



# Vente de pizza en ligne

Dossier de conception technique  
Version 1.0

Auteur  
Bernole Yohan  
*Analyste Programmeur*

# TABLE DES MATIÈRES

<b>1.</b>	<b>VERSIONS</b>	<b>3</b>
<b>2.</b>	<b>INTRODUCTION</b>	<b>4</b>
2.1.	Domaine fonctionnel	4
2.2.	Objet du document	4
2.3.	Références	4
<b>3.</b>	<b>ARCHITECTURE LOGICIELLE</b>	<b>5</b>
3.1.	Base de données	5
3.1.1.	Diagramme de classe	5
3.1.2.	Modèle physique de données	6
3.2.	Hébergement de la solution	6
3.3.	Service d'authentification	7
3.4.	Service de paiement	7
3.5.	Outil de géolocalisation	7
3.6.	Structure des sources	7
<b>4.</b>	<b>ARCHITECTURE DES COMPOSANTS</b>	<b>8</b>
4.1.	Diagramme de composants	8
4.2.	Explication textuelle des composants	8
<b>5.</b>	<b>ARCHITECTURE DE DÉPLOIEMENT</b>	<b>9</b>
5.1.	Diagramme de déploiement	9
5.2.	Explication textuelle des éléments	10
<b>6.</b>	<b>GLOSSAIRE</b>	<b>10</b>

# 1. VERSIONS

Auteur	Date	Description	Version
BERNOLE Yohan	28/10/2021	Création du document	1.0

## 2. INTRODUCTION

### 2.1. Domaine fonctionnel

Le groupe OC Pizza demande un site Internet pour que les clients puissent

- Passer leurs 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

Il faut donc intégrer:

- La gestion du compte utilisateur
- La gestion des commandes, de leur réception à leur livraison en passant par leur préparation
- Le stock d'ingrédients restants pour savoir quelles pizzas sont encore réalisables
- Un aide-mémoire aux pizzeraiolos indiquant la recette de chaque pizza

### 2.2. Objet du document

Le document fonctionnel fait précédemment, nous permet de voir toutes les fonctionnalités qui doivent être présentes dans l'application. Ainsi, le présent document constitue le dossier de conception technique de l'application. Il est destiné à l'attention des développeurs, mainteneurs et de l'équipe technique.

L'objectif de l'analyse technique étant de lister les contraintes spécifiques dont les développeurs vont devoir tenir compte pour coder l'application, ce document présentera les langages et les conventions de développement, l'architecture logicielle et de déploiement.

Les éléments du présent dossier découlent:

- De l'analyse des besoins de l'entreprise OC Pizza
- De la rédaction du dossier de conception fonctionnelle

### 2.3. Références

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

- Document du domaine fonctionnel - **Bernole\_Yohan\_1\_dossier\_fonctionnel\_092022.pdf**
- Dossier d'exploitation - **Bernole\_Yohan\_3\_dossier\_d'exploitation\_092022.pdf**
- Procès-verbal de livraison finale - **Bernole\_Yohan\_4\_PV\_de\_livraison\_finale\_092022.pdf**
- Dossier **Annexes**

## 3. ARCHITECTURE LOGICIELLE

### 3.1. Base de données

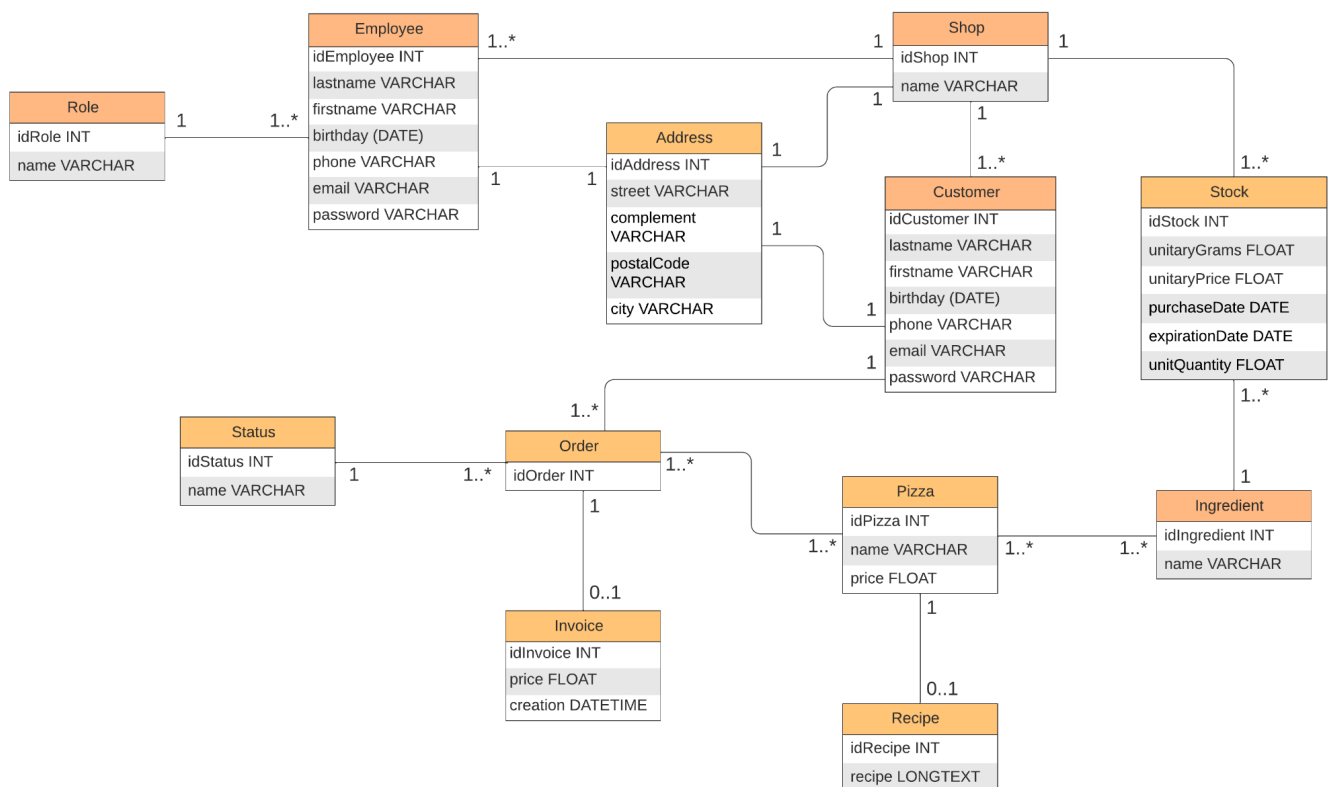
Afin de créer la base de données, plusieurs étapes ont dû être mises en œuvre. De la création du diagramme de classe en passant par le modèle physique de données jusqu'à la base de données elle-même.

#### 3.1.1. Diagramme de classe

Le diagramme de classe permet de voir

- Les entités
- Les associations entre les entités.

C'est la première ébauche de notre base de données qui provient de l'analyse du domaine fonctionnel.

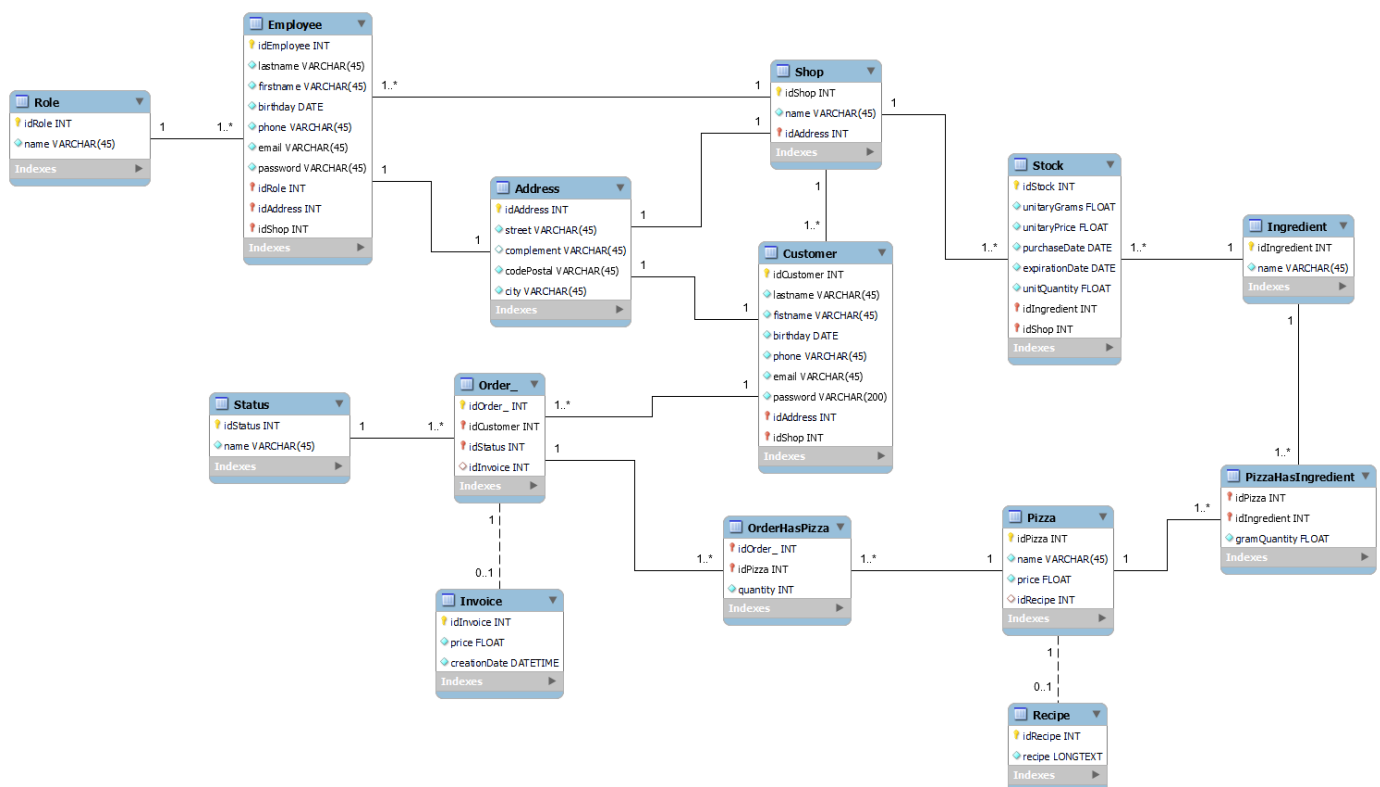


### 3.1.2. Modèle Physique de données

Le modèle physique de données décrit clairement la structure du système en modélisant

- Les classes
- Les attributs
- Les opérations
- Les relations entre les objets.

Il reprend le diagramme de classe en donnant des types aux données correspondant aux types SQL. Il indique aussi les cardinalités entre les associations ainsi que les clés primaires et les clés étrangères qui sont explicitement décrites.



\*Le MPD est disponible dans le dossier annexe

### 3.2. Hébergement de la solution

L'application sera développée en React Native côté client sur un serveur Node. Il sera hébergé grâce à la solution IONOS pour l'hébergement Web qui comprend 100 Go d'espace, un nom de domaine et la possibilité de créer plusieurs bases de données MySQL. Il inclut aussi le logiciel PhpMyAdmin qui permet de gérer la base de données.

<https://www.ionos.fr/hebergement/hebergement-web>

### 3.3. Service d'authentification

Les moyens d'authentification avec l'utilisation de OAuth 2.0 seront déployés conformément à la documentation dans les liens ci-dessous.

Pour Google

- <https://developers.google.com/identity/sign-in/web/sign-in>

Pour Facebook

- <https://developers.facebook.com/docs/facebook-login/web>

### 3.4. Service de paiement

Le service de paiement sera déployé grâce à la solution Stripe qui permet d'accepter les paiements par carte bancaire.

<https://stripe.com/fr/payments>

Pour déployer la solution Stripe avec React nous pouvons utiliser la documentation correspondante.

<https://stripe.com/docs/stripe-js/react>

### 3.5. Outil de géolocalisation

En ce qui concerne la géolocalisation des livreurs, pour le consommateur, la solution retenue pour indiquer la position est l'utilisation de Google Map API. L'utilisateur pourra se rendre sur son espace et suivre sa commande.

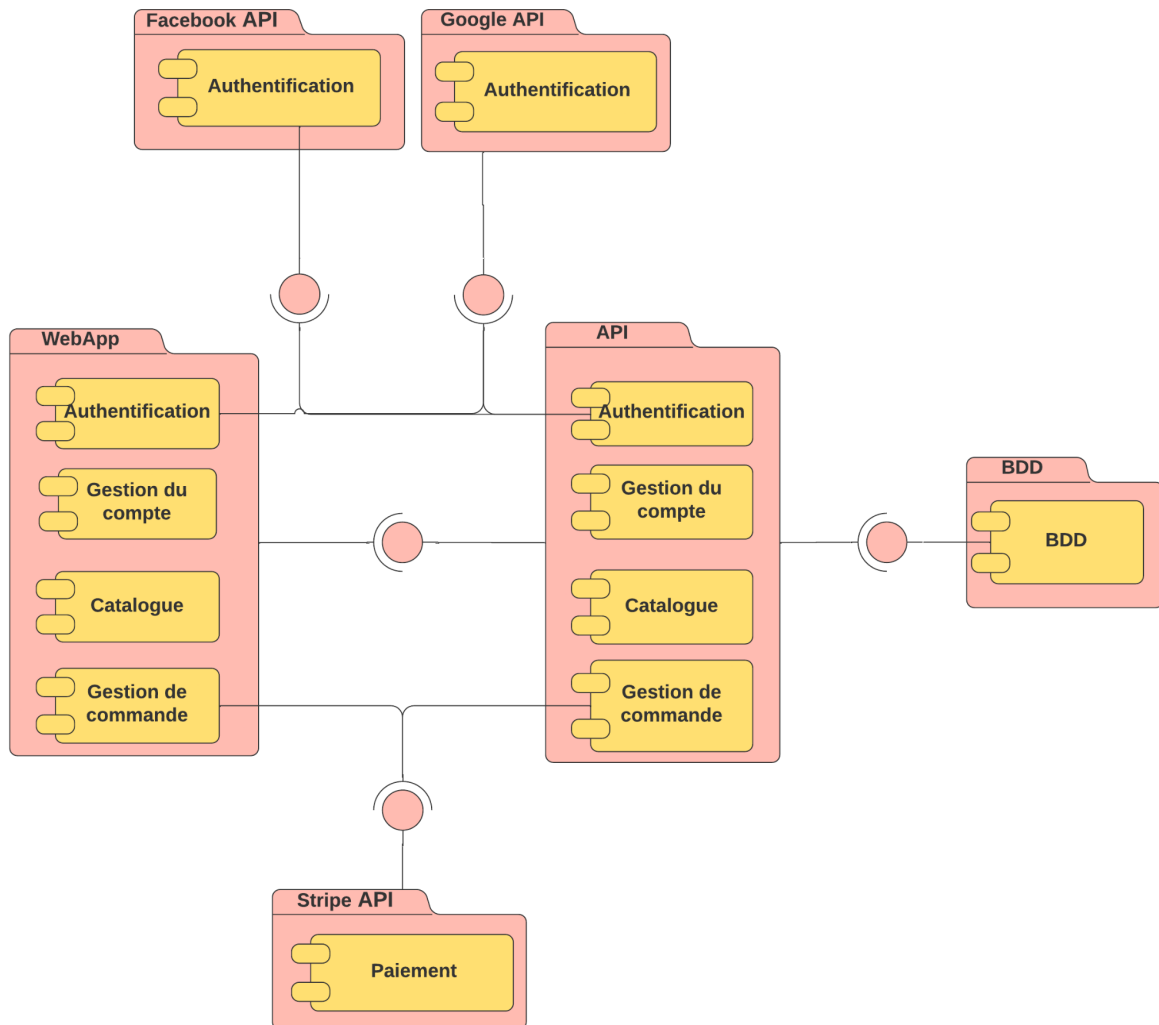
### 3.6. Structure des sources

L'application est développée en React Native. La structuration des répertoires du projet suit la logique suivante:

```
└─ /src
    ├── /assets
    ├── /components
    ├── /context
    ├── /hooks
    ├── /pages
    ├── /services
    ├── /utils
    └─ App.js
    └─ index.js
```

## 4. Architecture des composants

### 4.1. Diagramme de composants



### 4.2. Explication textuelle des composants

L'utilisateur interagit avec les composants de l'application web. L'API de l'application permet d'utiliser les fonctions pour obtenir et/ou modifier les composants dans ma base de données.

Le catalogue représente l'ensemble des produits, la gestion du compte représente l'utilisateur avec toutes les informations correspondantes. Puis il y a la gestion de commande de celui-ci.



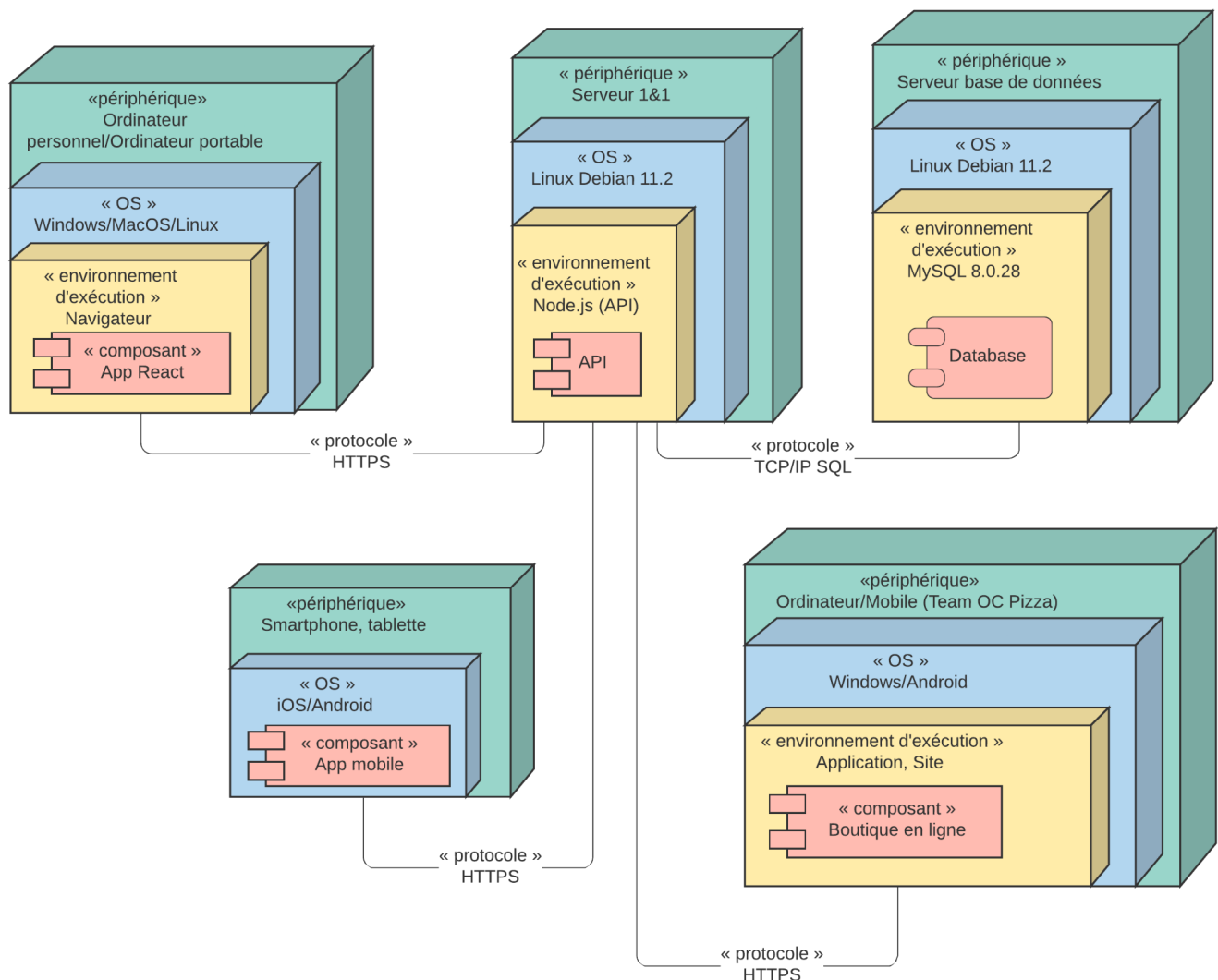
L'authentification peut se faire via la connexion Facebook, Google, mis en place avec les API correspondantes, ou par email & mot de passe.

Les API génèrent un jeton pour permettre de s'authentifier en relation avec l'application, et tout est stocké en lien avec l'API dans la base de données.

Le processus de paiement est similaire, l'API Stripe permet d'utiliser le paiement par carte. Elle s'occupe de tout le processus de paiement pour vérifier que les données correspondent bien et que l'opération est autorisée par la banque de l'utilisateur. Les données de la commande sont ainsi stockées dans la base de données.

## 5. Architecture de déploiement

### 5.1. Diagramme de déploiement



## 5.2. Explication textuelle du déploiement

Le site web sera responsive donc consultable sur ordinateur, tablette et mobile (iOS & Android) grâce au navigateur web. L'application est réalisée en React Native et est hébergée avec la solution IONOS vue ci-dessus sur un serveur Linux Debian avec Node installé sur le serveur.

La solution IONOS permet aussi d'héberger la base de données et de l'utiliser avec l'interface Phpmyadmin en lien avec le système de gestion de BDD MySQL.

Du côté d'OC Pizza, ils peuvent accéder à l'application avec un navigateur web sur ordinateur et tablette pour recevoir les commandes, gérer le catalogue, proposer le paiement, les aide-mémoire etc..

## 6. GLOSSAIRE

BDD = Base de données

API = Application Programming Interface

MPD = Modèle Physique de données