

Projet Flutter – Bibliotheque

Célia Baylet

-Android Studio

-Langage : Dart

Contexte

La bibliothèque municipale de la ville souhaite moderniser la gestion de son catalogue en proposant une application numérique. L'objectif est de permettre aux bibliothécaires de gérer facilement les livres disponibles, d'ajouter de nouveaux ouvrages, et de consulter les détails des auteurs. L'application est destinée à un usage interne, mais une version future pourrait être déployée pour les lecteurs. L'interface doit être simple, réactive, et permettre une gestion rapide des informations.

Architecture de l'application : MVVM (Modèle-Vue-ViewModel)

L'application va être développée en **Flutter** en adoptant une architecture **MVVM** pour garantir une bonne séparation des préoccupations, une évolutivité future, et faciliter la maintenance.

Modèle

Le **Modèle** contient la logique métier et les structures de données. Il représente la couche d'accès aux données via SQLite. Par exemple, les classes Livre et Auteur définissent les entités Livre et Auteur, et les fichiers LivreDatabase et AuteurDatabase gèrent les interactions avec la base de données.

Exemple :

- **Livre** : Une classe représentant un livre avec des propriétés telles que idLivre, nomLivre, idAuteur, etc.
- **LivreDatabase** : Un gestionnaire de base de données qui permet d'insérer, supprimer, mettre à jour, et récupérer des livres.

Vue

La **Vue** représente l'interface utilisateur. Ce sont les différentes pages qui affichent les données et permettent aux utilisateurs d'interagir avec l'application. Les vues ne contiennent aucune logique métier, elles se contentent de réagir aux changements de l'état des données.

Exemple :

- **LivreVue** : Affiche la liste des livres disponibles avec des options pour ajouter, modifier, ou supprimer un livre.
- **AjouterLivreVue** : Permet à l'utilisateur d'ajouter un nouveau livre.

VueModèle

Le **VueModèle** sert de médiateur entre la Vue et le Modèle. Il contient la logique de présentation et est responsable de la gestion de l'état. Il récupère les données du Modèle et les prépare pour l'affichage dans la Vue. Il informe également la Vue lorsqu'il y a des changements dans les données.

Exemple :

- **LivreVueModèle** : Gère la récupération des livres depuis la base de données, fournit une interface propre à la Vue, et notifie les modifications.

Fonctionnalités :

Gestion des livres : Ajouter, modifier, et supprimer des livres du catalogue.

Gestion des auteurs : Lier un livre à un auteur et gérer les informations sur les auteurs.

Affichage des détails : Consulter les détails d'un livre ou d'un auteur.

Gestion des utilisateurs :

-rôles (restreindre les actions réservées aux administrateurs)

-Ecran de connexion pour l'authentification

-écran pour afficher et gérer la liste des utilisateurs.

-formulaire pour ajouter ou modifier les utilisateurs.

