



DOSSIER DES SPECIFICATIONS TECHNIQUES



IT CONSULTING & DEVELOPMENT

SOMMAIRE

I - LE CONTEXTE

II – DOMAINE FONCTIONNEL

- 1 – Diagramme de classe
- 2 – Description des classes
 - a – Restaurant
 - b – User
 - c – Client
 - d – OCPizza
 - e – Contact
 - f – Ingredient
 - g – Stock
 - h - Recipe
 - i – Product
 - j – Order
 - k - Basket
 - l - Order_statusupdate

III – DIAGRAMMES

- 1 – Diagramme de classe
- 2 – Description des classes
 - a – Restaurant
 - b – User
 - c – Client
 - d – OCPizza
 - e – OCPizza_role
 - f – Contact
 - g – Ingredient
 - h - Ingredient_stock
 - i – Product
 - j – Product_recipe
 - k - Product_type
 - l – Product_price
 - m – Product_size
 - n – Order
 - o – Order_basket
 - p – Order_delivery
 - q – Order_payment
 - r – Order_status
 - s – Order_statusupdate

IV – DIAGRAMME DE COMPOSANTS

- 1 – Diagramme de composant général
- 2 – Description de composant API REST Google map
- 3 – Diagramme de composants système bancaire

V – DIAGRAMME DE DEPLOIEMENT

I – CONTEXTE

La société :

OC Pizza est une société spécialisée dans la vente de pizza en livraison et à emporter. Elle compte actuellement cinq points de vente et trois supplémentaires sont prévus.

L'objectif :

L'objectif est de mettre en place un système informatique pour l'ensemble des pizzerias du groupe

Fonctionnalités demandées :

- 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 :
 - 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

b – User



Cette classe possède les attributs communs à chaque utilisateurs du programme (à savoir les clients ou les employés d'OCPizza). Ses attributs sont :

User_firstname : le prénom de l'utilisateur

User_lastname : le nom de l'utilisateur

User_login : le login d'authentification de l'utilisateur

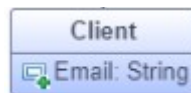
User_password : le mot de passe de l'utilisateur

Les classes **Client** et **OCPizza** lui sont associées par **héritage**, la complétant et permettant d'ajouter des attributs non-communs à ces deux classes.

La classe est associée à la classe **Order**.

Order : c'est une association **one-to-many**, un utilisateur pouvant interagir avec une ou plusieurs commande (que ce soit passer une commande, modifier le statut d'une commande etc.)

c – Client

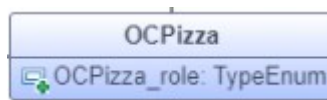


Elle ne possède qu'un attribut (**Email**) qui correspond à l'adresse mail du client.

Cette classe est associée par **héritage** à la classe **User**, qu'elle vient compléter. Elle est également associée avec la classe **Contact**.

Contact : c'est une association **one-to-many**, un client pouvant avoir une ou plusieurs adresses enregistrées (par exemple, si le client souhaite se faire livrer à une autre adresse que celle de son domicile).

d – OCPizza



Elle ne possède qu'un attribut (**Role**) qui correspond à l'affectation d'un employé OCPizza.

Cette classe est associée par **héritage** à la classe **User**, qu'elle vient compléter. Elle est également associée avec la classe **Restaurant** dont elle est un **composant**.

Restaurant : l'association a déjà été commentée précédemment.

e – Contact



Cette classe possède les attributs suivant :

City : nom de ville de l'entité

Address : adresse de l'entité

Level : Si appartement, étage de l'entité

Phone : Numéro de téléphone de l'entité

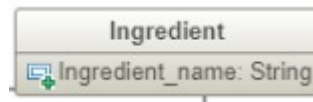
Comment : Commentaires de l'entité

La classe est associée aux classes **Restaurant**, **User** et **Order**.

Les deux premières associations ont été commentées précédemment.

Order : c'est une association **one-to-one**, dans le cas où la commande doit être livrée, celle-ci ne peut être livrée qu'à une seule adresse.

f – Ingredient



Elle ne possède qu'un attribut (**Ingredient_name**) qui correspond au nom de l'ingrédient.

La classe est associée aux classes **Restaurant** et **Product**.

Restaurant : l'association a déjà été commentée précédemment..

Product : c'est une association **many-to-many**, un (ou plusieurs) produit pouvant contenir un certain nombre d'ingrédient dans sa recette.

g – Stock



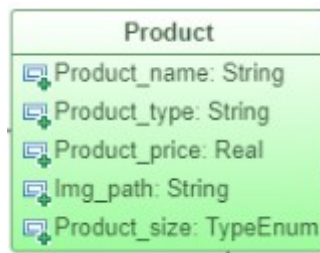
C'est une **classe d'association** qui permet d'ajouter l'attribut **Ingredient_quantity** à l'association entre les classes **Restaurant** et **Ingredient** (**many-to-many**). Cela permet de connaître la quantité d'un ingrédient dans le stock d'un restaurant.

h – Recipe



C'est une **classe d'association** entre les classes **Ingredient** et **Product** (**many-to-many**). Cela permet de connaître les ingrédients nécessaire à la préparation d'un ou plusieurs produits, en bref, la recette.

i – Product



Cette classe possède les attributs suivant :

Product_name : le nom du produit

Product_type : la catégorie du produit (pizza, boisson etc.)

Product_price : la valeur du produit

Product_size : la taille du produit

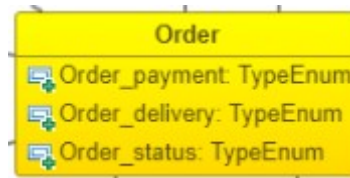
Image_path : un chemin vers une image/représentation du produit

La classe est associée aux classes **Ingredient** et **Order**.

Ingredient : l'association a déjà été commentée précédemment

Order : c'est une association **many-to-many**, un (ou plusieurs) produit pouvant être contenu dans une (ou plusieurs) commande.

j – Order



Cette classe possède les attributs suivant :

Order_payment : le mode de paiement de la commande

Order_delivery : le mode de livraison de la commande

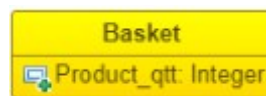
Order_status : le statut de la commande (en préparation, en livraison etc.)

La classe est associée aux classes **Restaurant**, **User**, **Contact**, **Product** et **Order_statusupdate**

Les quatre premières associations ont été commentées précédemment.

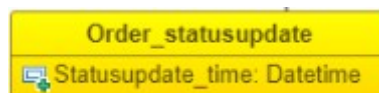
Order_statusupdate : c'est une association **one-to-many**, une commande ayant pu avoir une (ou plusieurs) mise à jour de statut.

k – Basket



C'est une **classe d'association** qui permet d'ajouter l'attribut **Product_qtt** à l'association entre les classes **Product** et **Order** (**many-to-many**). Cela permet de connaître la quantité d'un produit dans une commande.

l – Order_statusupdate



Elle ne possède qu'un attribut (**Statusupdate_time**) qui correspond à la date et l'heure de mise à jour de statut d'une commande.

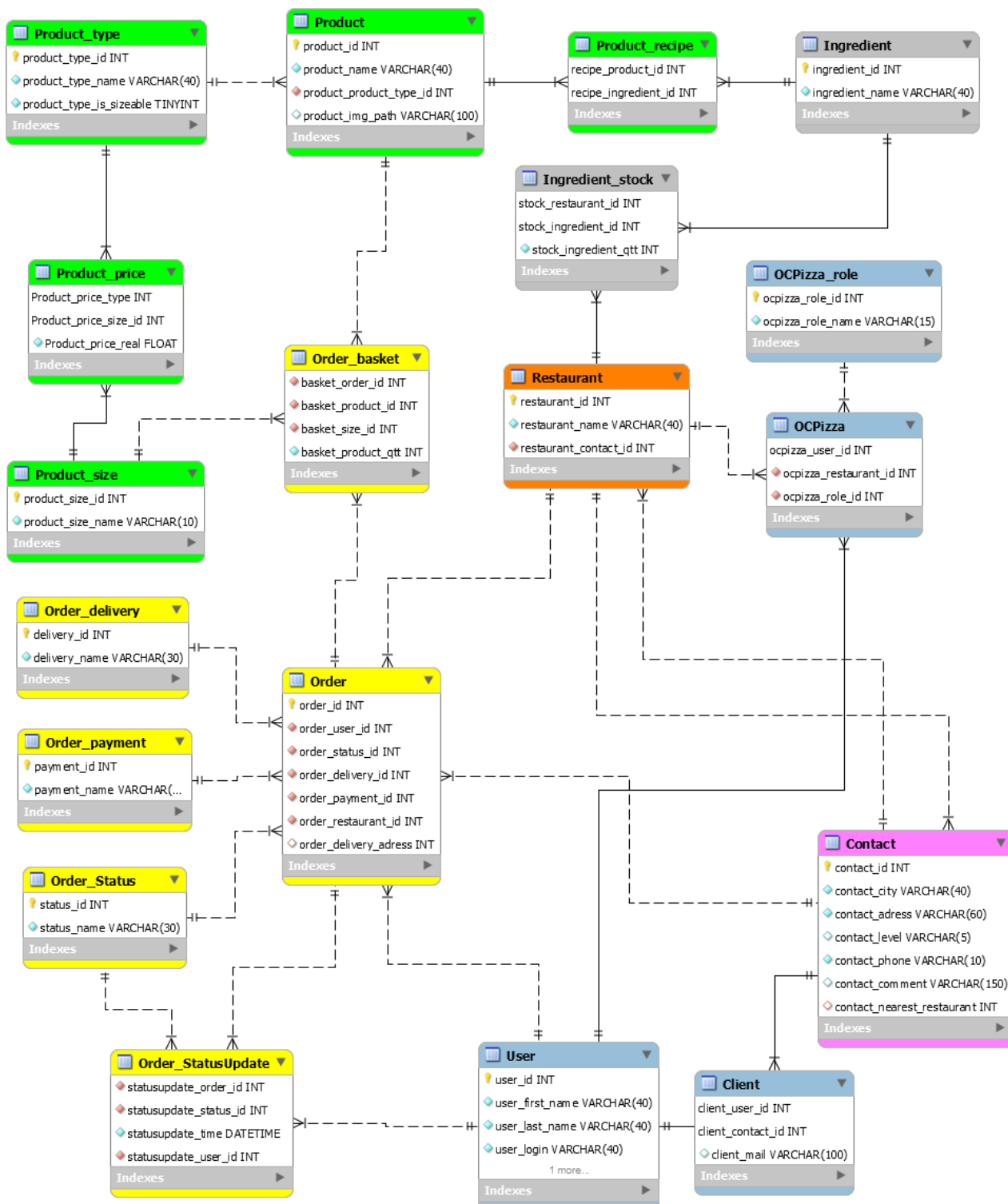
La classe est associée à la classe **Order**.

Order : l'association a déjà été commentée précédemment.

III – MODELE DE DONNEES

1 – Modèle de donnée

Ci-dessous, la représentation du modèle de donnée :



2 – Description des tables

a – Restaurant



Cette table regroupe les informations communes des restaurants OCPizza.

Toutes les colonnes doivent être renseignées (**NOT NULL**)

restaurant_id : clé primaire en **AUTO_INCREMENT** de type **INT**.

restaurant_name : nom du restaurant, type **VARCHAR(40)**

restaurant_contact_id : clé étrangère se référant à **contact_id** de la table **Contact**

b – User



Cette table regroupe les informations communes des utilisateurs.

Toutes les colonnes doivent être renseignées (**NOT NULL**)

user_id : clé primaire en **AUTO_INCREMENT** de type **INT**

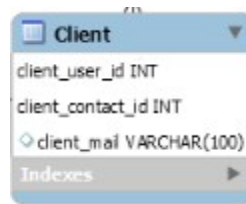
user_first_name : prénom de l'utilisateur, type **VARCHAR(40)**

user_last_name : nom de l'utilisateur, type **VARCHAR(40)**

user_login : login de l'utilisateur, type **VARCHAR(40)**

user_password : mot de passe de l'utilisateur, sera encrypté, type **VARCHAR(20)**

c – Client



Client	
client_user_id	INT
client_contact_id	INT
client_mail	VARCHAR(100)
Indexes	

Cette table fait le lien entre un utilisateur client, son adresse et son adresse mail.

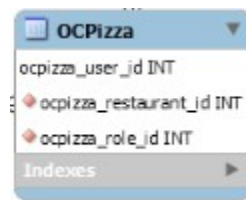
Toutes les colonnes doivent être renseignées (**NOT NULL**)

client_user_id : clé étrangère se référant à **user_id** de la table **User**

client_contact_id : clé étrangère se référant à **contact_id** de la table **Contact**

client_mail : adresse mail du client, type **VARCHAR(100)**

d – OCPizza



OCPizza	
ocpizza_user_id	INT
ocpizza_restaurant_id	INT
ocpizza_role_id	INT
Indexes	

Cette table fait le lien entre un utilisateur OCPizza, son rôle et son restaurant d'affiliation.

Toutes les colonnes doivent être renseignées (**NOT NULL**)

ocpizza_user_id : clé étrangère se référant à **user_id** de la table **User**

ocpizza_restaurant_id : clé étrangère se référant à **restaurant_id** de la table **Restaurant**

ocpizza_role_id : clé étrangère se référant à **role_id** de la table **Role**

e – OCPizza_role



OCPizza_role	
ocpizza_role_id	INT
ocpizza_role_name	VARCHAR(15)
Indexes	

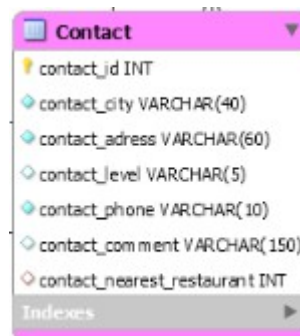
Cette table liste les rôles possibles pour les employés d'OCPizza.

Toutes les colonnes doivent être renseignées (**NOT NULL**)

role_id : clé primaire en **AUTO_INCREMENT** de type **INT**

role_name : nom du rôle (employé, livreur etc.), type **VARCHAR(15)**

f – Contact



Contact	
contact_id	INT
contact_city	VARCHAR(40)
contact_adress	VARCHAR(60)
contact_level	VARCHAR(5)
contact_phone	VARCHAR(10)
contact_comment	VARCHAR(150)
contact_nearest_restaurant	INT
Indexes	

Cette table regroupe les informations communes concernant les coordonnées des clients et restaurants.

Mise à part les colonnes **contact_level** et **contact_comment**, toutes les colonnes doivent être renseignées (**NOT NULL**)

contact_id : clé primaire en **AUTO_INCREMENT** de type **INT**

contact_city : ville de l'entité, type **VARCHAR(40)**

contact_adress : adresse de l'entité, type **VARCHAR(60)**

contact_level : étage de l'entité, type **VARCHAR(5)**

contact_phone : numéro de téléphone de l'entité, type **VARCHAR(10)**

contact_comment : commentaire de l'entité, type **VARCHAR(150)**

contact_nearest_restaurant : clé étrangère se référant à **restaurant_id** de la table **Restaurant**

g – Ingredient



Ingredient	
ingredient_id	INT
ingredient_name	VARCHAR(40)
Indexes	


Cette table liste les ingrédients utilisés par OCPizza.

Toutes les colonnes doivent être renseignées (**NOT NULL**)

ingredient_id : clé primaire en **AUTO_INCREMENT** de type **INT**

ingredient_name : nom de l'ingrédient, type **VARCHAR(40)**

h – Ingredient_stock



Ingredient_stock	
stock_restaurant_id	INT
stock_ingredient_id	INT
stock_ingredient_qtt	INT
Indexes	

Cette table fait le lien entre les ingrédients et les restaurants afin d'établir le stock de chaque enseigne OCPizza.

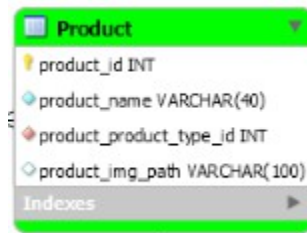
Toutes les colonnes doivent être renseignées (**NOT NULL**)

stock_restaurant_id : clé étrangère se référant à **restaurant_id** de la table **Restaurant**

stock_ingredient_id : clé étrangère se référant à **ingredient_id** de la table **Ingredient**

stock_ingredient_qtt : quantité de l'ingrédient en stock, type **INT**

i – Product



Product	
product_id	INT
product_name	VARCHAR(40)
product_product_type_id	INT
product_img_path	VARCHAR(100)
Indexes	

Cette table regroupe les informations communes des produits.

Mise à part la colonne **product_img_path**, toutes les colonnes doivent être renseignées (**NOT NULL**)

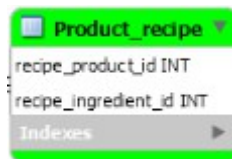
product_id : clé primaire en **AUTO_INCREMENT** de type **INT**

product_name : nom du produit, type **VARCHAR(40)**

product_product_type_id : clé étrangère se référant à **product_type_id** de la table **Product_type**

product_img_path : chemin vers image/représentation du produit, type **VARCHAR(100)**

j – Product_recipe



Product_recipe	
recipe_product_id	INT
recipe_ingredient_id	INT
Indexes	

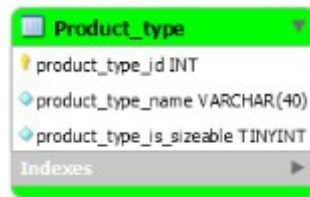
Cette table fait le lien entre les ingrédients et les produits, afin d'établir la recette d'un produit.

Toutes les colonnes doivent être renseignées (**NOT NULL**)

recipe_product_id : clé étrangère se référant à **product_id** de la table **Product**

recipe_ingredient_id : clé étrangère se référant à **ingredient_id** de la table **Ingredient**

k – Product_type



Product_type	
product_type_id	INT
product_type_name	VARCHAR(40)
product_type_is_sizeable	TINYINT
Indexes	

Cette table liste les catégories possibles des produits OCPizza.

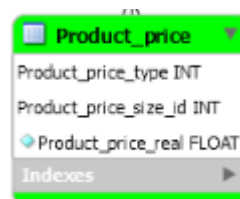
Toutes les colonnes doivent être renseignées (**NOT NULL**)

product_type_id : clé primaire en **AUTO_INCREMENT** de type **INT**

product_type_name : nom de la catégorie, type **VARCHAR(40)**

product_type_is_sizeable : détermine si un produit affilié à cette catégorie peut avoir un attribut de taille, type **TINYINT**

l – Product_price



Product_price	
Product_price_type	INT
Product_price_size_id	INT
Product_price_real	FLOAT
Indexes	

Cette table fait le lien entre une catégorie de produit et une taille afin de déterminer un prix.

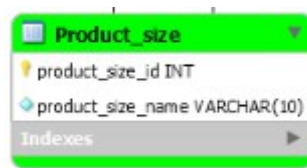
Toutes les colonnes doivent être renseignées (**NOT NULL**)

Product_price_type_id : clé étrangère se référant à **product_type_id** de la table **Product_type**

Product_price_size_id : clé étrangère se référant à **product_size_id** de la table **Product_size**

Product_price_real : prix d'un produit selon sa catégorie et sa taille, type **FLOAT**

m – Product_size



Product_size	
product_size_id	INT
product_size_name	VARCHAR(10)
Indexes	

Cette table liste les tailles possibles pour les produits OCPizza

Toutes les colonnes doivent être renseignées (**NOT NULL**)

Product_size_id : clé primaire en **AUTO_INCREMENT** de type **INT**

Product_size_name : nom de la taille du produit, une entrée « none » existe pour les produits aux formats fixe, type **VARCHAR(40)**

n – Order



Order	
order_id	INT
order_user_id	INT
order_status_id	INT
order_delivery_id	INT
order_payment_id	INT
order_restaurant_id	INT
order_delivery_adress	INT
Indexes	

Cette table regroupe les informations communes des commandes.

Mise à part la colonne **order_delivery_adress**, toutes les colonnes doivent être renseignées (**NOT NULL**)

order_id : clé primaire en **AUTO_INCREMENT** de type **INT**

order_user_id : clé étrangère se référant à **user_id** de la table **User**

order_status_id : clé étrangère se référant à **status_id** de la table **Order_status**

order_delivery_id : clé étrangère se référant à **delivery_id** de la table **Order_delivery**

order_payment_id : clé étrangère se référant à **payment_id** de la table **Order_payment**

order_restaurant_id : clé étrangère se référant à **restaurant_id** de la table **Restaurant**

order_delivery_adress : clé étrangère se référant à **contact_id** de la table **Contact** (si la commande doit être livrée)

o – Order_basket



Order_basket	
•	basket_order_id INT
•	basket_product_id INT
•	basket_size_id INT
•	basket_product_qtt INT
Indexes	

Cette table fait le lien entre une commande, un produit et sa taille, et détermine la quantité de produit sélectionné par un utilisateur.

Toutes les colonnes doivent être renseignées (**NOT NULL**)

basket_order_id : clé étrangère se référant à **order_id** de la table **Order**

basket_product_id : clé étrangère se référant à **product_id** de la table **Product**

basket_size_id : clé étrangère se référant à **product_size_id** de la table **Product_size**

basket_product_qtt : quantité d'un produit dans le panier, type **INT**

p – Order_delivery



Order_delivery	
•	delivery_id INT
•	delivery_name VARCHAR(30)
Indexes	

Cette table liste tout les mode de livraison possibles.

Toutes les colonnes doivent être renseignées (**NOT NULL**)

delivery_id : clé **primaire** en **AUTO_INCREMENT** de type **INT**

delivery_name : nom du mode de livraison, type **VARCHAR(30)**

q – Order_payment



Order_payment	
payment_id	INT
payment_name	VARCHAR(...)
Indexes	

Cette table liste tout les mode de paiement possibles.

Toutes les colonnes doivent être renseignées (**NOT NULL**)

payment_id : clé primaire en **AUTO_INCREMENT** de type **INT**
payment_name : nom du mode de paiement, type **VARCHAR(25)**

r – Order_status



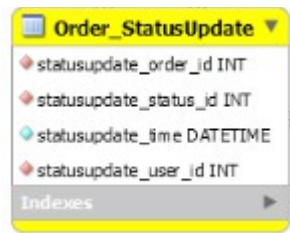
Order_Status	
status_id	INT
status_name	VARCHAR(30)
Indexes	

Cette table liste tout les statuts possibles.

Toutes les colonnes doivent être renseignées (**NOT NULL**)

status_id : clé primaire en **AUTO_INCREMENT** de type **INT**
status_name : nom du mode de paiement, type **VARCHAR(30)**

s – Order_StatusUpdate



Order_StatusUpdate	
statusupdate_order_id	INT
statusupdate_status_id	INT
statusupdate_time	DATETIME
statusupdate_user_id	INT
Indexes	

Cette table fait le lien entre une commande, un statut et un utilisateur afin de savoir qui a modifié le statut d'une commande, et quand.

Toutes les colonnes doivent être renseignées (**NOT NULL**)

statusupdate_order_id : clé étrangère se référant à **order_id** de la table **Order**

statusupdate_status_id : clé étrangère se référant à **status_id** de la table **Order_status**

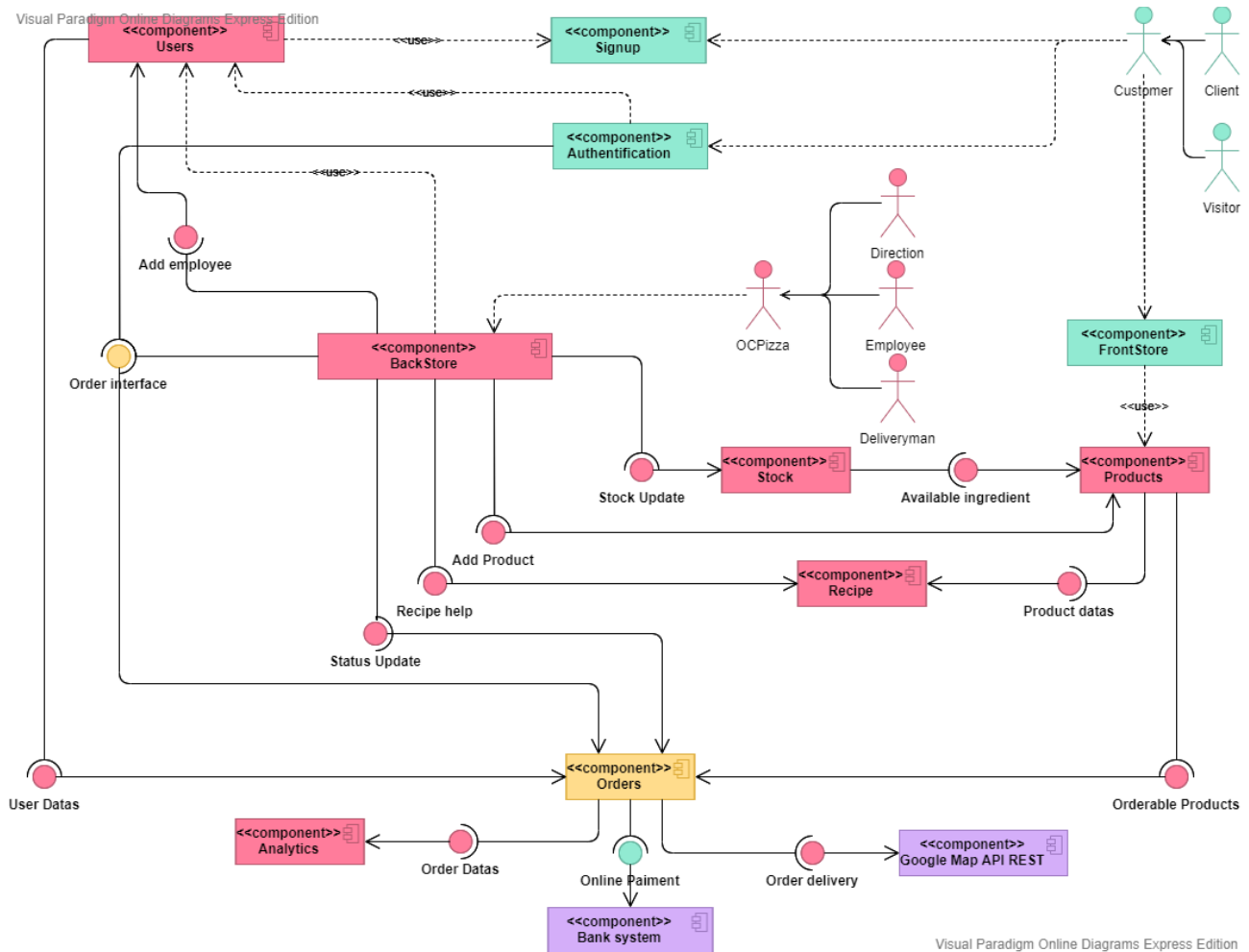
statusupdate_time : heure et date de la mise à jour du statut d'une commande, type **DATETIME**

statusupdate_user_id : clé étrangère se référant à **user_id** de la table **User**

IV – DIAGRAMME DE COMPOSANTS

1 – Diagramme de composant général

Ci-dessous, le diagramme de composant général :

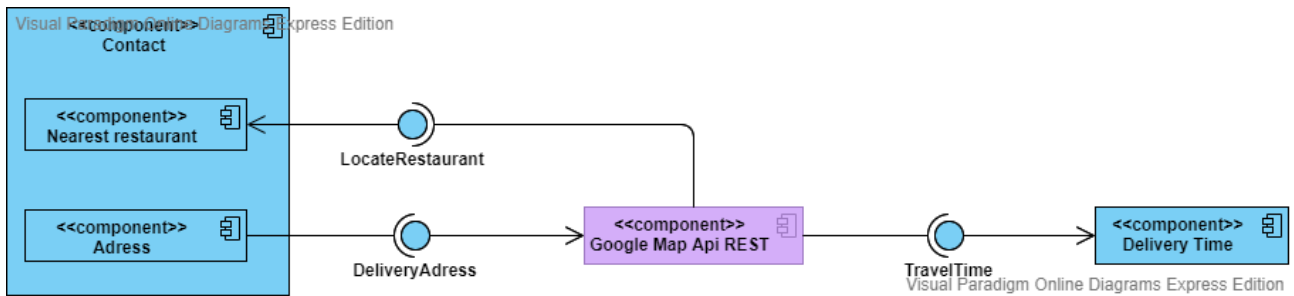


Notes :

- en **Vert** : composants et interface pouvant être utilisée par les utilisateurs hors-OC Pizza
- en **Jaune** : composants et interface pouvant être utilisée par tous les utilisateurs
- en **Rouge** : composants et interface pouvant être utilisée par les utilisateurs OC Pizza
- en **Violet** : composants extérieurs

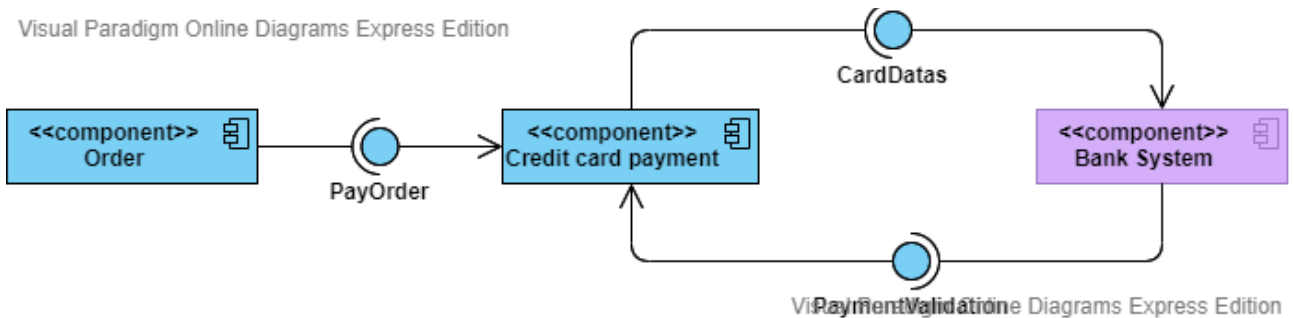
2 – Diagramme de composant API REST Google map

Ci-dessous, le diagramme de composant de l'API REST Google map :



3 – Diagramme de composant système bancaire

Ci-dessous, le diagramme de composant du système bancaire :



V – DIAGRAMME DE DEPLOIEMENT

Ci-dessous, le diagramme de déploiement :

