



IT Consulting & Development

**Dossier de Conception
Technique**

**Système
OC PIZZAS**

Chef de projet : Maxime ROUX

OPENCLASSROOMS

Parcours : Développeur d'application iOS

Entreprise : IC Consulting & Development
Chef de projet : Maxime ROUX
Email : maxime.roux@icconsulting.com
contact@icconsulting.com
Tél : 06XXXXXXXX
04XXXXXXXX

TABLE DES MATIÈRES

1. HISTORIQUE DES MODIFICATIONS	3
2. INTRODUCTION	4
2.1. Objet du document	
2.2. Références	
3. ARCHITECTURE TECHNIQUE	5
3.1. Composants généraux	
3.1.1. Package client	
3.1.2. Package employé	
3.1.3. Package point de vente	6
3.1.4. Composants externes	
3.2. Application Web	
3.3 Base de données	
3.3.1. Modèle physique de données	7
4. ARCHITECTURE DE DÉPLOIEMENT	8
4.1. Serveur	
5. ARCHITECTURE LOGICIELLE	9
5.1. Principes généraux	
5.1.1. Les couches	
5.1.2. Les modules	
5.1.3. Structures de source	10
6. POINTS PARTICULIERS	11
6.1. Gestion des logs	
6.2. Environnement de développement	
6.3. Procédure de packaging / livraison	
7. GLOSSAIRE	12

1. HISTORIQUE DES MODIFICATIONS :

Le tableau ci-dessous est un journal de de suivi des évolutions du document.

Si une modification significative intervient après partage du document, une nouvelle ligne serait ajoutée au tableau. Cette ligne indiquerait la nouvelle version, l'auteur Les différentes versions qui se sont substituées les unes aux autres sont renseignées dans ce tableau. Cela permet de s'assurer que l'on travaille bien même dernière version.

VERSION	AUTEUR	DATE	DÉSCRIPTION
V1	Maxime ROUX	18/01/2022	Création du document

2. INTRODUCTION :

2.1. Objet du document :

- » Le présent document constitue le dossier de conception technique de l'application du système informatique OC Pizza.
- » Les éléments du documents sont :
 - L'architecture des composants logiciels
 - L'architecture de déploiement
 - L'architecture logicielle
 - Les fichiers de configuration et de logs de l'application
 - L'environnement de développement

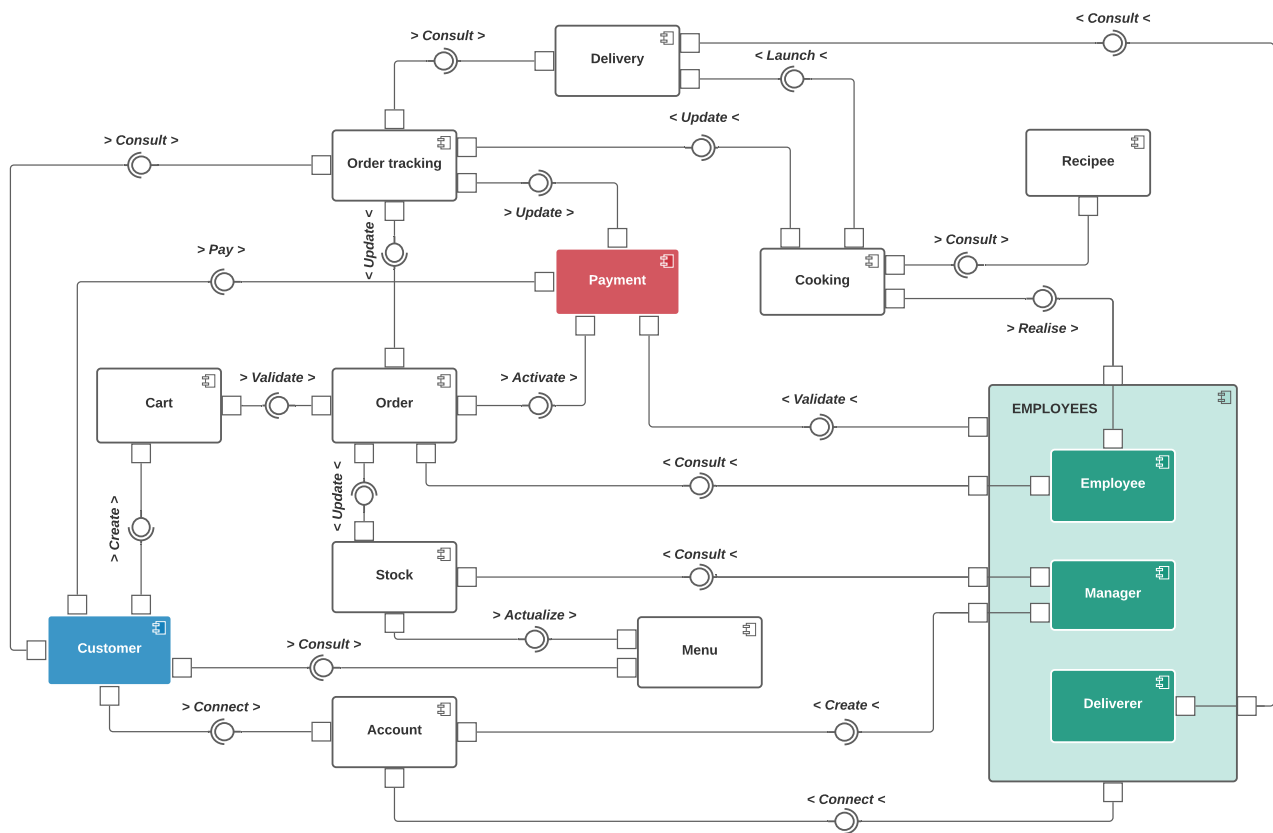
2.2. Références :

La réalisation de ce document Pour de plus amples informations, se référer :

- » **DCF** : Dossier de conception fonctionnelle de l'application
- » **DE** : Dossier d'exploitation

3. ARCHITECTURE TECHNIQUE :

3.1. Composants généraux :



3.1.1. Package client :



Ce package est l'interface web client. Elle est responsive, compatible à la fois avec les ordinateurs les appareils mobiles).

3.1.2. Package employé :



Ce package est l'interface web réservée aux collaborateurs OC Pizzas. Selon leur poste, ils n'auront pas accès aux mêmes fonctionnalités du package point de vente.

- » L'accès pizzaïolo permet la gestion de commande, aux aide-mémoires, à la consultation des stocks, de réaliser le paiement.
- » L'accès livreur permet de consulter les informations de livraison, de clôturer une commande et de réaliser le paiement.
- » L'accès Manager permet de réaliser toutes les actions précédentes, ainsi que de modifier les menus, les aide-mémoires, de créer les comptes employés et d'en gérer les autorisations, de gérer les stocks et d'accéder au suivi de l'activité / statistiques.

3.1.3. Package point de vente :



Ce package gère les fonctionnalités du site tels que la gestion des stocks, les menus, les commandes, les paniers, les comptes, les aide-mémoires, le suivi de l'activité...

3.1.4. Composants externes :



Le module de paiement est un module externe qui permet la validation du paiement via Merkanet (solution de paiement en ligne avec contrat de vente à distance proposé par BNP Paribas).

3.2. Application web :

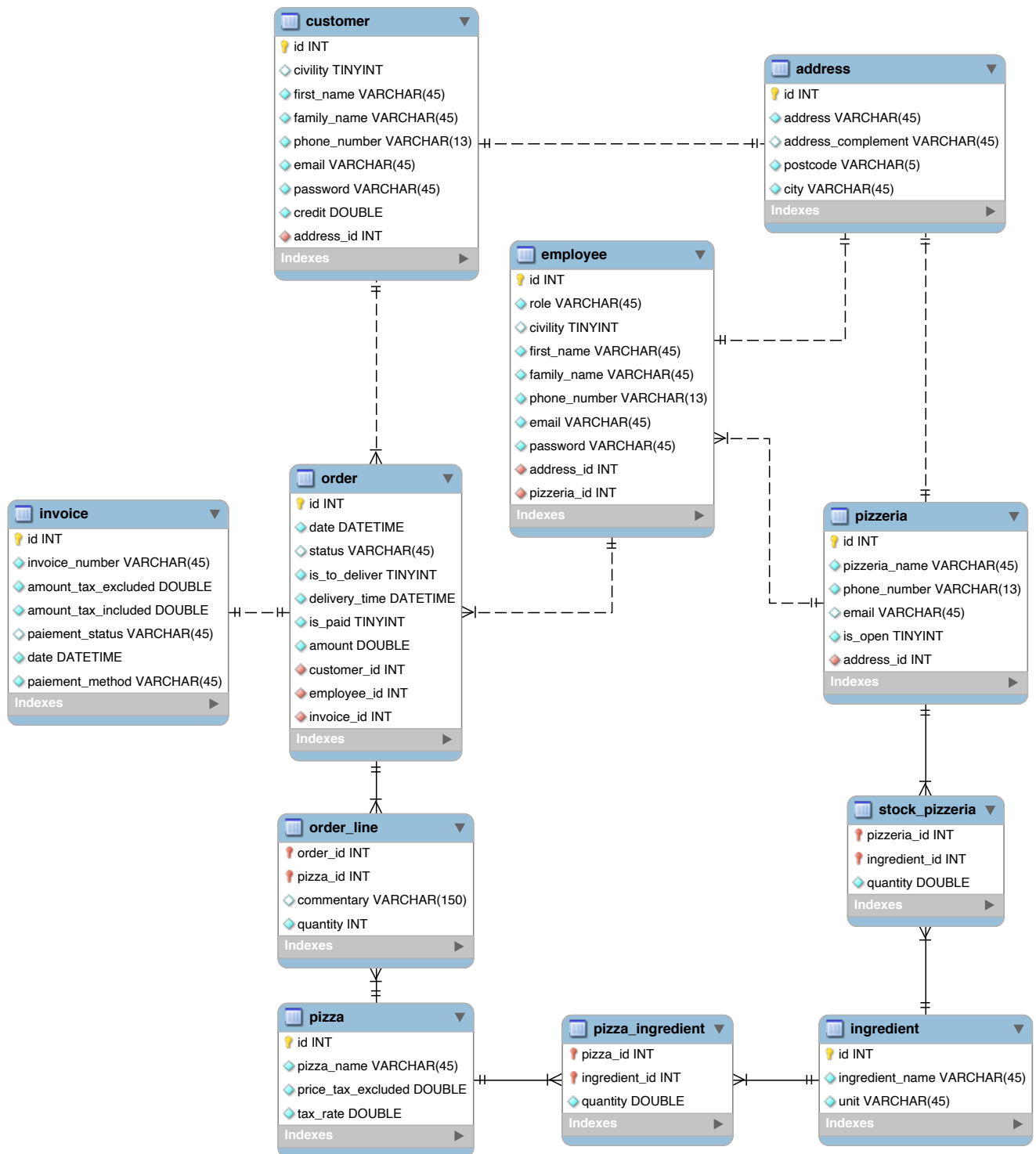
La pile logicielle est la suivante :

- » **Serveur Web** : Apache 2.4. C'est un logiciel installé sur le système permettant le transfert des données via le protocole HTTP.
- » **Langage PHP 8** qui est interprété par le serveur afin de pouvoir utiliser le Framework Symfony
- » **Framework Symfony** : c'est un ensemble de composants PHP ainsi qu'un framework MVC libre écrit en PHP. Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d'accélérer le développement d'un site web.

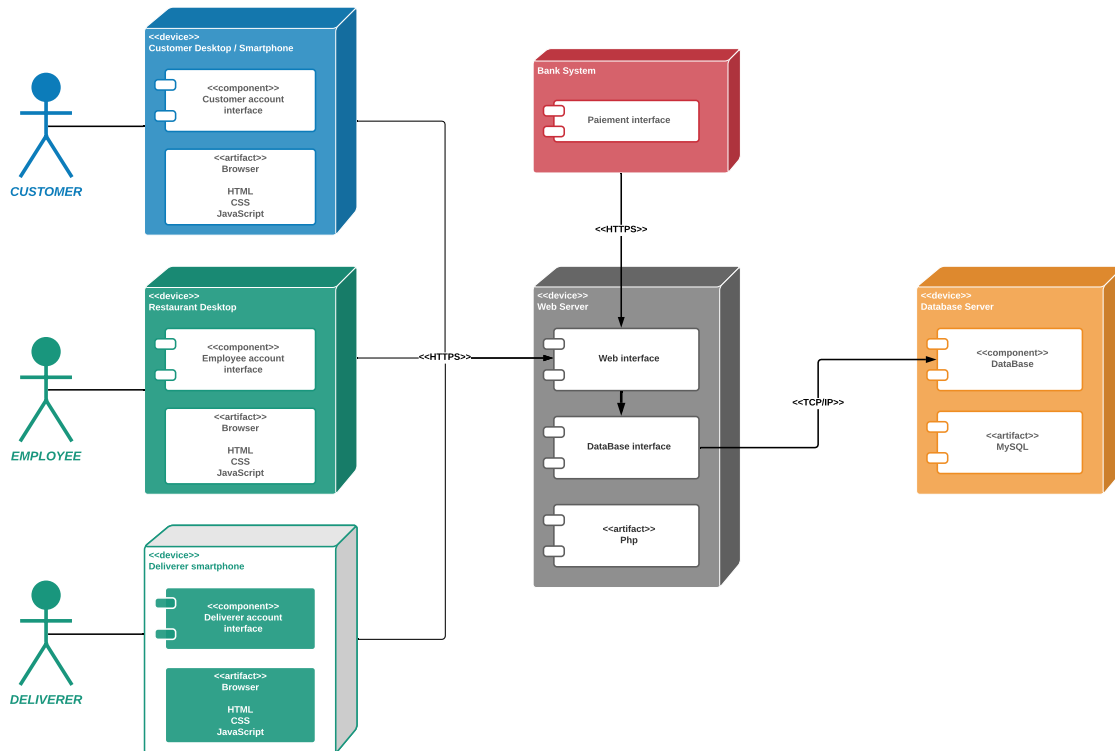
3.3. Base de données :

- » **Base de données** : MySQL 5.7. permet de traiter et stocker les données dont le système a besoin pour son utilisation.

3.3.1. Modèle physique de données :



4. ARCHITECTURE DE DÉPLOIEMENT :



4.1. Serveur :

- » Hébergement cloud
- » Ressources : jusqu'à 12 CPU threads, 48 Go de RAM en ressources allouées
- » Technologie : NVME 3.2GO/S
- » Nom de domaine Offert
- » Espace Disque : illimité
- » Base de Données : illimitées
- » Transfert Mensuel : illimité
- » Interface cPanel® : interface technique facilitant la gestion
- » Service Technique 24/7

5. ARCHITECTURE LOGICIELLE :

5.1. Principes généraux :

Les sources et versions du projet sont hébergées sur Github et sont gérées avec Git.

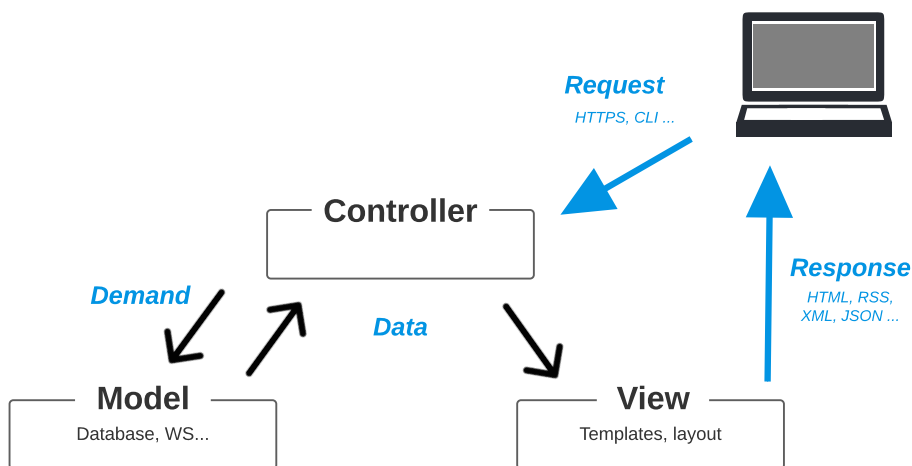
Les dépendances sont gérées avec Composer.

5.1.1. Les couches :

L'architecture applicative est la suivante :

- » **Model** : Implémentation du modèle des objets métiers
- » **View** : Implémentation de la vue qui s'affichera à l'écran et qui permettra à l'utilisateur d'interagir avec le système
- » **Controller** : interprète les interactions utilisateur, et commande la vue en fonction des règles données par le model

Structure MVC (Model, View, Controller) :



5.1.2. Les modules :

Le module **Doctrine** permet à l'interface de communiquer avec la base de données. Il est le module par défaut du Framework **Symfony**.

5.1.3. Structures de source :

```
your_project/
├── assets/
├── bin/
│   └── console
├── config/
│   ├── packages/
│   └── services.yaml
├── public/
│   ├── build/
│   └── index.php
├── src/
│   ├── Kernel.php
│   ├── Command/
│   ├── Controller/
│   └── ...
├── templates/
├── tests/
├── translations/
├── var/
│   ├── cache/
│   └── log/
└── vendor/
```

L'**arborescence** ci-dessus est créée automatiquement par le Framework **Symfony**. Elle permet une organisation claire des fichiers.

Les différents dossiers qui la composent sont :

- » **assets/** : contient les différentes ressources graphiques du site Web
- » **bin/** : contient les exécutables du projet, tels que ceux fournis avec le Framework Symfony (ex : la console) ou ceux des dépendances (phpunit, simple-phpunit, php-cs-fixer, phpstan)
- » **config/** : contient toutes les configurations du site
- » **public/** : contient les fichiers destinés à recevoir les requêtes utilisateurs
- » **src/** : contient l'application, les controllers et différentes ressources nécessaire à l'exécution du site
- » **templates/** : contient les templates utilisé dans le site
- » **tests/** : contient les différents tests de l'application
- » **translations/** : contient les traductions permettant de rendre le site multi-langues
- » **var/** : contient les fichiers de cache et de log
- » **vendor/** : contient les dépendances du site et les loaders

6. POINTS PARTICULIERS :

6.1. Gestion des logs :

Les logs permettent d'avoir un suivi de l'activité du site (erreurs, problème, cause de crash, ...). Ils sont gérés grâce à la librairie **Monolog**, qui est très répandue sur les serveurs PHP.

Monolog permet de trier les logs par jours et de supprimer automatiquement les plus anciens, afin de ne pas prendre inutilement de l'espace.

6.2. Environnement de développement :

- » Système d'exploitation : MacOS Big Sur version 11.6
- » Server Local : Apache : 2.4.52 / Php : 8.1.3 / MySQL : 8.0.27
- » Création du Modèle Physique de Données (MPD) : MySQL Workbench 8.0.27
- » IDE : SublimeText

6.3. Procédure de packaging / livraison :

Le système fera l'objet d'un déploiement sur l'hébergeur O2Switch lors de la livraison finale.

Lors de la souscription à l'offre d'hébergement O2Switch, les codes d'accès et un lien seront fournis au client.

Les livrables de l'application et les fichiers de la base de données seront disponibles sur le lien GitHub.

Les différents dossiers constituant les livrables de l'application seront ensuite transférés sur l'espace d'hébergement via le logiciel Filezilla.

Les fichiers de la base de données seront importés via l'outil PhpMyAdmin de l'interface d'hébergement cPanel.

Une formation d'une demi-journée sera proposée au client afin qu'il prenne en main le futur système et notamment le back office.

7. GLOSSAIRE :

TERME	DÉFINITION
Architecture	Elle désigne la structure générale inhérente à un système informatique, l'organisation des différents éléments du système (logiciels et/ou matériels et/ou humains et/ou informations) et des relations entre les éléments.
Dépendance	Bibliothèque nécessaire au fonctionnement de l'application.
Hébergement Web	C'est l'acte de mettre le site ou l'application web sur un serveur pour que tout le monde sur internet puisse y accéder.
MVC	MVC est un patron de conception (design pattern) très répandu pour réaliser des sites web. Il permet de séparer l'affichage des informations, les actions de l'utilisateur et l'accès aux données.
MPD	Un Modèle Physique de Données (MPD) est un outil de conception de base de données qui permet de définir la mise en oeuvre de structures physiques et de requêtes portant sur des données.
Responsive	Un site responsive est un site qui est conçu et développé de façon à pouvoir s'adapter à toutes les résolutions d'écran. C'est donc un seul et même site qui peut être consulté sur ordinateur, sur smartphone ou sur tablette.
Serveur Web	Un serveur web désigne à la fois une machine physique et un logiciel. Dans le premier cas, il s'agit d'un ordinateur relié à Internet et hébergeant des ressources. Ces ressources peuvent être des fichiers, des programmes ou des bases de données.