

OC PIZZA

SOLUTION TECHNIQUE

PROJET VI : CONCEVEZ LA SOLUTION TECHNIQUE D'UN SYSTÈME DE GESTION DE PIZZERIA

I. INTRODUCTION

1.1 CONTEXTE		3
1.2 OBJECTIFS		3
1.3 SOLUTION		3

II. ARCHITECTURE TECHNIQUE

1.1 COMPOSANTS		4
<i>Diagramme de composants</i>		4

III. ARCHITECTURE DE DEPLOIEMENT

1.1 DEPLOIEMENT		5
<i>Diagramme de déploiement</i>		5

IV. STRUCTURE BASE DE DONNEES

<i>Diagramme de classes</i>		6
<i>Modèle physique de données</i>		7

Lien du repository --> [\[GitHub\]](#)

1.1 CONTEXTE

OC Pizza est un jeune groupe de pizzeria en plein essor et spécialisé dans les pizzas livrées ou à emporter. Il compte déjà cinq points de vente et prévoit d'en ouvrir au moins trois de plus d'ici la fin de l'année.

Le client a déjà fait une petite prospection et les logiciels existants qu'il a pu trouver ne lui conviennent pas. C'est pourquoi le groupe a fait appel à notre équipe afin de mettre en place un système informatique pour l'ensemble de ses pizzerias.

1.2 OBJECTIFS

La solution à développer doit répondre à plusieurs critères évoqués lors du premier rendez-vous.

- 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

1.3 SOLUTION

Mise en place d'une application web à partir de laquelle chaque protagoniste pourra effectuer diverse action. Tel que la commande en ligne par le client, suivre son évolution, permettre aux préparateurs la consultation des recettes...

Externalisation de l'hébergement de l'application sur un Cloud en mode PAAS. La plateforme Héroku a été retenu par la direction d'OcPizza lors du dernier entretien.

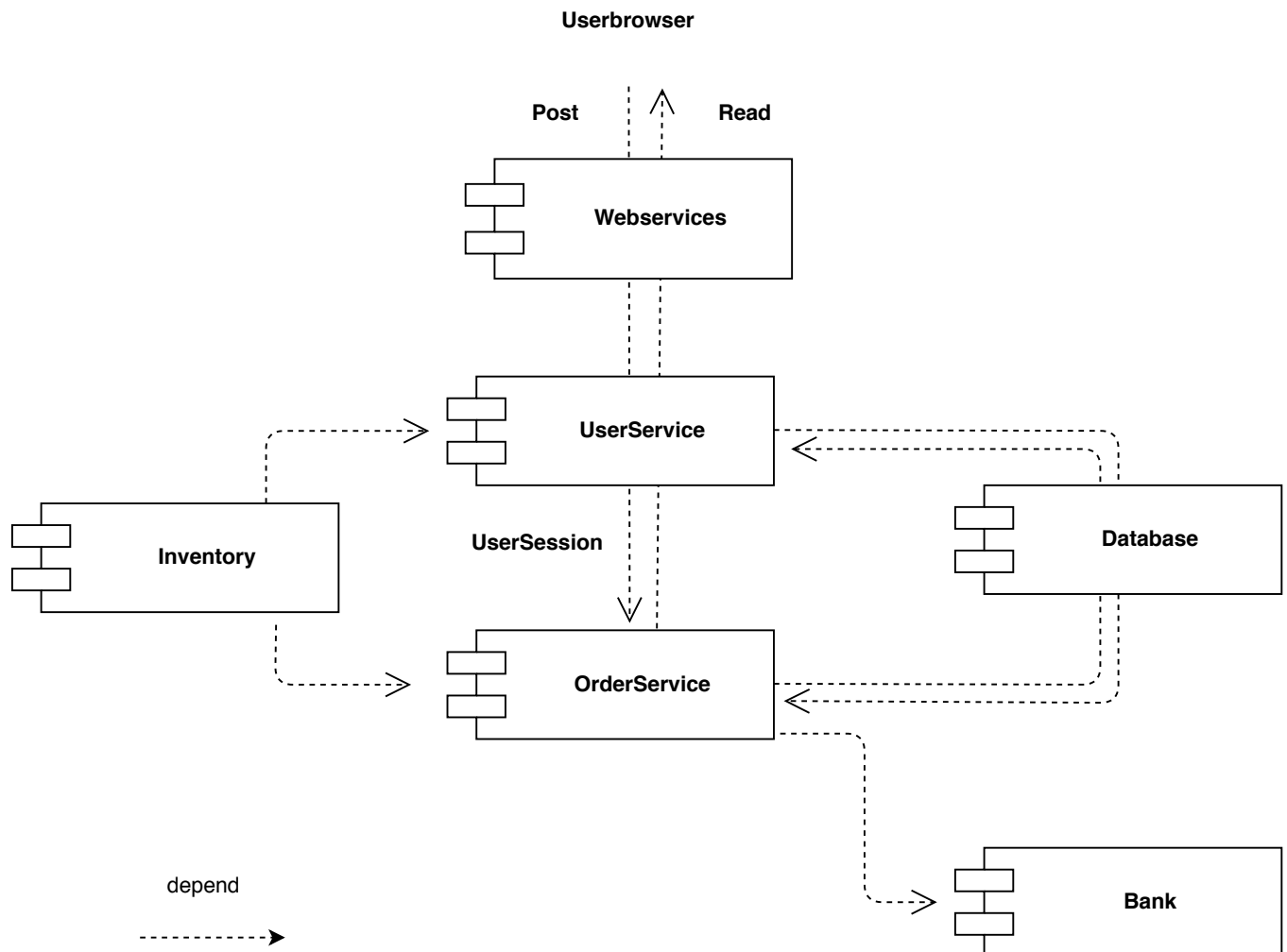
La base de données utilisée pour le fonctionnement de l'application utilisera le système de gestion de base de données PostgreSQL.

Pour la gestion des paiements il faudra faire appel à un prestataire de services de paiement qui permettra de gérer le paiement des commandes réalisés à partir du site en ligne.

1.1 COMPOSANTS

Description de l'organisation du système afin de mettre en évidence les dépendances entre les composants

L'utilisateur communique avec l'application distante via le web service utilisant le protocole HTTP comme moyen de transport, ainsi les communications s'effectuent sur un support universel.



1.1 DEPLOIEMENT

Le diagramme suivant sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux.

Il se compose de quatre noeuds :

User device représente la machine depuis laquelle l'utilisateur consulte le site web (*pc, tablette, smartphone*), avec un composant Front-end partie visible de l'application, destinée à être manipulée par un tiers.

Application server le serveur sur lequel sera stocké l'application, avec le composant Backend qui va permettre la liaison entre ces différents noeuds.

Database server le serveur sur lequel sera stocké la base de données, avec le composant database qui va stocker la totalité des informations relatives à l'activité d'OcPizza (*données utilisateurs, stocks...*).

Et enfin **Bank server** qui représente le prestataire de services de paiement avec le composant Bank transaction qui va permettre de gérer les règlements effectués à partir du site en ligne.

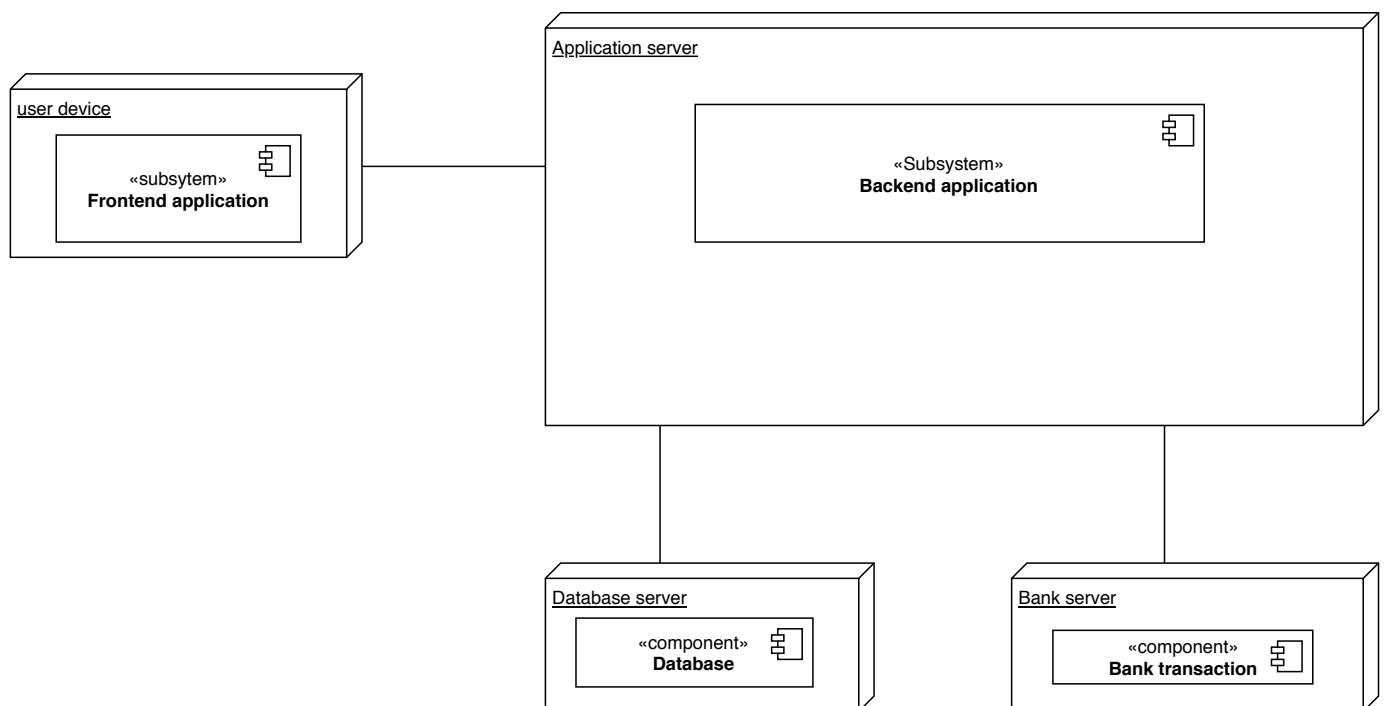


Diagramme de déploiement

Le diagramme suivant représente les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci.

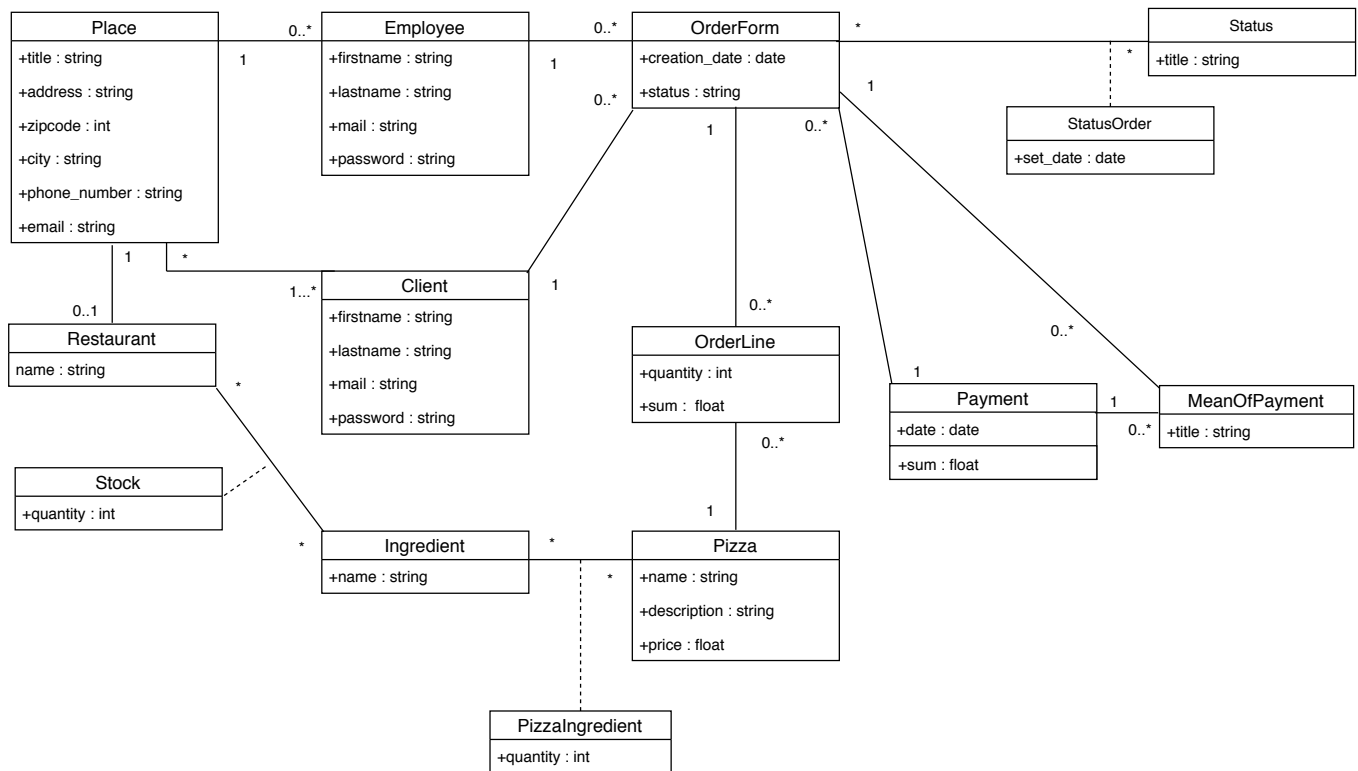


Diagramme de classes

Place : cette classe va contenir la totalité des adresses (Clients, Employés, Restaurants).

Employee et **Client** : contiennent les informations sur les différents utilisateurs, (prénoms, noms, mails, mots de passe).

Restaurant : le nom du restaurant

Stock : le stock sera géré par restaurant cette classe contient les quantités d'ingrédients disponibles.

Ingredient : le nom de chaque ingrédient

Pizza : Chaque pizza est matérialisé par un nom, une description et un prix.

PizzaIngredient : contient les ingrédients et leurs portions, nécessaires pour la réalisation d'une pizza.

OrderForm : Classe centrale de l'application elle contient les informations nécessaires pour la gestion de commande par conséquent elle fait le lien avec la plupart des autres classes.

OrderLine : une extension de la classe OrderForm elle sera matérialisé par la quantité d'article commandé et le coût total.

Status : contient les différentes étapes d'une commande.

StatusOrder : affiche en temps réel le statut de la commande.

MeanOfPayment : contient les types de paiement.

Payment : reprend des éléments de la commande à savoir le coût, la date, son numéro et le mode de paiement.

Traduction du diagramme de classes en modèle physique de données avec ajout des clés primaires et des clés étrangères.

