

**PRÉPARÉ PAR**

KHOUY Abderrazak  
ZENNOU Abdessalam  
KHANNOUS Anass  
ELMAKHFI Mohamed

**ENCADRÉE PAR**

ING . OUARRACHI

**PRÉSENTÉ LE**

JANVIER 2024

---

# DESCRIPTION PROJET MAUI

---

JAN 2024



.net MAUI



# Architecture utilisée : MVC

## Modèle (Model) :

Les modèles représentent les données et la logique métier de l'application.

Dans notre cas, les modèles sont définis dans le dossier Models et comprennent des entités telles que Etudiant, Filiere, Lesson, Absence, et Proff.

## Vue(view) :

Les vues sont responsables de l'interface utilisateur. Elles sont définies dans les fichiers XAML situés dans le dossier Views. Chaque page de notre application MAUI, comme MainPage.xaml, Inscription.xaml, MarkAbsence.xaml, PageInformation.xaml, et Search.xaml, a une vue correspondante. Ces vues définissent la structure et l'apparence de l'interface utilisateur.

## Contrôleur (Code-Behind) :

agit comme un intermédiaire entre la vue et le modèle, gérant la logique d'affichage, les interactions utilisateur, et transmettant les commandes au modèle.

Les fichiers code-behind, tels que MainPage.xaml.cs, Inscription.xaml.cs, MarkAbsence.xaml.cs, PageInformation.xaml.cs, et Search.xaml.cs, agissent en tant que contrôleurs.

Les contrôleurs gèrent les événements déclenchés par l'interface utilisateur et effectuent des actions correspondantes, comme la gestion des inscriptions, la recherche d'étudiants, la saisie des absences, etc.

## Entity framework and SQLite

utilise Notre application Entity Framework en combinaison avec SQLite pour la gestion de la base de données. Grâce à Entity Framework, la manipulation des données et les opérations sur la base de données, telles que la création de tables et l'accès aux données, sont simplifiées. Cette approche améliore l'efficacité du développement et offre une intégration transparente avec la base de données SQLite, facilitant ainsi la persistance et la récupération des informations métier au sein de notre application.