



SAÉ s2.01

Description
écrite de
l'architecture



ANTOINE PEREDERII, TITOUAN LOUVET
GROUPE 9-10 PREMIERE ANNEE

Description écrite de l'architecture

L'architecture de notre application mobile Android Banquale, développée avec .NET MAUI, peut être décrite de la manière suivante :

I. Patrons de conception

A. Modèle-Vue-ViewModel (MVVM)

Le patron MVVM est utilisé pour séparer la logique métier de l'interface utilisateur. Il favorise la modularité et facilite les tests unitaires.

B. Injection de dépendances

L'injection de dépendances est utilisée pour gérer les dépendances entre les différentes classes de l'application. Cela permet de rendre le code plus souple, maintenable et testable.

II. Dépendances et structure du projet

Notre projet est hébergé sur Git (codefirst) à l'adresse suivante : <https://codefirst.iut.uca.fr/git/antoine.perederii/Banquale/> . Le projet utilise .NET MAUI, une technologie multiplateforme pour le développement d'applications mobiles. Les dépendances spécifiques à .NET MAUI sont utilisées pour la création de l'interface utilisateur, la gestion des ressources et la navigation entre les écrans. Le projet est organisé en différents dossiers et projets selon une structure de projet standard pour une application .NET MAUI. Nous avons trois projets principaux, l'un avec les views, le data binding et le fonctionnement. Le second avec les modèles, classes de notre projet et enfin le dernier avec les différents tests de notre application. Nous utilisons les fonctionnalités de Git pour la gestion du code source, le suivi des modifications et la collaboration au sein du binôme.

III. Composants clés de l'application

A. Modèles (Models)

Ils représentent les entités métier de notre application (classes), comme les comptes bancaires, les transactions, les clients, etc. Ils sont généralement annotés avec des attributs de sérialisation pour faciliter la persistance des données.

B. Vues (Views)

Elles sont responsables de l'affichage de l'interface utilisateur et de la collecte des entrées de l'utilisateur. Les vues sont généralement définies en utilisant des fichiers XAML pour permettre une conception déclarative de l'interface.

C. Vue-Modèles (ViewModels)

Ils contiennent la logique métier de l'application et se connectent aux vues via la liaison de données. Les vue-modèles sont responsables de la récupération et de la manipulation des données, ainsi que de la gestion des interactions utilisateur. Ce sont les fichiers C# associés à nos vues.

D. Gestionnaire de persistance (Data contract persistence)

Il gère la sauvegarde et le chargement des données de l'application, en utilisant une interface définie (IPersistenceManager). Cela permet de séparer la logique de persistance spécifique de l'implémentation réelle, ce qui facilite les tests et les modifications ultérieures.

Cette description donne un aperçu général de l'architecture de notre application mobile Banquale. Pour plus de détails et une compréhension plus approfondie de l'application, il est recommandé d'examiner le code source de notre projet sur [codefirst](#) dans le répertoire src/ ainsi que la documentation dans le répertoire /Documentation.