

# OC PIZZA

## Digital OC Pizza

Dossier de conception technique

Version 1.1

**Auteur**

Joseph Herradi

*Développeur informatique*

# 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
<b>3 - Architecture Technique.....</b>	<b>5</b>
<b>4 - Rappel de la solution technique.....</b>	<b>6</b>
<b>5 - Architecture de Déploiement.....</b>	<b>7</b>
<b>6 - Architecture logicielle.....</b>	<b>9</b>
6.1.1 - Applications Spring.....	9
6.1.2 - Les couches.....	9
<b>7 - Modélisation du domaine fonctionnel:.....</b>	<b>10</b>
<b>8 - Modèle physique de données.....</b>	<b>12</b>

# 1 - VERSIONS

Auteur	Date	Description	Version
Joseph Herradi	02/04/2019	Creation du document	1.0
Joseph Herradi	05/09/2019	Mise à jour du document	1.1

## 2 - INTRODUCTION

### 2.1 - Objet du document

OC Pizza est un jeune groupe de pizzeria en plein essor et spécialisé dans les pizzas livrées ou à emporter. Il compte déjà 5 points de vente et prévoit d'en ouvrir au moins 3 de plus d'ici la fin de l'année. Un des responsables du groupe a pris contact avec nous afin de mettre en place un système informatique, déployé dans toutes ses pizzerias et qui lui permettrait notamment de gagner en efficacité dans les process internes et de proposer un site de command en ligne aux clients.

Dans ce document, nous détaillons les spécifications techniques de ce projet suite à l'analyse et la validation du besoin d'OC PIZZA (cf document des spécifications fonctionnelles).

### 2.2 - Références

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

1. **DSF - OCP8**: Dossier de spécifications fonctionnelles
2. **DXP - OCP8** : Dossier d'exploitation de l'application

### 3 - ARCHITECTURE TECHNIQUE

La programmation orientée composant s'intègre très bien dans le contexte de la programmation orientée objet; en effet il ne s'agit, finalement, que d'un facteur d'échelle. L'approche composant peut être assimilée à l'approche objet non pas au niveau algorithmique mais au niveau de l'architecture générale de l'application.

Un composant est un élément constitutif d'un logiciel représenté par un classeur structuré et comportant une ou plusieurs interfaces requises ou offertes. Plusieurs classes réalisent en général son comportement interne: seules ses interfaces sont visibles. Un port est un point de connexion entre un composant et son environnement.

Ce dernier est représenté par un petit carré sur la bordure du contour du classeur, il peut être associé à une interface requise/offerte ou directement à un autre port par un connecteur delegate.

Notre système de commande en ligne peut être matérialisé par 4 composants:

1. Une interface utilisateur associée à tous les autres composants, correspond à une "vue" FRONT afin d'utiliser et contrôler l'ensemble des fonctionnalités du système.
2. Un composant "administration des données" pour l'import, export, modification et suppression des données par un panel user friendly. Grâce à ce composant, il sera possible d'ajouter des produits à la carte, renseigner les stocks, rédiger des memos de recettes ...
3. Un composant "gestion de commande" qui gère l'ensemble du processus d'une commande de la création à la finalisation.
4. Un composant "gestion de paiement électronique" permettant l'intégration des fonctionnalités de paiement en ligne.

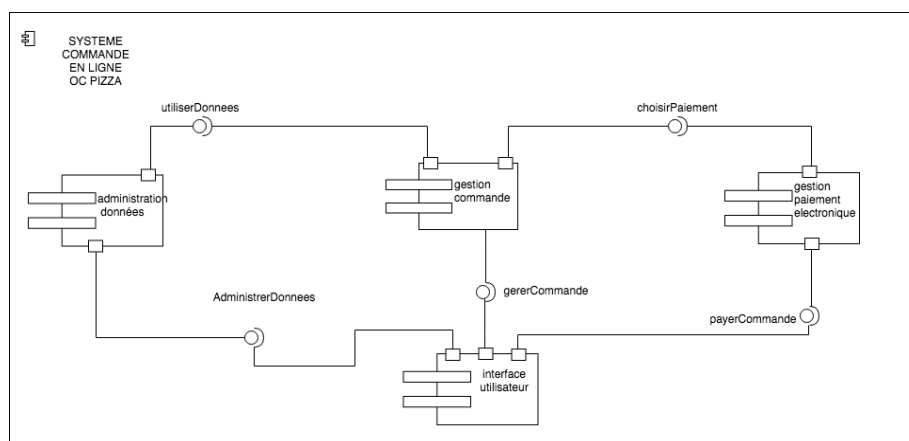


Diagramme des composants

## 4 - RAPPEL DE LA SOLUTION TECHNIQUE

A l'issue des échanges avec le client OC PIZZA, le choix technologique suivant a été validé pour la conception du système informatique:



L'application sera développée sous Java 8 avec une implémentation sous le framework Java JEE qui demeure une référence dans le développement web de part sa robustesse et sa fiabilité.



L'interface web sera réalisée à l'aide du framework Bootstrap disposant d'une vaste collection de création et d'outils d'interfaces graphiques et d'animations.



PostgreSQL

Système de gestion de base de données relationnelle Opensource disposant d'une grande communauté et régulièrement maintenu, il offre des performances remarquables.

PostgreSQL est un bon choix compte tenu de la structure de la base de données et la taille des données traitées.

## 5 - ARCHITECTURE DE DÉPLOIEMENT

Le système consistera en une application web déployée en ligne et pouvant être utilisée par l'ensemble des acteurs du projet cible (client, vendeur, cuisinier, livreur et responsable). Chacun des acteurs aura des droits spécifiques d'accès aux fonctionnalités de la solution selon les cas d'utilisation correspondants.

Une version mobile responsive de l'application sera particulièrement adaptée à l'utilisation du livreur pour la consultation de la commande et validation de la livraison.

Une application web backend facilitera l'administration des données du système notamment pour des utilisateurs non techniques afin de gérer les produits de la carte, les prix, le stock... de façon autonome. La base de données utilisée est PostgreSQL et reste un choix pertinent par rapport au dimensionnement du projet.

Nous proposerons au client OC pizza le déploiement sur un serveur dédié virtuel (VPS) Apache Tomcat d'un hébergeur web (exemple figure ci-dessous: Ikoula, lws, OVH ...). Les caractéristiques matérielles suffisantes pour installer l'environnement et garantir une bonne expérience utilisateur du système sont: 5 GO SSD - CPU 2 coeurs - 2 GO RAM. L'environnement du serveur sera une distribution Linux parfaitement adaptée à l'administration SSH à distance.

Hébergeur	Note Finale	Espace disque	Bande passante	RAM	CPU	Datacenter	Support client	Prix / mois (à partir de)
<b>Ikoula</b> VPS FlexiCloud	★★★★★	50 - 200 GB	100 MBS Full Duplex Max	1 - 16 GB	1 - 4 vCœurs	France	★★★★★	12,99 € <a href="#">Visiter Ikoula</a>
<b>LWS</b> VPS M	★★★★★	160 GB ou 80 GB SSD - 500 GB ou 180 GB SSD	Illimité	2 GB - 12 GB	1 - 4 vCœurs x 3,4 GHz	France	★★★★★	9,99 € <a href="#">Visiter LWS</a>
<b>1&amp;1 IONOS</b> VPS XL	★★★★☆	30 - 160 GB SSD	Illimité	512 MB - 8 GB	1 - 4 vCœurs	France, Europe, USA	★★★★☆	4,99 € <a href="#">Visiter 1&amp;1 IONOS</a>
<b>OVH</b> VPS SSD 2	★★★★☆	20 - 80 GB SSD	1 - 4 TB	2 GB - 8 GB	1 - 2 vCœurs x 2,4 GHz	France, Canada	★★★★☆	3,59 € <a href="#">Visiter OVH</a>

Différentes offres de VPS (source OpportunitésDigitales)

Enfin, le paiement électronique se fera grâce à l'intégration du système du prestataire de paiement retenu selon les critères d'OC PIZZA (notamment les frais de gestion...).

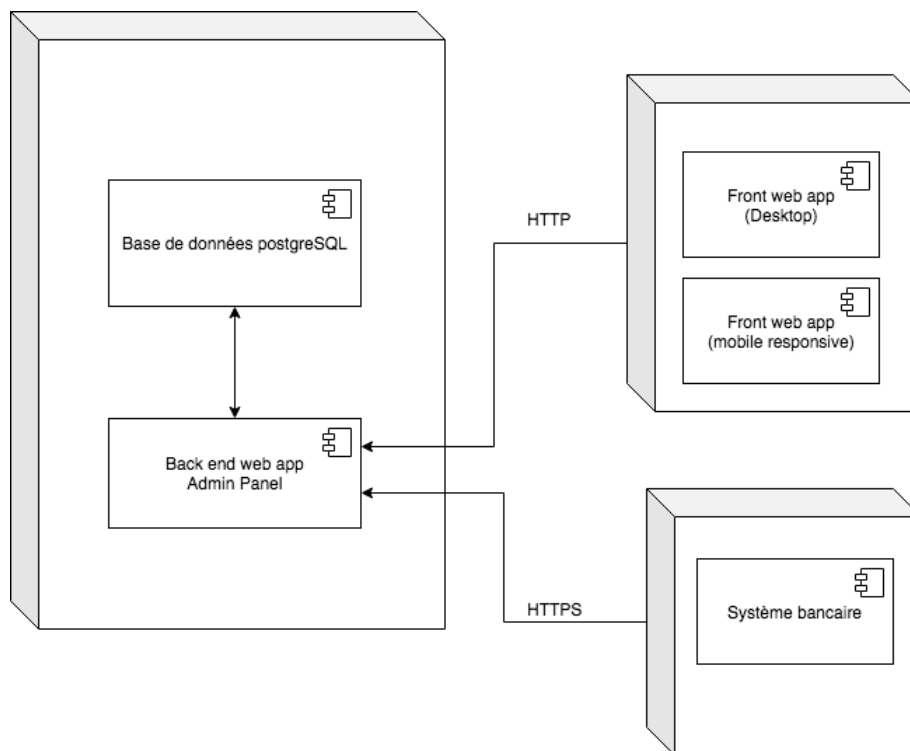


Diagramme de déploiement



## 6 - ARCHITECTURE LOGICIELLE

### 6.1.1 - Applications Spring

L'application web est constituée de 2 modules indépendants : Gestion des achats et gestion de la production.

Ces modules sont développés avec l'architecture MVC du Framework Spring.

### 6.1.2 - Les couches

L'architecture applicative est la suivante :

- une couche Présentation : responsable de l'interface client/application.
- une couche Modèle : responsable de la représentation et la persistance des données.
- une couche Métier : responsable de la logique métier et de la manipulation des données.

Couche	Rôle
Présentation	Regroupe les notions de vues. Elle représente la partie servant d'interface entre l'utilisateur et l'application.
Métier	Regroupe les fonctionnalités et algorithmes nécessaires au fonctionnement de l'application telles que : <ul style="list-style-type: none"><li>- L'affichage du catalogue en fonction des pizzas disponibles</li><li>- Choix de l'établissement en fonction de la distance du client</li></ul>
Modèle	Permet de manipuler les tables grâce aux entités et du mapping objet relationnel par Hibernate. Les entités utilisées peuvent être: <ul style="list-style-type: none"><li>- Produit</li><li>- Recette</li><li>- Commande</li><li>- Client</li></ul>

## 7 - MODÉLISATION DU DOMAINE FONCTIONNEL :

Suite à l'analyse du besoin du client OC PIZZA, il ressort le diagramme des classes suivant modélisant l'ensemble des concepts objets du domaine fonctionnel. Dans l'approche orientée objet, chaque classe est caractérisée par des attributs dont le type est indiqué ci-dessous.

Le multiplicité d'une association entre classes reflète la cardinalité des instances des classes entre elles. Nous utilisons également la notion d'énumération pour caractériser un attribut ayant des valeurs définies.

On distingue les classes suivantes :

- **Commande:** cette classe occupe une place centrale dans l'application. On remarque que plusieurs autres classes font le lien avec cette dernière. Elle permettra de définir pour un client et un établissement donnés, une commande dont le statut sera mis à jour selon le process interne. La définition du statut est faite dans l'énumération dans le diagramme ci-dessous. Les attributs `PaiementMode` et `ModeLivraisonRetrait` utilisent de même une énumération.
- **Ajoutproduit:** cette classe d'association détaille la quantité et le prix des produits choisis pour une commande donnée.
- **Produit:** Elle contient le nom des produits de la carte ainsi que leur disponibilité qui est obtenue grâce à un calcul utilisant le stock des ingrédients par établissement et la recette.
- **StockIngredients:** Cette classe permet la gestion du stock des ingrédients par établissement.
- **MemoRecette:** Elle permet de gérer les recettes des produits par les attributs ingrédients, quantités, temps de cuisson et température du four.
- **InfoEtablissement:** regroupe les coordonnées des différents établissements d'OC PIZZA.
- **Utilisateur** contenant les informations des utilisateurs et dont héritent 2 classes filles: **Client** (avec les attributs supplémentaires mail, adresse et tel) et **Personnel** avec une

agregation sur la classe InfoEtablissement.

- **Role:** cette classe permet de gérer les différents droits selon le type d'utilisateur. Une énumération sur l'attribut `type_role` est utilisée pour représenter les possibilités. Ainsi un client aura la possibilité de commander et de modifier une commande et non un livreur qui pourra uniquement consulter un récapitulatif et modifier le statut de la commande.
- **HistoriqueCommande:** Cette classe permet d'enregistrer l'évolution des statuts d'une commande avec l'identification des utilisateurs. Une méthode permet d'activer ou non cette fonctionnalité.
- **Facture:** Cette classe est nécessaire pour la génération d'une facture pour un client et une commande donnés.

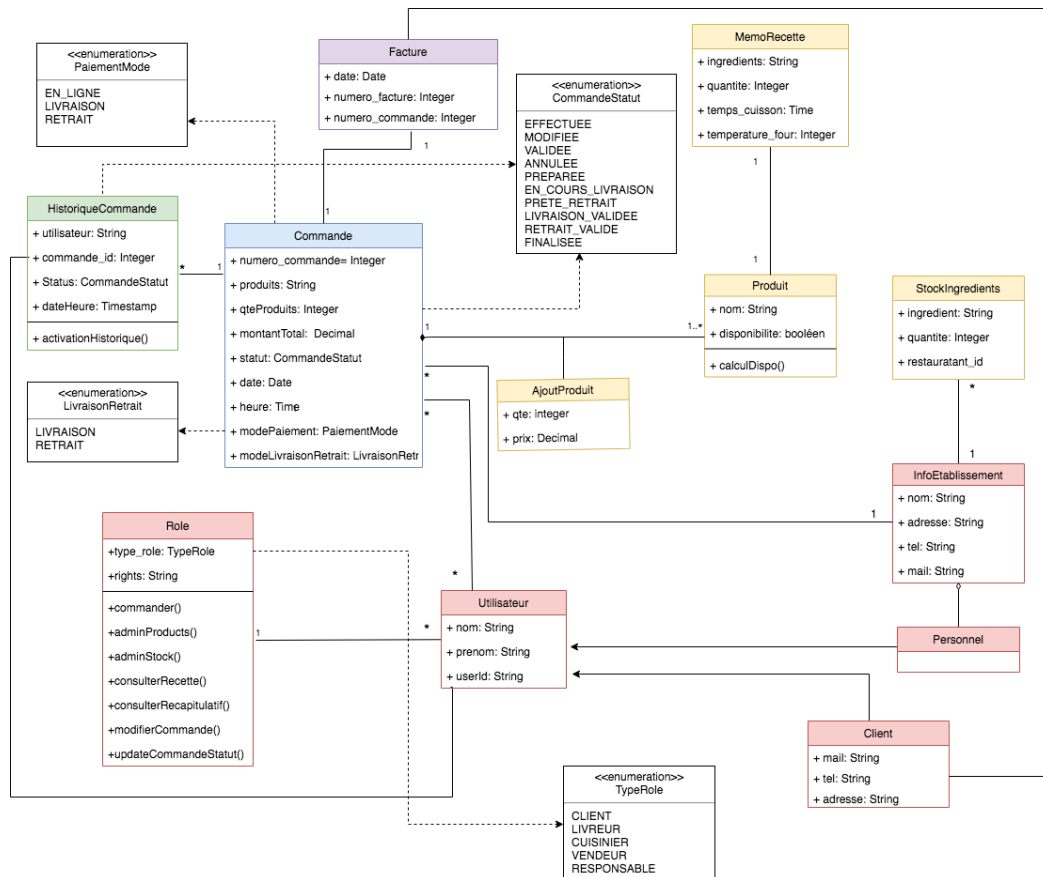
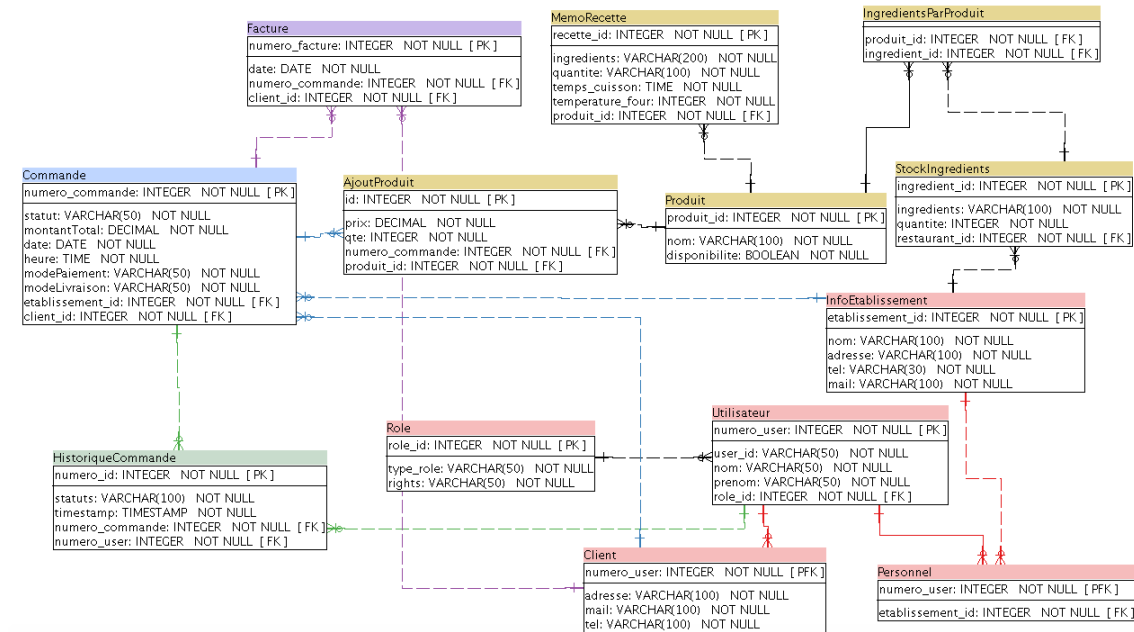


Diagramme des classes

## 8 - MODÈLE PHYSIQUE DE DONNÉES

Grâce au diagramme des classes préalablement réalisé, le MPD suivant a été élaboré par la création de tables avec les clés primaires correspondantes (attribut PK) . Les clés secondaires FK permettent quant à elles de faire le lien entre les différentes tables.

Enfin, concernant les tables Client et Personnel qui héritent de la table Utilisateur, elles utilisent comme clés primaires, la clé étrangère pointant vers la table de la superclasse.



## Modèle physique de données MPD

### Exemple de requêtes :

- Affichage des recettes des pizzas

```
select * from produit
join memorecette
on produit.produit_id=memorecette.produit_id
```

Résultats   Plans d'exécution

Lignes 5; select \* from produit join memorecette on produit.produit\_id=memorecette.produit\_id   Lignes: 1, Colonnes: 0

Résultats   Méta-données   Info   Prévisualisation / Graphiques   Table inversée   Résultats sous forme de texte

	produit_id	nom	disponibilite	recette_id	ingredients	quantite	temps_cuisson	temperature_fo...	produit_id
1		pizza 4 saisons	true	1	sauce tomate; mozzarella; champignons; poivrons; coeur ...	0.1;0.05;0.1;0.1;0.2	00:10:00	170	1
2		pizza du chef	false	2	sauce tomate; gorgonzola; champignons; viande hachee	0.1;0.15;0.15;0.2	00:07:00	180	2
3		pizza norvegienne	true	3	sauce tomate; mozzarella; saumon; crème fraiche	0.1;0.2;0.25;0.25;0.25	00:08:00	185	3
4		pizza hawaïenne	true	4	sauce tomate; mozzarella; coeur artichaut; ananas; crem...	0.1;0.2;0.15;0.2;0.2	00:10:00	175	4
5		pizza norvegienne	false	5	sauce tomate; mozzarella; anchois; capres	0.1;0.2;0.1;0.1	00:08:00	180	5

- Affichage de l'historique d'une commande depuis la création jusqu'à la finalisation avec les différents statuts et utilisateurs:

```

select historiquecommande.statuts,historiquecommande.timestamp,utilisateur.user_id,role.type_role
from historiquecommande
join utilisateur
on historiquecommande.numero_user=utilisateur.numero_user
join role
on utilisateur.role_id=role.role_id
where numero_commande=1
order by historiquecommande.timestamp

```

statuts	timestamp	user_id	type_role
EFFECTUEE	2019-03-19 18:45:00.0	dpatrick	CLIENT
VALIDEE	2019-03-19 18:49:00.0	dmichel	VENDEUR
PREPAREE	2019-03-19 19:00:00.0	rherve	CUISINIER
EN_COURS_LIVRAISON	2019-03-19 19:20:00.0	smichel	LIVREUR
LIVRAISON_VALIDÉE	2019-03-19 19:30:00.0	smichel	LIVREUR
FINALISEE	2019-03-19 19:35:00.0	smichel	LIVREUR

- Affichage des détails d'une commande:

```

select commande.statut, commande.montanttotal,produit.nom as pizza,commande.date,commande.heure, commande.modepaiement,commande.modelivraisonretrait,infoetablissement.nom as nom_pizzeria,
utilisateur.user_id, client.adresse
from commande
join utilisateur
on commande.client_id=utilisateur.numero_user
join client
on commande.client_id=client.numero_user
join infoetablissement
on commande.etablissement_id=infoetablissement.etablissement_id
join ajoutproduit
on commande.numero_commande=ajoutproduit.numero_commande
join produit
on ajoutproduit.produit_id=produit.produit_id
where commande.numero_commande=2

```

statut	montanttotal	pizza	date	heure	modepaiement	modelivraisonretrait	nom_pizzeria	user_id	adresse
EN_COURS_LIVRAISON	12	pizza hawaïenne	2019-03-19	19:45:00	LIVRAISON	LIVRAISON	republique	dpatrick	10 rue Pasteur, Paris