



SPÉCIFICATIONS TECHNIQUES

SOMMAIRE :

I.....	CONTEXTE
II.....	DESCRIPTION DU DOMAINE FONCTIONNEL
III.....	MODÈLE PHYSIQUE DE DONNÉES
IV.....	LES DIFFÉRENTS COMPOSANTS INTERNES ET EXTERNES DU SYSTÈME
V.....	ARCHITECTURE DE DÉPLOIEMENT

I. CONTEXTE :

OC Pizza :

Jeune groupe de pizzeria en plein essor et spécialisé dans les pizzas livrées ou à emporter. Compte 5 points de vente et prévoit d'en ouvrir au moins 3 d'ici la fin de l'année.

1. Objectif :

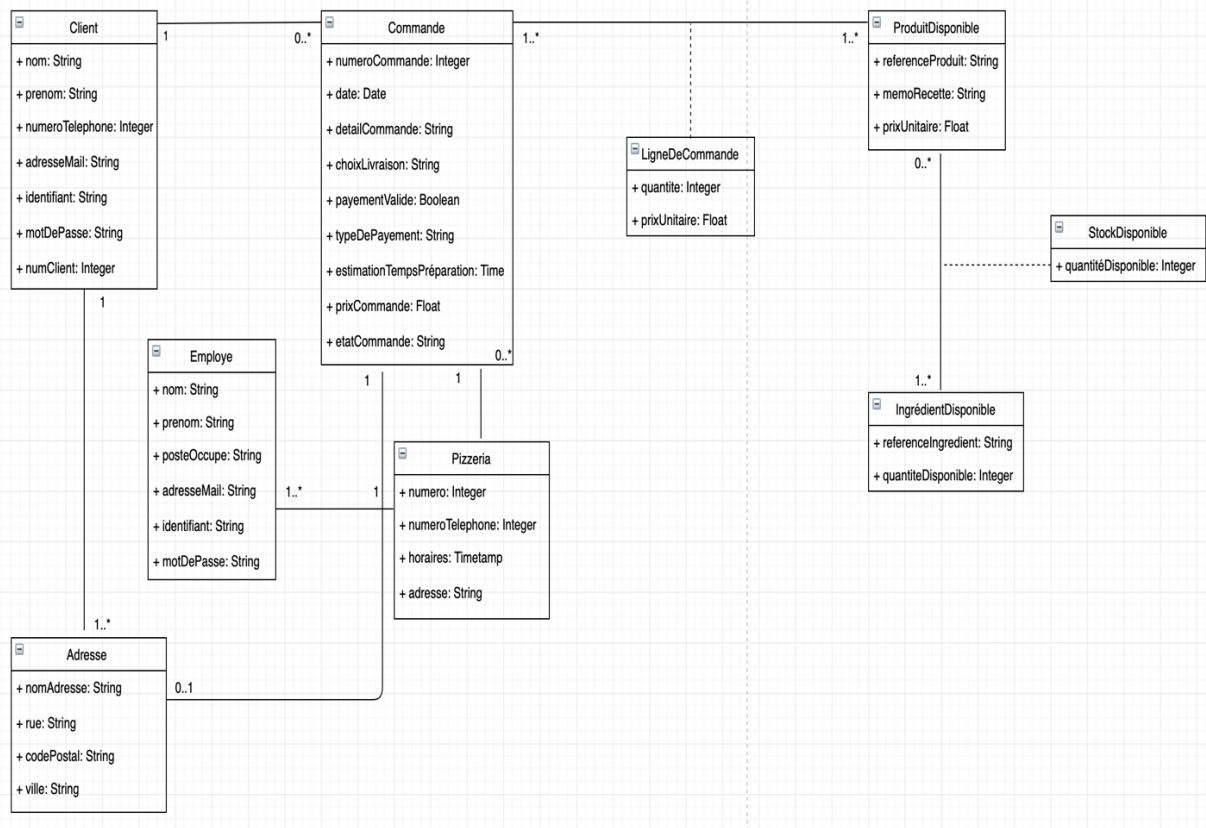
Réaliser une application web pour faciliter la gestion des pizzerias du groupe OC Pizza et permettre aux clients de passer leurs commandes en ligne.

2. Fonctionnalités attendues :

- Suivre en temps réel les commandes passées et en préparation
- Suivre en temps réel le stock d'ingrédients restants pour savoir quelles pizzas sont encore réalisables
- Passer des commandes en ligne
- Possibilité de payer en ligne
- Modifier ou annuler une commande tant que celle-ci n'a pas été préparée
- Proposer un aide-mémoire au pizzaïolo indiquant la recette de chaque pizza
- Informer ou notifier les clients sur l'état de leur commande

II. DESCRIPTION DU DOMAINE FONCTIONNEL :

Nous allons dans un premier temps définir les différents objets du domaine fonctionnel sous forme de diagramme de classe ce qui nous permettra par la suite de construire le modèle physique de données qui répondra à notre solution technique.



L'objet «**Client**» est lié à la commande et à l'adresse. Ses attributs permettent l'authentification du client.

L'objet «**Commande**» va permettre d'avoir un aperçu des différentes commandes, leur état de préparation et de livraison mais aussi permettre au manager de faire des statistiques sur l'activité des pizzerias pour ainsi améliorer l'efficacité de son service.

L'objet «**Adresse**» est lié au client et à la commande. Ainsi le client pourra renseigner plusieurs adresses de livraison et facturation.

L'objet «**Employe**» est lié à la pizzeria dans laquelle l'employé va travailler. Elle possèdera autant d'instance que d'employé.

L'objet «**Pizzeria**» possède pour le moment 5 instances car le groupe possède 5 pizzerias.

L'objet «**ProduitDisponible** » représente les différents types de produits que peuvent être vendus dans les pizzerias, en effet les pizzerias peuvent vendre des desserts et boissons par exemple.

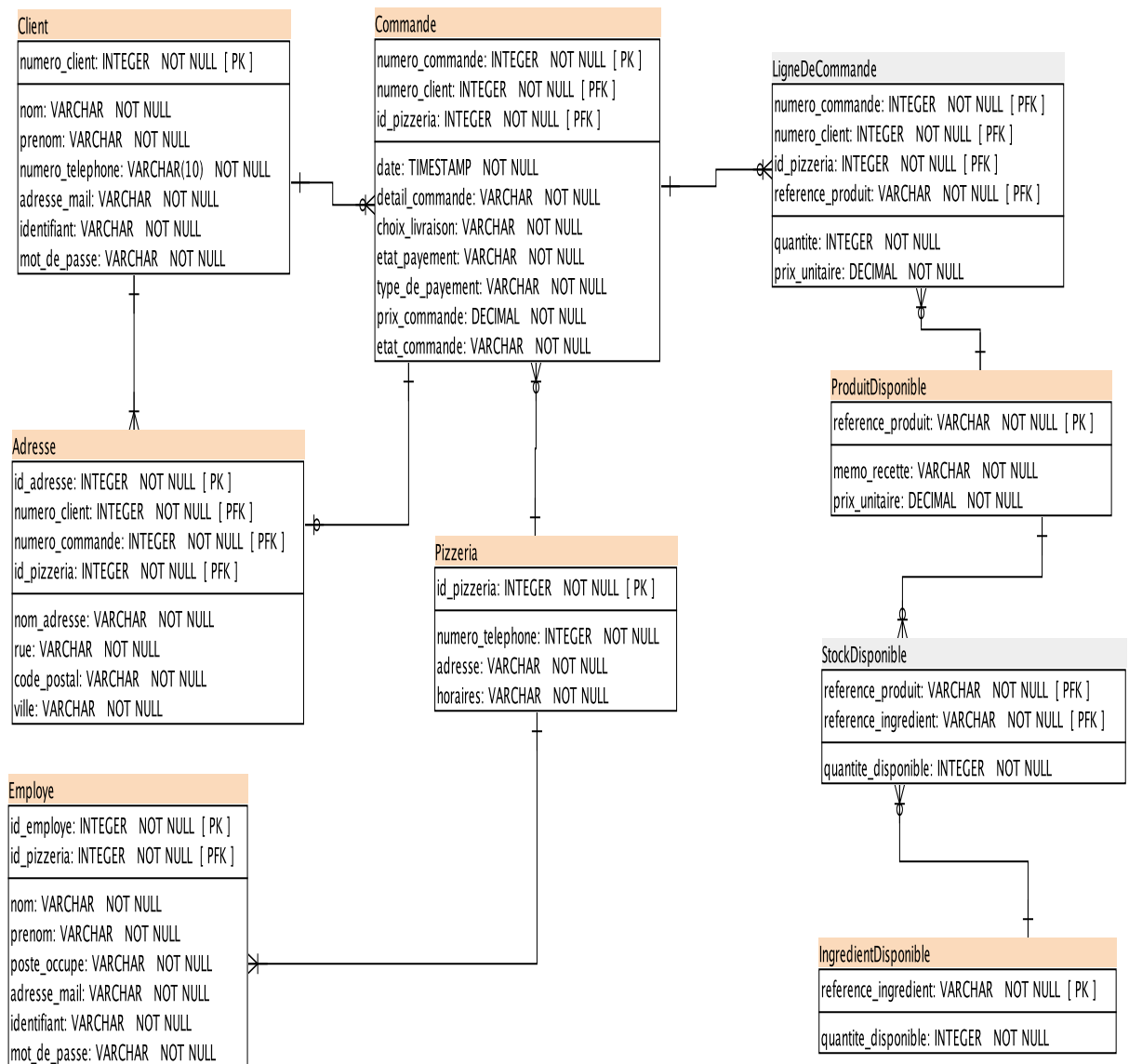
L'objet «**LigneDeCommande** » est une classe d'association qui regroupe les classes « Commande » et « ProduitDisponible ». En effet une commande peut être constituée de plusieurs produits et un produit peut appartenir à plusieurs commandes.

L'objet «**IngrédientDisponible** » correspond aux ingrédients présents et disponibles dans chaque pizzeria. Cette classe permet à chaque pizzeria et au groupe d'avoir un aperçu de son stock réel.

L'objet «**StockDisponible** » est une classe d'association regroupe les classes « ProduitDisponible » et « IngrédientDisponible » représente la quantité nécessaire d'ingrédient pour un produit. En effet un produit peut être constitué de plusieurs ingrédients et un ingrédient peut appartenir à plusieurs produits.

III. MODÈLE PHYSIQUE DE DONNÉES :

Le modèle physique de données va permettre de modéliser dans le détail la base de données relationnelle OC Pizza.



IV. LES COMPOSANTS INTERNES ET EXTERNES DU SYSTÈME :

Les Diagrammes de composants décrivent l'organisation du système du point de vue des éléments logiciels. Ils mettent en évidence les dépendances entre les composants, et décrivent ici les interfaces entre les composants internes du système OC Pizza et les composants externes.

1. Les composants internes :

1.1 Navigateur client :

Le navigateur client est un navigateur HTML standard compatible avec les formulaires et le langage JavaScript. Il s'agit d'un dispositif universel d'interface utilisateur.

Ainsi les navigateurs web classique tels que Firefox, Google Chrome, Internet explorer, Safari sont utilisables pour accéder à l'application.

1.2 Serveur d'application :

Nous utiliserons un serveur programmé en PHP en utilisant le framework Symfony. Symfony est un outil flexible, évolutif, stable et performant il intègre des design patterns reconnus et prouvés. Il peut répondre à n'importe quelle problématique technique et offre un cadre solide.

1.3 Serveur web :

Le système d'exploitation utilisé pour le serveur d'application sera Apache (opensource et gratuit).

Le protocole de communication entre notre projet Symfony et le serveur Apache est HTTP.

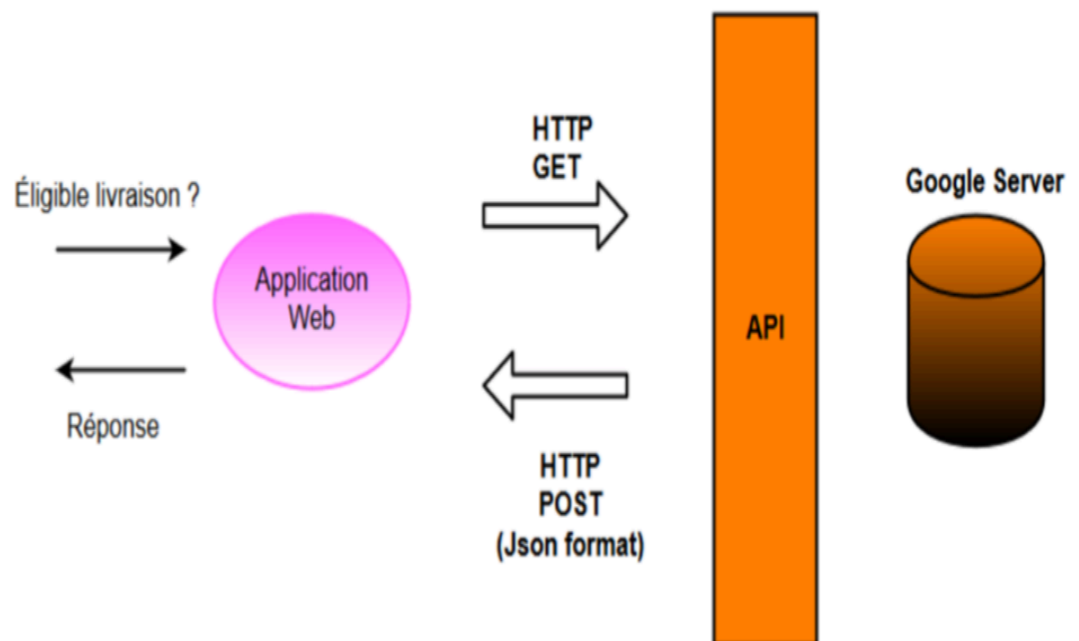
1.4 Serveur de données :

Nous utiliserons le système de gestion de base de données relationnelles PostgreSQL.

2. Les composants externes :

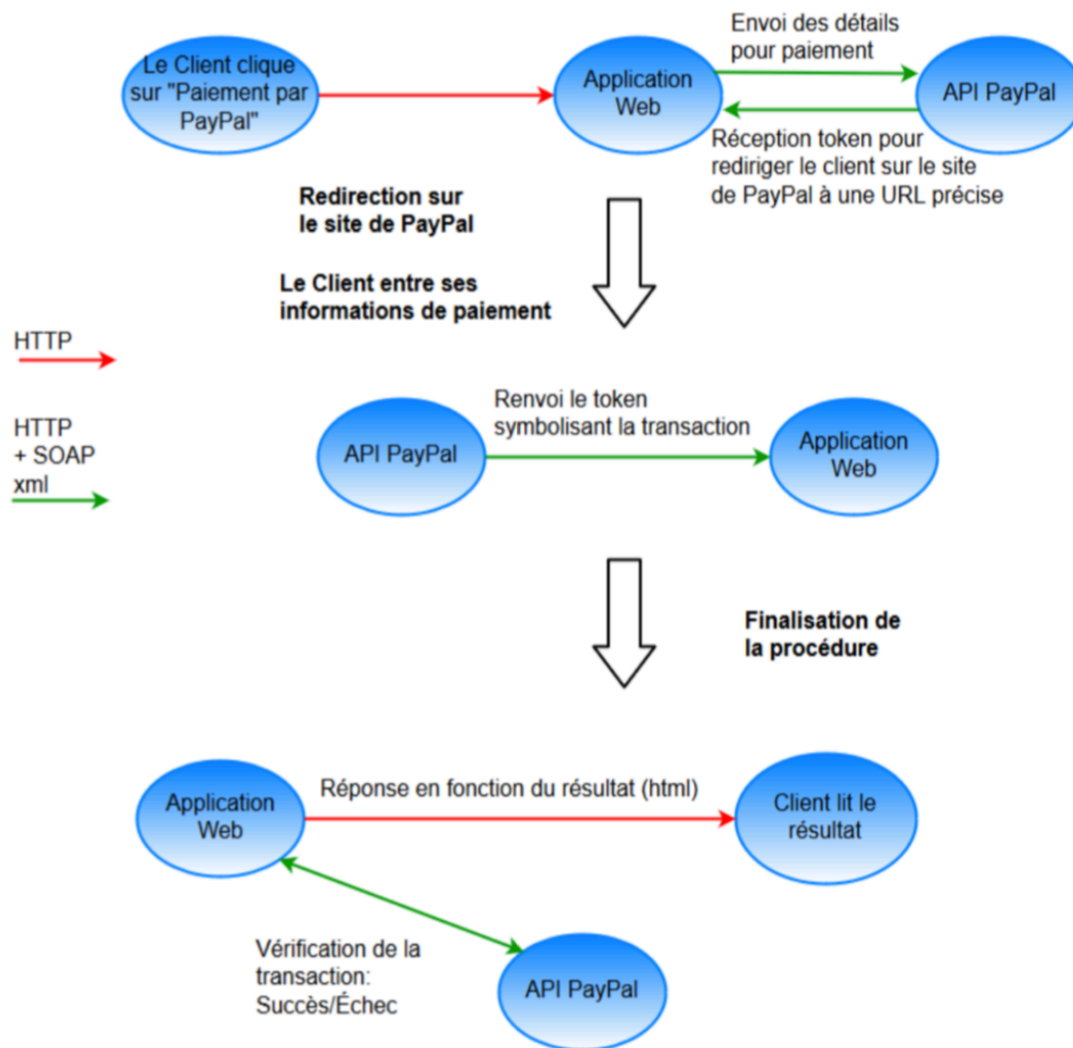
1. API Google Maps :

Afin de localiser la position du client par rapport à la pizzeria dans laquelle il souhaite commander, l'application se doit de communiquer avec une API de géolocalisation. Une requête http serait envoyée à l'API contenant l'adresse du client potentiel et l'adresse de la pizzeria en question. En retour une distance serait calculée par l'API afin confirmer ou non l'éligibilité du client pour une livraison.

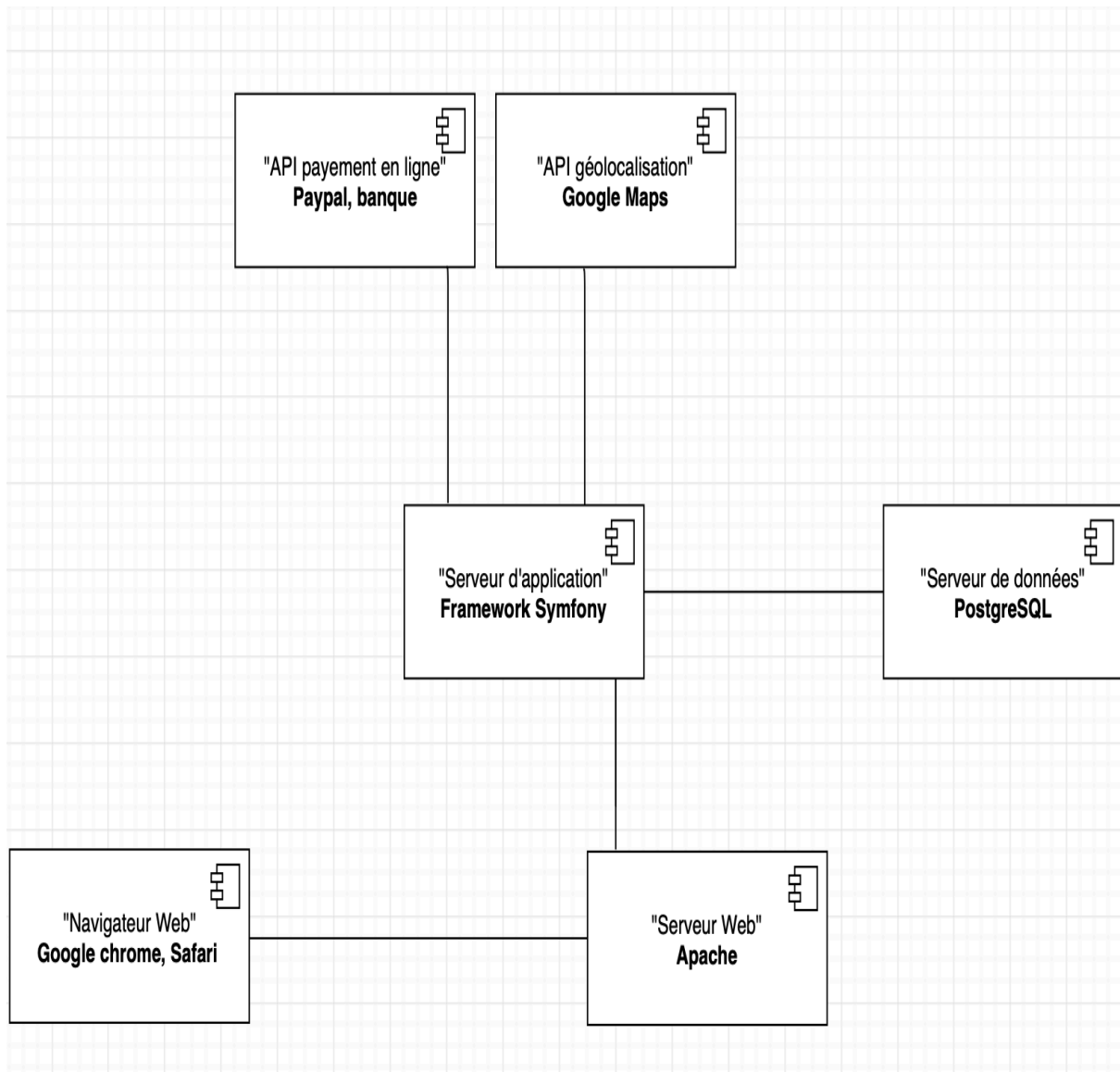


2. API Paypal :

Le système de paiement utilisé est Paypal. Pour communiquer avec l'API de Paypal, nous utiliserons le protocole SOAP et enverrons/recevrons des fichiers XML.



3. Diagramme des composants :



V. ARCHITECTURE DE DEPLOIEMENT :

