



IT Consulting & Development

OC PIZZA

Application web et gestion OC Pizza

Dossier de conception technique

Version 1.0

Auteur

Tony Gonzalez
Analyste programmeur

TABLE DES MATIERES

1 - Versions.....	3
2 - Introduction.....	4
2.1 - Objet du document	4
3 - Le domaine fonctionnel.....	5
3.1 - Le diagramme de classe	5
3.1.1 - La classe User	6
3.1.2 - La classe Command	6
3.1.3 - La classe CommandDetail.....	7
3.1.4 - La classe Pizza	7
3.1.5 - La classe Recipe	7
3.1.6 - La classe Ingredient.....	8
3.1.7 - La classe Stock	8
3.1.8 - La classe Local.....	8
4 - Architecture technique	9
4.1 - Le modèle physique de données.....	9
4.2 - Le diagramme de composant	10
5 - Architecture de Déploiement.....	11
6 - Les logiciels utilisés.....	12
6.1 - Base de données.....	12
6.2 - Partie front-office	12
6.3 - Partie back-office	13
7 - Glossaire.....	14

1 - VERSIONS

Auteur	Date	Description	Version
Tony Gonzalez	01/10/2020	Création du document	1.0

2 - INTRODUCTION

2.1 - Objet du document

Le présent document constitue le dossier de conception technique de l'application OC Pizza.

L'objectif de ce document est d'expliquer en détail la solution technique de notre application web pour le groupe OC Pizza.

Les éléments du présents dossiers découlent :

- de l'analyse du domaine fonctionnel
- de l'analyse de l'architecture technique

3 - LE DOMAINE FONCTIONNEL

3.1 - Le diagramme de classe

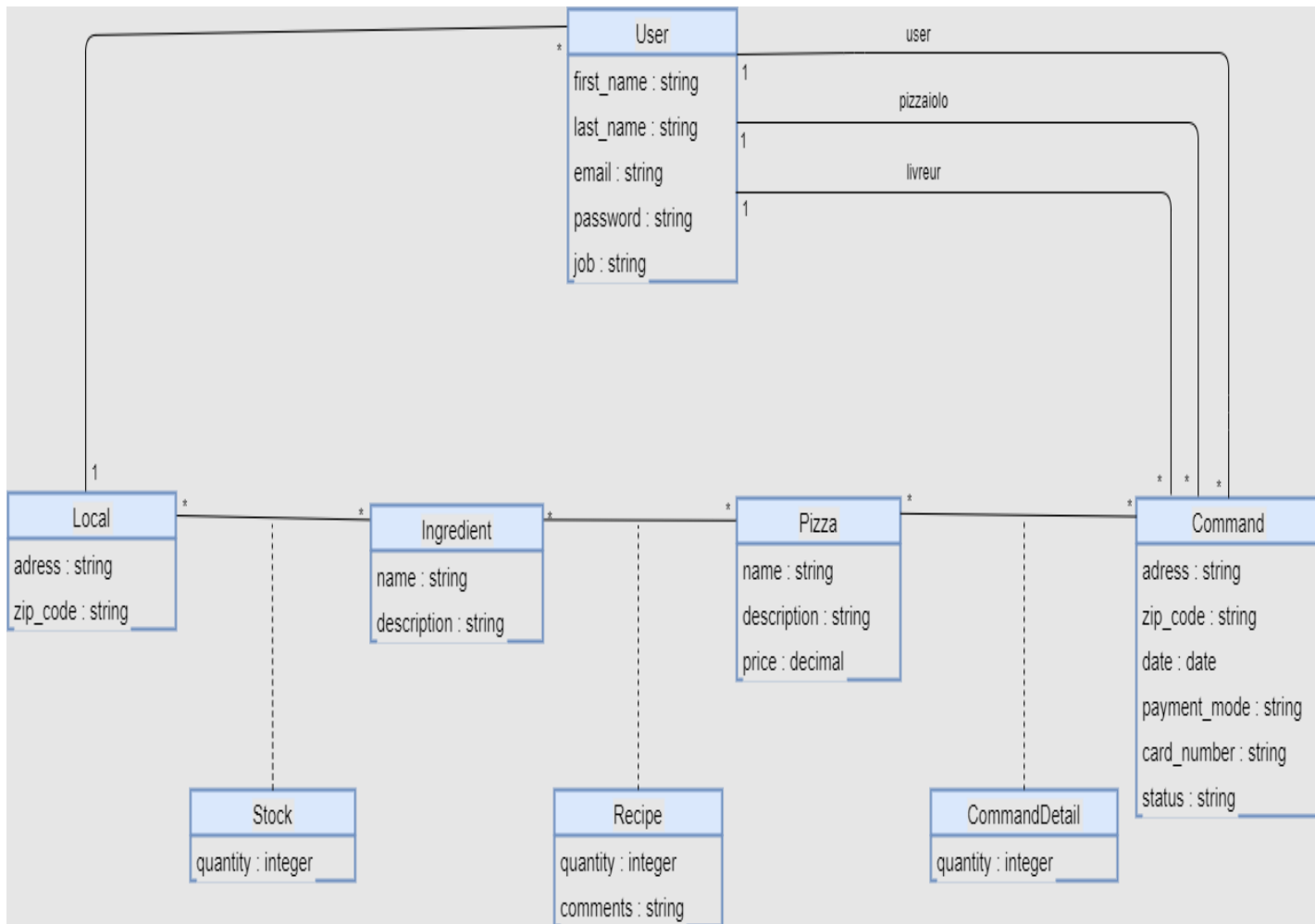


Diagramme UML de classes

Le diagramme ci-dessus, montre les différentes classes qui composeront notre application web, ainsi que leurs relations.

A noter que sur ce diagramme n'apparaissent pas les champs affectés par des clés étrangères.

3.1.1 - La classe User

Cette classe représente l'utilisateur de l'application web.

Attributs	Description
id	Identifiant
first_name	Prénom
last_name	Nom
email	Courriel de contact
password	Mot de passe
job	Poste occupé au sein du groupe OC Pizza

Il est à noter qu'un utilisateur qui n'est pas un employé du groupe Oc Pizza, utilisera une autre interface et que l'attribut « job » ne sera pas à préciser.

3.1.2 - La classe Command

Cette classe représente la commande de l'utilisateur.

Attributs	Description
id	Identifiant de la commande
adress	Adresse de livraison de la commande
zip_code	Code postal de l'adresse de livraison
date	Date de la commande
payment_mode	Mode de paiement utilisé
card_number	Numéro de carte de crédit
status	Etat de la commande

3.1.3 - La classe *CommandDetail*

Cette classe représente le détail de la commande de l'utilisateur.

Attributs	Description
quantity	Quantité commandée

3.1.4 - La classe *Pizza*

Cette classe représente les pizzas à commander.

Attributs	Description
id	Identifiant de la pizza
name	Nom de la pizza
description	Description de la pizza
price	Prix de la pizza

3.1.5 - La classe *Recipe*

Cette classe représente les recettes des pizzas qui serviront au pizzaiolo pour leur élaboration.

Attributs	Description
quantity	Quantité de l'ingrédient
comments	Commentaires sur la préparation

3.1.6 - La classe Ingredient

Cette classe représente les ingrédients disponibles pour la préparation des pizzas.

Attributs	Description
id	Identifiant de l'ingrédient
name	Nom de l'ingrédient
description	Description de l'ingrédient

3.1.7 - La classe Stock

Cette classe représente les stocks des pizzerias contenant les ingrédients.

Attributs	Description
quantity	Quantité

3.1.8 - La classe Local

Cette classe représente les différentes pizzerias.

Attributs	Description
id	Identifiant de la pizzeria
adress	Adresse de la pizzeria
zip_code	Code postal de l'adresse du local

4 - ARCHITECTURE TECHNIQUE

4.1 - Le modèle physique de données

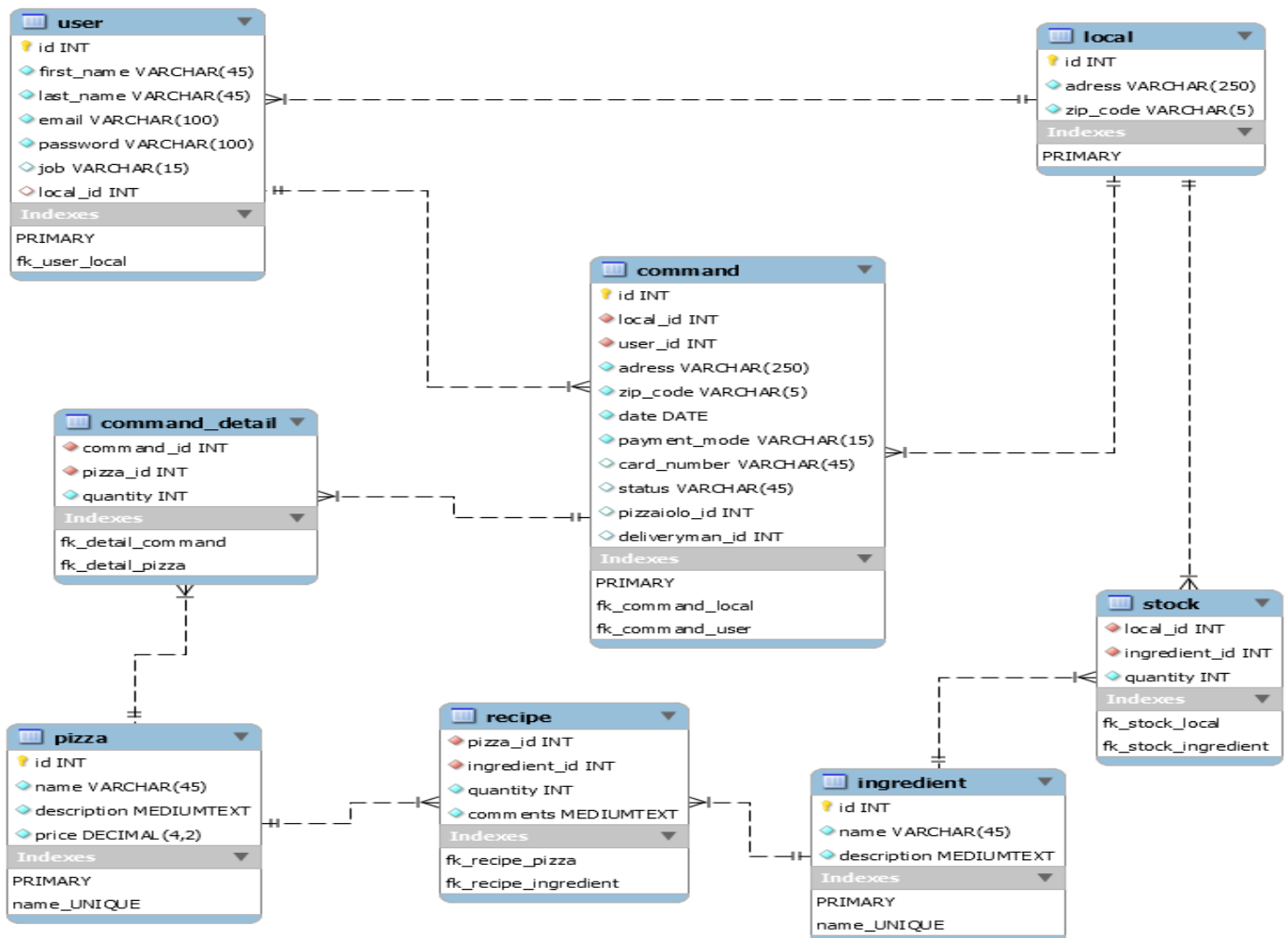


Diagramme de modèle physique de données

Le modèle physique de données est construit sur le modèle du diagramme de classe.

Il représente l'architecture de notre base de données.

Nous pouvons donc voir sur ce diagramme, les différentes tables et leurs champs, qui composeront la base de données de notre application, ainsi que les clés étrangères reliant certaines tables entre elles.

4.2 - Le diagramme de composant

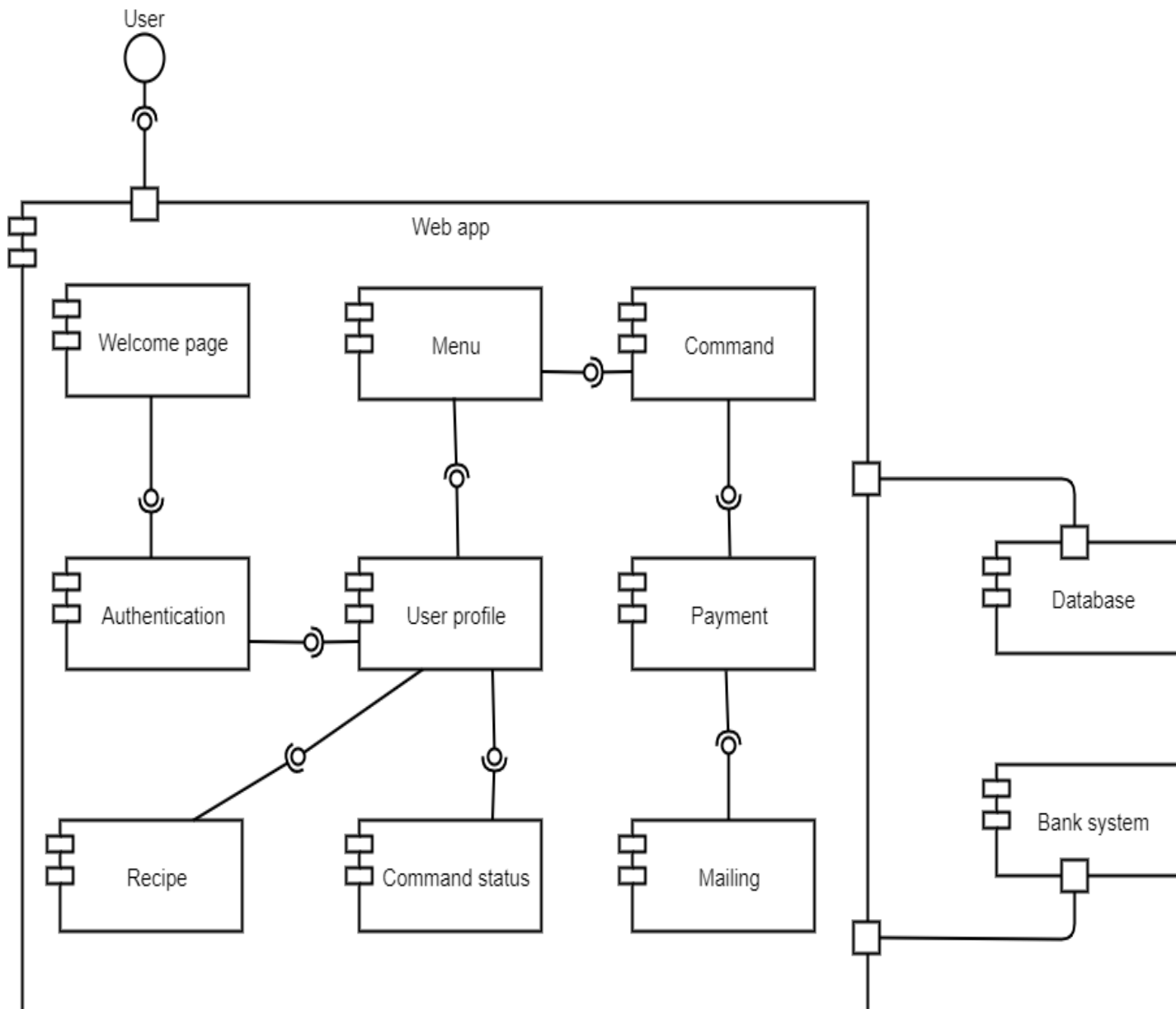


Diagramme UML de Composants

Le diagramme de composants ci-dessus décrit l'organisation du système du point de vue des éléments logiciels. Il permet de mettre en évidence les dépendances entre les composants.

5 - ARCHITECTURE DE DEPLOIEMENT

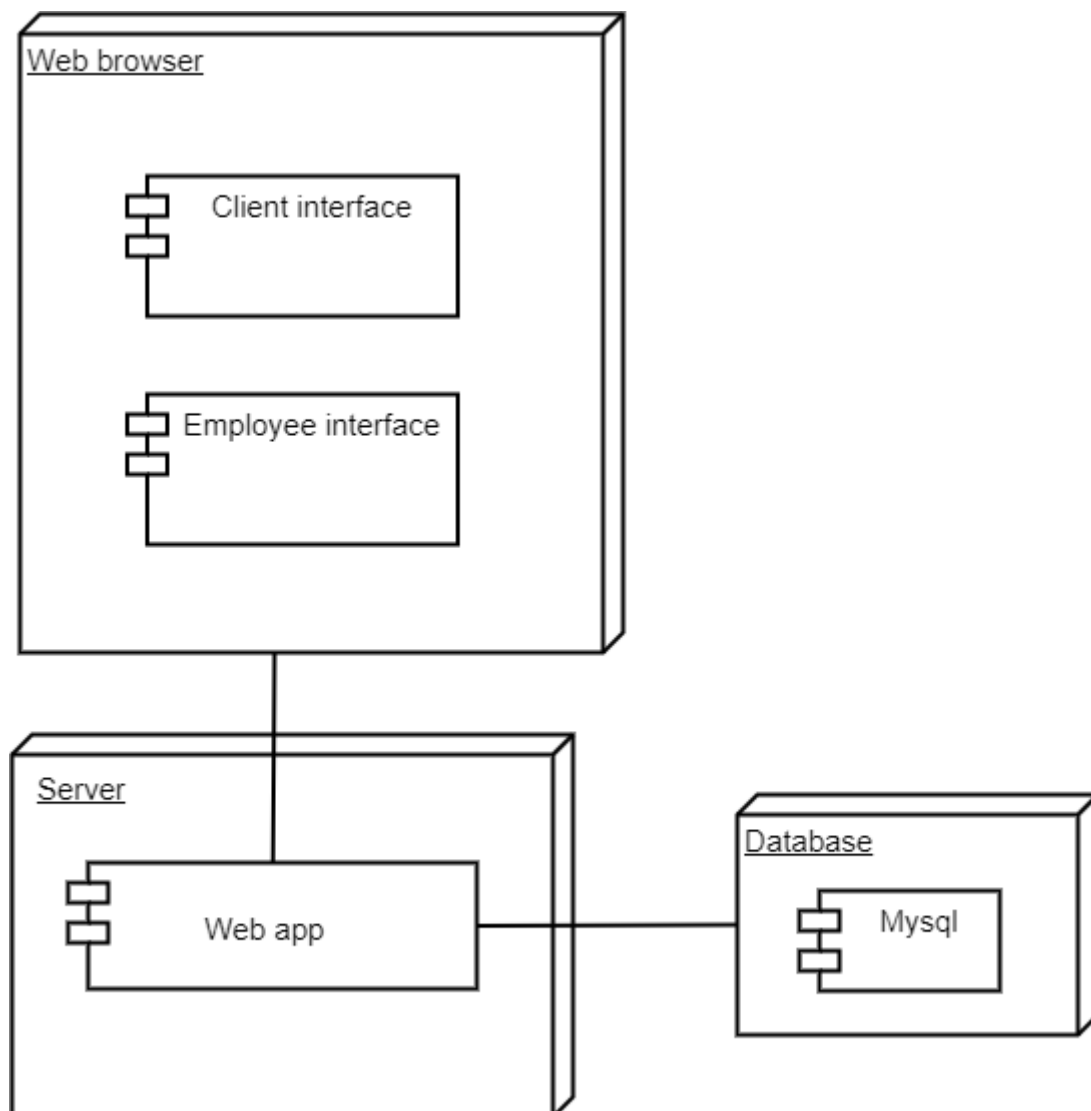


Diagramme UML de déploiement

Le diagramme de déploiement ci-dessus représente 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.

6 - LES LOGICIELS UTILISES

6.1 - Base de données

Le logiciel utilisé pour notre base de données sera MySQL.

C'est un outil idéal pour tout type de bases de données quelles que soit leurs tailles, tant pour sa sécurité que pour sa simplicité d'utilisation.



6.2 - Partie front-office

Pour la partie front-office, nous utiliserons les outils suivants :

- HTML 5
- CSS 3
- Javascript

Ces trois outils, d'utilisation simple et rapide sont les plus utilisés et les plus performants pour la création d'interface web.



6.3 - Partie back-office

Pour la partie back-office, nous utiliserons les langages de programmation suivants :

- Python3
- Django

Python est un langage de programmation puissant et de haut niveau. Aujourd'hui, il figure parmi les plus populaires. Instagram et Pinterest, pour ne citer qu'eux, l'utilisent.

Django est un cadre de développement web en Python. Il a pour but de rendre le développement web 2.0 simple et rapide.



7 - GLOSSAIRE

Front-office	En informatique, une application de front-office ou logiciel de front-office est la partie d'un système informatique accessible aux utilisateurs finaux ou aux clients, par opposition au back office.
Back-office	Le Back Office désigne l'ensemble des parties d'un système informatique auxquelles l'utilisateur final n'a pas accès.