SOFTWARE ENGINEERING SEMINAR

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Team Members:

Wilder Steven Hernández Manosalva - 20212020135

Jhon Javier Castañeda Alvarado - 20211020100

Systems Engineering

BookWiseUD Architecture Document

Bogotá D.C 2025

Library System (BookWiseUD)

1. Introduction

This document presents the initial specification for a Library System, whose purpose is to digitalize and optimize the management of bibliographic resources and lending services. The system will allow administrators to register, update, and consult books, manage user data, and efficiently control the borrowing and returning of copies. Likewise, it seeks to offer a user-friendly web interface to facilitate interaction for both librarians and registered users. The solution will be developed on a modular and scalable architecture that allows the integration of new functionalities in the future.

2. Objetives

2.1 General Objective

To develop a library management system that allows the administration of books, users, and loans in an organized and centralized manner, improving control and information query processes.

2.2 Specific Objectives

- Define the structure of the database.
- Implement basic management functionalities for books and users.
- Include a loan module to manage borrowings and returns.
- Design the main use cases and interaction diagrams.
- Develop an intuitive web interface for users.

3. Problem / Justification

Small libraries (schools, institutions, and communities) often manage their book inventory and loan records manually or with impractical tools such as spreadsheets. This leads to information loss, confusion about the status of items, and difficulties in performing quick queries. The system seeks to solve this problem by providing a digital tool that allows:

- Register and consult books with their main data.
- Register and manage library users.
- Control loans and returns with a usage history.

4. Scope

- Complete CRUD for books and users.
- Loan registration (create loan and mark return).
- REST API for frontend consumption.
- React frontend with views to list, create, and edit books and users.

5. Functional Requirements (summary)

1. RF-01: Register book (title, author, ISBN, year, category, status, quantity).
2. RF-02: Edit / Delete / Query book.
3. RF-03: Register user (name, ID, email, phone, role).
4. RF-04: Edit / Delete / Query user.
5. RF-05: Register loan (user, book, loan date, expected return date).
6. RF-06: Mark return (actual return date, change book status).

6. Non-Functional Requirements

- RNF-01: Documented REST API.
- RNF-02: Basic authentication in the backend (JWT) to protect endpoints.
- RNF-03: Modular architecture with separated layers.
- RNF-04: Possibility to expand with new functionalities.

7. Proposed Architecture

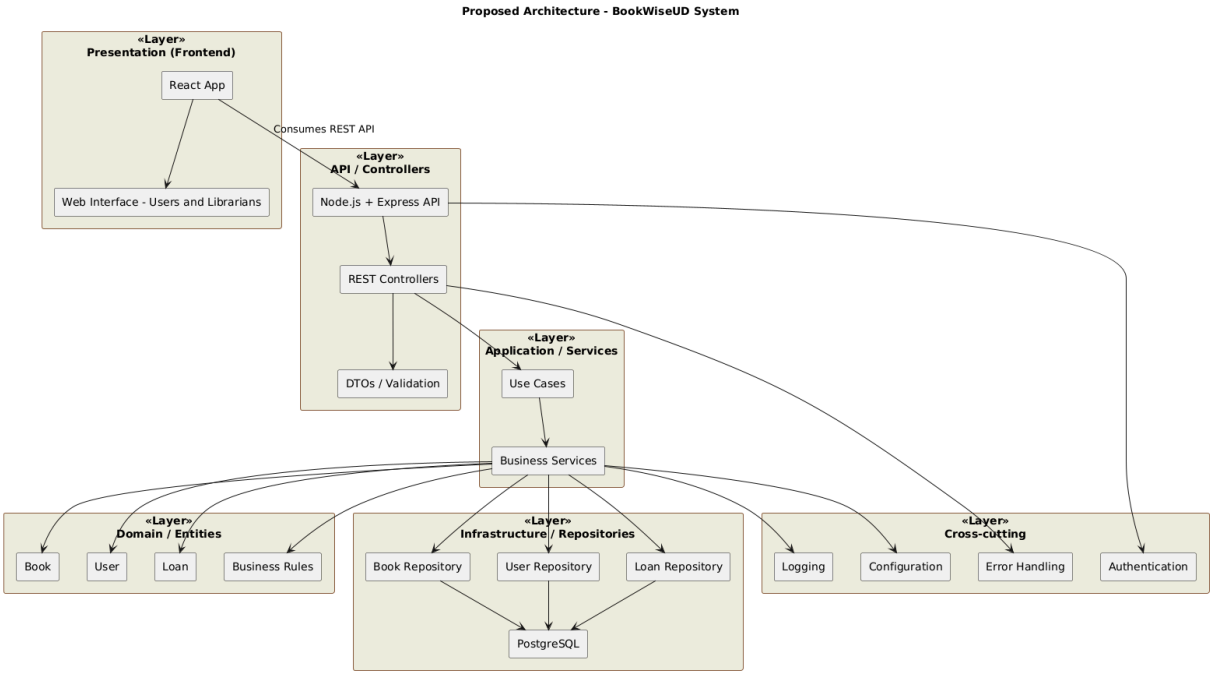

Image 1. Complete Architecture Diagram

General pattern: Layered architecture + Clean Architecture / Hexagonal principles applied in a lightweight manner.

Layers / Modules

- **API / Controllers:** REST endpoints, input validation (DTOs).
- **Application / Services:** Use case logic.
- **Domain / Entities:** Domain models (Book, User, Loan) and business rules.
- **Infrastructure / Repositories:** Concrete persistence implementations (PostgreSQL).

- **Cross-cutting:** Authentication, logging, configuration, error handling.

8. Suggested Technologies and Tools

- Frontend: React + TypeScript.
- Backend: Node.js + TypeScript y Express.
- Database: PostgreSQL.
- Backend testing: Jest + Supertest.
- Frontend testing: React Testing Library + Jest.
- Containerization: Docker.

9. Use Cases

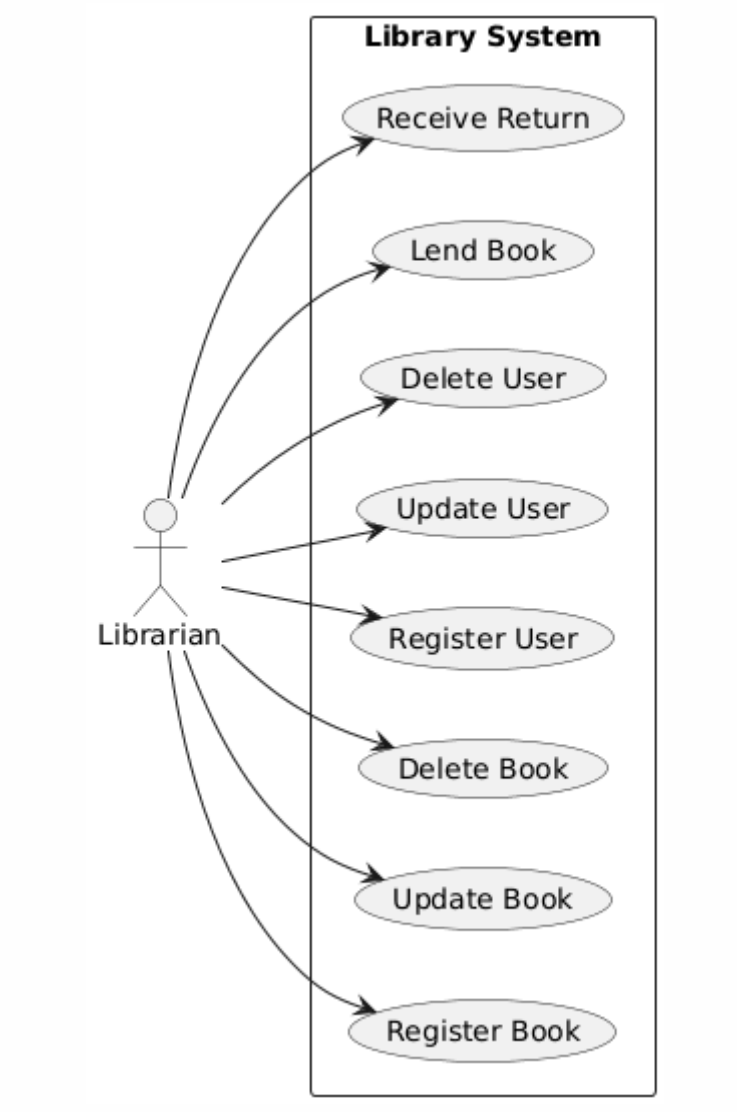

Image 2. Use Case Diagram for the Librarian.

The librarian is responsible for the comprehensive administration of the system. Their main functions include registering new books in the database, ensuring that information such as title, author, category, and availability is up to date. They can also modify or delete book records when necessary, guaranteeing proper inventory management. In addition, the librarian is responsible for managing users, allowing the registration, updating, or deletion of their personal information. Another essential task is to control loans: assigning a book to a user, verifying its availability, and recording the return date. Finally, the librarian also receives users' returns, updating the status of each copy and ensuring it becomes available in the system again.

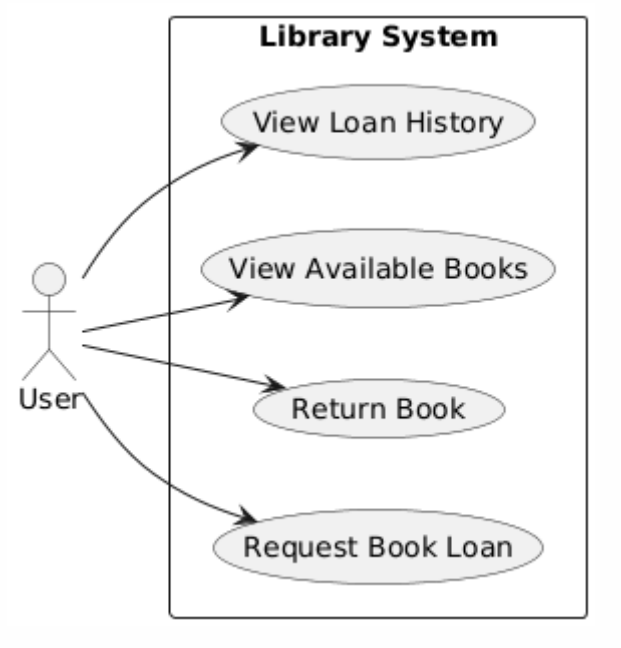

Image 3. Use Case Diagram for the User.

The user interacts with the system to access library services. They can consult the catalog of available books to see which copies are on loan or available. They have the option to request a loan, which is recorded in the system along with the corresponding dates. Once finished, they must return the book, allowing the librarian to update the inventory. Additionally, they can review their history of previous loans.

Proposed Mockup

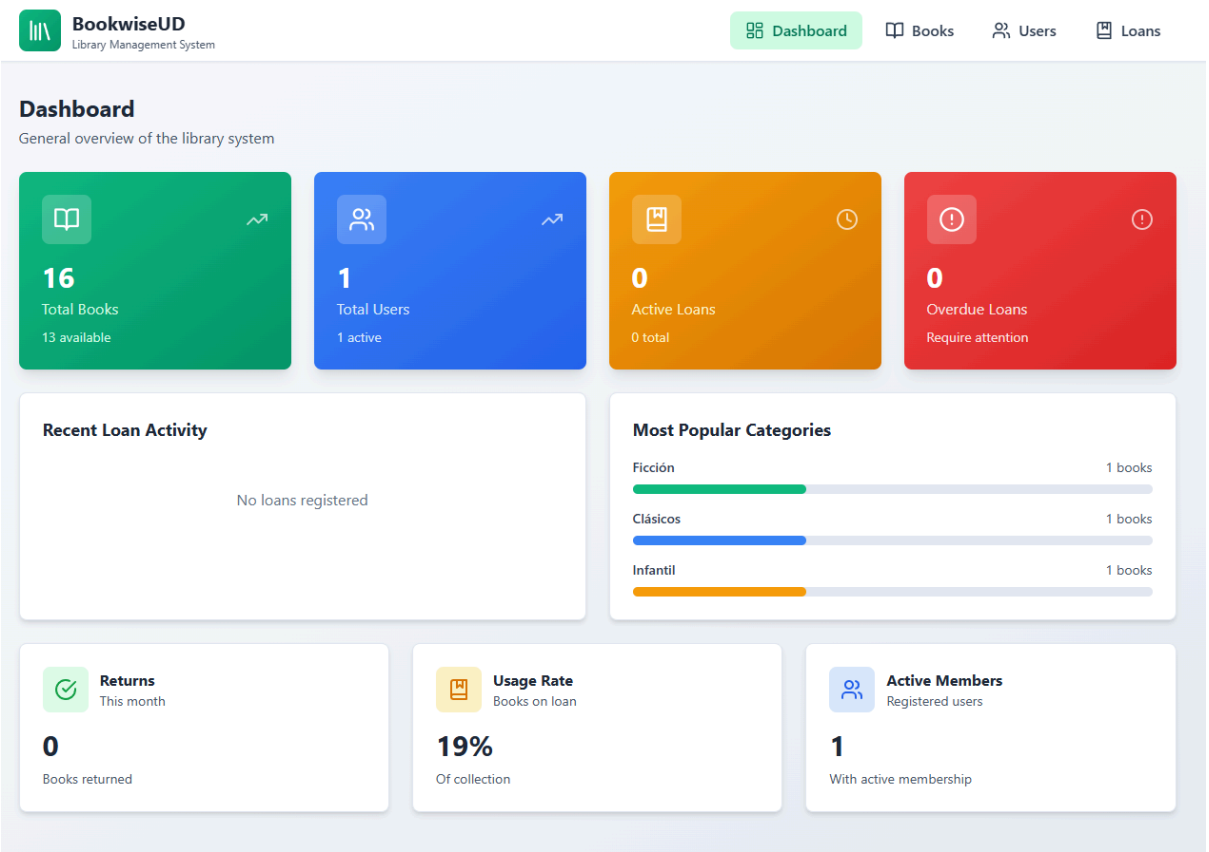

Image 4. Main dashboard with information on books and users.

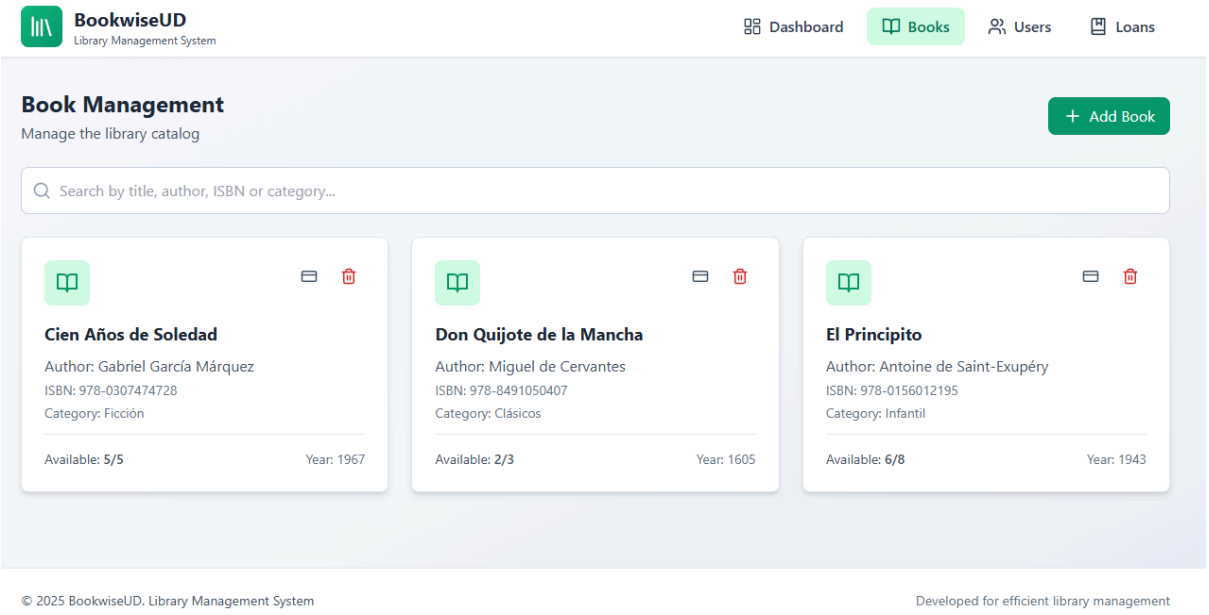

Image 4. Gallery showing all books and their availability.

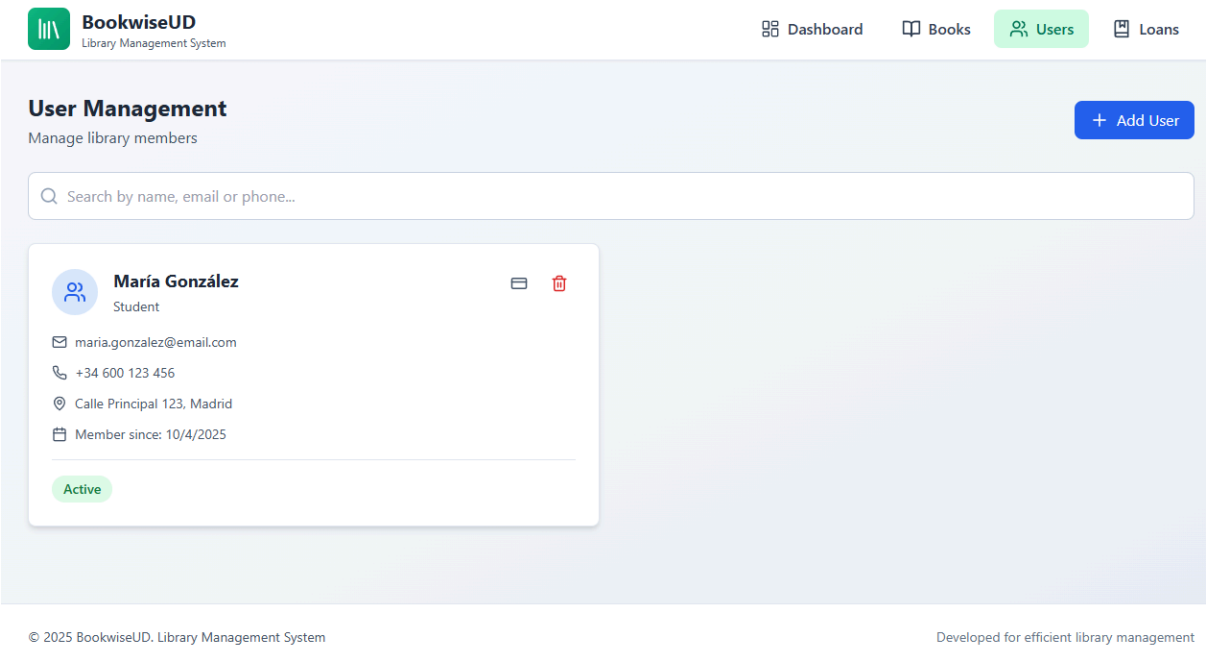

Image 5. User gallery with contact information.

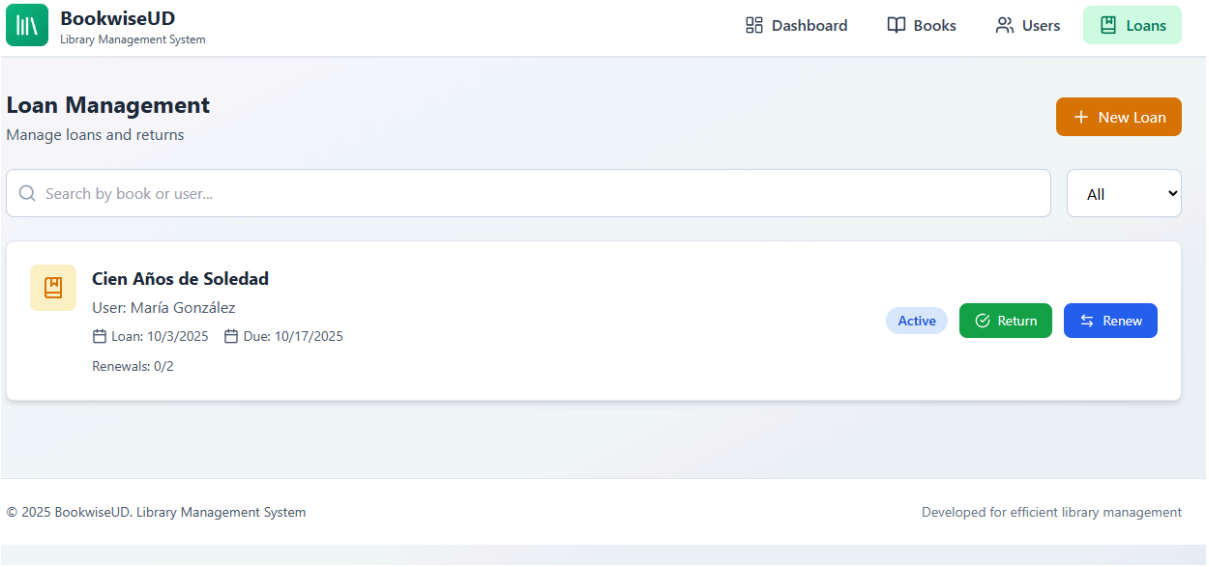

Image 6. Book loan manager for users.

Conclusions

The development of the library system represents a modern and efficient technological solution for managing bibliographic resources. Its implementation allows the automation of tasks that were traditionally performed manually, such as book registration, user management, and the control of loans and returns. This facilitates the organization of information, reduces human errors, and speeds up internal processes. Furthermore, the design of an intuitive web interface and a modular architecture provides a system that is adaptable,

scalable, and accessible from different devices, optimizing the experience for both librarians and end users.

The project demonstrates the importance of applying software engineering principles, such as layered separation, scalability, and the use of modern technologies (React, Node.js, and PostgreSQL). These elements ensure the system's maintainability and its capacity for future evolution. Likewise, the proposed model lays the foundation for incorporating new functionalities, such as notifications or usage statistics, without compromising stability. Overall, the library system not only digitalizes library management but also promotes efficiency, traceability, and the modernization of information services.