OC PIZZA

Boutique en ligne
&
Gestionnaire centralisé

Dossier de spécifications techniques

Version 2.0

Auteur
Stephen AOGOLO
Analyste-programmeur

TABLE DES MATIERES

1 - VERSIONS

Auteur	Date	Description	Version
Stephen AOGOLO	18/05/2020	Création du document	1.0
Stephen AOGOLO	18/05/2020	Mise à jour du document – Ajout des corrections suivantes : • *Ajout* du diagramme de classe.	1.1
Stephen AOGOLO	19/05/2020	Mise à jour du document – Ajout des corrections suivantes : • *Correction* du diagramme de classe.	1.2
Stephen AOGOLO	21/05/2020	Mise à jour du document – Ajout des corrections suivantes : • *Correction* du diagramme de classe.	1.3
Stephen AOGOLO	21/05/2020	Mise à jour du document – Ajout des corrections suivantes : • *Correction* du diagramme de classe.	1.4
Stephen AOGOLO	28/05/2020	Mise à jour du document – Ajout des corrections suivantes : • *Ajout* du MPD.	1.5
Stephen AOGOLO	29/05/2020	Mise à jour du document – Ajout des corrections suivantes : • *Correction* du MPD.	1.6
Stephen AOGOLO	09/06/2020	Mise à jour du document – Ajout des corrections suivantes : • *Ajout* des diagrammes de composants et de déploiement.	1.7
Stephen AOGOLO	16/06/2020	Mise à jour du document – Ajout des corrections suivantes : *Correction* du diagramme de déploiement.	1.8
Stephen AOGOLO	16/06/2020	Mise à jour du document – Ajout des corrections suivantes : • *Correction* du diagramme de classe (cardinalités, utilisateur/commande). • *Correction* du modèle physique de données (clés primaires, etrangères). • *Correction* du diagramme de composants (suppression navigateur, interfaces). *Correction* du diagramme de déploiement (précision de localisation).	1.9
Stephen AOGOLO	14/01/2021	Mise à jour du document – Ajout des corrections suivantes : • *Correction* du type de base de données. Anciennement Mysql, Postgresql est retenue pour la gestion des données. • *Correction* du système d'exploitation. Anciennement Debian, Ubuntu est retenue pour le l'hébergement de la solution. • *Correction* des versions logicielles : Ubuntu 20.04.1, Gunicorn 20.0.4, Postgresql 13.	2.0

2 - INTRODUCTION

2.1 - Objet du document

Ce dossier de spécifications techniques (***version 2.0***) fait suite au dossier de spécifications fonctionnelles (***version 1.4***). Il reprend la perception et l'organisation fonctionnelle de ce dernier.

Ce dossier a pour objectif d'apporter une analyse technique approfondie afin de décrire par des moyens réels et existants, la conception et la réalisation du projet. Ce document présente et décrit donc la solution technique du projet **OC PIZZA – Boutique en ligne & Gestionnaire centralisé**.

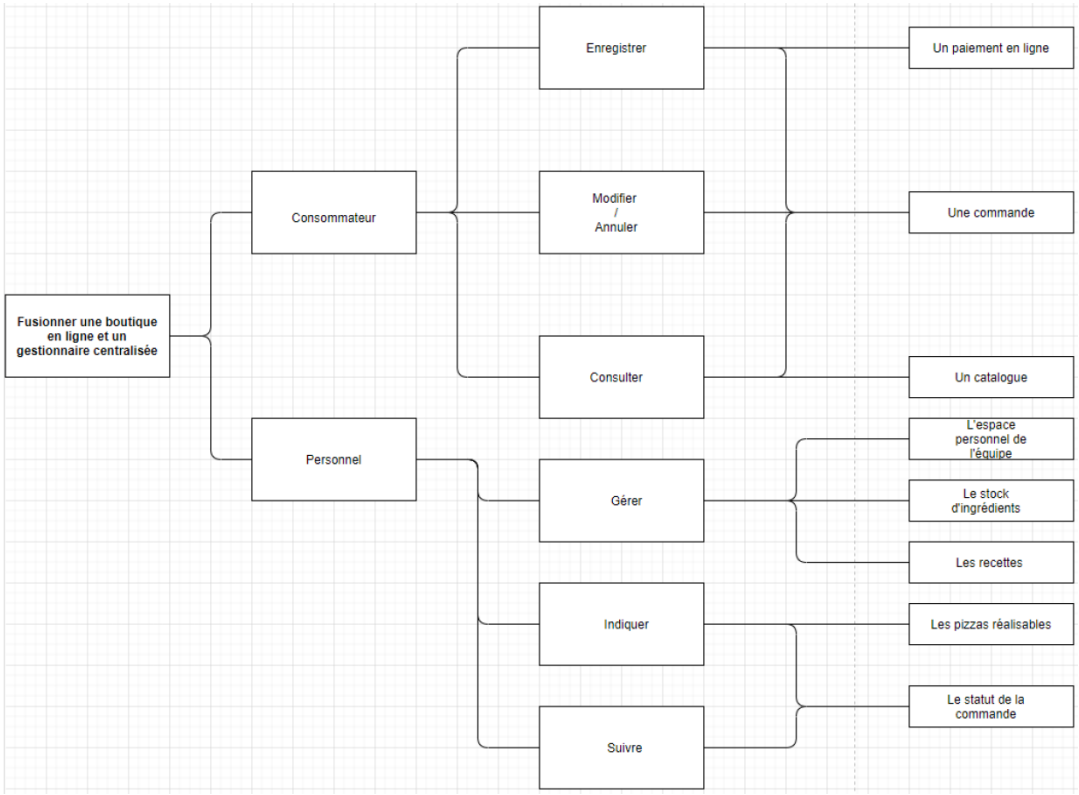
La solution technique est décrite sous trois aspects :

- La description du domaine fonctionnel.
- L'énumération des composants vitaux du système.
- L'organisation physique des composants vitaux du système.

3 - LE DOMAINE FONCTIONNEL

3.1 - Référentiel

Pour rappel, le schéma ci-dessous permet de reprendre le périmètre fonctionnel, identifié durant la phase d'élaboration des spécifications fonctionnelles. (***3.3.1- Impact mapping - spécifications fonctionnelles - version 1.4)*** :



Le tableau suivant reprend une brève description de la solution retenue, en couplant aux besoins fonctionnels les solutions fonctionnelles identifiées (***6. Application Web – Spécifications fonctionnelles – version 1.4***) :

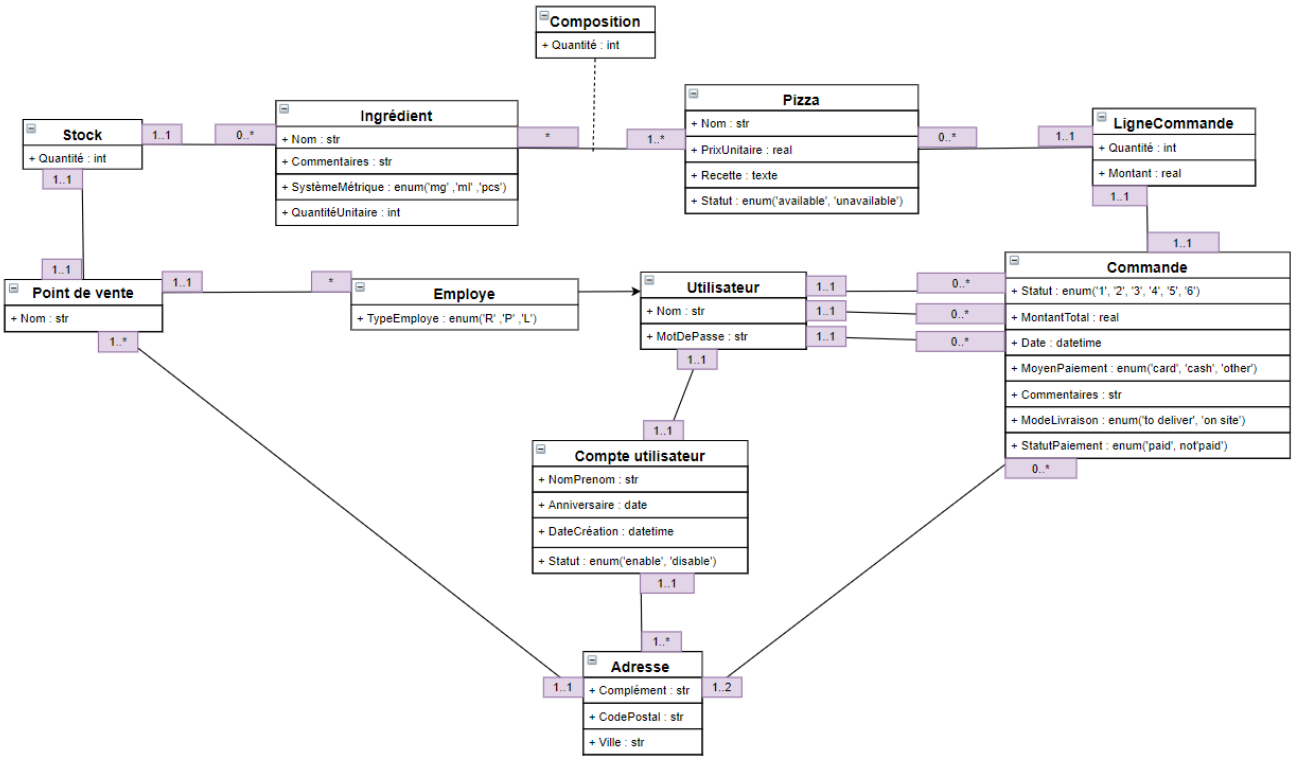
Acteurs cibles	*Besoins Fonctionnels*	*Solutions Fonctionnelles*
Personnel	Une gestion centralisée de toutes les pizzerias…	6.1 | D1 | D2
Personnel	Suivre ce qui se passe dans les points de ventes…	4.4 | 4.5 | 6.1 | D2
Personnel	Indiquer que la livraison est effectuée…	4.1 | 6.1
Personnel	Gestion des commandes, de leur réception à leur livraison …	4.1 | 4.4 | 4.5 | 6.1
Personnel	Suivre en temps réel les commandes …	4.4 | 4.5 | 6.1
Personnel	Suivre … le stock d'ingrédients…	6.2 | D2
Personnel	Savoir quelles pizzas peuvent encore être réalisées…	4.4 | D1
Personnel | Clientèle	Site internet…	Back Office | Front Office
Clientèle	Passer les commandes…	1.2
Clientèle	Payer en ligne…s'ils le souhaitent…sinon..à la livraison…	B3 | B31 | B32
Clientèle	Modifier Annuler leur commande.. Tant que celle-ci n'a pas été préparée…	1.3 | 1.4
Personnel	Proposer…aide mémoire..recette..chaque pizza…	6.3 | D1

	INDICE	CAS D'UTILISATION
PRINCPALES	1.1	Consulter catalogue
	1.2	Enregistrer une commande
	1.3	Modifier une commande
	1.4	Annuler une commande
	2.1	Créer un compte
	3.1	Consulter l'historique
	3.2	Obtenir une promotion
	4.1	Modifier statut commande
	4.4	Consulter les pizzas réalisables
	4.5	Consulter les commandes à réaliser
	6.1	Suivre le statut de la commande
	6.2	Modifier le stock d'ingrédients
	6.3	Modifier les recettes de pizzas
	7.1	Gestion utilisateurs
INTERNES	A1	Fournir informations personnelles
	B1	Constituer panier
	B2	Obtenir récapitulatif
	B3	Payer une commande
	B5	Consulter une commande
	B31	Payer en ligne
	B32	Payer à la livraison
	C1	S'authentifier
	C11	Authentification interne
	C12	Authentification externe
	D1	Consulter les recettes de pizzas
	D2	Consulter le sotck d'ingrédients

L'ensemble des besoins fonctionnels délimitent le périmètre fonctionnel de la solution. C'est donc dans ce domaine fonctionnel que la solution proposée s'articulera.

3.2 - Le diagramme de classes

Dans le domaine fonctionnel, précédemment décrit, le corps de la solution se constitue d'un ensemble de classes. Chaque classe a pour but d'isoler ou de regrouper les éléments sujets des fonctionnalités du système. Cet ensemble de classe est schématisé par le diagramme qui suit.



3.2.1 - Règles de gestion

Ci-dessous, chaque classe ainsi que leurs attributs sont décrits

3.2.1.1 - Utilisateur

Utilisateur est une super classe. Les attributs suivants sont propres à celle-ci :

- ***Nom*** : attribut reprenant le nom et le prénom de la personne concernée. ***[TYPE : str]***
- ***MotDePasse*** : attribut reprenant le mot de passe de la personne concernée. ***[TYPE : str]***

3.2.1.2 - Employe

Cette classe hérite de la super-classe Utilisateur, elle reprend donc les attributs de cette dernière. Les attributs suivants sont propres à cette classe :

- ***TypeEmploye*** : code précisant le type de personnel de la personne concerné. ***[TYPE : enum(R, P, L)]***

3.2.1.3 - Compte utilisateur

Cette classe gère les espaces utilisateurs, Employé et Client :

- ***NomPrenom*** : attribut reprenant le nom et le prénom de la personne concernée. ***[TYPE : str]***
- ***Anniversaire*** : attribut reprenant la date de naissance de la personne concernée. ***[TYPE : date]***
- ***DateCreation*** : attribut reprenant la date de création de l'espace utilisateur. ***[TYPE : int]***
- ***Statut*** : attribut informant sur le statut d'activité de l'espace utilisateur. ***[TYPE : enum(actif, inactif)]***

3.2.1.4 - Adresse

Cette classe gère l'ensemble des informations constituant une adresse postale et numérique :

- ***Complement*** : attribut reprenant la 1ère partie de l'adresse postale (porte, étage, bâtiment, numéro, rue). ***[TYPE : str]***
- ***CodePostal*** : attribut reprenant la 2nd partie de l'adresse postale (code postale, cedex). ***[TYPE : str]***
- ***Ville*** : attribut reprenant la 3ème partie de l'adresse postale (ville). ***[TYPE : str]***
- ***Pays*** : attribut reprenant la dernière partie de l'adresse postale (pays). ***[TYPE : str]***
- ***Email*** : attribut reprenant la 1ère partie de l'adresse numérique (courriel). ***[TYPE : str]***
- ***Telephone*** : attribut reprenant la seconde partie de l'adresse numérique (téléphone). ***[TYPE : str]***

3.2.1.5 - Commande

Cette classe gère les informations qui constituent une commande :

- ***Statut*** : attribut informant sur l'état de préparation d'une commande. ***[TYPE : enum(1,2,3,4,5,6)]***
- ***MontantTotal*** : attribut informant le montant total de la commande concernée. ***[TYPE : real]***
- ***Date*** : attribut informant sur l'horodatage de la commande concernée. ***[TYPE : datetime]***
- ***MoyenPaiement*** : attribut reprenant les différents moyens de paiement. ***[TYPE : enum(carte_bancaire, espece, autre)]***
- ***Commentaires*** : attribut reprenant des commentaires de précision liées à la commande concernée. ***[TYPE : str]***
- ***ModeLivraison*** : attribut informant sur le type de livraison (à livrer ou sur place). ***[TYPE : enum(a_livrer, sur_place)]***
- ***StatutPaiement*** : attribut informant l'état de paiement de l'opération (payée ou non-payée). ***[TYPE : enum(paiement_effectue, paiement_non_effectue)]***

3.2.1.6 - Stock

Cette classe gère les informations relatives au contenu de l'entrepôt :

- ***Quantite*** : attribut informant sur la quantité d'un ingrédient présent dans l'entrepôt. ***[TYPE : int]***

3.2.1.7 - Pizza

Cette classe gère les informations relatives à une pizza :

- ***Nom*** : attribut reprenant le nom de la pizza concernée. ***[TYPE : str]***
- ***PrixUnitaire*** : attribut reprenant le prix unitaire de la pizza concernée. ***[TYPE : real]***
- ***Recette*** : attribut informant sur la recette de préparation de la pizza concernée. ***[TYPE : text]***
- ***Statut*** : attribut informant sur la disponibilité de la pizza concernée. ***[TYPE : enum(disponible, indisponible)]***

3.2.1.8 - Ingrédient

Cette classe gère les informations relatives à un ingrédient :

- ***Nom*** : attribut reprenant le nom de l'ingrédient concernée. ***[TYPE : str]***
- ***Commentaires*** : attribut reprenant des commentaires de précision liées à l'ingrédient concernée. ***[TYPE : str]***
- ***SystèmeMetrique*** : attribut informant sur les unités de mesure de l'ingrédient concerné (mg, ml, pièce). ***[TYPE : enum(g, ml, pieces)]***
- ***QuantiteUnitaire*** : attribut informant sur la quantité unitaire d'un ingrédient. Par exemple, Farine est considéré comme un ingrédient lorsque sa quantité est de 500g. ***[TYPE : int]***

3.2.1.9 - Point de Vente

Cette classe gère les informations relatives à un point de vente :

- ***Nom*** : attribut reprenant le nom du point de vente concernée. ***[TYPE : str]***

3.2.1.10 - Ligne Commande

Cette classe gère la sélection de pizzas vis-à-vis d'une commande :

- ***Quantite*** : attribut informant sur la quantité de pizzas sélectionnées pour une commande. ***[TYPE : int]***
- ***Montant*** : attribut informant sur le montant total de la sélection. ***[TYPE : real]***

3.2.1.11 - Composition

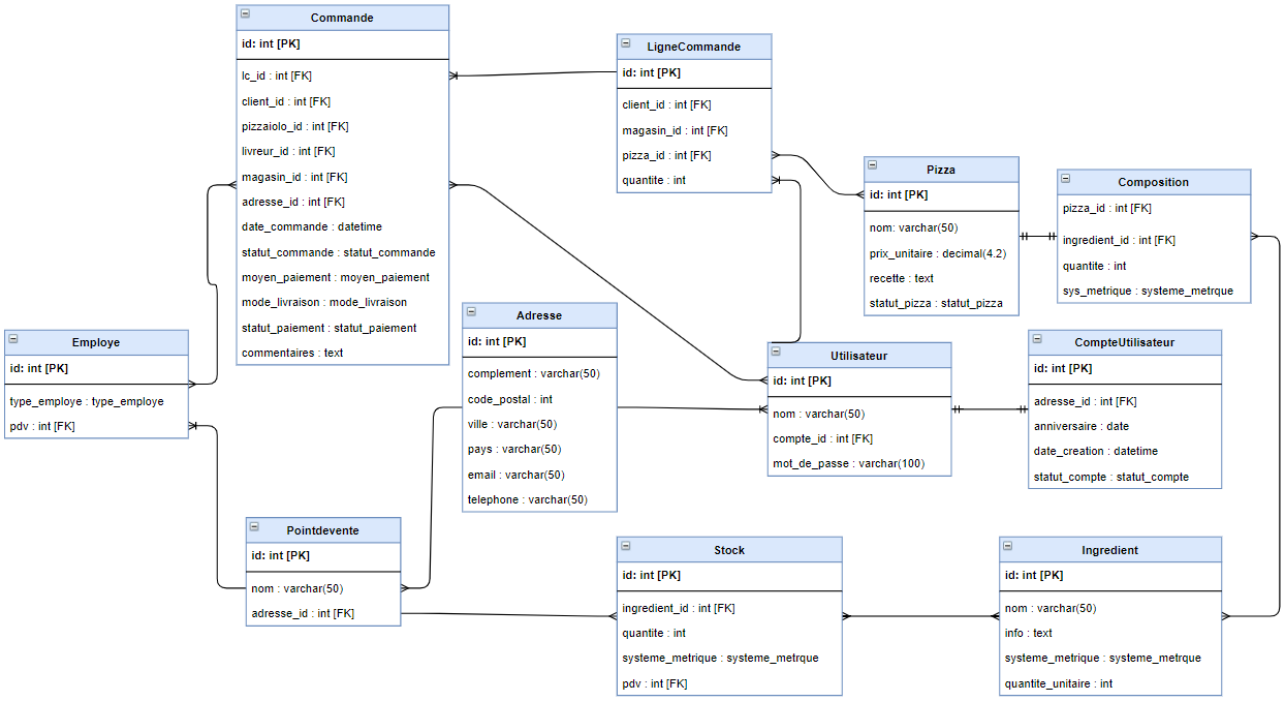
Cette association regroupe les ingrédients qui constituent chaque pizza :

- ***Quantite*** : attribut informant sur la quantité d'ingrédients constituant une pizza. ***[TYPE : int]***

3.3 - Le modèle physique de données

Le diagramme de classe décrit précédemment, met en lumière les instances existantes de la solution. Ces instances manipulent des données qui seront stockées dans une base de données prévu à cet effet. Afin de structurer le stockage des données manipulées par ces-dernières, l'élaboration d'un modèle physique de données est nécessaire pour visualiser clairement l'organisation de la base de données.

Ci-dessous, se trouve le modèle physique de données du système proposée.



3.3.1 - Règles de gestion

Ci-dessous, une description sommaire de chaque table constituant le MPD.

3.3.1.1 - Table Utilisateur

Utilisateur	
id	clé primaire, identifiant l'utilisateur
nom	nom et prénom de l'utilisateur
compte_id	clé étrangère de ID Compte, identifiant l'espace personnel d'un utilisateur
mot_de_passe	mot de passe de connexion à un espace personnel

3.3.1.2 - Table Employe

Employe	
id	clé primaire, identifiant l'employé
type_employe	code précisant le poste de l'employé concerné
pdv	clé étrangère de ID PointdeVente , identifiant le point de vente

3.3.1.3 - Table CompteUtilisateur

CompteUtilisateur	
id	clé primaire, identifiant un espace personnel
adresse_id	clé étrangère de ID Adresse, identifiant l'adresse de livraison
anniversaire	date de naissance de l'utilisateur concernée
date_creation	date de création de l'espace personnel
statut_compte	statut d'activité du compte

3.3.1.4 - Table Adresse

Adresse	
id	clé primaire, identifiant une adresse de livraison
complement	1ere partie de l'adresse postale, porte etage batiment numero rue
code_postal	2nd partie de l'adresse postale, code postale cedex
ville	3ème partie de l'adresse postale, ville
pays	4ème partie de l'adresse postale, pays
email	1ere partie de l'adresse numérique, courriel
telephone	2nd partie de l'adresse numérique, téléphone

3.3.1.5 - Table Commande

Commande	
id	clé primaire, identifiant la commande
lc_id	clé étrangère de ID LigneCommande, identifiant le panier
client_id	clé étrangère de ID Utilisateur, identifiant le client
pizzaiolo_id	clé étrangère de ID Employe, identifiant le pizzaiolo
livreur_id	clé étrangère de ID Employe, identifiant le livreur
magasin_id	clé étrangère de ID PointdeVente , identifiant le point de vente
adresse_id	clé étrangère de ID Adresse, identifiant l'adresse de livraison
date_commande	date d'enregistrement de la commande
statut_commande	statut de préparation de la commande
moyen_paiement	moyen de règlement de la commande
mode_livraison	mode de livraison de la commande
statut_paiement	statut de règlement de la commande
commentaires	informations complémentaires saisies par le client

3.3.1.6 - Table Stock

Stock	
id	clé primaire, identifiant le stock d'un ingrédient
ingredient_id	clé étrangère de ID Ingredient, identifiant l'ingrédient
quantite	quantité d'ingrédients présent dans le stock concerné
système_metrique	unité de mesure attachée à l'ingrédient concernée
pdv	clé étrangère de ID PointdeVente , identifiant le point de vente

3.3.1.7 - Table Pizza

Pizza	
id	clé primaire, identifiant la pizza
nom	nom identifiant une pizza
prix_unitaire	prix unitaire de la pizza concernée
recette	recette de préparation de la pizza concernée
statut_pizza	statut informant sur la disponibilité de la pizza concernée

3.3.1.8 - Table Ingredient

Ingredient	
id	clé primaire, identifiant un ingrédient
nom	nom de l'ingrédient
info	Informations complémentaires saisies par le responsable
système_metrique	unité de mesure attachée à l'ingrédient concernée
quantite_unitaire	quantité nécessaire pour l'estimation d'une unité.

3.3.1.9 - Table PointdeVente

PointdeVente	
id	clé primaire, identifiant un point de vente
nom	nom du point de vente
adresse_id	clé étrangère de ID Adresse, identifiant l'adresse de livraison

3.3.1.10 - Table LigneCommande

LigneCommande	
id	clé primaire, identifiant une sélection de pizzas
client_id	clé étrangère de ID Utilisateur, identifiant le client
magasin_id	clé étrangère de ID PointdeVente , identifiant le point de vente
pizza_id	clé étrangère de ID Pizza, identifiant la pizza
quantite	quantité de pizza sélectionnée

3.3.1.11 - Table Composition

Composition	
pizza_id	clé étrangère de ID Pizza, identifiant la pizza
ingredient_id	clé étrangère de ID Ingredient, identifiant l'ingrédient
quantite	quantité d'ingrédients constituant une pizza
sys_metrique	unité de mesure attachée à l'ingrédient concernée

4 - Regles de gestion Architecture Technique

La solution est développée selon un ensemble de logiciels qui en constituent une pile. Dans cette partie, la pile logicielle va être décrite afin d'orienter la conception et la réalisation de la solution.

4.1 - Application Web

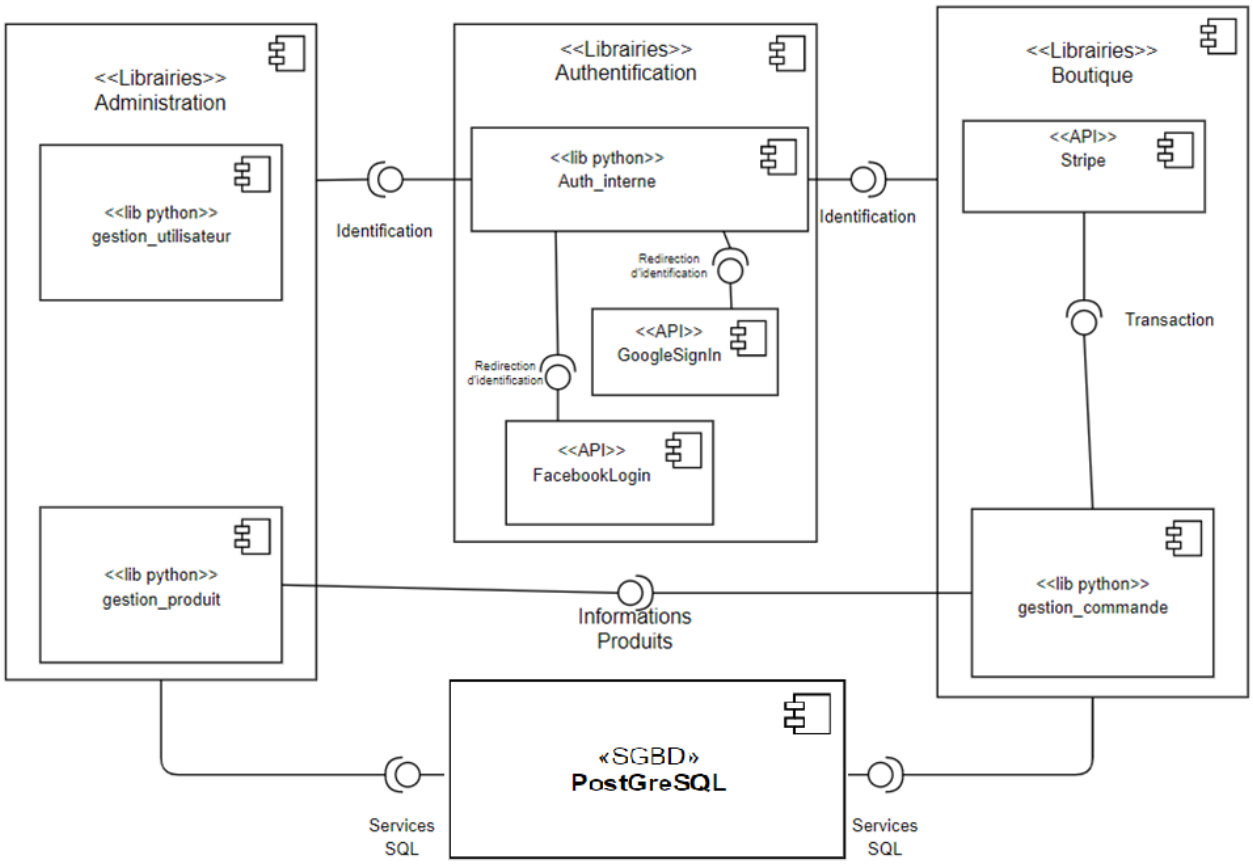
Ci-dessous se trouve l'énumération du contenu de la pile logicielle, sur laquelle est supportée la solution applicative :

- Un système d'exploitation
- Un serveur web
- Une base de données
- Plusieurs Langage de programmation
- Plusieurs interfaces de programmation applicative

4.2 - Le diagramme de composants

La solution applicative du projet s'appuie sur plusieurs composants physiques, dédiés à l'exécution des processus, des services et autres types d'opérations logicielles. Dans cette solution applicative, les composants sont majoritairement des librairies et des scripts.

Ci-dessous se trouve le diagramme de composants, reprenant ainsi la disposition de ces différents composants :



D'après le diagramme de composants, le système est constitué de cinq principaux composants. Certains composants en regroupent d'autres. Ci-dessous, une description des objectifs attachés à chaque composant.

4.2.1 - Le navigateur Web

Ce composant, externe à la solution, est un programme informatique permettant de parcourir l'arborescence de la solution tout en visualisant les données distribuées par cette dernière. Son but est de supporter le traitement de documents HTML, PNG, JPEG, GIF, etc. Il doit supporter les contenus interactifs manipulés par JavaScript.

4.2.2 - La librairie Administration

Ce grand composant est une librairie qui concentre les scripts exécutés lors des processus d'administration de données, autour de l'utilisateur et du produit. Elle regroupe les principales tâches utiles lors de la gestion centralisée des points de ventes.

4.2.2.1 - Gestion Utilisateur

Ce composant, ensemble de scripts ***PYTHON*** interne à la solution, délivre tous les services de manipulation centrés autour de l'utilisateur, employé ou client.

4.2.2.2 - Gestion Produit

Ce composant, ensemble de scripts ***PYTHON*** interne à la solution, délivre tous les services de manipulation centrés autour du produit, la pizza.

4.2.3 - La librairie Authentification

Ce grand composant est une librairie englobant les scripts et modules API, exécutés lors des processus d'authentification d'un utilisateur. Elle regroupe les principales tâches d'identification et vérification, utiles pour la sécurisation des opérations.

4.2.3.1 - Authentification interne

Ce composant, ensemble de scripts ***PYTHON*** interne à la solution, délivre tous les services d'identification et de vérification, portées sur un utilisateur.

4.2.3.2 - Authentification externe GOOGLE Sign-in

Ce composant, ***API*** externe à la solution, délivre tous les services d'identification et de vérification, portées sur un utilisateur.

4.2.3.3 - Authentification externe FACEBOOK Login

Ce composant, ***API*** externe à la solution, délivre tous les services d'identification et de vérification, portées sur un utilisateur.

4.2.4 - La librairie Boutique

Ce grand composant est une librairie qui concentre les scripts exécutés lors des processus de ventes et de services avant-après ventes.

4.2.4.1 - Gestion commande

Ce composant, ensemble de scripts ***PYTHON*** interne à la solution, délivre tous les services de gestion et de suivis, centrés autour d'une commande.

4.2.4.2 - Gestion transactions bancaires STRIPE

Ce composant, ***API*** externe à la solution, délivre tous les services de vérification et d'autorisation bancaire, portées sur une transaction.

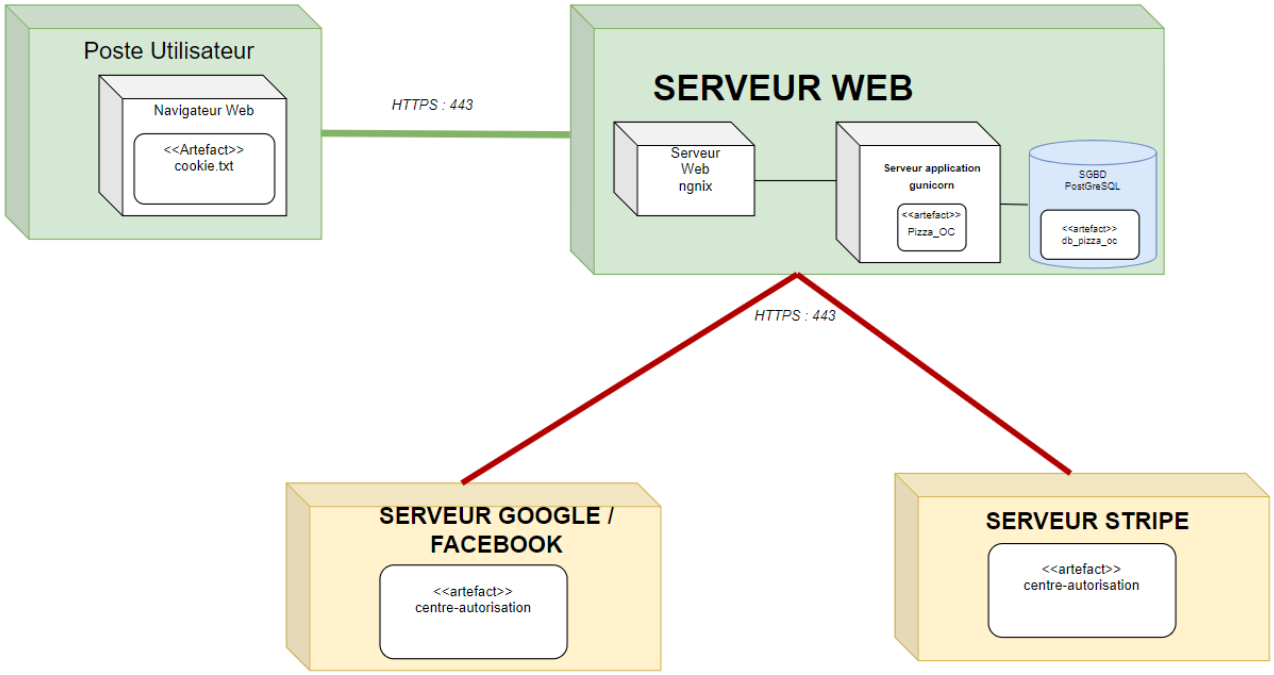
4.2.5 - SGBD

Ce composant, interne à la solution, permet d'offrir les services de base de données ***PostGreSQL***, essentiels au stockage et requêtage des données vitales du système.

5 - Architecture de Deploiement

5.1 - Le diagramme de déploiement

Afin d'obtenir une vision globale du projet, le diagramme de déploiement matérialise les acteurs qui exécuteront les composants de la solution. Voir ci-dessous :



Le tableau suivant reprend les éléments du diagramme de déploiement :

Nœud	*Contenu*	*Déploiement*
Poste Utilisateur	Navigateur Web	Cookie.txt
Serveur Web	Serveur Web ngnix	Application
	Serveur application gunicorn	PIZZA_OC
	SGBD postgresql	db_PIZZA_OC
Serveur Google / Facebook	Application	centre autorisation
Serveur Stripe	Application	centre autorisation

Ci-dessous se trouve les détails techniques des éléments, constituant la pile logicielle et entités physiques, essentiels au respect du déploiement de la solution.

5.1.1 - Système d'exploitation

Le système d'exploitation est de type ***Linux Ubuntu Version 20.0.4.1.***

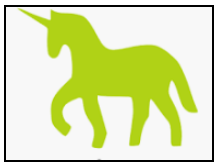
Ce système d'exploitation est l'interface centrale qui supporte les parties WEB et WSGI. Cet OS a été sélectionné pour obtenir une certaine homogénéité avec les composants applicatifs qu'ille aura à gérer.

5.1.2 - Serveur WEB

Le serveur WEB est de type ***NGNIX Version 1.18.0.***

Ce serveur traite les demandes du client WEB avant de les transmettre au WSGI. Il est aussi sollicité dans le sens inverse, ce qui lui confère un rôle de passerelle entre les terminaux WEB et le serveur d'applications. NGNIX possède une bonne relation avec GUNICORN en termes de déploiement et de maintenance.

5.1.3 - Serveur d'applications



Le serveur d'application est de type ***GUNICORN Version 20.0.4***

Le WSGI assure la mise en forme du flux HTTP, provenant du serveur WEB, en un flux adapté à la solution applicative. En plus de sa fonction de passerelle, ce serveur fourni des services standards pour améliorer le développement, le fonctionnement et la maintenance de la solution. Il améliore également la portabilité de la solution applicative

5.1.4 - Langage de programmation

Le Framework de développement est de type ***DJANGO version 3.1.2.***

Ce Framework python permet de garantir une solide chaine de cohérence entre le langage de programmation PYTHON et le WSGI. Il délivre des outils performants, permettant une rapide réalisation de la solution applicative, ainsi qu'un bon comportement durant le fonctionnement de cette dernière.

Le langage de programmation (script) est de type ***PYHTON version 3.8.***

PYTHON est le principal langage de programmation de la solution applicative. Il occupe la majorité des parties du FRONT-OFFICE et du BACK-OFFICE. Sa modularité permet une bonne gestion des fonctionnalités de la solution.

Le Framework JS est de type ***VueJs Version 2.6 (macross).7***

Ce Framework JavaScript a été sélectionné pour sa simplicité d'usage. Dans la solution applicative, Il est configuré pour apporter le minimum de fonctions nécessaires vis-à-vis de JavaScript.

Le langage de programmation web est de type ***JavaScript Version ES6.***

Ce langage est principalement présent pour structurer et styliser le contenu visuel et dynamique de la solution. Sa présence dans la solution reste donce moyenne voire faible.

Le langage de programmation web (structure) est de type ***HTML Version 5.***
Le langage de programmation web (style) est de type ***CSS Version 3.***

Ce couple de langage est principalement présent pour structurer et styliser le contenu visuel et statique de la solution.

5.1.5 - APIs

L'interface de programmation applicative dédiée aux transactions en ligne est de type ***STRIPE version 2020-08-27.***

L'interface de programmation applicative dédiée à l'authentification externe est de type ***GOOGLE SIGN-IN (OAuth 2.0).***

L'interface de programmation applicative dédiée à l'authentification externe est de type ***FACEBOOK LOGIN Version 2018-07-02.***

5.1.6 - Base de données

La base de données est de type ***POSTGRESQL Version 13.***

6 - GLOSSAIRE

API	Sigle anglais signifiant « Application Programming Interface », interface de programmation applicative en français. C'est une librairie de classes, fonctions et méthodes qui proposent un service WEB particulier.
Classe / Super-Classe	Classe : Terme Français utilisé en programmation orienté objet. Elle désigne un ensemble d'attributs et de méthodes permettant de concevoir et gérer une instance. Super-Classe : Terme français utilisé en programmation orienté objet. C'est une classe standard qui possède de plus, des classes héritées. On peut parler de classes filles ou classes enfants.
ID	Diminutif anglais de « IDentifiant ».
Impact Mapping	Méthode de planification permettant d'identifier les relations dynamiques importantes d'un projet. Elle permet d'adapter un plan d'action efficace en ayant une visibilité claire sur le périmètre fonctionnel du projet.
MPD	Sigle français signifiant « Modèle Physique de Données ». C'est un schéma qui reprend la structure d'une base de données.
OS	Sigle anglais signifiant « Operating System », Système d'exploitation en français.
PDV	Sigle français signifiant « Point de Vente ». Il est utilisé pour caractériser la partie physique d'une boutique.
Quantité Unitaire	Dans ce projet, ce terme désigne la quantité d'ingrédients requise pour considérer cet ensemble comme étant un ingrédient. Cela permet d'améliorer la gestion entre les ingrédients en stock. (Par exemple, Farine est considéré comme un ingrédient lorsque sa quantité est de 500g)
SGBD	Sigle français signifiant « Système de Gestion de Base de Données ».
Système Métrique	Terme français utilisé pour regrouper les unités de mesures standard. On peut aussi parler de « système international d'unités.
WSGI	Sigle anglais signifiant « Web Server Gateway Interface ». C'est un type de serveur, voire une spécification qui définit une interface entre des serveurs et des applications web pour le langage Python.