OC Pizza

Mise en place d'un nouveau système informatique pour l'ensemble des pizzerias du groupe.

Dossier de conception technique

Version 1

Auteur
Cossu Denis
Développeur d'applications Android

TABLE DES MATIÈRES

1 - VERSIONS

Auteur	Date	Description	Version
Cossu Denis	29/08/19	Création du document	1

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier de conception technique de l'application OC Pizza.

L'objectif du document est de faire l'analyse technique des besoins du client pour mettre en place un nouveau système informatique pour l'ensemble des pizzerias du groupe OC Pizza.

Les éléments du présent dossier découlent des besoins du client.

2.2 - Références

Pour de plus amples informations, se référer également aux éléments suivants :

1. Le dossier de conception fonctionelle de l'application
2. Le dossier d'exploitation de l'application

2.3 - Contexte

«OCPizza» est un jeune groupe de pizzeria en plein essor. Créé par Franck et Lola, le groupe est 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 6 mois. Le système informatique actuel ne correspond plus aux besoins du groupe car il ne permet pas une gestion centralisée de toutes les pizzerias. De plus, il est très difficile pour les responsables de suivre ce qui se passe dans les points de ventes. Enfin, les livreurs ne peuvent pas indiquer « en live » que la livraison est effectuée.

2.4 - Enjeux et Objectifs

Enjeux :

-Etre plus efficace dans la gestion des commandes, de leur réception à leur livraison en passant par leur préparation.

-Suivre en temps réel les commandes passées, en préparation et en livraison.

-Suivre en temps réel les stocks d'ingrédients restants pour savoir quelles pizzas peuvent encore être réalisées.

-Proposer un site internet pour que les clients puissent :

-Passer leur commande, en plus de la prise de commande par téléphone ou sur place.

-Payer en ligne leur commande s'ils le souhaitent – sinon, ils payeront directement à la livraison.

-Modifier ou annuler leur commande tant que celle-ci n'a pas été préparée.

-Proposer un aide-mémoire aux pizzaiolos indiquant la recette de chaque pizza.

Objectif :

Réaliser le nouveau système informatique avant l'ouverture des trois nouvelles pizzerias dans 6 mois.

(Voir document de spécifications fonctionnelles pour plus de détails)

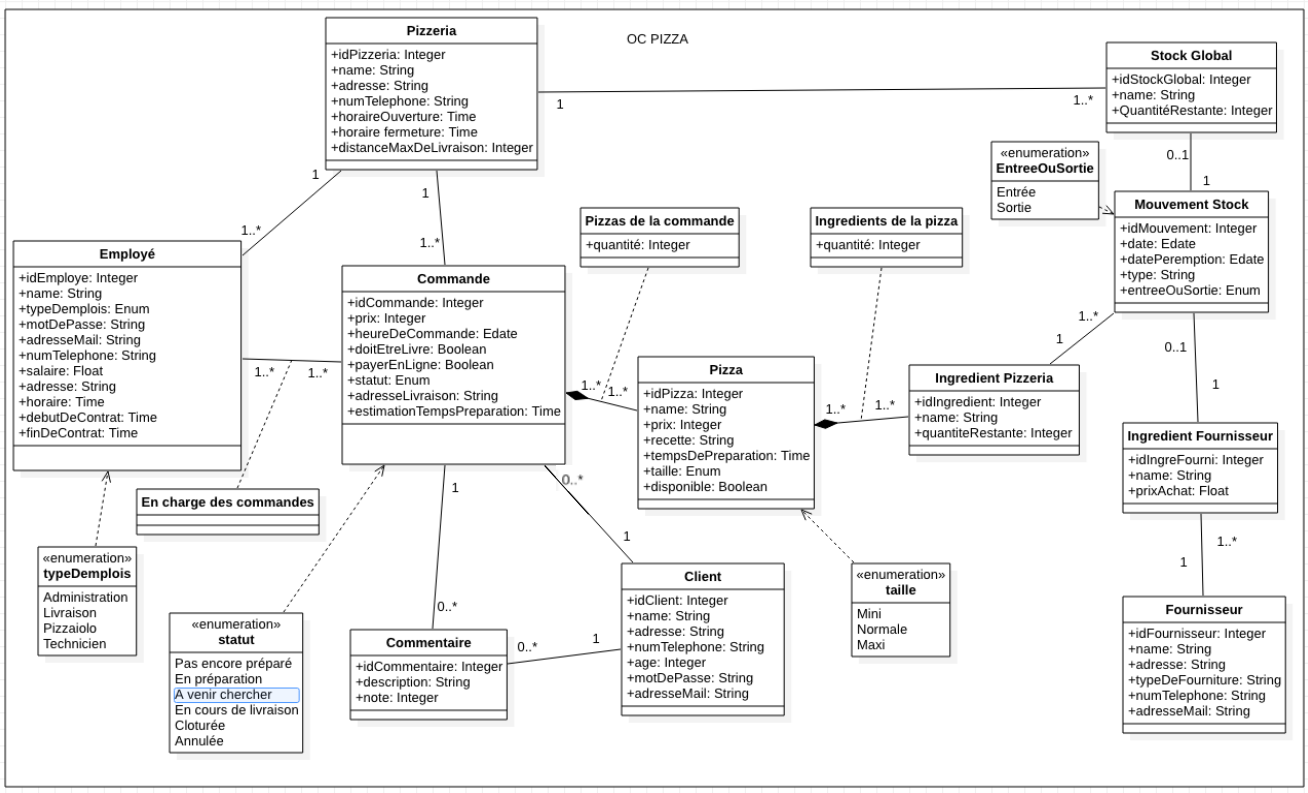
3 - Diagramme de classes et modèle physique de données

La persistance des données sera gérée à l'aide des bases de données relationnelles. Pour mettre en place ces bases de données notre choix se porterait plus sur PostgreSQL qui est performant, robuste, stable et open-source. Pour aider à mettre en place cette persistance des données nous avons réalisé un diagramme de classe ainsi qu'un modèle physique de donnée.

3.1 - Diagramme de Classes

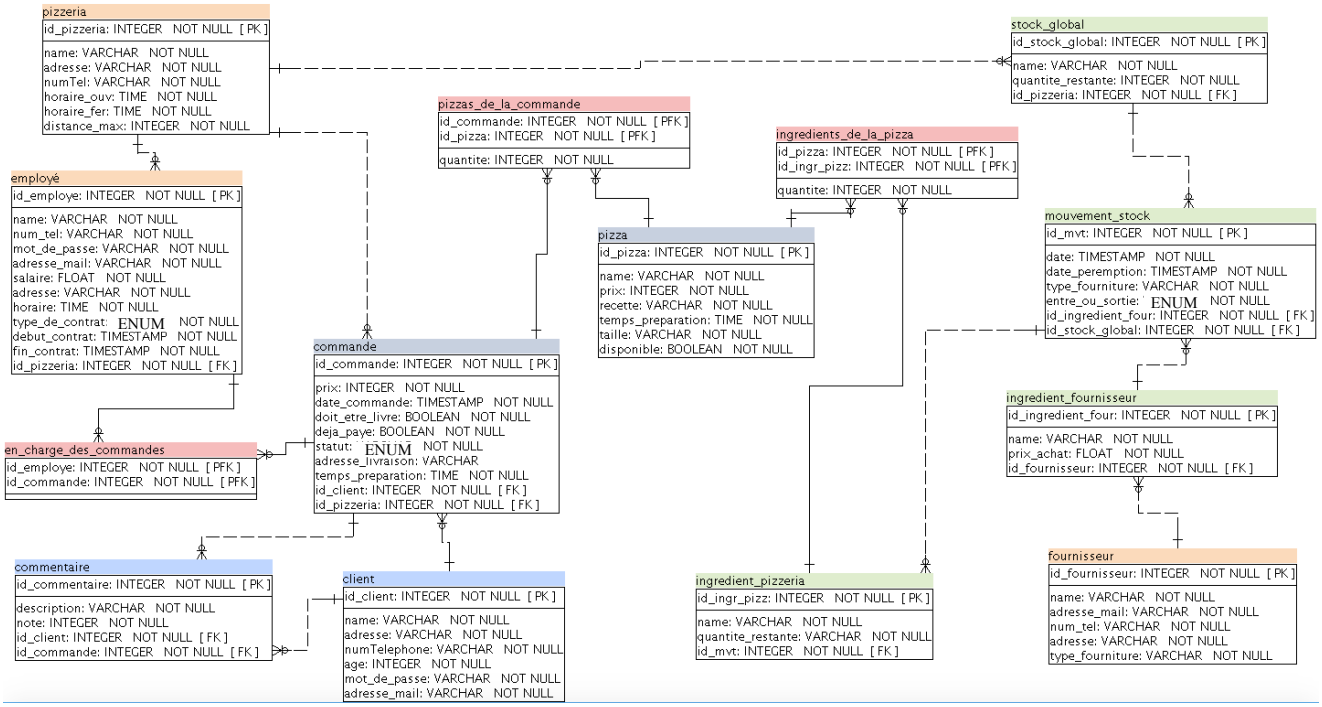
Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

Ce diagramme aidera notamment à l'élaboration du modèle physique de données du nouveau système informatique de OC Pizza.



3.2 - Modèle physique de données

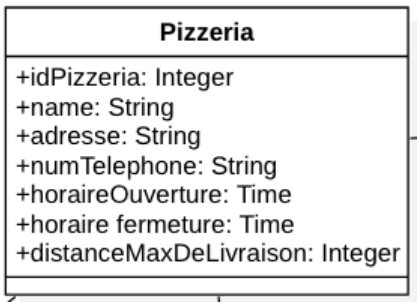
Le modèle physique de données est réalisé avec SQL Power Architect qui permet de générer automatiquement la base de données.



4 - Domaine fonctionnel (Description)

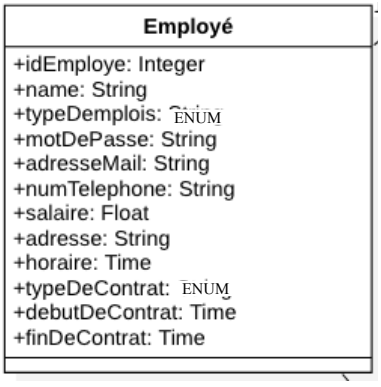
Au sein de cette partie, nous allons voir ensemble les descriptions des différentes composantes du domaine fonctionnel qui vont permettre l'élaboration du modèle physique de données adapté à la solution technique souhaitée.

4.1 - Pizzeria



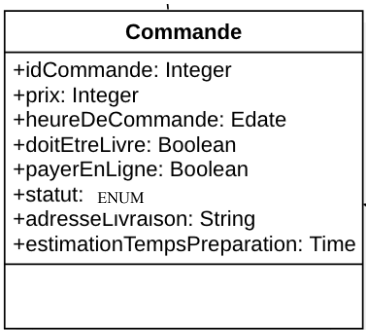
Comme le groupe OC Pizza est composé pour l'instant de 5 pizzerias, il y aura donc 5 instances de la classe Pizzeria. Chacune des instances est reliée aux classes 'Employé', 'Commande', 'Stock Global'... permettant de s'assurer de la bonne des gestion des pizzerias.

4.2 - Employé



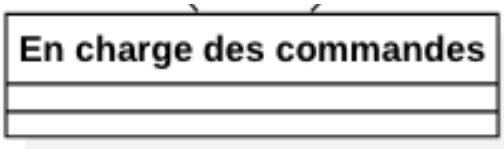
Un employé peut être un membre de l'administration, un technicien, un pizzaiolo ou un livreur. La classe 'Employé' est lié à une seule pizzeria et peut selon le type d'emploi peut être lié à une ou plusieurs commandes via la classe 'En charge des commandes'.

4.3 - Commande



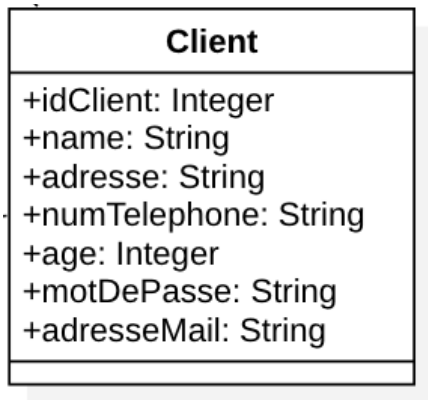
Une commande a lieu dans une pizzeria par un client. Le client peut si il le souhaite lui associer un commentaire. Son statut ('pas encore préparé', 'en préparation', 'à venir cherché', 'en cours de livraison', 'annulée', 'cloturée') est disponible en temps réel. Si le client a payé lors de la commande et qu'il annule sa commande il est remboursé. Si il n'a pas payé à ce moment il doit payer lors de la réception de la commande. La commande peut être gérée par un ou plusieurs employés de la pizzeria via la classe 'En charge des commandes'.

4.4 - En charge des commandes



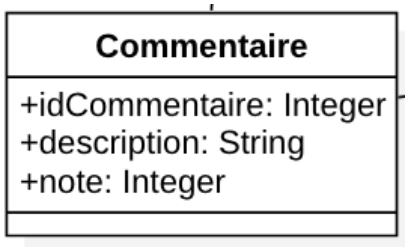
Classe d'association matérialisant la relation « plusieurs à plusieurs » (many-to-many) entre les employés de la pizzeria et les commandes. En effet un employé peut s'occuper de plusieurs commandes et une commande peut être gérée par plusieurs employés.

4.5 - Client



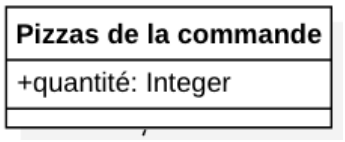
Le client est la cible principale de l'application, il peut passer une ou plusieurs commandes à une pizzeria et peut associer un commentaire à chaque commande si il le souhaite. Il peut donner une autre addresse que la sienne lors de la commande si il le souhaite.

4.6 - Commentaire



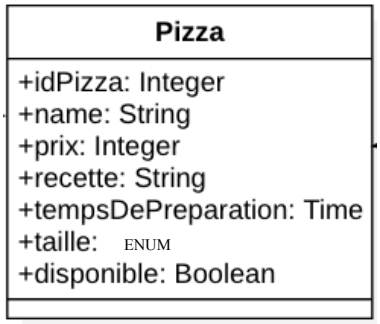
Un commentaire peut être laissé par un client lors d'une commande, il peut donner une note à sa commande, dire ce qu'il en a pensé. Si un client énonce un problème, l'équipe de techniciens corrigera le problème ou le fera remonter aux personnes concernées.

4.7 - Pizzas de la commande



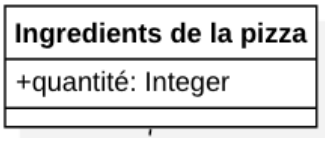
Classe d'association matérialisant la relation « plusieurs à plusieurs » (many-to-many) entre les pizzas et les commandes. En effet un même type de pizza peut se trouver dans diffèrentes commandes et une commande peut être composée de plusieurs pizzas. Avec l'attribut 'quantité' le client peut commander plusieurs fois la même pizza dans sa commande.

4.8 - Pizza



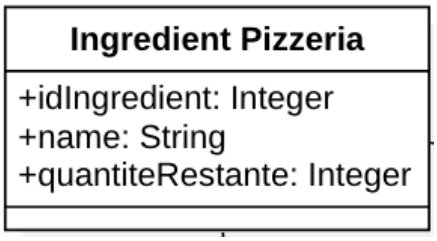
Une pizza est composée de plusieurs ingrédients et une ou plusieurs plusieurs pizzas peuvent former une commande. Chaque pizza est associée à un nom, un prix, un temps de préparation, et une recette pourra potentiellement servir d'aide-mémoire pour le pizzaiolo. Les pizzas seront disponibles en diffèrentes tailles par exemple 'mini', 'normal' et 'maxi'. Si le nombre d'ingrédient pour faire la pizza n'est pas suffisant la pizza devient indisponible.

4.9 - Ingredients de la pizza



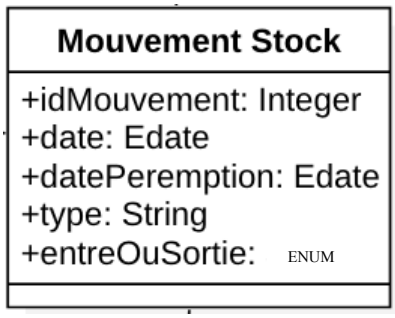
Classe d'association matérialisant la relation « plusieurs à plusieurs » (many-to-many) entre les pizzas et les ingrédients. En effet une pizza peut être composé de plusieurs ingrédients et un ingrédient peut être utilisé dans diffèrentes pizzas. L'attribut 'quantité' indique le nombre d'un même type d'ingrédient utilisé pour la pizza, lors de la réalisation d'une pizza ce nombre sera déduit du stock d'ingrédients.

4.10 - Ingredient Pizzeria



Cette classe représente les stocks d'ingrédients d'une pizzeria, quand la quantité restante d'un ingrédient devient faible, un mouvement du stock global est lancé pour remplir à nouveau les stocks d'ingrédients de la pizzeria.

4.11 - Mouvement Stock

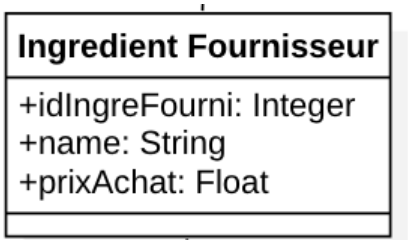


Cette classe permet de renseigner sur les entrées et les sorties du stock global de OC Pizza, on a ainsi la date du mouvement du stock, si le mouvement est une entrée ou une sortie du stock global, le type de stock (quel ingrédient ou lot de plusieurs ingrédients). Les stocks avec des dates de peremption courtes seront à déplacer et à utiliser en premier.

4.12 - Stock Global

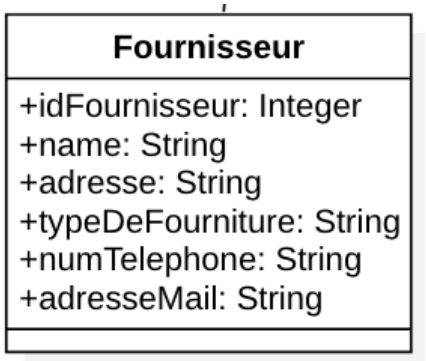
Cette classe représente le stock global d'ingrédients de OC Pizza. Quand la quantité restante d'un ou plusieurs ingrédients devient faible, un achat est lancé envers les fournisseurs pour remplir à nouveau les stocks. Si les stocks ne sont pas remplis assez vite les pizzas concernées par le manque d'ingrédients deviennent indisponibles.

4.13 - Ingredient Fournisseur



Cette classe représente les ingrédients pouvant être achetés à un fournisseur. Le prix d'achat est à prendre en compte avant tout achat pour éviter les dépenses inutiles.

4.14 - Fournisseur



Cette classe représente les fournisseurs qui vont permettre de remplir les stocks d'ingrédients des pizzerias lorsqu'ils sont proches de l'épuisement. Les pizzerias n'ont pas forcément le même fournisseur pour les mêmes ingrédients, en effet puisqu'il y a 5 pizzerias (et de nouvelles ouvriront) elles ne sont pas localisées au même endroit, selon leur localisation un fournisseur peut-être préférable à un autre.

5 - Les composants du système (internes et externes)

5.1 - Serveur de données (interne)

Nous avons choisi Linux comme système d'exploitation qui est opensource contrairement à UNIX par exemple. Comme dit précedemment nous avons choisi PostgreSQL comme système de gestion de base de données relationnelles. La base de données portera le nom de l'entreprise c'est à dire OC Pizza, une seule base de données sera utilisée pour les diffèrentes pizzerias.

5.2 - Serveur web (interne)

Linux sera également le système d'exploitation utilisé pour nos serveurs web, pour le serveur web nous utiliserons Apache qui est très connu, qui est open-source et plus performant que Nginx pour les contenus dynamiques par exemple. Pour le serveur HTTP nous utiliserons également uWSGI qui est simple à configurer qui demande peu d'espace à allouer et qui est rapide. Pour conserver les avantages de la programmation objet en python le protocole de communication entre Apache et uWSGI sera WSGI ce qui permettre aux requètes HTTP d'être transformées en objet Python.

5.3 - Serveur d'application (interne)

Le serveur sera programmé en Python avec Django qui est un framework opensource qui permet de mettre au point rapidement des applications webs de qualité.

Les bonnes pratiques de la PEP 8 seront respéctées (Une ligne doit contenir 80 caractères maximum. L'indentation doit être de 4 espaces. Ajoutez deux lignes vides entre deux éléments de haut niveau, des classes par exemple, pour des questions d'ergonomie. Séparez chaque fonction par une ligne vide. Les noms ne doivent pas contenir d'accent...)

Nous utiliserons HTML, CSS et JavaScript ainsi que Jquery et les requètes AJAX pour engendrer les pages HTML.

- CSS et JavaScript vont permettre de soigner la présentation des pages webs, de les rendre interactives,

fluides et ergonomiques lorsqu'elles seront demandées par l'utilisateur.

- jQuery est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web.

- L'architecture informatique AJAX permet de construire des applications Web et des sites web dynamiques interactifs sur le poste client. Elle permettra d'obtenir des réponses rapides en limitant les échanges client/serveur.

5.4 - Navigateur client (interne)

Le navigateur client est un navigateur HTML classique qui peut être utilsé avec les formulaires et langages HTML, CSS et JavaScript. Firefox, Internet Explorer, Safari, Google Chrome et bien d'autres navigateurs webs connus pourront être utilisés pour accèder à l'application. Cet accès marchera tout aussi bien que ce soit sous OS Windows, MacOS ou Linux. Ainsi l'utilisateur pourra passer rapidement d'une page HTML envoyée par le serveur à une autre à l'aide du navigateur. Dans la fenêtre client, les pages obtenues par le navigateur auront une interface adaptées pour une utilisation simple et rapide.

5.5 - Localisation (externe)

Notre base de données PostgreSQL envoie à l' API REST Google Maps Directions les données telles que l'adresse des diffèrentes pizzerias ainsi que les adresses des clients d'OC Pizza. Ces informations permettront de calculer les trajets optimals (et le temps estimé pour les parcourir) entre les clients et les pizzerias. Cela permettra aux clients de voir les pizzerias les plus proches et de s'y rendre rapidement si ils le souhaitent et aux livreurs de gagner du temps sur leur livraison. Cela permettra également au client d'avoir une estimation du temps prévu pour la livraison (temps de préparation + temps de trajet) avant même qu'il ne valide sa commande. Si le trajet est trop long (basé sur la 'DistanceMaxDeLivraison') le client ne pourra pas demandé à être livré par cette pizzeria.

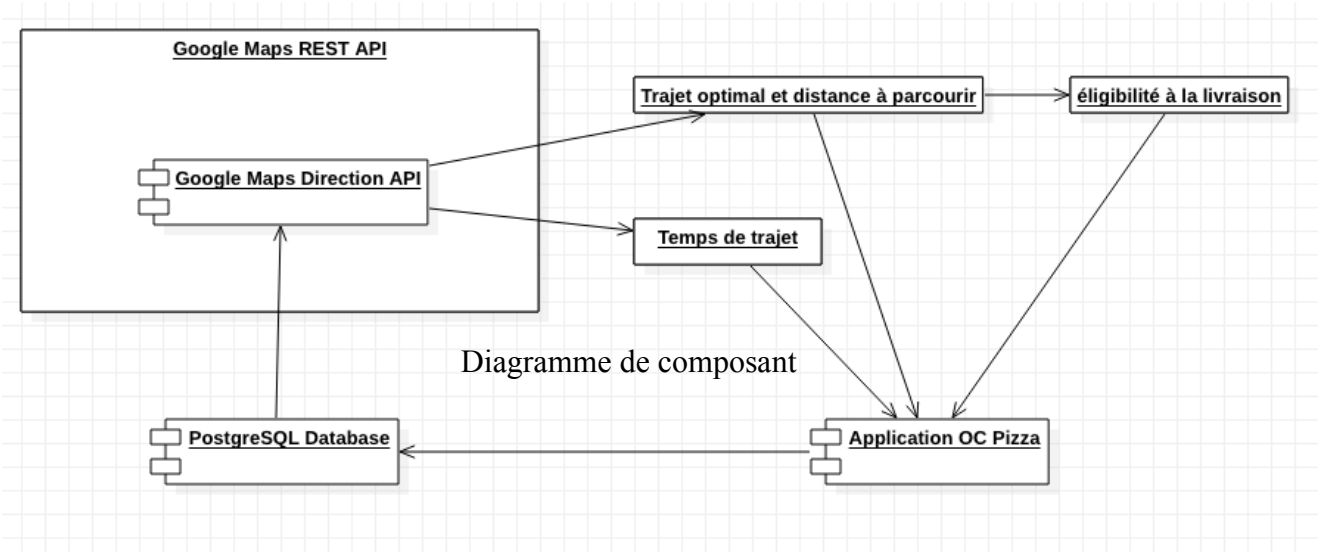
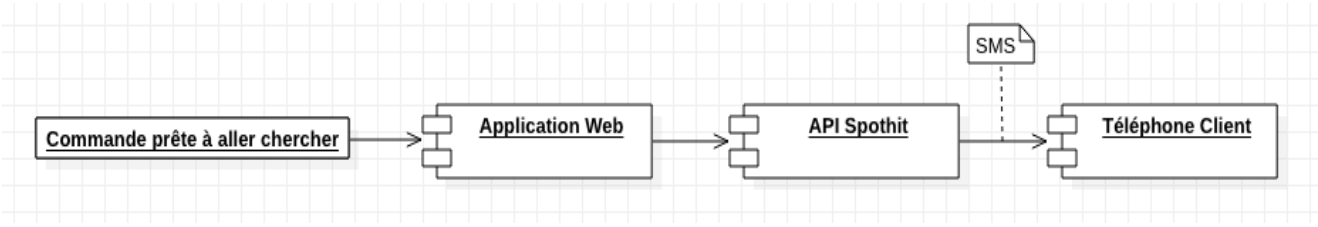


Diagramme de composant

5.6 - Notification (externe)

Pendant une commande, les employés peuvent recevoir des notifications (ne pas oublier telle commande, rappeler telles informations...) ainsi que le client (commande prêtes à aller chercher...). Pour automatiser l'envoi des notifications sous forme de SMS nous utiliserons l'API Spothit. L'utilisateur de l'API a juste besoin d'écrire dans un fichier Json les informations nécessaire à l'envoi du message comme le numéro du destinataire du message et le contenu de la notification. Les SMS sont ensuites envoyés via la méthode POST à l'aide de requêtes HTTP.

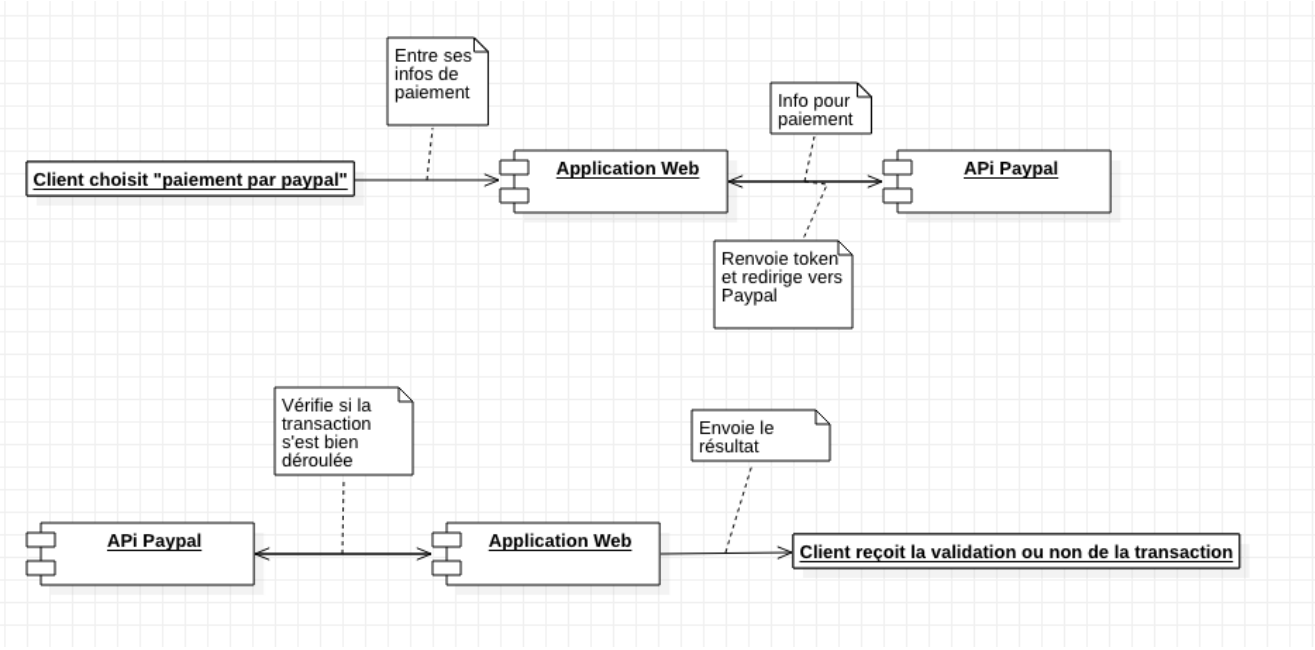
Schema d'un exemple de fonctionnement de l'API Spothit :



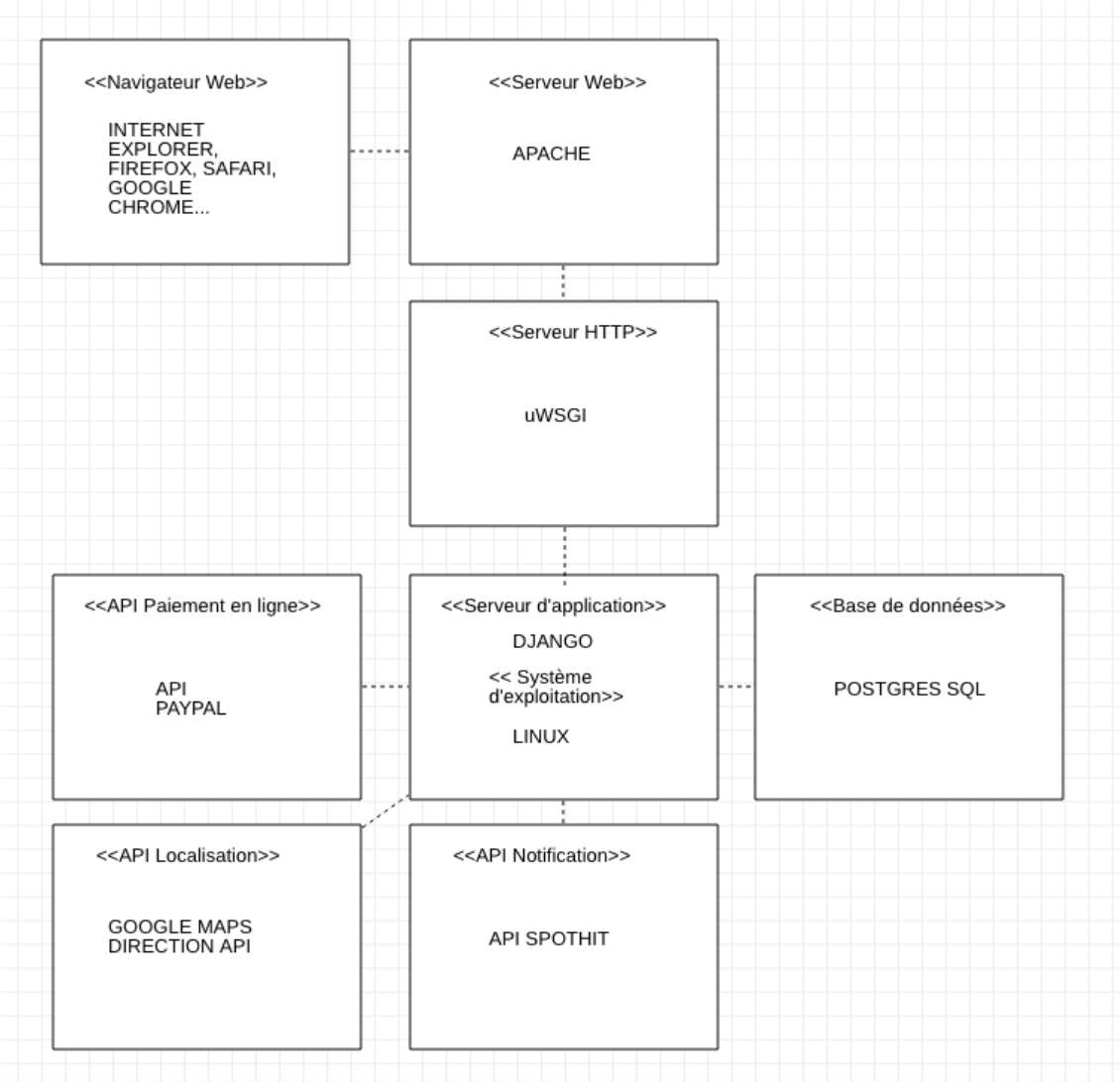
5.7 - Paiement (externe)

Pour payer par internet nous avons choisi d'utiliser Paypal comme mode de paiement, nous utiliserons le protocole SOAP qui permettra d'envoyer et de recevoir des fichiers XML pour communiquer avec l'API Paypal. Cet API marche avec des tokens (permet de générer des numéros en fonctions d'un algorithme précis, et permet une authentification sur un serveur d'un utilisateur, en utilisant un mot de passe unique (chaque numéro aléatoire n'est généré qu'une seule fois)) : nous envoyons à l'API un fichier XML avec les informations nécessaires au paiement et nous recevons un token qui permet de rediriger le client sur le site paypal pour lui permettre de payer.

Schéma simplifié du fonctionnement :



5.8 - Schéma récapitulatif



6 - ARCHITECTURE DE DÉPLOIEMENT

6.1 - Introduction

Après avoir illustré les composants internes et externes du système informatique avec les Diagrammes de composants, nous allons identifier les éléments physiques et leur connexions entre eux pour compléter la conception architecturale avec le Diagramme de déploiement UML.

6.2 - Matériel utilisateur

Du moment qu'il y a un accès à internet n'importe quel appareil (tablette, ordinateur, téléphones...) appartenant à un utilisateur (que ce soit un client ou un membre d'une pizzeria) peut accèder à l'application. En effet l'application sera conçue pour fonctionner avec n'importe quel système d'exploitation et avec la grande majorité des navigateurs.

Toutefois pour recevoir les notifications les utilisateurs auront également besoin d'un smartphone pouvant recevoir des SMS sur un numéro valide.

6.3 - Hébergement des serveurs

Pour ne pas héberger les serveurs en interne et pour faciliter les potentiels modification à faire à l'avenir (comme augmenter la taille des serveurs si OC Pizza venait à ouvrir encore plus de pizzeria par exemple) nous avons choisi d'héberger les serveurs sur le cloud. Nous pourrions par exemple nous servir d'OVH, une entreprise française spécialisée dans les services de cloud computing, pour héberger nos serveurs.

6.4 - Diagramme de déploiement