

Cahier des charges

Application mobile **Biblio Fac**

1. Contexte et objectifs du projet

Dans le cadre du cours de Programmation Mobile (L4 LMD), ce projet constitue l'examen final. Il est réalisé par groupe de cinq étudiants et consiste à concevoir, développer et défendre une application mobile complète, fonctionnelle et bien architecturée.

Le sujet retenu est le **Sujet 2 : Gestion de Bibliothèque Universitaire**. L'application, nommée **Biblio Fac**, vise à digitaliser la gestion d'une bibliothèque universitaire afin de faciliter l'accès aux livres, le suivi des emprunts et la gestion administrative.

2. Présentation générale de l'application

Nom de l'application : Biblio Fac

Plateforme : Mobile (Android)

Technologie principale : Flutter

Backend : Firebase (Authentication et Cloud Firestore)

L'application permet aux étudiants de consulter un catalogue de livres, vérifier leur disponibilité, effectuer des réservations et suivre leur historique d'emprunts. Elle offre également une interface d'administration destinée aux bibliothécaires.

3. Rôles et types d'utilisateurs

3.1 Étudiant

Les fonctionnalités accessibles à l'étudiant sont :

- Inscription et connexion à l'application
- Consultation du catalogue de livres
- Recherche de livres par titre, auteur ou ISBN
- Vérification de la disponibilité des livres
- Réservation et emprunt de livres
- Consultation de l'historique personnel des emprunts

3.2 Administrateur (Bibliothécaire)

Les fonctionnalités accessibles à l'administrateur sont :

- Gestion complète du catalogue de livres (ajout, modification, suppression)
- Validation ou refus des emprunts et réservations
- Consultation des emprunts en cours
- Supervision globale de la bibliothèque

4. Fonctionnalités principales

4.1 Authentification et gestion des utilisateurs

L'application utilise Firebase Authentication avec les méthodes suivantes :

- Inscription par email et mot de passe
- Connexion par email et mot de passe
- Connexion via Google (provider externe)
- Persistance de la session utilisateur
- Déconnexion sécurisée

4.2 Gestion du catalogue de livres

- Affichage de la liste des livres disponibles
- Ajout de nouveaux livres par l'administrateur
- Mise à jour des informations d'un livre
- Suppression d'un livre
- Gestion du nombre total et du nombre d'exemplaires disponibles

4.3 Emprunts et réservations

- Emprunt d'un livre disponible
- Réservation d'un livre indisponible
- Suivi du statut des emprunts (en cours, retourné)
- Historique des emprunts par utilisateur

4.4 Intégration d'une API externe

Afin d'enrichir le catalogue, l'application consomme la **Google Books API**. Cette API permet :

- La recherche de livres par titre, auteur ou ISBN
- L'import automatique des informations (auteur, description, couverture)

5. Architecture logicielle

L'application respecte strictement le pattern **MVC (Modèle – Vue – Contrôleur)**.

5.1 Organisation du projet

```
lib/  
  models/  
  views/  
  controllers/  
  services/  
  providers/  
  main.dart
```

5.2 Description des couches

- **Models** : Représentation des entités (User, Book, Loan)
- **Views** : Interfaces utilisateur et écrans
- **Controllers** : Logique métier et coordination
- **Services** : Accès Firebase et API externes
- **Providers** : Gestion de l'état de l'application

6. Gestion de l'état

La gestion de l'état est assurée par le package **Provider**. Cette approche permet de :

- Séparer la logique métier de l'interface utilisateur
- Améliorer la maintenabilité du code
- Faciliter la mise à jour dynamique des données

7. Base de données Cloud Firestore

7.1 Collections principales

- **users** : informations des utilisateurs
- **books** : catalogue de livres
- **loans** : emprunts
- **reservations** : réservations

7.2 Description des données

Chaque collection est structurée en documents identifiés par un identifiant unique. Les relations entre utilisateurs, livres et emprunts sont gérées via des références d'identifiants.

8. Navigation et expérience utilisateur

L'application offre une navigation fluide avec :

- Indicateurs de chargement
- Messages de confirmation et d'erreur
- Boîtes de dialogue pour les actions critiques

9. Outils de collaboration

- Dépôt Git commun
- Historique des commits utilisé pour évaluer la contribution individuelle

10. Livrables attendus

- Lien vers le dépôt Git
- Application mobile fonctionnelle
- Documentation technique (README)

11. Conclusion

Le projet **Biblio Fac** vise à proposer une solution mobile moderne, structurée et conforme aux exigences pédagogiques. L'architecture MVC, l'intégration de Firebase et l'utilisation d'une API externe garantissent une application robuste, évolutive et adaptée au contexte universitaire.