

# APPLICATION PIZZERIA FLUTTER

Lien du depot github : <https://github.com/Melanie-devv/Pizzeria>

## DESCRIPTION DU PROJET :

Il s'agit d'un projet individuel réalisé à l'école du 29/09/23 au 10/11/23.

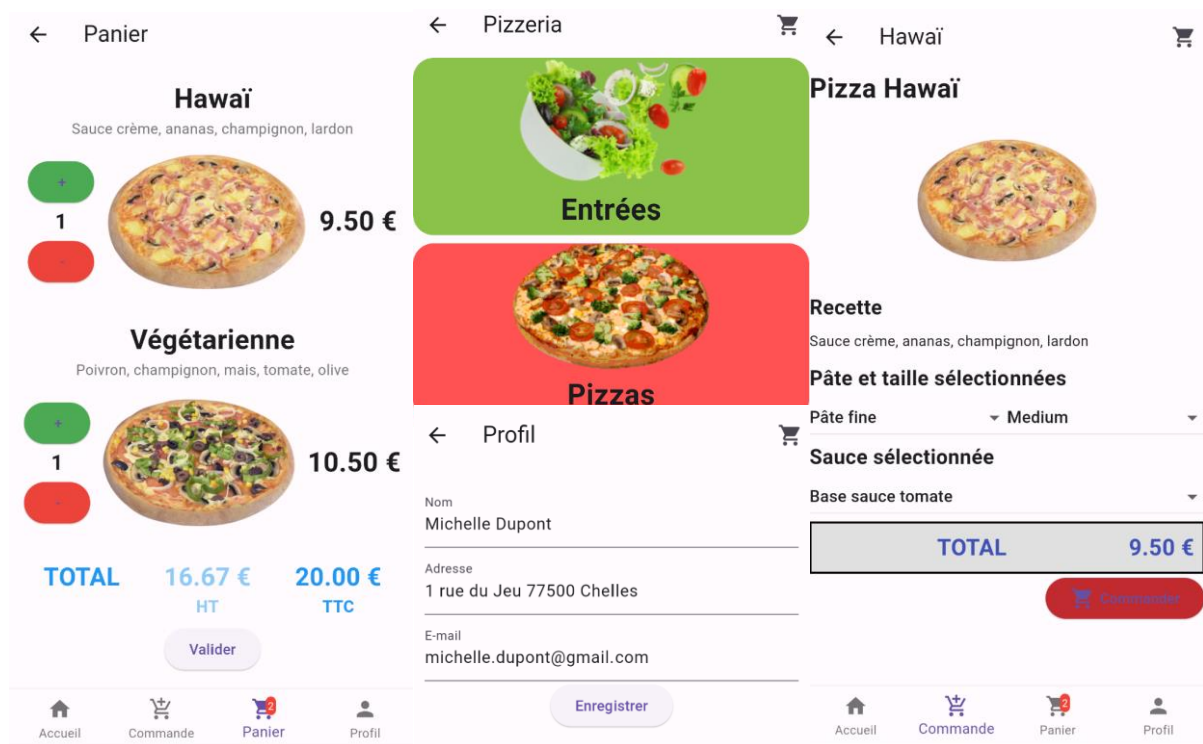
C'est une application android Flutter (ainsi qu'une web app sur navigateur) qui comprend différentes pages pour commander une pizza. Plusieurs options sont possibles, on peut sélectionner la sauce que l'on souhaite pour sa pizza, sa taille, son type de pâte, etc...

L'utilisateur peut aussi visualiser son panier et son total, ajouter ou supprimer des pizzas dans sa commande, et enfin, commander.

Il y a aussi une page profil modifiable avec des informations qui peuvent s'enregistrer.

Les données des pizzas sont récupérées à partir d'un fichier (.json) sur un serveur local (wamp).

Voici quelques exemples de visuels :



# COMPETENCES DEVELOPPEES

## Gérer le patrimoine informatique

- " ▶ Recenser et identifier les ressources numériques
- ▶ Exploiter des référentiels, normes et standards adoptés par le prestataire informatique
- ▶ Mettre en place et vérifier les niveaux d'habilitation associés à un service
- ▶ Vérifier les conditions de la continuité d'un service informatique
- ▶ Gérer des sauvegardes
- ▶ Vérifier le respect des règles d'utilisation des ressources numériques"

Pour la réalisation de ce projet, nous devons suivre un cahier des charges défini et « Exploiter des référentiels, normes et standards adoptés par le prestataire informatique » qui est notre professeur. Nous avons à notre disposition des « morceaux » de code que nous devons compléter. Il nous fallait donc adopter son style de codage et ses normes. De plus, nous devons « Vérifier le respect des règles d'utilisation des ressources numériques » à travers différents tests au sein du projet. Ainsi que « Gérer des sauvegardes », notamment au niveau des données liées aux pizzas qui étaient dans un serveur distant.

## Répondre aux incidents et aux demandes d'assistance et d'évolution

- " ▶ Collecter, suivre et orienter des demandes
- ▶ Traiter des demandes concernant les services réseau et système, applicatifs
- ▶ Traiter des demandes concernant les applications"

Ce projet s'inscrit dans un cadre où il nous a été demandé (via une consigne) de répondre aux demandes d'évolution d'une application de pizza via plusieurs étapes de réalisation. De plus, une revue a été faite par notre prof nous demandant de traiter ses demandes, corrections, etc... Il a donc fallu « collecter, suivre et orienter des demandes ». Au fur et à mesure de l'avancée du projet, de nouvelles demandes émergeaient.

## **Mettre à disposition des utilisateurs un service informatique**

- " ▶ Réaliser les tests d'intégration et d'acceptation d'un service
- ▶ Déployer un service
- ▶ Accompagner les utilisateurs dans la mise en place d'un service"

Bien sur, nous avons dû tester notre projet pour s'assurer de sa bonne conformité et voir s'il répondait bien à toutes les contraintes et demandes du cahier des charges (compétence « Réaliser les tests d'intégration et d'acceptation d'un service »). De plus, l'application était fonctionnelle et bénéficiait d'un fichier contenant les détails des pizzas stockés sur un serveur local (« Déployer un service »). Nous avons également dû expliquer comment notre application fonctionnait par le biais d'une démo et d'un échange avec notre professeur, nous avons donc « Accompagné les utilisateurs dans la mise en place d'un service ».

## **Organiser son développement professionnel**

- " ▶ Mettre en place son environnement d'apprentissage personnel
- ▶ Mettre en œuvre des outils et stratégies de veille informationnelle
- ▶ Gérer son identité professionnelle
- ▶ Développer son projet professionnel"

Ce projet s'inscrit directement dans le cadre de mon projet professionnel puisque j'ai effectué un stage dans une entreprise qui avait besoin de faire une application en flutter. Les différents tutos que j'ai appris au cours de ce projet m'ont donc permis de « Mettre en place son environnement d'apprentissage personnel » ainsi que de « Développer son projet professionnel » puisque plus tard, je voudrais devenir développeuse full stack.

## DESCRIPTION DU PROJET

Tout d'abord, nous avons utilisé des widgets du Framework Flutter tels que Scaffold, ListView, InkWell, etc... pour faire l'UI. Au début les images étaient locales dans le fichier assets. (nous verrons les détails de la création du web service pour récupérer ces images depuis le fichier json plus tard).

**Pour ce qui est des pages**, nous avons la page principale qui permet de choisir entre boissons, desserts, pizza... Ensuite une page PizzaList qui affiche une liste de toutes les pizzas pouvant être commandées. Au clic d'une pizza s'affiche la page PizzaDetail, plus précise, qui permet de faire le choix de la sauce, la pâte,... et de commander sa pizza en fonction du prix affiché. Ensuite nous avons la page panier et la page de profil de l'utilisateur.

**Pour ce qui est du panier**, il y a différentes stratégies

Le panier va être disponible sur toutes les pages. Il faut donc que l'objet Panier soit disponible depuis l'ensemble des autres objets.

- Passage par paramètre

Le plus simple est de créer un objet panier dans la classe racine de notre application : la classe MyHomePage. L'objet sera ensuite transmis, via les constructeurs, à tous les autres objets.

- Le Design Pattern Singleton

Le Design Pattern Singleton peut être utilisé pour centraliser les données.

- Les Provider

Le package Provider permet d'implémenter le Design Pattern Observable. Il se base donc sur un objet pouvant notifier une liste d'objets d'un changement.

La classe Panier serait chargée de notifier tout changement aux autres classes de type UI.

- Le Design Pattern Bloc (Business Logic Component)

Ce Design Pattern est basé sur les Streams et les événements pour échanger les données. Il est plus difficile à appréhender.

## Stratégie retenue

Dans un premier temps, nous allons prendre le passage par paramètre. Ensuite, nous allons implémenter un Provider (notamment pour afficher la notification au dessus de l'icone panier qui nous informe de combien de pizza nous avons commandé)

**Pour ce qui est de la navigation**, j'ai créé une app bar et une bottom navigation bar personnalisée et intégrée sur toutes les pages.

**Pour ce qui est du web service**, on récupère l'ensemble des pizzas dans le widget "PizzaList". Ce widget est avec état. L'obtention des données se fera donc dans la méthode initState() grâce à une classe qui accède au Web Service.

La classe Pizza va convertir le JSON en objet.

On doit aussi créer un service web de type Apache (vous pouvez bien sûr créer un service nginx ou autre si vous préférez).

```
<?php
header("Content-Type: application/json; charset=UTF-8");

$json = array();
$json[] = array(
    'id' => 1,
    'title' => 'Barbecue',
    'garniture' => 'La garniture',
    'image' => 'http://localhost/static/images/pizzas/pizza-bbq.jpg',
    'price' => 6.5
);
$json[] = array(
    'id' => 2,
    'title' => 'Hawaï',
    'garniture' => 'La garniture',
    'image' => 'http://localhost/static/images/pizzas/pizza-hawai.jpg',
    'price' => 9.5
);
$json[] = array(
    'id' => 3,
    'title' => 'Epinards',
    'garniture' => 'La garniture',
    'image' => 'http://localhost/static/images/pizzas/pizza-spinach.jpg',
    'price' => 8.5
);
$json[] = array(
    'id' => 4,
    'title' => 'Végétarienne',
    'garniture' => 'La garniture',
    'image' => 'http://localhost/static/images/pizzas/pizza-vegetable.jpg',
    'price' => 10.5
);
echo json_encode($json);
```

Dans le répertoire racine de notre serveur web, nous avons créé le répertoire différents répertoire ainsi qu'un fichier index.php qui contient des données json de ce type (voir ci-contre).

Afin d'obtenir les données à partir d'un Web Service de type RESTful, j'ai utilisé un objet http dans mon application flutter pour contacter (faire un appel http) sur le Webservice. Puis convertir les données JSON en objet Pizza.

Maintenant que les images des pizzas sont chargées depuis une URL, des modifications au niveau du code s'impose. Par exemple, il faudra utiliser Image.network au lieu de Image.asset.