

# SOLUTION TECHNIQUE D'UN SYSTÈME DE GESTION DE PIZZERIA :

OC PIZZA souhaite

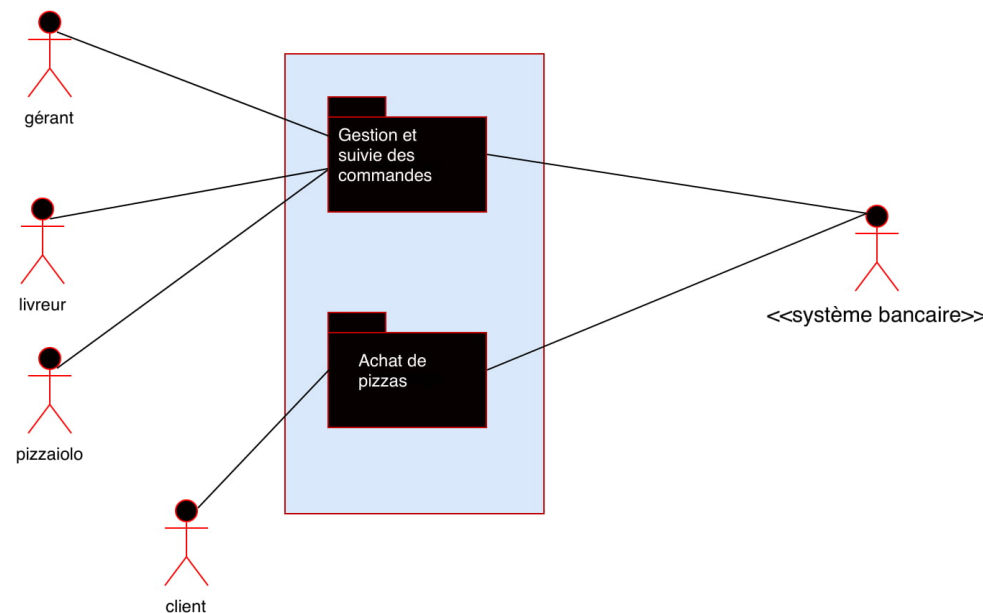
suite à la réalisation du dossier de spécification fonctionnelles ayant permis de dresser les diagrammes de package, de cas d'utilisation et d'activités. Ce dossier vient détailler les solutions techniques à travers 3 grands diagrammes :

\*\*\* Diagramme de classe

\*\*\* Diagramme de composants

\*\*\* Diagramme de déploiement

Le modèle Physique de données en fin de document établie à partir du diagramme de classe, permet de décrire concrètement et physiquement les tables de la base de données ainsi que leurs relations de clef primaires étrangères, les index détaillés.



# 1 Domaine fonctionnel du système : Classes et instances du domaine fonctionnel

## \*\*\* introduction :

le diagramme de classe : ce diagramme représente toutes les entités internes ou externes à votre programme. On appelle toute ces entités des "classes". Ici, votre schéma contient une quinzaine de classe, répartie en 4 grandes catégories de classes que sont : La pizzeria, le système de paiement, Le stock de pizza, et la gestions clients.

## \*\*\* Développement :

on retrouve entre vos classes 3 grands types de relations décrivant les interactions et échanges entre ces classes. des relations de composition, d'agrégation et de multiplicités.

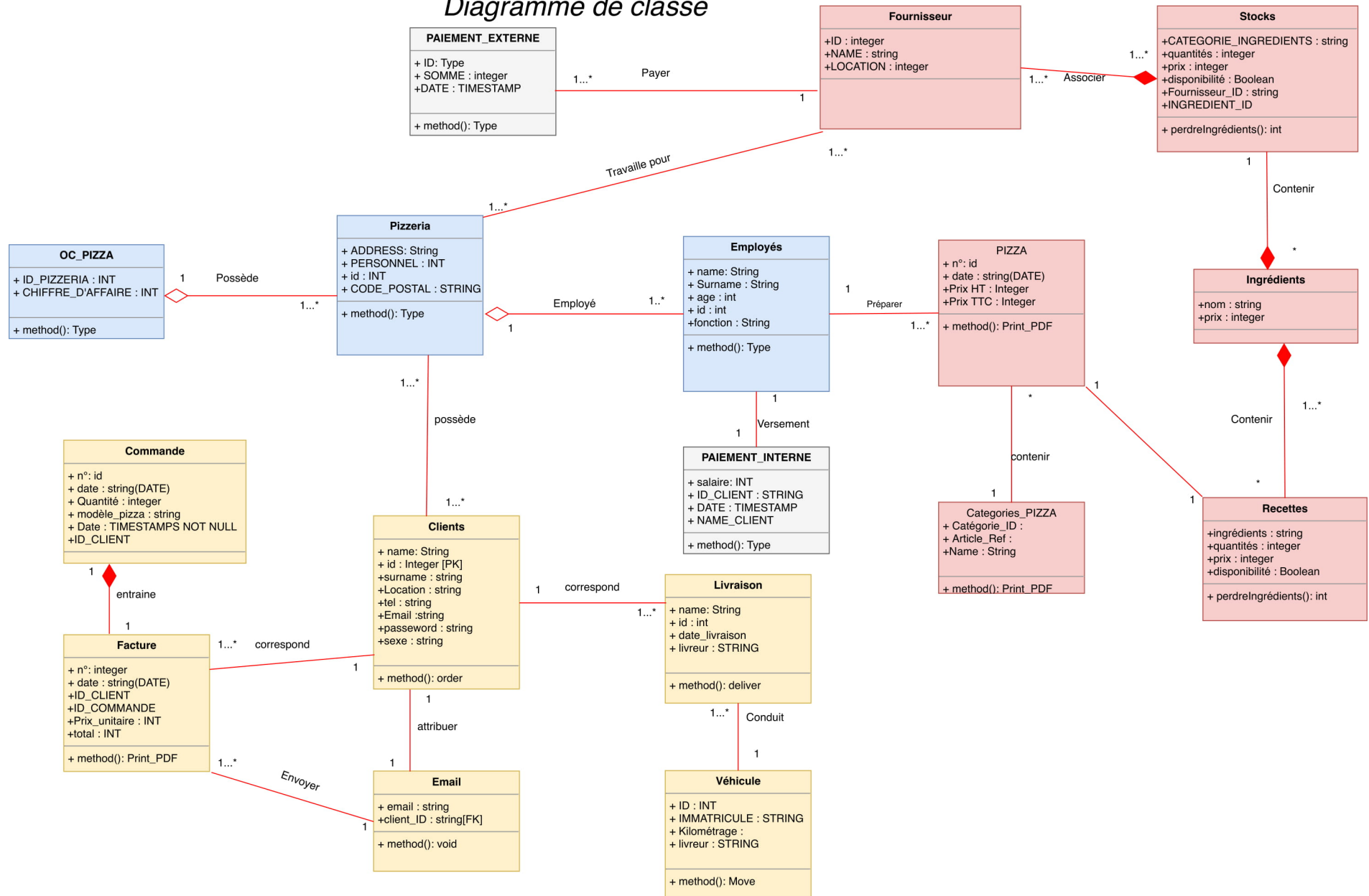
- \* Pour exemple, ici la facture dépend du système de commande et ne peut être présente sans commande, vos commandes seront donc composés de factures. Ces mêmes factures sont en relations avec l'email et le client. De ce fait, toutes ses données sont donc dépendantes des informations saisie par le client lors de la commande et la facture est le résultat de ces autres classes.
- \* Par ailleurs, on voit que la gestion des stocks est découpée en 3 entités "recettes", "ingrédients" et "stocks". la classe stock aura pour attributs des quantités, une disponibilité sous forme de vrai ou faux (aussi appelé Booléen) mais aussi une catégorie d'ingrédients. Un fournisseur sera donc associer à votre stock et pourra prendre en charge une ou plusieurs catégories d'ingrédients.
- \* Le système de paiement, est lui divisé en 2 classes externe et une interne. on retrouve comme dans chaque classe un ID (ici de paiement). Mais aussi une Date, un entier réel indiquant le montant du versements. Les paiements externes sont donc ici liées à vos fournisseurs. Tandis que les paiements internes sont liées à vos clients. On retrouve un ID en clef étrangère ce qui permet de référencer avec exactitude la colonne du receveur dans la table paiement
- \* Les pizzas sont associées chacune à des catégories permettant de plus vite effectuer les requêtes et recherches dans la future base de données. Ces mêmes pizzas seront préparés par un seul employés par pizzeria, cet employé lui peut travailler pour toutes les pizzeria du groupe OC PIZZA mais est assigné à une spécifique.

## \*\*\*

A partir de ces relations entre classes permettant de poser les bases d'échanges entre vos classes, j'ai crée un modèle physique et concret de données représentant votre Base de données finale. Aussi appellé MPD, voici en détail sa présentation

# Diagramme de classe

Diagramme de classe



# 2 Composants du système : Interaction des composants internes et externes du programme

## \*\*\* Introduction :

Le diagramme de composants permet d'identifier graphiquement les différents composants concrets, techniques internes et externes à votre site web de pizzeria. Ils contiennent par exemple les exécutables, les bibliothèques, les fichiers. Mais aussi des données comme la base de données. On révèle ici en évidence les dépendances (d'utilisations entre ces composants)

## \*\*\* Relations :

\* Ces mêmes composants sont répertoriés en 2 grands systèmes de composants : 1 les composants tiers et 2 les programmes serveurs. Ces grands paquets interagissent ensemble et s'échangent des informations dynamiquement pour les fournir ensuite à l'utilisateur. En effet le serveur et le Back-end (c'est-à-dire la surface invisible mais nécessaire au fonctionnement de votre programme, fournissant l'interaction entre les données entrantes et sortantes) sont contenus dans un grand composant "Serveur".

\* De l'autre côté on retrouve le navigateur clients affichant les informations après interprétation du code HTML fournie par le Back End. L'API Google maps et du paiement font aussi partie de cet exemple car ils sont extérieurs au système principal.

\* Techniquement votre base de données est gérée par un ORM propre au Framework Django du langage Python. Cet ORM (object-relational mapping) simule une base de données relationnelles orientées objet, comme vue précédemment dans le diagramme de classe. La base de données POSTGRESQL communique avec L'ORM

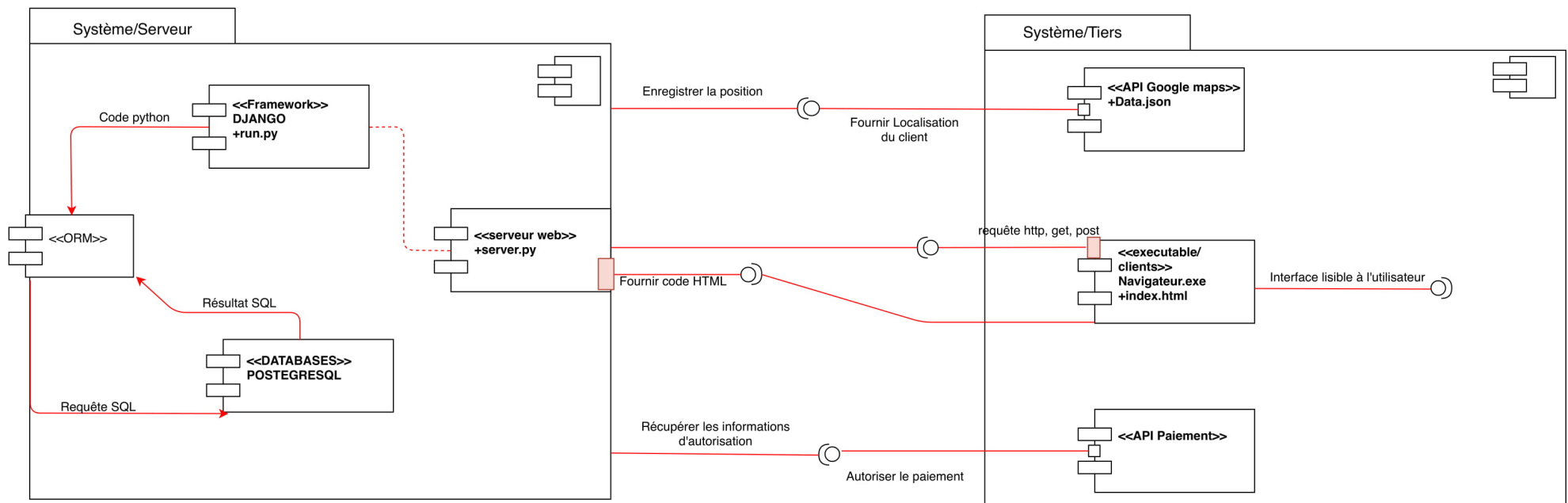
\*\*\*

Ces composants ici détaillés en détail sont de nouveaux présents dans le déploiement de ces composants expliqué dans la page suivante.

# Diagramme de composant

Diagramme de composants

V2



# 3 Déploiement du système : Organisation physique des composants, déploiement détaillé

## \*\*\* Introduction :

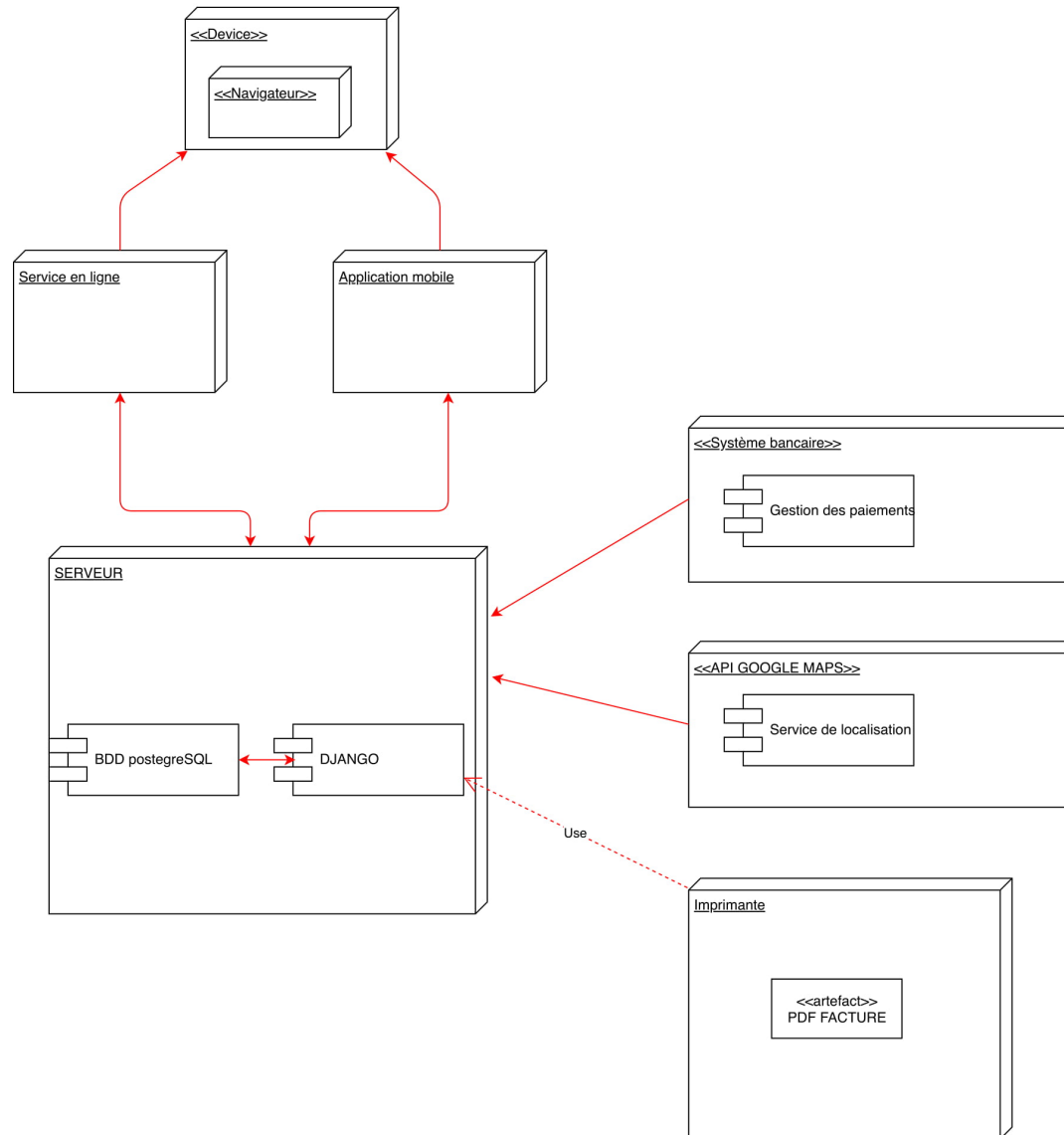
Le diagramme de déploiement est représenté ici par 5 "noeuds". Ces noeuds sont des classeurs ayant la forme de boîtes en 3 dimensions contenant les composants régissant le programme. On décrit donc ici le déploiement physique des informations générées par le Logiciel Principal OC PIZZA, ces informations sont appelées artefacts dans le jargon.

## \*\*\* Relations :

- \* Au sein de ce diagramme, les lignes d'associations représentées par des flèches représentent l'échange d'information entre le back end et le site web en ligne/l'application mobile.
- \* Le système bancaire et l'API fournissent eux des autorisations et des informations géographiques, votre site web sera donc dépendant de ces composants illustrés ici par les pointillés.
- \* L'espace utilisateur contient lui une sous-boîte représentant le navigateur et plus spécifiquement le "décodage" graphique du HTML5 pour afficher des informations compréhensibles à l'utilisateur. L'échange d'informations allant dans les deux sens c'est ici une double flèche qui caractérise ce même échange.
- \* L'imprimante ici présente interagit directement avec le programme Back end fournissant automatiquement le PDF imprimable de la commande du client.

# Diagramme de déploiement

Diagramme de déploiement



# 4 Modèles relationnel de la base de données : interaction entre les tables



# Modèle physique de données

