



SPECIFICATIONS TECHNIQUES OC PIZZA

RESUME

Ce document décrit la spécification technique suivant la suite d'une analyse de vos besoins



OC PIZZA

Projet 06 :

Concevez la solution technique d'un système de gestion
de pizzeria

Dossier de conception technique version 1.0

Sulyvan Dercourt



TABLE DES MATIERES

1 -Versions	3
2 -Introduction	4
2.1 -Objet du document	4
2.2 -Références	4
2.3 - Modèles fonctionnelle	5
2.3.1 - Règle de gestion	5
3 -Architecture Technique.....	7
3.1 -Composants généraux	7
3.1.1 - Interface de connexion	7
3.1.2 - Interface de navigation pour les utilisateurs.....	7
3.1.3 - Interface de réalisation des commandes	7
3.1.4 - La base de données	7
3.2 -Application Web - Diagramme de composant	7
3.2.1 -Composants << UserManagement >>	9
3.2.2 -Composants << UserNavigation >>	9
3.2.3 -Composants << DeliverieOrganisation >>	9
3.2.4 -Composants << Database >>	10
4 -Architecture de Déploiement.....	11
4 - Diagramme de déploiement	11
4.1 -Serveur Web NGINX	12
4.1 -Serveur WSGI	12
4.1 -Serveur de Base de données MySQL	12



1 - VERSIONS

Auteur	Date	Description	Version
Sulyvan	08/08/2019	Création du document	1.0



2 - INTRODUCTION

2.1 - Objet du document

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

- Description du fonctionnement de la base de données,
- Description des composants qui seront utilisés,
- Description du déploiement de l'application et des serveurs qui seront utilisés.

Les éléments du présent dossier découlent :

- De la spécification fonctionnelle *OC Pizza*
- Des différents échanges avec le client.

2.2 - Références

Pour mieux saisir ce présent document, vous pouvez vous référer sur le document des

- [Spécifications fonctionnelles.](#)
- [Architecture fonctionnelle du projet.](#)



2.3 - Modèles fonctionnelle

Ce présent diagramme UML décrit le modèle fonctionnel de la base de données,
Les liens qui seront utilisé pour que les tables puissent communiquer.

2.3.1 - Règle de gestion

Ici nous avons plusieurs tables et plusieurs sections, nous pouvons donc obtenir plusieurs relations de différents types :

Pour les Tables :

Relations entre les tables :

- Employés et Acteur la relation est = 1 à 1

Nous avons donc **un** employé pour **un** Acteur

- Restaurant et Mail la relation est = 1 à 1

Nous avons donc **un** Restaurant pour **un** Mail

- Statut et Commande la relation est = 1 à 1*

Nous avons donc **un** statut pour **plusieurs** Commandes

- Facture et Acteur la relation est = 1 à 1*

Nous avons donc **une** Facture pour **plusieurs** Acteur

- Ingrédient et Produit la relation est = 1* à 1*

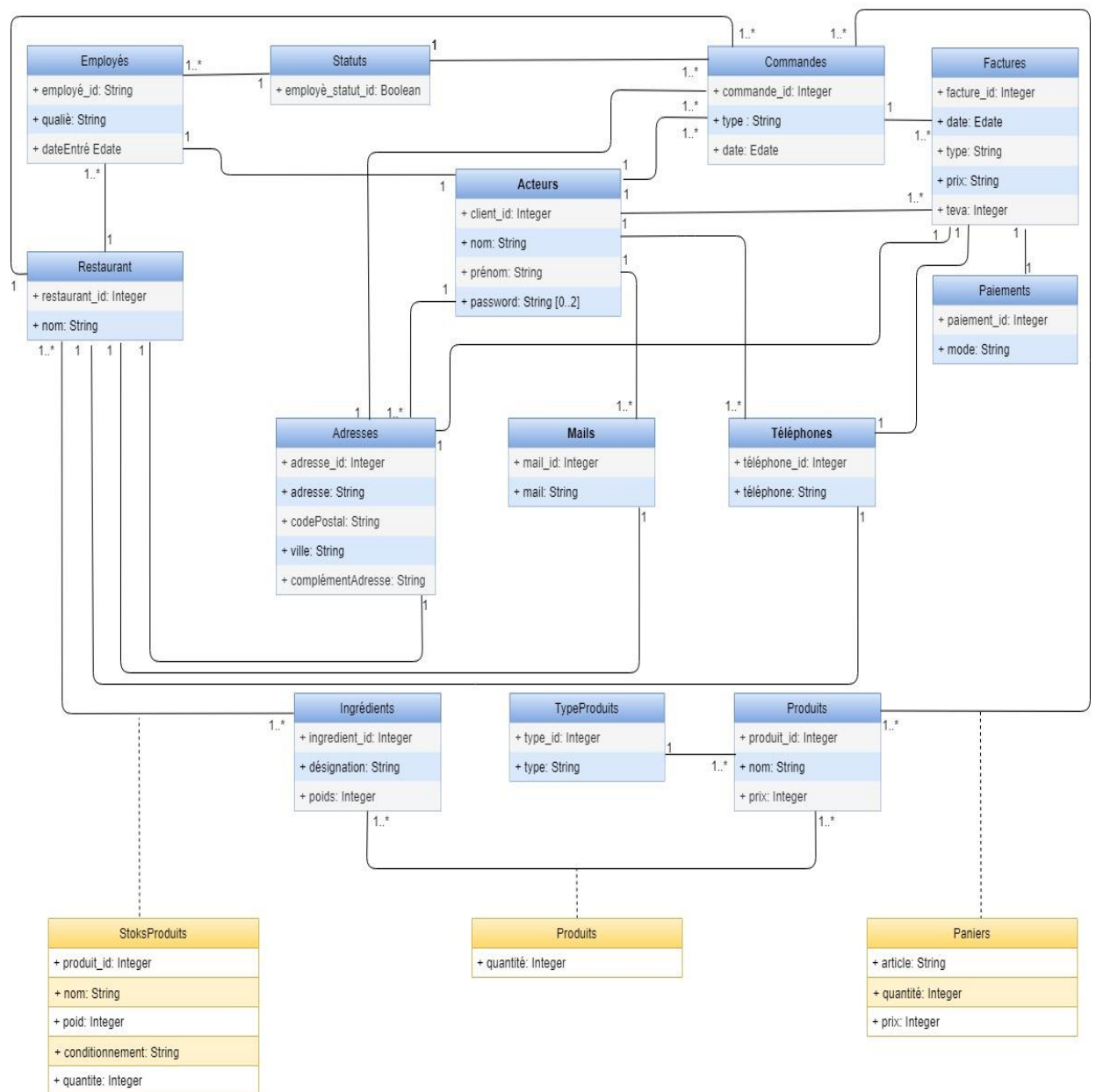
Nous avons donc **plusieurs** Ingrédient pour **plusieurs** Produits

- Commande et Produits la relation est = 1* à 1*

Nous avons donc **plusieurs** Commande pour **plusieurs** Produits



Class Diagramme





3 - ARCHITECTURE TECHNIQUE

3.1 - Composants généraux

3.1.1 - Interface de connexion

Pouvoir se connecter en tant qu'employé ou en tant que client.

3.1.2 - Interface de navigation pour les utilisateurs

Pouvoir visiter chaque page du site web OC Pizza.

3.1.3 - Interface de réalisation des commandes

Pouvoir réaliser une commande mais offre aussi la possibilité de personnaliser sa pizza,

Offre aussi la possibilité de consulter les recettes, consultez les réalisations encore possible

Car cette interface est en permanence en lien avec la base de données.

3.1.4 - La base de données

Pouvoir stocker toutes les données, utilisateurs, restaurants, administration, produits.

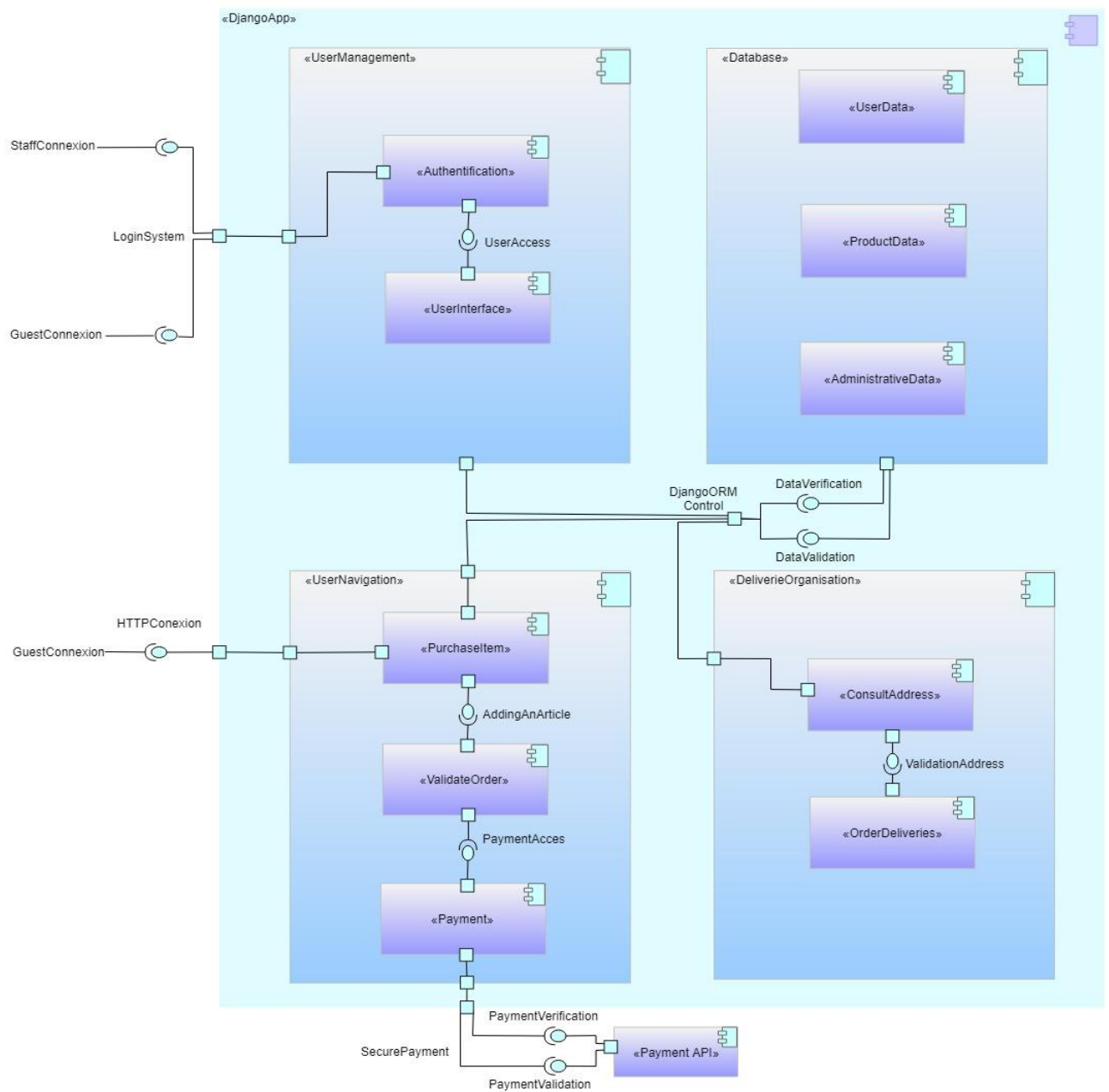
3.2 - Application Web

La pile logicielle est la suivante :

- Python : Application Django, BD Server MySQL
- HTML, CSS, JavaScript : Site web
- Python, JavaScript, Framework, API : Organisation des utilisateurs, commandes, paiements.



component diagram





3.2.1 - Composants << UserManagement >>

>> Authentification : Auras pour responsabilité de connecter chaque acteur et de rediriger celui-ci Vers leurs interfaces.

>> UserInterface : Chaque acteur possédera des « droits utilisateur » en fonction de leur statut,

- Client
- Gérant
- Hôtesse de caisse
- Livreur
- Pizzaiolo
 - Le rôle de chaque acteur sera prédéfini en amont, de façon à éviter que le pizzaiolo Puisse accéder à l'interface du Gérant par exemple.

3.2.2 - Composants << UserNavigation >>

>> PurchaseItem : Un composant qui va permettre d'effectuer une liste d'achat et de les ajouter dans le « Panier d'achat »

>> ValidateOrder : L'étape qui interroge l'utilisateur, afin de valider sa commande, une étape clé qui interrogera les stocks en temps réel, permet de confirmer une commande ou tout simplement de proposer une alternative en cas de rupture de stock d'un produit.

>> Payement : Composant de paiement en ligne, pas obligatoire après validation d'une commande, mais facilite la livraison, et permet à l'utilisateur de se libérer de cette tâche, si celle-ci c'est bien déroulé sur l'espace de paiement sécurisé.

3.2.3 - Composants << DeliveryOrganisation >>

>> ConsultAddress: Ce composant consultera l'adresse fourni soit par la commande initialement validé soit par l'adresse fourni par le client l'or de son inscription sur le site OC Pizza,



- L'adresse de livraison peut spontanément changer selon les préférences du client, il peut ainsi commander dans un autre restaurant OC Pizza sans devoir changer son adresse de livraison habituelle.

>> OrderDeliveries : Un composant qui offre aux livreurs une organisation sûre et rapide sans compromettre sa tournée, un GPS qui indiquera aux livreurs,

- Une organisation des livraisons en fonctions de ces distances, quantités, demande spécifique du client,
- des changements de livraisons si des déviations sont à déjoué.

3.2.4 - Composants << Database >>

>> UserData : Pour y stocké toutes les informations liées à l'acteur,

- Qui s'inscrit sur le site Oc Pizza,
- Le nouvel employé qui rejoint la grande équipe Oc Pizza.

>> ProductData : Pour y stocker les informations des produits,

- Les produits présents sur le menu et sur le site Oc Pizza,
- Les produits ajoutés lors d'une réception de commande,
- Leurs liens vont permettre de suivre son état du stock en temps réel et offrir ainsi une commande sur et personnalisable à souhait.
- Création de divers recette, pizza, salades, dessert et une interface qui prend en compte les modifications d'ingrédient ou tout simplement un moyen de consulter chaque recette.

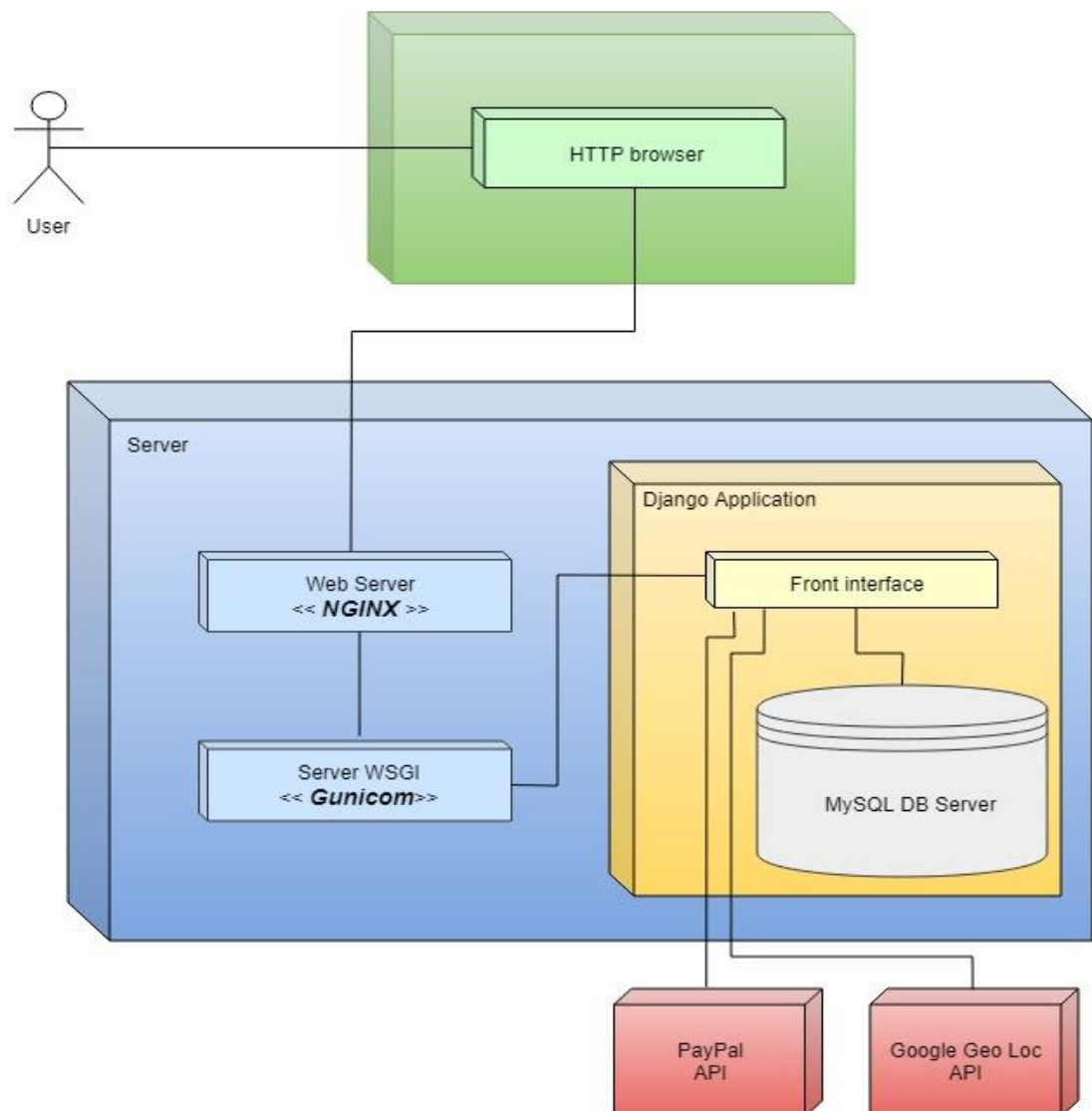
>> AdministrativeData : Un composant qui va stocker tous types de mouvement,

- Les commandes réalisées avec leurs factures et le moyen de paiement
- Cela peut permettre de réaliser tous types d'analyse afin de déterminer les changements à apporter afin d'améliorer ou modifier en fonctions des tendances, des préférences des clients.



4 - ARCHITECTURE DE DEPLOIEMENT

Deployment diagram





4.1 - Serveur Web NGINX

Serveur http Open source Asynchrone

Permet que chaque requête soit traitée avec un processus dédié, pour permettre un traitement rapide.

Conçu spécialement pour un grand trafic ce qui optimise le site et ouvre des possibilités.

NGINX est à ce jour le serveur le plus utilisé au monde.

4.2 - Serveur WSGI

Soit « **Web Server Gateway Interface** », Un server qui demande de respecter un certains nombres de conventions afin d'assurer une compatibilité avec les applications web et une bonne communication entre celui-ci et le serveur web afin de générer les pages demander par l'utilisateur.

6.4 - Serveur de Base de données MySQL

Un serveur dédié uniquement pour la base de données et sera intégré à l'application Django afin d'optimiser la communication sur chaque interface de communication.