

OC PIZZA

P6

Dossier de conception technique

Version 1

Auteur

Ewen Jeannenot

Analyste-programmeur



IT Consulting & Development

TABLE DES MATIERES

1 - Versions	4
2 - Introduction	5
2.1 - Objet du document	5
2.2 - Références.....	5
2.3 - Besoin du client	5
2.3.1 - Contexte.....	5
2.3.2 - Enjeux et Objectifs	5
3 - Domaine Fonctionnel	7
3.1 - Diagramme de classes	7
3.1.1 - Description diagramme de classes	7
3.2 - Modèle physique de données	9
3.2.1 - Description modèle physique de données	9
4 - Architecture Technique.....	12
4.1 - Composants généraux	12
4.1.1 - Description diagramme de composants : interface web	12
4.1.2 - Description diagramme de composants : process commande	13
4.1.3 - Description diagramme de composants : stock.....	13
4.1.4 - Description diagramme de composants : gestion	13
5 - Architecture de Déploiement	14
5.1 - Serveur de Base de données	14
5.2 - Serveur d'Application.....	14
5.3 - Utilisateurs	15
5.4 - Serveur API	15
6 - Glossaire	16

1 - VERSIONS

Auteur	Date	Description	Version
Ewen	23/05/2021	Création du document	001

2 - INTRODUCTION

2.1 - Objet du document

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

L'objectif du document est de présenter le domaine fonctionnel et l'architecture technique de la solution proposée.

Les éléments du présent dossier découlent :

- De l'étude préalable du dossier de conception fonctionnelle

2.2 - Références

Pour de plus amples informations, se référer également aux éléments suivants :

1. **DCF - 001** : Dossier de conception fonctionnelle de l'application

2.3 - Besoin du client

2.3.1 - Contexte

« OC Pizza » est un jeune groupe de pizzeria en plein essor. Créé par Franck et Lola, le groupe est spécialisé dans les pizzas livrées ou à emporter. Il compte déjà 5 points de vente et prévoit d'en ouvrir au moins 3 de plus d'ici 6 mois. Le système informatique actuel ne correspond plus aux besoins du groupe car il ne permet pas une gestion centralisée de toutes les pizzerias. De plus, il est très difficile pour les responsables de suivre ce qui se passe dans les points de ventes. Enfin, les livreurs ne peuvent pas indiquer « en live » que la livraison est effectuée.

2.3.2 - Enjeux et Objectifs

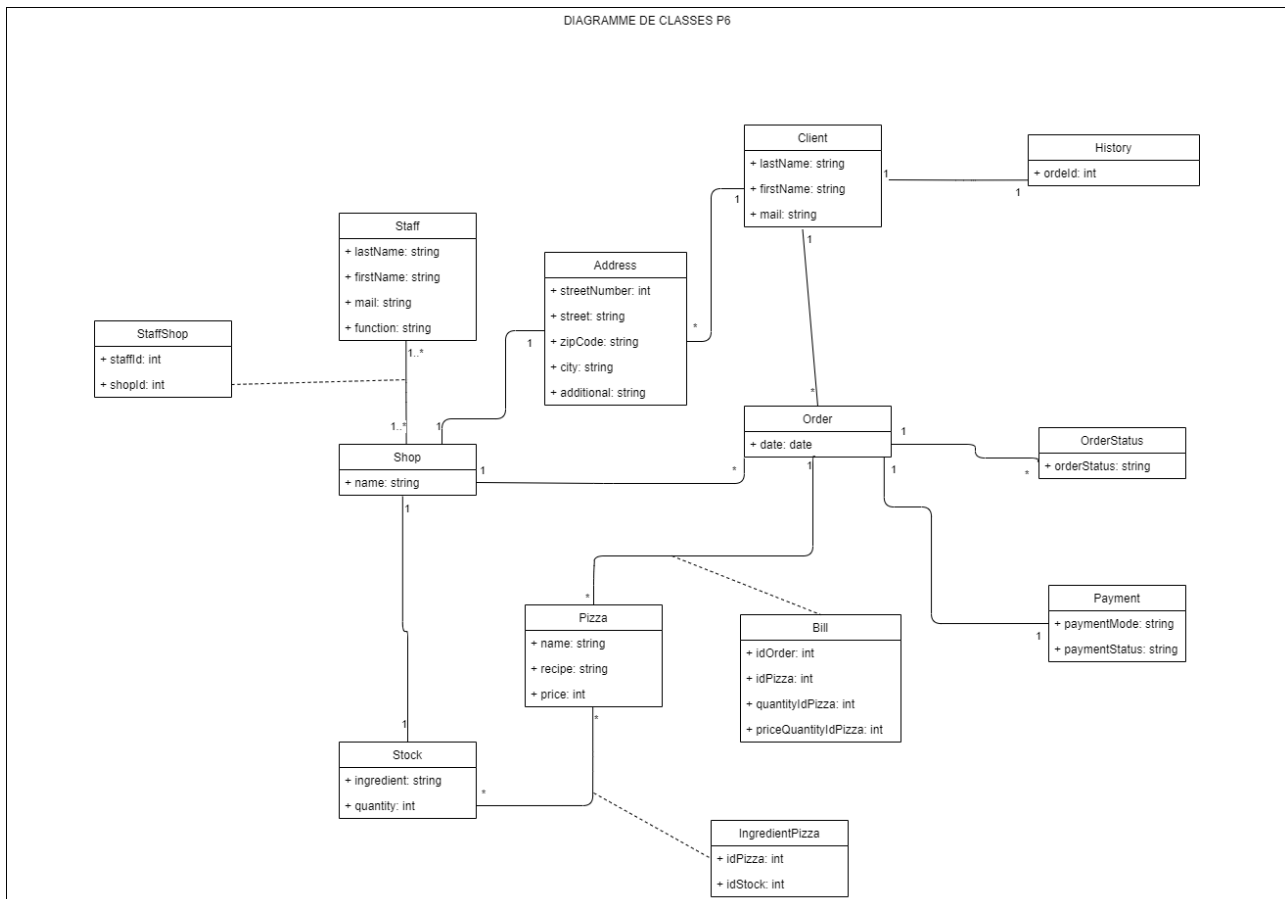
Besoins exprimés par le client :

- être plus efficace dans la gestion des commandes, de leur réception à leur livraison en passant par leur préparation ;
- suivre en temps réel les commandes passées, en préparation et en livraison ;
- suivre en temps réel le stock d'ingrédients restants pour savoir quelles pizzas peuvent encore être réalisées ;
- proposer un site Internet pour que les clients puissent :

- o passer leurs commandes, en plus de la prise de commande par téléphone ou sur place ;
 - o payer en ligne leur commande s'ils le souhaitent – sinon, ils paieront directement à la livraison ;
 - o modifier ou annuler leur commande tant que celle-ci n'a pas été préparée.
- proposer un aide-mémoire aux pizzaiolos indiquant la recette de chaque pizza.
 - d'informer ou notifier les clients sur les l'état de leur commande

3 - DOMAINE FONCTIONNEL

3.1 - Diagramme de classes

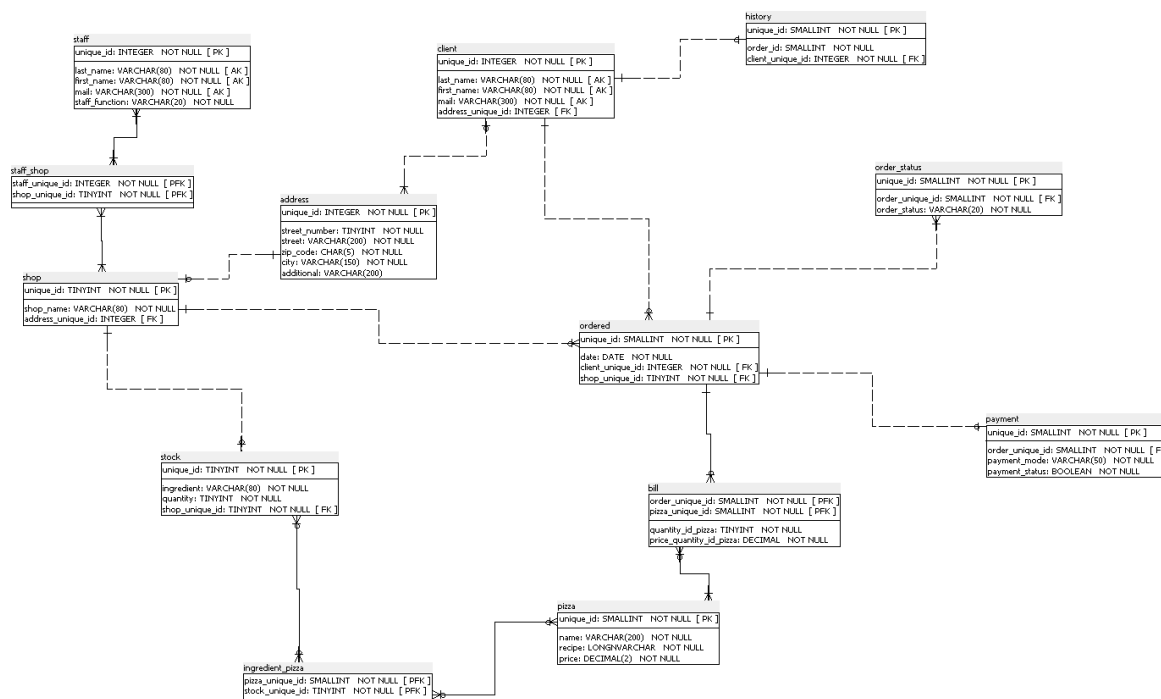


3.1.1 - Description diagramme de classes

CLASSE	ATTRIBUT	DESCRIPTION
Staff	lastName	Nom de l'employé
	firstName	Prénom de l'employé
	mail	Adresse email de l'employé
	function	Fonction de l'employé
Shop	Name	Nom du point de vente

Address	streetNumber	Numéro de la rue
	Street	Nom de la rue
	zipCode	Code postale de la ville
	City	Ville
	Additional	Complément d'information
Client	lastName	Nom de famille
	firstName	Prénom
	Mail	Adresse mail du client
History	orderId	Numéro des commandes
Order	Date	Date de la commande
OrderStatus	orderStatus	Etat de la commande
Pizza	Name	Nom de la pizza
	Recipe	Recette de la pizza
	price	Prix de la pizza
Stock	Ingredient	Nom d'un ingrédient
	Quantity	Quantité en stock de l'ingrédient
Payment	paymentMode	Choix du mode de paiement
	paymentStatus	Etat du paiement
Bill : Classe d'association entre Order et Pizza	quantityIdPizza	Quantité d'une pizza dans la commande
	priceQuantityIdPizza	Prix correspondant pour la quantité d'une même pizza
IngredientPizza : Classe d'association entre Pizza et Stock		
StaffShop : Classe d'association entre Staff et Shop		

3.2 - Modèle physique de données



3.2.1 - Description modèle physique de données

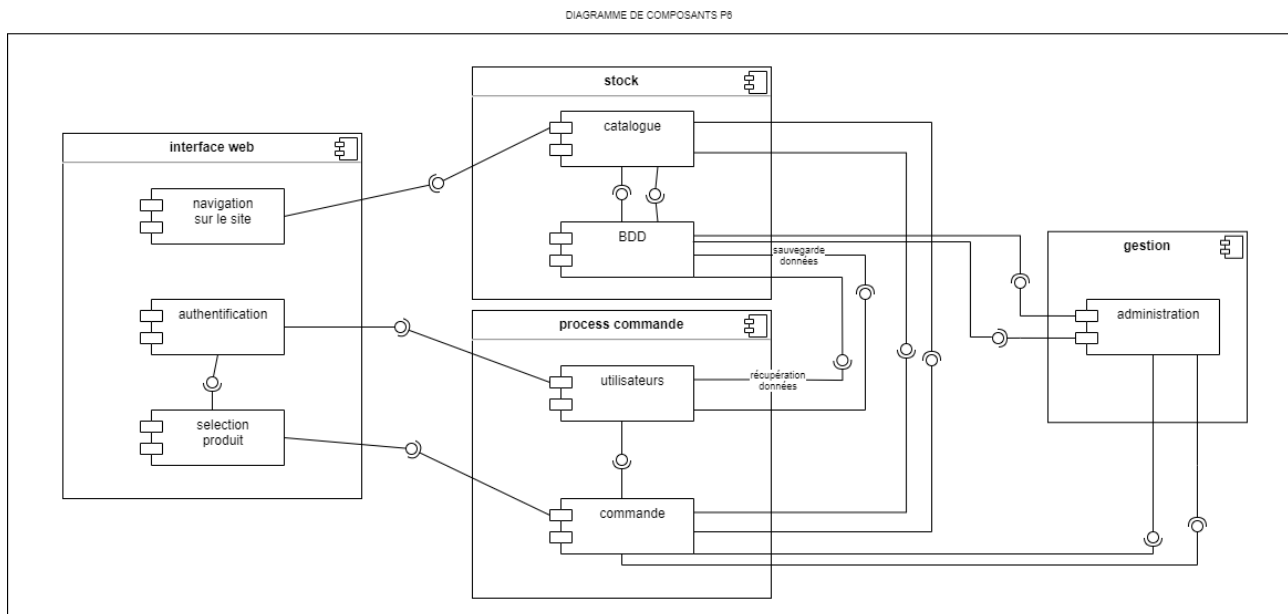
CLASSE	ATTRIBUT	DESCRIPTION
staff	unique_id	PK / INTEGER / NOT NULL
	lastName	VARCHAR(80) / NOT NULL / Index contrainte unicité
	firstName	VARCHAR(80) / NOT NULL / Index contrainte unicité
	mail	VARCHAR(80) / NOT NULL / Index contrainte unicité
	function	VARCHAR(20) / NOT NULL
shop	unique_id	PK / TINYINT / NOT NULL
	name	VARCHAR(80) / NOT NULL
	Adress_unique_id	FK / INTEGER / NOT NULL

address	unique_id	PK / SMALLINT / NOT NULL
	streetNumber	TINYINT / NOT NULL
	Street	VARCHAR(200)
	zipCode	TINYINT / NOT NULL
	city	VARCHAR(150)
	additional	Varchar(200)
	client_unique_id	FK / INTEGER / NOT NULL
Client	unique_id	PK / INTEGER / NOT NULL
	lastName	VARCHAR(80) / NOT NULL / Index contrainte unicité
	firstName	VARCHAR(80) / NOT NULL / Index contrainte unicité
	Mail	LONGNVARCHAR / NOT NULL / Index contrainte unicité
History	unique_id	PK / SMALLINT / NOT NULL
	order_id	SMALLINT / NOT NULL
	client_unique_id	FK / INTEGER / NOT NULL
Ordered	unique_id	PK / SMALLINT / NOT NULL
	Date	DATE / NOT NULL
	client_unique_id	FK / INTEGER / NOT NULL
	shop_unique_id	FK / TINYINT / NOT NULL
Order_Status	unique_id	PK / SMALLINT / NOT NULL
	order_unique_id	FK / SMALLINT / NOT NULL
	Order_status	VARCHAR(20) / NOT NULL
Pizza	unique_id	PK / SMALLINT / NOT NULL
	Name	Varchar(200) / NOT NULL
	Recipe	LONGNVARCHAR / NOT NULL
	price	DECIMAL(4,2) / NOT NULL
Stock	unique_id	PK / TINYINT / NOT NULL
	Ingredient	VARCHAR(80) / NOT NULL
	Quantity	TINYINT / NOT NULL
	shop_unique_id	FK / TINYINT / NOT NULL
Payment	unique_id	PK / SMALLINT / NOT NULL
	order_unique_id	FK / SMALLINT / NOT NULL
	paymentMode	VARCHAR(50) / NOT NULL

	paymentStatus	BOOLEAN / NOT NULL
Bill : Classe d'association entre Order et Pizza	Order_unique_id	PFK / SMALLINT / NOT NULL
	Pizza_unique_id	PFK / SMALLINT / NOT NULL
	quantityIdPizza	TINYINT / NOT NULL
	priceQuantityIdPizza	DECIMAL(8,2) / NOT NULL
Ingredient_Pizza : Classe d'association entre Pizza et Stock	pizza_unique_id	PFK / SMALLINT / NOT NULL
	stock_unique_id	PFK / TINYINT / NOT NULL
Staff_Shop : Classe d'association entre Staff et Shop	staff_unique_id	PFK / INTEGER / NOT NULL
	shop_unique_id	PFK / TINYINT / NOT NULL

4 - ARCHITECTURE TECHNIQUE

4.1 - Composants généraux



4.1.1 - Description diagramme de composants : interface web

- *Navigation sur le site* : gère la navigation / affichage du catalogue pour un utilisateur
- *Authentification* : gère l'authentification d'un utilisateur avec les différents droits accordés en fonction de son statut (gérant, manager, livreur, pizzaiolo et client)
- *Sélection produit* : gère les choix de l'utilisateur dans le catalogue pour passer une commande

4.1.2 - Description diagramme de composants : process commande

- *Utilisateurs*: en fonction des informations d'authentification, va permettre d'interagir avec la base de données pour envoyer et recevoir les informations des utilisateurs
- *Commande*: gère le processus d'une commande en fonction de l'utilisateur (passer et consulter pour le client, mise à jour, changement d'état...)

4.1.3 - Description diagramme de composants : stock

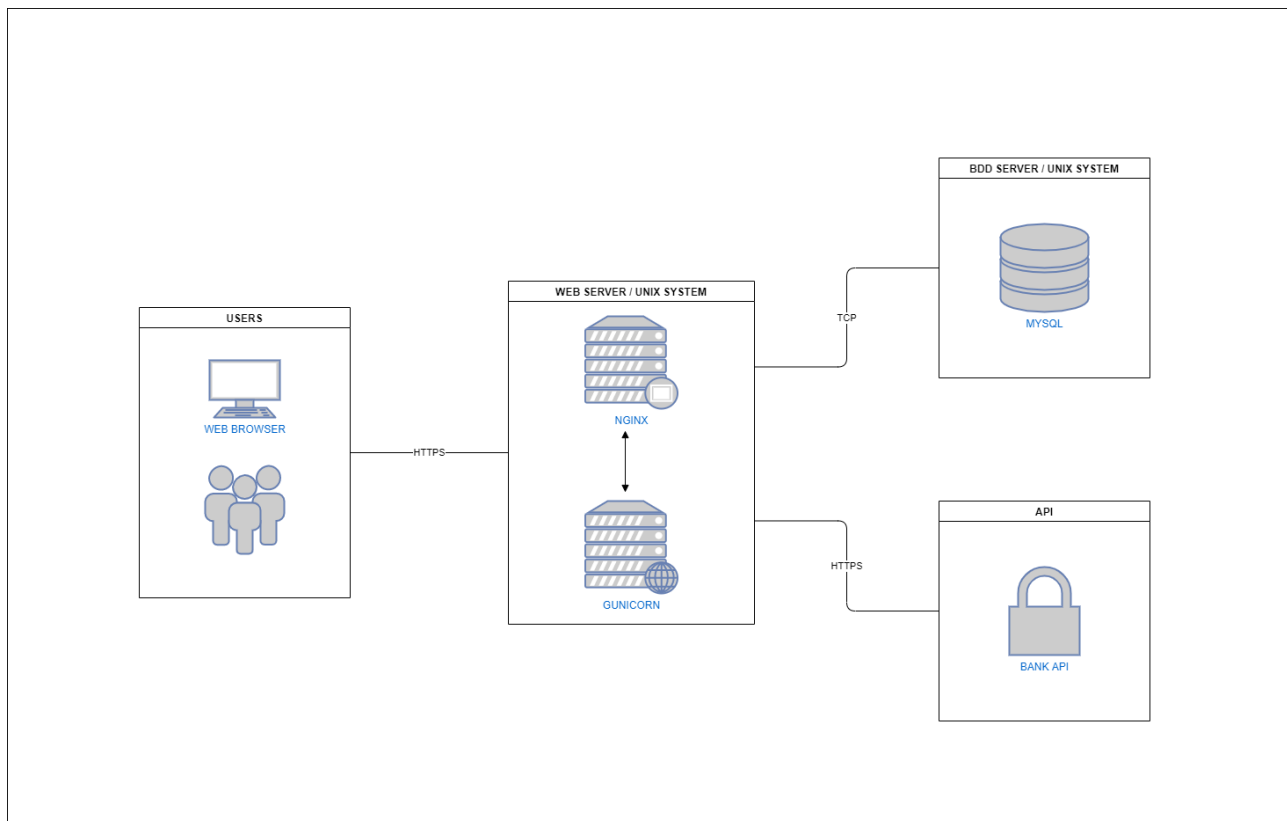
- *Catalogue*: Gère les produits du catalogue, en se mettant à jour (disponibilité du produit) en fonction des informations reçu par le composant commande.
- *BDD*: Gère les différentes informations stockées, du produit à l'utilisateur

4.1.4 - Description diagramme de composants : gestion

- *Administration*: S'occupe de la gestion des différents composants du système, stock et process commande

5 - ARCHITECTURE DE DEPLOIEMENT

Diagramme UML de déploiement



5.1 - Serveur de Base de données

Description :

Le serveur de base de données utilisera le SGBDR MySQL et sera installé sur un système unix afin de rester homogène avec les autres serveurs.

La communication entre le serveur de base de données et le serveur d'application s'effectue par le protocole TCP/IP

5.2 - Serveur d'Application

Description :

Le serveur d'application comporte le serveur web Gunicorn et le serveur Nginx pour son rôle

de reverse proxy. Tous les deux seront installés sur un système Unix en raison des problèmes de compatibilité avec windows.

Le serveur d'application communique avec l'utilisateur final ainsi que l'API Bancaire via le protocole HTTP / HTTPS.

La communication avec le serveur de base de données se fait par le protocole TCP/IP

5.3 - Utilisateurs

Description :

L'utilisateur final qu'il soit utilisateur, client ou employé va se connecter au serveur d'application à l'aide d'un navigateur web par le biais du protocole HTTP / HTTPS.

Cela pourra donc se faire depuis n'importe quel terminal pouvant utiliser un navigateur :

Système Android / Windows / UNIX avec un ordinateur ou un smartphone.

5.4 - Serveur API

Description :

La partie API bancaire reçoit les requêtes du serveur d'application web via HTTPS

6 - GLOSSAIRE
