

OC PIZZA

Système de gestion de restaurant

DOCUMENTATION TECHNIQUE

Version 2

1. Version du document	4
2. Introduction	5
2.1. Objet du document	5
2.2. Récapitulatif - Cas d'utilisation (spécification fonctionnelles).	5
3. Domaine fonctionnel	6
3.1. Référentiel	6
3.1.1. Règles de gestion	6
Utilisateur - Rôle	6
Utilisateur - Adresse	6
Utilisateur - Commande	6
Commande - Statut	7
Commande - Adresse	7
Commande - Restaurant	7
Commande - Produit	7
Produit - Catégorie	7
Produit - Ingrédient	7
Ingrédient - Unité	7
Restaurant- Ingrédient	7
Restaurant- Adresse	7
4. Architecture technique	8
4.1 Application web	8
4.1.1 Composant Restaurant en ligne (sous-système)	9
4.1.1.1. Composant Menu	9
4.1.1.2. Composant Panier	9
4.1.1.3. Composant Authentification	10
4.1.2 Composant Administration (sous-système)	11
4.1.2.1. Composant Restaurant	11
4.1.2.2. Produit & recette	11
4.1.2.3. Ingrédient	12
4.1.2.4. Stock	12
4.1.2.5. Utilisateur	12
4.1.3 Composant Vente (sous-système)	13
4.1.3.1. Commande	13
4.1.3.2. Paiement	13
4.1.4 Composant Centre analyse(sous-système)	14
4.1.3.1. Statistique	14
5. Architecture de déploiement	15
5.1. Serveur dédié	15
5.1.1. Description:	15

5.1.2.Caractéristiques:	15
5.2. Serveur web	16
5.2.1.Description:	16
5.2.2.Caractéristiques:	16
5.3. Système de Gestion de base de donnée	16
5.3.1.Description:	16
5.3.2.Caractéristiques:	16
6. Glossaire	17

1. Version du document

Date	Nom	Description	V	Auteur
06/10/2020	<i>OC Pizza - Système de gestion de restaurant - Documentation technique</i>	1ère version.	1	Thomas PIERGIOVANNI
15/10/2020	<i>OC Pizza - Système de gestion de restaurant - Documentation technique</i>	Modification dans <i>Section 3.1</i> suite à un retour soutenance.	2	Thomas PIERGIOVANNI

2. Introduction

2.1. Objet du document

Ce document a pour objet la description de la conception technique de l'application de gestion de restaurant à développer pour OC Pizza.

Il décrit le domaine fonctionnel ainsi que l'architecture technique et de déploiement de l'application.

2.2. Récapitulatif - Cas d'utilisation (spécification fonctionnelles).

Package Authentification:

- s'inscrire
- se connecter.
- récupérer identifiants

Package Admin:

- gérer compte
- gérer restaurant.
- gérer produit et recettes
- gérer stock (responsable)
- gérer ingrédient
- Consulter les statistiques

Package Commande:

- choisir pizzeria
- constituer une commande
- constituer un panier
- compléter information commande
- valider commande
- consulter commande
- modifier commande
- supprimer commande
- modifier état commande
- consulter les produits et recettes
- payer à la remise
- payer en ligne

3. Domaine fonctionnel

3.1. Référentiel

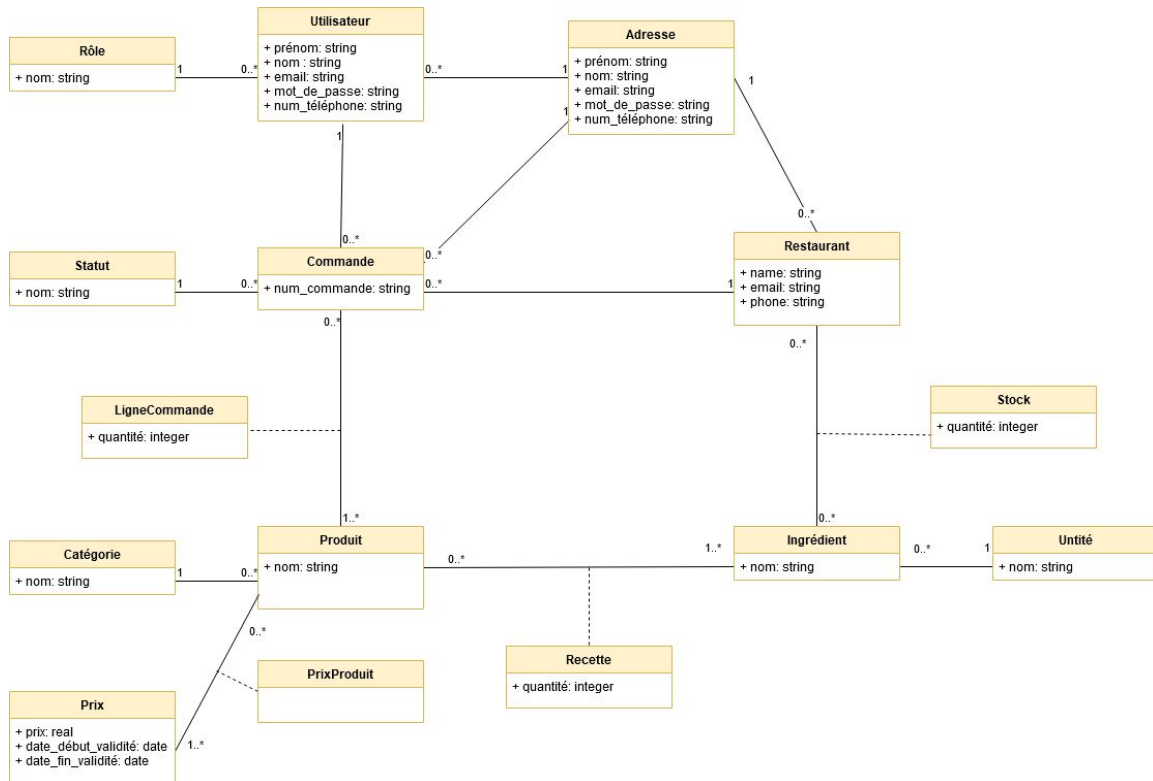


Diagramme de classe - Système de gestion de restaurant

3.1.1. Règles de gestion

Utilisateur - Rôle

- Un *Utilisateur* ne peut avoir qu'un seul *Rôle*.
- Un *Rôle* peut être attribué à zéro ou à plusieurs *Utilisateurs*.

Utilisateur - Adresse

- Un *Utilisateur* ne peut avoir qu'une seule *Adresse*.
- Une *Adresse* peut appartenir à zéro ou à plusieurs *Utilisateurs*.

Utilisateur - Commande

- Un *Utilisateur* peut avoir zéro ou plusieurs *Commandes*.
- Une *Commande* ne peut appartenir qu'à un *Utilisateur*.

Commande - Statut

- Une *Commande* ne peut avoir qu'un seul *Statut*.
- Un *Statut* peut être attribué à zéro ou plusieurs *Commande*.

Commande - Adresse

- Une *Commande* ne peut avoir qu'une seule *Adresse*.
- Une *Adresse* peut appartenir à zéro ou à plusieurs *Commandes*.

Commande - Restaurant

- Une *Commande* ne peut appartenir qu'à une seule *Restaurant*.
- Une *Restaurant* peut avoir zéro ou plusieurs *Commandes*.

Commande - Produit

- Une *Commande* peut avoir un ou plusieurs *Produits*.
- Un *Produit* peut composer zéro ou plusieurs *Commandes*.

Produit - Catégorie

- Un *Produit* ne peut avoir qu'une seule *Catégorie*.
- Une *Catégorie* peut être attribuée à zéro ou plusieurs *Produits*.

Produit - Prix

- Un *Produit* peut avoir un ou plusieurs *Prix*.
- Un *Prix* peut être attribué à zéro ou plusieurs *Produits*.

Produit - Ingrédient

- Un *Produit* peut être composé par un ou plusieurs *Ingrédients*.
- Un *Ingrédient* peut composer zéro ou plusieurs *Produits*.

Ingrédient - Unité

- Un *Ingrédient* ne peut avoir qu'une seule *Unité*.
- Une *Unité* peut être attribuée à zéro ou plusieurs *Ingrédients*.

Restaurant- Ingrédient

- Un *Restaurant* peut stocker zéro ou plusieurs *Ingrédients*.
- Un *Ingrédient* peut être stocké par zéro ou plusieurs *Restaurants*.

Restaurant- Adresse

- Une *Restaurant* ne peut avoir qu'une seule *Adresse*.
- Une *Adresse* peut appartenir à zéro ou à plusieurs *Restaurants*.

4. Architecture technique

4.1 Application web

La pile logicielle est la suivante :

- Serveur dédié: Debian 10.6
- Serveur web: Nginx 1.19.2.
- Application: Python 3.8
- Système de Gestion de Base de Données : MySQL 8.0.

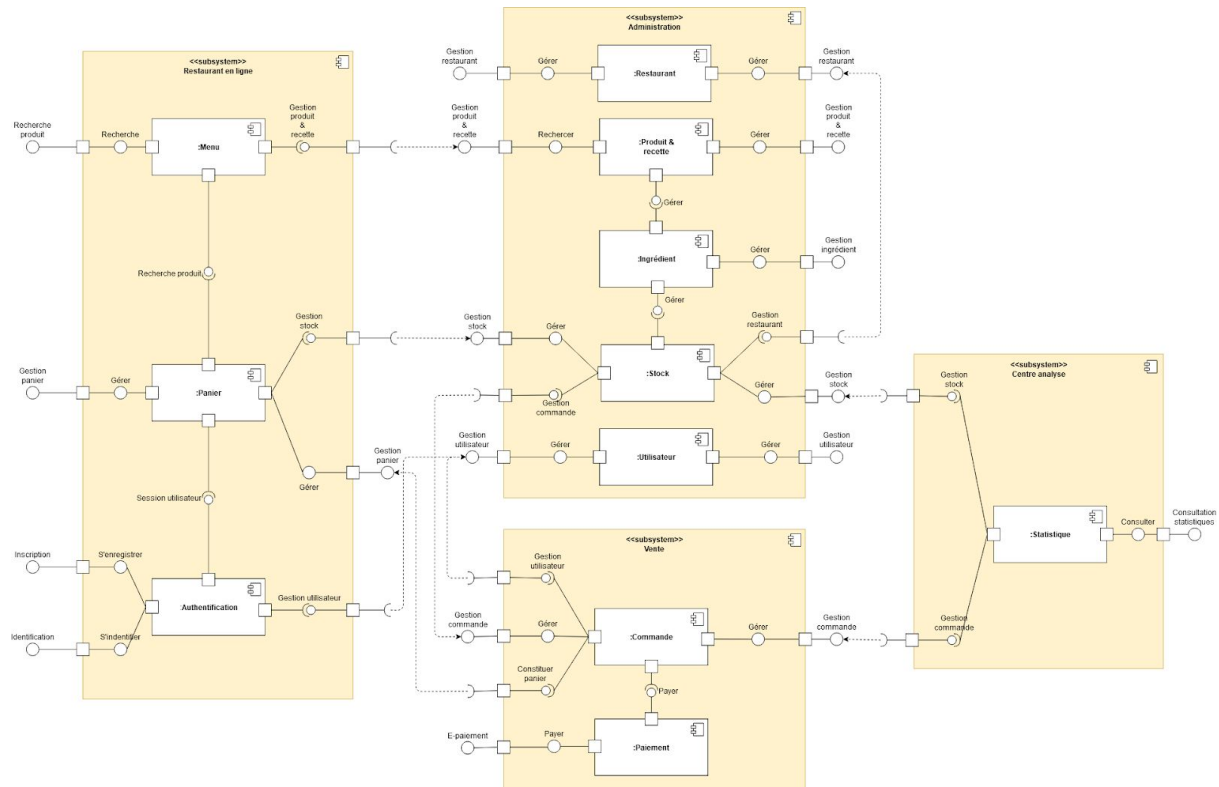


Diagramme de composants - Système de gestion de restaurant

4.1.1 Composant *Restaurant en ligne* (sous-système)

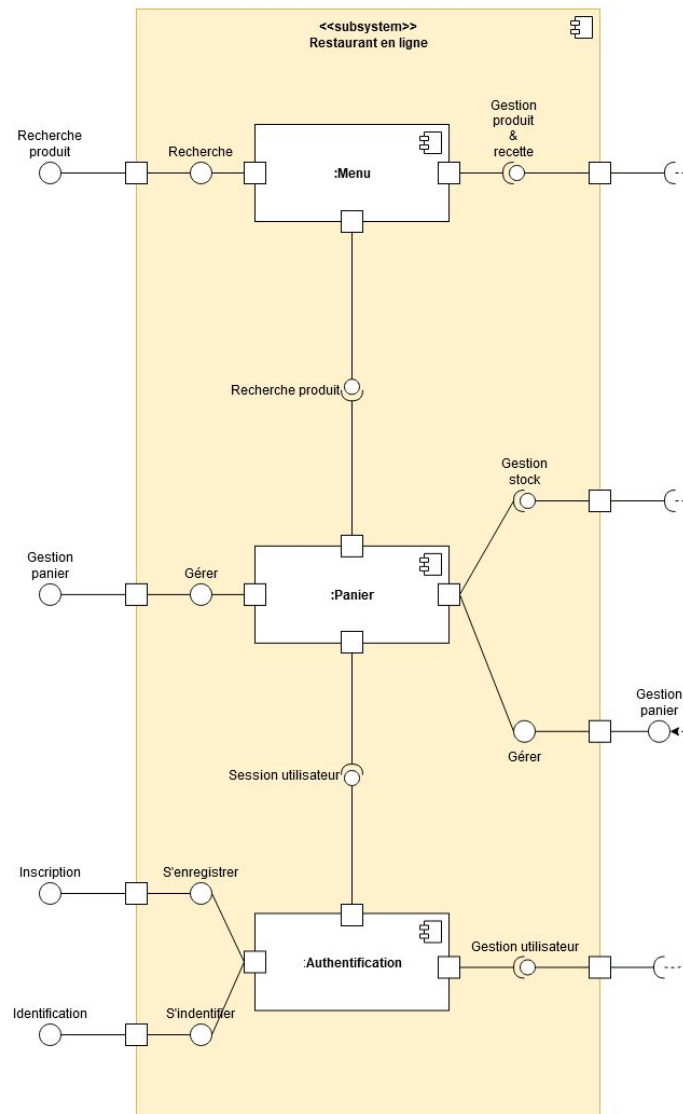


Diagramme de composants - Système de gestion de restaurant, sous système “Restaurant en ligne”

4.1.1.1. Composant *Menu*

- *Menu* fournit l'interface *Recherche Produit* afin de permettre à l'utilisateur d'effectuer une recherche dans le menu.
- *Menu* requiert l'interface *Gestion produit & recette* afin de disposer des informations des *Produits* (ex: Pizza Margherita, Regina, Coca-Cola, etc.) mis en vente.

4.1.1.2. Composant *Panier*

- *Panier* fournit l'interface *Gestion panier*.

- afin de permettre à l'utilisateur de gérer son panier d'achat.
- afin de permettre à *Commande* de disposer des informations du panier.
- *Panier* requiert l'interface *Gestion stock* afin de disposer des informations de stock disponible.
- *Panier* requiert, si l'utilisateur le souhaite, l'interface *Identification* afin de disposer des informations de l'utilisateur.

4.1.1.3. Composant *Authentication*

- *Authentication* fournit l'interface *Inscription* afin de permettre à l'utilisateur (acteur visiteur*) de s'inscrire.
- *Authentication* fournit l'interface *Identification*:
 - afin de permettre à l'utilisateur (acteur client·e*, responsable*, pizzaiolo*, livreur·se*) de s'identifier.
 - afin de permettre à *Panier*, si l'utilisateur le souhaite, de disposer des informations de l'utilisateur.
- *Authentication* requiert l'interface *Gestion utilisateur* afin de disposer des informations utilisateurs disponibles.

4.1.2 Composant *Administration* (sous-système)

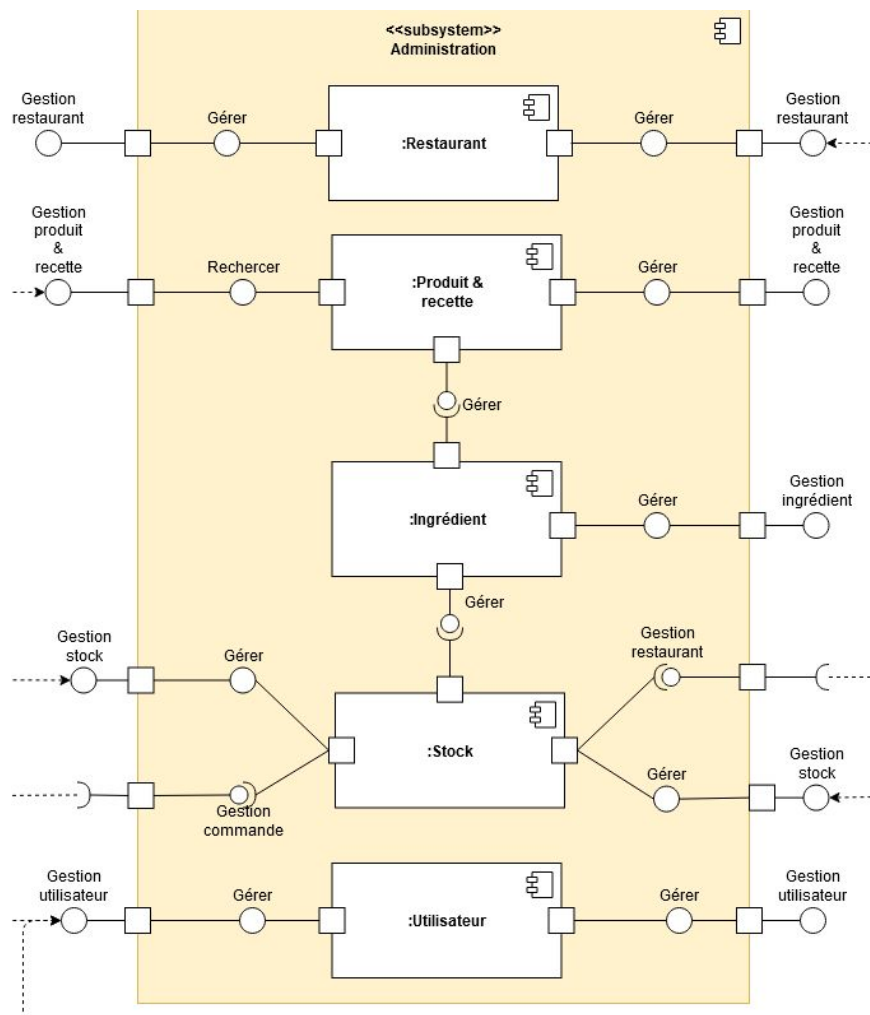


Diagramme de composants - Système de gestion de restaurant, sous système "Administration"

4.1.2.1. Composant *Restaurant*

- *Restaurant* fournit l'interface *Gestion restaurant*:
 - afin de permettre à l'utilisateur de gérer le restaurant (p.ex. ajouter un nouveau restaurant, modifier le nom d'un restaurant existant, etc.).
 - afin de permettre à *Stock* de se référer à restaurant.

4.1.2.2. Produit & recette

- *Produit et recette* fournit l'interface *Gestion produit & recette*:
 - afin de permettre à l'utilisateur de gérer ses produits et ses recettes (p.ex. ajouter une pizza à la liste de produits à vendre, modifier la recette d'un dessert).
 - afin de permettre à *Menu* de disposer des informations des *Produits* mis en vente.

- afin de permettre à *Ingrédient* d'être ajusté en fonction des recettes.

4.1.2.3. Ingrédient

- *Ingrédient* fournit l'interface *Gestion ingrédient*:
 - afin de permettre à l'utilisateur de gérer ses ingrédients.
 - afin de permettre à *Stock* d'être ajusté en fonction des ingrédients.
- *Ingrédient* requiert l'interface *Gestion produit & recette* afin d'être ajusté en fonction des recettes.

4.1.2.4. Stock

- *Stock* fournit l'interface *Gestion stock*:
 - afin de permettre à l'utilisateur de gérer ses stocks.
 - afin de permettre à *Panier* de disposer des informations de stock disponible.
 - afin de permettre à *Statistique* de disposer des informations des stocks.
- *Stock* requiert l'interface *Gestion ingrédient* afin d'être ajusté en fonction des ingrédients.
- *Stock* requiert l'interface *Gestion restaurant* afin de se référer à restaurant.
- *Stock* requiert l'interface *Gestion commande* afin de s'ajuster en fonction des commandes.

4.1.2.5. Utilisateur

- *Stock* fournit l'interface *Gestion utilisateur*:
 - afin de permettre à l'utilisateur de gérer "son compte" si l'utilisateur est de type client* et "les comptes" si l'utilisateur est de type responsable*.
 - afin de permettre à *Authentification* de disposer des informations utilisateurs disponibles

4.1.3 Composant *Vente* (sous-système)

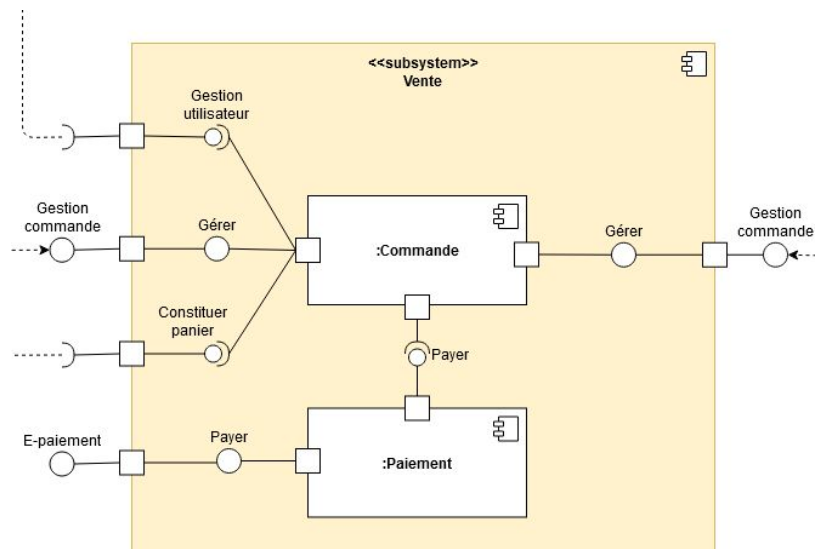


Diagramme de composants - Système de gestion de restaurant, sous système “Vente”

4.1.3.1. Commande

- **Commande** fournit l'interface *Gestion commande* :
 - afin de permettre à l'utilisateur de gérer la commande.
 - afin de permettre à *Stock* de s'ajuster en fonction des commandes.
 - afin de permettre à *Statistique* de disposer des informations des commandes.
- **Commande** requiert l'interface *Gestion utilisateur* afin de se référer à un utilisateur.
- **Commande** requiert l'interface *Gestion panier* afin de disposer des informations du panier.
- **Commande** requiert, si l'utilisateur le souhaite, l'interface *E-paiement* afin d'effectuer le paiement en ligne.

4.1.3.2. Paiement

- **Paiement** fournit l'interface *E-paiement* afin de permettre à **Commande**, si l'utilisateur le souhaite, d'effectuer le paiement en ligne.

4.1.4 Composant *Centre analyse(sous-système)*

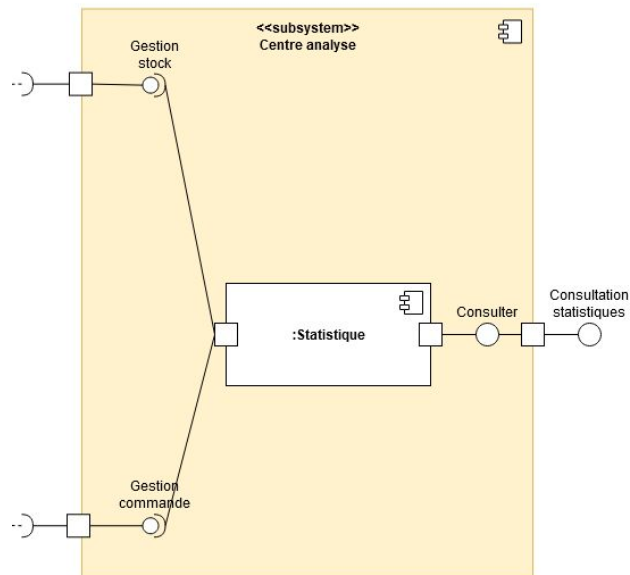


Diagramme de composants - Système de gestion de restaurant, sous système "Centre analyse"

4.1.3.1. Statistique

- *Statistique* fournit l'interface *Consultation statistique* afin de permettre à l'utilisateur de consulter les statistiques.
- *Statistique* requiert l'interface *Gestion stock* afin de disposer des informations des stocks.
- *Statistique* requiert l'interface *Gestion commande* afin de disposer des informations des commande.

5. Architecture de déploiement

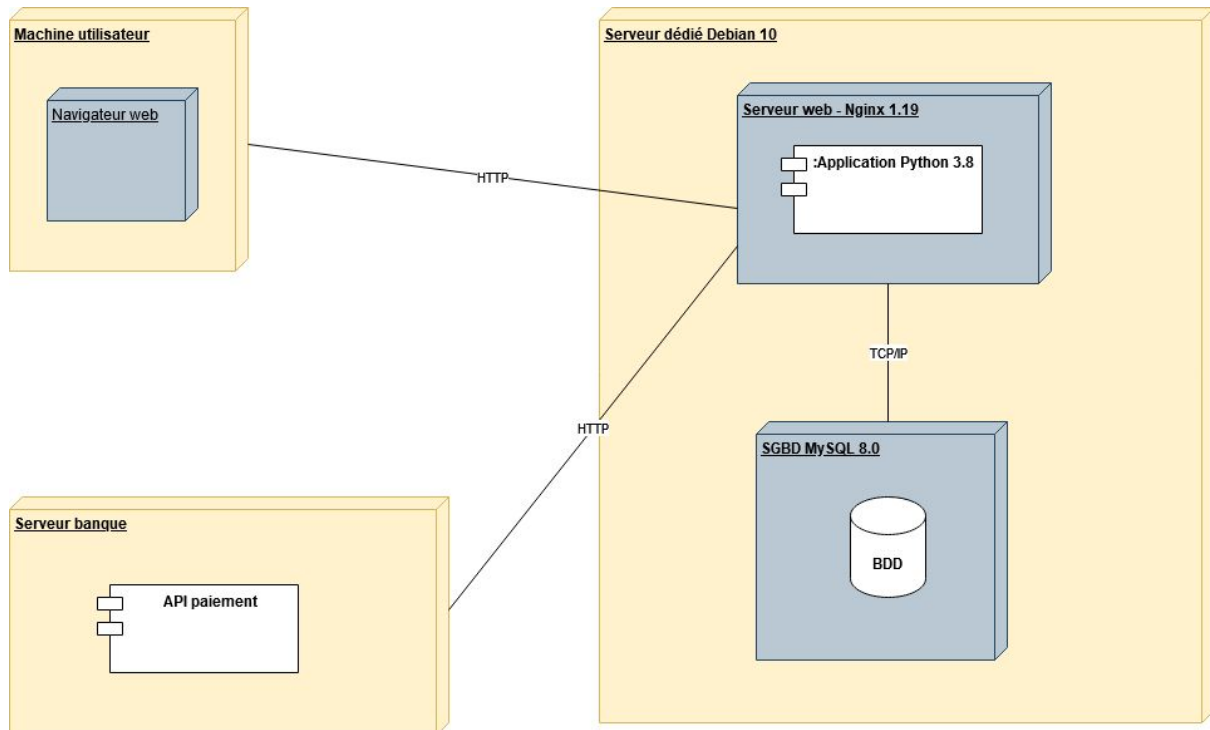


Diagramme de déploiement - Système de gestion de restaurant

- Le navigateur web de la machine utilisateur communique avec l'application se trouvant sur un serveur web via un protocole HTTP.
- L'application située sur le serveur web communique avec la base de donnée MySQL se trouvant sur un système de gestion de base de donnée via un protocole TCP/IP.
- L'application située sur le serveur web communique avec l'API de paiement se trouvant sur le serveur de la banque via un protocole HTTP.

5.1. Serveur dédié

5.1.1. Description:

Serveur mis à disposition du seul client uniquement.

5.1.2. Caractéristiques:

- Debian 10.6:
 - Système d'exploitation libre (noyau Linux)
 - Fourni avec 59000 paquets (logiciels) libre.

5.2. Serveur web

5.2.1.Description:

Programme qui fournit les pages des pages web demandé à l'utilisateur.

5.2.2.Caractéristiques:

- Nginx 1. 19.2:
 - Serveur web libre
 - Systèmes d'exploitation supportés: Mac OS X, Windows

5.3. Système de Gestion de base de donnée

5.3.1.Description:

Le serveur de base de donnée sert à stocker et gérer les données du Système de gestion de restaurant dans une base de données.

5.3.2.Caractéristiques:

- MySQL 8.0.:
 - Serveur de base de donnée relationnelles SQL.
 - Licence publique générale GNU (GPL) ou propriétaire selon utilisation.
 - Systèmes d'exploitation supportés: Linux, Mac OS X, Windows (entre autres).

6. Glossaire

Mot	Définition	Référence
Visiteur-se	Un-e client-e ne disposant pas de compte dans la chaîne de pizzerias.	Voire <i>Documentation Fonctionnelle</i> , section 2.1. <i>Diagramme de contexte</i>
Client-e	Un-e client-e disposant d'un compte dans la chaîne de pizzerias.	idem
Responsable	Responsable des pizzerias	idem
Pizzaiolo	Pizzaiolo des pizzerias	idem
Livreur-se	Livreur-se des pizzerias	idem