La solution technique retenue pour ce projet.

Une application en mesure de répondre aux besoin des clients, c'est-à-dire pouvoir passer une commande par internet de façon rapide, efficace et simple mais aussi de modifier ou de la supprimer si nécessaire (sous certaines conditions, voir dans la spécification fonctionnelles).

Les employés de l'entreprise OC pizza pourront également effectuer des commandes par téléphone ou directement sur place. pour ce faire ils pourront créer un compte directement lié au compte client pour passer une commande pour lui

Le pizzaïolo pourra consulter avec l'application, les différentes recette de pizza. Les livreurs pourront changer l'état de la commande en "livrée" en utilisant l'application une fois celle-ci livrée au client.

La solution technique retenue pour ce projet : Le serveur

Un serveur d'application Gunicorn + Nginx en reverse proxy

Gunicorn est un serveur d'application web très adapté pour les projets python avec le framework django de plus il permet d'être couplé avec Nginx qui est un autre serveur web mais ici il sera configuré configuré de manière à servir d'intermédiaire pour améliorer les performances des requêtes car il aura pour but de servir les fichier statique du site avant de passer par gunicorn pour le reste ce qui permet de répartir la source de travail pour les serveur

cette configuration à beaucoup d'avantage :

Intermédiaire de sécurité : le mandataire inverse protège un serveur Web des attaques provenant de l'extérieur. En effet, la couche supplémentaire apportée par les mandataires inverses peut apporter une sécurité supplémentaire. La réécriture programmable des URL permet de masquer et de contrôler, par exemple, l'architecture d'un site web interne. Mais cette architecture permet surtout le filtrage en un point unique d'accès aux ressources Web.

Répartition de charge : le mandataire inverse peut distribuer la charge d'un site unique sur plusieurs serveurs Web applicatifs. Selon sa configuration, un travail de réécriture d'URL sera donc nécessaire

cache : le mandataire inverse ou (reverse proxy) peut décharger les serveurs Web de la charge de pages/objets statiques (pages HTML, images) par la gestion d'un cache local. La charge des serveurs Web est ainsi généralement diminuée, on parle alors d'« accélérateur web » ou d'« accélérateur HTTP ».

Pour résumer:

Visiteur > Web > Votre Serveur > Nginx > Gunicorn > Django

source : http://sametmax.com/nginx-en-reverse-proxy-gunicorn-pour-vos-apps-django/

La solution technique retenue pour ce projet : Le langage de programmation

le langage de programmation utilisé sera Python

Python est le langage de programmation le plus utilisé au monde. C'est due au fait qu'il est utilisable dans tous les domaines de la programmation. Du web à la finance, python peut faire absolument tout.

Python est puissant et peut être utilisé pour n'importe quoi, grâce aux nombreuses fonctionnalités qu'il offre par défaut et avec des bibliothèques standard couvrant presque toutes les tâches de programmation. Que vous souhaitez effectuer des calculs scientifiques, traiter des images ou développer des interfaces ou des protocoles de systèmes d'exploitation, Python vous fera gagner un temps précieux.

De plus, Python convient parfaitement aux produits que l'on souhaite lancer le plus rapidement possible. Cela signifie un retour sur investissement (ROI) plus rapide et une possibilité d'adapter le produit en fonction des retours d'utilisateurs réels.

Python as quelques gros avantages lui permettant d'être un des langages de programmation les plus intéressant pour le web :

- Une très bonne gestion du JSON / XML
- Une gestion des sockets UDP / TCP très intuitive
- De très bonnes librairies de gestion d'email
- Des moyens très simples d'utiliser le FTP

Source : https://leblogducodeur.fr/pourquoi-utiliser-python/

La solution technique retenue pour ce projet : Le Framework

le Framework utilisé sera Django

Django est avant tout un framework qui permet de développer un site web orienté contenu très rapidement. Il simplifie la gestion des cas classiques comme la création de pages statiques, la pagination, les listes d'objets, les systèmes d'authentification, etc.

Django est rapide, et pas seulement en terme de développement. De nombreux sites à fort trafic choisissent Django pour ses performances Web : temps de chargement des pages, affichage des contenus, etc.

Enfin, en terme de sécurité, Django vous aide à vous protéger contre les attaques les plus fréquentes : injection SQL, cross-site scripting, clickjacking, etc.

Source : https://www.appvizer.fr/services-informatiques/framework/django

La solution technique retenue pour ce projet : La base de données

la base de données utilisé sera PostgreSQL

il y a de nombreux avantage par rapport à mysql:

Support de la syntaxe Oracle : si l'ancienne base de données était sous Oracle, la migration sera plus facile sur PostgreSQL ;

PostgreSQL est plus fiable et l'intégrité des données y est plus performante ;

Tout ce qui est lié aux requêtes est meilleur avec PostgreSQL. PostgreSQL dispose d'un planificateur de requêtes sophistiqué et d'un optimiseur de requêtes ;

Dans un contexte de data-mining / data warehouse, PostgreSQL est plus performant ;

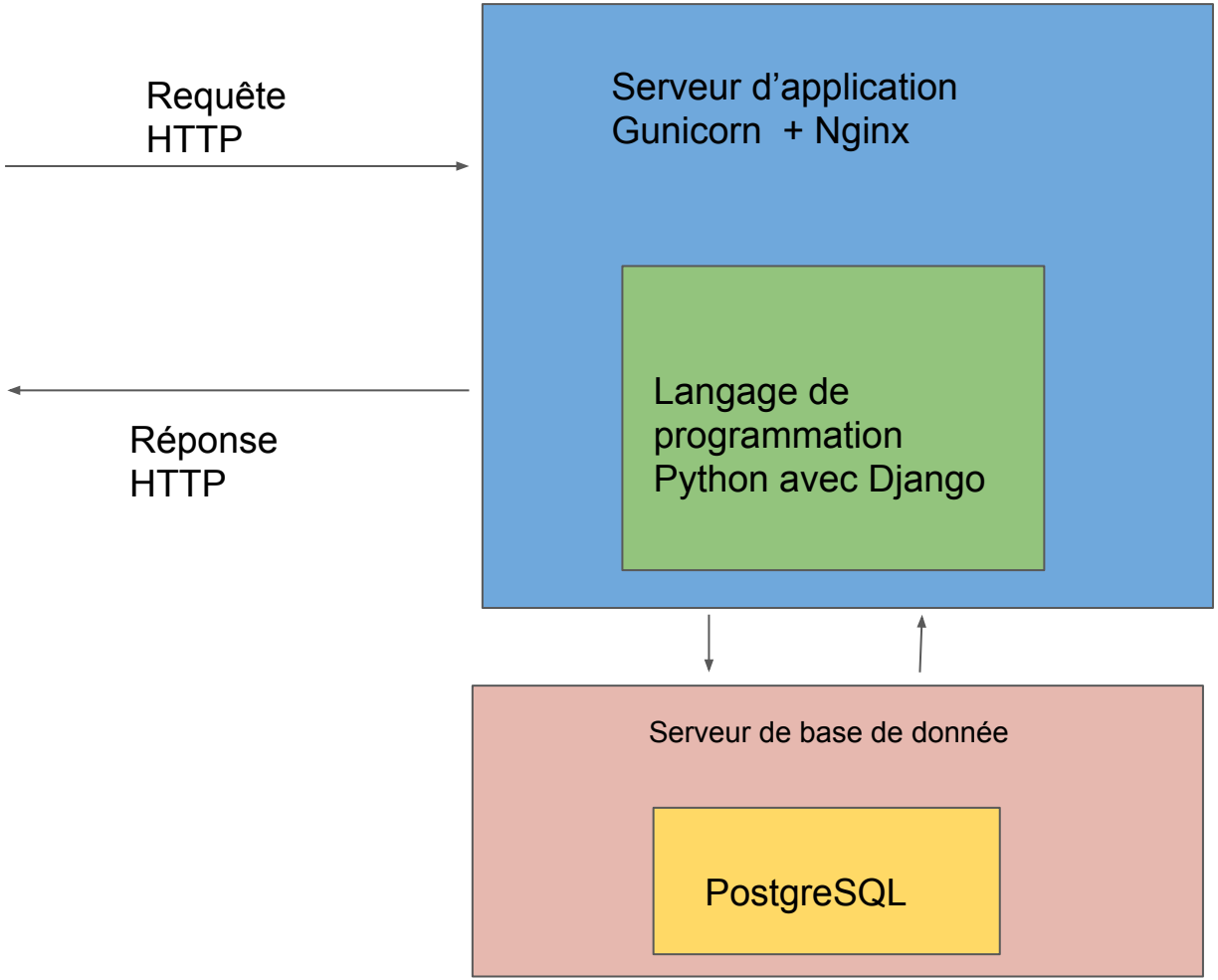
La documentation de PostgreSQL est plus dense et complète et de façon général le support est meilleur ;

PostgreSQL traîne moins de casseroles (astuces, limitations…) que MySQL ;

PostgreSQL est meilleur pour les requêtes avec sous-requêtes

il permet de traiter de plus gros volume de données

source: https://www.base-de-donnees.com/preferer-postgresql-a-mysql/


Requête
HTTP
Serveur d'application
Gunicorn + Nginx
Langage de
programmation
Python avec Django
Réponse
HTTP
Serveur de base de donnée
PostgreSQL