

Cahier des Charges - Application de Gestion de Bibliothèque

DEVMOB-ApcPédagogie-16 - Application Mobile de Bibliothèque de Quartier

1. Fiche d'Identité du Projet

Élément	Détails
Nom du projet	DEVMOB-ApvPédagogie
Version	1.0
Type	Application Mobile Multiplateforme
Public cible	Bibliothèques associatives, de quartier
Durée de développement	1 mois minimum
Technologies principales	Flutter, Dart, Firebase
Architecture	MVC (Model-View-Controller)

2. Objectif Principal

Créer une application mobile complète pour gérer une bibliothèque locale permettant :

- La gestion numérique du catalogue et des prêts
 - La facilitation des échanges entre membres et bénévoles
 - L'organisation d'événements culturels
 - La digitalisation des processus d'emprunt
-

3. Personnes et Rôles Utilisateurs

a. Visiteur (Non authentifié)

- Consulte le catalogue limité
- S'inscrit comme membre
- Prend connaissance des événements

b. Membre (Usager authentifié)

- Consulte le catalogue complet
- Réserve et emprunte des livres
- Gère ses emprunts en cours
- Participe aux événements
- Communique avec la communauté

c. Bibliothécaire/Bénévole (Admin)

- Gère le catalogue complet
 - Valide les emprunts/retours
 - Administre les membres
-

- Organise les événements
 - Modère les contenus
-

4. Fonctionnalités Détaillées

a. Fonctionnalités Globales (Tous les utilisateurs)

1. Gestion du Catalogue

- Recherche avancée (titre, auteur, genre, ISBN)
- Filtrage par disponibilité, genre, popularité
- Affichage des couvertures et résumés
- Système de notation et d'avis
- Tags et catégorisation

2. Système d'Emprunt Intelligent

- Réservation avec file d'attente automatique
- Scan QR/Code-barres pour emprunt/retour
- Calcul automatique des dates de retour
- Notifications de rappel
- Prolongation d'emprunt (si autorisée)

3. Gestion des Événements

- Calendrier interactif des activités
- Inscription en ligne avec gestion des places
- Rappels automatiques
- Photos et comptes-rendus

4. Communication Communautaire

- Messagerie interne sécurisée
- Forum de discussion par genre littéraire
- Annonces importantes
- Système de notifications push

b. Espace Membre

Tableau de Bord Personnel

dart

// Modèle de données membre

```
class Membre {  
    String uid;  
    String nom;  
    String email;  
    String telephone;  
    DateTime dateAdhesion;  
    List<String> genresPrefere;  
    int nbEmpruntsEnCours;
```

```
StatutMembre statut; // Actif, Suspendu, En attente
}
```

Fonctionnalités Spécifiques

- Historique complet des emprunts
- Liste de souhaits (wishlist)
- Recommandations personnalisées
- Signalement de livres endommagés
- Gestion du profil et préférences

c. Espace Administrateur

Tableau de Bord Admin

- Métriques en temps réel (livres empruntés, membres actifs)
- Alertes (retards, réservations en attente)
- Rapports automatiques

Outils de Gestion

- **Catalogue** : CRUD complet, import/export
 - **Membres** : Validation inscriptions, suspension, statistiques
 - **Prêts** : Gestion manuelle, historique, statistiques
 - **Contenu** : Modération avis, messages
-

5. Architecture Technique

Stack Technologique

Couche	Technologie
Frontend	Flutter 3.x, Dart 3.x
Backend	Firebase (Firestore, Auth, Storage)
État	Provider/Riverpod
Base de données	Cloud Firestore
Authentification	Firebase Auth
Stockage	Firebase Storage
Notifications	Firebase Cloud Messaging

Structure du Projet

```
lib/
├── models/          # Entités métier
│   ├── livre.dart
│   ├── membre.dart
│   ├── emprunt.dart
│   ├── reservation.dart
│   ├── evenement.dart
│   └── message.dart
├── controllers/     # Logique métier
│   ├── auth_controller.dart
│   ├── livre_controller.dart
│   ├── emprunt_controller.dart
│   ├── evenement_controller.dart
│   └── message_controller.dart
├── views/           # Interfaces utilisateur
│   ├── auth/
│   ├── catalogue/
│   ├── emprunts/
│   └── evenements/
├── messengerie/
│   └── admin/
├── services/        # Services externes
│   ├── firestore_service.dart
│   ├── auth_service.dart
│   ├── storage_service.dart
│   ├── notification_service.dart
│   └── scan_service.dart
├── widgets/         # Composants réutilisables
│   ├── livre_card.dart
│   ├── emprunt_item.dart
│   ├── custom_appbar.dart
│   └── dialogs/
└── utils/           # Utilities
    ├── constants.dart
    ├── validators.dart
    └── helpers.dart
```

6. Design et Expérience Utilisateur

Chartre Graphique

- **Style** : Moderne et chaleureux, inspiré "bibliothèque"
- **Couleurs** : Palette bleu/jaune/orange (couleurs littéraires)
- **Typographie** : Lisible (Roboto, Merriweather)
- **Mode** : Sombre/Clair automatique

Navigation Principale

```
dart
BottomNavigationBar(
  items: [
    BottomNavigationBarItem(icon: Icon(Icons.home), label: 'Accueil'),
    BottomNavigationBarItem(icon: Icon(Icons.menu_book), label: 'Catalogue'),
    BottomNavigationBarItem(icon: Icon(Icons.event), label: 'Événements'),
    BottomNavigationBarItem(icon: Icon(Icons.chat), label: 'Messages'),
    BottomNavigationBarItem(icon: Icon(Icons.person), label: 'Profil'),
  ],
)
```

Écrans Obligatoires (Maquette Figma)

1. **Écran d'accueil** - Recommandations + nouveautés
2. **Catalogue** - Liste/filtre/recherche
3. **Fiche livre** - Détails + actions
4. **Mes emprunts** - État et historique

5. **Calendrier événements** - Inscriptions
 6. **Messagerie** - Conversations
 7. **Admin** - Tableau de bord gestion
-

7. Sécurité et Permissions

Règles Firestore

```
javascript
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    // Membres peuvent lire le catalogue
    match /livres/{document} {
      allow read: if request.auth != null;
      allow write: if hasRole('admin');
    }

    // Chaque membre voit seulement ses emprunts
    match /emprunts/{document} {
      allow read, write: if request.auth != null
        && request.auth.uid == resource.data.membreId;
      allow read, write: if hasRole('admin');
    }

    function hasRole(role) {
      return request.auth != null
        &&
get(/databases/{database}/documents/membres/{request.auth.uid}).data.role ==
role;
    }
  }
}
```

8. Matrice des Permissions

Action	Visiteur	Membre	Admin
Voir catalogue limité	✓	✗	✗
Voir catalogue complet	✗	✓	✓
Emprunter	✗	✓	✓
Gérer catalogue	✗	✗	✓
Modérer contenu	✗	✗	✓

9. Fonctionnalités Avancées

Services d'Intégration

- **Scan ISBN** : Plugin barcode_scan pour ajout automatique
- **Notifications** : Firebase Cloud Messaging
- **Recherche** : Algolia (optionnel) pour recherche plein texte
- **Hors-ligne** : Hive/SQLite pour cache local

Fonctionnalités Bonus

- **Club de lecture** : Groupes de discussion
- **Recommandations IA** : Basé sur historique
- **Export PDF** : Reçus, rapports statistiques
- **Multi-langue** : Français/Anglais (intl)

10. Critères d'Évaluation

Obligatoires

Critère	Détails	Pondération
Fonctionnel	Catalogue, prêt/retour, réservations, messagerie	40%
Architecture	MVC bien implémenté, séparation claire	20%
Code qualité	Lisible, documenté, modulaire	15%
UI/UX	Ergonomique, responsive, fidèle maquette	15%
Git	20+ commits, messages clairs	10%

Bonus

- Tests unitaires/widget (10% bonus)
- Déploiement effective (Firebase App Distribution)
- Performance optimisée
- Accessibilité respectée

11. Planning de Développement

Semaine 1-2 : Fondations

- Setup projet Flutter + Firebase
- Architecture MVC et modèles de données
- Authentification et profils
- Catalogue livres (CRUD basique)

Semaine 3-4 : Fonctionnalités cœur

- Système emprunt/réservation
- Interface membre complète
- Recherche et filtres
- Notifications de base

Semaine 5-6 : Avancé

- Espace admin
- Messagerie et événements
- Scan code-barres
- Optimisations et tests

Semaine 7 : Finalisation

- Corrections bugs
 - Tests utilisateurs
 - Documentation
 - Déploiement
-

12. Mesures de Succès

Indicateurs Techniques

- Temps de chargement < 2s
- Taux de crash < 1%
- Couverture code > 70% (bonus)

Indicateurs Utilisateurs

- Réduction temps gestion bénévoles > 30%
 - Augmentation emprunts > 20%
 - Satisfaction utilisateurs > 4/5 étoiles
-

13. Livrables Attendus

1. **Code source** complet sur GitHub/GitLab
 2. **APK** testable ou lien Firebase App Distribution
 3. **Maquettes Figma/XD** interactives
 4. **Documentation** technique et utilisateur
-

5. **Présentation** démo (5-10 minutes)

Ce cahier des charges sert de référence contractuelle pour le développement de l'application DEVMOB-ApcPédagogie. Toute modification devra être documentée et validée par l'ensemble des parties prenantes.