Project Title

App for a Librarian

1. Project Relevance

Libraries still play an important role in education, but many small and medium libraries continue to rely on manual or outdated systems to manage books, readers, and lending records. This often leads to errors, lost information, and inefficient use of time for librarians.

The **App for a Librarian** is proposed to simplify daily library operations by providing a centralized digital system. The application will help librarians manage book inventory, track borrowed books, and organize reader information more efficiently. This project is relevant because it modernizes library management while remaining simple and accessible for everyday use.

2. Target Users

The primary target users of the system are:

Librarians in schools and universities

Staff of small and medium-sized libraries

Library assistants responsible for book lending

The system is intended for internal use by library staff and not directly for readers at this stage.

3. Competitor Analysis

There are existing library management systems available, but many of them are either too complex or not suitable for small libraries.

Competitor	Strengths	Weaknesses
Koha	Open-source, feature-rich	Complex setup and configuration
LibSys	Professional library system	Expensive, over-featured
Manual record systems	Simple	Error-prone, inefficient

The proposed application focuses on **simplicity, clarity, and essential features**, making it more suitable for smaller libraries with limited technical resources.

4. Planned Features

The system is planned to include the following features:

Librarian authentication and access control

Book catalog management (add, update, remove books)

Reader registration and management
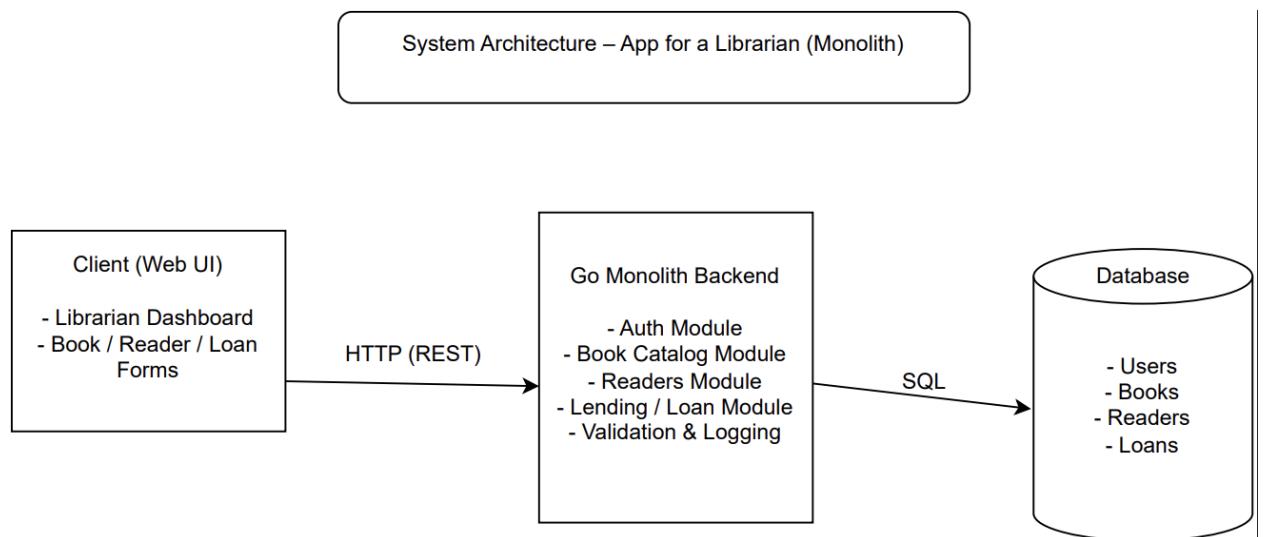
Book lending and return tracking

Search and filter functionality for books and readers
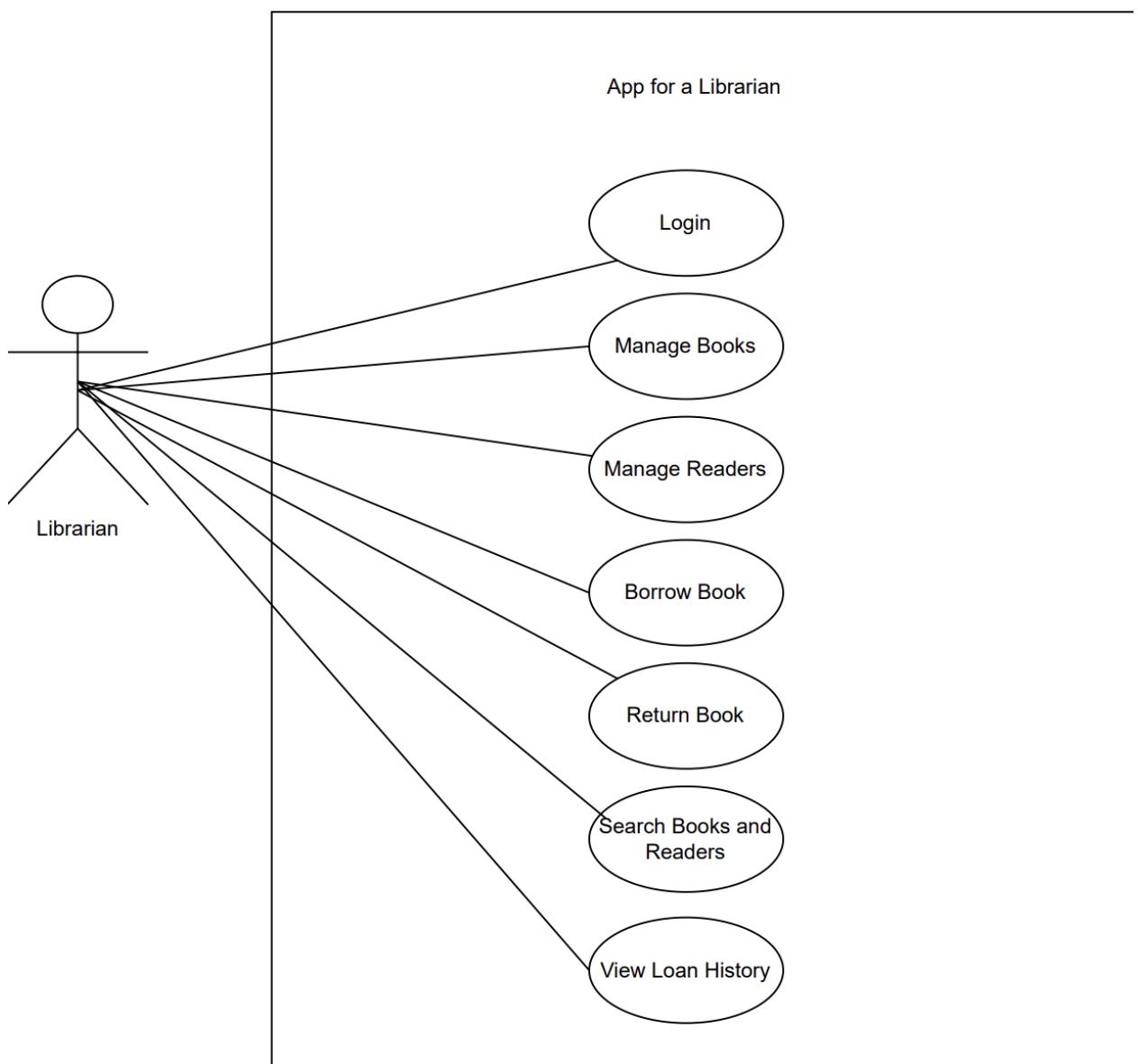
The exact user interface and final workflows will be defined in later stages of the project. At this milestone, the focus is on system planning and architecture rather than full implementation details.

Conclusion
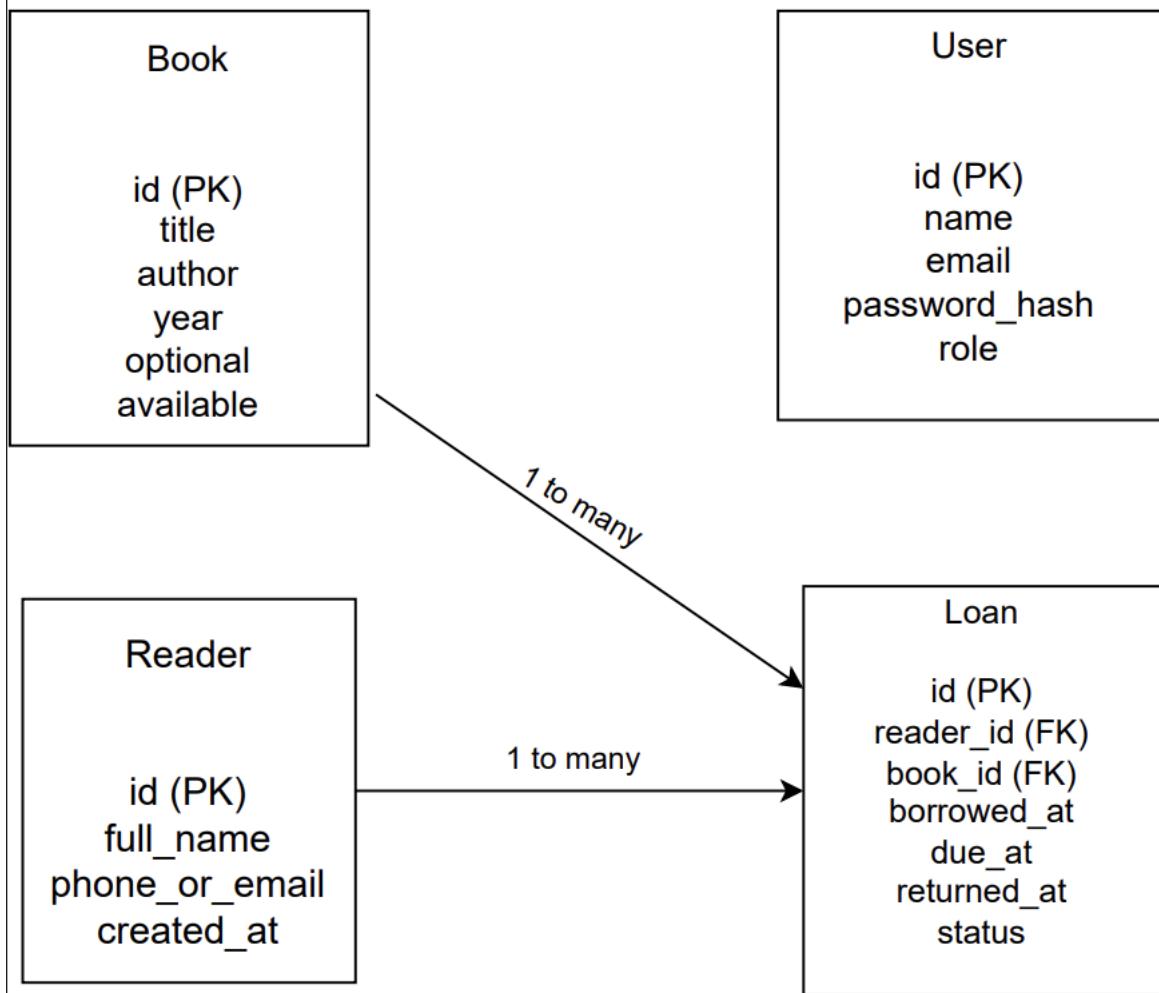
The App for a Librarian addresses real-world challenges faced by libraries by offering a lightweight and efficient management solution. The project emphasizes usability and clear structure, making it suitable for both academic purposes and potential real-world application.

System Architecture – App for a Librarian (Monolith)

Client (Web UI)

- Librarian Dashboard
- Book / Reader / Loan Forms

HTTP (REST)

Go Monolith Backend

- Auth Module
- Book Catalog Module
- Readers Module
- Lending / Loan Module
- Validation & Logging

SQL

Database

- Users
- Books
- Readers
- Loans
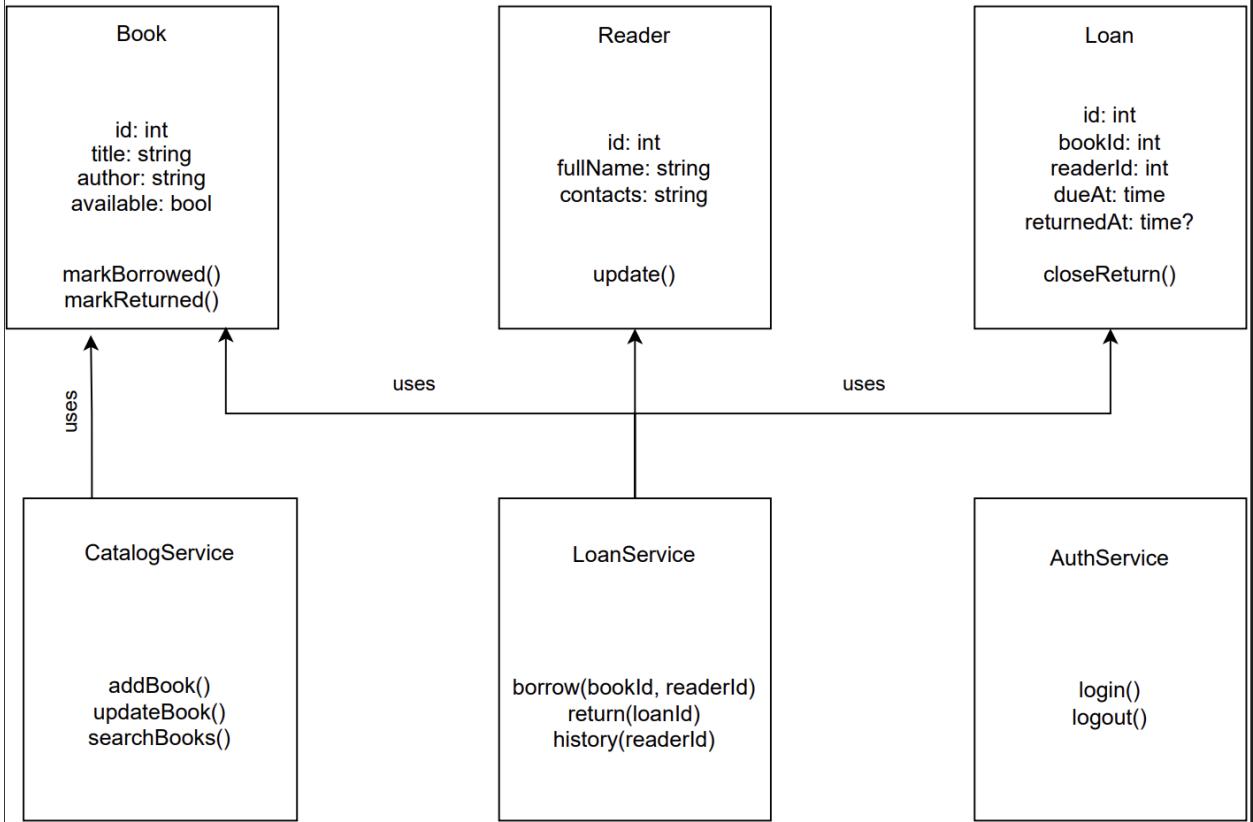
Use Case Diagram – App for a Librarian

App for a Librarian

Librarian

Login

Manage Books

Manage Readers

Borrow Book

Return Book

Search Books and Readers

View Loan History

ERD – App for a Librarian

Book

id (PK)
title
author
year
optional
available

User

id (PK)
name
email
password_hash
role

Reader

id (PK)
full_name
phone_or_email
created_at

Loan

id (PK)
reader_id (FK)
book_id (FK)
borrowed_at
due_at
returned_at
status

1 to many

1 to many

UML Class Diagram – App for a Librarian

Book

id: int
title: string
author: string
available: bool

markBorrowed()
markReturned()

Reader

id: int
fullName: string
contacts: string

update()

Loan

id: int
bookId: int
readerId: int
dueAt: time
returnedAt: time?

closeReturn()

uses   uses   uses

CatalogService

addBook()
updateBook()
searchBooks()

LoanService

borrow(bookId, readerId)
return(loanId)
history(readerId)

AuthService

login()
logout()

Project Plan: Librarian App

	Week 7	Week 8	Week 9	Week 10
Member A	**Project Infrastructure:** Git repository setup, folder structure, and Go environment configuration.	**Book Management:** Implementation of CRUD (Create, Read, Update, Delete) logic for book records.	**Transaction System:** Core logic for issuing books and tracking return deadlines.	**Module Integration:** Connecting the service layer with the repository and basic error handling
Member B	**Project Proposal:** Describe relevance, competitor analysis, target users, and core features	**Data Architecture:** Design and implementation of the ERD (Entity-Relationship Diagram).	**User Management:** Registration system for readers/students and search functionality.	**System Testing:** Unit testing of core functions and final documentation for defense.