

# Présentation de l'équipe

- Chef de projet : Zifan YAO
- Equipe Back-end : Hugo MARTIN, Ye YUAN, Ali AIT FORA
- Equipe Front-end : Tianhao WU, Hugues VOGEL
- Test & QA : Wui WONG



#### Introduction

- Une application web, pour ceux habitants en résidence.
- Pour les jeunes... et les moins jeunes.
- Pour faire les courses, depuis chez soi!
- Aucun tracas de paiement : tout est automatique, et sécurisé!



## Ce que notre application peut offrir

- Commander des produits autour de chez soi (même frais!),
- Choisir sa date de livraison,
- Recevoir les articles dans un casier sécurisé, en bas de chez soi,
- Payer en ligne, satisfait ou remboursé.



# Pourquoi choisir fais-mes-courses (fmc) ?

#### clients:

- trop occupé!
- ou juste pas envie de sortir :)
- un réel besoin ressenti en tant qu' étudiant

#### magasins:

- augmenter ses ventes
- publicité pour toucher plus de public
- diminuer le risque du stock

#### livreurs:

- faire du dollar
- sans trop de déplacements
- faire du sport en même temps!



## **Avantages de FMC**

- Sécurité: casiers verrouillés, pas de contact direct client/livreur, paiements par service sûr, réputation des livreurs et clients
- Rapidité: pouvoir se faire livrer au plus tôt 1 heure après la commande.
- **Liberté**: 24h pour récupérer ses articles aux casiers.
- Anticipation: pouvoir commander une semaine en avance.

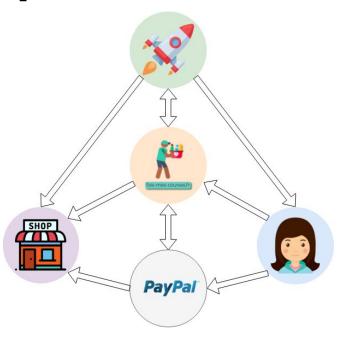


## Etude de la concurrence

- Amazon : service à domicile, pour des produits variés, à des prix très compétitifs.
- Foodora/Deliveroo : livraison de repas à domicile, le jour même, avec léger surcoût.
- Carrefour/Auchan... Drive : courses préparées pour vous, à récupérer en voiture.



## **Description fonctionnelle**



+ API Google Maps

- + API Paypal
- + API Datagram



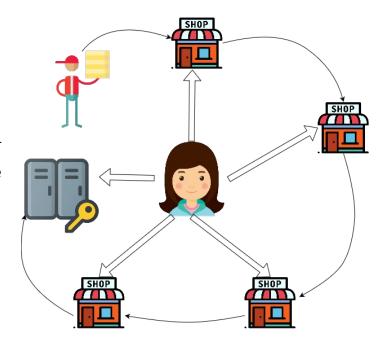
#### **Conditions d'utilisation**

- Limites sur la commande : prix limite, poids limite, taille des courses limite.
- Limites sur le rayon des magasins : on ne peut commander que chez des magasins à proximité.
- Limites de temps : date de livraison et de commande.
- Responsabilités du client : prélèvement différé avec vérification des fonds financiers, responsabilité lors de la récupération de la commande.
- Responsabilités du magasin : responsabilité de disponibilité
- Responsabilités du livreur : engagement contractuel



## Cas d'utilisation

- Client: Passer une commande et payer
- Livreur: Recevoir l'ordre de commande
- Magasin: Préparer la commande
- Livreur: Récupérer les produits du magasin, déposer dans le casier
- Client: Récupérer ses articles





# **Description des casiers**





actualisation des données à 11:45 et 17:45

Système réfrigéré

Verrouillage et déverrouillage par NFC/RFID

Système anti-vol



fais-mes-courses.fr

## Gestion de paiement

Utilisation d'un service tiers type **Paypal / Stripe** pour la gestion des transactions financières.

L'exécution des transactions est en logique interne.

Typiquement : exécution d'une autorisation de prélèvement, garantie de la satisfaction client, puis capture des fonds.







## **Exigences non fonctionnelles**

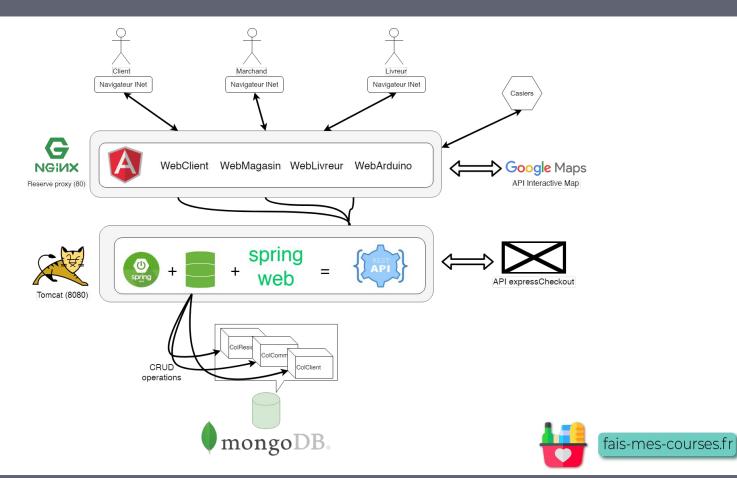
- Modularité/cloisonnement : Spring et AngularJS
- Scalabilité : mongoDB
- Look & feel : responsivité sur toute plateforme

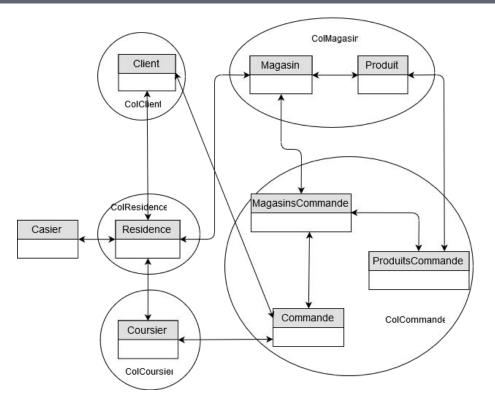


# **Architecture technique**

- Spring Boot avec intégration Spring Data, Spring Web
- AngularJS
- mongoDB







# Schéma mongoDB



## **Présentation POC**

- Réalisation d'un scénario "Happy Path" simplifié.
  - o Authentification via e-mail et mdp
  - Choix d'un magasin
  - Sélection d'articles
  - Paiement via Paypal API

0



## **Axes d'amélioration**

- Implémentation de toute les fonctionnalités décrites
- Exportation de l'application sur plateforme mobile
- Système de paiement propriétaire
- Implémentation d'un protocole de communication entre casiers et système type CoAP
- Améliorer l'approche de la logique métier : proposer un meilleur modèle économique, faire un système pour résoudre les litiges d'une meilleure façon, etc...



