

# OC PIZZA



## SOLUTION DE GESTION DE PRISE DE COMMANDE VIA UNE APPLICATION ANDROID

Dossier de conception technique

Version 1.1

Dossier réf. : **DCT-ITCD20230323OCP**

**Auteur**

HAMMER JULIEN

*Lead développeur*

# TABLE DES MATIÈRES

<b>1 - Versions.....</b>	<b>3</b>
<b>2 - Introduction .....</b>	<b>4</b>
2.1 - Objet du document .....	4
2.2 - Références .....	4
2.3 - Besoins .....	4
<b>3 - Architecture technique.....</b>	<b>5</b>
3.1 - Composants généraux.....	5
3.1.1 - Infrastructure : .....	5
3.1.2 - Base de données : .....	5
3.2 - Application Android OC Pizza .....	5
3.2.1 - Nom de l'application : .....	5
3.2.2 - Langage et environnement de développement : .....	5
3.2.3 - Architecture de développement : .....	5
3.2.4 - Modèle : .....	6
3.2.5 - Vue : .....	6
3.2.6 - ViewModel : .....	6
3.2.7 - Gestion de l'authentification : .....	6
3.2.8 - Gestion des notifications : .....	6
3.2.9 - Compatibilité et optimisation : .....	6
3.2.10 - Services tiers et intégrations : .....	7
3.2.11 - Sécurité : .....	7
<b>4 - Architecture de déploiement.....</b>	<b>8</b>
4.1 - Serveur de base de données.....	8
4.1.1 - Description : .....	8
4.1.2 - Caractéristiques techniques : .....	8
4.1.3 - Informations importantes / points particuliers : .....	8
4.1.4 - Déploiement de l'application : .....	8
<b>5 - Architecture logicielle .....</b>	<b>9</b>
5.1 - Principes généraux.....	9
5.2 - Application Android .....	9
5.2.1 - Modèle : .....	9
5.2.2 - Vue : .....	9
5.2.3 - ViewModel : .....	9
<b>6 - Points particuliers.....</b>	<b>10</b>
6.1 - Visibilité .....	10
6.2 - Caractéristiques.....	10
6.3 - Intuitivité.....	10
6.4 - Systèmes d'exploitation .....	10
6.5 - Bugs.....	11
<b>7 - Glossaire .....</b>	<b>12</b>

# 1 - VERSIONS

Auteur	Date	Description	Version
HAMMER JULIEN	23/03/2023	Création du document	1.1
HAMMER JULIEN	24/03/2023	Finalisation du document	1.1

## 2 - INTRODUCTION

### 2.1 - Objet du document

Le présent document constitue le dossier de conception technique de l'application Android dans le but de remplacer l'ancien système informatique du groupe OC PIZZA devenu obsolète.

L'objectif de ce document est donc de présenter les fonctionnalités du nouveau système informatique.

Les éléments du présent dossier découlent :

- de la demande initiale du client Franck et Lola d'OC Pizza,
- des différents échanges avec ce client.

### 2.2 - Références

Pour de plus amples informations, se référer également aux éléments suivants :

1. **DCF – réf. DCF-ITCD20230323OCP** : Dossier de conception fonctionnelle de l'application
2. **DE – réf. DE-ITCD20230323OCP** : Dossier d'exploitation

### 2.3 - Besoins

Les besoins identifiés pour ce projet sont les suivants :

- Être plus efficace dans la gestion des commandes de la réception à la livraison en passant par la préparation ;
- Suivre en temps réel les commandes passées, en préparation et en livraison ;
- Suivre en temps réel le stock d'ingrédients restants pour savoir quelles pizzas peuvent encore être réalisées ;
- Proposer un site Internet pour que les clients puissent :
  - Passer des commandes, en plus de la prise de commande par téléphone ou sur place ;
  - Payer en ligne leur commande s'ils le souhaitent – sinon, ils paieront directement à la livraison ;
  - Modifier ou annuler leur commande tant que celle-ci n'a pas été préparée.
- Proposer un aide-mémoire aux pizaiolos indiquant la recette de chaque pizza.

## 3 - ARCHITECTURE TECHNIQUE

### 3.1 - Composants généraux

#### 3.1.1 - Infrastructure :

L'infrastructure de l'application Android OC Pizza repose sur une architecture client-serveur. Le serveur est responsable de la gestion des données, de l'authentification des utilisateurs et de la communication avec les clients mobiles via des API Restful.

#### 3.1.2 - Base de données :

La base de données est conçue pour stocker et gérer les informations essentielles de l'application, telles que les données clients, les commandes, les produits et les tarifs. PostgreSQL, MySQL ou Amazon Relational Database Service sont utilisés pour assurer la cohérence et l'intégralité des données.

### 3.2 - Application Android OC Pizza

#### 3.2.1 - Nom de l'application :

Le nom de l'application est OC Pizza.

#### 3.2.2 - Langage et environnement de développement :

L'application est développée en Java, en utilisant Android Studio comme environnement de développement intégré.

#### 3.2.3 - Architecture de développement :

L'application Android suit l'architecture Modèle-Vue-ViewModel (MVVM) pour séparer les préoccupations et faciliter la maintenance et l'évolutivité.

### **3.2.4 - Modèle :**

Le modèle représente les structures de données et la logique métier de l'application. Il interagit avec la base de données et l'API pour récupérer et stocker les données.

### **3.2.5 - Vue :**

La vue est responsable de l'affichage des données de l'utilisateur et de la collecte des entrées utilisateur. Elle implémente les principes de Material Design pour offrir une expérience utilisateur cohérente et agréable. La vue est passive et n'a pas connaissance du modèle.

### **3.2.6 - ViewModel :**

Le ViewModel agit comme un intermédiaire entre la vue et le modèle. Il expose les données et les commandes pour la vue et s'occupe de la logique de présentation. Le ViewModel récupère les données du modèle, les traite et les expose sous une forme adaptée à la vue.

### **3.2.7 - Gestion de l'authentification :**

L'application intègre un système d'authentification et d'autorisation pour garantir que seuls les utilisateurs autorisés aient accès aux fonctionnalités pertinentes. Les services d'authentification Facebook et Google sont utilisés pour faciliter la connexion des utilisateurs.

### **3.2.8 - Gestion des notifications :**

Un système de notifications est mis en place pour informer du statut de la commande.

### **3.2.9 - Compatibilité et optimisation :**

L'application est optimisée pour fonctionner sur une variété d'appareils Android, en tenant compte des différentes tailles d'écran et résolutions, ainsi que des différentes versions d'Android.

- Téléphones mobiles
- Tablettes

### **3.2.10 - Services tiers et intégrations :**

Les services de paiement sont gérés par Stripe et l'application est initialement distribuée via Google Play pour les appareils Android.

### **3.2.11 - Sécurité :**

L'accès aux comptes d'administration est limité aux co-fondateurs Lola et Franck.

Le respect des normes RGPD sont respectés et les données bancaires ne sont pas conservées.

## 4 - ARCHITECTURE DE DÉPLOIEMENT

L'architecture de déploiement représente les différents serveurs et composants impliqués dans le fonctionnement de l'application OC Pizza. Elle montre comment les composants interagissent entre eux et avec les services tiers utilisés.

### 4.1 - Serveur de base de données

#### 4.1.1 - Description :

Le serveur de base de données est responsable de la gestion et du stockage des informations relatives aux commandes, aux clients, aux pizzas, aux stocks d'ingrédients et toutes les autres données nécessaires. Il permet l'accès sécurisé et rapide aux données pour l'application OC Pizza.

#### 4.1.2 - Caractéristiques techniques :

MySQL et Amazon Relational Database Service sont utilisés pour mettre en place le système de gestion de bases de données relationnelles.

#### 4.1.3 - Informations importantes / points particuliers :

Le serveur de base de données doit être configuré avec des paramètres de sécurité appropriés pour assurer la protection des données.

Des sauvegardes régulières de la base de données doivent être effectuées pour prévenir la perte de données en cas de problème.

#### 4.1.4 - Déploiement de l'application :

L'application sera déployée sur le google play store via un compte développeur créé pour un projet. Une release note sera rédigé pour la version déployée.

Le déploiement sera tout d'abord effectué en internal testing avec une liste prédéfinie d'utilisateurs et une durée d'un mois avant d'être déployée en production.

L'accès à l'application (via login) sera tout de même limité aux utilisateurs inscrit dans le système par l'administrateur de OC Pizza.



## 5 - ARCHITECTURE LOGICIELLE

### 5.1 - Principes généraux

Les sources et versions du projet sont gérées par Git, les dépendances et le packaging par Apache Maven / Grunt.

De plus L'architecture d'un logiciel décrit la manière dont seront agencés les différents éléments d'une application et comment ils interagissent entre eux. Cette étape est donc l'une des premières du développement logiciel et intervient lors de la phase de conception. Elle est généralement réalisée par un architecte logiciel ou un architecte solution, élément central du projet de développement.

Il faut déterminer les éléments nécessaires pour le bon déroulement du projet.

### 5.2 - Application Android

L'application Android suit l'architecture Modèle-Vue-ViewModel (MVVM) pour séparer les préoccupations et faciliter la maintenance et l'évolutivité.

#### 5.2.1 - Modèle :

Le modèle représente les structures de données et la logique métier de l'application. Il interagit avec la base de données et l'API pour récupérer et stocker les données.

#### 5.2.2 - Vue :

La vue est responsable de l'affichage des données à l'utilisateur et de la collecte des entrées utilisateur. Elle implémente les principes de Material Design pour offrir une expérience utilisateur cohérente et agréable. La vue est passive et n'a pas connaissance du modèle.

#### 5.2.3 - ViewModel :

Le ViewModel agit comme un intermédiaire entre la vue et le modèle. Il expose les données et les commandes pour la vue et s'occupe de la logique de présentation. Le ViewModel récupère les données du modèle, les traite et les expose sous une forme adaptée à la vue.

## 6 - POINTS PARTICULIERS

### 6.1 - Visibilité

Il est important de faire comprendre l'importance de l'utilisation de l'application auprès des futurs utilisateurs d'OC Pizza. Ils devront comprendre l'utilité de l'application sans avoir à supporter un nombre conséquent de publicités.

### 6.2 - Caractéristiques

Une application mobile consomme des données importantes en fonction de son contenu et de sa performance. Si l'utilisateur dispose de peu de stockage dans son téléphone, il ne l'installera pas sur son appareil.

Dès sa conception, l'application doit se démarquer de ses concurrents avec un design intuitif et irréprochable. Il faut que l'application attire l'attention des utilisateurs

Des enquêtes de satisfaction doivent être réalisées auprès des utilisateurs après son déploiement afin de sonder les retours d'expériences sur l'application et ainsi proposer une offre qui soit la plus adaptée.

### 6.3 - Intuitivité

L'application doit être simple de compréhension et remplir son rôle.

Pour capter l'attention des utilisateurs, l'apparence de l'application doit être soignée, ainsi que son ergonomie, sa résolution et la vitesse de chargement des pages.

### 6.4 - Systèmes d'exploitation

Pour le moment, nous nous concentrons sur une application fonctionnant avec Android mais par la suite il faudra très certainement envisager IOS et éventuellement un site internet.

## 6.5 - Bugs

Il faudra bien réaliser les tests sur l'application et éviter les bugs sur :

- Le téléchargement et l'utilisation globale ;
- L'accès à tous les onglets ;
- L'interactivité ;
- Le téléchargement des mises à jour ;
- Le chargement de toutes les pages ;
- Le poids de l'application.

## 7 - GLOSSAIRE

<b>Client</b>	Un client est un logiciel qui a pour but d'envoyer une demande à travers des requêtes afin de communiquer avec le serveur (l'interroger et attendre son retour).
<b>Serveur</b>	Un serveur est un dispositif informatique (matériel et logiciel) qui offre des services à un ou plusieurs clients. Il va répondre aux différentes requêtes émanant des clients.
<b>Base de données</b>	Une base de données est un « conteneur » stockant des données telles que des chiffres, des dates ou des mots, pouvant être retraités par des moyens informatiques pour produire une information. Par exemple, des chiffres et des noms assemblés et triés pour former un annuaire.
<b>MySQL</b>	MySQL, de l'anglais Structured Query Language, est un système de gestion de bases de données relationnelles et un logiciel gratuit
<b>Amazon Relational Database Service</b>	C'est un service web qui facilite la configuration, l'exploitation et la mise à l'échelle d'une base de données relationnelle dans le Cloud. Il fournit des capacités redimensionnables, à faible coût, pour les bases de données relationnelles classiques, et gère les tâches courantes d'administration de base de données.
<b>Orienté objet</b>	Il s'agit de briques logicielles appelées objets. Un objet représente un concept, une idée ou toute entité du monde physique, comme une voiture, une personne ou encore une page d'un livre. Il possède une structure interne et un comportement, et il sait interagir avec ses pairs.
<b>Modèle</b>	Le modèle représente les structures de données et la logique métier de l'application. Il interagit avec la base de données et l'API pour récupérer et stocker les données.
<b>Vue</b>	La vue est responsable de l'affichage des données à l'utilisateur et de la collecte des entrées utilisateur. Elle implémentera les principes de Material Design pour offrir une expérience utilisateur cohérente et agréable. La vue est passive et n'a pas connaissance du modèle.
<b>ViewModel</b>	Le ViewModel agit comme un intermédiaire entre la vue et le modèle. Il expose les données et les commandes pour la vue et s'occupe de la logique de présentation. Le ViewModel récupère les données du modèle, les traite et les expose sous une forme adaptée à la vue.
<b>Stripe</b>	Le logiciel et les API de Stripe permettent de mettre en place les paiements, d'effectuer des virements et de gérer leurs activités en ligne.

<b>API</b>	<p>Les interfaces de programmation d'application, de l'anglais Application Programming Interface (API) permettent à un produit ou service de communiquer avec d'autres produits et services sans connaître les détails de leur mise en œuvre. Elles offrent plus de flexibilité, simplifient la conception, l'administration et l'utilisation.</p> <p>Le Rest API (ou API Restfull) de l'anglais Representational State Transfer Application Program Interface, est un style architectural qui permet aux logiciels de communiquer entre eux sur un réseau ou sur un même appareil.</p>
<b>Material Design</b>	<p>Google a utilisé le Material Design pour unifier le style graphique de ses applications et de ses plateformes. Son interface présente donc l'avantage d'être très adaptée à une utilisation pour tout appareil et toutes résolutions d'affichage. Elle est également simple d'utilisation. Le Material Design optimiserait l'expérience utilisateur, tout en offrant des ressources aux développeurs et aux concepteurs d'interfaces.</p>
<b>Git</b>	<p>C'est un système de contrôle de version d'un projet et un logiciel gratuit. Il permet de traquer tous les fichiers d'un projet. On garde un historique de tous les changements et à tout moment nous pourrions revenir en arrière.</p>
<b>IOS</b>	<p>Il s'agit d'un système d'exploitation d'Apple qui équipe actuellement les smartphones iPhone et les tablettes iPad. C'est le système qui apparaît à l'écran et qui permet d'utiliser votre appareil : téléphoner, écrire un message, lancer une application, ...</p>
<b>Android</b>	<p>C'est un système d'exploitation créé par Google qui permet la communication entre un utilisateur et son appareil. C'est un petit peu comme Windows, mais sur smartphone et tablettes.</p>
<b>Bug</b>	<p>Un bug est un fonctionnement non désiré dans un ordinateur ou un programme informatique. On parle de bug quand le logiciel ne fait pas ce qu'on lui a demandé, qu'il se bloque et ne répond plus ou qu'il plante.</p>
<b>Internal testing</b>	<p>"Internal testing" (ou "test interne" en français) est une pratique courante dans le développement de logiciels. Il s'agit de tester un logiciel ou une application avant de le ou la rendre disponible au public ou aux clients. Les tests internes sont effectués par l'équipe de développement elle-même ou par une équipe de testeurs dédiée, qui vérifie que le logiciel fonctionne correctement et qu'il est exempt de bugs et d'erreurs. Les tests internes peuvent inclure des tests de fonctionnalité, des tests de performance, des tests de sécurité et des tests de compatibilité. Ils permettent de s'assurer que le logiciel est prêt pour une utilisation plus large et qu'il répond aux normes de qualité attendues.</p>

<b>Normes RGPD</b>	RGPD signifie « Règlement Général sur la Protection des Données » (anglais : GDPR « General Data Protection Regulation »). Elles encadrent le traitement des données personnelles sur le territoire de l'Union européenne (UE). Le contexte juridique s'adapte pour suivre les évolutions des technologies et des sociétés (usages accrus du numérique, développement du commerce en ligne...). Ce règlement harmonise les règles en Europe en offrant un cadre juridique unique aux professionnels. Il permet de développer leurs activités numériques au sein de l'UE en se fondant sur la confiance des utilisateurs.
<b>Release note</b>	Il s'agit d'un document qui accompagne la sortie d'une nouvelle version d'un logiciel ou d'une application. Elle contient toutes les informations sur les nouvelles fonctionnalités, les améliorations, les corrections de bugs et les problèmes connus de la version. Elle est destinée aux utilisateurs finaux, aux administrateurs système et aux développeurs pour les aider à comprendre les changements à apporter à la nouvelle version et donc les aider à mettre à jour le logiciel ou l'application.