MATHIEU FERRANDEAU

PARCOURS OPENCLASSROOMS DÉVELOPPEUR D'APPLICATION PYTHON

Projet 6 : Concevez la solution technique d'un système de gestion de pizzeria

https://github.com/MathieuFerrandeau/project6

Ce document résume la réalisation du 6ème projet du parcours de développeur d'application – Python dispensé par OpenClassrooms.

Version	Modifié le	Par	Modifications
1	26/07/2019	Mathieu	Création du document
2	27/07/2019	Mathieu	Ajout Objet du document / Besoin du client
3	30/07/2019	Mathieu	Ajout Architecture technique / de l’application
4	31/07/2019	Mathieu	Ajout Architecture BDD / diagramme de composants / diagramme de déploiement
5	01/07/2019	Mathieu	Ajout Acteurs externes
6	02/07/2019	Mathieu	Mise en forme du document

SOMMAIRE

I. Objet du document

Objet

Ce document représente le dossier de conception technique de l'application OC Pizza.
Il à pour but d'analyser la demande du client OC Pizza afin de :

- modéliser les objets du domaine fonctionnel ;
- identifier les différents éléments composant le système à mettre en place et leurs interactions ;
- décrire le déploiement des différents composants envisagés ;
- élaborer le schéma de la base de données.

II. Besoin du client

Contexte

« OC PIZZA » est un jeune groupe de pizzeria en plein essor et spécialisé dans les pizzas livrées ou à emporter. Il compte déjà 5 points de vente et prévoit d'en ouvrir au mois 3 de plus d'ici la fin de l'année.

Enjeux et objectifs

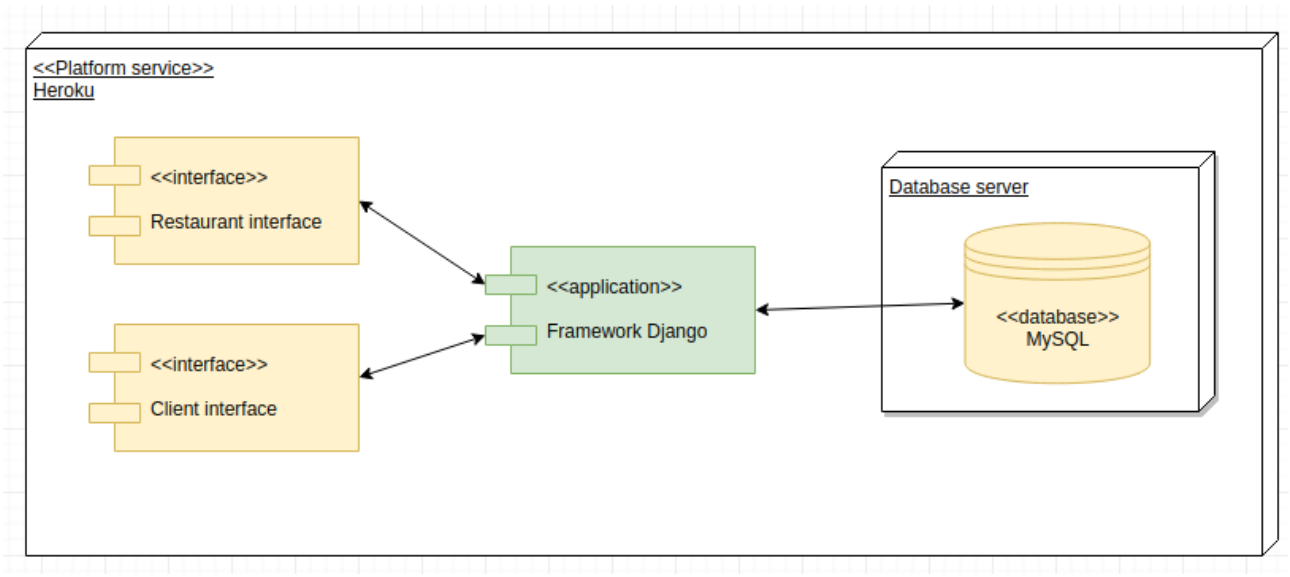
Un des responsables du groupe a pris contact avec nous afin de mettre en place un système informatique, déployé dans toutes ses pizzerias et qui lui permettrait notamment :

- d'être plus efficace dans la gestion des commandes, de leur réception à leur livraison en passant par leur préparation ;
- de suivre en temps réel les commandes passées et en préparation ;
- de suivre en temps réel le stock d'ingrédients restants pour savoir quelles pizzas sont encore réalisables ;
- de proposer un site internet pour que les clients puissent :
  - passer leurs commandes, en plus de la prise de commande par téléphone ou sur place,
  - payer en ligne leur commande s'ils le souhaitent – sinon, ils paieront directement à la livraison ;
- modifier ou annuler leur commande tant que celle ci n'a pas été préparée ;
- de proposer un aide mémoire aux pizzaiolos indiquant la recette de chaque pizza ;
- d'informer ou notifier les clients sur l'état de leur commande.

III. Architecture technique

Le logiciel :

- sera composé de 2 interfaces différentes, une interface front pour les clients et une interface back pour les employés ;
- se basera sur un socle technique et une base de données commune ;
- sera hébergé à l'aide d'une solution cloud Heroku.



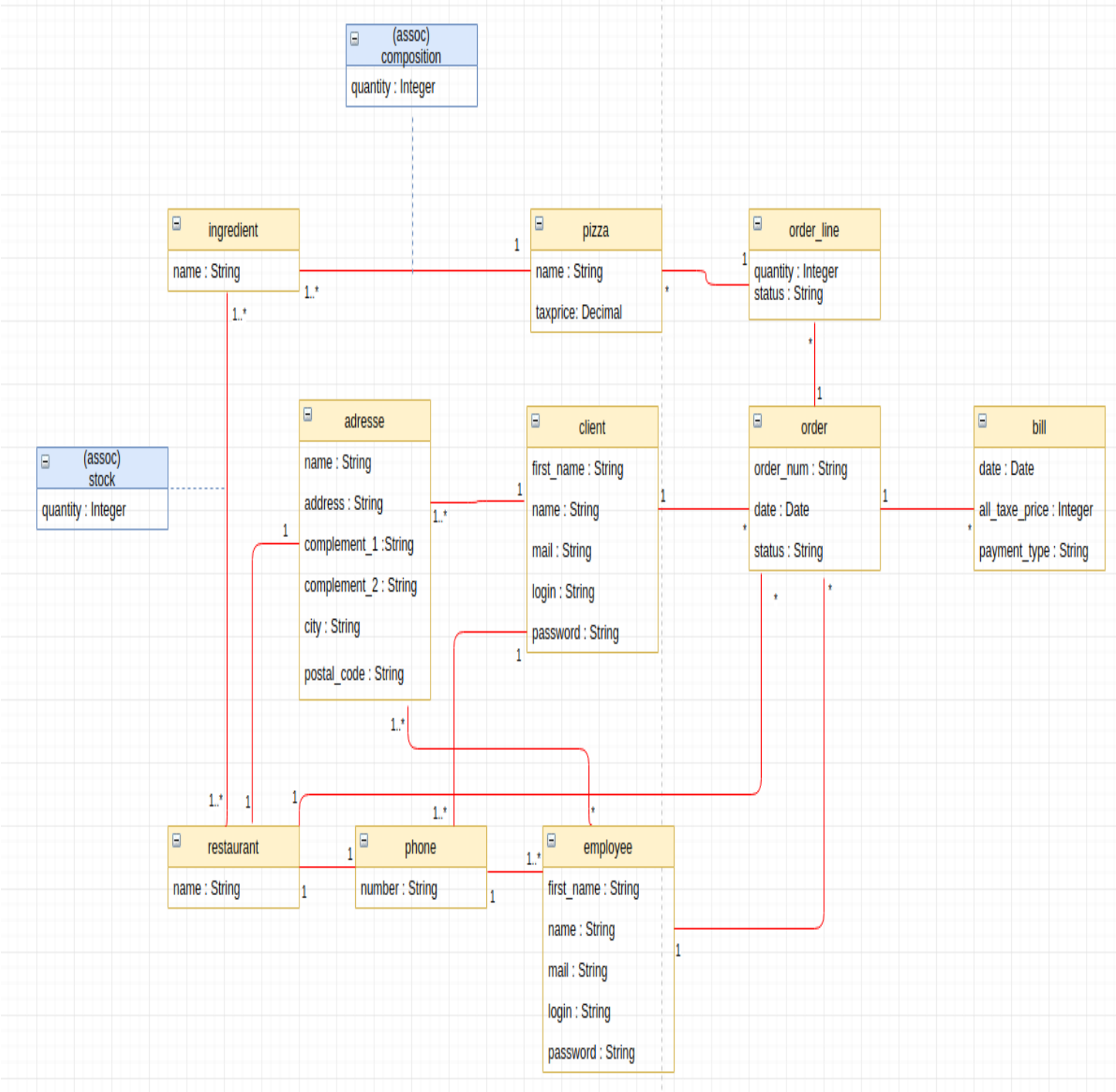
Avantages et inconvénients :

L'avantage de séparer les interfaces du cœur de l'application sera de faire une application qui sera plus modulable et plus extensible.
Cela en fera une solution bien plus simple à maintenir dans le temps avec les évolutions techniques futures.

Déployer l'application sur une plateforme cloud ôtera la contrainte budgétaire et temporaire d'installation et de gestion des serveurs.

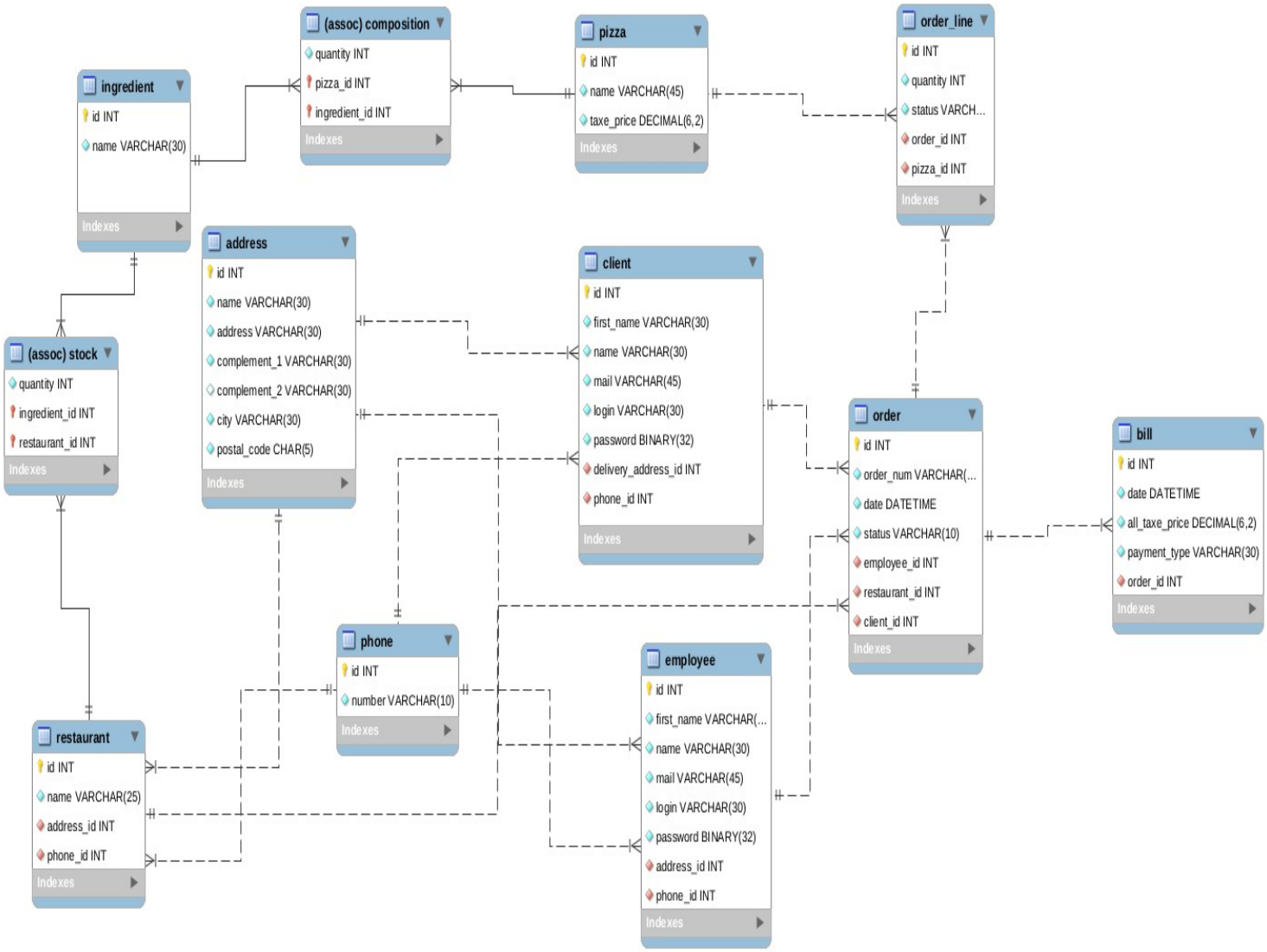
Le seul inconvénient est le fait de passer par un prestataire tiers et donc d'être dépendant de ses futures évolutions stratégiques.

IV. Architecture de l’application



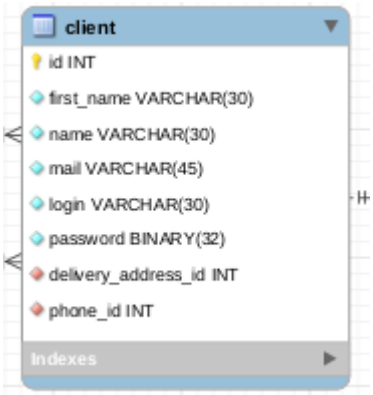
Afin de réaliser la structure de la base de données, nous avons tout d’abord réalisé le diagramme de classes ci-dessus.
Ce diagramme nous a permis de définir comment l’application sera structurée et ainsi réaliser les tables de la base de données qui seront nécessaires.

V. Architecture de la base de données



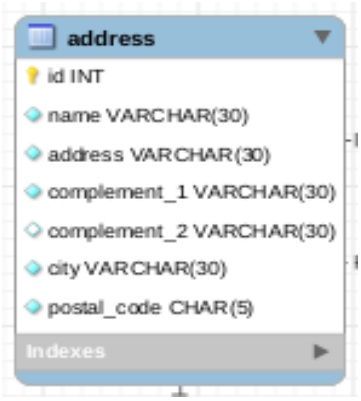
La base de données est composée des tables présentent sur l'image ci-dessus.

Sur les pages suivantes, vous retrouverez un aperçu de chaque table et une description des points importants ainsi qu'une explication des clés étrangères.



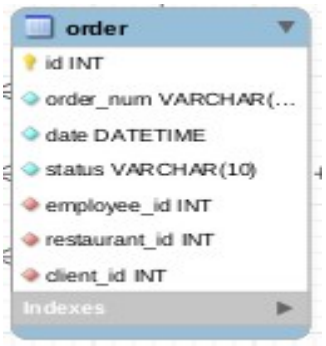
La table **client** permet de stocker certaines informations des clients tel que leurs logins, leurs noms et prénoms.

password : Stock un hash du mot de passe dans la base de données.
FK : delivery_address_id permet de rattacher un client à son adresse de livraison.
FK : phone_id permet de rattacher un client à son numéro de téléphone.



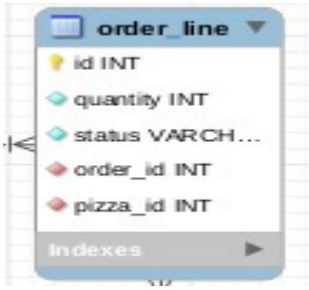
La table **address** permet de stocker toutes les adresses (clients, restaurants).

La table **phone** stocke tous les numéros de téléphone (clients, restaurants).



La table **order** stocke le numéro de commande, sa date et son statut.

FK : employee_id permet de rattacher une commande a un employé.
FK : restaurant_id permet de rattacher une commande à un restaurant.
FK : client_id permet de rattacher une commande à un client.



La table **order_line** permet de stocker la la quantité de pizza présente dans une commande et son statut de préparation.

FK : order_id permet de rattacher la ligne de commande à sa commande ;
FK : pizza_id permet de rattacher la ligne de commande à sa pizza.

La table **pizza** permet de stocker le nom et le prix TTC d'une pizza.

La table **composition** permet de savoir la composition d'une pizza.

PFK : pizza_id permet de rattacher la composition à une pizza.
PFK : ingredient_id permet de rattacher la composition a un ingrédient.

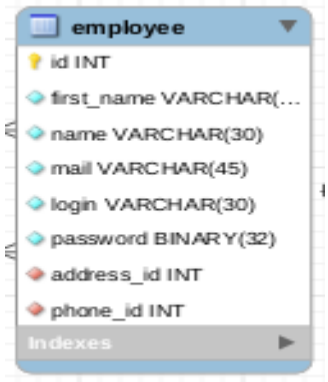
La table **ingredient** stocke le nom des ingrédients.

La table **stock** permet de connaître la quantité d'ingrédients en stock par restaurant.

PFK : ingredient_id permet de rattacher le stock à un ingrédient.
PFK : restaurant id permet de rattacher le stock à un restaurant.

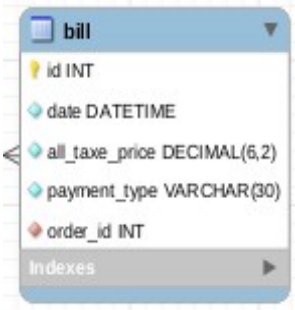
La table **restaurant** permet de stocker le nom du restaurant.

FK : adress_id permet de rattacher le restaurant à son adresse.
FK : phone_id permet de rattacher le restaurant à son numéro.



La table **employee** permet de stock le noms et les identifiants des employés.

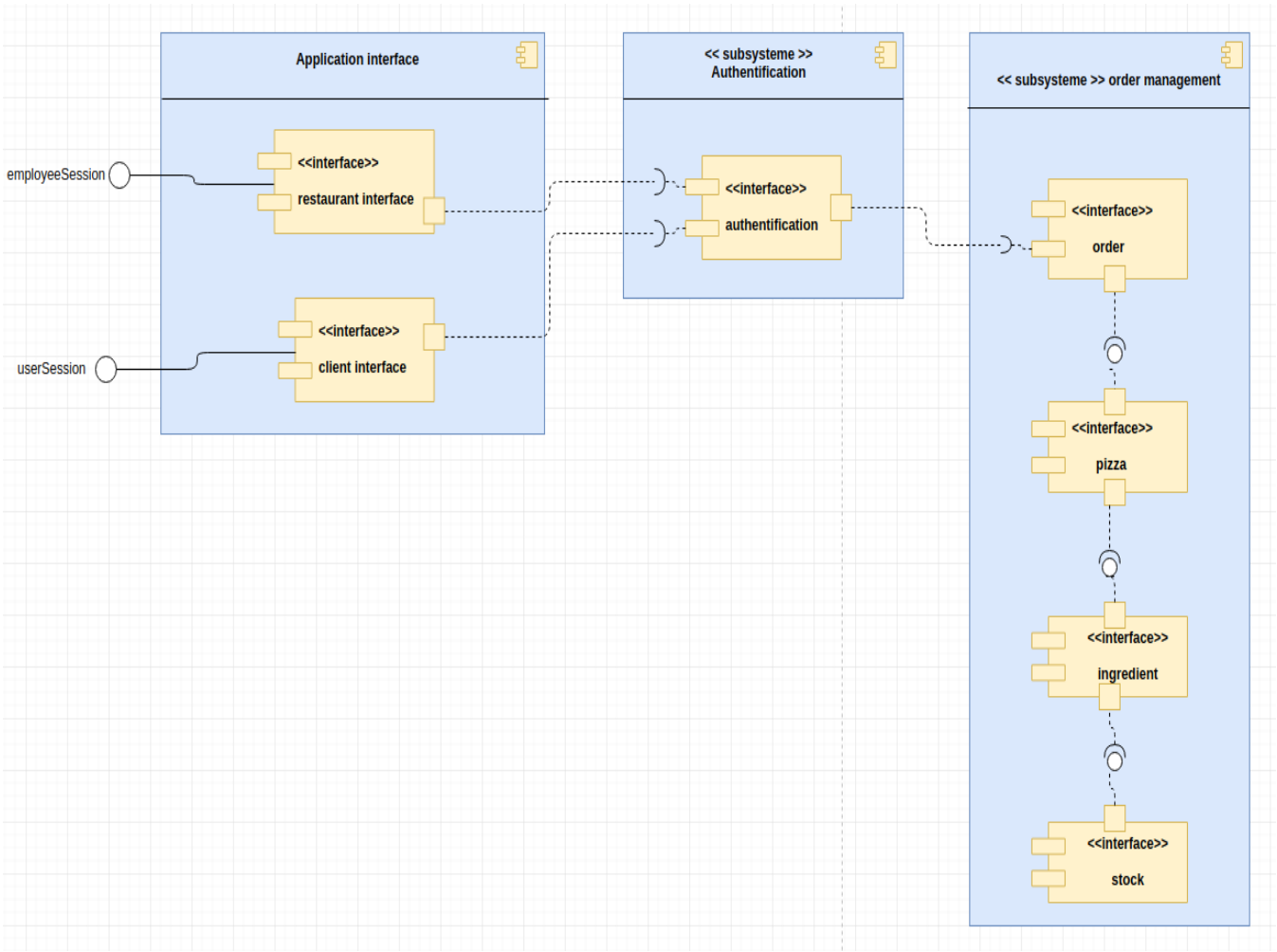
FK : adress_id permet de rattacher un employé à son adresse.
FK : phone_id permet de rattacher un employé à son numéro.



La table **bill** stock la date le prix TTC d'une commande et le type de paiement qui à été utilisé.

FK : order_id permet de rattacher une commande à sa facture.

VI. Architecture de composants



Ce diagramme de composants décrit l'organisation du système du point de vue des éléments logiciels. Ce diagramme nous permet de mettre en évidence les dépendances entre les composants.

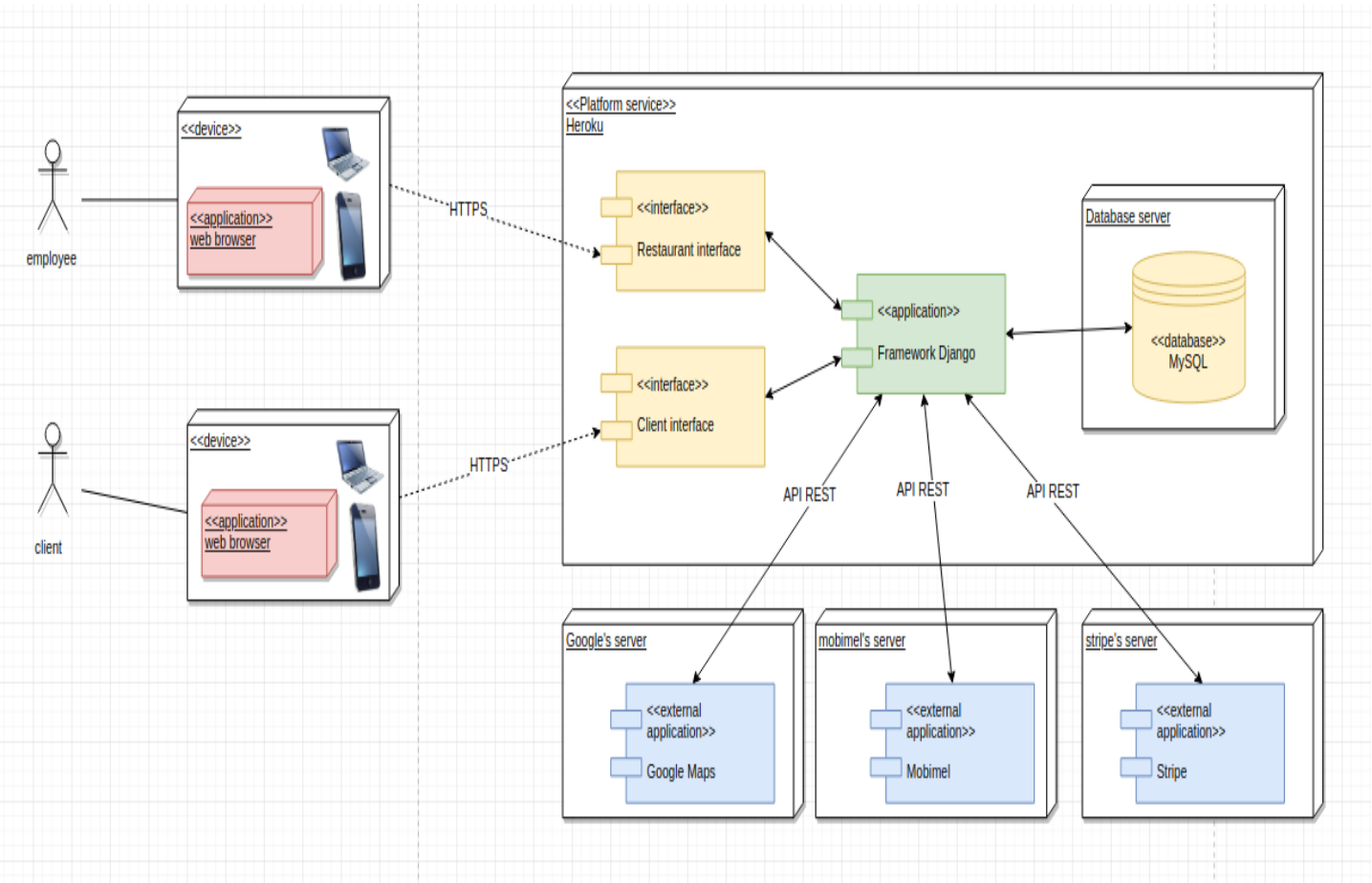
Nous pouvons voir que pour utiliser le module **authentification** nous avons besoin d'un client qui passe par l'interface utilisateur ou d'un employé qui passe par l'interface restaurant.

Que pour pouvoir être exécuté, le module **order** dépend du module authentification.

L'utilisation du module **order** permettra l'utilisation du module **pizza** qui fera appel au module **ingredient** qui appellera à son tour le module **stock**.

Ce diagramme nous permet de voir les dépendances entre les composants lorsqu'un employé ou un client souhaite valider une commande.

VI. Architecture de déploiement



Le diagramme de déploiement nous permet de comprendre comment sera déployée l'application et par quel moyen chaque acteur devrait y accéder.

A.Interfaces

Les utilisateurs (clients/employés) accèdent à l'application via leur terminal favori et un navigateur web installé sur celui-ci. La communication se fait via le protocole HTTPS.
Les clients et les employés ont accès à une interface différente.

B. Application

Le logiciel Python regroupe l'ensemble des fonctionnalités qui effectuent la liaison entre les interfaces et les serveurs externes des API et la base de données de l'application.
L'intérêt de cette démarche est de réduire le temps et le coût de développement, d'améliorer la sécurité global du logiciel et permettre le développement d'interfaces supplémentaires si nécessaires.

C. Serveur de l’application

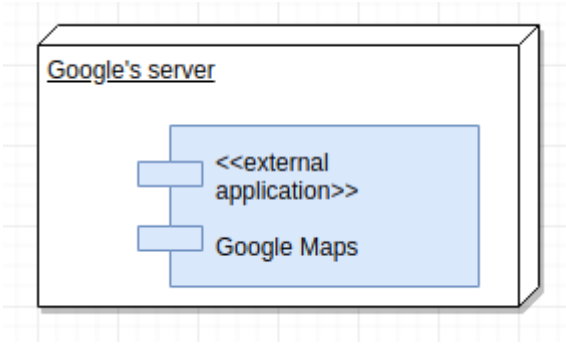
L’application sera déployée sur Heroku, une plateforme cloud d’application.
Cela permettra à OC Pizza de payer une solution en fonction de son utilisation et ainsi payer un tarif en fonction de l’affluence sur la plateforme.

D. La base de données

La base de données MYSQL sera déployée sur le serveur Heroku via ClearDB MYSQL.
Le choix d’une base de données MYSQL est préférable car bien intégrée avec Heroku.
MYSQL est une base de données très puissante qui est utilisée pour beaucoup de projet de e-commerce.

VIII. Acteurs externes

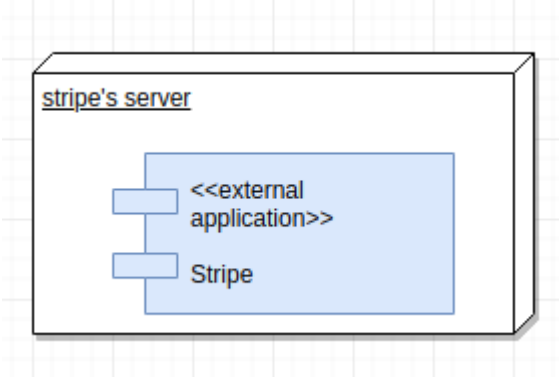
1. Maps : Google Maps



- Permet au livreur de se rendre efficacement chez le client en affichant l’itinéraire le plus rapide en tenant compte des travaux en cours et du trafique ;
- Affiche la position des restaurants pour l’utilisateur.

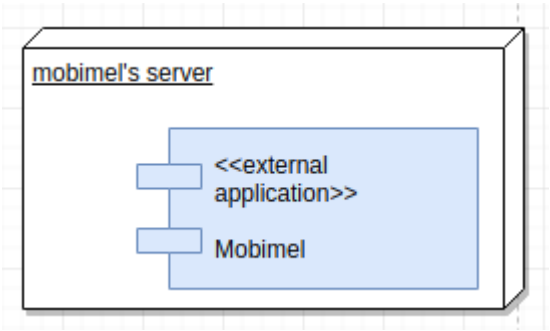
Il existe une librairie créée par la communauté Python pour utiliser facilement Google Maps au sein d’une application Python/Django.

2. Paiements : stripe



Cette Api est un système de gestion des paiements via applications web.
Elle est sécurisée, simple à mettre en place et est répandu sur les sites de e-commerces comme Deliveroo, Heetch ou Drivy.

3. Notifications : Mobimel



Permet de notifier l'utilisateur par SMS via des requêtes HTTPS de l'API Mobimel.
La base de données communiquerait au cœur de l'application le numéro de téléphone, la commande associé et son statut actuel.
Permettant à OC Pizza de notifier son client par sms.