

# OC Pizza

---



## Spécifications techniques



## Version du document :

Date	Auteur	Description	Version
21/05/19	Sandrine	Initial document	V1

## Objet du document :

Ce document permet de dégager les règles de gestion fonctionnelles du projet “système de gestion de pizzas” pour la société OC Pizza, il comprend :

- les différents acteurs interagissant avec le système proposé
- le descriptif des fonctionnalités
- les règles de gestion fonctionnelles
- le cycle de vie des commandes

## Contexte :

La société OC Pizza est un jeune groupe de pizzerias 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. OC Pizza a déjà fait une petite prospection et les logiciels existants qu'il a pu trouver ne lui conviennent pas.

## Objectifs :

L'application que nous proposons permet d'être plus efficace dans la gestion des commandes, de leur réception à leur livraison en passant par leur préparation.

Notre outil permet :

- de suivre en temps réel les commandes passées et en préparation ;
- de suivre en temps réel le stock d'ingrédients restants pour permettre l'actualisation des cartes à pizzas pour n'afficher que celles qui sont encore réalisables ;
- de proposer 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
- de proposer un aide mémoire aux pizzaiolos indiquant la recette de chaque pizza

## **Acteurs/Rôles et fonctionnalités attendues :**

### **Acteurs / Rôles :**

- Le/La responsable de toutes les pizzerias
- Le/La manager d'une pizzeria
- Le/La pizzaiolo
- Le/La Standardiste d'une pizzeria
- Le/La livreur
- Le client (ou client potentiel)

### **Les fonctionnalités attendues côté client :**

- Passer une commande
- Créer un compte "client"
- S'authentifier
- Gérer son compte "client"
- Modifier le statut d'une commande

### **Les fonctionnalités attendues côté personnel :**

- S'authentifier :
- Passer une commande
- Modifier le statut d'une commande

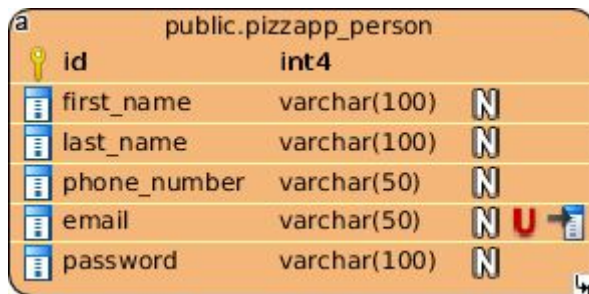
### **Les fonctionnalités attendues côté administration :**

- S'authentifier
- Gérer son personnel
- Afficher les statistiques
- Gérer la carte à pizza
- .Gérer le stock

# Specifications techniques :

## Modélisation du domaine fonctionnel :

- **Person**



The diagram shows a table named 'public.pizzapp\_person' with the following columns: 'id' (int4, primary key), 'first\_name' (varchar(100), nullable), 'last\_name' (varchar(100), nullable), 'phone\_number' (varchar(50), nullable), 'email' (varchar(50), nullable, unique), and 'password' (varchar(100), nullable). Each column has a small icon representing its data type or constraints.

public.pizzapp_person		
id	int4	
first_name	varchar(100)	N
last_name	varchar(100)	N
phone_number	varchar(50)	N
email	varchar(50)	N U
password	varchar(100)	N

Tout acteur est représenté par une personne. Ce modèle est composé d'un champ

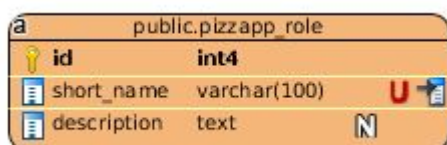
- ❖ id (clé primaire),
- ❖ nom, pouvant être null, limité à 100 caractères,
- ❖ prénom, pouvant être null, limité à 100 caractères,
- ❖ téléphone, pouvant être null, limité à 50 caractères,
- ❖ email de connection, pouvant être null, limité à 50 caractères, unique
- ❖ password de connection, pouvant être null, limité à 100 caractères.

Le nom, prénom et téléphone ne sont pas obligatoire, car un client qui commande depuis l'application, et qui prend toujours ses pizzas à emporter, peut ne pas remplir ces informations.

L'email et le password sont également non obligatoire, car un client qui se déplace à la pizzeria pour commander ses pizzas, peut ne pas remplir ces informations.

Le choix de laisser les identifiants de connection sur le modèle personne, est stratégique, car dans toutes les situations, la personne peut fournir son email afin de recevoir les publicités et promotions.

- **Role**



The diagram shows a table named 'public.pizzapp\_role' with the following columns: 'id' (int4, primary key), 'short\_name' (varchar(100), nullable, unique), and 'description' (text, nullable).

public.pizzapp_role		
id	int4	
short_name	varchar(100)	N U
description	text	N

Chaque acteur de l'application sera identifié par un rôle. Ce modèle est composé d'un champ

- ❖ nom court qui est unique et non null limité à 100 caractères,

- ❖ description qui peut être nulle et sans limite de caractère.

En fonction du rôle connecté, l'application filtrera les fonctionnalités afin de ne rendre accessible que ce que l'acteur pourra effectuer comme action.

- **Employed**

public.pizzapp_employed		
id	int4	
is_active	bool	
registration_date	timestamp(35) with time zone	
inactivity_date	timestamp(35) with time zone	N
comment	text	N
person_id	int4	U
pizzeria_id	int4	

L'employé peut être un responsable administrateur, un standardiste, pizzaiolo, livreur.

Chaque employé est constitué d'un champ

- ❖ is\_active, boolean permettant de savoir si l'employé est actif dans la pizzeria, True par défaut
- ❖ date d'enregistrement, datetime avec timezone,
- ❖ date d'inactivation, datetime avec timezone
- ❖ commentaire lié à l'employé, qui peut être null et sans limite de caractère.
- ❖ personne unique,
- ❖ pizzeria dans laquelle il travail.

L'employé est également constitué d'une

- ❖ liste de rôles, ce qui permet en fonction de ses rôles d'accéder à certains fonctionnalités.

public.pizzapp_employed_roles		
id	int4	
employed_id	int4	U
role_id	int4	U

- **Customer**

public.pizzapp_customer	
id	int4
registration_date	timestamp(35) with time zone
comment	text N
person_id	int4 U

Le customer est le client, il peut commander ses pizzas par téléphone, sur place ou avec l'application. Il est composé d'une

- ❖ personne, qui est unique
- ❖ date d'enregistrement, au format datetime avec prise en charge de la timezone,
- ❖ commentaire qui peut être nul et sans limite de caractère.

## • **Address**






public.pizzapp_address	
id	int4
title	varchar(100) N
address_one	varchar(400)
address_two	varchar(400) N
zip_code	varchar(20)
city	varchar(100)
country	varchar(100) N
location	int4 N
google_place_id	text N U
loc_address	text N

L'adresse est un modèle de données qui peut être géo-localisé par l'API Google. Une adresse peut être utilisée par plusieurs objets (customer, supplier, pizzeria). Elle est composée d'un champ

- ❖ titre, limité à 100 caractère et non obligatoire,
- ❖ première adresse, correspondant au numéro et nom de la rue, obligatoire et limitée à 400 caractères,
- ❖ deuxième adresse, si complément de rue, non obligatoire et limitée à 400 caractères,
- ❖ code postal, obligatoire et limitée à 20 caractères,
- ❖ ville, obligatoire et limitée à 100 caractères,
- ❖ pays, non obligatoire et limitée à 100 caractères,
- ❖ location, qui est un champ point contenant latitude et longitude, élément fourni par l'API Google, non obligatoire
- ❖ identifiant google place, text sans limite de caractère, élément unique fourni par l'API Google, non obligatoire,

- ❖ loc adresse qui correspond à l'adresse complète envoyé à l'API Google contenant numéro de rue, rue, code postal et ville, il n'est pas obligatoire et sans limite de caractère.




- **TypeAddress**

a public.pizzapp_typeaddress	
 id	int4
 type	varchar(100)
 address_id	int4 
 customer_id	int4

Le type d'adresse est un modèle de données, qui permet d'enregistrer plusieurs adresse client avec un type différent. Ce modèle contient un champ

- ❖ type, permettant de donner un titre à l'adresse, limité à 100 caractères et obligatoire
- ❖ une adresse,
- ❖ un client.














- **Pizzeria**

a public.pizzapp_pizzeria	
 id	int4
 name	varchar(100)
 address_id	int4

La pizzeria est modèle qui contient un champ

- ❖ nom, limité à 100 caractère et obligatoire,
- ❖ adresse.

- **Supplier**

a public.pizzapp_supplier	
 id	int4
 company_name	varchar(100)
 phone_number	varchar(50) 
 email	varchar(50) 
 is_active	bool
 registration_date	timestamp(35) with time zone
 inactivity_date	timestamp(35) with time zone 
 comment	text 
 address_id	int4



Le fournisseur est le modèle contenant un champ

- ❖ nom de société limité à 100 caractère et obligatoire,
- ❖ numéro de téléphone, limité à 50 caractère et non obligatoire,
- ❖ email, limité à 50 caractère et non obligatoire,
- ❖ is\_active, boolean permettant de savoir si le fournisseur est actif, True par défaut,
- ❖ date d'enregistrement, datetime avec timezone,
- ❖ date d'inactivité, datetime avec timezone, non obligatoire,
- ❖ commentaire, texte sans limite de caractère, non obligatoire,
- ❖ adresse.

- **Contact**

public.pizzapp_contact	
id	int4
is_active	bool
registration_date	timestamp(35) with time zone
inactivity_date	date
comment	text
person_id	int4
supplier_id	int4

Le contact est le modèle correspondant aux contacts de personnes travaillant pour un fournisseur. Il contient un champ

- ❖ actif, boolean à True par défaut, permettant de savoir si le contact est actif,
- ❖ date d'enregistrement, datetime avec timezone,
- ❖ date d'inactivité, datetime avec timezone, non obligatoire,
- ❖ commentaire, texte sans limite de caractère, non obligatoire,
- ❖ adresse.
- ❖ personne unique,
- ❖ fournisseur.

- UnitOfMeasure
- Component
- ComponentPrice
- Pizza
- PizzaLine

- PizzaCard
- Status
- Order
- OrderLine
- Command
- CommandLine
- StockMovement
- PaymentMethod
- Payment