



Projet 6 – Parcours Développeur d’application – Python

OPENCLASSROOMS

Concevez la solution technique d’un système de gestion pour un groupe de pizzeria **OC PIZZA**

Dossier de conception fonctionnelle

Version 1.0

Auteur

Lamia EL RALIMI

Etudiante



TABLE DES MATIERES

1 -Versions.....	3
2 -Introduction	4
2.1 -Contexte	4
2.2 -Objectif du document	4
3 -Description du domaine fonctionnel.....	5
3.1 -Diagramme de classe	5
3.2 -Relation entre les classes	5
4 -Modèle Physique de Données	6
5 -Composants du système	7
5.1 -Diagramme de composants	7
5.2 -Description	8
6 -Déploiement des composants	9
6.1 -Diagramme de déploiement	9
6.2 -Description	9

1 - VERSIONS

Auteur	Date	Description	Version
Lamia EL RALIMI	02/10/2020	Création du document	1.0

2 - INTRODUCTION

2.1 - Contexte

« OC Pizza » est un jeune groupe de pizzeria 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 vous afin de mettre en place un système informatique, déployé dans toutes ses pizzerias et qui lui permettrait notamment :

- d'être plus efficace dans la gestion des commandes, de leur réception à leur livraison en passant par leur préparation ;
- de suivre en temps réel les commandes passées et en préparation ;
- de suivre en temps réel le stock d'ingrédients restants pour savoir quelles pizzas 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
- d'informer ou notifier les clients sur l'état de leur commande

2.2 - Objectif du document

Ce document permet de définir le domaine fonctionnel du futur système ainsi que de concevoir l'architecture technique de la solution.

Pour cela, nous avons :

- décrit le domaine fonctionnel avec un diagramme de classe,
- élaboré un modèle physique de données,
- identifié les composants du système dans un diagramme de composants,
- réalisé un diagramme de déploiement de ces composants.



3 - DESCRIPTION DU DOMAINE FONCTIONNEL

3.1 - Diagramme de classe

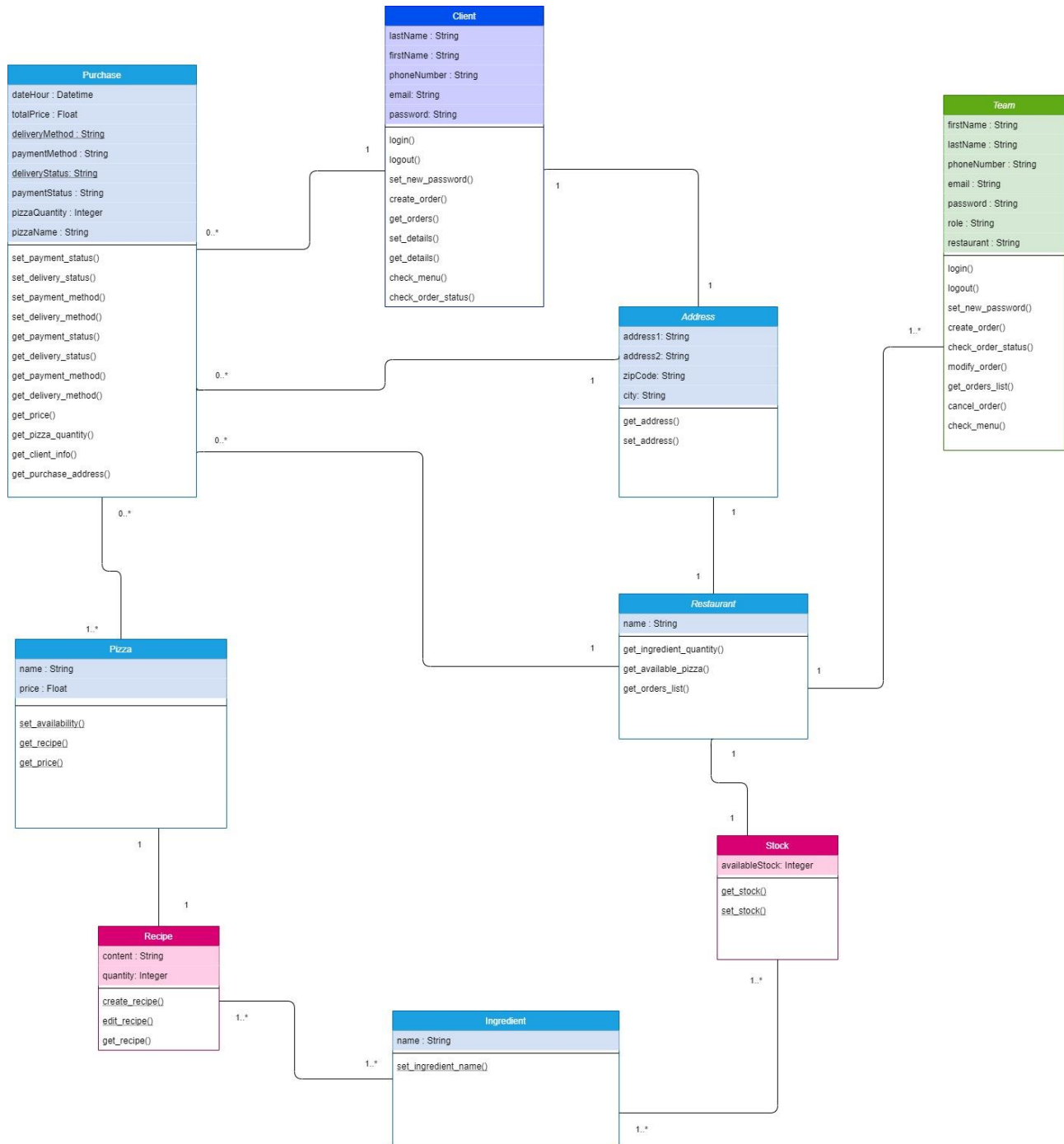


Diagramme de classe



3.2 - Relation entre les classes

Purchase – Client

- **Purchase** représente la commande passée par le client.
- **Client** représente un client de(s) pizzeria(s).

Un client est associé à zéro, une ou plusieurs commandes. Une commande est associée à un seul client.

Purchase – Address

- **Purchase** représente une commande.
- **Address** représente l'adresse de livraison d'une commande.

Une commande est associée à une seule adresse de livraison. Une adresse de livraison est associée à zéro, une ou plusieurs commandes.

Purchase – Restaurant

- **Purchase** représente une commande.
- **Restaurant** représente un restaurant de OC Pizza.

Une commande est associée à un seul restaurant. Un restaurant est associé à zéro, une ou plusieurs commandes.

Purchase – Pizza

- **Purchase** représente une commande.
- **Pizza** représente une pizza.

Une commande est associée à une ou plusieurs pizzas. Une pizza est associée à zéro, une ou plusieurs commandes.

Client – Address

- **Client** représente un client de(s) pizzeria(s).
- **Address** représente l'adresse de facturation du client.

Un client est associé à une seule adresse de facturation. Une adresse est associée à un client.



Address – Restaurant

- **Address** représente une adresse de restaurant.
- **Restaurant** représente un restaurant du groupe OCpizza.

Une adresse de restaurant est associée à un seul restaurant. Un restaurant est associé à une seule adresse de restaurant.

Restaurant – Team

- **Restaurant** représente un restaurant du groupe OCpizza.
- **Team** représente un employé.

Un restaurant est associé à un ou plusieurs employés. Un employé est associé à un restaurant.

Restaurant – Stock

- **Restaurant** représente un restaurant du groupe OCpizza.
- **Stock** représente un stock d'ingrédients.

Un restaurant est associé à un seul stock d'ingrédients. Un stock d'ingrédients est associé à un seul restaurant.

Pizza – Recipe

- **Pizza** représente une pizza.
- **Recipe** représente une recette.

Une pizza est associée à une seule recette. Une recette est associée à une seule pizza.

Ingredient – Recipe

- **Ingredient** représente un ingredient de pizza.
- **Recipe** représente une recette.

Un ingredient est associé à une ou plusieurs recette(s). Une recette est associée à une ou plusieurs ingredient(s).

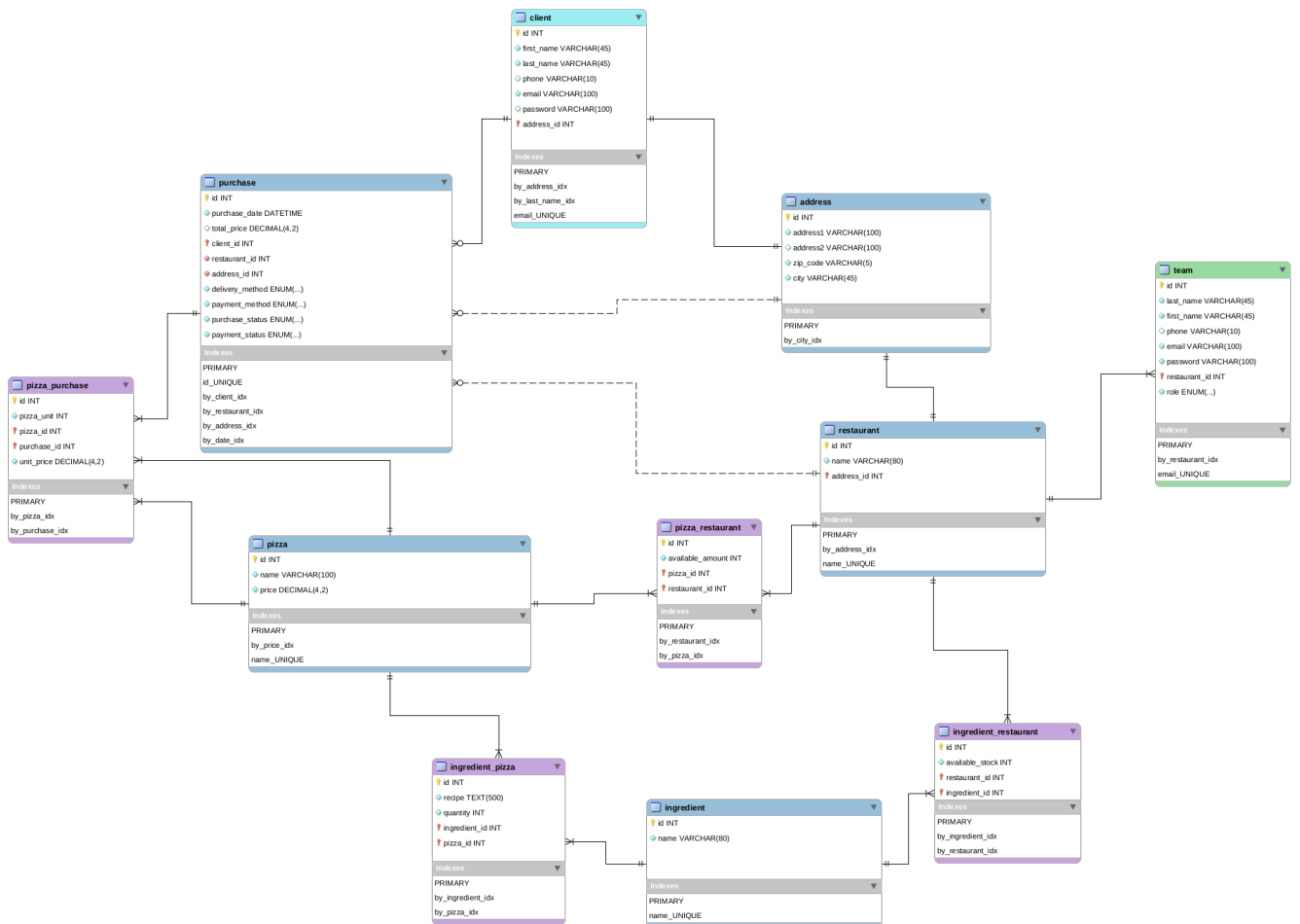
Ingredient – Stock

- **Ingredient** représente un ingredient de pizza.
- **Stock** représente un stock d'ingrédients.

Un ingredient est associé à un ou plusieurs(s) stock(s). Un stock d'ingrédients est associé à un ou plusieurs ingredient(s).



4 - MODÈLE PHYSIQUE DE DONNÉES



Modèle physique de données

5 - COMPOSANTS DU SYSTÈME

5.1 - Diagramme de composants

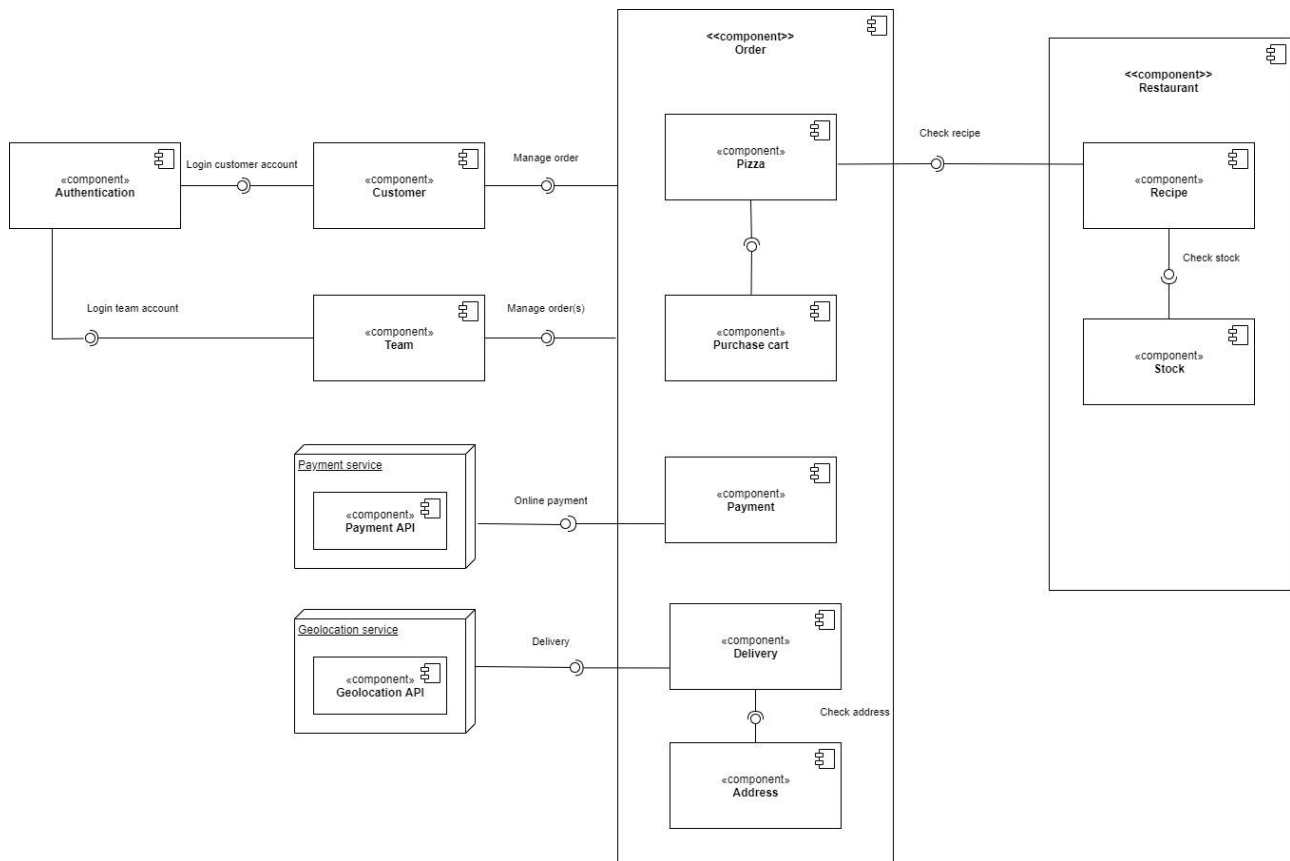


Diagramme de composants

5.2 - Description

Composants internes :

◇ **Authentication**

Composant d'authentification à un compte d'utilisateur pour accéder à l'application en fonction du rôle « client » ou « employé ».



◇ **Customer**

Le compte « client » permet de créer/gérer des commandes, régler en ligne, consulter les statuts de celles-ci, consulter les anciennes commandes, de gérer ses informations personnelles.

◇ **Team**

Le compte « employé » permet de créer des commandes pour les clients, de modifier/consulter les statuts de celles-ci, de gérer ses informations personnelles et de consulter les anciennes commandes.

◇ **Order**

Composant qui permet de gérer les commandes. Ce composant comporte la constitution/gestion du panier, l'ajout de l'adresse de livraison, le traitement du paiement et de la livraison le cas échéant.

◇ **Restaurant**

Le composant restaurant permet de gérer les recettes des pizzas et les stocks.

Composants externes :

◇ **Payment API**

Composant permettant de gérer les paiements en ligne via l'API de Stripe.

◇ **Geolocation API**

Composant permettant de gérer les livraisons via l'API de géolocalisation.

6 - DÉPLOIEMENT DES COMPOSANTS

6.1 - Diagramme de déploiement

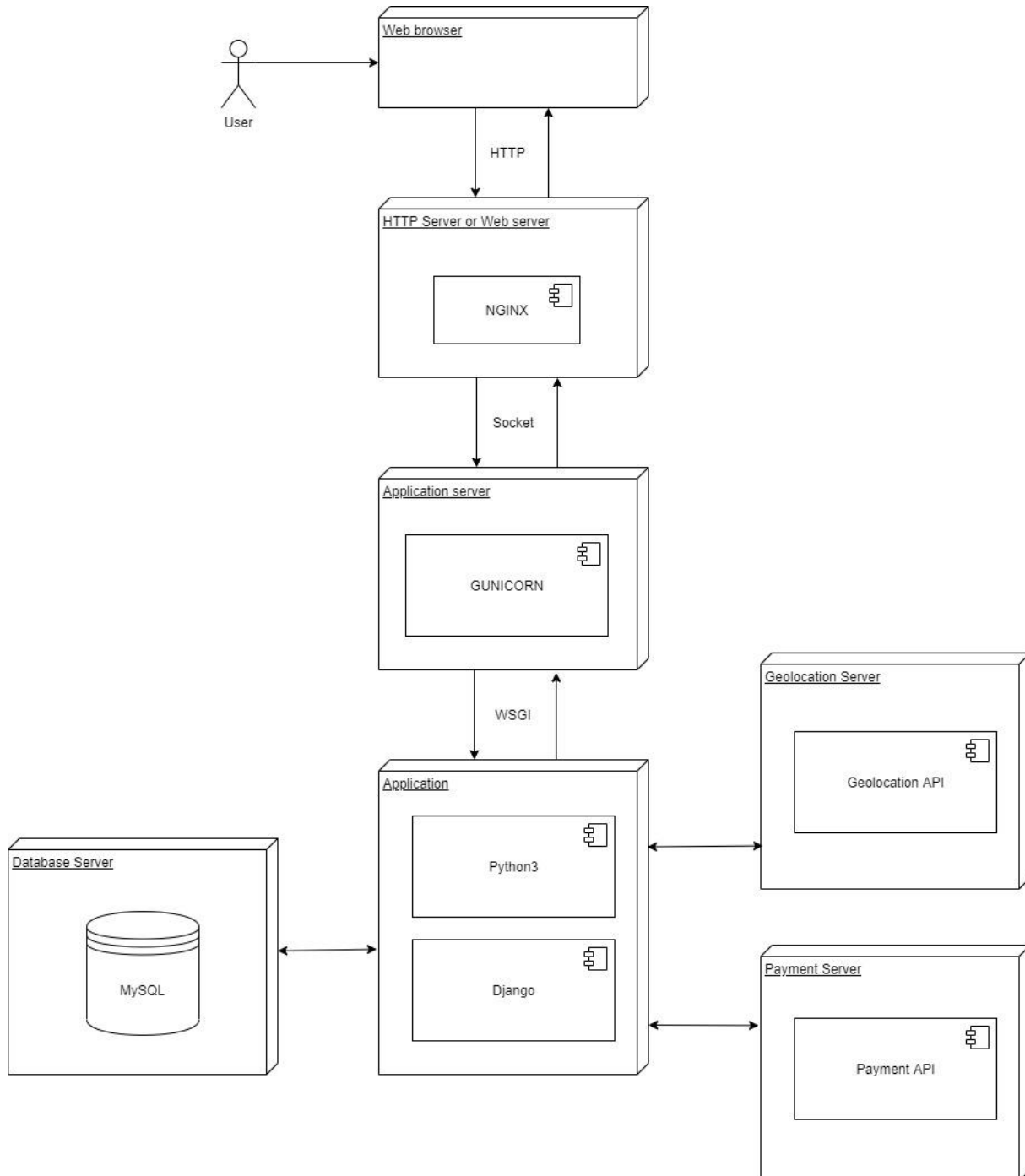


Diagramme de déploiement



6.2 - Description

- L'application sera développée grâce au framework Django. La base de données sera créée à l'aide d'un système de gestion de base de données relationnelles (SGBDR) MySQL qui communiquera avec le serveur d'application.
- Les transactions bancaires en ligne seront gérées via l'API de Stripe avec laquelle le serveur d'application communiquera.
- Le système de géolocalisation pour les livraisons sera géré via l'API de géolocalisation avec laquelle le serveur d'application communiquera.
- L'application sera déployée sur un serveur d'application utilisant Gunicorn.
- Les utilisateurs, via le navigateur de leur appareil (ordinateur, tablette ou smartphone), interagissent avec un serveur HTTP Nginx.