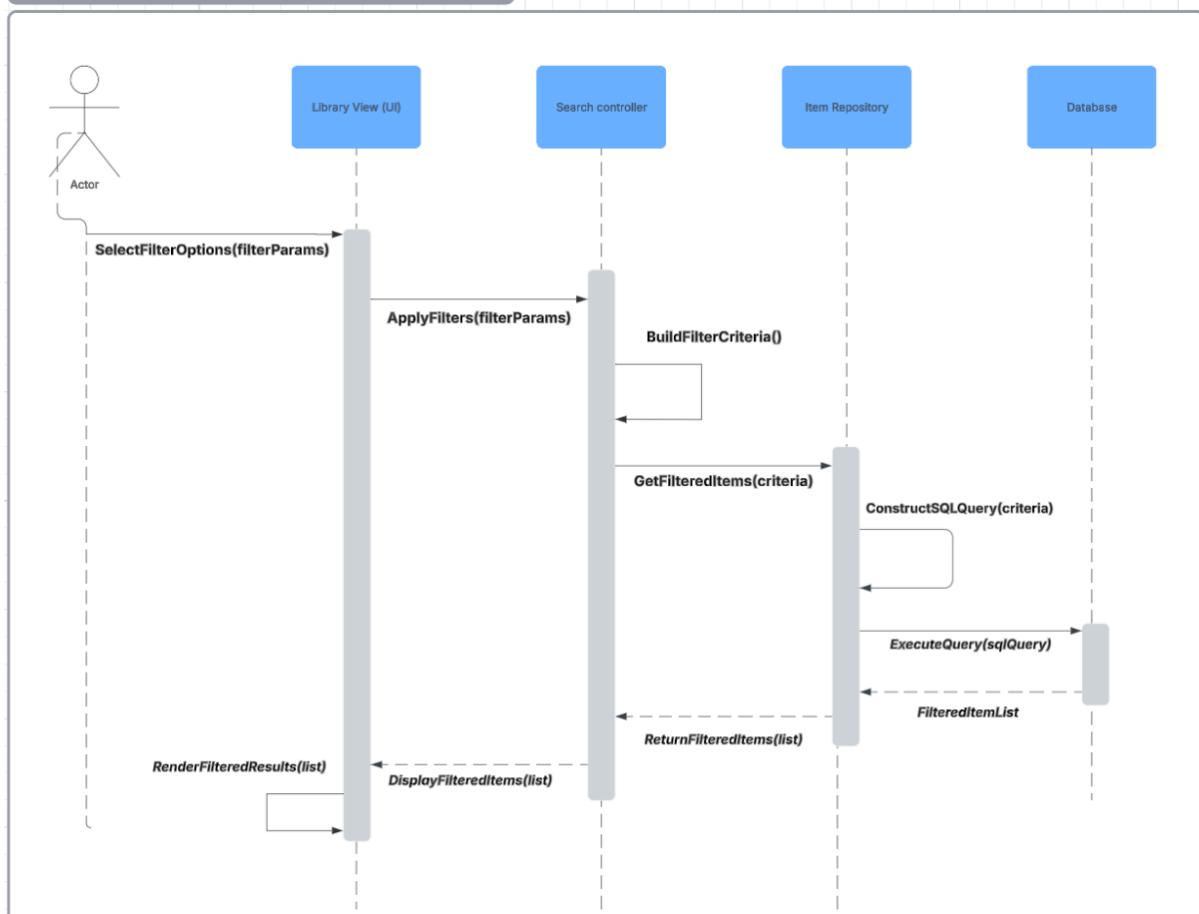


#### Logging in with Social Media (Google)



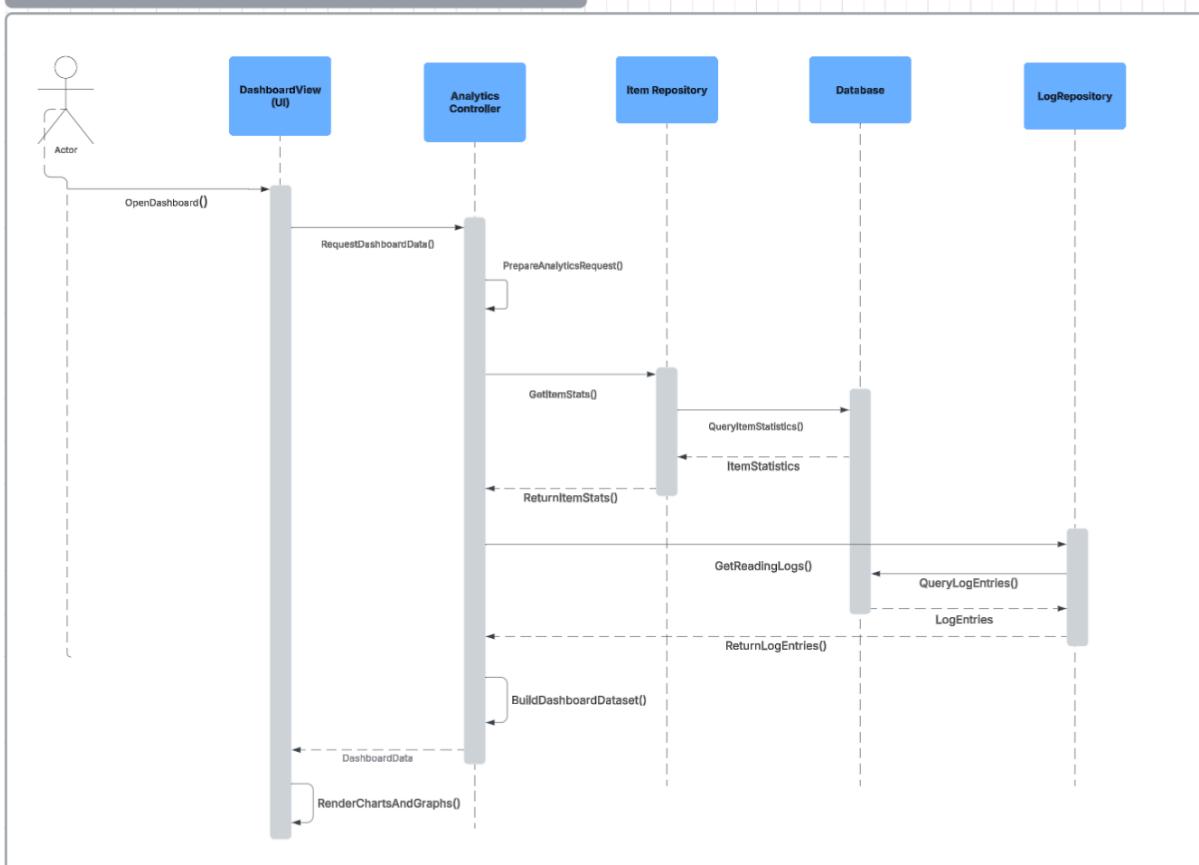
	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

## Applying Advanced Filtering



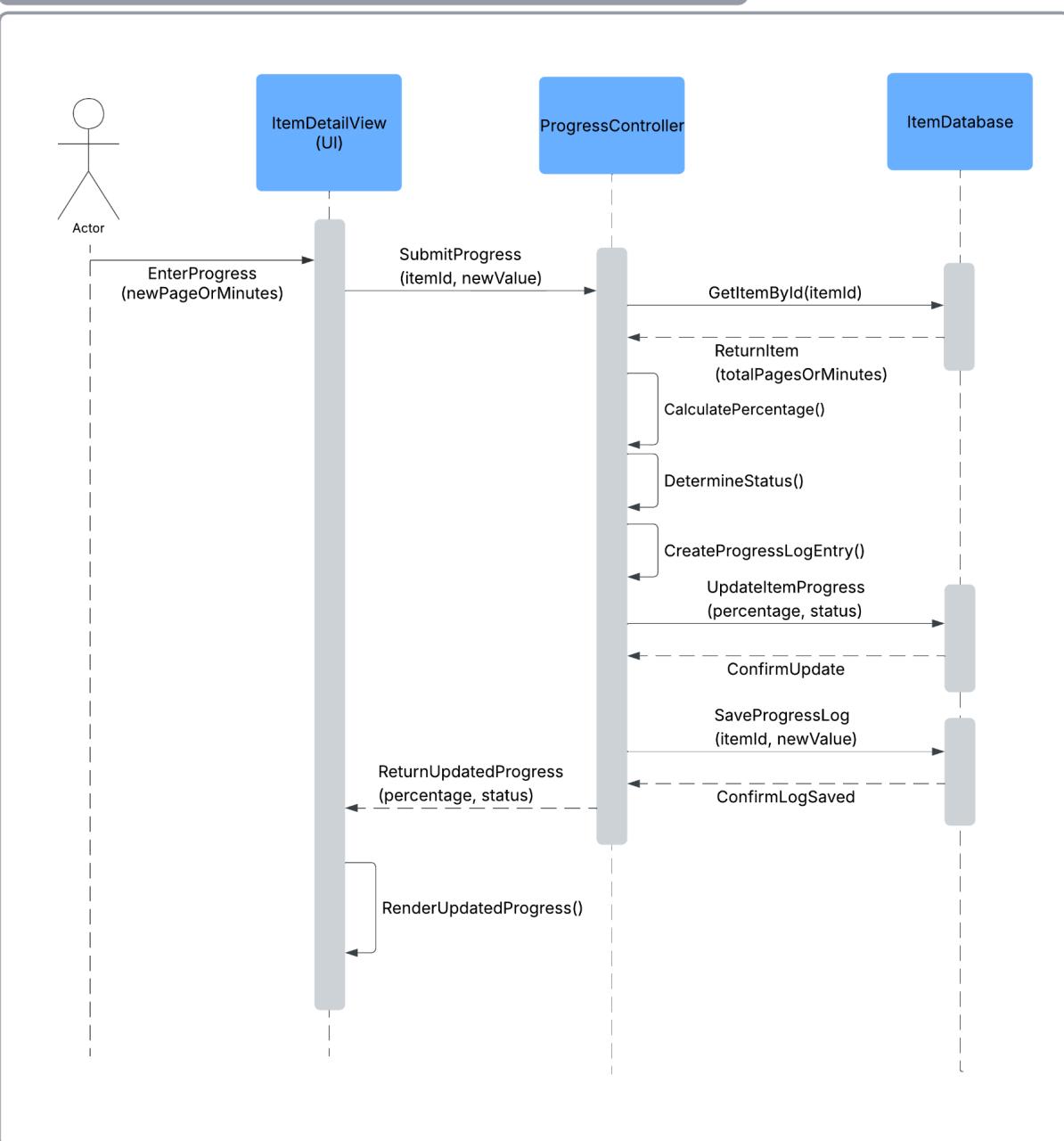
	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	<b>Issue No: 1.0 Issue Date: Nov 2025</b>
---	--	---

### Viewing the Data Visualization Dashboard



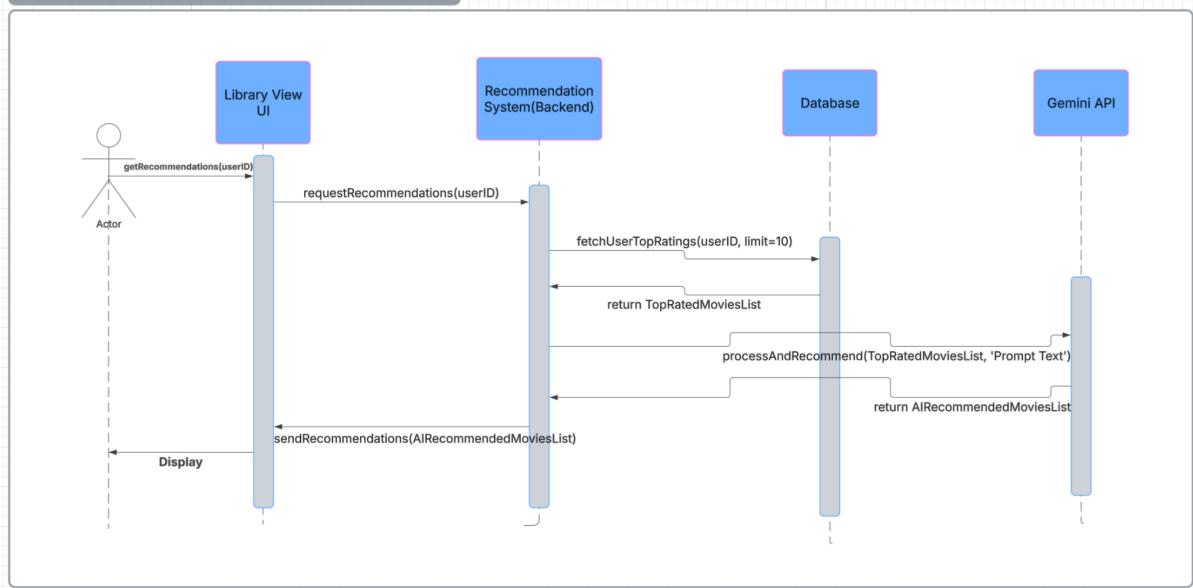
	<b>REQUIREMENTS ANALYSIS DOCUMENT</b> for <b>PERSONAL LIBRARY MANAGEMENT</b> <b>SYSTEM</b>	<b>Issue No: 1.0</b> <b>Issue Date: Nov 2025</b>
---	---	---

## Updating Reading/Watching Progress

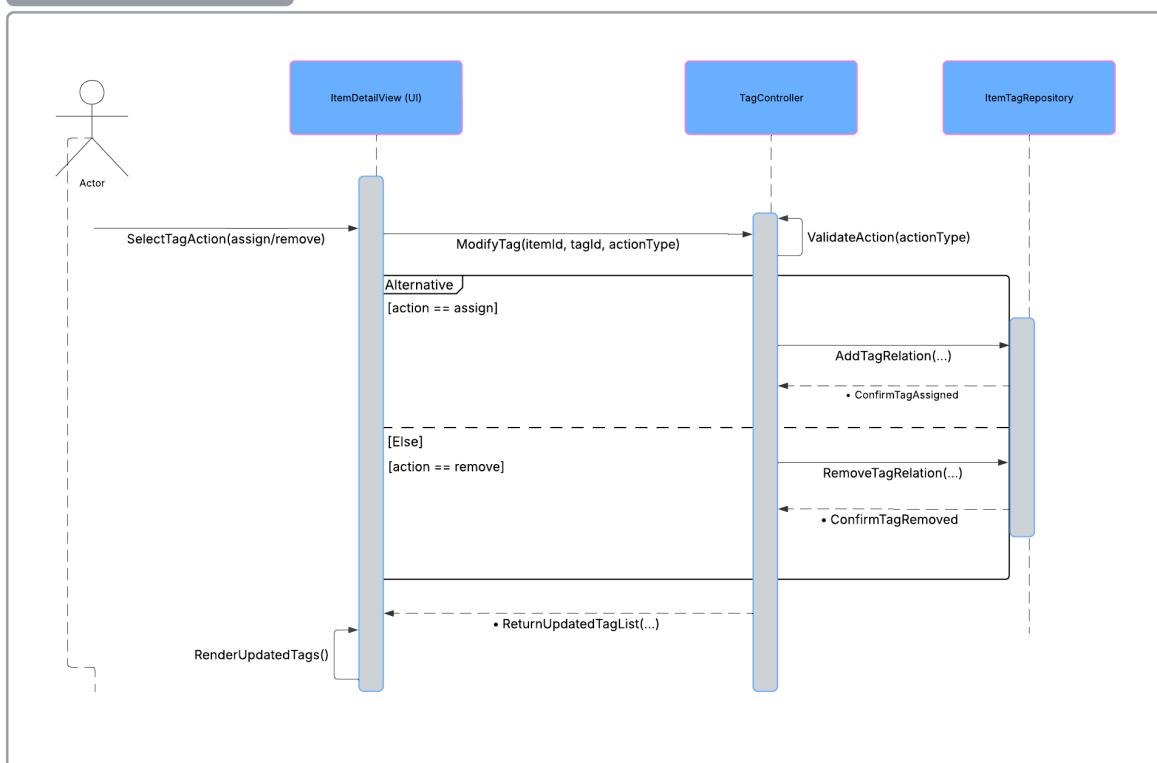


	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	<b>Issue No: 1.0 Issue Date: Nov 2025</b>
---	--	---

### Movie Recommendation Generation Sequence

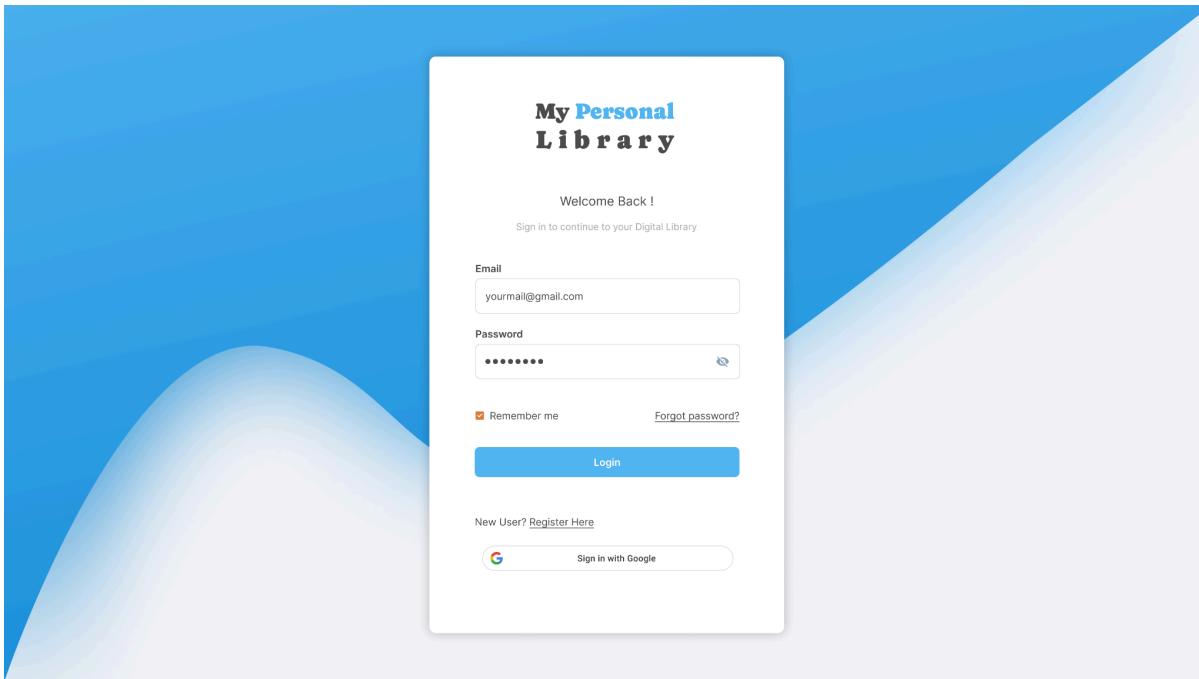


### Tag Management



	<p align="center"><b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b></p>	<p><b>Issue No: 1.0</b> <b>Issue Date: Nov 2025</b></p>
---	--	---

### 3.4.5 User interfaces and Mock-ups



**Good Morning**

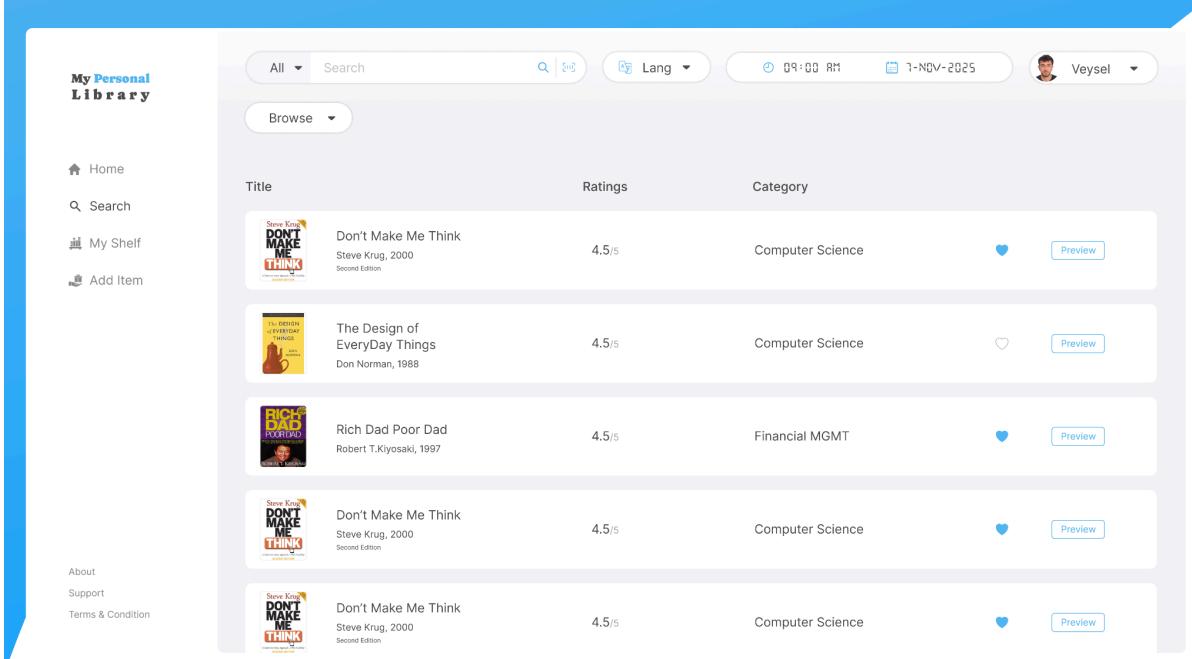
**Recommended for You**

Steve Krug: <b>DON'T MAKE ME THINK</b> A Common Sense Approach to Web Usability, Second Edition Don't Make Me Think Steve Krug, 2000 4.5/5	The Design of Every... Don Norman, 1988 4.5/5	Sprint : How to solve... Jake Knapp, 2000 4.5/5	LEAN UX Designing Great Products Jeff Gothelf and Josh Seiden Learn UX : Design Gr... Jeff Gothelf, 2018 4.5/5	The Road to React Steve Krug, 2000 4.5/5	Rich Dad Poor Dad Robert T. Kiyosaki, 1997 5/5	Harry Potter and The... J.K. Rowling, 2002 4.5/5	You Don't Know JS: Scope Closures Kyle Simpson, 2014 4.5/5
--	---	---	---	--	--	--	--

**Recent Readings**

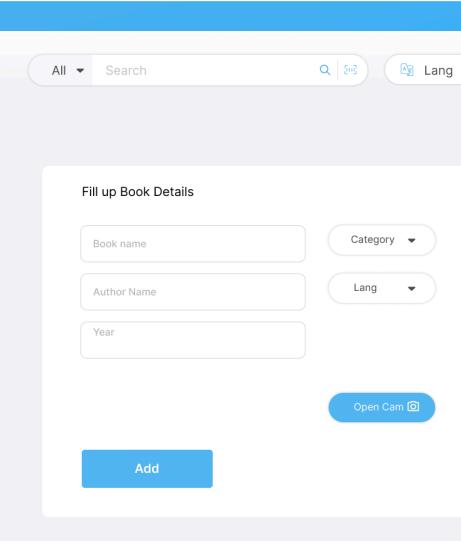
Steve Krug: <b>DON'T MAKE ME THINK</b> A Common Sense Approach to Web Usability, Second Edition Don't Make Me Think Steve Krug, 2000 4.5/5	The Design of Every... Don Norman, 1988 4.5/5	Sprint : How to solve... Jake Knapp, 2000 4.5/5	LEAN UX Designing Great Products Jeff Gothelf and Josh Seiden Learn UX : Design Gr... Jeff Gothelf, 2018 4.5/5	The Road to React Steve Krug, 2000 4.5/5	Rich Dad Poor Dad Robert T. Kiyosaki, 1997 5/5	Harry Potter and The... J.K. Rowling, 2002 4.5/5	You Don't Know JS: Scope Closures Kyle Simpson, 2014 4.5/5
--	---	---	---	--	--	--	--

	<p align="center"><b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b></p>	<b>Issue No: 1.0</b> <b>Issue Date: Nov 2025</b>
---	--	---



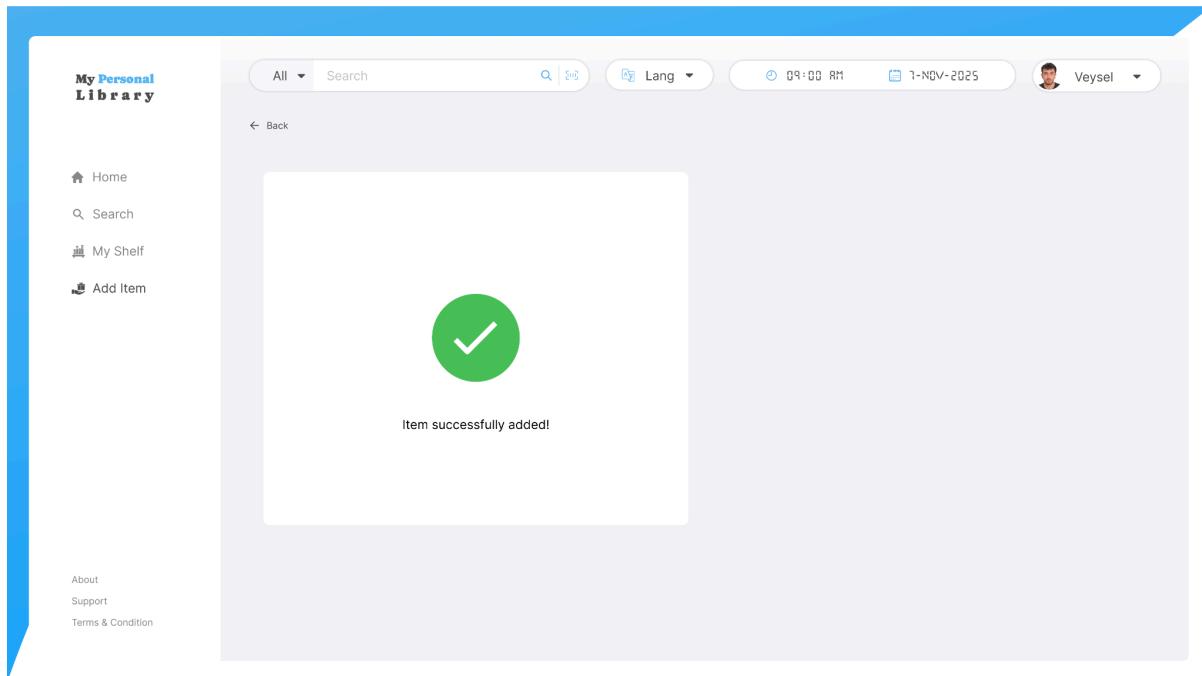
The screenshot shows the main interface of the "My Personal Library" application. On the left, there's a sidebar with links for Home, Search, My Shelf, and Add Item. The main area displays a list of books with columns for Title, Ratings, and Category. Each book entry includes a small thumbnail, the title, author, rating (4.5/5), category (e.g., Computer Science or Financial MGMT), a heart icon for favoriting, and a "Preview" button.

Title	Ratings	Category
Don't Make Me Think Steve Krug, 2000 Second Edition	4.5/5	Computer Science
The Design of EveryDay Things Don Norman, 1988	4.5/5	Computer Science
Rich Dad Poor Dad Robert T.Kiyosaki, 1997	4.5/5	Financial MGMT
Don't Make Me Think Steve Krug, 2000 Second Edition	4.5/5	Computer Science
Don't Make Me Think Steve Krug, 2000 Second Edition	4.5/5	Computer Science



The screenshot shows the "Add Item" form. It has a header "Fill up Book Details" and fields for Book name, Author Name, and Year. There are dropdown menus for Category and Lang. A "Open Cam" button is available for scanning barcodes. At the bottom is a large blue "Add" button.

	REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------



## 4. Glossary of Terms

Term	Definition
<b>Media Item</b>	An entry in the user's library, such as a book or DVD.
<b>Book</b>	A media item with attributes like title, author, and page count.
<b>DVD</b>	A media item with attributes like title, duration, director.
<b>Category</b>	A classification of a media item (e.g., Book, DVD).
<b>Tag</b>	A user-defined label applied to one or more media items to support flexible organization.

	REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

<b>User List</b>	A custom list created by the user to group selected media items (e.g., “Favorites”).
<b>Progress Tracking</b>	Recording reading or viewing progress on a media item.
<b>Progress Log</b>	Historical list of progress updates.
<b>Recommendation Engine</b>	External AI service used to generate personalized media suggestions.
<b>External Metadata Service</b>	APIs such as Google Books or IMDb used to enrich item information.
<b>Graceful Degradation</b>	System behavior in which core features remain operational even when external APIs fail.
<b>Filtering</b>	Showing items that match selected criteria.
<b>Sorting</b>	Ordering items by a chosen attribute (title, date, etc.).
<b>Search Query</b>	Keywords or parameters entered to locate items in the library.
<b>User Consent</b>	Explicit approval required to send personal data to the AI recommendation service.
<b>Authentication</b>	Verifying user identity using OAuth 2.0.

## 5. Traceability

This section establishes the relationship between the problems identified in the **Current System** (Section 2) and the requirements defined in Sections **3.2 (Functional Requirements)** and **3.3 (Non-Functional Requirements)**. Each problem in the project's problem statement is mapped to the system capabilities that resolve it, ensuring that the design and implementation directly address every identified deficiency.

	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

## 5.1 Traceability to Problem Statement

Problem in Current System	Description	Requirements that Address the Problem
<b>Lack of Personal Focus</b>	Existing tools are built for institutions, not personal collections. They don't track personal details like ownership, condition, or loan status.	<ul style="list-style-type: none"> <li>• FR-1 (Media CRUD)</li> <li>• FR-2 (Categorization)</li> <li>• FR-3 (Tag Management)</li> <li>• FR-5 (User Lists)</li> <li>• NFR-US (Usability)</li> <li>• NFR-SUP (Maintainability)</li> </ul>
<b>No Progress Tracking</b>	Users cannot track reading or viewing progress.	<ul style="list-style-type: none"> <li>• FR-6.1 (Track Book Progress)</li> <li>• FR-6.2 (Track DVD Progress)</li> <li>• FR-6.3 (Progress Logging)</li> <li>• FR-6.4 (Visual Indicators)</li> </ul>
<b>Limited Media Support</b>	Tools focus mostly on books and do not support mixed media.	<ul style="list-style-type: none"> <li>• FR-2.1 (Categorization)</li> <li>• FR-1 (CRUD Operations)</li> <li>• FR-8.2 (DVD Metadata)</li> <li>• NFR-SUP-1 (Modularity for new media types)</li> </ul>
<b>No Custom Lists or Tagging</b>	Users cannot create flexible lists or tag items based on personal needs.	<ul style="list-style-type: none"> <li>• FR-3 (Tag Management)</li> <li>• FR-5 (List Operations)</li> </ul>
<b>Absence of Content Enrichment</b>	Systems do not pull metadata automatically from the internet.	<ul style="list-style-type: none"> <li>• FR-8.1 (Book Metadata)</li> <li>• FR-8.2 (DVD Metadata)</li> <li>• FR-8.3 (Metadata Updates)</li> <li>• NFR-REL-2 (Graceful Degradation)</li> </ul>
<b>No Recommendation Features</b>	Systems do not provide personalized suggestions.	<ul style="list-style-type: none"> <li>• FR-9 (AI-Based Recommendations)</li> <li>• NFR-DATA-1 (User Consent)</li> </ul>

	<b>REQUIREMENTS ANALYSIS DOCUMENT for PERSONAL LIBRARY MANAGEMENT SYSTEM</b>	Issue No: 1.0 Issue Date: Nov 2025
---	--	---------------------------------------

<b>Fragmented User Experience</b>	Users rely on notebooks, spreadsheets, or mismatched apps.	<ul style="list-style-type: none"> <li>• FR-1 to FR-9 combined</li> <li>• NFR-US (mobile-first)</li> <li>• NFR-PERF (performance)</li> <li>• NFR-PKG-1 (containerization)</li> <li>• NFR-REL-1 (availability)</li> </ul>
<b>No unified filtering/searching</b>	Existing tools lack deep filtering, sorting, and search.	<ul style="list-style-type: none"> <li>• FR-4 (Search, Filter, Sort)</li> <li>• FR-5.8 (Search in Lists)</li> <li>• FR-5.9 (Filter in Lists)</li> </ul>

## 5.2 Traceability Summary

Every issue raised in the problem statement directly corresponds to at least one requirement in the proposed system. The PLMS addresses all deficiencies by providing:

- Mixed-media support
- Deep personalization (tags, lists, progress)
- Automated metadata enrichment
- AI-powered recommendations
- Robust search/filtering
- Strong security, maintainability, and performance guarantees

This ensures that the system not only fills the gaps identified in existing solutions but exceeds them through intelligent features and modern architectural practices.