LibMan App: Purpose, Features, and Technical Overview

Software Requirements Specification and Sample Design Document

Akinaw Mengesha Gebregiorgies

November 6 , 2024

1. Introduction

The LibMan app is an android-based library management tool designed to streamline library operations, enhance user experience, and improve efficiency for both librarians and patrons. This Software Requirements Specification (SRS) document details the purpose, core features, technical stack, and overall architecture of LibMan.

2. Purpose of the Application

LibMan aims to digitize and modernize library management, providing a centralized platform for both librarians and patrons.

Key objectives include: Efficient Library Management: LibMan consolidates essential library functions, reducing time spent on administrative tasks and increasing operational efficiency.

User-Friendly Patron Interaction: With an intuitive interface, patrons can independently browse resources, reserve books, and manage their borrowing history.

Data Security and Organization: Through secure user authentication and organized data handling, LibMan protects user information and ensures a safe, efficient library environment.

The app's features are designed to reflect the requirements of modern libraries, promoting streamlined resource management, effective user engagement, and a secure experience for all stakeholders.

3. Key Features of the Application

Key Feature 1 - User Authentication

Overview

User authentication is a fundamental feature in LibMan, securing access to various app functionalities and safeguarding user data. This feature is critical for both librarians and patrons, ensuring only authorized users can access sensitive information or manage library resources.

Functionality

The authentication system enables two core functions:

User Registration: New users can create accounts, which allow them to access features like book reservations, personalized recommendations, and viewing borrowing history.

Secure Login: Registered users can log in with password protection, ensuring credentials remain safe through secure storage methods.

Benefits for Users

Personalized Experience: Users can access their borrowing history, manage reservations, and receive personalized recommendations based on past activity.

Data Security: The system prioritizes data protection, providing a secure environment for user information.

Administrative Access for Librarians: Only librarians can access certain management features, such as adding, editing, or deleting book records and overseeing member data.

User authentication supports the app's foundation of security and organization, allowing users to interact confidently with library services.

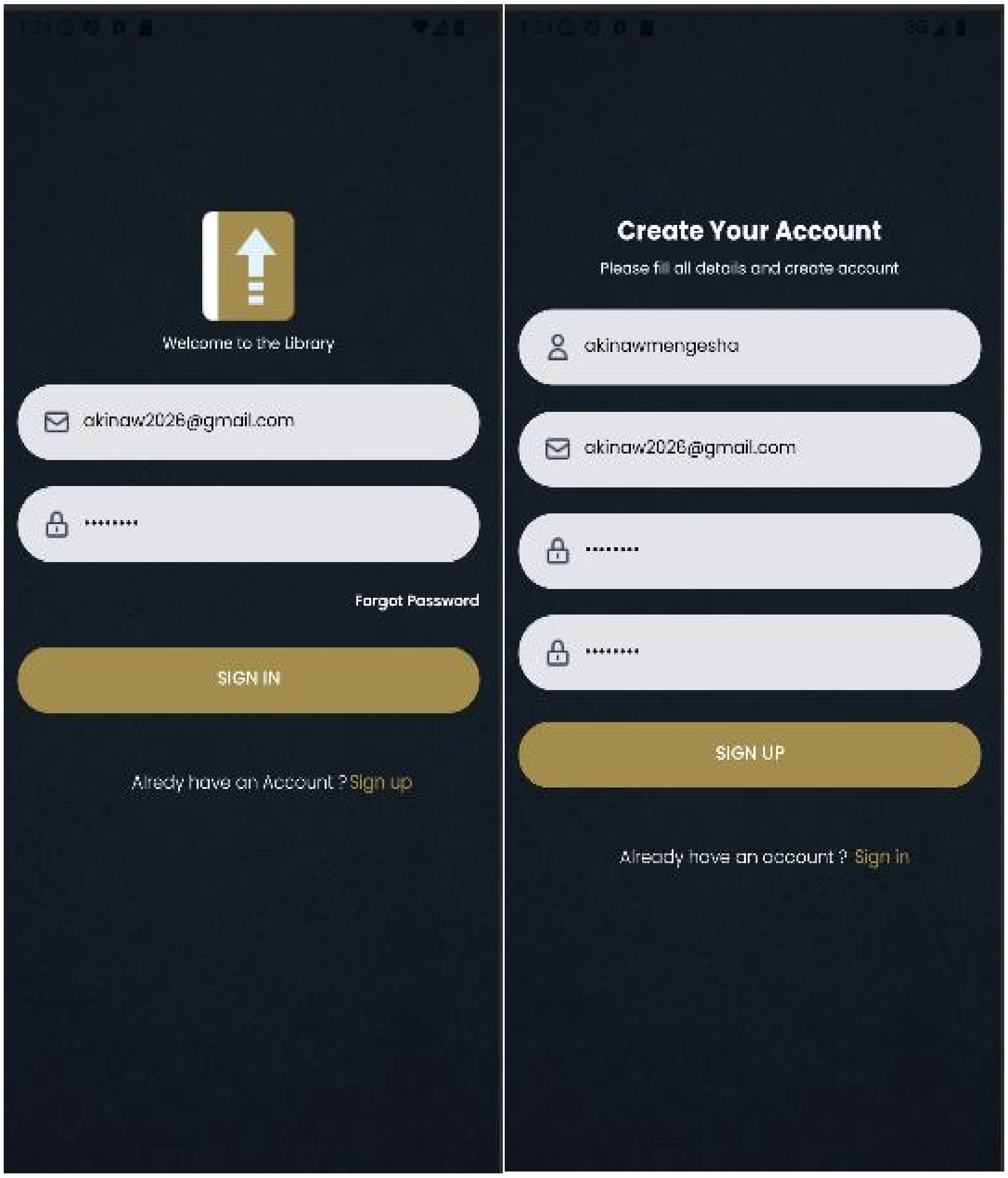


Figure 1 Sample Design of User Authentication

Key Feature 2 - Book Management

Overview

LibMan's Book Management feature is the core of the library's digital catalog, enabling librarians to systematically organize and maintain the library's book collection for easy accessibility and accurate tracking.

Functionality

The Book Management system offers several capabilities to ensure effective cataloging:

Adding New Books: Librarians can input comprehensive details for each book, including title, author, genre, publication year, ISBN, and availability status.

Updating Book Records: This function enables librarians to keep book information up to date, ensuring accuracy for users.

Viewing Book Details: Both patrons and librarians can view detailed information on any book in the catalog, making it simple to locate and learn about available resources.

Deleting Books: Outdated or damaged books can be removed from the catalog to maintain an accurate and relevant inventory.

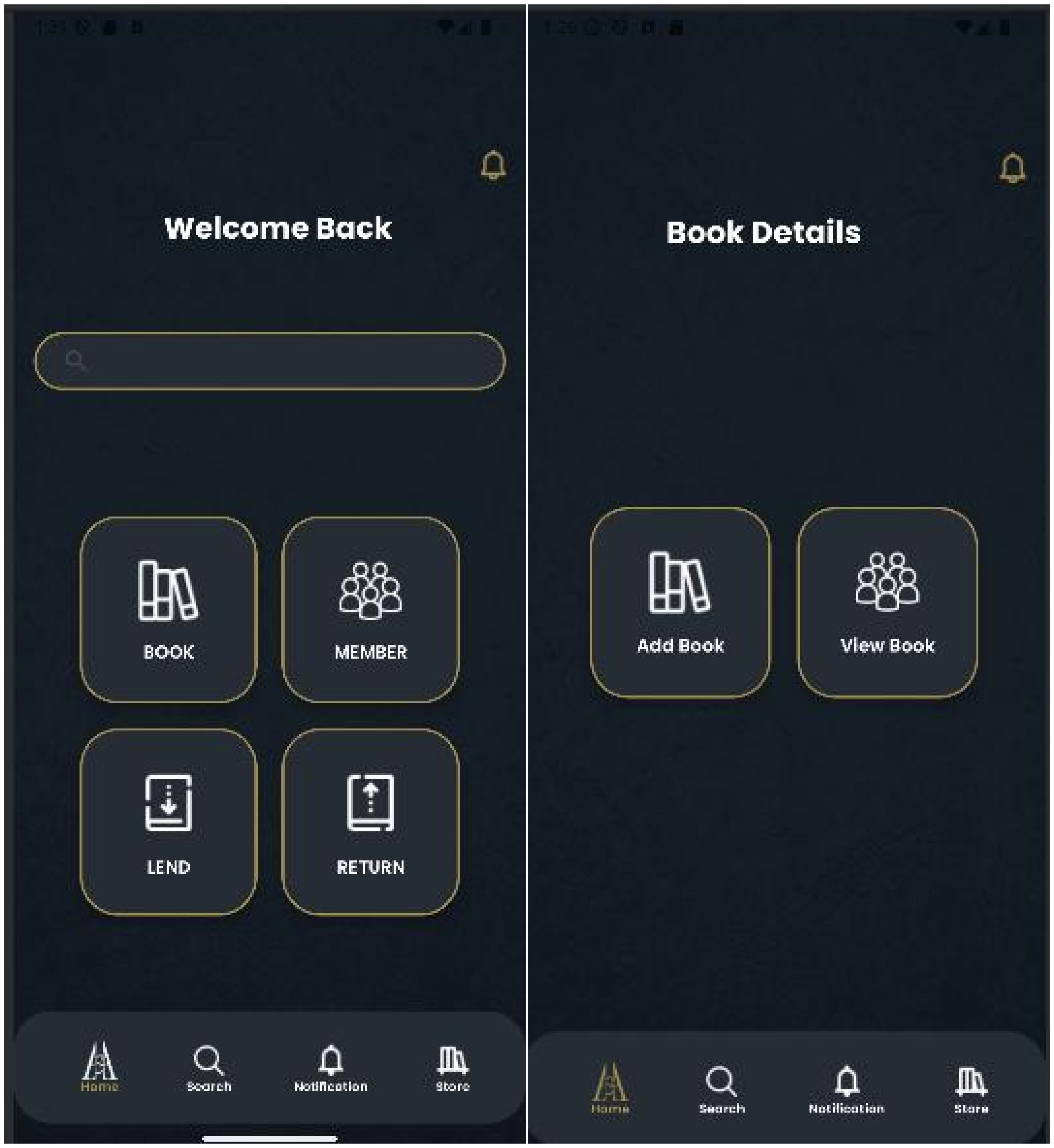
Benefits for Librarians

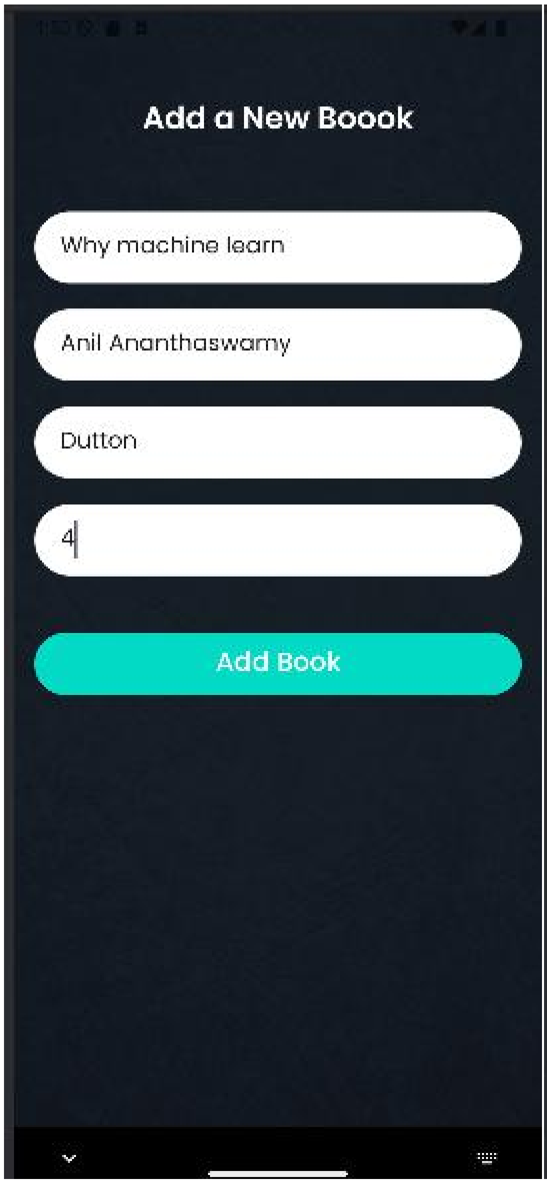
Centralized Inventory Control: The feature serves as a centralized database for all book data, making it easy for librarians to manage and locate resources.

Improved Access to Resources: By organizing books by genre, author, and availability, librarians can quickly find information, improving service efficiency.

Enhanced Patron Experience: Patrons benefit from accurate data about book availability, making it simple to find and reserve books as needed.

The Book Management feature is essential for maintaining an orderly and user-friendly catalog, enabling the smooth operation of daily library activities.





Add a New Boook

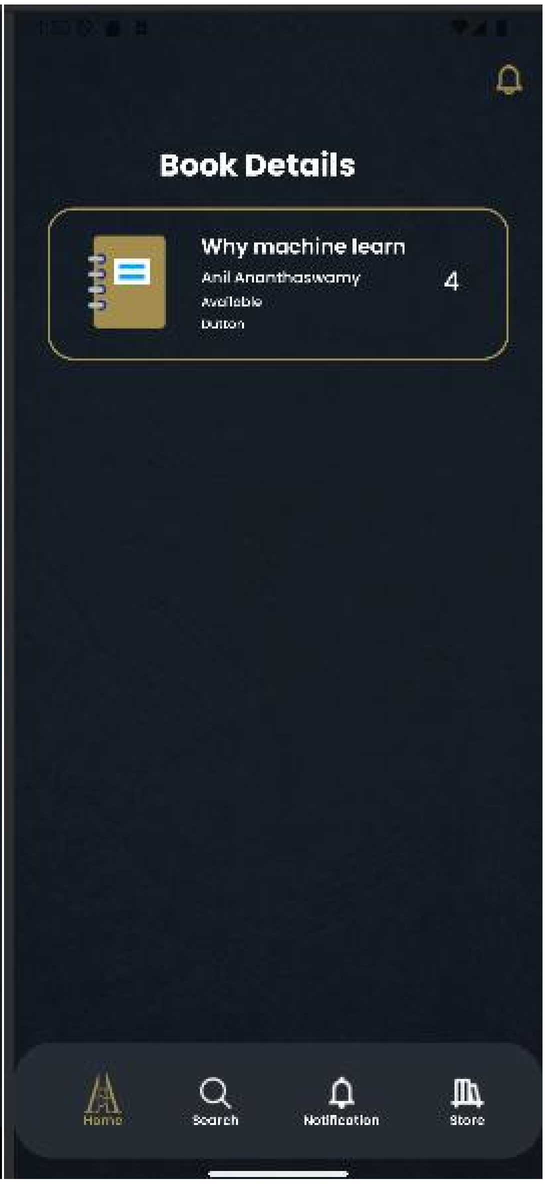
Why machine learn

Anil Ananthaswamy

Dutton

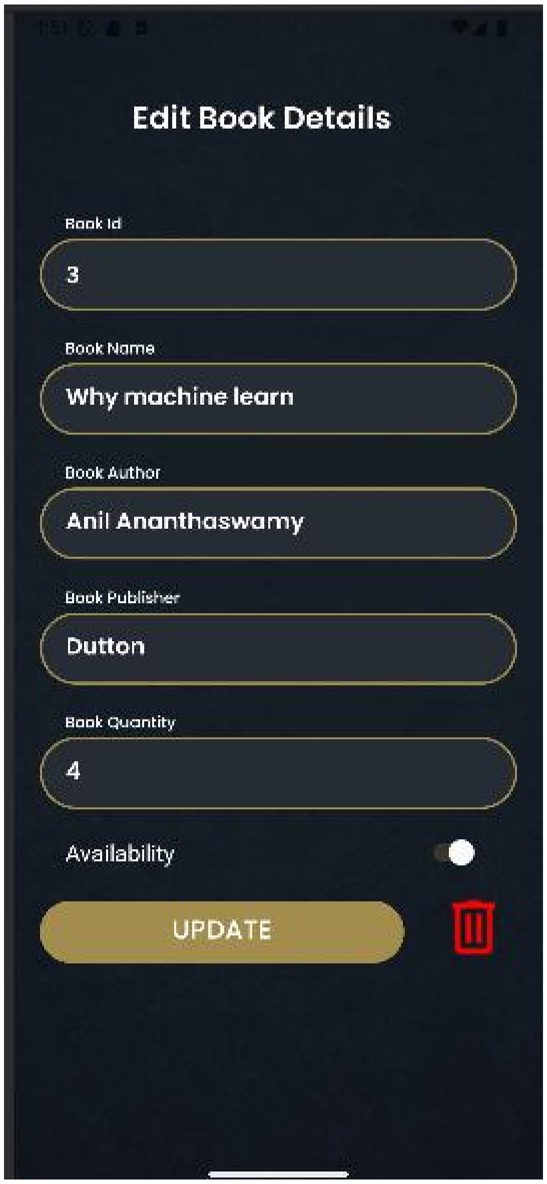
4

Add Book



Book Details

Why machine learn
Anil Ananthaswamy
Available
Dutton
4


Edit Book Details

Book Id: 3

Book Name: Why machine learn

Book Author: Anil Ananthaswamy

Book Publisher: Dutton

Book Quantity: 4

Figure 2 : Sample Design of User Management

Key Feature 3 - Member Management

Overview

Member Management is crucial for librarians to maintain a well-organized record of all patrons, including borrowing history, outstanding fines, and membership status, building a reliable database of users.

Functionality

This system allows librarians to:

Add New Members: Librarians can register new patrons, gathering key details like name, contact information, and membership start date.

Update Member Records: Member records can be easily updated to reflect any changes, ensuring all information remains current.

Track Borrowing History: Librarians can review each member's borrowing history, which helps manage outstanding loans and fines effectively.

Delete or Deactivate Memberships: For policy violations or unpaid fines, librarians can restrict access, helping to maintain accountability.

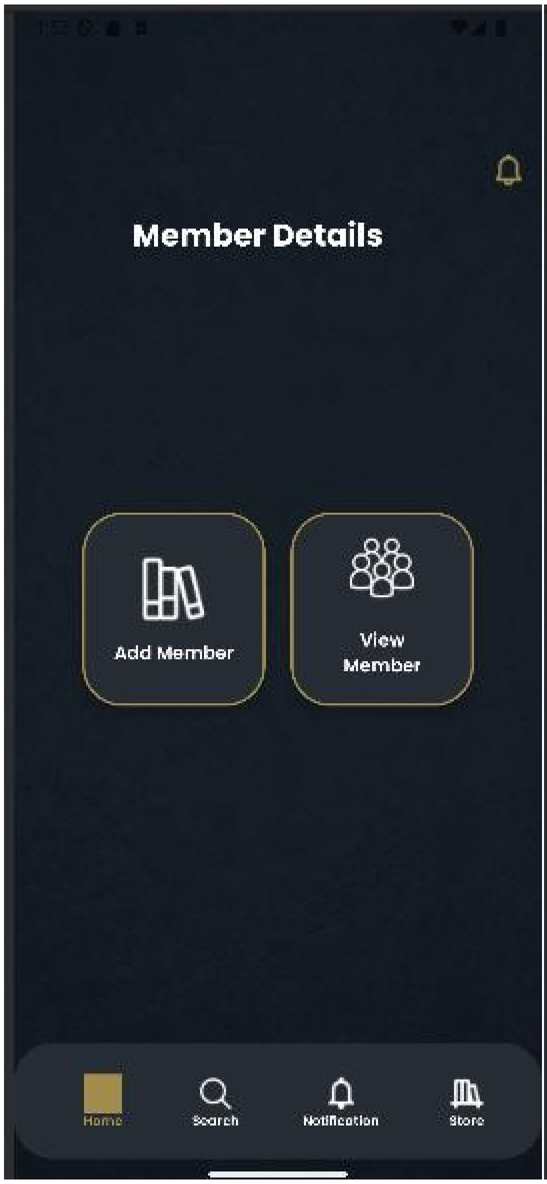
Benefits for Libraries and Patrons

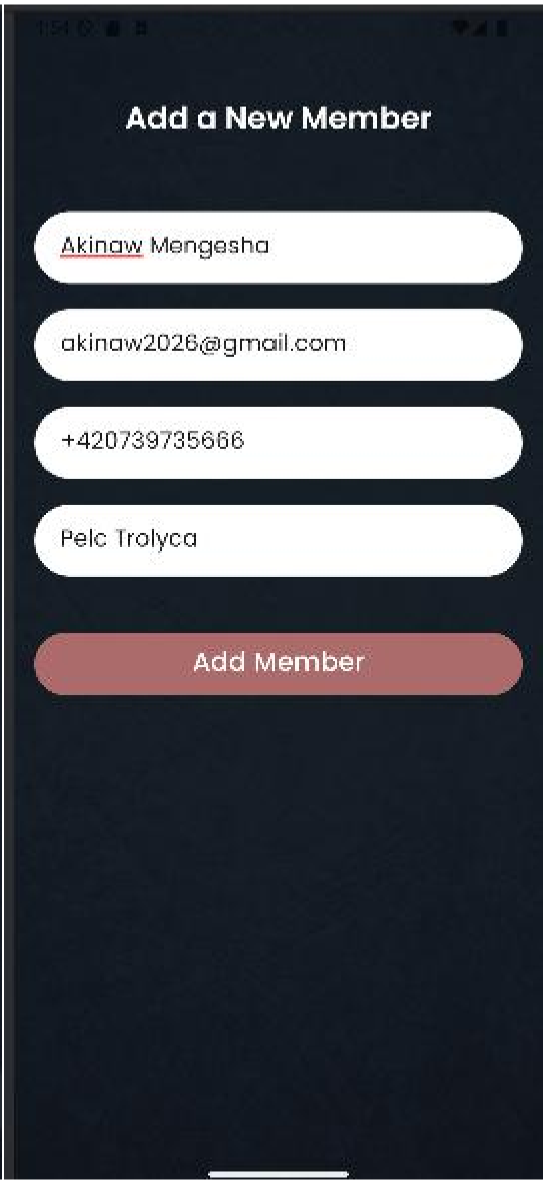
Efficient Record-Keeping: Librarians can maintain accurate records of all patrons, supporting efficient member communication and engagement.

Improved Accountability: By tracking borrowing histories and fines, librarians enforce library policies more effectively.

Enhanced Member Experience: Members can enjoy easy access to their borrowing records, making interactions with library staff smoother.

LibMan's Member Management feature allows for streamlined library administration, enabling librarians to dedicate more time to engaging with patrons and enhancing library services.



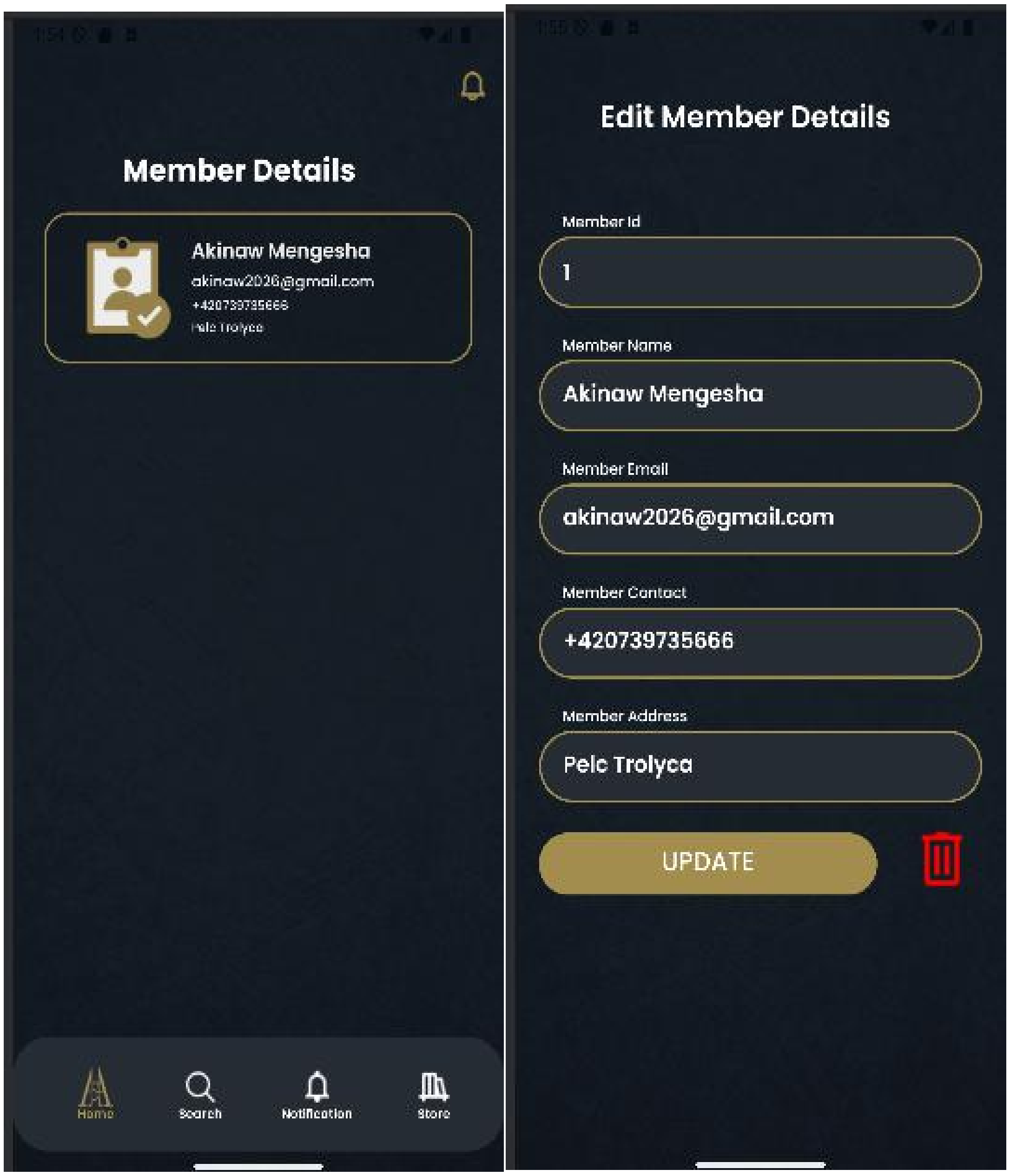

Add a New Member

Akinaw Mengesha

akinaw2026@gmail.com

+420739735666

Pelc Trolyca


Member Details

Akinaw Mengesha
akinaw2026@gmail.com
+420739735666
Pelc Trolyca

Edit Member Details

Member Id
1

Member Name
Akinaw Mengesha

Member Email
akinaw2026@gmail.com

Member Contact
+420739735666

Member Address
Pelc Trolyca

Figure 3 : Sample Design of Member Management

Key Features 4 and 5 - Book Reservation and Return Systems

Feature 4 - Book Reservation System

The Book Reservation feature allows patrons to reserve titles in advance, giving them assured access to popular or limited-availability books.

Functionality

The reservation system enables:

Placing Reservations: Patrons can view a book's current availability and place a reservation directly through the app.

Notifications for Patrons: When a reserved book is available, patrons are notified for collection.

Reservation Management for Librarians: Librarians can monitor active reservations, ensuring that reserved books are set aside for timely pick-up.

Benefits of Book Reservation

Increased Patron Satisfaction: Patrons can reserve books and avoid wait times for popular items.

Efficient Access to Resources: The reservation system prevents overbooking and allows the library to manage demand for high-traffic titles.

Feature 5 - Book Return System

The Book Return feature simplifies the return process, making it easy for librarians to manage inventory and track due dates.

Functionality

The return system includes:

Tracking Due Dates: The app automatically tracks each book’s due date, alerting librarians and patrons as return dates approach.

Sending Overdue Reminders: Patrons receive notifications for overdue items, encouraging timely returns.

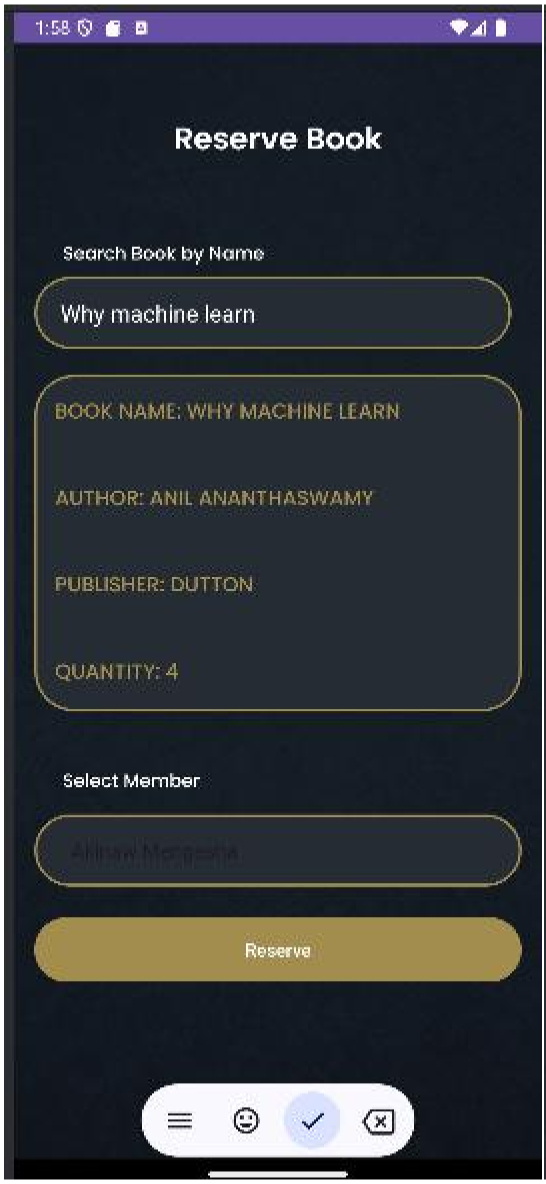
Updating Inventory upon Return: Librarians update book status in the system upon return, making the item available to other patrons.

Benefits of Book Return System

Inventory Control: The return system ensures that library records accurately reflect available stock.

Reduced Late Returns: Automated reminders help patrons return books on time, benefiting both the library and other patrons.

Together, the book reservation and return systems enable a complete, streamlined cycle for borrowing and returning resources, enhancing user experience and supporting efficient library management.



Reserve Book

Search Book by Name

Why machine learn

BOOK NAME: WHY MACHINE LEARN

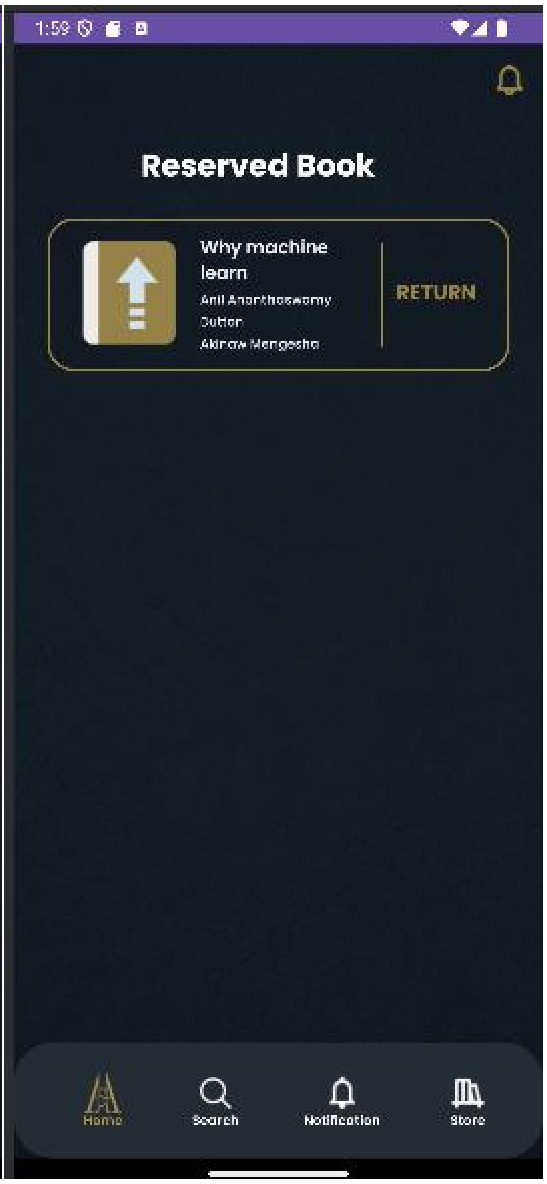
AUTHOR: ANIL ANANTHASWAMY

PUBLISHER: DUTTON

QUANTITY: 4

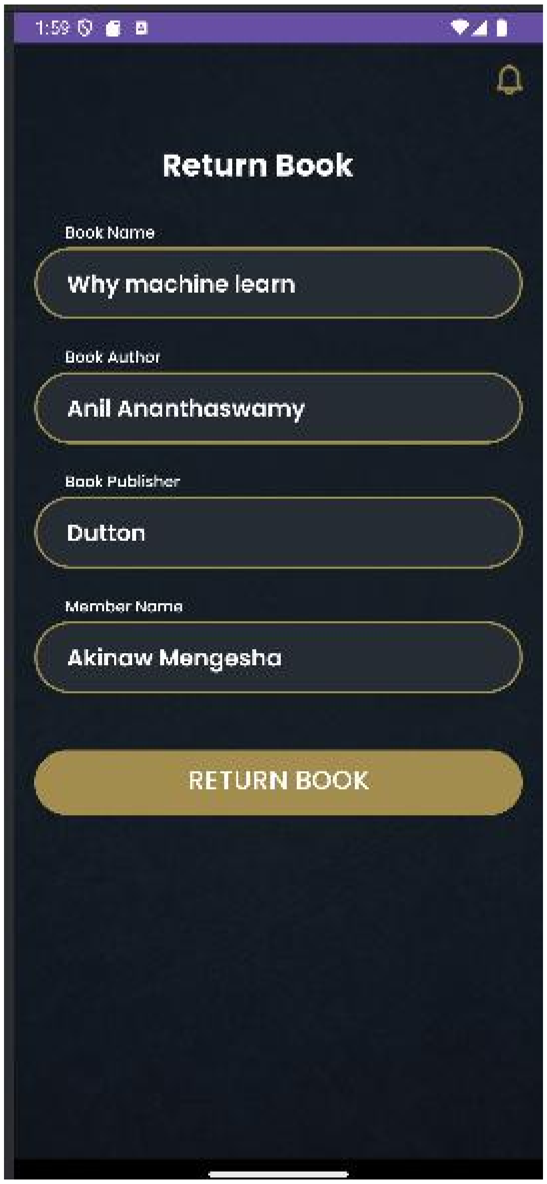
Select Member

Akinaw Mengesha



Reserved Book

Why machine learn
Anil Ananthaswamy
Dutton
Akinaw Mengesha


Return Book

Book Name
Why machine learn

Book Author
Anil Ananthaswamy

Book Publisher
Dutton

Member Name
Akinaw Mengesha

Figure 4 : Sample Design of Book Reservation and Return Systems

4. Technical Stack and Architecture

LibMan is a mobile application with a frontend and backend architecture, specifically designed to streamline library management operations for librarians and patrons. Android Studio, Java, and SQLite, LibMan supports a comprehensive range of features to manage library collections, track members, and facilitate book reservations and returns.

Frontend (Client-Side):

Android Studio: The primary Integrated Development Environment (IDE) for building the LibMan app, enabling Android-specific tools for layout design, testing, and debugging.

Java and XML:

Java: Serves as the primary programming language for the app, handling client-side logic and interactions. It powers the user interface (UI) responses, interactions with backend components, and error handling on the device.

XML: Used for defining UI layouts and components. Through XML, the app design adheres to Material Design guidelines for a consistent user experience.

Material Design Components: By following Material Design principles, LibMan achieves a professional and user-friendly interface that aligns with Android's UI/UX standards.

Backend (Data Management and Business Logic):

Java: While Java handles frontend interactions, it is also used for the backend logic within the app itself. Java code is responsible for data processing, business logic (such as handling book availability checks and reservations), and interfacing with the SQLite database.

SQLite Database: Used for local data storage, SQLite enables the app to operate offline, storing essential data like user accounts, book records, member details, and borrowing histories on the device itself.

Gradle and Kotlin Scripts (.kts):

Gradle Build System: The build automation tool Gradle is used to compile, test, and package the app. With Android-specific Gradle plugins, it supports the integration of libraries and external resources needed for development.

Kotlin Script (.kts): Kotlin scripting in Gradle (.kts) files helps define project configurations and dependencies in a concise, type-safe way. For instance, libraries required for UI design, data handling, and testing are declared in build.gradle.kts files.

Key Dependencies and Libraries:

Implementation Dependencies: The build.gradle.kts files specify libraries necessary for the application, such as: androidx libraries for enhanced UI components and layout structures.

Configuration and Build Types: build.gradle.kts files allow configurations for different environments (e.g., debug vs. release) and set up ProGuard for code obfuscation in production builds.

Project Structure and Build Workflow:

Module-Based Architecture: The app's core modules (e.g., app, data, and ui) are defined within the build.gradle.kts files, allowing for modularity and better management of dependencies.

Build Variants and Product Flavors: The Kotlin scripts (build.gradle.kts) enable the use of multiple build variants or product flavors if the app needs different versions (e.g., a library-specific version with custom branding).

Automated Tasks: Gradle automation tasks, defined within build.gradle.kts, streamline building, testing, and deploying processes. Common tasks include cleaning the build, assembling the APK, and running tests.

User Authentication and Security:

User Authentication: With Java handling user authentication, the app secures access to various functions, such as viewing account details and managing reservations. This involves user registration, login, and data access restrictions within the app.

Data Privacy and Integrity: SQLite databases are securely stored on the device, and Kotlin scripts manage encrypted libraries (if implemented) to enhance data protection.

Development and Maintenance

The use of Gradle with Kotlin DSL in build.gradle.kts files allows for easier configuration management, reducing errors and improving code readability. By leveraging the capabilities of both Java and Kotlin, LibMan can be scaled efficiently, with potential for remote capabilities, cloud storage, and enhanced data security features.

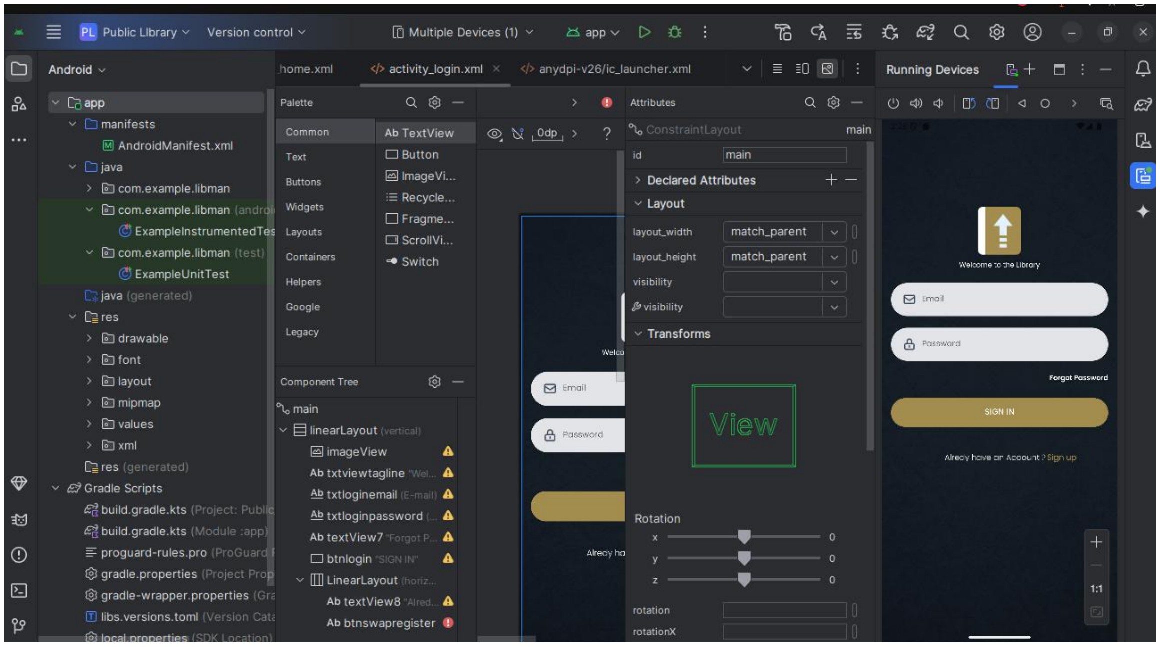


Figure 5 : Technical Stack and Architecture of LibMan

5. Conclusion

The LibMan app, designed as a mobile library management solution, seamlessly integrates frontend and backend architectures to support a range of essential library functions. Built using Android Studio, Java, and SQLite, the app enables effective book and member management, reservation systems, and more. On the frontend, Android Studio and Java ensure a responsive, user-friendly experience, with XML adhering to Material Design principles. The backend leverages Java and SQLite to manage data storage and business logic, enabling offline functionality while securely handling user and book information.

To facilitate development, LibMan relies on Gradle and Kotlin scripts, which streamline project configuration, dependency management, and automate tasks like testing and deployment. Key dependencies like AndroidX, are managed through these scripts, allowing for modular code organization and flexible build variants.

Security and user authentication are central to LibMan's architecture. Java handles secure user registration, login, and access restrictions, while Kotlin scripts support encrypted data storage and device-level security. Overall, the LibMan app is designed for scalability and potential enhancements, such as cloud storage, making it a versatile, reliable tool for modern libraries.

This SRS document outlines how LibMan is positioned to meet the dynamic needs of libraries, supporting effective resource management, secure data handling, and seamless interaction for users.