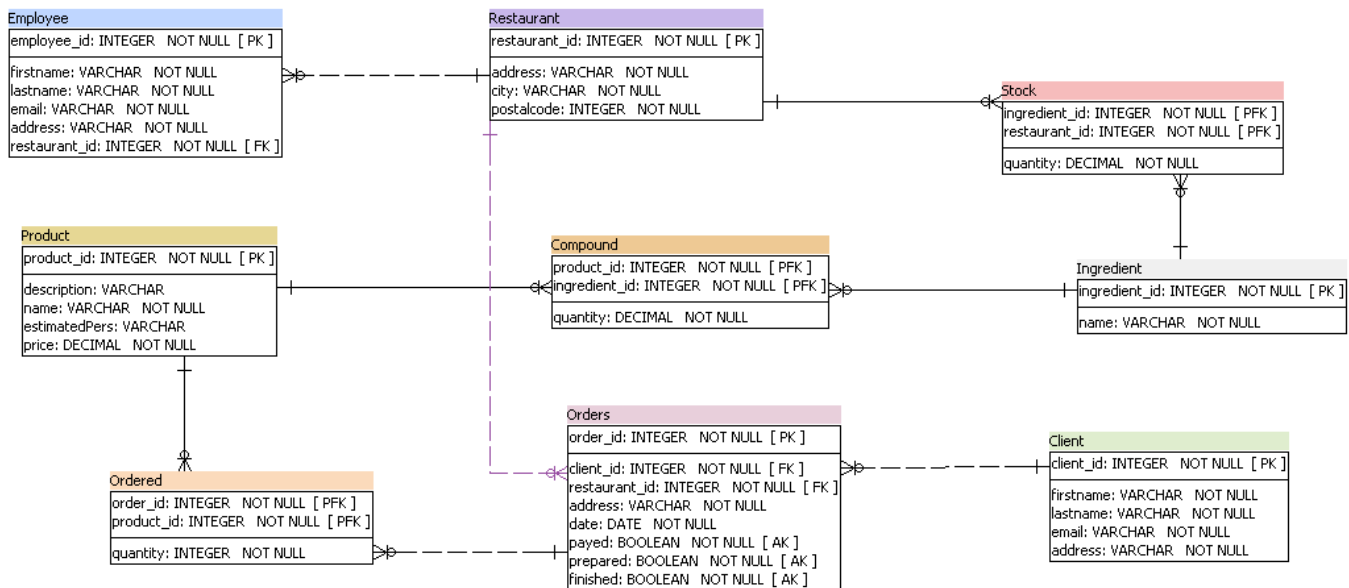


I) Domaine fonctionnel



Employee : Permet de stocker les employés ainsi que de savoir dans quel restaurant ils sont employés.

⇒ Restaurant : Utilise la clé étrangère « Restaurant_id » pour identifier le restaurant dans lequel un employé travail

Restaurant : Désigne un restaurant et sa position

Client : Permet de stocker un client ainsi que ses informations personnelles.

Ingredient : Permet de stocker un type d'ingrédient par son nom

Product : Permet de stocker un produit, sa description, le nombre de personne estimée à sa consommation et son prix.

Compound : Liaison entre un produit et les ingrédients, définissant combien d'ingrédients est nécessaire à la fabrication du produit en question.

⇒ Ingredient/Product: Partage de la fonction de clé primaire (product_id, ingredient_id) afin de lier les composants

Stock : Permet de stocker la quantité d'ingrédient disponible dans chaque restaurant.

⇒ Ingredient/Restaurant: Partage de la fonction de clé primaire étrangère (ingredient_id, restaurant_id), pour désigner chaque type d'ingrédient pour chaque restaurant.

Orders : Permet de stocker les commandes passées par les clients, ainsi que de suivre leurs statuts, si elles ont été préparées, payées et livrées.

⇒ Client : Utilise la clé étrangère client_id pour désigner le propriétaire d'une commande.

⇒ Restaurant : Utilise la clé étrangère restaurant_id pour désigner dans quel restaurant la commande a été passée.

Ordered : Fait la liaison entre une commande et ses produits, permet donc de savoir combien de produits sont présents dans une dite commande.

⇒ Product/Orders: Partage de la fonction de clé primaire étrangère (product_id, order_id), pour désigner les différents produits d'une commande.

⇒

II) Composants du système

Site :

- Utilisation de Django, le site sera divisé en deux applications, une pour la vente destinée au client et une pour le visionnage des commandes, leurs états et les stocks réservés au personnel.

API (Rest):

- Permet de récupérer information sur les stocks présents dans la base de données.
- L'API sera utilisée pour le Dashboard utilisé par les employées en cuisine leur permettant de voir les stocks en temps réel ainsi que l'état des commandes sur la timeline.
- L'intégration de l'API nécessitera d'installer le Framework Django Rest.

III) Composants externes du système

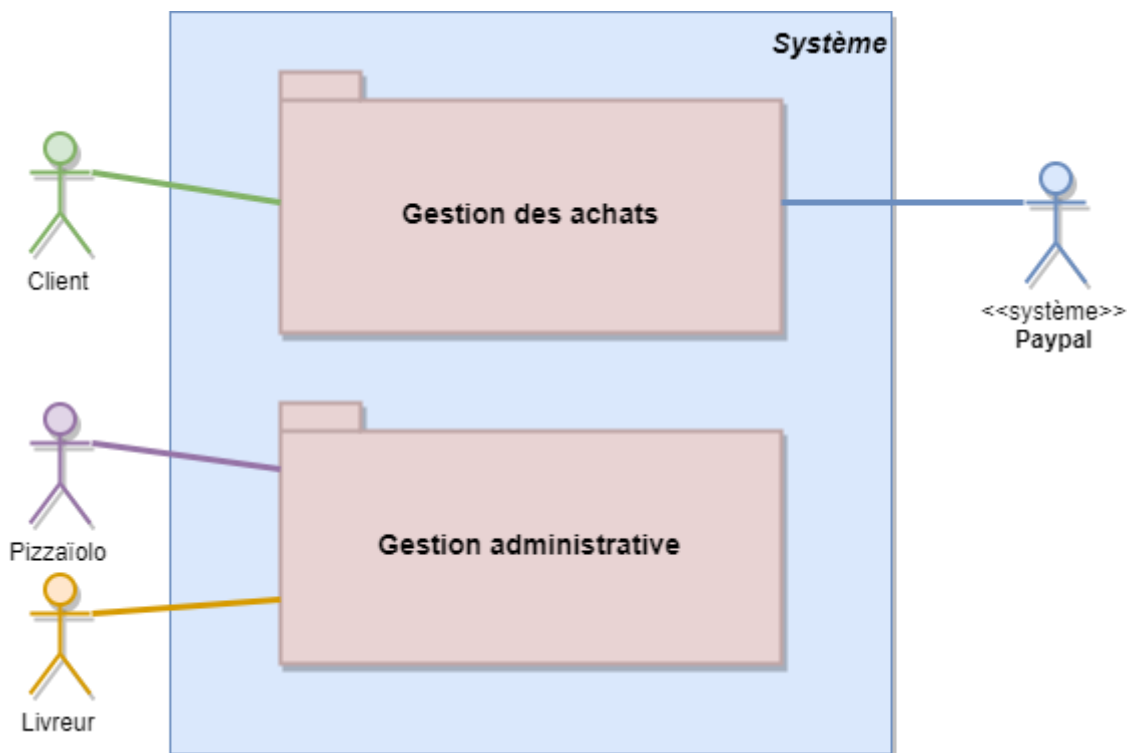
PayPal :

- Permet la gestion des paiements sur le site internet.
- Présent lors du passage de commande si l'utilisateur choisit de payer depuis le site web.
- L'intégration se fait en copiant un contenu HTML généré par l'outil de PayPal dans la page HTML du site.

Google Map :

- Permet d'afficher visuellement la position des points de vente pour l'utilisateur.
- Présent sur la page « Locate » afin de trouver le restaurant le plus proche.
- Tout comme pour Paypal, l'intégration se fait en copiant un « embed » dans la page HTML du site.

IV) Organisation physique



Le système utilisera Django sur un serveur sous une distribution UNIX et sera divisé en deux applications distinct le site client et le site employé contenant un Dashboard pour la préparation des commandes et gestion des stocks.

