



## OC Pizza

### Spécifications techniques Version 1.0

## TABLE DES MATIÈRES

I.	<u>Versions.....</u>	<u>1</u>
II.	<u>Introduction.....</u>	<u>1</u>
III.	<u>Etude du contexte.....</u>	<u>2</u>
IV.	<u>Etude des packages.....</u>	<u>4</u>
V.	<u>Etude du cycle de vie d'une commande.....</u>	<u>6</u>
VI.	<u>Description du domaine fonctionnel.....</u>	<u>8</u>
VII.	<u>Etudes techniques.....</u>	<u>13</u>
VIII.	<u>Les composants internes.....</u>	<u>15</u>
	A. <u>Modèle physique de donnée.....</u>	<u>17</u>
IX.	<u>Les composants externes.....</u>	<u>18</u>
X.	<u>Glossaire.....</u>	<u>20</u>

**Auteur**  
BESEVIC Ivan

## Versions

Auteur	Date	Description	Version
IB	06/02/2019	Création du document	1.0

## Introduction

Le présent document constitue le dossier de conception technique de l'application OC\_PIZZA. L'objectif du document est de présenter les outils, les technologies, et les méthodes mises en œuvre pour réaliser l'application.

# Etude du contexte

En étudiant le cahier des charges on peut distinguer les différents acteurs qui interagissent avec le système de la pizzeria.

Voici la liste :

- ❖ Le client
- ❖ Le commercial
- ❖ Le préparateur de commande
- ❖ Le Livreur
- ❖ Le manager
- ❖ Le système bancaire
- ❖ Le système de géolocalisation

## Acteurs principaux

**Les clients** pourront passer des commandes via le site internet de la pizzeria qui proposera des produits en fonction du stock disponible.

Les clients auront la possibilité de payer en ligne ou en espèce à la livraison du produit.

**Les commerciales** pourront passer les commandes des clients qui sont aux téléphones ou sur places dans le restaurant.

**Les préparateurs de commandes** utilisent le système pour suivre les commandes validés en attente de préparation ainsi que contrôler la disponibilité de la matière première dans le stock.

**Les livreurs** pourront récupérer les coordonnées des dernières commande en attente de livraison et livrer la commande au client.

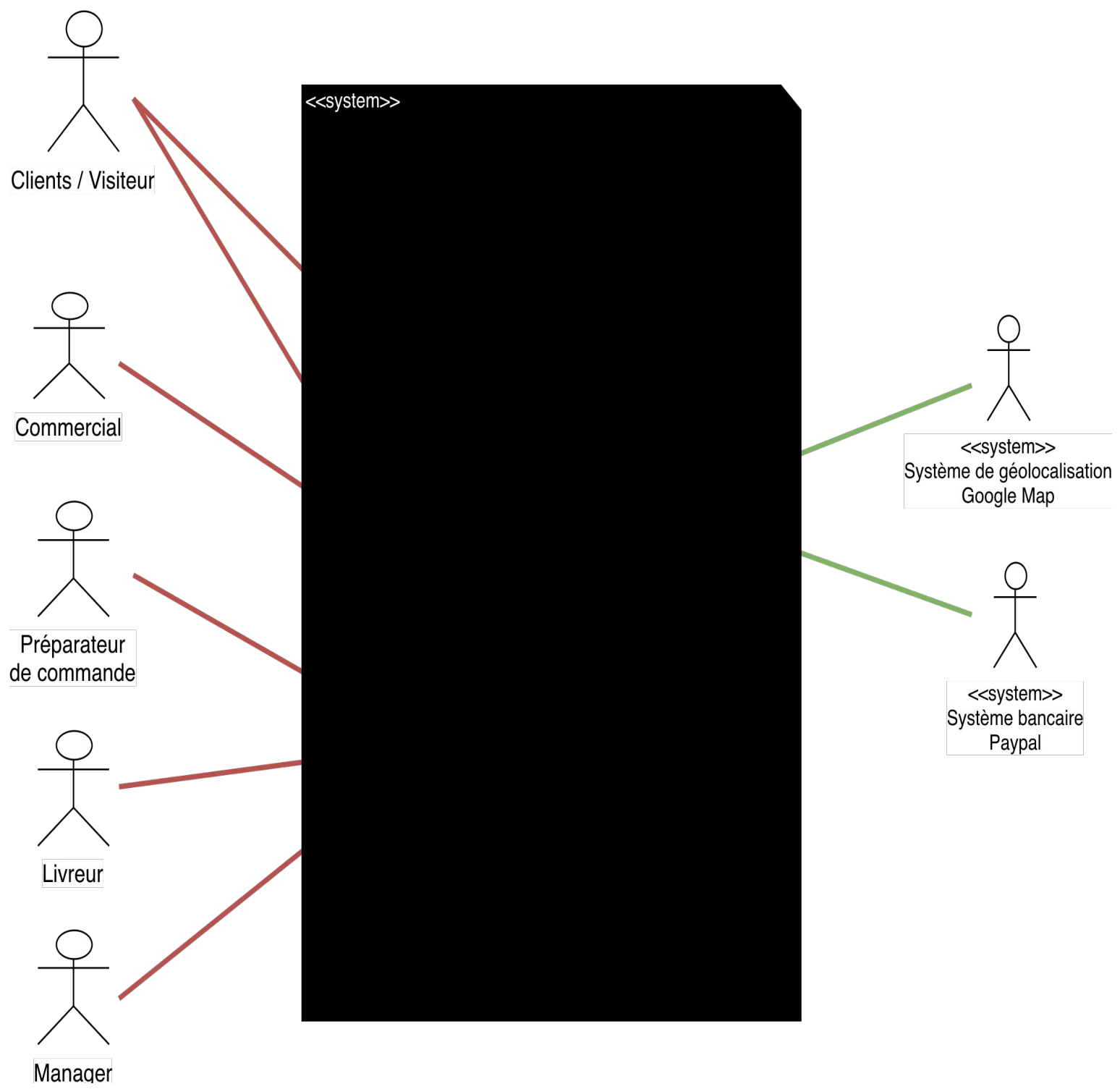
**Les managers** pourront accéder au chiffre d'affaires, ajouter ou modifier un produit dans le catalogue et lire les messages d'erreur provenant du système.

## Acteurs secondaire

**Le système bancaire** pour payer en ligne.

**Le système de géolocalisation** est utilisé pour savoir si un client est éligible à la livraison, par exemple, si un client habite trop loin des restaurants il ne pourra pas commander sur le site Internet.

# Etude du contexte - Diagramme de contexte



## Etude des packages

Le système peut-être divisé en quatre parties

**Le package “interface d’achat”** regroupe la partie site web qui permettra au client de commander des produits en ligne. Il pourra visualiser une liste de produits, créer un panier pour stocker tous ses articles, de proposer divers moyens de paiement et de vérifier l'avancée de sa commande.

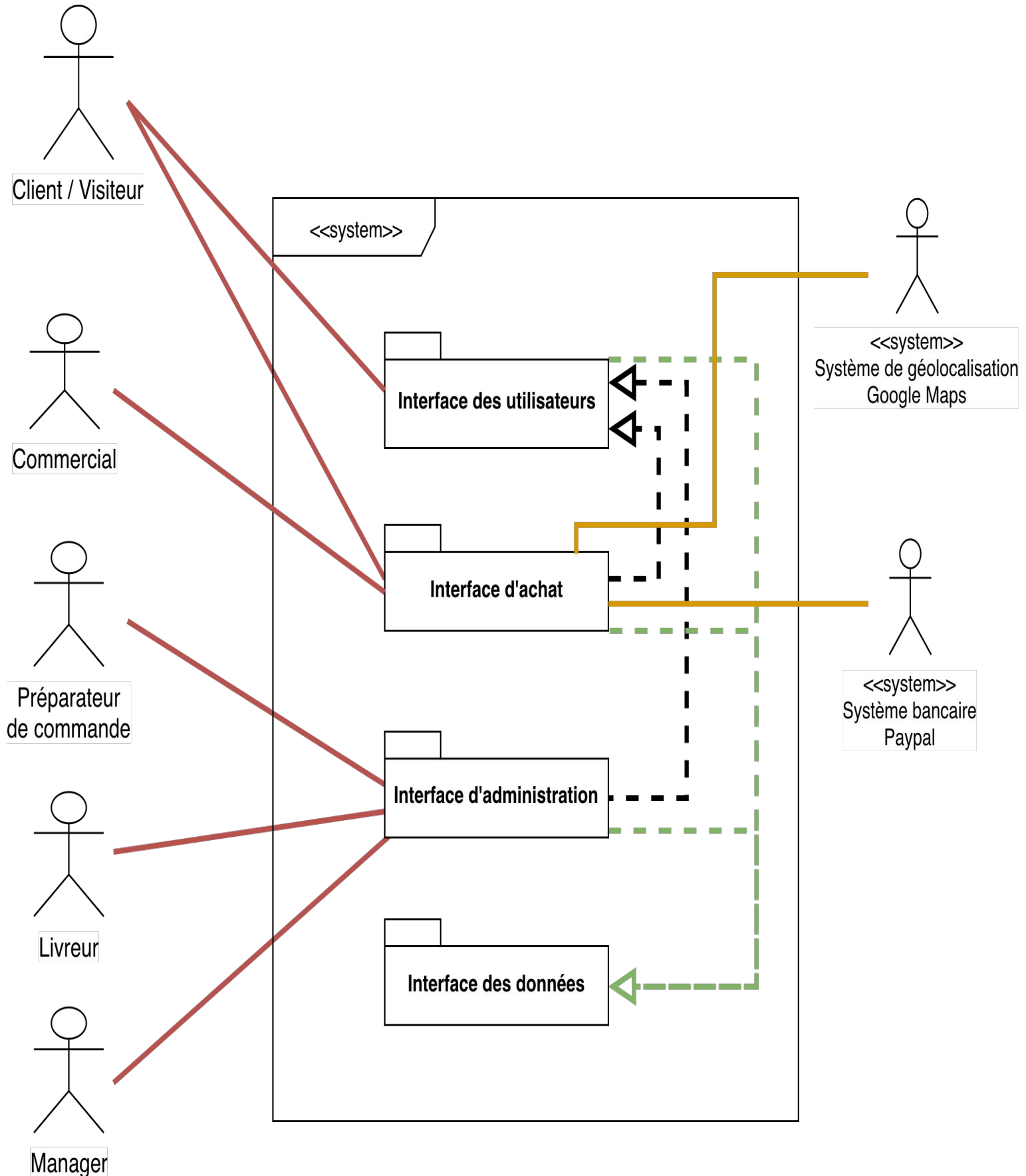
**Le package “interface d’administration”** regroupe les fonctionnalités de gestions administrateur que propose le site internet de la pizzeria.

**Le package “Interface des utilisateurs”** s’occupe de la collecte des informations du client et de l’identification des différents types d’acteurs pouvant communiquer avec le site internet.

**Le package “Interface des données”** s’occupe de stocker les informations concernant les utilisateurs et les restaurants.

Acteur	Interface d’achat	Interface d’administration	Interface des utilisateurs	Interface des données
Client / Visiteur	X		X	
Commercial	X		X	
Préparateur de commande		X		
Livreur		X		
manager		X		
système bancaire	X			

# Etude des packages - Diagramme de packages



# Etude du cycle de vie d'une commande

Le cycle de vie d'une commande est constitué de plusieurs étapes, allant de la validations du panier par le client jusqu'à la livraison. Le client à la possibilité supprimer ou modifier son panier avant sa validation.

## Statuts d'une commande

Une commande peut avoir quatre statuts :

- **Statut «en attente» :**

Ce statut est généré par l'action du client «Valider son panier ».

C'est aussi le statut de création de la commande. Ce statut veut dire que la commande est passée au système mais n'a pas encore été prise en charge par un préparateur de commande.

- **Statut « en cours de préparation » :**

Ce statut est généré par l'action d'un préparateur de commande, lorsqu'il commence la préparation de la commande.

- **status « prêt pour livraison au comptoir » :**

Si la commande est de la part d'un client qui se trouve au comptoir.

- **Statut « prêt pour livraison » :** Ce statut est généré par l'action d'un préparateur de commande, lorsqu'il fini la préparation de la commande.

- **Statut «en livraison» :**

Ce statut est généré par l'action d'un livreur lorsqu'il reçoit la commande préparé.

- **Statut «Livré» :**

Pour que ce statut soit atteint il faut que la commande soit livrée et payée. C'est le livreur qui prend en charge cette étape final et la mise à jour du statut.

- **Statut «Non livré» :**

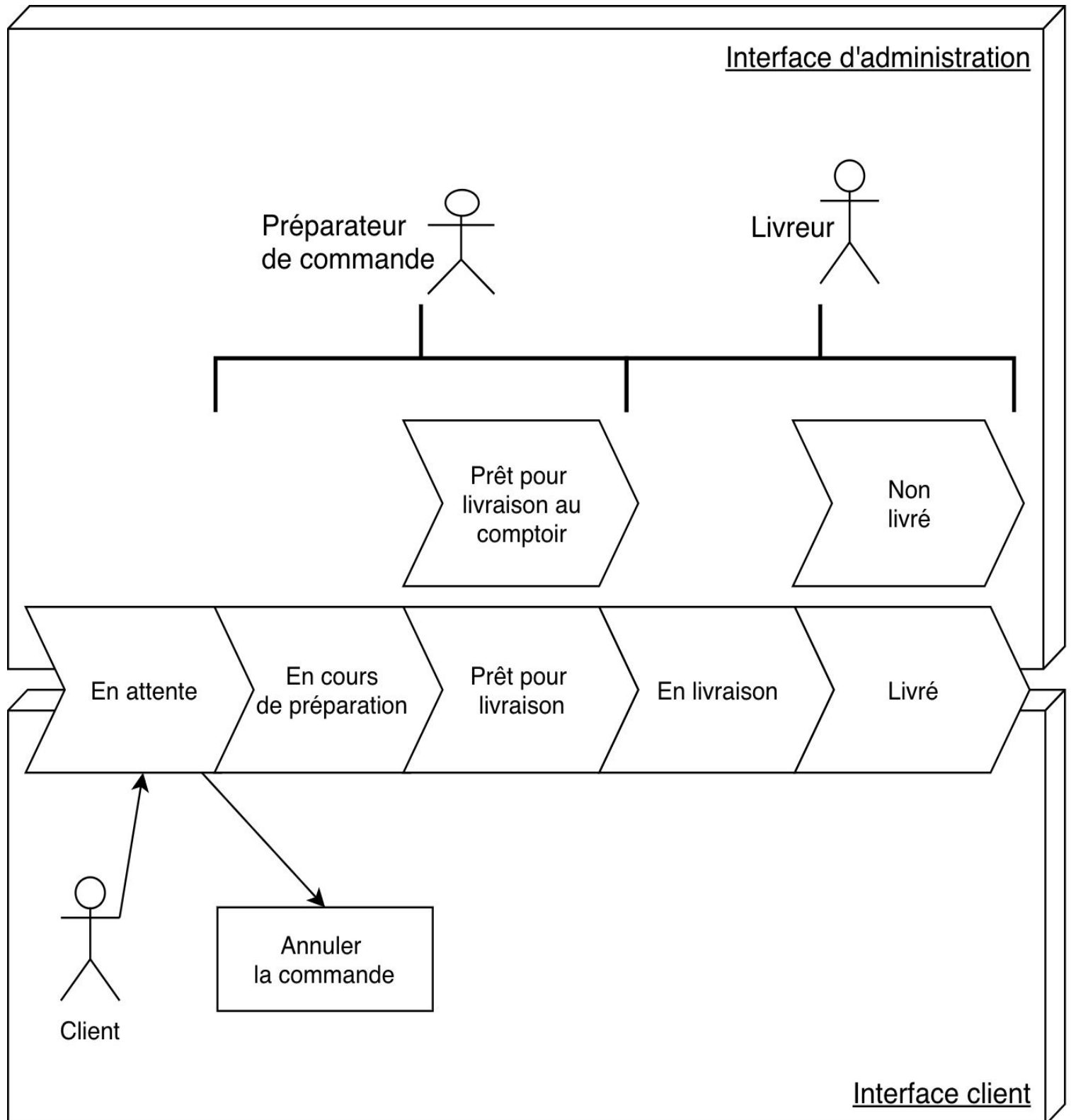
Le livreur a reçu une mauvaise adresse et le client est injoignable.

## Annulation d'une commande

Après validation du panier, l'annulation d'une commande peut encore être faite par l'action du client : elle n'est possible que lorsque la commande est en statut « en attente ». Dans ce cas, le système supprime la commande de la base de données et rembourse le client si il a payé en ligne.

## Etude du cycle de vie d'une commande

- Synthèse

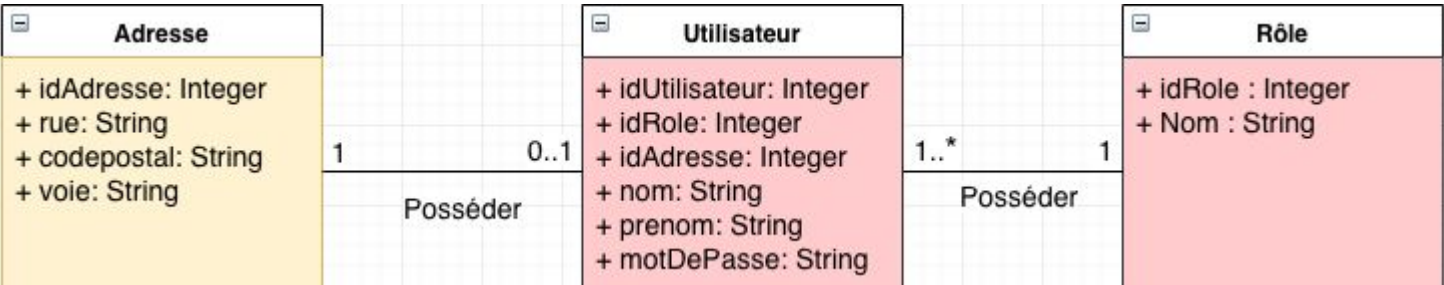




# DESCRIPTION DU DOMAINE FONCTIONNEL :

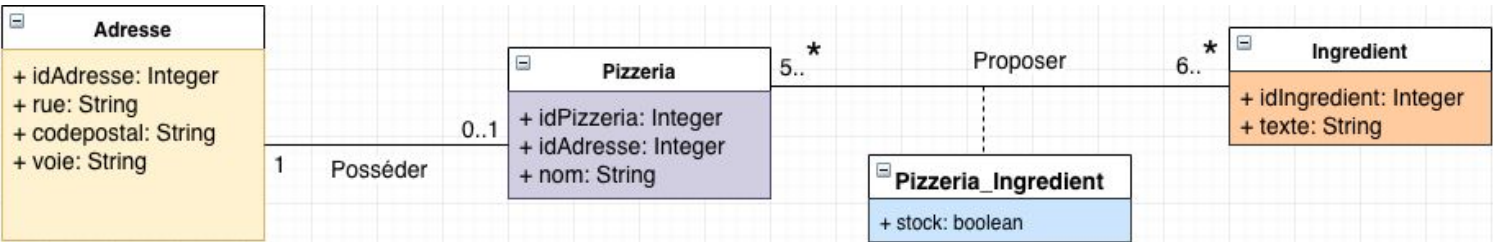
Dans cette partie, nous expliciterons les différentes entités du domaine fonctionnel permettant la construction d'un modèle physique de données.

## L'utilisateur:



Un **utilisateur** possède une et une seul **adresse** et une **adresse** peut être détenue par aucun ou un **utilisateur**.  
Un **utilisateur** possède un et un seul **rôle** et un **rôle** doit être détenue par au moins un **utilisateur**.

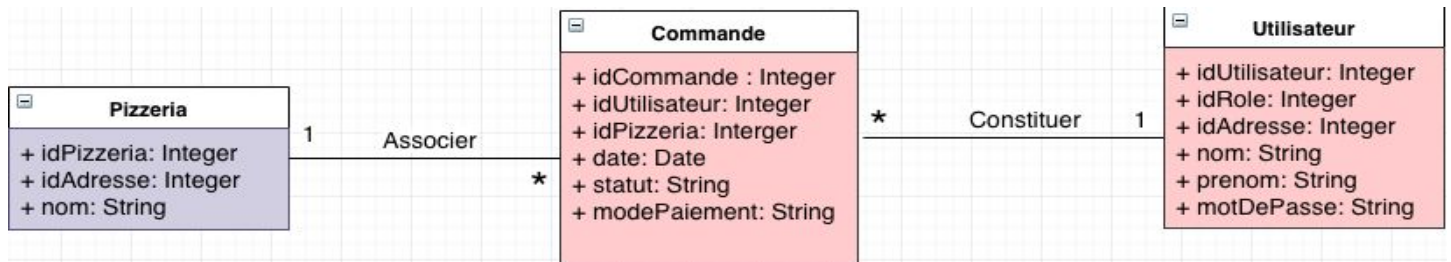
## Les pizzerias:



Une **pizzeria** possède une et une seul **adresse** et une **adresse** peut être détenue par aucune ou une **pizzeria**.  
Une **pizzeria** doit proposer au moins six un **ingrédient** et un **ingrédient** doit être proposé par au moins cinq **pizzerias**.

## DESCRIPTION DU DOMAINE FONCTIONNEL (suite) :

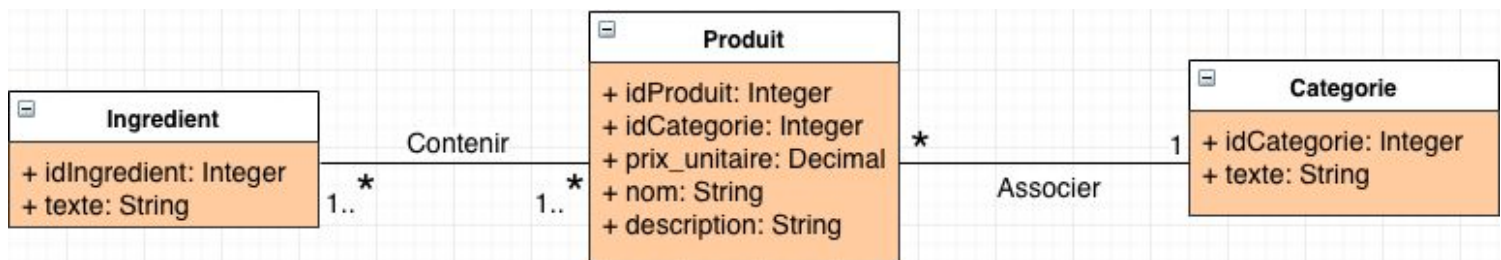
### Les commandes:



Une **commande** est associée à une et une seule **pizzeria** et une **pizzeria** peut être associée à aucune, une ou plusieurs **commandes**.

Une **commande** est constituée par un et un seul **utilisateur** et un **utilisateur** peut constituer aucune, une ou plusieurs **commandes**.

### Les produits:

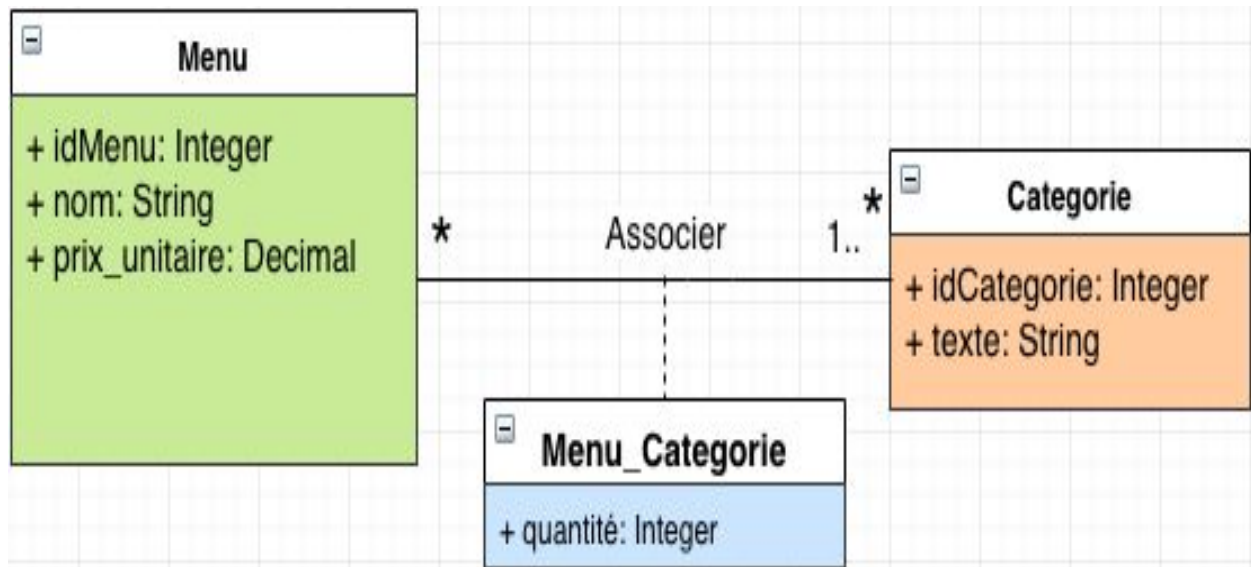


Un **produit** contient au moins un **ingrédient** et un **ingrédient** est contenu dans au moins un **produit**.

Un **produit** est associé à une et une seule **catégorie** et une **catégorie** est associée à aucun, un ou plusieurs **produits**.

## DESCRIPTION DU DOMAINE FONCTIONNEL (suite) :

### Les menus:

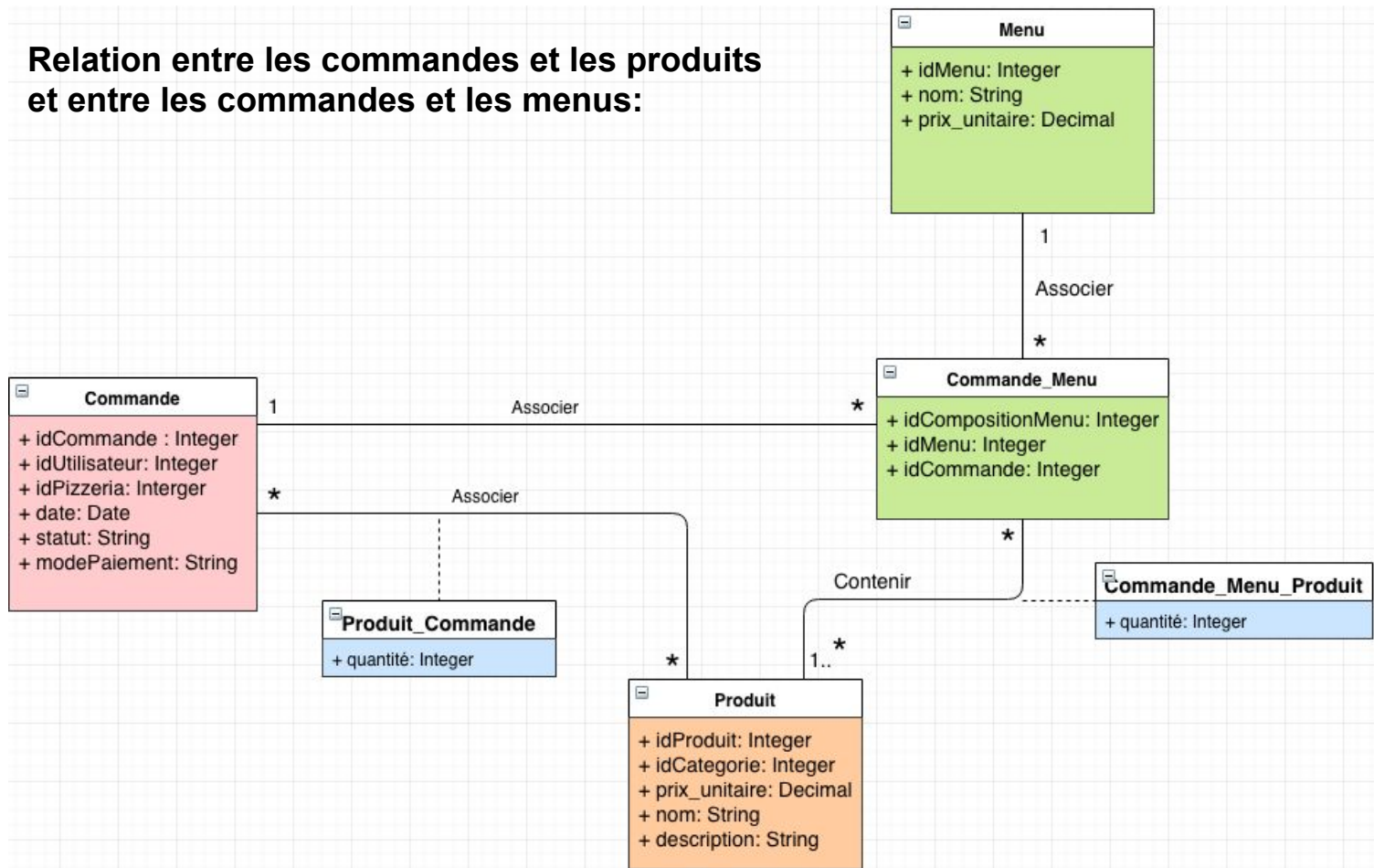


Un **menu** est associé à au moins une **catégorie** et une **catégorie** peut être associée à aucun, un ou plusieurs **menus**.

Par exemple un menu peut être associé à une entrée, deux pizzas et une boisson.

## DESCRIPTION DU DOMAINE FONCTIONNEL (suite) :

Relation entre les commandes et les produits  
et entre les commandes et les menus:



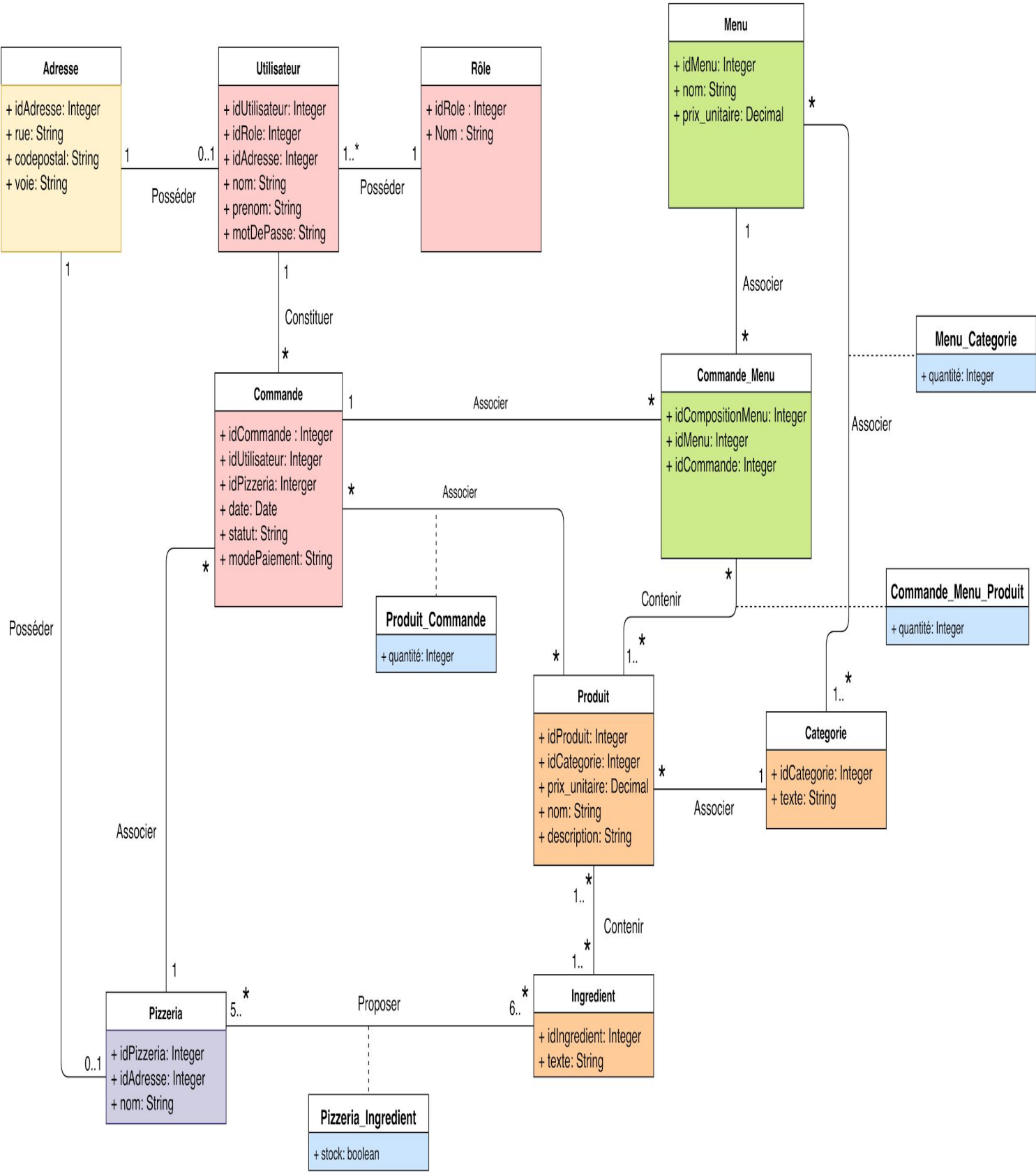
Une **commande** peut être associée à aucun, un ou plusieurs **produits** et un **produit** peut être associé à aucune, une ou plusieurs **commandes**.

Une **commande** peut être associée à aucune, une ou plusieurs "**commande\_menu**" et une "**commande\_menu**" doit être associée à une et une seul **commande**.

Une "**commande\_menu**" est associée à un et un seul **menu** et un **menu** est associé à aucune, une ou plusieurs "**commande\_menu**".

Une "**commande\_menu**" contient au moins un **produit** et un **produit** peut être contenu dans aucune, une ou plusieurs "**commande\_menu**".

# Synthèse des entités retenus dans un diagramme modélisant le domaine fonctionnel :



## Etudes techniques

Pour une mise en œuvre fiable, efficace et évolutive des besoins clients décrit ci dessus, il est proposé d'utiliser un ensemble de technologies web parmi les plus fiable et au développement très actif.

### **Django framework**

Django est un framework web libre et open-source, écrit en Python facilitant la création de sites Web complexes. Django permet une adaptabilité en cas d'évolution future car il comporte de nombreuses bibliothèques pré-codées.

### **PostgreSQL**

PostgreSQL est un système de gestion de base de données relationnelle et objet. C'est un outil libre.

### **Serveur web :**

Nous utiliserons NGINX qui est un logiciel libre de serveur Web.

### **PayPal**

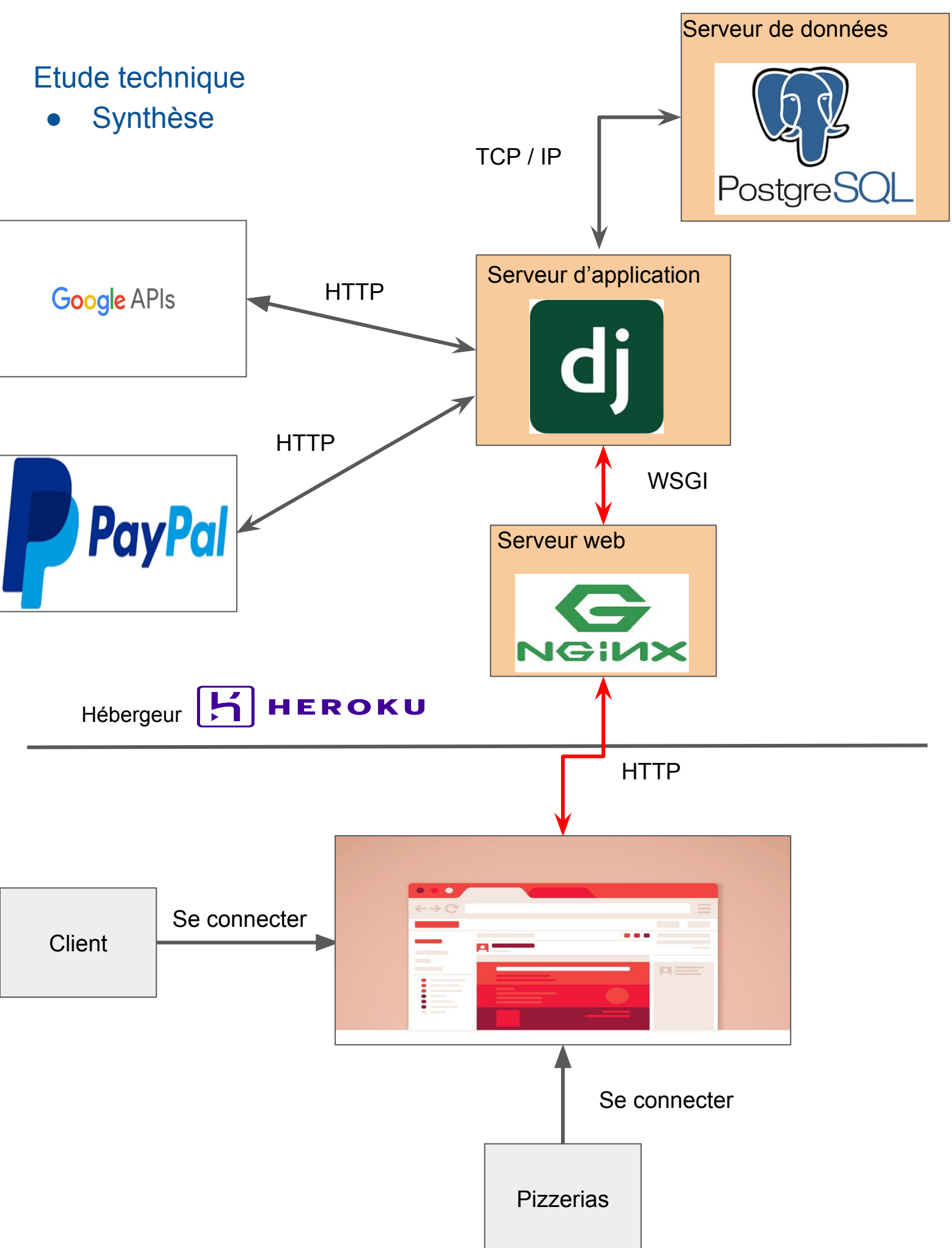
PayPal est un service de paiement en ligne. Paypal est accepté comme moyen de paiement par plus de 16 millions de commerçants dans le monde.

### **Google MAP API Distance Matrix API**

LA Distance Matrix API est un service qui fournit la distance et le temps pour une matrice d'origines et de destinations. L'API renvoie des informations basées sur l'API Google Map.

Etude technique

- Synthèse





## Les composants internes


- Explication

### **Serveur d'application :**

- Nous utiliserons un serveur programmé en Python en respectant les bonnes pratiques de la PEP 8. Toute la logique métier se trouvera sur ce même serveur.
- L'interface python-postgresql utilisé sera `psycopg2`.
- Pour améliorer les pages HTML, nous utiliserons Bootstrap et JavaScript avec AJAX :
  - Bootstrap permettra de modifier facilement la présentation des pages web demandées par l'utilisateur.
  - JavaScript pour rendre les pages web plus ergonomiques et interactive.
  - AJAX pour mettre à jour une partie de la page web en limitant les échanges client/serveur.
- La base de données portera le nom de « oc\_pizza » (l'unique base de données du projet).

### **Hébergement des serveurs :**

Nous opterons pour un hébergement sur le cloud ceci permettant notamment de facilement modifier la taille du serveur si le nombre de connexion sur le site internet augmente.

Les serveurs seront hébergés sur  **HEROKU**  
un Hébergeur simple, rapide et infogéré.

### **Modèle physique de données**

Le modèle physique de donnée présenté ci-dessous se base sur le diagramme de classe détaillée dans les pages 8 à 12.



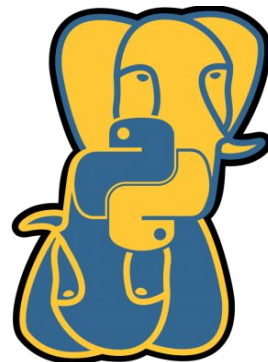
## Le serveur d'application

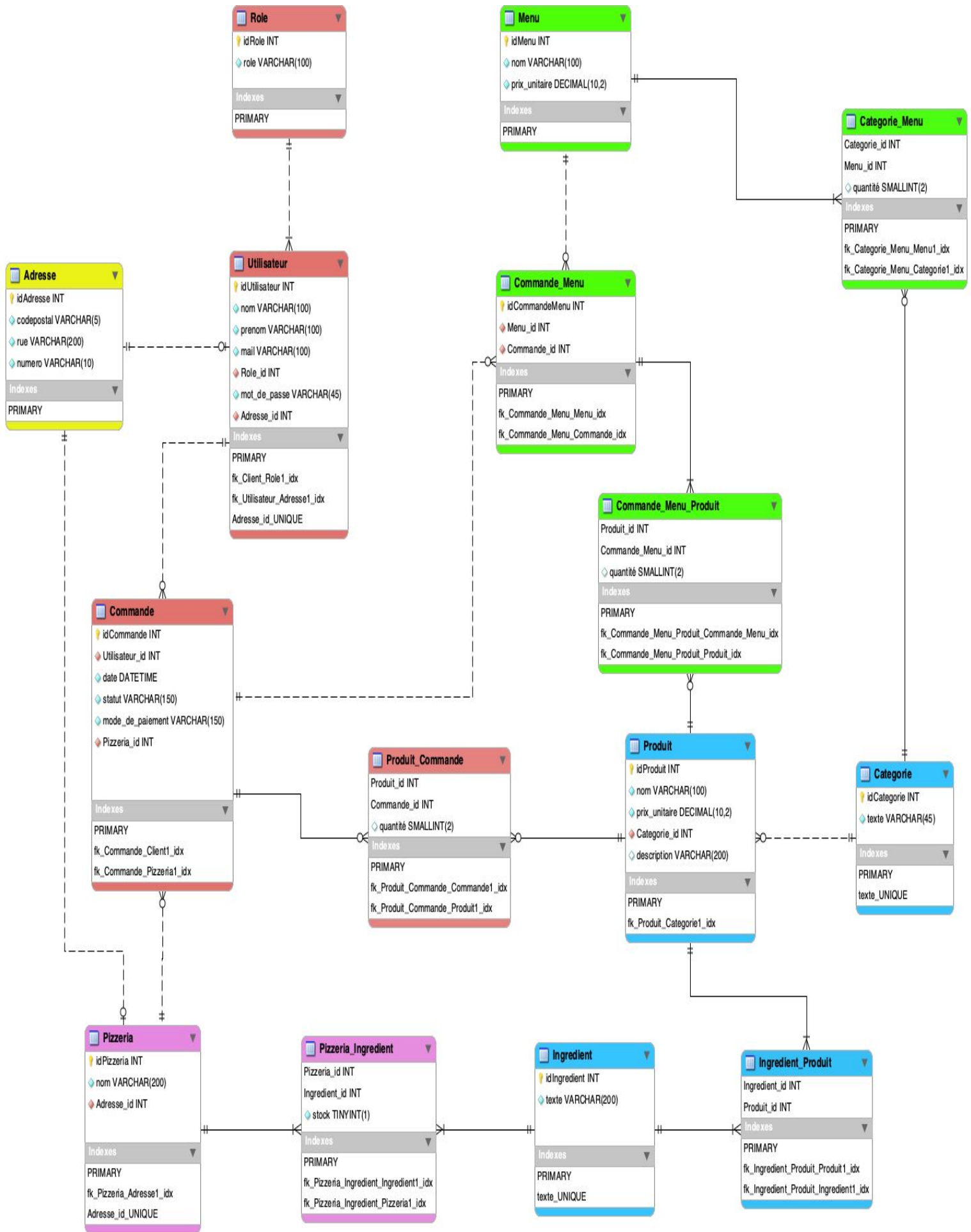
- Synthèse

### Front-End



### Back-End





## Les composants externes

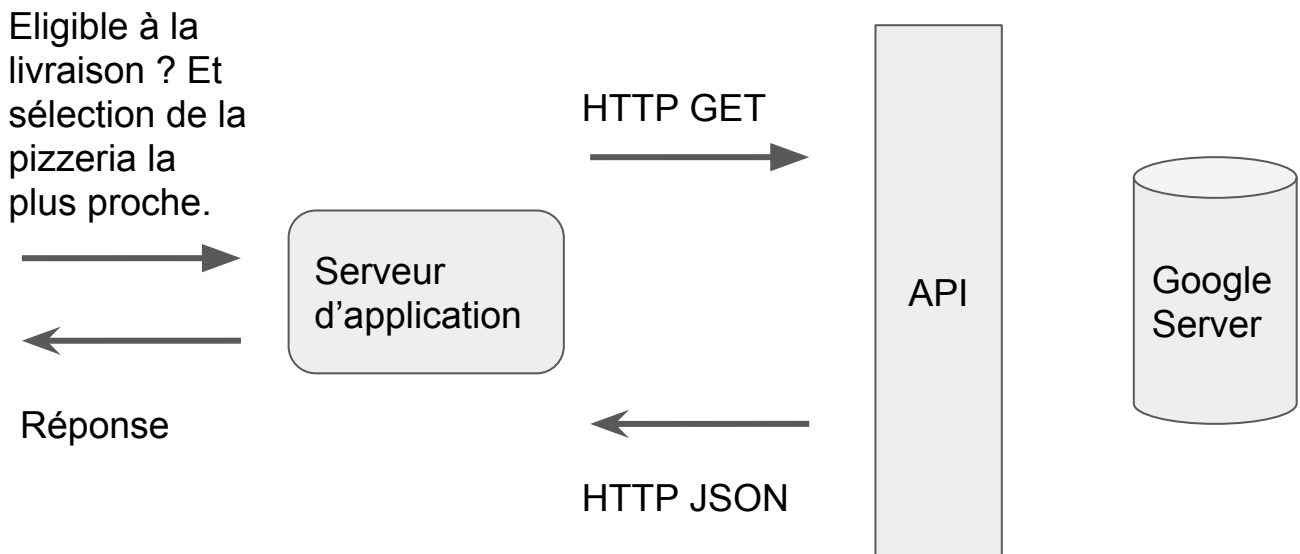
- Explication d'utilité

### Google Maps Distance Matrix API:

Afin de localiser la position du client par rapport aux différentes pizzeria, l'application va communiquer avec une API de géolocalisation. Nous avons sélectionné Google Maps Distance Matrix API.

L'idée serait d'envoyer une requête HTTP contenant l'adresse du client et des différentes adresses des pizzerias à l'API. Nous attendons en retour des distances calculée par l'API nous permettant de sélectionner la pizzeria la plus proche et confirmer ou non l'éligibilité du client pour une livraison basée sur un critère de « distance maximale de livraison ».

### Synthèse:



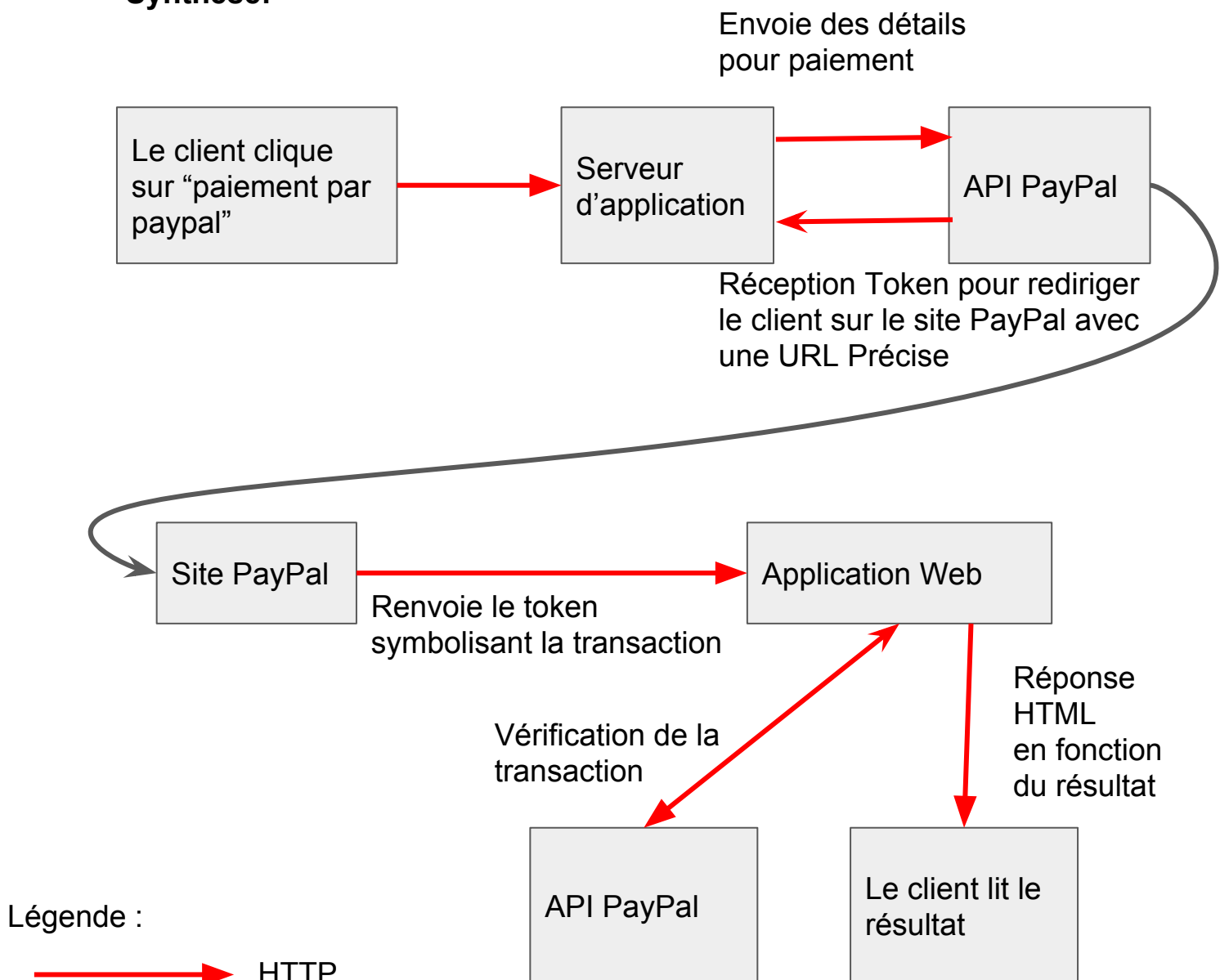
## Les composants externes

- Explication d'utilité (suite)

### API PayPal

Le client clique sur « Paiement avec PayPal » lors de la création de la commande, nous faisons donc appel au service PayPal en lui indiquant dans une requête http les informations de paiement, celui-ci nous réponds en nous envoyant un « token » qu'il nous ait possible d'utiliser pour rediriger le client sur le site PayPal afin de finaliser la transaction.

### Synthèse:



- **Glossaire**

**HTTP** : Hypertext Transfer Protocol

**WSGI** : Web Server Gateway Interface (**WSGI**) est une spécification qui définit une interface entre des serveurs et des applications web pour le langage Python.

**TCP / IP** : TCP (Transmission Control Protocol) et IP (Internet Protocol).