|  |
| --- |
| {{Client}}  {{Projet}}  Dossier de conception technique  Version {{Version}} |
| **Auteur**  {{Auteur}}  {{Auteur\_Role}} |

A Faire :

* Remplacer manuellement les valeurs entre <>
* Les valeurs entre {{}} sont à renseigner dans les propriétés du document

Table des matières

1 - Versions 4

2 - Introduction 5

2.1 - Objet du document 5

3 - Architecture Technique 6

3.1 - Composants généraux 6

1.1.1 - Commande 6

1.1.2 - Employe 7

1.1.3 - Compte 7

1.1.4 - Client 7

1.1.5 - Adresse 7

1.1.6 - Pizza 7

1.1.7 - Panier 7

1.1.8 - Ingredient 8

1.1.9 - Recette 8

1.1.10 - Restaurant 8

1.1.11 - Stock 8

3.2 - Application Web 9

3.2.1 - Composant Panier 9

3.2.2 - Composant Authentification 9

3.2.3 - Composant Accueil 10

3.2.4 - Composant Commande 10

3.2.5 - Composant utilisateur 10

3.2.6 - Composant Pizza 10

3.2.7 - Composant Ingrédient 10

3.3 - Application XXX... 10

4 - Architecture de Déploiement 11

4.1 - Serveur de Base de données 11

4.2 - Serveur XXX 12

5 - Architecture logicielle 13

5.1 - Principes généraux 13

5.1.1 - Les couches 13

5.1.2 - Les modules 13

5.1.3 - Structure des sources 13

5.2 - Application Web 14

5.3 - Application Xxx 14

6 - Points particuliers 15

6.1 - Gestion des logs 15

6.2 - Fichiers de configuration 15

6.2.1 - Application web 15

6.2.1.1 - Datasources 15

6.2.1.2 - Fichier xxx.yyy 15

6.2.2 - Application Xxx 15

6.3 - Ressources 15

6.4 - Environnement de développement 15

6.5 - Procédure de packaging / livraison 15

6.6 - XXX 15

7 - Glossaire 16

# Versions

|  |  |  |  |
| --- | --- | --- | --- |
| Auteur | Date | Description | Version |
| Xxx | JJ/MM/AAAA | Création du document | XXX |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Objet du document

Le présent document constitue le dossier de conception fonctionnelle de l'application "Pizza Easy Order".

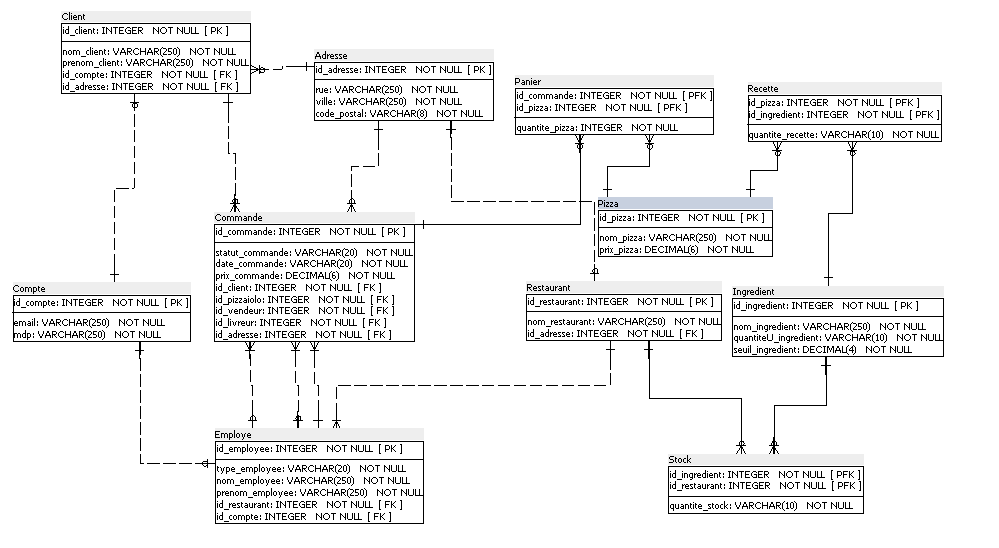
Le but du document est de présenter les fonctionnalités techniques de l'application et les différents acteurs.

Les éléments du présent dossier découlent :

* des échanges avec le client
* de l'analyse des besoins du client
* de la représentation UML découlant de l'analyse des besoins

# Architecture Technique

## Composants généraux



Modèle physique de données

### Commande

La classe commande regroupe les commandes en cours et passées.

Elle est composée de l'id, du statut de la commande, de la date de la commande, du prix de la commande et d'un ensemble de clés étrangères :

* id\_client : une commande a 1 et 1 seul client (et un client peut avoir 0 ou plusieurs commandes)
* id\_pizzaiolo : une commande a 1 et 1 seul pizzaiolo (et un pizzaiolo a 1 ou plusieurs commadnes)
* id\_vendeur : une commande a 0 ou 1 vendeur (et un vendeur a 1 ou plusieurs commandes)
* id\_livreur : une commande a 0 ou 1 livreur (et un livreur a 1 ou plusieurs commandes)
* id\_adresse : une commande a 1 et 1 seul adresse (et une adresse a 0 ou plusieurs commandes)

Commande a une relation many-to-many avec la table Pizza.

### Employe

La classe employe rassemble les informations d'un employé.

Elle est composée de l'id, du type d'employé, du nom, du prénom et de deux clés étrangères :

* id\_restaurant : un employé appartient à 1 et 1 seul restaurant (et un restaurant a 1 ou plusieurs employés)
* id\_compte : un employé a 1 et 1 seul compte (et un compte est à 0 ou 1 employé)

### Compte

La classe compte comporte les informations de connexion de chaque utilisateur.

Elle est composée de l'id, d'un email et d'un mot de passe.

### Client

La classe client contient les informations clés d'un client.

Elle est composée de l'id, du nom, du prénom et de deux clés étrangères :

* id\_compte : un client a 1 et 1 seul compte (et un compte est à 0 ou 1 client)
* id\_adresse : un client a 1 et 1 seule adresse (et une adresse est à 0 ou plusieurs clients)

### Adresse

La classe adresse comporte l'ensemble des adresses nécessaire pour la pizzeria.

Elle est composée de l'id, de la rue, de la ville et du code postale.

### Pizza

La classe pizza comporte l'ensemble des pizzas proposé par le groupe.

Elle est composée de l'id, du nom et du prix.

Pizza a une relation many-to-many avec Commande et une avec Ingredient

### Panier

La classe panier sert de relation many-to-many entre commande et pizza.

Elle est donc composée des id de ces deux classes et d'une quantité de pizza.

### Ingredient

La classe ingredient comporte l'ensemble des ingrédients nécessaire pour le groupe.

Elle est composée de l'id, du nom, de la quantité unitaire et du seuil minimal.

Ingredient a une relation many-to-many avec Pizza et Restaurant.

### Recette

La classe recette sert de relation many-to-many entre Pizza et Ingredient.

Elle est donc composée des id de ces deux classes et de la quantité d'ingrédient par recette.

### Restaurant

La classe restaurant sert à rassembler l'ensemble des établissements du groupe.

Elle est composée de l'id, du nom et d'une clé étrangère :

* id\_adresse : un restaurant a 1 et 1 seule adresse (et une adresse a 0 ou 1 restaurant)

Restaurant a une relation many-to-many avec Ingredient

### Stock

La classe stock sert de relation many-to-many entre la classe Restaurant et Ingredient.

Elle est donc composée des id de ces deux classes et d'une quantité d'ingrédient en stock par restaurant.

## Application Web

La pile logicielle est la suivante :

* Application Python(django)
* Serveur d'application Ngnix
* Base de données PostgreSQL

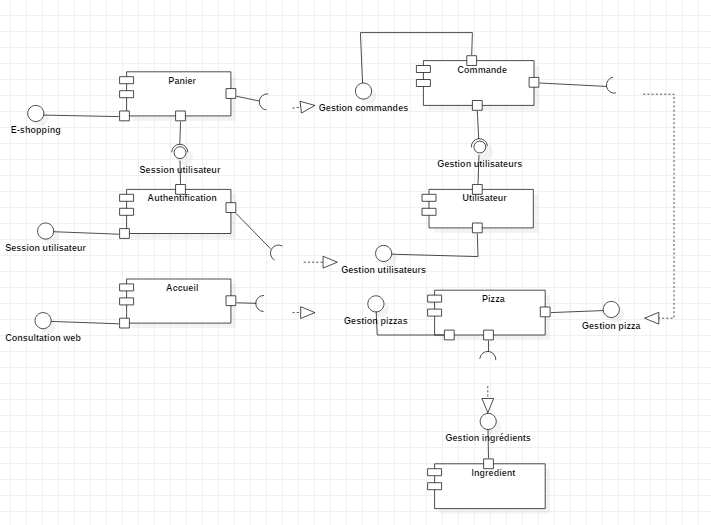


Diagramme UML de Composants

### Composant Panier

Le rôle de ce composant est de rassembler un ensemble d'information afin de préparer une commande.

### Composant Authentification

Le rôle de ce composant est de gérer les connections et inscription sur la solution web.

### Composant Accueil

Le rôle de ce composant est de rassembler les informations nécessaires à afficher sur la page d'accueil : les différentes pizzas, les promotions du moment et autres informations.

### Composant Commande

Le rôle de ce composant est de gérer les commandes en cours et passées.

### Composant utilisateur

Le rôle de ce composant est de gérer les fonctions disponibles suivant l'utilisateur et d'y gérer les paramètres du compte.

### Composant Pizza

Le rôle de ce composant est de gérer les pizzas suivant leurs disponibilités et recette.

### Composant Ingrédient

Le rôle de ce composant est de gérer les ingrédients nécessaires pour les pizzas et de gérer les stocks.

## Application XXX...

# Architecture de Déploiement

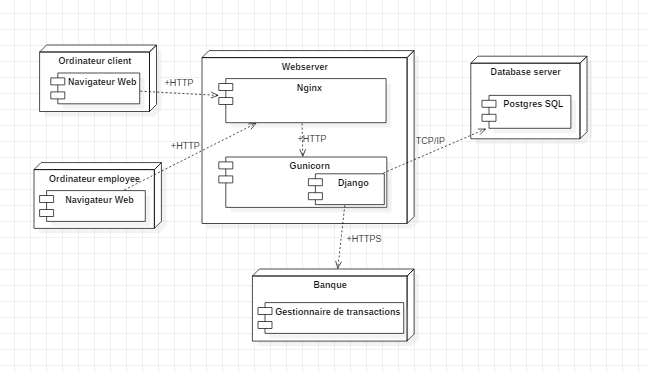


Diagramme UML de déploiement

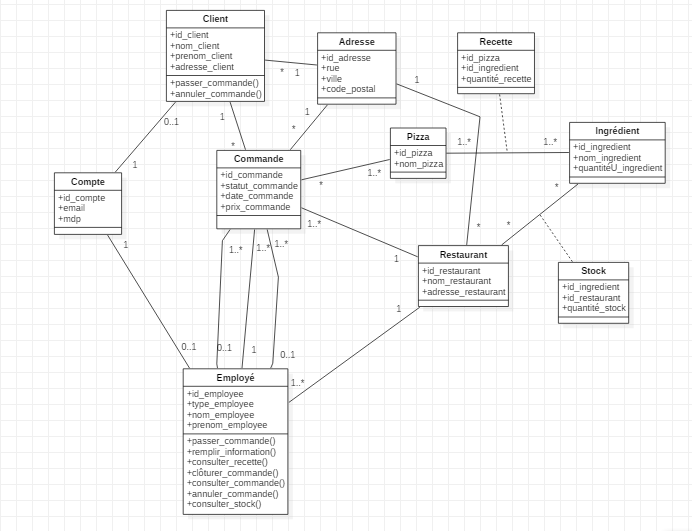
## Serveur de Base de données

Le serveur contient la base de données nécessaire à la gestion des commandes des pizzeria.

Les données de toutes les pizzerias sont réunies sur ce serveur.

La Base de données est gérée en PostregSQL.

*Modèle physique de données*



*Diagramme de Classe*

## Serveur XXX

...

# Architecture logicielle

## Principes généraux

Les sources et versions du projet sont gérées par **Git**, les dépendances et le packaging par **Apache Maven / Grunt / ...**.

...

### Les couches

L'architecture applicative est la suivante :

* unecouche **business** : responsable de la logique métier du composant
* unecouche **model** : implémentation du modèle des objets métiers
* …
  + …
* ...

### Les modules

Ex: modules Maven dans le cas d’application multi-module...

### Structure des sources

La structuration des répertoires du projet suit la logique suivante :

* les répertoires sources sont crées de façon à respecter la philosophie Maven (à savoir : « convention plutôt que configuration »)

racine  
 ├─ *pom.xml*  
 ├─ <moduleX>  
 │ ├─ *pom.xml*  
 │ └─ src  
 │ ├─ main  
 │ │ ├─ java  
 │ │ └─ resources  
 │ └─ test  
 │ ├─ java  
 │ └─ resources  
 ├─ <moduleY>  
 │ ├─ *pom.xml*  
 │ └─ src  
 │ ├─ main  
 │ │ ├─ java  
 │ │ └─ resources  
 │ └─ test  
 │ ├─ java  
 │ └─ ressources  
 └─ src  
 └─ lib

* ...

## Application Web

…

Si besoin, diagramme UML de composants pour monter les différents modules et leur inter-dépendances

## Application Xxx

…

# Points particuliers

## Gestion des logs

…

## Fichiers de configuration

### Application web

...

#### Datasources

...

#### Fichier xxx.yyy

...

### Application Xxx

...

## Ressources

...

## Environnement de développement

## Procédure de packaging / livraison

## XXX

…

# Glossaire

|  |  |
| --- | --- |
|  |  |
|  |  |