

Table des matières

I. Objet du document.....	2
II. Contexte actuel	2
III. Synthèse des besoins.....	2
IV. Administration des pizzerias	3
V. Définition des acteurs	3
A. Acteurs principaux.....	3
B. Acteurs secondaires.....	4
VI. Définition des packages	5
A. Interface Client	5
B. Interface pizzeria	5
VII. Etude de cas d'utilisations	6
VIII. Solution technique	10

I. Objet du document

Analyse des besoins d'« **OC Pizza** » en terme de développement d'une solution logicielle visant à améliorer la gestion des commandes de pizzas et à permettre aux clients de commander et, éventuellement de payer en ligne.

Ce document a pour objet de définir avec OC Pizza le périmètre global de la solution, il évoluera au fil du temps et selon les principes agiles d'interaction entre notre équipe et OC Pizza.

II. Contexte actuel

« **OC Pizza** » est un groupe de pizzerias spécialisé dans les ventes à emporter ou à livrer. Le groupe compte 5 points de ventes et prévoit d'en ouvrir 3 autres dans les temps à venir.

La direction d'OC Pizza nous sollicite pour le développement de cette solution logicielle car aucun produit sur le marché ne fournit le suivi des commandes, la gestion de stock en temps réel et une interface client permettant de commander.

III. Synthèse des besoins

le client souhaite :

- une application qui gère toutes les pizzerias de l'entreprise.
- Une meilleure automatisation des tâches pour aller plus vite.
- Un suivi en temps réel des commandes.
- Un suivi en temps réel des stocks d'ingrédients.
- Pouvoir passer des commandes sur internet (en plus du téléphone et sur place)
- Payer en ligne (sinon au moment de la livraison)
- Modifier ou annuler la commande tant que sa préparation n'a pas commencé.
- Proposer un aide-mémoire aux pizzaiolos.

IV. Administration des pizzerias

L'application communiquera avec Google map, qui répertoriera toutes les pizzerias du groupe. Le client (ou le vendeur qui passera la commande au client) sélectionnera une adresse qui sera entrée dans cette map, et la pizzeria, dont l'itinéraire le moins long par rapport à l'adresse tapée, sera choisie pour traiter sa commande.

V. Définition des acteurs

Voir diagramme de contexte

Suite à une analyse primaire du cahier de charges, nous pouvons distinguer 6 acteurs, dont 5 principaux et 1 secondaire :

- Client
- Vendeur
- Pizzaiolo
- Livreur
- Patron
- Système bancaire

A. Acteurs principaux

- 4 acteurs internes à la structure OC Pizza : Le vendeur, le pizzaiolo, le livreur, le patron.
- 1 acteur externe : Le client.

Les vendeurs auront une version du logiciel très proche de celle du client, étant donné qu'ils joueront le rôle d'intermédiaire entre le système informatique et le client dans le cas où ce dernier ne passe pas par le site internet afin de passer une commande, mais soit il vient sur place ou il appelle au téléphone.

Les pizzaiolos utiliseront le logiciel pour :

- Saisir dans le système informatique la quantité de chaque élément du stock. Ce sera le stock initial, sur lequel se basera le logiciel pour estimer le nombre de pizzas réalisables, et ce pour chaque type de pizza.
- la mise à jour des commandes passées et en préparation.
- Le suivi en temps réel du stock d'ingrédients. Cette fonctionnalité sera automatisée par le logiciel, mais en liaison avec le pizzaiolo. c'est-à-dire

que dès qu'une commande est passée en état de préparation par le pizzaiolo, le logiciel mettra à jour le stock d'ingrédients, en retranchant le nombre de pizzas commandés du nombre de pizzas réalisables, et ce pour chacun des types de pizzas du menu.

Les livreurs pourront récupérer les détails (pizzas commandées, téléphone, adresse, nom, prénom) afin de livrer les clients. ils mettront à jour l'état des commandes, une fois livrées.

Le client utilisera le système via un site internet afin de passer une commande; le site internet proposera au client un large choix de pizzas.
le client pourra aussi consulter sa commande passée et la modifier ou l'annuler tant qu'elle n'est pas passée en état de préparation.
Il aura la possibilité de payer en ligne, ou en espèces lors de la livraison de sa commande.

Le patron n'est pas explicitement demandé dans le cahier des charges mais son intégration s'avère utile dans le bon fonctionnement du groupe de pizzerias, il pourra accéder au chiffre d'affaire ou à différents indices de satisfaction des clients, tels que à titre d'exemple, le temps total mis pour une commande de sa réservation jusqu'à sa livraison, etc..

B. Acteurs secondaires

Le système bancaire servira pour l'enregistrement d'un règlement par carte bancaire, autant par un client que pour lui à travers un vendeur. Le client pourra faire cela en ligne, mais le vendeur peut également le faire pour le client lors d'une vente sur place.

Acteurs	Acteurs principaux	Acteurs secondaires
Le client	X	
Le vendeur	X	
Le pizzaiolo	X	
Le livreur	X	
Le patron	X	
Le système bancaire		X

VI. Définition des packages

Après une analyse du cahier des charges, le système peut être décomposé en deux packages :

Voir diagramme de packages.

A. Interface Client

Le package **Interface Client** regroupe la partie site web qui permettra au client de commander des pizzas en ligne, de consulter le menu des pizzas, de constituer un panier de commande, de consulter sa commande passée, de la modifier ou de l'annuler, de proposer divers moyens de paiement.

B. Interface pizzeria

Le package **Interface Pizzeria** regroupe la gestion des commandes ainsi que celle du stock d'ingrédients.

Différents acteurs auront accès à cette interface :

Les pizzaiolos auront accès aux commandes passées des clients ainsi qu'au stock.

Le livreur aura accès aux coordonnées des clients grâce aux numéros de commandes.

Le patron pour le contrôle et la gestion administrative.

Acteurs	Interface Pizzeria	Interface Client
Le client		X
Le vendeur		X
Le pizzaiolo	X	
Le livreur	X	
Le patron	X	
Le système bancaire		X

VII. Etude de cas d'utilisations

Cette partie de l'étude recense tous les cas d'utilisations possibles des différents acteurs.

Voir diagramme de cas d'utilisation général

Les acteurs internes à la structure "OC Pizza" (vendeur, pizzaiolo, livreur, patron) devront tous **s'authentifier** afin d'être légitimés à accéder au système. Ensuite, le système attribue à chaque acteur les données d'identité pour cette session. Il y aura du coup une session patron, une session vendeur, une session pizzaiolo, ainsi de suite...

Par rapport au client (acteur externe), s'il passe une commande en ligne, il devra **s'authentifier** (s'inscrire s'il ne l'est pas). S'il passe une commande sur place ou par téléphone, un vendeur lui crée une fiche client avec son nom, son prénom, son numéro de téléphone, son adresse de livraison s'il souhaite se faire livrer.

Le client :

Pour passer une commande, le client aura trois possibilités :

1. **Sur place**, commander et régler en espèces ou par carte bancaire (via un terminal de paiement électronique).
2. **Par téléphone** en communiquant avec **un vendeur**. Le règlement, en espèces ou par carte bancaire (TPE), sur place lors de la récupération de sa commande, ou en espèces à la livraison.
3. Commander **en ligne** via le site internet. Le règlement par carte bancaire ou en espèces à la livraison.

En ligne, Le client pourra choisir les pizzas dans un menu établi sur le site de la pizzeria.

Si le client décide de passer une commande, il devra forcément renseigner son adresse de livraison, soit l'adresse de son domicile enregistrée lors de son inscription au site (pour ne pas devoir la retaper à chaque fois), soit de saisir une nouvelle adresse de livraison. (il peut par exemple souhaiter se faire livrer à son lieu de travail)

Le système va calculer le temps de préparation et proposera au client un temps approximatif de livraison. Si le client valide la proposition, on lui proposera de

régler l'achat via le site internet ou en espèces auprès du livreur lors de la livraison.

Après validation de la commande, le système va générer un bon de commande de numéro de commande unique.

Si la situation l'exige, le client peut revenir à tout moment sur sa commande en indiquant le numéro de commande, afin de :

- Suivre l'état de la commande passée.
- Modifier les choix qui ont été confirmés. (si aucune pizza n'est passée en état de préparation)
- Annuler sa commande. (si aucune pizza n'est passée en état de préparation)

Le vendeur :

- Le vendeur pourra passer des commandes pour les clients, sur place ou par téléphone.
- Au cas où un client demande à modifier ou annuler une commande en attente de préparation, le vendeur, muni du numéro de commande du client, accède à la liste des pizzas en attente de préparation, et réalise son ordre (modification ou annulation).

Seules les pizzas qui sont en attente de préparation peuvent être modifiées ou annulées.

Le pizzaiolo :

- Le pizzaiolo aura la tâche de gestion du stock d'ingrédients, en liaison avec le logiciel.
- Il devra recevoir les commandes passées et les préparer tout en mettant à jour chaque commande pour informer le client de l'état d'avancement de la préparation.
- À tout moment, le pizzaiolo peut consulter un aide-mémoire afin de connaître les ingrédients de chaque pizza et sa recette de préparation.

Le livreur :

- Après finalisation de la préparation d'une commande, le livreur aura accès aux coordonnées du client via le numéro de commande.

- Lors de l'interaction avec le système, le livreur pourra vérifier si la commande a été réglée en ligne ou s'il devrait être réglé en espèces à la livraison.
- Le livreur valide les commandes livrées et en récupère des nouvelles en attente de livraison.

Le patron :

- Après authentification, le patron aura une vision globale de tout le système, il disposera de toutes les permissions, lui permettant de contrôler le bon fonctionnement du système.
- La gestion administrative. (chiffre d'affaires, ...)

Voir diagramme de cas d'utilisation interface client

Voir diagramme de cas d'utilisation interface pizzeria

Acteurs	Fonctionnalités principales	Fonctionnalités internes
Client	<ul style="list-style-type: none"> • Consulter menu pizzas • Passer commande • Consulter commande 	<ul style="list-style-type: none"> • Authentification • Constituer panier • Saisir informations pour livraison • règlement commande <ul style="list-style-type: none"> ○ en espèces ○ par carte bancaire • Modifier commande • Annuler commande
Vendeur	Idem que pour le client	<p>Idem que pour le client, en plus de :</p> <ul style="list-style-type: none"> • Créer fiche client
Pizzaiolo	<ul style="list-style-type: none"> • Préparer commande • Suivi stock • Consulter aide-mémoire 	<ul style="list-style-type: none"> • Authentification • Recevoir commande • Mettre à jour état commande • Saisir stock initial • Estimer nombre pizzas réalisables • Mettre à jour stock ingrédients • Consulter recettes pizzas
Livreur	<ul style="list-style-type: none"> • Préparer livraison 	<ul style="list-style-type: none"> • Authentification • Consulter détails livraison • Mettre à jour état commande
Patron	<ul style="list-style-type: none"> • Supervision - Gestion 	<ul style="list-style-type: none"> • Authentification • Gestion administrative (chiffre d'affaires, etc.)
Système bancaire	<ul style="list-style-type: none"> • Échange avec le système bancaire 	

VIII. Solution technique

Afin de répondre aux besoins d'OC Pizza et de réalisation d'une application répondant aux spécifications fonctionnelles détaillées dans ce document, nous nous proposons de réaliser la solution à l'aide des technologies suivantes :

- L'essentiel de l'application sera programmé via le langage Java, un langage performant et multiplateforme qui semble adapté à la structure du projet.
- L'interface web du logiciel sera réalisée également en Java via des frameworks comme Spring, Apache Struts.
- Enfin, toutes les données de l'application seront stockées dans un système de gestion de base de données relationnel dont la structure sera définie par la suite en collaboration avec OC Pizza.

Le choix de PostgreSQL semble opportun pour la réalisation de cette base de données en raison du volume potentiel de celle-ci et de la structure des données qui seront traitées.

Apache http Server

Le serveur HTTP Apache, communément appelé Apache, est un serveur web multiplateforme libre et open-source.

Il a joué un rôle clé dans la croissance initiale du World Wide Web et est resté le plus populaire depuis avril 1996. En juillet 2016, on estimait qu'il desservait 46 % de tous les sites Web actifs et 43 % des millions de sites Web les plus actifs.

JBoss et *Apache Tomcat* sont tous les deux des serveurs d'applications Java, mais *JBoss* est bien plus que cela. La différence substantielle entre les deux serveurs est que *JBoss* fournit un stack complet Java Enterprise Edition (JEE), incluant les *Enterprise JavaBeans* et plusieurs autres technologies utiles aux développeurs travaillant sur des applications Java. *Apache Tomcat* est beaucoup plus limité. En résumé, *JBoss* est un stack JEE qui inclut un conteneur servlet et un serveur web, alors que *Apache Tomcat* est en soi un conteneur servlet et un serveur web, présentant deux principaux avantages :

- Beaucoup moins de complexité et d'utilisation de ressources.
- Modularité.

Apache Struts

Struts est un framework pour applications web, développé par le projet Jakarta de la fondation Apache. C'est le plus populaire des frameworks pour le développement d'applications web avec Java.

Struts met en oeuvre le modèle MVC 2 basé sur une seule servlet faisant office de contrôleur et des JSP pour l'IHM. L'application de ce modèle permet une séparation en trois parties distinctes de l'interface, des traitements et des données de l'application.

Struts se concentre sur la vue et le contrôleur. L'implémentation du modèle est laissée libre aux développeurs : ils ont le choix d'utiliser des JavaBeans, un outil de mapping objet/relationnel, des EJB ou toute autre solution.

Spring

Spring est un framework très riche permettant de structurer, d'améliorer et de simplifier l'écriture d'applications Java EE.

Spring est un framework libre, un conteneur dit « léger », c'est à dire une infrastructure similaire à un serveur d'application Java EE. Il prend donc en charge la création d'objets et la mise en relation d'objets par l'intermédiaire d'un fichier de configuration. Le gros avantage par rapport aux serveurs d'application est qu'avec SPRING, les classes n'ont pas besoin d'implémenter une quelconque interface pour être prises en charge par le framework (au contraire des serveurs d'applications J2EE et des EJBs). C'est en ce sens que SPRING est qualifié de conteneur "léger".

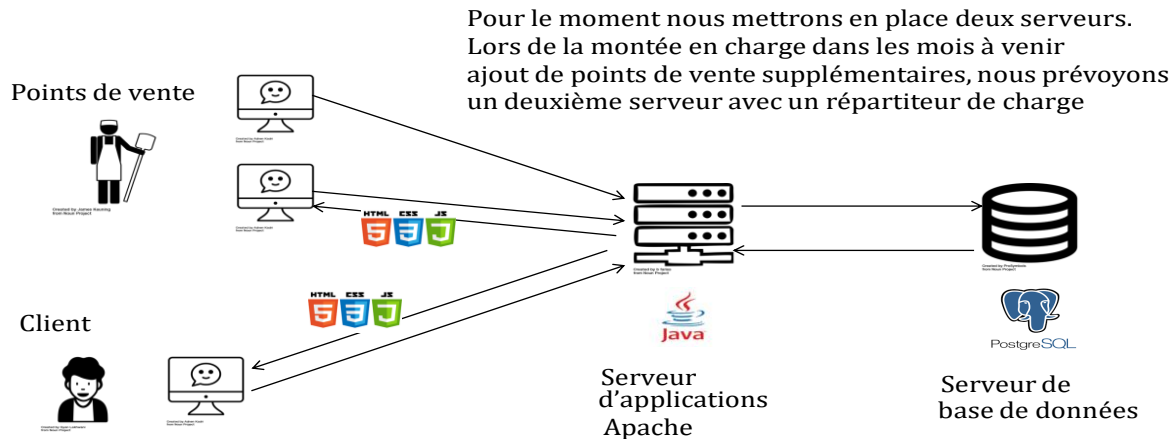
Paypal

PayPal est un service de paiement en ligne qui permet (entre autre) de payer des achats sans avoir à recommuniquer ses coordonnées bancaires, une adresse de courrier électronique et un mot de passe étant suffisants.

Paypal est accepté comme moyen de paiement par plus de 16 millions de commerçants dans le monde. Plus de 203 millions de particuliers se servent de PayPal pour acheter, vendre ou envoyer de l'argent.

Architecture générale

Nous déploierons la solution logicielle sous forme d'une architecture client / serveur.



Choix technologiques



- Java très populaire, largement utilisé.
- Modularité, portabilité et lisibilité.

- Apache Maven, Spring, Apache Struts
- Frameworks qui simplifient le développement de l'application.



- Développement web Front-end
- création de site web : technologie HTML /CSS - Bootstrap – JavaScript



- Base de données PostgreSQL
- comment les données sont enregistrées

Gestion de projet

- Projet Agile / Methode SCRUM
- Découpe du projet en fonctionnalités.
- Intégration du client dans la definition du projet et de la gestion des priorités. Collaboration tout au long du projet.
- Adaptation aux changements
- Intégration continue (local, test, production), livraison sur des cycles courts, ajouts des fonctionnalités au fur et à mesure.
- Test continu