

Application de gestion de bibliothèque de quartier

Nom du projet 15 (variant) : DEVMOB-BookShare

Objectif

Créer une application mobile pour gérer une petite bibliothèque de quartier ou une bibliothèque associative : catalogue des livres, prêt / retour, réservations, suivi des emprunts, communication entre membres, et gestion des événements (heures de lecture, ateliers).

C'est quoi l'idée ?

Une application mobile pour les lecteurs et les bénévoles d'une bibliothèque locale afin de :

- consulter le catalogue,
- réserver et emprunter des livres,
- suivre l'historique et les retards (pénalités si souhaité),
- organiser des événements et ateliers,
- communiquer entre membres et bénévoles,
- partager des documents (règlement, fiches de prêt).

Qui utilise l'application ?

- **Tous les membres (usagers)** : consulter le catalogue, réserver, emprunter, noter un livre, participer aux événements, envoyer des messages.
- **Bénévoles / Bibliothécaires (admin)** : valider emprunts/retours, gérer le catalogue, ajouter/supprimer livres, gérer membres, modérer la messagerie, publier événements.
- **Visiteurs** : consulter le catalogue limité, s'inscrire en tant que membre.

Fonctionnalités attendues

Pour tous les membres

- Rechercher et parcourir le catalogue par titre/auteur/genre.
- Voir disponibilité (disponible / prêté / réservé).
- Réserver un livre (file d'attente si plusieurs réservations).
- Emprunter / retourner (scan code-barres ou QR pour faciliter).
- Voir son historique d'emprunts et dates de retour.
- Recevoir notifications (livre disponible, rappel date de retour, invitation à événement).
- Noter et laisser un court avis sur un livre.
- Participer au calendrier d'événements (lectures, ateliers).
- Messagerie interne pour contact entre membres et bénévoles.

- Stocker et consulter documents (règlement, formulaire d'adhésion).

Espace membre (utilisateur)

- Créer un compte / s'authentifier.
- S'inscrire comme membre de la bibliothèque (formulaire, justificatif optionnel).
- Voir et gérer ses réservations et prêts en cours.
- Prolonger un prêt (si autorisé).
- Signaler un livre endommagé.
- Consulter recommandations (basées sur genres favoris).

Espace admin (bibliothécaire / bénévole)

- Gérer le catalogue : ajouter/modifier/supprimer livres (titre, auteur, ISBN, code-barres, image, résumé, tags).
- Gérer les membres : validation, suspension, droits.
- Valider emprunts et retours (scan QR/code-barres + mise à jour automatique).
- Gérer les réservations et files d'attente.
- Publier et gérer événements (places limitées, inscriptions).
- Générer rapports : livres populaires, taux de rotation, membres actifs.
- Modérer messages et avis.

Aspects techniques

Flutter & Dart

- Flutter pour application multi-plateforme.
- Architecture : Clean Architecture ou MVC.
- Gestion d'état : Provider, Riverpod ou Bloc.
- UI responsive pour mobile (portrait/landscape) et tablette.

Backend

- Firebase (Auth + Firestore + Storage) ou Supabase pour rapidité de prototypage.
- **Optionnel : API REST (Node.js / Laravel / Spring Boot) si besoin de logique métier plus lourde.(optionnel)**
- **Utilisation possible d'un service local (SQLite) pour cache hors-ligne.(optionnel)**

Services additionnels

- Recherche plein-texte (Firestore + indexation ou Algolia en option).
- Scan de code-barres / QR (plugin Flutter).
- Notifications push (Firebase Cloud Messaging).
- Export / impression des reçus de prêt en PDF.

Comment ça marche techniquement ?

- **Auth** : Firebase Auth (email/mot de passe, Google) pour l'accès.
- **Catalogue & Données** : Firestore pour livres, membres, prêts, réservations, messages.

- **Stockage** : Firebase Storage pour images de couvertures et documents PDF.
- **Temps réel** : Firestore ou WebSocket pour notifications et messagerie en temps réel.
- **Scan** : Intégration d'un plugin pour scanner ISBN / QR afin d'enregistrer rapidement un prêt/retour.

UI/UX

- Design clair et accueillant, visuel orienté « bibliothèque » (couvertures, tags).
- Mode sombre / clair.
- Thème personnalisable (couleurs de la bibliothèque).
- Navigation avec BottomNavigationBar (Catalogue, Mes Emprunts, Événements, Messages, Profil).
- Écrans requis en maquette : accueil (recommandations + nouveautés), page livre (détails + actions), emprunts/reservations, calendrier événements, ajout livre (admin), messagerie.

Sécurité & rôles

- Authentification sécurisée via Firebase Auth.
- Rôles : visiteur / membre / bibliothécaire (admin).
- Permissions sur Firestore rules (ex : seuls admins peuvent modifier le catalogue).
- Validation et vérification des uploads de documents (type, taille).

Maquette visuelle (obligatoire)

- Fournir une maquette interactive (Figma / Adobe XD).
- Présenter au minimum ces écrans : accueil, fiche livre, emprunts, réservation, événements, ajouter livre (admin), messagerie.
- Bonus si le design est fidèlement implémenté dans l'application finale.

Structure du projet recommandée

lib/

```
├─ models/      # Livre, Membre, Emprunt, Reservation, Evenement, Message
├─ providers/    # authProvider, catalogueProvider, loansProvider
├─ services/     # AuthService, BookService, LoanService, EventService, FileService,
ScanService
```

views/

```
├─ home/        # Accueil, recommandations
├─ catalogue/   # Liste, filtre, fiche livre
├─ loans/       # Emprunts, réservations
├─ events/      # Calendrier, inscription
├─ admin/       # Gestion catalogue & membres
├─ chat/        # Messagerie
```

```
├─ widgets/     # couverture_card, book_item, dialogs, badges
```

└─ main.dart

Critères d'évaluation

Critère	Détail
Durée	1 mois minimum
Commits Git	Minimum 20, messages clairs
Fonctionnel	Recherche/catalogue, prêt/retour, réservations, messagerie, événements
Architecture	Clean Architecture ou MVC
Qualité du code	Lisible, organisé, modulable
Interface	Ergonomique, responsive, navigation fluide
Démo / APK	Application testable ou déployée (ex : Firebase App Distribution)
Tests (bonus)	Tests unitaires / widget tests

Suggestions bonus

- Intégration d'un scanner ISBN pour ajout automatique des métadonnées (via API publique).
- Gestion des pénalités et paiements (si bibliothèque le souhaite).
- Recommandations basées sur historique de lecture.
- Export des rapports (PDF/CSV) : livres les plus empruntés, statistiques membres.
- Intégration multi-langue (français / anglais).
- Fonction « club de lecture » : créer groupes et partager listes de lecture.
- Mode hors-ligne pour consultation du catalogue mis en cache.