



COLLECTED WISDOM, EMPOWERED SOFTWARE: YOUR ESSENTIAL ARMORY

A digital Library platform project Proposal

Project Title : Developing Digital Library System (Offline-first Mobile App with Periodic Sync)

Client: በ ኢ/አ/ተ/ቤ አቶና ደብረ ጽዮን ቅ/ማርያም ቤ/ያን ፈለገ ሕይወት ሰ/ት/ቤ/ት ቤተ-መጻሕፍት ክፍል

Project lead developer Info:

Name: Natinael Samuel (Bsc in SoftwareEngineering)

Phone: +251904161978

Email: afritioalberts1216@gmail.com

Portfolio: <https://natinael-samuel.netlify.app>

12 September 2025

Wolaita sodd, Ethiopia

Prepared By Natinael Samuel Full-Stack Developer | Software Engineer

1) Project Goal

The goal of this project is to build a **low-cost, offline-first mobile app** for librarians that enables:

- Registering books and generating/printing QR labels.
 - Registering users using the existing **Fayda National ID** (16-digit ID, barcode).
 - Checking books in/out by scanning the user ID barcode + book QR (borrow) and scanning book QR (return).
 - Storing all activity locally and syncing to a central online database when Wi-Fi is available (semi-automatic, e.g., every X hours).
 - Producing simple reports (overdue, top books, top readers, counts by category).
-

2) Success Criteria

The project will be considered complete when:

- Librarians can register books and produce QR labels.
 - Librarians can register users with Fayda ID and identify users by scanning their Fayda barcode.
 - Borrow/return transactions work offline and sync reliably when internet is available.
 - Reports are accessible within the app.
 - Sync conflict handling prevents record corruption.
 - The app runs smoothly on **basic Android devices**.
-

3) Minimum Viable Product (MVP) Features

- **Book registration form** (title, author, category, notes, unique ID).
 - **QR generation** for books (save/share/export).
 - **User registration** by Fayda ID (16-digit ID, name, phone, optional photo).
 - **Borrow flow**: Scan Fayda barcode → scan book QR → save record.
 - **Return flow**: Scan book QR → mark as returned.
 - **Local storage**: SQLite database with robust backups.
 - **Sync system**: Semi-automatic (every X hours) + manual sync option.
 - **Reports**: Overdue books, top borrowed, counts by category.
 - **Search**: Local book search by title, author, category.
-

4) Phase 2 (Nice-to-Have Features)

- Advanced search + filters.
 - User management (blocking, fines, borrowing limits).
 - Notifications (SMS) for overdue books.
 - Multi-device conflict resolution with logs.
 - Role-based accounts for librarians.
 - Cloud-hosted dashboard (web) for church admin.
 - Report export to CSV/PDF.
-

5) Tech Stack (Practical, Low-Cost)

- **Mobile framework:** React Native (Expo).
 - **Local database:** SQLite (via react-native-sqlite-storage).
 - **QR handling:** Expo barcode scanner + QR generation library.
 - **Backend/Sync:** Supabase (Postgres + Auth + Storage).
 - **Hosting:** Supabase free tier or cheap VPS when needed.
-

6) Data Model (Simplified Schema)

Users: fayda_id, name, phone, photo, status.

Books: book_code, title, author, category, notes.

Transactions: tx_id, book_code, fayda_id, type (borrow/return), timestamp, sync_status.

SyncLog: sync_id, device_id, status, details.

7) Workflow Overview

- **Book QR:** Contains book_code, scanned for transactions.
 - **User Fayda ID:** Scanned to identify users.
 - **Borrow Flow:** Scan Fayda → scan Book → record saved.
 - **Return Flow:** Scan Book → mark as returned.
 - **Reports:** Generate overdue, top books, top readers.
-

8) Offline-First Sync Strategy

- Every action writes immediately to local SQLite.
- Sync triggers: automatic (on Wi-Fi every X hours) + manual “Sync Now” button.
- Data uploaded in **batch bundles** to Supabase.
- Conflict resolution: duplicate filtering & admin conflict list.

9) Core App Screens

1. Login / Shift start.
 2. Dashboard with key counts.
 3. Scan (borrow/return mode).
 4. Register User.
 5. Register Book + Generate QR.
 6. Books List & Search.
 7. Transactions History.
 8. Reports.
 9. Sync page.
 10. Settings.
-

10) Privacy & Security

- Store minimal user data.
 - Encrypt sensitive Fayda ID if possible.
 - HTTPS-only sync.
 - Role-based access control.
 - Limit photo retention.
-

11) Equipment & Costs

- Basic Android phones: Available by the librarian
 - QR label printing at local print shops: 20 Birr per sticker.
 - Hosting: Supabase free tier.
-

12) Implementation Plan

Phase 0 (Planning): Approvals & workflow confirmation.

Phase 1 (MVP: 2–4 weeks): Book & user registration, QR generation, borrow/return, local DB, manual sync.

Phase 2: Sync backend (Supabase), multi-device conflict handling, reports export.

Phase 3: Extra features (notifications, dashboard, advanced search).

13) Risks & Mitigation

- **Risk:** Forgetting to sync → **Mitigation:** auto-sync + reminders.
 - **Risk:** Poor barcode scanning → **Mitigation:** good QR printing + phone flashlight.
 - **Risk:** Data corruption during sync → **Mitigation:** batch uploads & validation.
-

14) Conclusion

This **Digital Library System** will modernize the church's library operations with a **low-cost, offline-first solution**, ensuring smooth daily operations, reliable data storage, and minimal equipment costs.