OC Pizza

Mise en place du nouveau système informatique

Dossier de conception technique

Version 1.1

Auteur
Alexandre Simões Mendes
Développeur d'applications android

VERSIONS

Auteur	Date	Description	Version
Simoes Mendes Alexandre	15/10/2021	Création du document	1.0
Simoes Mendes Alexandre	01/11/2021	Reformulation, modification du document	1.1

INTRODUCTION

Objet du document

Le présent document constitue le dossier de conception technique du système informatique pour l'entreprise OC Pizza.

Objectif du document est de présenter différents outils mis à disposition d'OC Pizza lors de la mise en place du nouveau système informatique.

Les éléments du présent dossier découlent :

- Des différents entretiens avec les dirigeants de l'entreprise OC Pizza
- De l'analyse des besoins effectuée par ITCons&Dev

Références

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

1. DCT - 001 : Dossier de conception fonctionnelle
2. DCT - 003 : Dossier d'exploitation

ARCHITECTURE TECHNIQUE

Site Web

Les clients auront accès au « Frontend » de ce site soit la devanture du magasin, les autres acteurs auront accès quant à eux au « Backend » soit l'arrière-boutique de celui-ci.

Pour répondre aux divers besoins du nouveau système informatique, nous recommandons fortement de construire un site internet sur mesure, donc de partir de zéro. Cela permettra d'avoir un produit qui répondra aux besoins actuels, et qui pourra facilement évoluer dans le temps si de nouveaux besoins venaient à apparaître. Nous verrons plus bas dans le document plus en détails cette solution.

Pour ce qui est du frontend du site internet que nous souhaitons mettre en place, nous utiliserons deux langages : HTML5, CSS3 et enfin le langage de programmation PHP, permettant ainsi de réaliser un site dynamique. PHP est d'ailleurs une bonne solution pour utiliser notre base de données, mais nous verrons cela plus tard.

Pour ce qui est du backend, nous avons préféré nous orienter vers une solution codée en J2EE sur le Framework Spring. Ce site devra être parfaitement adapté aux navigateurs mobiles (Responsive Web Design). Nous verrons dans la partie suivante, concernant l'application mobile, en quoi ce point est crucial dans notre solution.

Application Mobile

Grace au site internet développé pour s'adapter parfaitement à une utilisation sur navigateur mobile, comme vu plus haut, nous avons décidé de partir sur une application mobile simple mais très efficace.

En effet, dès que l'utilisateur lancera l'application « OC Pizza », le site s'affichera directement dans celle-ci. Cette application « CRUD » (« Create, Read, Update, Delete » / « Créer, Lire, Mettre à jour, Supprimer ») fonctionnera directement en lien avec :

- Le site internet (avec le protocole http, qui permet d'échanger des pages web entre le client et le serveur)

- La base de données (avec le langage SQL pour communiquer avec cette dernière)

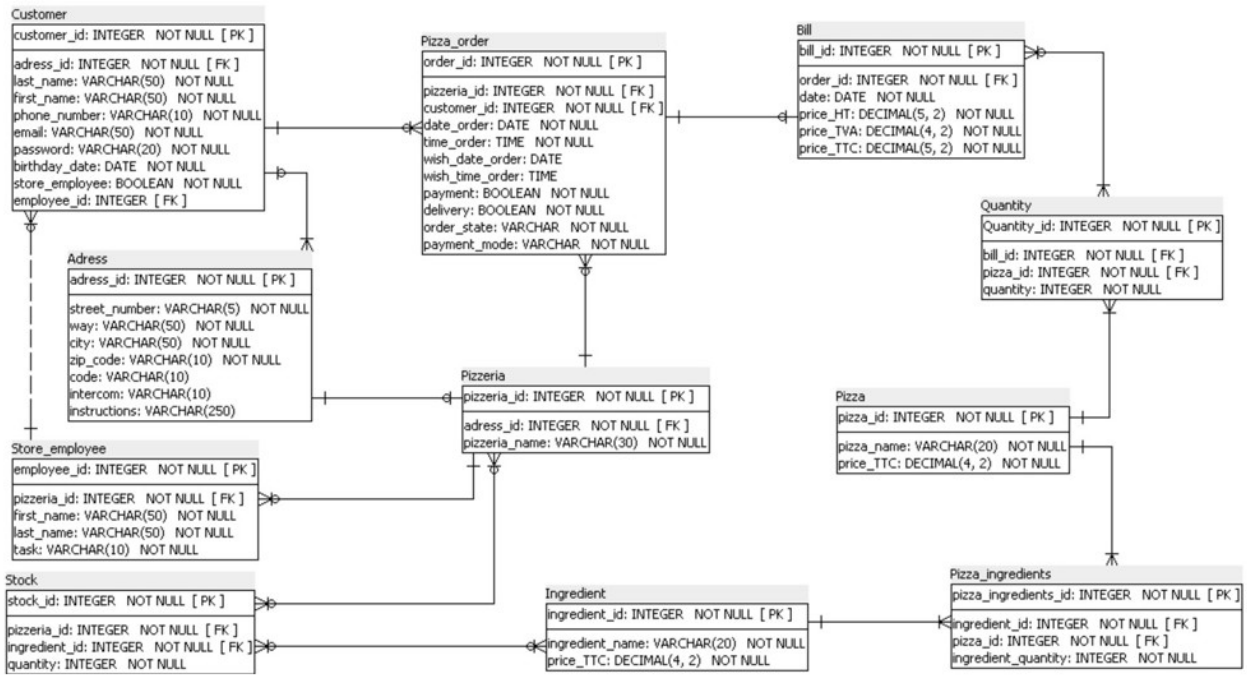
L'application sera développée grâce à Angular, un framework qui permet de créer des applications web dynamiques et immersives en utilisant TypeScript comme langage de programmation. Elle sera cross-plateforme, c'est à dire utilisable aussi bien avec un téléphone sous Android qu'avec un téléphone sous IOS.

Base de données

Pour répondre aux besoins du nouveau système informatique, la gestion des données devra se faire à travers une base de données (SGBD) stockée sur un cloud. Nous avons privilégié l'utilisation de MS SQL, le système de gestion de base de données conçu par Microsoft, compatible avec le langage PHP et facilement utilisable dans notre solution de mise en place du nouveau système informatique.

Bien que payante, nous verrons, là aussi, plus bas dans le document les détails de cette solution.

Modèle Physique de données



Le Modèle Physique de Données (MPD en abrégé ou Physical Data Model en anglais) va nous servir à modéliser la base de données relationnelle d'OC Pizza. Intéressons-nous plus en détails aux différentes tables de ce MPD.

A. La table Pizza_order

Presque toutes les colonnes de cette table doivent être renseignées (« NOT NULL »), les seules exceptions étant « whish_date_order » (date souhaitée de livraison) et « wish_time_order» (heure souhaitée de livraison) .

Nous retrouvons au sein de cette table l'attribut « order_id », qui est la clé primaire de cette table. Cette clé primaire servira de clé unique qui ne correspondra uniquement qu'à une commande. Ainsi, chaque commande ne possèdera qu'un numéro d'identification. Le type de cet attribut est « INTEGER ».

L'attribut « date_order », de type « DATE », nous permet quant à lui de définir la date à laquelle la commande a été passée par le client au format « YYYY-MM-DD ».

L'attribut « wish_date », est de même type, mais n'est utilisée que si le client le souhaite, dans le cas d'une livraison qui n'est pas immédiate dans la journée, voir qui n'est pas du tout pour le jour même. Il va de pair avec « wish_time » qui permet, lui, de définir l'heure à laquelle le client souhaite recevoir sa commande. « wish_time », est de type « TIME », comme « time_order », qui nous permet de définir l'heure à laquelle la commande a été passée par le client, au format « HH:mm:ss »

Nous avons aussi au sein de cette table, l'attribut « payment », qui est de type « BOOLEAN ». Celui-ci permet de connaitre si oui ou non, la commande d'un client a déjà été réglée. Les seules réponses possibles pour cet attribut sont donc « TRUE » ou « FALSE » (vrai ou faux). Un autre attribut de ce type fait partie de cette table, « delivery », qui permet quant à lui de savoir si le client souhaite une livraison ou un retrait en magasin. Là aussi, deux uniques réponses sont possibles « TRUE » ou « FALSE ».

Les deux derniers attributs de la table « Pizza order » ont la particularité de renvoyer tous les deux à des énumérations. En effet, « order_state » qui permet de connaitre l'état de la commande à un instant « t » et « payment_mode » qui lui, nous renseigne sur le moyen de paiement sélectionné par le client, renvoient tous les deux aux tableaux « <<Enumerate>> Order State » et « <<Enumerate>> Payment Mode » vu un peu plus tôt dans ce document. Les deux attributs sont donc de type « VARCHAR », configuré avec une capacité de 20 caractères.

B. La table Customer

Dans cette table aussi, presque toutes les colonnes doivent être renseignées (« NOT NULL »), la seule exception étant « employee_id » (numéro d'identification de l'employé) qui n'est utilisé que si le client est passé par un employé du magasin pour passer sa commande.

La clé primaire de cette table est « customer_id », un numéro d'identification unique pour chaque client du restaurant. Notons, que dans cette table, nous avons aussi trois clés étrangères : « order_id », « address_id » et « employee_id ». Voyons plus en détails ces trois clés ci.

L'attribut « order_id », est une clé étrangère qui permet d'associer un client avec un numéro de commande établit auparavant dans la table « Pizza order ». La deuxième clé étrangère, comme nous l'avons vu, est « adress_id », qui permet d'attribuer à un client une ou plusieurs adresses, qu'il aura préalablement définit sur son compte ou en magasin. Cette clé, permet de faire le lien avec la table « Address », que nous verrons plus en détails plus tard. La dernière clé étrangère est donc « employee_id », qui permet d'associer la classe « Customer » avec la classe « Store_employee ». Ces trois attributs sont de type « INTEGER ».

Revenons quelques instants sur cette dernière clé étrangère. Elle a la particularité de n'être utilisée que si l'attribut « Store_employee », de type « BOOLEAN » s'est vu attribuer une valeur « TRUE ». Ainsi, si la commande est prise par un employé du magasin et non passée par internet, nous pouvons connaitre la personne étant intervenu dans le processus de commande. Bien entendu, nous retrouvons ici des attributs qui permettent de renseigner des informations sur le client, comme « last_name », qui permet de récupérer le nom de famille du client, mais aussi comme « first_name », qui permet lui de connaitre le prénom d'un client. Les deux sont de types « VARCHAR », avec une capacité totale de 50 caractères chacun, afin de permettre aux clients de renseigner le mieux possible leurs informations personnelles. Toujours concernant les informations personnelles, l'attribut « birthday_date », de type « DATE » permettra de recueillir la date d'anniversaire de la personne, afin de lui transmettre des offres ou des promotions la semaine de cet évènement, par exemple.

Pour ce qui est des informations de contact, l'attribut « email » nous est utile afin de récupérer le mail de l'utilisateur. Son type est de « VARCHAR » avec une capacité de 50 caractères, ce qui nous semble l'idéal dans l'optique où un utilisateur aurait utilisé tous ses noms et prénoms pour composer son adresse mail. L'attribut « phone_number » permet quant à lui de pouvoir avoir un numéro de téléphone de contact, que ce soit pour le magasin ou le livreur. Son type est « VARCHAR » là aussi, mais avec une capacité de 10 caractères, suffisant que ce soit pour un numéro fixe ou mobile.

Enfin, l'attribut « password », de type « VARCHAR » lui aussi, mais avec une capacité de 20 caractères, permettra de récupérer le mot de passe définit par l'utilisateur.

C. La table Adress

Dans cette table aussi, presque toutes les colonnes doivent être renseignées (« NOT NULL »), les seules exceptions étant « code », « intercom » et « instructions »

La clé primaire de cette table est « adress_id », un numéro d'identification unique pour les adresses renseignées. Notons, que dans cette table, nous avons aussi une clé étrangère : « pizzeria_id», qui permet d'associer une pizzeria à une adresse. Celle-ci est de type « VARCHAR ».

Pour éviter de lourdes répétitions, partons du postulat que les prochains attributs concernent aussi bien le domicile du client que celui d'un des restaurants. L'attribut « street_number » permet de pouvoir renseigner le numéro de l'habitation, « way » quant à lui permet de renseigner la voie, « city » lui la ville et finalement « zip_code », le code postal. Avec ces quatre attributs, tous de type « VARCHAR » nous avons déjà de quoi pouvoir récupérer les principales informations concernant une adresse. Pour permettre une livraison à domicile, dans le cas où cela est nécessaire, nous pouvons récupérer le « code » d'entrée l'immeuble, l' « intercom » ou interphone, que ce soit un nom ou un numéro, mais surtout « instructions » qui permet de saisir des instructions précises concernant la livraison. Là aussi, tous sont de type « VARCHAR » ; notons que les instructions peuvent contenir jusqu'à 255 caractères, ce qui laisse la possibilité à l'utilisateur d'être le plus précis possible.

D. La table Pizzeria

Toutes les colonnes doivent être renseignées (« NOT NULL ») pour cette table.

La clé primaire de cette dernière est l'attribut « pizzeria_id » qui permet à la table « Pizzeria » d'être associée à la table « Address ». Ainsi, lorsqu'un client passe une commande, selon son adresse, le restaurant le plus proche de chez lui, lui sera attribué.

Cette table possède aussi une clé étrangère, « order_id » de type « INTEGER » qui renvoie au numéro de commande attribué par la table « Pizza order ».

Enfin, « pizzeria_name » permet d'avoir la raison sociale de l'établissement concerné, son type est donc évidemment de type « VARCHAR », avec une capacité de trente caractères, afin de parer toute éventualité.

E. La table Store_employee

Pour cette table aussi, toutes les colonnes doivent être renseignées (« NOT NULL »).

La clé primaire de « Store_employee » est « employee_id », un « INTEGER » qui permet d'attribuer un numéro unique à chaque employé du réseau OC Pizza. Une seule clé étrangère est ici présente, « pizzeria _id » qui indique pour un numéro d'employé donné, la pizzeria qui l'emploie. Nous avons ici affaire avec un attribut de type « VARCHAR ».

Les attributs « last_name » et « first_name », comme dans la table « Customer », permettent de récupérer le nom et prénom, mais ici, d'un employé, en fonction de son numéro d'identification. Les deux attributs sont de type « VARCHAR » avec une limite de 50 caractères.

Enfin, l'attribut « task », permet de connaitre la fonction de l'employé du magasin. Il renvoie à une énumération de quatre possibilités, comme vu lors du chapitre précédent.

F. La table Bill

Toutes les colonnes doivent être renseignées (« NOT NULL ») pour cette table aussi.

La clé principale de cette table est « bill_id » qui permet de connaitre le numéro de la facture, un numéro unique. Son type est donc logiquement un « INTEGER ». Type partagé avec la clé étrangère, « order_id » qui renvoie au numéro de la commande à laquelle fait référence la facture. L'attribut « date » permet de connaitre la date à laquelle a été établie la facture par le magasin, son type est donc « DATE ».

Enfin, pour établir une facture, connaitre les différents prix est une obligation ; ainsi, grâce à « Price_HT » qui permet de connaitre le prix total hors taxe, « Price_TVA » qui renseigne le prix de la TVA sur la commande, et « Price_TTC » qui renseigne le montant total de la commande nous pouvons récupérer les informations nécessaires. Les trois attributs sont de même type, « DECIMAL ».

G. La table Quantity

Toutes les colonnes doivent être renseignées (« NOT NULL »).

Cette table permet d'établir pour une commande client, la quantité demandée pour une pizza. Ainsi, sa clé primaire n'est autre que «quantity_id », et comme tous les numéros d'identifications ici, il est de type « INTEGER ». Les deux clés étrangères de cette table permettent de récupérer le numéro d'identification unique et donc la facture du client (« bill_id ») et le numéro d'identification des pizzas demandées par le client.

Enfin, l'attribut « quantity » permet de connaitre exactement la quantité pour chaque pizza d'une Commande.

H. La table Pizza

Toutes les colonnes doivent être renseignées (« NOT NULL »).

Pour la table « Pizza », la clé primaire est « pizza_id » qui permet de connaitre le numéro d'identification unique de la pizza. Il n'y a pas de clé étrangère au sein de cette table, qui permet de lister l'ensemble des pizzas proposés par OC Pizza.

Nous retrouvons aussi au sein de cette table, l'attribut « pizza_name », de type « VARCHAR » qui nous permet de connaitre le nom d'une pizza, tel qu'il est indiqué sur le menu, tant sur internet, que sur les divers supports du restaurant.

Enfin, nous avons aussi l'attribut « Price_TTC », qui permet de connaitre le prix de la pizza.

I. La table Ingredient

Toutes les colonnes doivent être, là aussi, renseignées (« NOT NULL »).

Pour la table « Ingrédient », la clé primaire est « ingredient_id » qui permet d'attribuer un numéro d'identification unique pour un ingrédient, pour une pizza lors d'une commande. Nous n'avons aucune clé étrangère au sein de cette table, qui permet de lister l'ensemble des ingrédients dont dispose chaque magasin OC Pizza.

Nous retrouvons aussi au sein de cette table, l'attribut « ingredient_name », de type « VARCHAR » qui nous permet de connaitre le nom d'un ingrédient, tel qu'il est indiqué sur le menu, tant sur internet, que sur les divers supports du restaurant.

Enfin, nous avons aussi l'attribut qui permet de connaitre le prix de cet ingrédient à travers « price_TTC ».

J. La table Pizza_ingredients

Toutes les colonnes doivent être renseignées (« NOT NULL »).

La clé primaire de cette table est là aussi un numéro d'identification, ici, « pizza_ingredients_id ». Le reste de la table est composé uniquement de deux clés étrangères, « pizza_id » et « ingredient_id », permettant d'associer à une pizza, les ingrédients nécessaires à sa préparation. Enfin, « ingredient_quantity » permet de connaitre la quantité nécessaire d'un ingrédient. Tous les attributs sont ici des « INTEGER ».

K. La table Stock

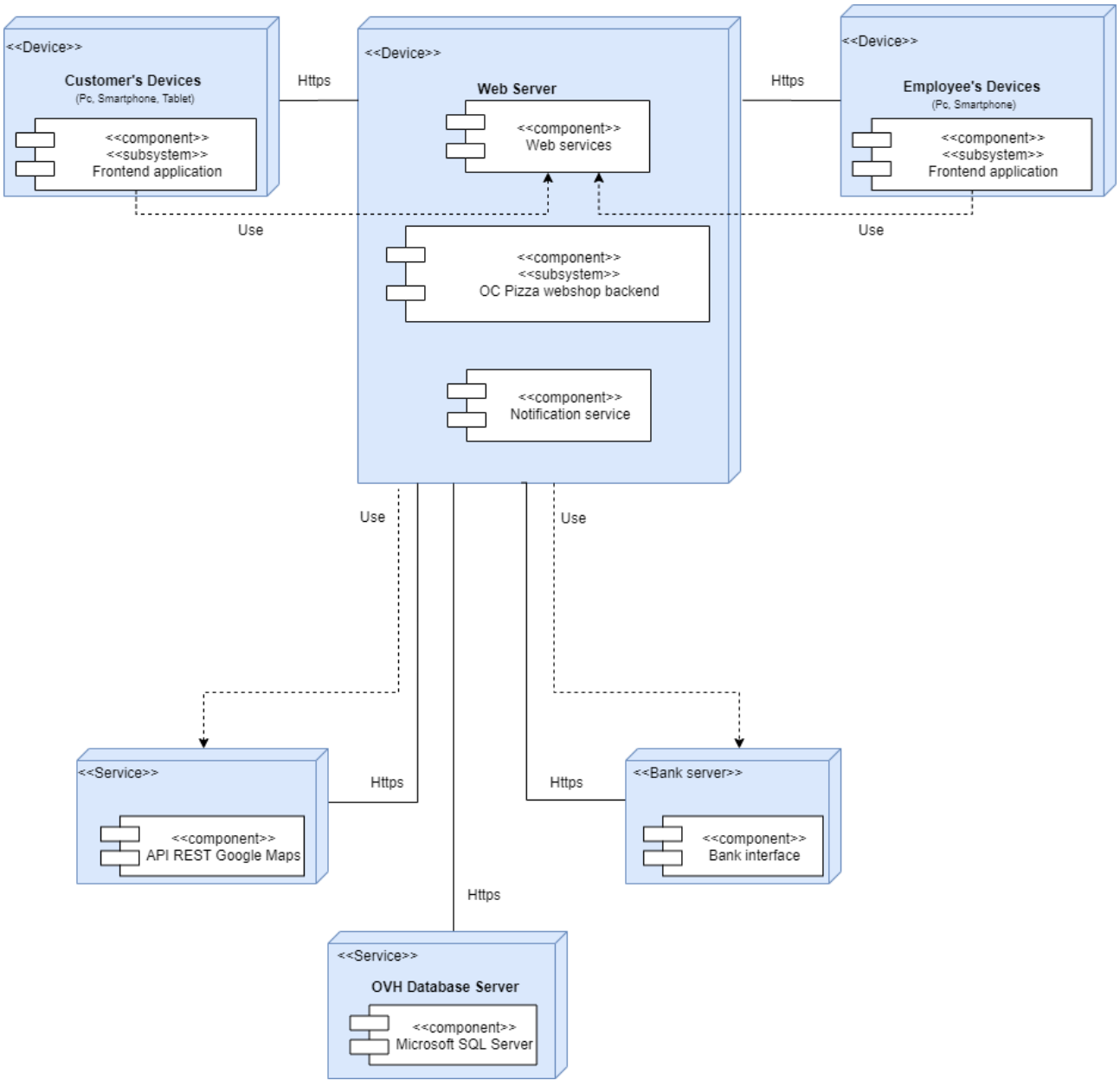
Toutes les colonnes doivent être renseignées (« NOT NULL »)

La clé primaire de cette table est « stock_id », un « INTEGER » qui est aussi un numéro d'identification. Les deux clés étrangères de la table sont aussi de ce type, que ce soit « pizzeria_id » ou « ingredient_id ».

Idem pour « quantity » qui permet de connaitre la quantité disponible d'un ingrédient (grâce à « ingredient_id ») dans une pizzeria donnée (grâce à « pizzeria_id »).

ARCHITECTURE DE DÉPLOIEMENT

Déploiement de l'application



Intéressons-nous un peu plus en détail au diagramme de déploiement. Nous pouvons constater qu'il y a deux nœuds qui représentent les différents matériels que les différents acteurs peuvent utiliser lorsqu'ils utilisent le site internet ou l'application mobile OC Pizza.

D'un côté, nous avons le nœud « Customer's devices », qui représente les appareils utilisés par les clients du restaurant pour passer une commande. Ces utilisateurs ont accès à une « Frontend Application ».

De l'autre côté nous retrouvons le nœud « Employee's devices », qui quant à lui représente les supports matériels utilisés par les employées des différents magasins pour accéder aux différentes interfaces à leur disposition.

Que ce soit à travers le site internet ou l'application, c'est à travers des requêtes HTTPS (HyperText Transfer Protocol Secure, ou protocole de transfert hypertextuel sécurisé) qu'ils communiquent tous deux avec un troisième nœud, le « Web Server ».

Ce serveur web, comporte plusieurs composants tel que le « Web Service » qui permet aux différents éléments du diagramme de communiquer entre eux et d'échanger des données. « OC Pizza webshop Backend » est, comme son nom l'indique, le composant qui représente « l'arrière-boutique » du restaurant, du moins de son site internet, uniquement utilisé par les différents salariés du magasin.

Le dernier composant de « Web Server » est « Notification service » qui est quant à lui le composant qui permet d'informer, à travers le système informatique, les différents acteurs, comme par exemple, à travers un « tracker » qui suit l'évolution de la préparation d'une commande, d'une livraison.

Pour finir, nous pouvons aussi dire que le « Web Server » permet de communiquer, là aussi en HTTPS, avec trois autres nœuds, qui représentent ici trois services externes. Tout d'abord l'utilisation d'une API REST, ici Google Maps, qui sera utilisée pour les informations relatives de près ou de loin à la géolocalisation des clients et des restaurants OC Pizza. Ensuite l'accès à l'interface et aux services de la banque partenaire de la chaine de restaurants, afin de pouvoir procéder à la vérification des données renseignées par les clients et relative au paiement de leurs commandes, interface qui permettra donc de valider ou non le paiement de ces dernières. Enfin, l'accès aux serveurs loués chez OVH et donc l'accès aux bases de données des magasins.

Serveur de déploiement : OVH

Nous avons décidé de profiter des services de l'entreprise OVH, qui nous permet :

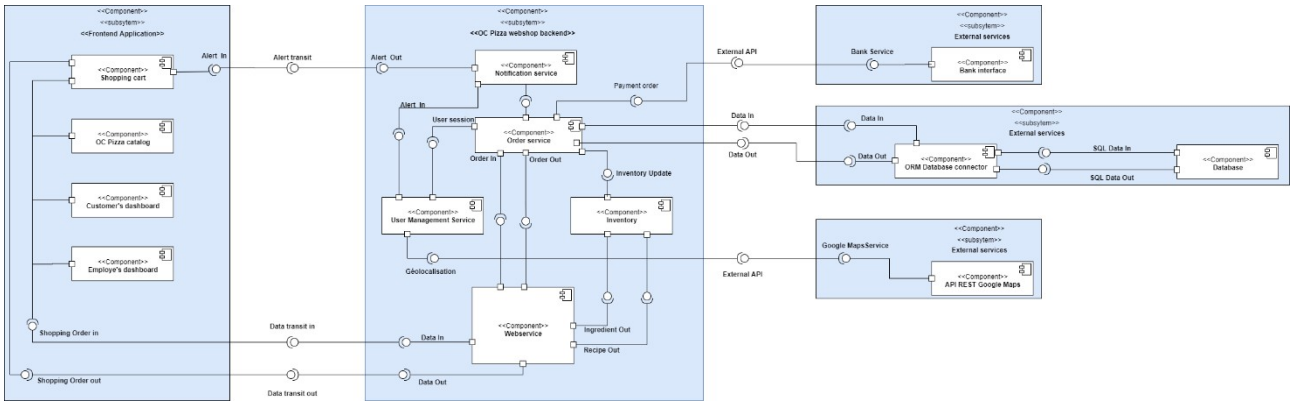
- D'acheter le nom de domaine "OCPizza.store" pour 1.49 euros la première année.

- D'héberger le site internet grâce à l'offre "hébergement business" à 11.99 euros par mois, offre adaptée aux magasins en ligne notamment.

- Location d'un server cloud afin d'héberger la base de données et bénéficier de Microsoft SQL avec l'offre "MS SQL Server Web Edition" à 27.60 euros par mois.

ARCHITECTURE LOGICIELLE

Les différents composants



Prenons le temps de nous intéresser un peu plus en détails au diagramme de composant ci-dessus et en le détaillant plus explicitement.

La gestion des données se fera grâce à une base de données (SGBD) stockée sur un cloud, Pour l'ensemble de ces solutions (server et SGBD) nous passerons par l'intermédiaire d'un prestataire : OVH. Leurs nombreuses offres groupées permettent d'obtenir les outils nécessaires quant à la mise en place de la solution proposée et cela à un coût attractif, notamment un partenariat avec Microsoft pour la mise en place d'une base de données, avec MS SQL Server, ainsi que la possibilité de bénéficier des services nécessaires en cloud, ce qui présente de nombreux avantages.

Le « Webshop » communiquera avec une base de données à distance, en cloud, comme vu plus haut dans le document. La solution du cloud est privilégiée car elle permet un gain de place non négligeable, mais surtout d'éviter tous les problèmes que pourraient subir des serveurs dit « physiques », comme les pannes, les incidents (incendies, inondations ou tout autre problème du même type) qui pourraient causer la perte des données. Le cloud permet aussi de mieux gérer l'afflux de clients sur le site internet, et donc d'éviter tout incident relatif à un trop plein de visiteurs.

Pour la gestion des paiements en ligne, nous passerons là aussi par une API externe, celui de la banque avec laquelle OC Pizza travaille. Le client remplira d'abord un formulaire afin de renseigner toutes les informations nécessaires afin de procéder au règlement, en ligne, par carte bancaire, comme le numéro de carte, le type de carte, le nom, la date d'expiration et le cryptogramme. Et cela à partir du site internet ou de l'application OC Pizza. L'API externe communiquera ainsi les informations renseignées par l'utilisateur à la banque partenaire, qui validera ou non, instantanément, le paiement. Si ce dernier est validé, la commande l'est aussi.

L'utilisation de l'API REST Google MAPS quant à elle permet de géolocaliser un client. Ainsi, lors de son inscription sur le site internet, ou en magasin, lorsque l'adresse de ce dernier est renseignée, l'API

permet de connaitre le magasin le plus proche de son domicile et le client est donc « rattaché » à celui-ci. Cette API est aussi pratique pour les livreurs, qui pourront consulter l'itinéraire, le temps de trajet jusqu'au domicile du client, et même se servir de ce service sur leur téléphone portable comme GPS à travers l'interface de leur application. Ces informations sont aussi complétées par le temps de trajet estimé, qui s'ajoute alors au temps de préparation de la commande, ce qui permet au final d'estimer un temps de livraison dont le client sera informé sur son « tracker » disponible sur son espace personnel, sur la page de la commande concernée.

Le Frontend de l'application ou du site internet est composé de quatre éléments distincts. D'abord le« Shopping cart », qui n'est autre que le panier de l'utilisateur. Ce dernier peut ainsi décider d'ajouter ou supprimer des produits au sein de son panier. Lorsque le panier est validé, une alerte est envoyée au composant « Notification Service », faisant partie du backend.

Nous retrouvons aussi « OC Pizza catalog », permet d'afficher l'ensemble du catalogue des restaurants OC Pizza sur le site web ou l'application mobile. C'est à partir du catalogue que le client va pouvoir constituer son panier, où que le vendeur va pouvoir constituer un panier pour un client. Ensuite nous retrouvons le « Customer's dashboard », qui n'est autre que le tableau de bord du client. C'est à travers ce composant que l'utilisateur peut modifier ses informations personnelles préalablement renseignées lors de son inscription, consulter son historique de commande, suivre la préparation de sa commande etc.

Enfin, le « Employee's dashboard » quant à lui, est le tableau de bord des différents employés du magasins. C'est à travers ce composant qu'ils peuvent consulter les différentes commandes passées par les clients, suivre la préparation d'une commande, retrouver un compte client, consulter les informations d'un client et les modifier etc.

De l'autre côté nous avons le Backend du webshop est composé de cinq éléments. Commençons par «Notification service », dont nous avons déjà parlé précédemment, qui permet de recevoir des alertes lorsque le panier d'un client est validé et que la commande est elle aussi validée.

Nous retrouvons aussi le composant « Order Service » qui permet de gérer les différentes commandes, passées et en cours grâce aux informations communiquées par le « Webservice », de changer leurs statuts mais aussi de communiquer avec la base de données pour la mettre à jour, mais aussi avec le service bancaire externe afin de procéder au paiement des commandes. Ce composant permet aussi de mettre à jour le composant « Inventory » en actualisant les stocks, les ingrédients et les recettes disponibles. Ce dernier permet d'ailleurs d'avoir accès aux différents inventaires et stocks, magasin par magasin, ingrédient par ingrédient, grâce aux informations communiquées par « Order Service ».

Enfin, « User Management Service » permet de son côté de gérer différents processus comme l'authentification d'un client, la modification d'un mot de passe, la modification d'informations relatives au compte d'un client. C'est aussi ce composant qui nous permet de géolocaliser un client, et donc de lui attribuer un restaurant, ou simplement de fournir un itinéraire au livreur du magasin.

POINTS PARTICULIERS

Ressources

Celles-ci devront être fournie par OC Pizza, que ce soit :

- La chartre graphique nécessaire à la création et développement du site internet
- Les données nécessaires à la mise en place de la base de données pour les magasins.

Environnement de développement

Pour le front-end du site :

- Un éditeur de texte pour le HTML et le CSS, comme NotePad++, Atom.io ou SimpleText.
- Le framework Symfony pour le PHP.

Pour le back end du site :

- Utilisation du framework Spring. Utilisation du langage J2EE.

Pour l'application mobile :

- Utilisation du framework Angular. Utilisation du langage TypeScript

Pour la base de données :

- Utilisation du SGBD MS SQL et du langage SQL.

Procédure de packaging / livraison

Le site et la base de données seront déployés au moment de la livraison, directement sur le serveur OVH.

L'application mobile sera déployée sur le Playstore et l'AppStore, là aussi, au moment de la livraison.

GLOSSAIRE

Framework / IDE	Environnement de travail.
MPD	Modèle Physique de données.
SGBD	Système de gestion de base de données.
Frontend	Partie visible de l'iceberg, élément d'une page web avec lesquels un utilisateur peut interagir.
Backend	Partie cachée de l'iceberg, sert la partie frontend, directement ou par l'intermédiaire d'un programme.