

## OC PIZZA

# CENTRALISER LE SYSTEME INFORMATIQUE POUR L'ENSEMBLE DES RESTAURANTS

avec  
**OC SOFT**

Dossier de conception technique

Version 2.0

**Auteur**  
Jean-Maxime GUTH  
*Développeur*

## TABLE DES MATIERES

<b>1 -Versions .....</b>	<b>3</b>
<b>2 -Glossaire .....</b>	<b>4</b>
<b>3 -Introduction .....</b>	<b>5</b>
3.1 -Objet du document .....	5
3.2 -Références.....	5
3.3 -Résumé de la solution fonctionnelle.....	6
<b>4 -Architecture technique .....</b>	<b>7</b>
4.1 -Diagramme de composants.....	7
4.1.1 -Approche du diagramme .....	7
4.1.2 -Diagramme de composants.....	8
4.1.3 -Nomenclature des composants .....	9
<b>5 -Architecture de déploiement.....</b>	<b>10</b>
5.1 -Diagramme de déploiement.....	10
<b>6 -Architecture logicielle .....</b>	<b>11</b>
6.1 -Diagramme de classe .....	11
6.1.1 -Approche du diagramme .....	11
6.1.2 -Diagramme de classe .....	12
6.1.3 -Aide à la compréhension.....	13
6.2 -Modèle physique de données .....	21
<b>7 -Notes et remarques.....</b>	<b>23</b>

# 1 - VERSIONS

Auteur	Date	Description	Version
JM GUTH	16/03/2021	Création du document	1.0
JM GUTH	23/12/2021	Modification d'hébergeur (OVH pour DigitalOcean) + Modification du diagramme des classes : <ul style="list-style-type: none"><li>- Stock_min et stock_max permutés dans classe item au lieu de stock</li></ul> + Mise en page corrections	2.0

## 2 - GLOSSAIRE

<b>UML</b>	Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) est un langage visuel utilisant différents diagrammes pour modéliser la conception et le fonctionnement de logiciels informatiques. C'est cette méthodologie que nous à travers ce dossier.
<b>Composant</b>	En UML, le terme « composant » fait référence à un module de classes qui représente des systèmes ou des sous-systèmes indépendants ayant la capacité de s'interfacer avec le reste du système.
<b>Classe</b>	En programmation orientée objet, une classe représente donc une catégorie d'objets. On peut considérer que c'est une sorte de moule.
<b>Méthode de classe</b>	En programmation orientée objet, une méthode est une routine membre d'une classe. C'est une fonction.
<b>TCP/IP</b>	Le protocole TCP/IP (Transmission Control Protocol/Internet Protocol) réunit les deux protocoles TCP et IP. Il s'agit donc d'une suite de protocoles associée au domaine d'Internet pour lequel elle facilite le transfert de données.
<b>HTTPS</b>	(Hyper Text Transfer Protocol Secure) Protocole de transmission permettant à l'utilisateur d'accéder à des pages web par l'intermédiaire d'un navigateur.
<b>Agrégation</b>	L'agrégation permet de définir une entité comme étant liée à plusieurs entités de classe différentes. C'est une relation particulière qui attribue à l'une des classes le rôle d'agrégat et à l'autre classe le rôle d'agrégé. L'agrégation peut être assimilée à une appartenance - faible -.
<b>Composition</b>	Il s'agit d'une appartenance forte. La vie de l'objet composant est liée à celle de son composé.

## 3 - INTRODUCTION

### 3.1 - Objet du document

Le présent document constitue le dossier de conception technique de l'application **OCSOFT**.

Objectif du document :

Ce document a pour vocation de modéliser toute l'architecture technique de l'application **OCSOFT**, d'identifier les différents composants nécessaires et leur déploiement. La méthodologie utilisée est UML.

### 3.2 - Références

Ce document découle du précédent document « **dossier de conception fonctionnelle OCSOFT V1.0** » validé par le client « **OC PIZZA** » le 22/12/2020 et remis à jour le 23/12/2021 (V2)

Il a été complété par :

- Le **dossier de conception technique** crée le 16/03/2021 (V1) et remis à jour le 23/12/2021 (V2)
- Le **PV de livraison** établi le 31/12/2021

### 3.3 - Résumé de la solution fonctionnelle retenue.

Par suite du rendez-vous du 22/12/2020 avec le client « OC PIZZA » pour la présentation de la conception fonctionnelle de l'application OCSOFT, la solution du dossier V1.0 a été retenue et validé par le client.

En voici un résumé dans les grandes lignes :

L'application comportera 3 interfaces :

#### INTERFACE CLIENT

Site web marchand qui permet :

- Créer un compte.
- Passer des commandes.
- Utiliser les services clients

#### INTERFACE COLLABORATEUR (En restaurant)

ERP qui permet de :

- Enregistrer et gérer les commandes de leur création à leur livraison.

#### INTERFACE ADMINISTRATION (Pour la direction)

ERP qui permet de :

- Gérer et suivre tout le système en temps réel.
- Gérer, les achats, la comptabilité, la

- Les clients pourront passer des commandes, soit en ligne, soit par téléphone, soit sur place.
- Il sera possible pour les clients de suivre l'avancement de leur commande en temps réel.
- Il sera possible pour les clients de géolocaliser les livreurs sur une carte.
- L'application affichera les recettes des pizzas pour les pizzaiolos.
- L'application mettra à jour le stock automatiquement en temps réel et empêchera la possibilité de commander des produits si le stock relatif n'est pas disponible.
- L'application permettra à l'administration de suivre l'activité en temps réel de chaque application.

#### Partie technique :

- La partie Backend de l'application sera codé en Python3 avec le Framework Django.
- La partie Frontend de l'application sera codé en HTML/CSS/JS.
- La BDD sera sur un serveur PostgreSQL.
- Le serveur d'hébergement sera chez DigitalOcean.

## 4 - ARCHITECTURE TECHNIQUE

### 4.1 - Diagramme de composants

#### 4.1.1 - Approche du diagramme

Ce diagramme de composants (*en page 7*) identifie les différents composants logiciels et leurs relations entre eux.






Le diagramme se décompose en trois paquets correspondant au **diagramme de paquetage du dossier de conception fonctionnel** :

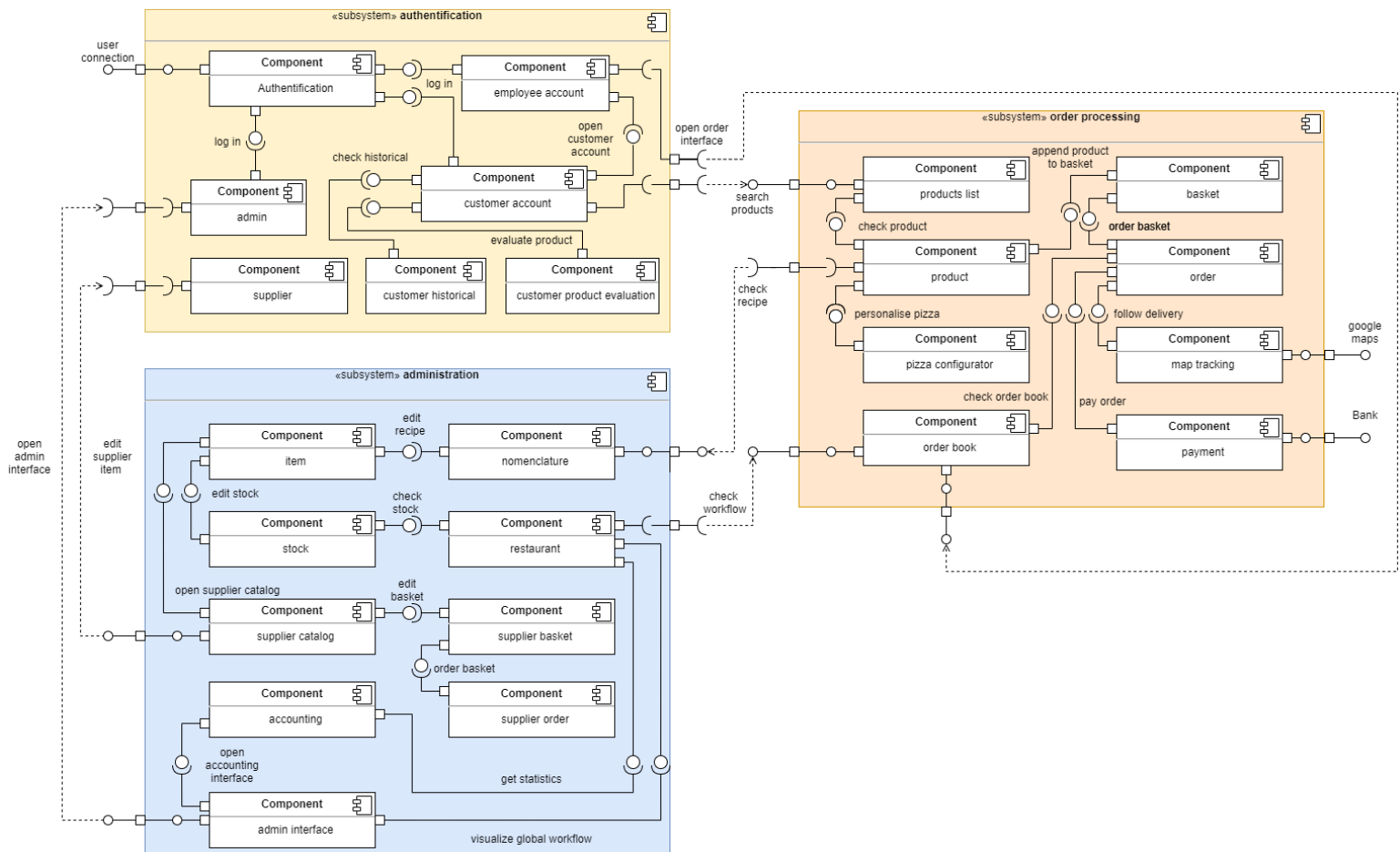
Note : le code couleur utilisé est le suivant :

Authentification	Gestion de commandes	Administration
<ul style="list-style-type: none"><li>• Gère la connexion des utilisateurs clients ou collaborateurs.</li><li>• Gère les accès aux données des compte clients et des historiques.</li><li>• Gère les comptes fournisseurs.</li></ul>	<ul style="list-style-type: none"><li>• Gère tout le système de suivi de commande de la création à la clôture des commandes.</li><li>• Gère la liste des produits.</li><li>• Inclus un configurateur pour personnaliser les pizzas à la demande.</li><li>• Est connecté à Google Maps pour le suivi des livreurs.</li><li>• Est connecté au système bancaire pour la gestion des paiements.</li></ul>	<ul style="list-style-type: none"><li>• Gère les achats.</li><li>• Gère la comptabilité et statistique.</li><li>• Gère les stocks.</li><li>• Gère le suivi des restaurants.</li><li>• Gère les recettes des pizzas</li></ul>

#### 4.1.2 - Diagramme de composants :

##### Légende

-  Port
-  interface émise
-  Connecteur d'assemblage
-  interface requise
-  dépendance





#### 4.1.3 - Nomenclature des composants :

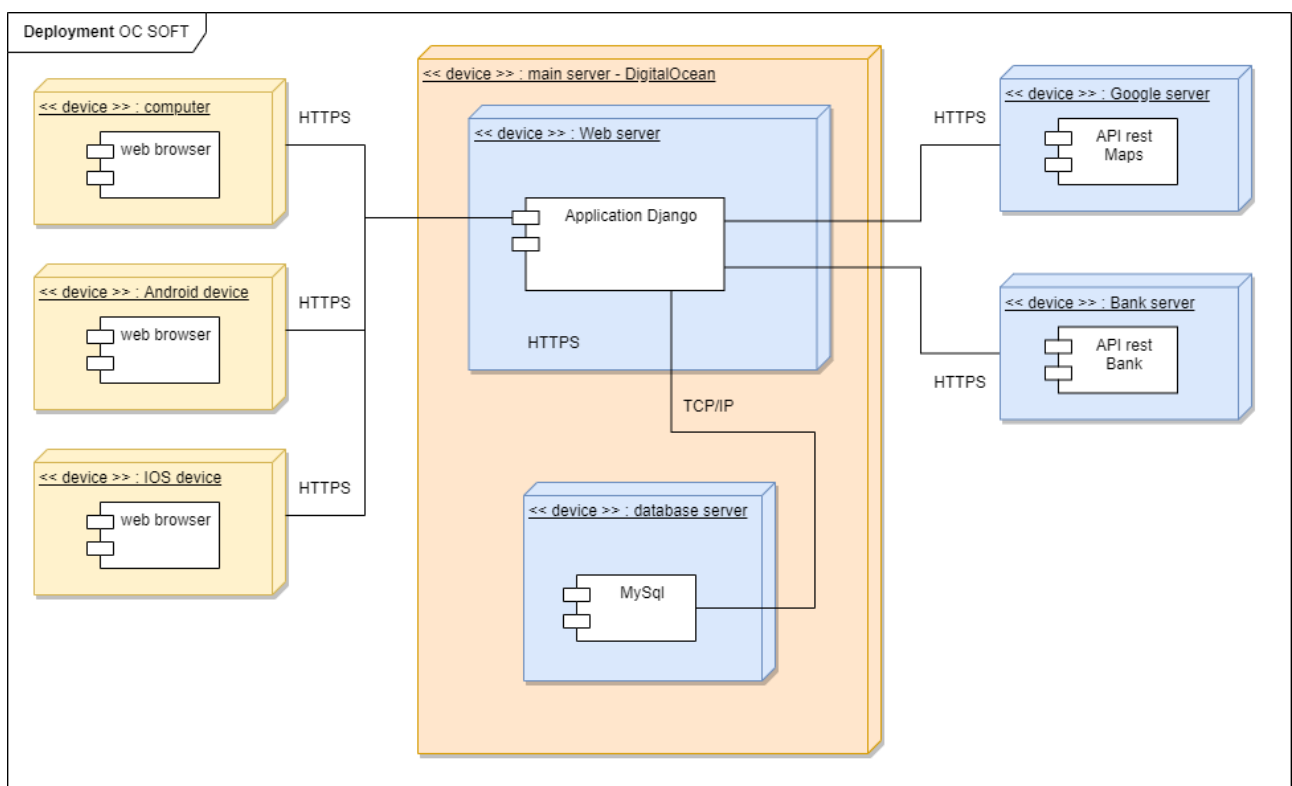
Composant	Description
Authentification	Composant qui permet de s'identifier avec login et mot de passe.
Customer account	Composant qui gère et affiche le compte du client.
Employee account	Composant qui gère et affiche le compte d'un employé.
Admin	Composant qui gère et affiche le compte d'un administrateur.
Supplier	Composant qui gère et affiche le compte d'un fournisseur.
Customer historical	Composant qui gère et affiche l'historique d'un client.
Customer product evaluation	Composant qui permet à un client de pouvoir écrire et laisser un avis sur un produit.
Product	Composant qui gère et affiche les produits destinés à la vente.
Products list	Composant qui affiche l'ensemble des produits
Pizza configurator	Composant qui permet de personnaliser une pizza
Order	Composant qui gère et affiche les commandes client
Order book	Composant qui affiche l'ensemble des commandes client
Basket	Composant qui permet à un client de constituer un panier afin de passer une commande.
Map tracking	Composant qui utilise les données de Google Maps pour pister les livreurs en temps réel.
Payment	Composant qui gère le paiement des commandes et traite avec la banque.
Item	Composant qui gère les articles. (A la différence des produits, les articles sont destinés à être achetés, stockés et composés entre eux pour donner des produits qui seront vendu)
Nomenclature	Composant qui gère la composition des articles pour donner un produit. C'est la ou sont édité les recettes de pizzas, plusieurs articles « ingrédients » donneront un produit « pizza ». Dans le cas d'une boisson (une canette), un seul article donnera un produit « canette »
Stock	Composant qui gère et affiche le stock des articles.
Restaurant	Composant qui gère le restaurant et permet d'y voir son activité.
Supplier catalog	Composant qui gère le catalogue des articles dédiés a un fournisseur. Un article peut également se trouver dans plusieurs catalogues dans le cas ou on peut l'acheter à plusieurs fournisseurs.
Supplier basket	Composant qui permet à un admin de constituer un panier afin de passer une commande fournisseur
Supplier order	Composant qui gère et affiche les commandes fournisseurs
Accounting	Composant qui gère et affiche la comptabilité et les statistiques
Admin interface	Composant qui affiche une fenêtre administrateur et permet à l'admin d'accéder à tous les composants associés.

## 5 - ARCHITECTURE DE DEPLOIEMENT

### 5.1 - Diagramme de déploiement

Ce diagramme modélise le déploiement des composants.

- Serveur principal d'hébergement : DigitalOcean
- Les utilisateurs interagissent avec un serveur web http
- Application codée en python avec le Framework Django.
- Serveur de base de données SGBDR MySQL
- Transactions bancaires par l'API REST de la banque
- Suivi des livreurs par L'API REST de Google Maps



## 6 - ARCHITECTURE LOGICIELLE

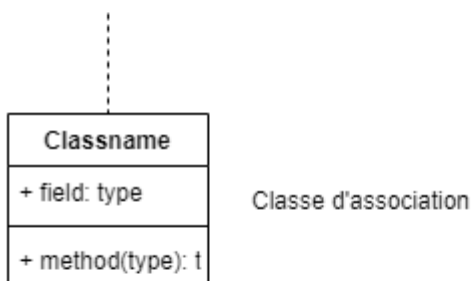
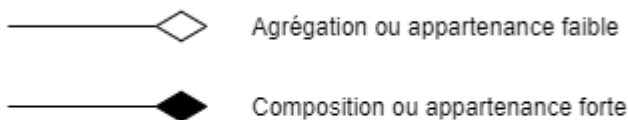
### 6.1 - Diagramme de classes

#### 6.1.1 - Approche du diagramme

Ce diagramme (*en page 10*) représente les différentes classes et méthodes liées de l'application.

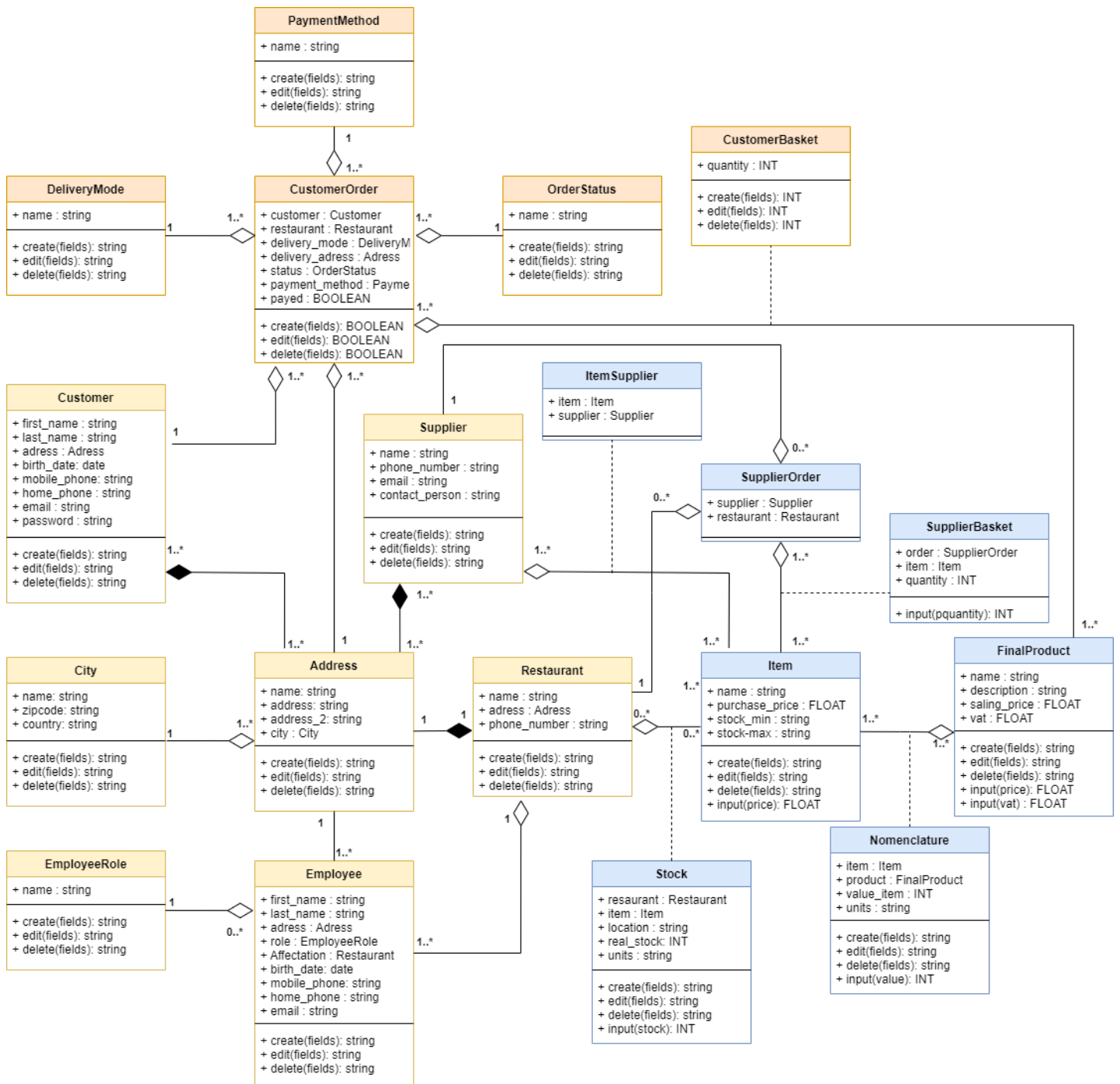
	Classes liées à l'authentification
	Classes liées à la gestion de commande
	Classes liées à l'administration

#### LEGENDE



- 1 multiplicité d'association : exactement 1
- 1..\* multiplicité d'association : au moins 1
- 0..\* multiplicité d'association : aucune ou plusieurs

## 6.1.2 - Diagramme de classe :



### 6.1.3 - Aide à la compréhension du diagramme de classe :

Pour mieux interpréter ce diagramme, ce dossier propose une aide à la compréhension pour chaque classe. Les informations relatives à chaque classe sont disponibles selon l'index ci-dessous :

Index :

N°	Nom de la classe	Page
1	Customer	p.14
2	Address	p.14
3	City	p.15
4	Employee	p.15
5	EmployeeRole	p.15
6	Supplier	p.16
7	Restaurant	p.16
8	OrderStatus	p.16
9	DeliveryMode	p.16
10	PaymentMethod	p.17
11	CustomerOrder	p.17
12	CustomerBasket	p.17
13	FinalProduct	p.18
14	Item	p.18
15	Nomenclature	p.19
16	ItemSupplier	p.19
17	SupplierOrder	p.19
18	SupplierBasket	p.20
19	Stock	p.20

1)

Classe	Rôle	Agrégats ou composite de :	Agrégée ou composante de :
Customer	<i>Classe qui permet de créer et éditer des comptes clients. Elle prend en argument :</i> <ul style="list-style-type: none"> <li>Nom</li> <li>Prénom</li> <li>Adresse (classe composante « Address »)</li> <li>Date de naissance</li> <li>Téléphone mobile</li> <li>Téléphone fixe</li> <li>Courriel</li> <li>Mot de passe</li> </ul>	<div>Adress</div>	<div>CustomerOrder</div>

2)

Classe	Rôle	Agrégats ou composite de :	Agrégée ou composante de :
Address	<i>Classe qui permet de créer et éditer des adresses. Elle prend en argument :</i> <ul style="list-style-type: none"> <li>Nom</li> <li>Adresse</li> <li>Ville (classe composante « city »)</li> </ul>	<div>City</div>	<div>CustomerOrder</div> <div>Employee</div> <div>Customer</div> <div>Restaurant</div> <div>Supplier</div>

3)

Classe	Rôle	Agrégats ou composite de :	Agrégée ou composante de :
City	<i>Classe qui permet de créer et éditer des villes Elle prend en argument :</i> <ul style="list-style-type: none"> <li>Nom</li> <li>Code postal</li> <li>Pays</li> </ul>		<div>Address</div>

4)

Classe	Rôle	Agrégats ou composite de :	Agrégée ou composante de :
Employee	<i>Classe qui permet de créer et éditer des comptes employés. Elle prend en argument :</i> <ul style="list-style-type: none"> <li>Nom</li> <li>Prénom</li> <li>Adresse (classe composante « Address »)</li> <li>Rôle (classe composante « EmployeeRole »)</li> <li>Affectation (classe composante « Restaurant »)</li> <li>Date de naissance</li> <li>Téléphone mobile</li> <li>Téléphone fixe</li> <li>Courriel</li> </ul>	<div>EmployeeRole</div> <div>Restaurant</div> <div>Address</div>	

5)

Classe	Rôle	Agrégats ou composite de :	Agrégée ou composante de :
EmployeeRole	<i>Classe qui permet de créer et éditer des rôles d'employé (Manager, pizzaiolo, livreur etc...) Elle prend en argument :</i> <ul style="list-style-type: none"> <li>Nom</li> </ul>		<div>Employee</div>

6)

Classe	Rôle	Agrégats ou composite de :	Agrégée ou composante de :
Supplier	<i>Classe qui permet de créer et éditer des comptes fournisseurs. Elle prend en argument :</i> <ul style="list-style-type: none"> <li>Nom</li> <li>Adresse (classe composante « Address »)</li> <li>Numéro de téléphone</li> <li>Contact</li> </ul>	Address	
Classe associée avec <span>Item</span> via <span>ItemSupplier</span>			

7)

Classe	Rôle	Agrégats ou composite de :	Agrégée ou composante de :
Restaurant	<i>Classe qui permet de créer et éditer les comptes des restaurants. Elle prend en argument :</i> <ul style="list-style-type: none"> <li>Nom</li> <li>Adresse (classe composante « Address »)</li> <li>Numéro de téléphone</li> </ul>	Address	

8)

Classe	Rôle	Agrégats ou composite de :	Agrégée ou composante de :
OrderStatus	<i>Classe qui permet de créer et éditer les statuts des commande (en préparation, en livraison etc..). Elle prend en argument :</i> <ul style="list-style-type: none"> <li>Nom</li> </ul>		CustomerOrder

9)

Classe	Rôle	Agrégats ou composite de :	Agrégée ou composante de :
DeliveryMode	<i>Classe qui permet de créer et éditer les types de livraison pour une commande (à emporter, à livrer). Elle prend en argument :</i> <ul style="list-style-type: none"> <li>Nom</li> </ul>		CustomerOrde



10)

Classe	Rôle	Agrégats ou composite de :	Agrégée ou composante de :
PaymentMethod	<i>Classe qui permet de créer et éditer les types de paiement pour une commande (espèces caisse, CB caisse, CB livreur, CB en ligne etc....) Elle prend en argument :</i> <ul style="list-style-type: none"> <li>Nom</li> </ul>		<div>CustomerOrder</div>

11)

Classe	Rôle	Agrégats ou composite de :	Agrégée ou composante de :
CustomerOrder	<i>Classe qui permet de créer des commandes clients Elle prend en argument :</i> <ul style="list-style-type: none"> <li>Client (classe composante « customer »)</li> <li>Restaurant (classe composante « restaurant »)</li> <li>Mode de livraison (classe composante « DeliveryMode »)</li> <li>Adresse de livraison (classe composante « Adress »)</li> <li>Méthode de paiement (classe composante « PaymentMethod »)</li> <li>Payé</li> </ul>	<div>Customer</div> <div>Restaurant</div> <div>Address</div> <div>DeliveryMode</div> <div>PaymentMethod</div>	
<i>Classe associée avec</i> <div>FinalProduct</div> <i>via</i> <div>CustomerBasket</div>			

12)

Classe	Rôle	Classe d'association entre :
CustomerBasket	<i>Classe d'association qui permet de rattacher des produits à une commande. Elle prend en argument :</i> <ul style="list-style-type: none"> <li>Client (classe composante « customer »)</li> <li>Produit (classe composante « FinalProduct »)</li> <li>quantité</li> </ul>	<div>CustomerOrder</div> <div>FinalProduct</div>

13)

Classe	Rôle
FinalProduct	<i>Classe qui permet de créer ou d'éditer des produits destinés à la vente. Elle prend en argument :</i> <ul style="list-style-type: none"> <li>Nom</li> <li>Description</li> <li>Prix de vente</li> <li>TVA</li> </ul>
Classe associée avec	<div>CustomerOrder</div> <div>via</div> <div>CustomerBasket</div>
Et avec	<div>via</div> <div>Item</div> <div>Nomenclature</div>

14)

Classe	Rôle
Item	<i>Classe qui permet de créer ou d'éditer des articles. Elle prend en argument :</i> <ul style="list-style-type: none"> <li>Nom</li> <li>Description</li> <li>Prix de vente</li> <li>TVA</li> <li>Stock mini</li> <li>Stock maxi</li> </ul>
Classe associée avec	<div>Restaurant</div> <div>via</div> <div>Stock</div>
Également avec	<div>FinalProduct</div> <div>via</div> <div>Nomenclature</div>
Également avec	<div>SupplierOrder</div> <div>via</div> <div>SupplierBasket</div>
Également avec	<div>Supplier</div> <div>via</div> <div>ItemSupplier</div>

15)

Classe	Rôle	Classe d'association entre :
Nomenclature	<i>Classe d'association qui permet de rattacher un ou plusieurs articles (ex : des ingrédients) à un produit fini (ex : une pizza). Elle prend en argument :</i> <ul style="list-style-type: none"> <li>Article (classe composante « Item »)</li> <li>Produit (classe composante « FinalProduct »)</li> <li>Quantité</li> <li>Unité</li> </ul>	<div>Item</div> <div>FinalProduct</div>

16)

Classe	Rôle	Classe d'association entre :
ItemSupplier	<i>Classe d'association qui permet de rattacher un ou plusieurs articles à un ou plusieurs fournisseurs. Elle prend en argument :</i> <ul style="list-style-type: none"> <li>Article (classe composante « Item »)</li> <li>Fournisseur (classe composante « Supplier »)</li> </ul>	<div>Item</div> <div>Supplier</div>

17)

Classe	Rôle	Agrégats ou composite de :
SupplierOrder	<i>Classe qui permet de créer, éditer une commande fournisseur. Elle s'apparente à une classe d'association. Elle prend en argument :</i> <ul style="list-style-type: none"> <li>Fournisseur (classe composante « Supplier »)</li> <li>Restaurant (classe composante « Restaurant »)</li> </ul>	<div>Supplier</div> <div>Restaurant</div>
<i>Classe associée avec</i> <div>Item</div> <i>via</i> <div>SupplierBasket</div>		

18)

Classe	Rôle	Classe d'association entre :
SupplierBasket	<p>Classe d'association qui permet de rattacher un ou plusieurs articles à une ou plusieurs commande fournisseur. Elle prend en argument</p> <ul style="list-style-type: none"> <li>Article (classe composante « Item »)</li> <li>Commande fournisseur (classe composante « SupplierOrder »)</li> <li>Quantité</li> </ul>	<div>Item</div> <div>SupplierOrder</div>

19)




Classe	Rôle	Classe d'association entre :
Stock	<p>Classe d'association qui permet de créer, éditer une commande fournisseur. Elle prend en argument :</p> <ul style="list-style-type: none"> <li>Restaurant (classe composante « Restaurant »)</li> <li>Article (classe composante « Item »)</li> <li>Emplacement</li> <li>Stock réel</li> <li>Unité de stock</li> </ul>	<div>Item</div> <div>Restaurant</div>

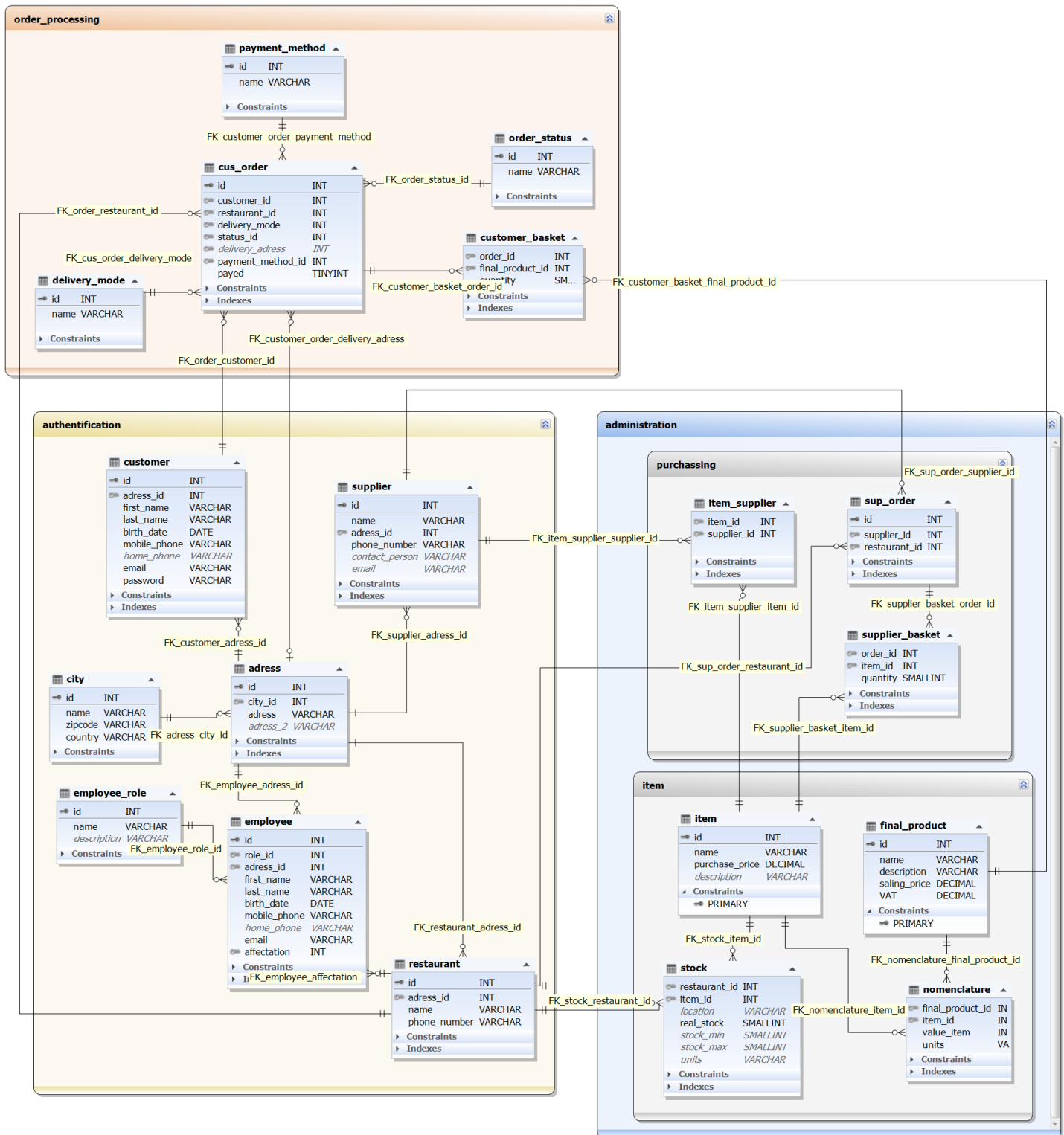
## 6.2 - Modèle physique de données

Le modèle physique de données (MPD) ou modèle relationnel de la base de données, décrit l'architecture de la base de données PostgreSQL et les différentes relations entre les tables. Ce diagramme est très semblable à celui du diagramme de classes.

(Diagramme en p. 22)

Exemple de table :

		<b>customer</b>	
Clé primaire (PK) →		<b>id</b>	INT
Clé étrangère (FK) →		<b>adress_id</b>	INT
		first_name	VARCHAR
		last_name	VARCHAR
		birth_date	DATE
Caractère noir = NOT NULL →		<b>mobile_phone</b>	VARCHAR
Caractère gris = NULL →		<i>home_phone</i>	<i>VARCHAR</i>
		email	VARCHAR
		password	VARCHAR
		► <b>Constraints</b>	
		► <b>Indexes</b>	



## 7 - NOTES ET REMARQUES

Sujet	Remarques / observations	Page	Auteur

Sujet	Remarques / observations	Page	Auteur