



Application "OC Pizza Gestion"

DOSSIER DE CONCEPTION TECHNIQUE
Version V1



Auteur

Karine Danjard

Développeur d'Applications



TABLE DES MATIÈRES

1 - VERSIONS.....	2
2 - INTRODUCTION.....	3
2.1 - Objet du document.....	3
2.2 - Références.....	3
Rappel des demandes du client.....	4
3 - DIAGRAMME DE CLASSES.....	5
4 - MODELE PHYSIQUE DE DONNEES (MPD).....	6
5 - DIAGRAMME DE COMPOSANTS.....	8
6 - DIAGRAMME DE DEPLOIEMENT.....	9



1 - VERSIONS

Auteur	Date	Description	Version
Karine Danjard	30/04/2020	Création du document	V1



2 - INTRODUCTION

2.1 - Objet du document :

Le présent document constitue le dossier de conception technique de l'application "OC Pizza Gestion"

Objectif du document : fixer les solutions techniques du logiciel "OC Pizza Gestion"

2.2 - Références :

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

- 1 - DCF - Dossier de conception fonctionnelle de l'application OC Pizza Gestion
- 2 - Diaporama - Projet OC Pizza Gestion

Si nous revenons à ces dossiers, nous pouvons voir que le système de base de données choisi est PostgreSQL et le serveur proposé est un serveur Tomcat Java JEE.



Rappel des demandes du client :

- être plus efficace dans la gestion de commandes, de leur réception à leur livraison en passant par la préparation,
- suivre en temps réel les commandes passées, en préparation et en livraison,
- suivre en temps réel le stock d'ingrédients restants pour savoir quelles pizzas peuvent être encore réalisés,
- 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,
 - ◆ proposer un aide mémoire aux pizzaiolos indiquant la recette de chaque pizza.



Le diagramme de classes ci-dessous va nous permettre d'établir les différentes classes ainsi que les différentes relations entre celles-ci.

Visual Paradigm Online Diagrams Express Edition

[illegible]

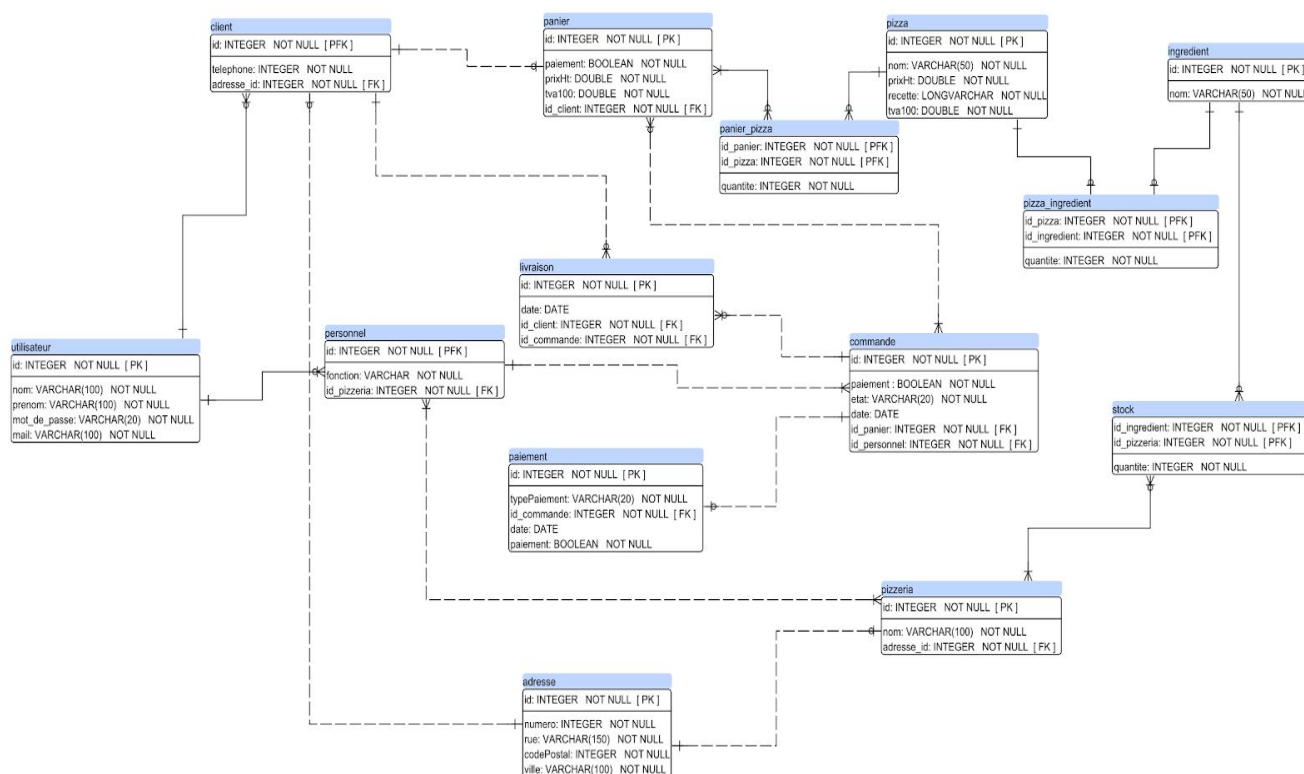
Nous pouvons voir dans ce diagramme les différentes classes dont nous avons besoin pour ce projet ainsi que les différents attributs prévus. Nous pouvons voir également les multiplicités entre elles.



4 - MODELE PHYSIQUE DE DONNEES (MPD)

A partir du diagramme de classes ci-dessus, un Modèle Physique de Données a pu être réalisé.

Ce dernier nous permettra d'implémenter les données dans la base de données.



Nous retrouvons ici toutes les tables ainsi que leurs relations et attributs. Ensuite avec ce Modèle Physique de Données, nous pourrons générer un script qui nous permettra de l'intégrer à la base de données PostgreSQL.(Réalisé dans ce projet avec SQL Power Architect).



Extrait du script créé à partir du MPD (Script entier fourni en annexe) :

```
CREATE SEQUENCE public.adresse_id_seq;
```

```
CREATE TABLE public.adresse (  
    id INTEGER NOT NULL DEFAULT nextval('public.adresse_id_seq'),  
    numero INTEGER NOT NULL,  
    rue VARCHAR(150) NOT NULL,  
    codePostal INTEGER NOT NULL,  
    ville VARCHAR(100) NOT NULL,  
    CONSTRAINT adresse_pk PRIMARY KEY (id)  
);
```

```
ALTER SEQUENCE public.adresse_id_seq OWNED BY public.adresse.id;
```

```
CREATE SEQUENCE public.pizzeria_id_seq;
```

```
CREATE TABLE public.pizzeria (  
    id INTEGER NOT NULL DEFAULT nextval('public.pizzeria_id_seq'),  
    nom VARCHAR(100) NOT NULL,  
    adresse_id INTEGER NOT NULL,  
    CONSTRAINT pizzeria_pk PRIMARY KEY (id)  
);
```

.....

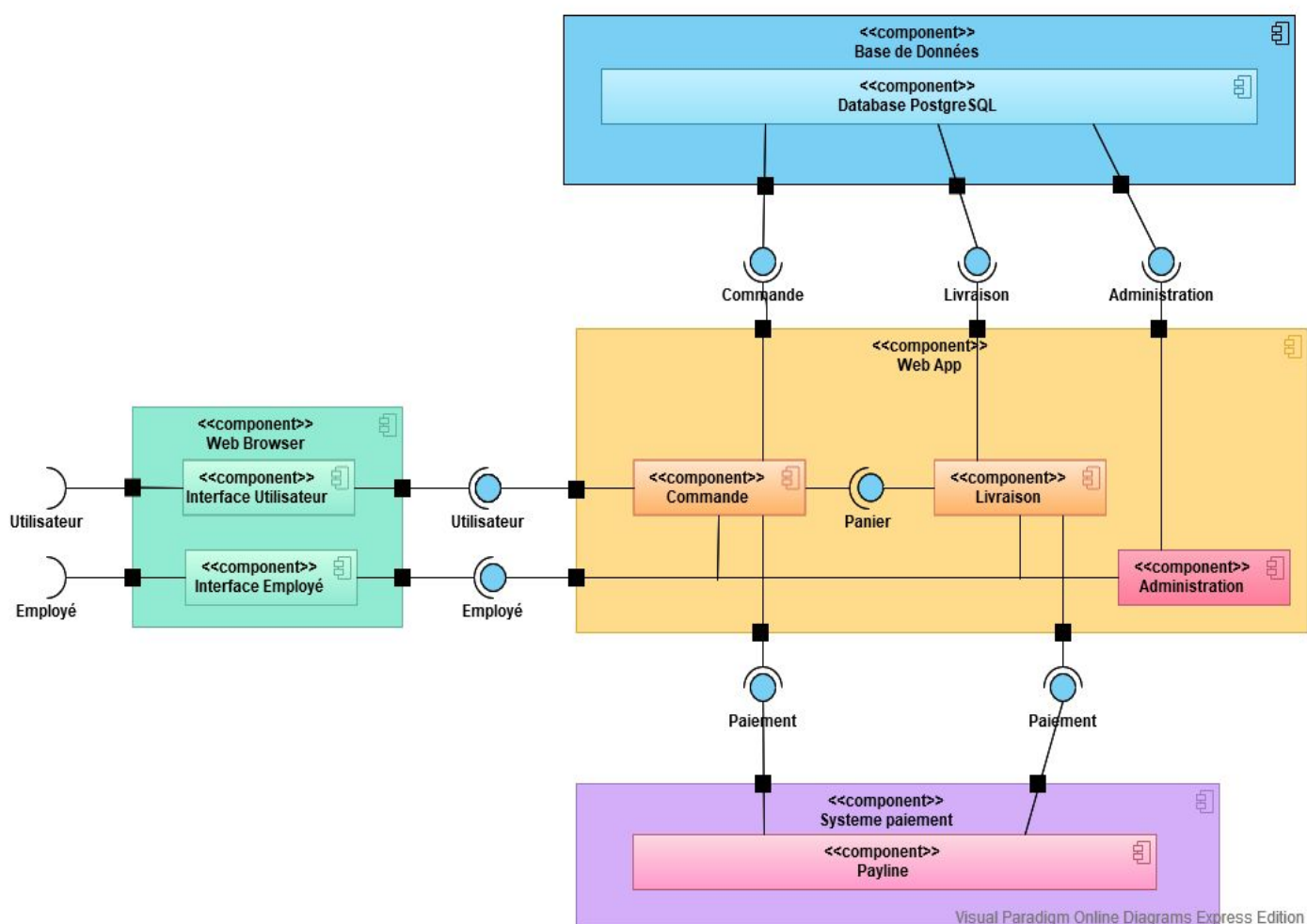


5 - DIAGRAMME DE COMPOSANTS :

Ce diagramme va nous permettre de mettre en évidence les dépendances entre les composants (qui utilise quoi).

Visual Paradigm Online Diagrams Express Edition

DIAGRAMME DE COMPOSANTS





6 - DIAGRAMME DE DEPLOIEMENT :

Le diagramme de déploiement va nous permettre de décrire le déploiement physique des informations générées par l'application sur des composants matériels.

Visual Paradigm Online Diagrams Express Edition

DIAGRAMME DE DEPLOIEMENT

