



# **Projet OC Pizza**

Dossier de conception fonctionnelle  
(Solution technique)

LINOCENT Timothée  
*IT Consulting & Development*



## TABLE DES MATIERES

<b>1 - Introduction .....</b>	<b>3</b>
1.1 -Objet du document .....	3
1.2 -Besoin du client.....	3
<b>2 - Le domaine fonctionnel .....</b>	<b>4</b>
2.1 -Diagramme de classe .....	4
2.2 -Modèle physique de données .....	5
2.3 -Description .....	5
<b>3 - Architecture.....</b>	<b>14</b>
3.1 -La base de données.....	14
3.2 -Les composants internes et le déploiement technique .....	15
<b>4 - Glossaire .....</b>	<b>16</b>

# 1 - INTRODUCTION

## 1.1 - Objet du document

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

L'objectif du document est de présenter la solution technique comprenant les classes associées à la solution trouvée, une ébauche d'architecture, un aperçu de la base de données.

## 1.2 - Besoin du client

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 d'ici 6 mois.

Le système informatique actuel ne répond 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.

Enjeux et objectifs :

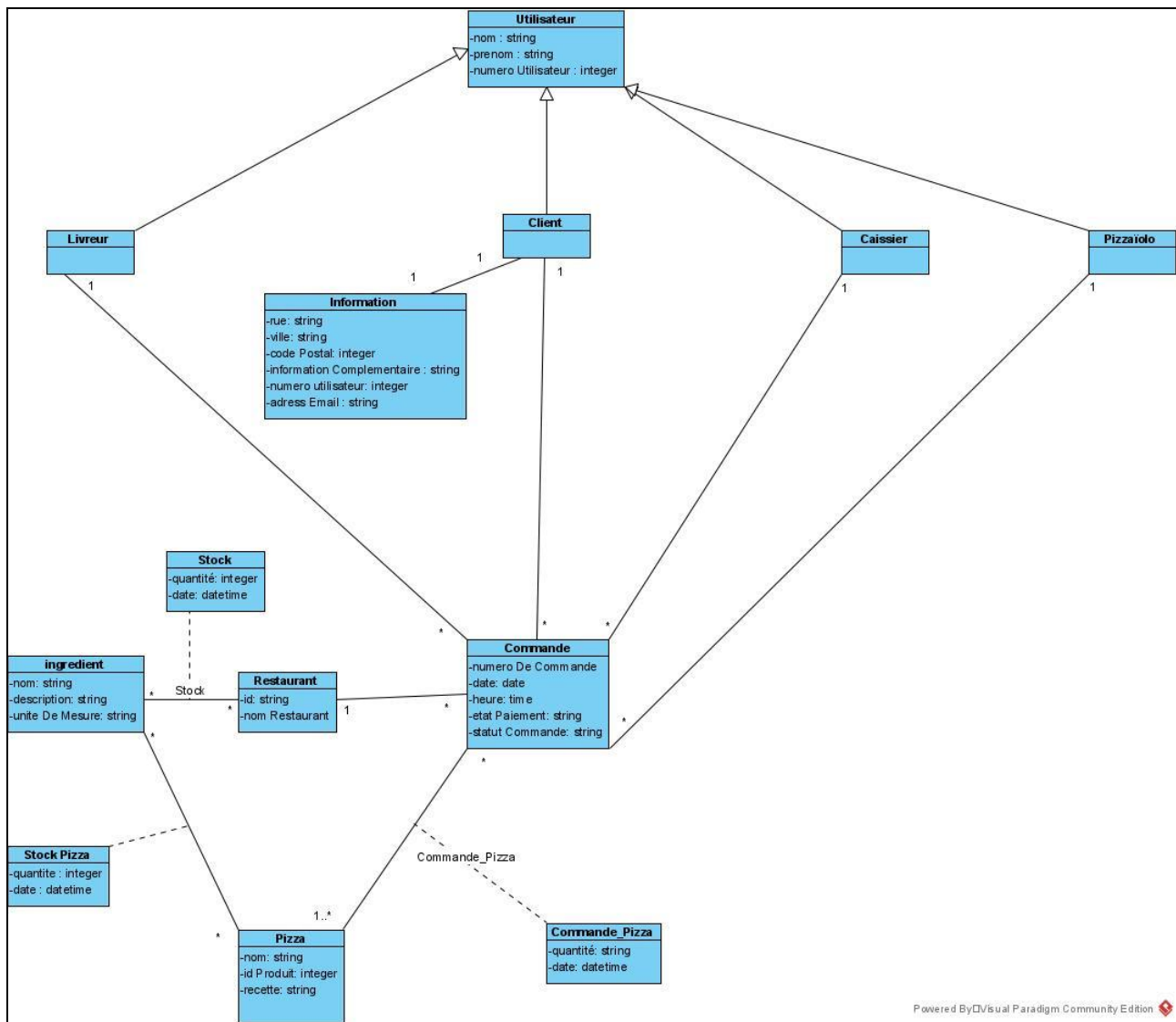
Le client souhaite :

- Être plus efficace dans la gestion des commandes, de leur réception à leur livraison en passant par leur préparation ;
- Suivre en temps réel le stock d'ingrédients restant pour savoir quelles pizzas peuvent encore être réalisées ;
- 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.
- Proposer un aide-mémoire aux pizzaiolos indiquant la recette de chaque pizza.

Le client souhaite obtenir le système informatique sous 6 mois, pour l'ouverture des trois nouvelles pizzerias.

## 2 - LE DOMAINE FONCTIONNEL

### 2.1 - Diagramme de classe

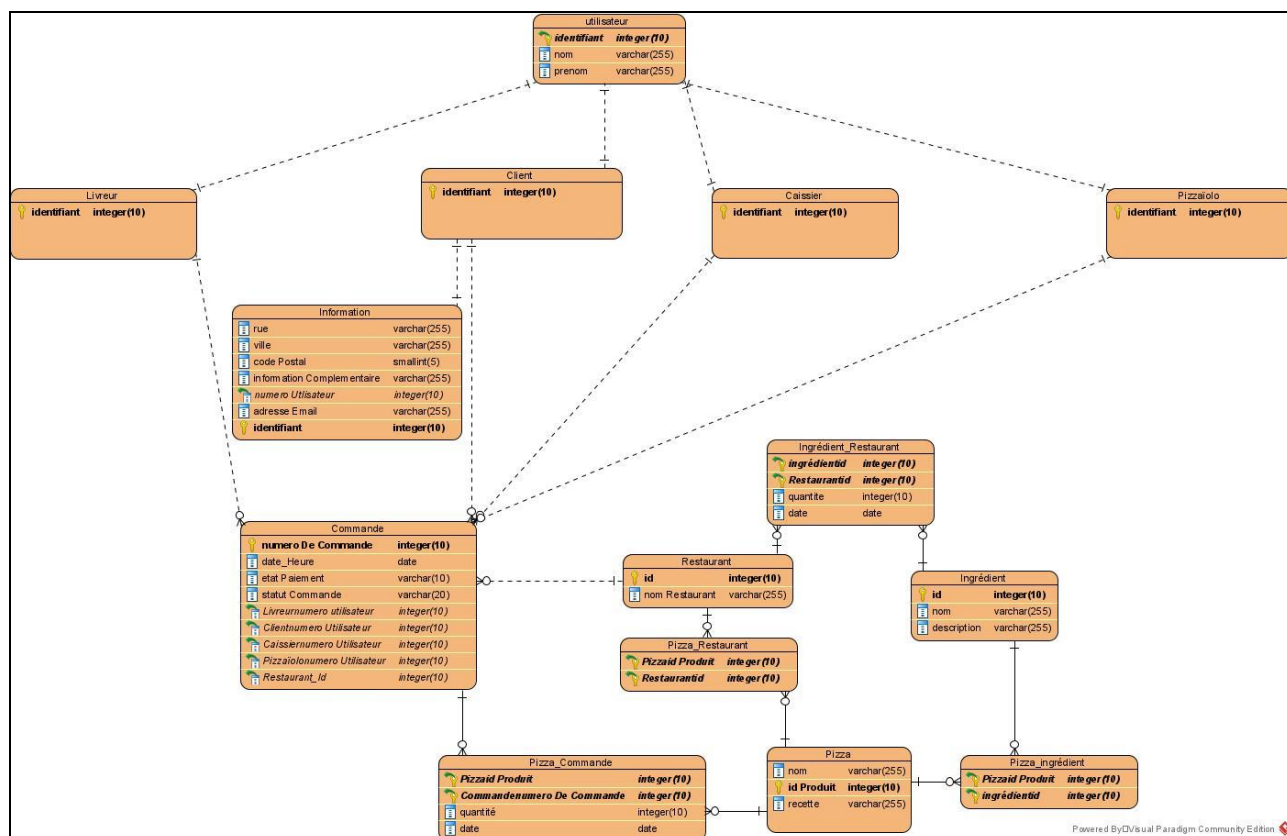


Le diagramme de classe UML ci-dessus est une représentation des informations qui vont être manipulées. Ce qui nous permet d'avoir une vision globale du système et de l'organisation des données.

Nous apercevons également les différentes interactions qu'il y aura entre chaque classe, leur cardinalité et leur filiation.

## 2.2 - Modèle physique de données

A partir de notre diagramme de classe, nous pouvons réaliser le diagramme du modèle physique de données qui modélise la base de données.



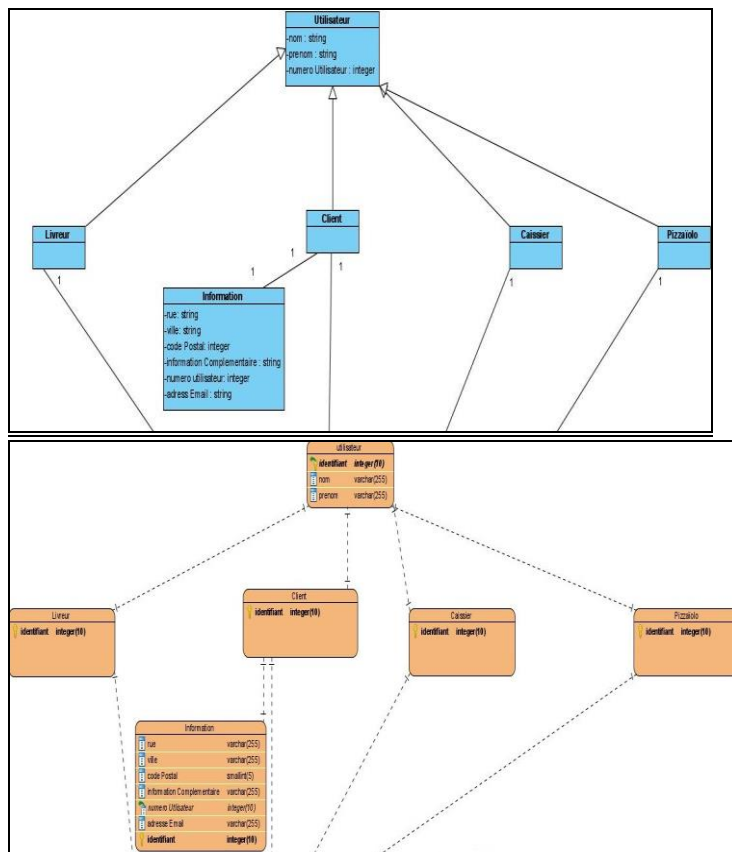
Ce diagramme nous présente visuellement ce à quoi ressemblera notre base de données, les relations entre les tables et le contenu des tables, avec le type de chaque colonne.

Nous détaillerons tous ceci juste après.

## 2.3 - Description

Nous allons à présent décrire les classes et tables précédemment présenter. Dans la mesure où les deux diagrammes sont étroitement liés, nous présenterons la table et la classe qui lui est associée simultanément.

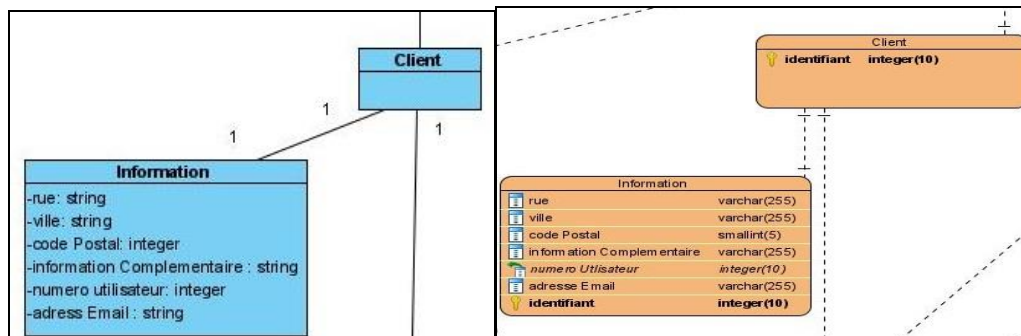
## Les classes/tables client, livreur, caissier et pizziolo.



Ces classes servent à identifier le statut de chaque utilisateur. Chaque utilisateur est identifié par un numéro utilisateur qui est unique à chacun.

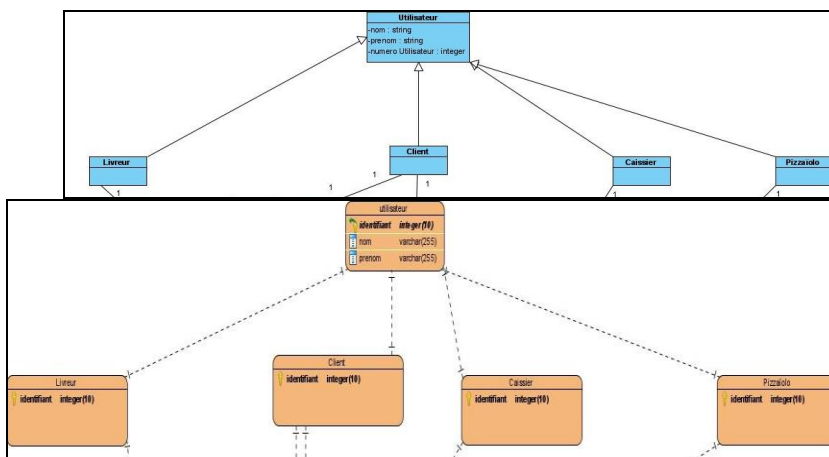
### **Relations**

Classe : client – information. Cardinalité de 1 à 1.



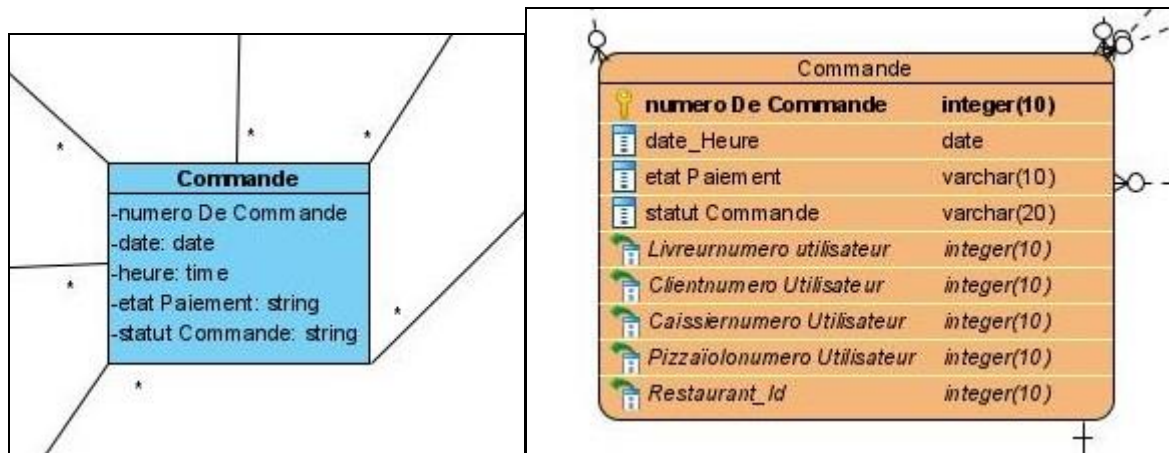
La classe information est reliée à la classe client. Grâce à la clé étrangère, chaque ligne de la table information correspondra à un unique client. Ce qui permettra d'avoir différent client avec un même nom ou prénom, ou encore avoir deux clients à la même adresse.

Classe : livreur/caissier/pizzaiolo – utilisateur. Cardinalité de 1 à 1.



La classe utilisateur décrit ces utilisateurs en stockant leurs nom et prénom. La clé étrangère permet de faire correspondre chaque ligne à un utilisateur unique. Ce qui permet d'avoir différent utilisateur avec un même nom ou prénom.

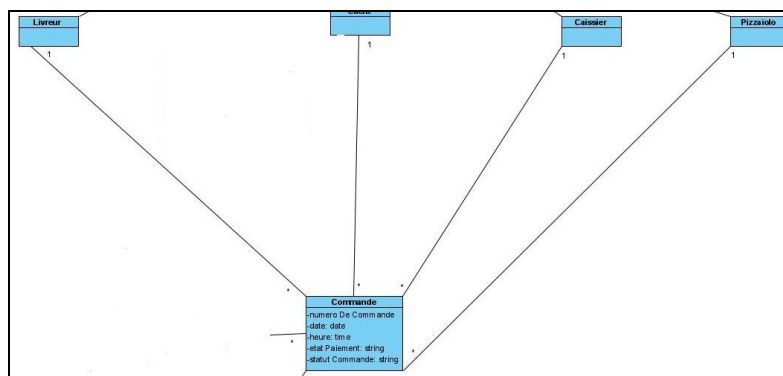
## La classe commande.



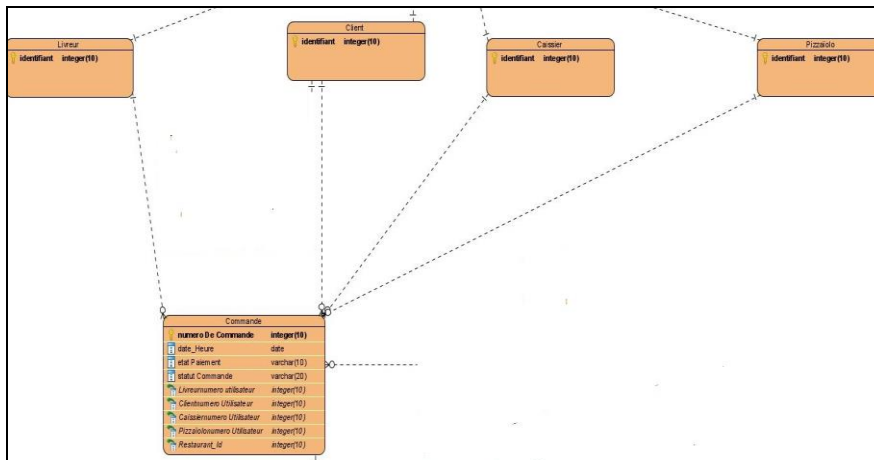
Cette classe décrit une commande dans son intégralité en répertoriant le numéro de commande, la date et l'heure de la commande, l'état du paiement de la commande (payé, impayé) et le statut de la commande (en préparation, en livraison, etc.).

## Relations

Classes : client/livreur/caissier/ Pizzaiolo – commande. Cardinalité 1 à 1, n.

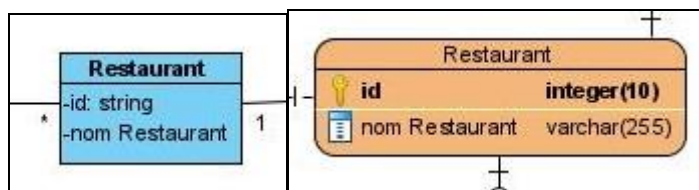






Ces relations permettent d'identifier chaque utilisateur ou client ayant passer une commande, et aux livreurs et pizzaiolo de consulter la commande et d'interagir avec.

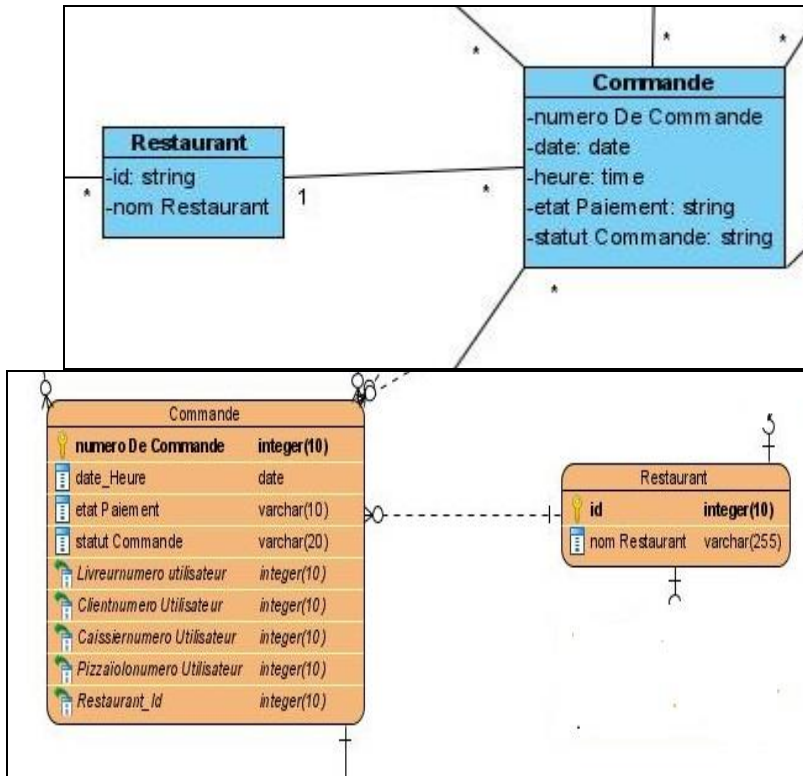
### La classe restaurant



La table restaurant contiendra les noms de chaque restaurant. Chaque restaurant aura un nom qui lui est propre pour éviter les confusions.

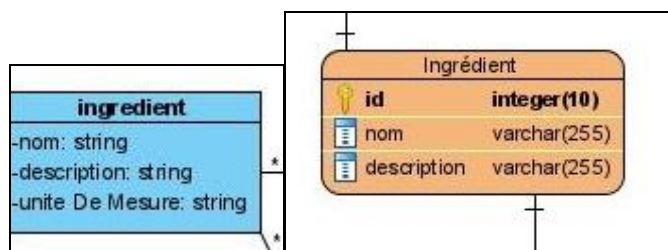
## Relations

Classe : restaurant – commande. Cardinalité 1 à n.



A chaque restaurant correspondra au moins une commande passée. Ce qui permettra de suivre le nombre de commande passer dans chaque restaurant.

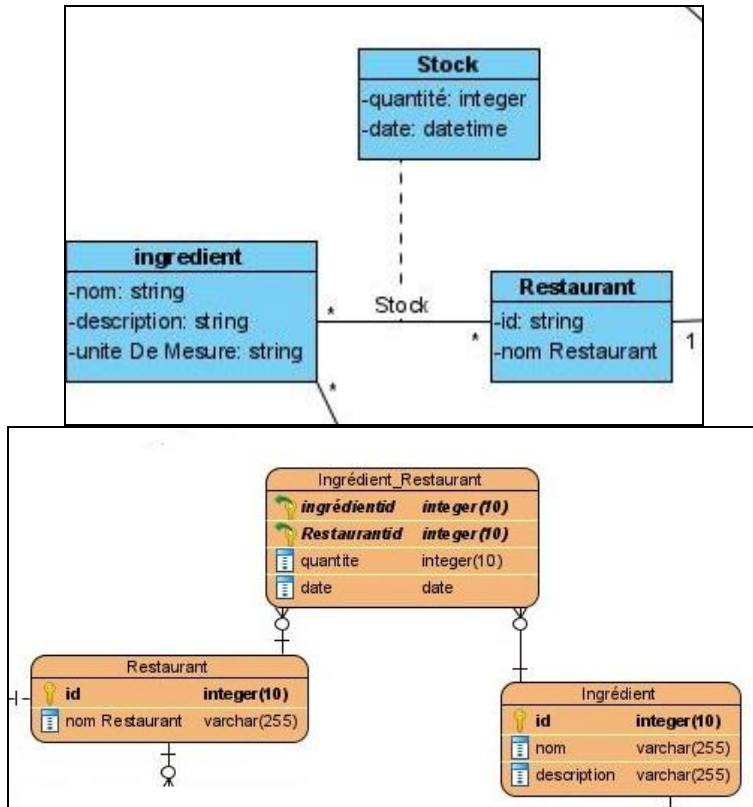
## La classe ingrédient



La classe ingrédient décrit l'ensemble des ingrédients nécessaires à la réalisation de l'ensemble des produits proposé à la vente.

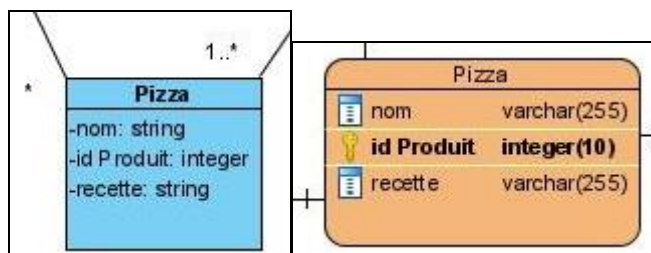
## Relation

Classe : ingrédient – stock – restaurant. Cardinalité 1,n à 1,n.



La classe stock permet de visualiser le stock d'ingrédient disponible pour chaque magasin.

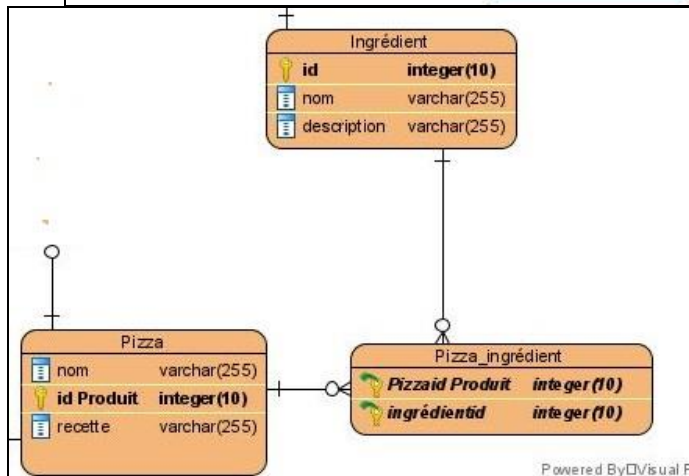
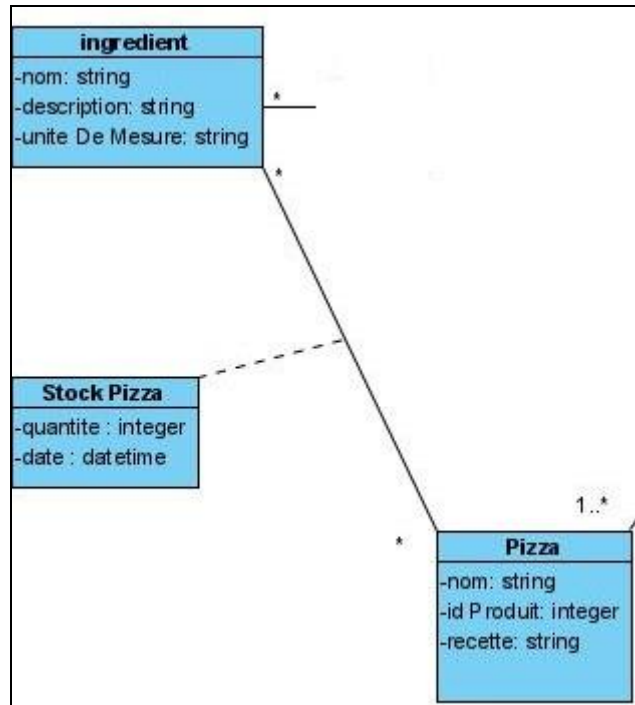
## La classe Pizza



Cette classe présente le nom de la pizza, les ingrédients qui la compose.

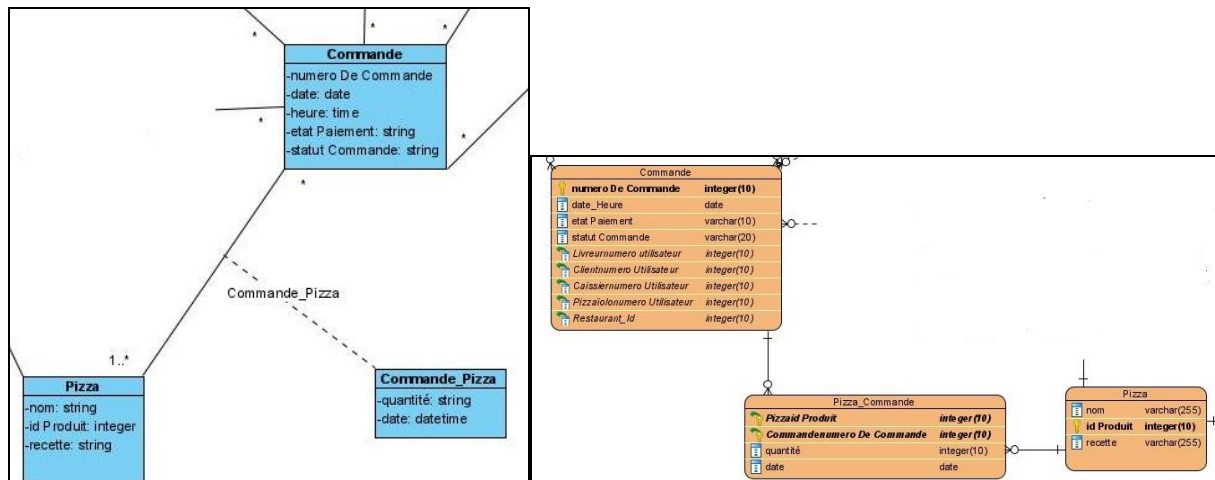
## Relation

Classe : ingrédient – stock pizza – Pizza. Cardinalité : 1,n à 1,n.



Cette relation permet de visualiser le nombre de pizza théorique disponible à la vente en fonction de la quantité d'ingrédient restant. La classe

Classe : pizza – commande pizza – commande. Cardinalité : 1,n à 1,n.

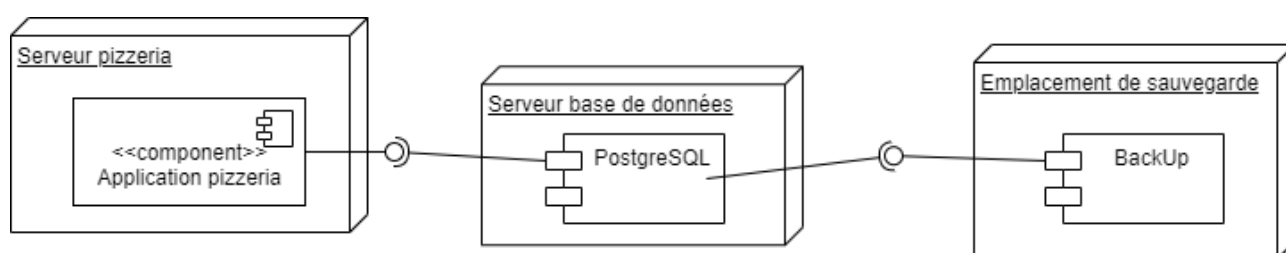


Cette relation permet de connaitre le nombre de pizza commander. La classe commande\_pizza contient les quantités commander et les dates.

## 3 - ARCHITECTURE

Les diagrammes suivants présenteront les composants nécessaires à l'application. Ils mettront en lumière les dépendances entre les composants internes et externes.

### 3.1 - La base de données

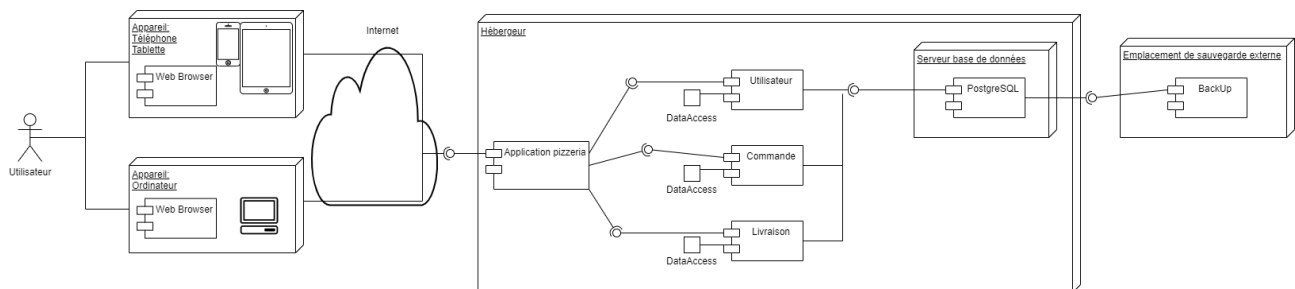


Nous utiliserons PostgreSQL comme base de données relationnel. Il y aura deux sauvegardes, une première sauvegarde sur la machine où sera héberger le serveur de l'application et la seconde sera sur une machine externe afin de protéger les données en cas de problème – comme ce fut le cas pour l'incendie chez OVH récemment. Un simple ordinateur ou disque dur suffira pour la BackUp.

Les sauvegardes seront quotidiennes et devront s'organiser en dehors des heures d'ouvertures des restaurants, donc probablement de nuit.

Si les restaurants font de la vente 24/24, nous commencerons par effectuer la sauvegarde au milieu de la nuit, vers 3h00, et changeront l'horaire afin de choisir la période la plus creuse pour gêner le moins possible les utilisateurs -clients comme employés.

### 3.2 - Les composants interne et le déploiement technique.



Sur ce schéma, nous pouvons voir les composants interne de notre future application. Les différentes fonctionnalités nécessaires à notre application. Nous voyons les différents supports nécessaires au fonctionnement de notre application.

De plus la plateforme sera la même que l'utilisateur soit un visiteur, client, employés ou administrateur. Par plateforme nous voulons dire, ici, serveur.

Nous pouvons observer qu'un utilisateur pourra utiliser notre application avec tout support, il s'agit d'une application REST. De ce fait, un utilisateur pourra utiliser un téléphone, une tablette ou un ordinateur. L'utilisateur aura également besoin d'un accès à un navigateur internet quel qu'il soit.

## 4 - GLOSSAIRE ET RÉFÉRENCES

<b>UML</b>	De l'anglais <i>Unified Modeling Language</i> , est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système.
<b>Clé étrangère</b>	Une clé étrangère identifie une colonne ou un ensemble de colonne d'une table comme référençant une colonne ou un ensemble de colonnes d'une autre table (la table référencée).