

Analysez les besoins de votre client pour son groupe de pizzerias

Solution fonctionnelle et technique

Table des matières

Contexte :	3
Solution fonctionnelle :	3
Etude du contexte :	3
Rôle factuel des entités :	4
Utilisateurs :	4
Client :	4
Préparateur :	4
Livreur :	5
Entité secondaire :	5
Etude des packages :	5
Package Achat :	5
Package interne :	5
Package Livraison :	5
Etudes des acteurs :	6
L'utilisateur :	6
Le client :	6
Le préparateur :	6
Le livreur :	7
Solution technique :	7

Analysez les besoins de votre client pour son groupe de pizzerias

Contexte :

« OC Pizza » est un jeune groupe de pizzeria en plein essor et spécialisé dans les pizzas livrées ou à emporter. Il compte déjà 5 points de vente et prévoit d'en ouvrir au moins 3 de plus d'ici la fin de l'année. Un des responsables du groupe a pris contact avec vous afin de mettre en place un système informatique, déployé dans toutes ses pizzerias et qui lui permettrait notamment :

- D'être plus efficace dans la gestion des commandes, de leur réception à leur livraison en passant par leur préparation ;
- De suivre en temps réel les commandes passées et en préparation ;
- De suivre en temps réel le stock d'ingrédients restants pour savoir quelles pizzas sont encore réalisables ;
- De 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
- De proposer un aide-mémoire aux pizzaiolos indiquant la recette de chaque pizza

NB : Merci de vous référer au document spécification fonctionnelle pour tout complément.

Solution fonctionnelle :

Suite à la première étude du contexte, le développeur se propose de réaliser l'application pour répondre aux besoins du client.

Dans le but de répondre à ces besoins spécifiques, le développeur liste ci-après les éléments de compréhension du projet, à valider par le client avant réalisation du projet.

Etude du contexte :

Le développeur estime que le rôle des entités à prévoir pour l'application sont pour un état non-connecté :

- Utilisateur non connecté

Le développeur estime que le rôle des entités à prévoir pour l'application sont pour un état connecté :

Analysez les besoins de votre client pour son groupe de pizzerias

- Le client (Manager de la société OC Pizza)
- Les clients du client (dénommés utilisateur pour le reste du cahier de solution fonctionnelle)
- Les préparateurs
- Les livreurs

Rôle factuel des entités :

Utilisateurs (connecté) :

Les utilisateurs utiliseront une application web, disponible sur tout type de terminal, afin de pouvoir consulter le menu du restaurant ainsi que gérer leurs commandes.

Il aura la possibilité de payer en ligne lors de l'enregistrement de sa commande mais également la possibilité de payer à la livraison.

Il lui sera également possible de modifier sa commande si celle-ci n'est pas préparée.

Utilisateur non connecté :

L'utilisateur non connecté aura accès à l'application web. Toutefois, il ne pourra pas effectuer de commande (ou autre action) sans s'authentifier ou s'inscrire.

Client :

Le client, aussi appelé Manager, aura tous les accès pour la lecture et écriture de recettes, lecture et écriture de menu, lecture et écriture de stock, lecture et écriture de commandes. Il pourra également consulter les commandes en cours.

Son rôle est pour l'instant limité au bon fonctionnement de l'entreprise, toutefois il sera possible et préférable pour répondre à ses attentes, de mettre en place des outils permettant de faciliter l'étude de statistiques dégagés par l'application.

Il peut endosser le rôle du client pour écrire une commande reçue par téléphone, il peut également préparer la commande si nécessaire et modifier son statut.

Dans un premier temps, son rôle est de contrôler le bon fonctionnement de l'application pour permettre un développement itératif des fonctionnalités qui lui seront plus spécifiques à terme.

Préparateur :

Le préparateur a une fonction définie par l'entreprise de cuisinier. Il reçoit les commandes, modifie leurs statuts et consulte les recettes spécifiques aux produits demandés par l'utilisateur.

Il peut également voir le stock en cours.

En revanche, son rôle n'est uniquement composé que d'action de lectures. Il ne peut pas modifier une commande ou modifier le stock (à l'exception de l'écriture du statut de la commande), de même qu'il lui est impossible de consulter les informations de l'utilisateur.

Ceci a été prévu par le développeur pour permettre une compartimentalisation des compétences ainsi qu'une meilleure étude des fonctions de son application.

En effet, il serait difficile pour le développeur de proposer une application fiable dont le contenu pourrait être modifié par tout type d'utilisateur.

Analysez les besoins de votre client pour son groupe de pizzerias

Attention : La modification du stock par le préparateur n'est qu'une opération négative. Il ne peut pas modifier ou ajouter volontairement des produits. Le stock se déduit seul des quantités à l'acceptation de la commande.

Livreur :

Le Livreur est l'entité qui possède le moins de fonctions. Il ne possède uniquement que les fonctions nécessaires à son travail. Il peut lire les informations de l'utilisateur et livrer.

Il est prévu par le développeur qu'avec cette solution, le risque de « mauvaise livraison » est ainsi limité.

Entité secondaire :

Le système bancaire est l'unique entité secondaire de l'application. Son rôle se cantonne à vérifier les coordonnées bancaires de l'utilisateur et recevoir le paiement de l'utilisateur.

Etude des packages :

Package Achat :

Le package achat n'est accessible uniquement que par l'utilisateur et le client. En effet, il n'est possible uniquement que d'effectuer des commandes dans cette partie.

Le client a par contre des droits d'administration disponibles sur les menus.

Package interne :

Le package interne est accessible par les préparateurs et le client. Dans ce package, les informations des commandes sont accessibles pour permettre leurs préparations. Les stocks sont également indiqués pour permettre au client de vérifier les stocks en cours et s'assurer de la bonne préparation des commandes. Dans un souci de suivi des stocks, le client sera seul à pouvoir modifier les stocks.

Le package interne concerne également la préparation. Le client a encore une fois les droits « admin » et peut préparer la commande en modifiant son état s'il le souhaite. Un rôle de préparateur est également confié à d'autres employés, ceux-ci se chargeront des commandes et modifieront leurs états avant et après préparation. Ainsi le suivi sera en temps réel pour le client.

Package Livraison :

Le package livraison est isolé. Le livreur ne peut avoir accès qu'aux informations strictement nécessaires dans le cadre de son travail pour des raisons de sécurité.

De plus, le fait d'externaliser complètement le statut du livreur permettra de gérer plus facilement le turn-over conséquent ainsi qu'une possible externalisation du service avec d'autres plateformes telle qu'UberEat ou Kangaroo.

Analysez les besoins de votre client pour son groupe de pizzerias

Etudes des acteurs :

L'utilisateur (connecté) :

L'utilisateur est le client final, c'est pour lui que le développeur développe cette application.

Aussi, il paraît évident que les solutions retenues par le développeur doivent être pratiques pour le client, mais également l'utilisateur final qui l'utilisera quotidiennement pour faire des réservations.

La solution doit donc être autant visuelle que possible et surtout intuitive et penser à une expérience utilisateur conforme aux attentes des utilisateurs.

L'utilisateur aura donc un choix à faire :

- Il peut commander en boutique
- Il peut commander par téléphone
- Il peut commander par internet et suivre l'état de sa commande.

Il devra donc naturellement avoir accès à la lecture du menu pour l'aider à faire son choix. Par la suite, il pourra effectuer sa réservation en précisant son mode de paiement et en renseignant les champs d'adresse pour que sa commande puisse correctement lui être livrée s'il le souhaite. Il peut sinon la retirer directement à la boutique.

Côté système, l'utilisateur sera défini par un numéro de commande correspondant à un id unique de préparation afin de pouvoir correctement l'identifier. Un nom sera également associé à la commande pour limiter le risque de fraude et/ou comportements malveillants.

Une fois sa commande enregistrée, il peut la modifier tant que celle-ci n'est pas encore préparée.

L'utilisateur (non-connecté) :

L'utilisateur non connecté correspond à l'état par défaut d'un visiteur dans l'application. Il pourra être par la suite caractérisé par un rôle (client, préparateur ou livreur) une fois authentifié.

Cet acteur a des actions limitées, il ne peut que consulter le menu et s'authentifier.

Le client :

Le client est le demandeur du projet (ou assimilé).

Il souhaite un outil lui permettant de mieux organiser son travail, d'être plus efficace dans la gestion de ses commandes, de ses stocks, du suivi de ses commandes mais également d'avoir une présence sur l'espace internet afin de faire connaître son commerce à un plus grand nombre.

Il a un rôle de « masterKey » pour la plupart des opérations. Il peut effectuer des opérations de lecture et écriture pour les réservations de commandes, de menu, de stock.

Il pourra par la suite mettre en place des outils statistiques afin de mieux suivre l'évolution de son entreprise.

Le préparateur :

Le préparateur est un exécutant aux ordres du client. Il reçoit les commandes et les prépare. De ce fait, il a un droit de lecture sur la commande et de lecture/écriture sur l'état de la commande.

Sa mission est limitée à la simple préparation des commandes. Il ne modifie pas le stock et ne prend pas de commandes.

Analysez les besoins de votre client pour son groupe de pizzerias

Le livreur :

Le livreur n'a qu'une fonction : livrer. Pour ce faire, il ne peut avoir accès qu'aux informations client pour limiter les risques de fraudes dues au turn-over fréquent pour ce type de poste. Cette spécialisation peut avoir une solution pratique ultérieure pour le client : l'appel d'offre à des sociétés de livraison à domicile telle que UberEat.

Solution technique :

Le développeur retient une application en deux parties :

- Une partie Front-end :

Le Front, ou partie visuelle du site, sera géré en HTML/CSS et javascript avec Angular. L'approche retenue par le développeur consiste en une approche mobile-first, ainsi l'application sera optimisée sur tout type d'écran : smartphones, tablettes et ordinateurs afin de maximiser les parts de marchés.

La bibliothèque Material Design sera utilisée pour les animations et effets de style.

- Une partie Back-end :

La partie Backend sera gérée en Java pour sécuriser l'application et définir les entités. Cette partie, invisible pour le client, sera également la partie en charge d'effectuer les requêtes en SQL dans la base de données afin de pouvoir modifier, lire, écrire ou supprimer des informations (CRUD).

NodeJS est également présent pour permettre le développement du projet (gestion de packages nécessaires à l'application).

L'application sera hébergée avec Docker pour permettre la limitation de l'impact d'une panne, ou d'une attaque : il suffira en effet de redémarrer le conteneur.

Angular est un Framework JavaScript. Il paraît pertinent au développeur de l'utiliser pour une raison pratique : la gestion de l'application se fera avec des routes et non pas des liens. L'utilisateur aura donc un site épuré qui aura l'impression d'une SPA (Single Page Application). Ce Framework est développé et maintenu par Google.

Material Design est également une bibliothèque que Google a mise en place pour les développeurs.

La gestion de projet se fera avec SCRUM, une méthode agile pour faciliter la validation client et s'assurer d'un produit au plus proche de ses besoins et de ses évolutions futures. Il sera profitable au développeur également afin de comprendre le métier pour lui apporter des solutions plus concrètes et personnalisées.

Enfin, la librairie Socket.io sera utilisée afin de permettre une application en temps « réel » ou avec une latence minimale et sans avoir besoin de rafraîchir l'application. Cette librairie sera surtout utile côté pizzeria afin de limiter la demande de données sur la bande passante lors de l'arrivée d'une nouvelle livraison.

Le projet sera également versionné sur Github afin de suivre une évolution probable du projet.