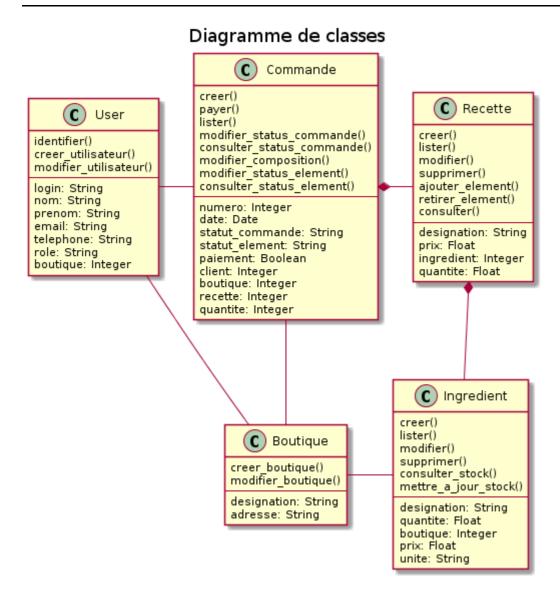
original sur https://github.com/Zepmanbc/oc_dapython_pr6/blob/master/doc/demarche.md

Contexte

L'analyse fonctionnelle de la solution du site de gestion de pizzeria étant faite, il est maintenant demandé de décrire le domaine fonctionnel, les composants, le déploiement ainsi que la structure de la base de donnée.

Description du domaine fonctionnel



Description des composants du système

Diagramme de composants

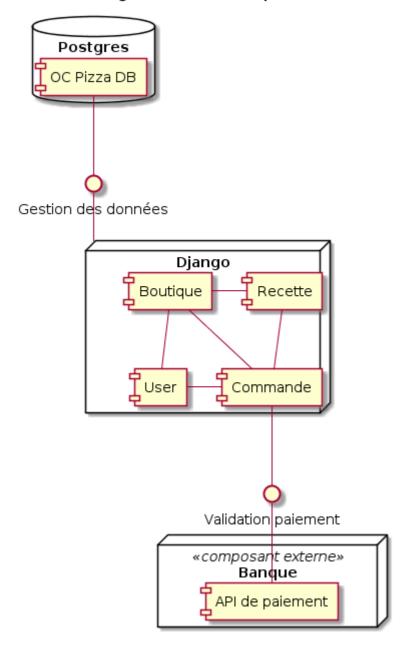
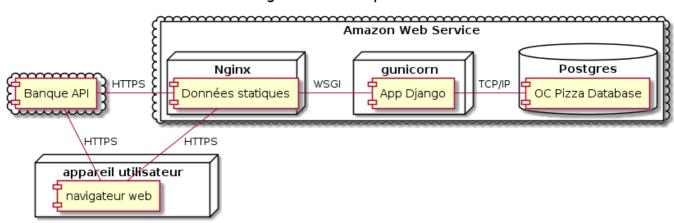


Diagramme de déploiement



Modèle Physique de données

Modèle physique de données commande_composition status_commande [PK] id: INTEGER NOT NULL status_composition paiement_type [PK] id: INTEGER NOT NULL FK] commande_id: INTEGER NOT NULL [FK] recette_id: INTEGER NOT NULL [FK] status_id: INTEGER NOT NULL (quantité = nombre de lignes) [PK] id: INTEGER NOT NULL [PK] id: INTEGER NOT NULL designation: VARCHAR designation: VARCHAR User commande [PK] id: INTEGER NOT NULL [PK] id: INTEGER NOT NULL recette [FK] boutique_id: INTEGER [FK] role_id: INTEGER NOT NULL [PK] id: INTEGER NOT NULL FK] dient_id: INTEGER NOT NULL [FK] boutique_id: INTEGER NOT NULL [FK] status_id: INTEGER NOT NULL [FK] paiement_type_id: INTEGER NOT NULL 0..× login: VARCHAR() UNIQUE NOT NULL nom: VARCHAR telephone: VARCHAR(12) password: CHAR(64) NOT NULL (SHA-256) nom: VARCHAR(256) designation_commerciale: VARCHAR designation_technique: VARCHAR prix: FLOAT(4,2) prenom: VARCHAR(256) email: VARCHAR(256) paiement: BOOLEAN 0..3 0...* 0..1 0..3 boutique recette_composition role [PK] id: Integer NOT NULL [PFK] recette_id: INTEGER NOT NULL [PFK] ingredient_id: INTEGER NOT NULL [PK] id: Integer NOT NULL designation: VARCHAR(256) nom: String adresse: String horaires: String quantite: INTEGER(4) stock ingredient [PFK] ingredient_id: INTEGER NOT NULL [PFK] boutique_id: INTEGER NOT NULL [PK] id: INTEGER NOT NULL nom: VARCHAR unite: VARCHAR quantite: INTEGER(12)

Base de données

Création de la base:

01 database structure.sql

02 database basics.sql

03_database_view_procedure.sql

remplissage de la base

10 stock.sql

30 clients.sql (https://mockaroo.com/schemas/154661)

commandes.sql (commandes.py)

Dump de la base : oc_dapython_pr6.sql

mysqldump --routines --events oc_dapython_pr6 > oc_dapython_pr6.sql"

Mise en situation

 Voir les pizzas disponibles en fonction des ingrédients (procédure recette_vue ou vue v_0_recettes_possible)

- Voir les commandes en cours dans une boutique (vues *v_1_commandes_boutique_**)
- Voir l'inventaire dans une boutique (vue *v_1_stock_boutique_nord**)
- Afficher une recette (procedure *voir_recette*)
- Calculer le chiffre d'affaire des boutiques (affiche les commandes payées du jour par boutique, vue v_0_chiffre_d_affaire_journee)
- Mise à jour du statut de la commande en fonction de l'avancé des préparations (trigger after_update_commande_composition)
- Mise à jour du stock quand une pizza passe en préparation (trigger after update commande composition)
- Mise à jour du statut de la commande en fonction du paiement (et du statut) (trigger before_update_commande)
- Création du stock en fonction des modifications de boutiques ou d'ingrédients (triggers after_delete_boutique after_insert_boutique after_delete_ingredient after_insert_ingredient)