# Documentation Technique du Projet Flutter : Gestion de Bibliothèque Numérique

## Introduction

Ce projet Flutter est une application de gestion de bibliothèque numérique qui suit l'architecture MVVM (Model-View-ViewModel). L'application permet de gérer les auteurs, les livres, et les utilisateurs avec des opérations CRUD (Création, Lecture, Mise à jour, Suppression). Elle inclut une gestion des rôles utilisateur (admin et user) pour contrôler les fonctionnalités accessibles.

## Architecture MVVM

### 1. Modèle (« Model »)

Représente les entités principales de l'application. Il s'agit des classes de données pour les auteurs, les livres et les utilisateurs.

- Fichiers principaux :

- Auteur.dart

- Livre.dart

- User.dart

### 2. Repository

Le repository gère les interactions avec la base de données SQLite via Sqflite. Chaque entité (« Model ») dispose d'un fichier pour encapsuler les opérations liées à la persistance.

- Fichiers principaux :

- Database.dart : Configuration de la base de données et création des tables.

- AuteurDatabase.dart : Gestion des auteurs dans la base de données.

- LivreDatabase.dart : Gestion des livres dans la base de données.

- UserDatabase.dart : Gestion des utilisateurs dans la base de données.

### 3. ViewModel

Les ViewModels gèrent la logique métier et servent d'intermédiaire entre les modèles et les vues. Ils utilisent ChangeNotifier pour informer les vues des changements d'état.

- Fichiers principaux :

- AuteurViewModel.dart

- LivreViewModel.dart

- UserViewModel.dart

### 4. Vue (« View »)

Les vues sont responsables de l'interface utilisateur. Elles affichent les données provenant des ViewModels et envoient les actions utilisateur aux ViewModels.

- Fichiers principaux :

- AjouterAuteurView.dart

- ModifierAuteurView.dart

- AuteurListView.dart

- AjouterLivreView.dart

- LoginView.dart

- HomeScreen.dart

### Fonctionnalités

1. Gestion des Auteurs

- Ajouter, modifier, supprimer et afficher les auteurs.

- Afficher les livres associés à un auteur.

2. Gestion des Livres

- Ajouter, modifier, supprimer et afficher les livres.

- Associer chaque livre à un auteur.

- Charger des jaquettes de livres à partir de fichiers locaux.

3. Gestion des Utilisateurs

- Créer des utilisateurs avec des rôles (admin ou user).

- Login avec hachage de mot de passe (bcrypt).

4. Gestion des Rôles

- Les administrateurs peuvent gérer tous les utilisateurs.

- Les utilisateurs simples ont un accès restreint.

Technologies Utilisées

- Flutter : Framework principal pour l'interface utilisateur.

- Sqflite : Bibliothèque SQLite pour la persistance des données.

- Provider : Gestion de l'état avec ChangeNotifier.

- BCrypt : Hachage des mots de passe.

### Conclusion

Ce projet met en œuvre une architecture MVVM solide, tout en exploitant les meilleures pratiques Flutter. Il est évolutif et offre une base robuste pour ajouter de nouvelles fonctionnalités.