

# ShopEase - Une application Mobile Android de commerce électronique

Naif ASSWIEL

Amine KESSEMTINI

16/04/2025

## Introduction

Dans le cadre du module **DevOps - 178UD10** en **M1 Informatique - Intelligence Artificielle** à l'Université du Mans, nous vous présentons dans ce document **ShopEase**, une application de commerce électronique Android développée avec Flutter.

Il s'agit d'une application similaire à celles que vous rencontrez dans votre vie quotidienne, comme Amazon ou Fnac, mais enrichie de quelques fonctionnalités innovantes, telles que l'ajout d'articles côté vendeur via un code-barres, la recherche d'articles par commande vocale, ou encore l'utilisation de gestes de balayage dans le panier (glisser à droite pour acheter un article directement, ou à gauche pour le supprimer). L'application propose également la possibilité de choisir sa localisation directement sur une carte.

C'est de là que vient le nom *ShopEase*. Bien entendu, nous avons intégré toutes les fonctionnalités classiques d'une application de commerce électronique.

## Objectif

L'objectif et le choix de cette application découlent du fait qu'une application de commerce électronique permet de mettre en œuvre une grande variété de fonctionnalités, telles que l'inscription, la connexion, la gestion de stock, les méthodes de paiement, et la géolocalisation. Cela nous offre l'opportunité de maîtriser un maximum de logiques de code. Par conséquent, l'objectif principal de ce module sera pleinement atteint.

## User Story

vous trouvez ci dessous les users story sommarizé qui peut etre effectué dans ShopEase

- **Gestion de compte (Client & Vendeur)** : création, connexion et déconnexion
- **Découverte produits** : parcourir le catalogue et consulter les détails
- **Recherche produits** : saisie textuelle ou commande vocale
- **Panier** : ajouter, modifier (quantité), supprimer et consulter le total
- **Commande** : initier depuis le panier, confirmer l'adresse, paiement simulé et confirmation
- **Interface vendeur** : consulter et rechercher son inventaire
- **Gestion catalogue (Vendeur)** : ajouter, modifier et supprimer des produits

## Diagramme de Cas d'utilisation

nous avons commence par construire un Diagramme de cas d'utilisation qui matche les user story que nous avons mis au dessus. vous trouvez le diagramme de cas d'utlisation dans [ce lien](#)

## Maquette

Dans cette mini-version du rapport, nous incluons uniquement le lien vers les maquettes. Celles-ci ont été réalisées avec Figma. Vous pouvez consulter l'ensemble des interfaces créées en suivant [ce lien](#).

# Diagramme de Classe

Nous avons mis en place ce diagramme de classe qui correspond aux principales entités du système et aux relations entre elles, afin de visualiser l'architecture de l'application et de faciliter la compréhension du modèle de données. vous pouvez l'écarter dans [ce lien](#)

## Choix techniques

Pour réaliser *ShopEase*, nous avons choisi des outils adaptés à nos besoins et compétences :

- **Figma** : pour concevoir rapidement des maquettes interactives et collaborer facilement grâce au partage en ligne.
- **Flutter** : pour développer une application mobile fluide avec un seul code en Dart, avec l'avantage du *hot reload* et d'un rendu performant.
- **FastAPI** : pour créer un back-end rapide et asynchrone, avec des API RESTful faciles à mettre en place.
- **Firebase** : pour gérer l'authentification des utilisateurs, stocker les données de manière sécurisée (Firestore) et faciliter l'intégration de services backend sans devoir déployer de serveur dédié.

## Diagramme d'Architecture

Après la réalisation du projet, nous avons mis en place deux architectures : l'une est simplifiée et décrit les différents composants de *ShopEase*, auxquels nous pouvons accéder via des API, que vous pouvez consulter [ici](#), et l'autre est plus détaillée, que vous pouvez consulter [ici](#).

## Problème Récontré

- **Environnement de développement** : difficulté à exécuter **Android Studio** de manière fluide sur nos machines personnelles en raison de ressources système limitées (RAM).
- **Transition vers Linux** : nécessité d'installer un environnement Linux pour résoudre les problèmes de reconnaissance du SDK Android sous Windows, ce qui a engendré une perte de temps.
- **Utilisation de Visual Studio Code** : alternative testée mais lente sur nos configurations, moins intégrée que **Android Studio**.
- **Apprentissage de nouvelles technologies** : prise en main de **FastAPI** et d'autres outils a nécessité un investissement significatif en temps d'apprentissage.
- **Problèmes techniques dans le code** : gestion du panier et mise à jour du prix total après ajout, suppression ou achat d'articles.

## Pratique de codage suivie

- Sur le dépôt GitHub, nous avons créé une branche par fonctionnalité pour conserver un historique de commits dédié et faciliter la restauration en cas d'erreur.
- Nous avons élaboré un widget servant de « template » pour chaque classe, afin de le réutiliser sur différentes pages.

## Conclusion

Cette expérience autour de ShopEase nous a permis de mettre en pratique nos compétences et de découvrir de nouvelles technologies. Nous sommes fiers d'avoir terminé tout ce qui a été demandé dans les moindres détails et espérons avoir la chance à l'avenir de travailler sur des projets aussi motivants.

vous trouvez un le dépôt github [ici](#), ainsi une version plus détaillé (que nous recommandons de le lire) du rapport [ici](#)