

SPÉCIFICATIONS TECHNIQUES

OC Pizza



Open Classrooms
Auteur : Mickaël HORN – Développeur Junior

Table des matières

<i>Contexte du projet</i>	3
<i>Comparatif des méthodes</i>	4
<i>Choix de la méthode</i>	5
<i>Macroplanning</i>	6
<i>RACI (Responsibility Assignment Matrix).....</i>	8
<i>Diagramme de classe (MCD)</i>	10
<i>Modèle Logique de Données (MLD).....</i>	12
<i>Modèle Physique de Données (MPD).....</i>	13

Contexte du projet

OC Pizza est un groupe récent de pizzeria qui connaît un franc succès et se développe rapidement.

Fondé par Franck et Lola, OC Pizza est spécialisé dans les livraisons et dans les pizzas à emporter.

Aujourd'hui, le groupe compte 5 points de ventes et a pour projet d'en ouvrir 3 de plus d'ici 6 mois.

OC Pizza cherche à changer de système informatique car il ne correspond plus aux besoins du groupe.

En effet, le système actuel ne permet pas de gérer de manière centralisée et individuelle les différentes pizzerias.

Les responsables ne peuvent pas suivre correctement ce qui se passe dans les points de ventes et les livreurs n'ont pas la possibilité d'indiquer en temps réel que la livraison a bien été effectuée.

Voici une liste exhaustive des besoins du groupe :

- Améliorer la gestion des commandes en passant par le cycle de vie complet, c'est-à-dire la réception, la préparation et pour finir, la livraison
- Avoir un suivi en temps réel des commandes en préparation et en livraison
- Disposer du stock en direct pour savoir quelles pizzas peuvent être réalisées ou non
- Mise en place d'un site Internet
 - Passer des commandes
 - Payer en ligne
 - Modifier ou annuler la commande si elle n'a pas encore été préparée
- Aider les pizzaiolos en leur proposant un aide-mémoire qui contiendra la recette de chaque pizza
- Recevoir les notifications quand la commande est prête à être livrée ou réceptionnée.

Comparatif des méthodes

Avant de commencer, il nous faut choisir une méthodologie pour notre projet.

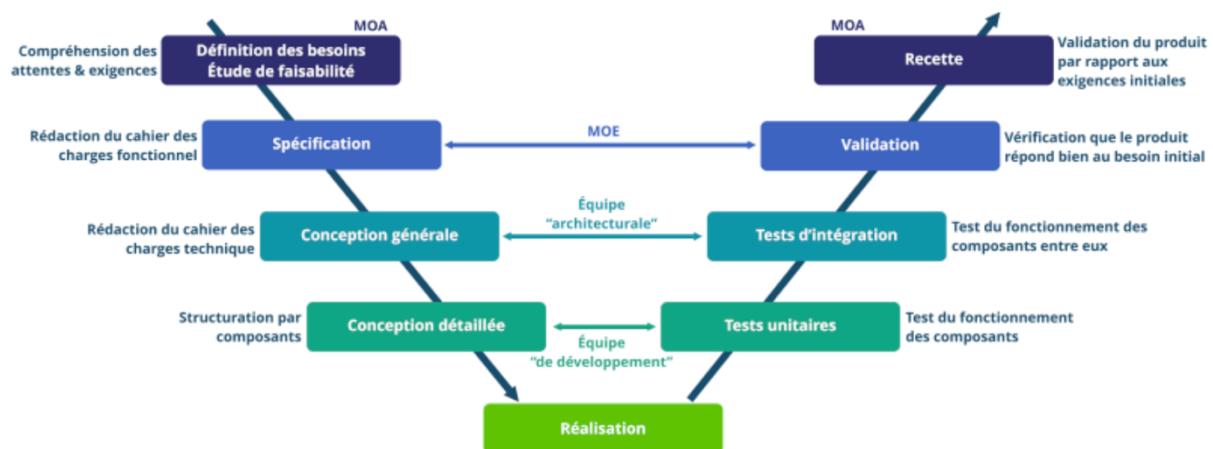
Voici un comparatif des différentes méthodologies à disposition qui aidera à choisir laquelle est la plus concrète pour notre situation.

	Méthode Agile	Méthode en Cascade	Cycle en V
Principes clés	L'équipe, soit des individus et des interactions, plutôt que des processus et des outils L'application, c'est-à-dire des fonctionnalités opérationnelles plutôt que de la documentation exhaustive La collaboration avec le client, plutôt que la contractualisation des relations L'acceptation du changement, plutôt que le suivi d'un plan	La livraison des livrables se fait à une date précise et est définie lors du cadrage du projet Une phase ne peut commencer que si la précédente est terminée La production des livrables définis au tout début du projet	Chaque partie de la phase ascendante fait écho à chacune de la partie descendante (la phase ascendante est là pour valider la phase descendante)
Avantages	Flexibilité Collaboration et communication fréquente avec le client Meilleure visibilité du client sur le projet Le client peut changer de direction s'il le souhaite Maîtrise des coûts	Une documentation solide (étapes clairement définies) Estimation des coûts dès le début du projet Une structure ou les phases de projet sont clairement délimitées Evite les aller-retours incessants, comme tout est défini au préalable Plus grande simplicité (processus continu) que dans le cycle en V (simultané)	Elaboration d'un plan de tests dès le début du projet Qualité et fiabilité maximisés, risques minimisés, grâce aux nombreux tests
Inconvénients	Documentation maigre car le dialogue est privilégié Les clients doivent rester disponibles Pas adapté pour les entreprises aux structures hiérarchiques très fortes, à cause du fonctionnement collaboratif Malgré un bon contrôle des coûts, la vision du budget pour la totalité du projet est difficile	Une bonne documentation peut également être lourd et fastidieux Si une des phases prend du retard, les prochaines aussi De par sa construction séquentielle et linéaire, le retour en arrière est difficile Les tests arrivent tardivement	Le processus étant plus complexe que la méthode en cascade, le cycle en V est plus coûteux
Projets à proscrire	Les projets parfaitement bien cadrés dont l'échéance est bien définie, qui ne nécessite pas de personnalisations	Ne pas appliquer la méthode en cascade ou le cycle en V dans le cas des projets où le client fait des changements réguliers	

Choix de la méthode

Après avoir pris en compte les différences parmi les méthodologies de projet citées plus haut, nous devons prendre une décision sur quelle méthode appliquer.

Le cycle en V semble adapté à notre cas, et nous allons détailler pourquoi.



Les étapes du cycle en V

Le client a clairement exprimé ses besoins, sous forme de liste, que nous avons préalablement étayés dans les spécifications fonctionnelles, ce qui est déjà très rassurant pour la réalisation de ce projet avec une méthodologie comme le cycle en V.

Il semble savoir où il va et on peut donc en déduire qu'il ne devrait pas y avoir de modifications importantes qui pourraient compromettre notre méthodologie séquentielle et linéaire.

Nous pouvons nous passer de la méthodologie agile, car nous n'avons pas besoin d'avancer « à petit pas » étant donné qu'on a affaire à un projet bien cadré.

La méthodologie en cascade est plus simple, mais le cycle en V, du fait de ses tests complets, nous assurera une livraison de qualité.

Pour terminer, je m'appuie sur un élément concret qui est survenu lors de la construction des spécifications fonctionnelles qui est la date de livraison.

En effet, le client souhaite que le projet soit livré pour l'ouverture des 3 nouvelles pizzéries, c'est-à-dire dans 6 mois.

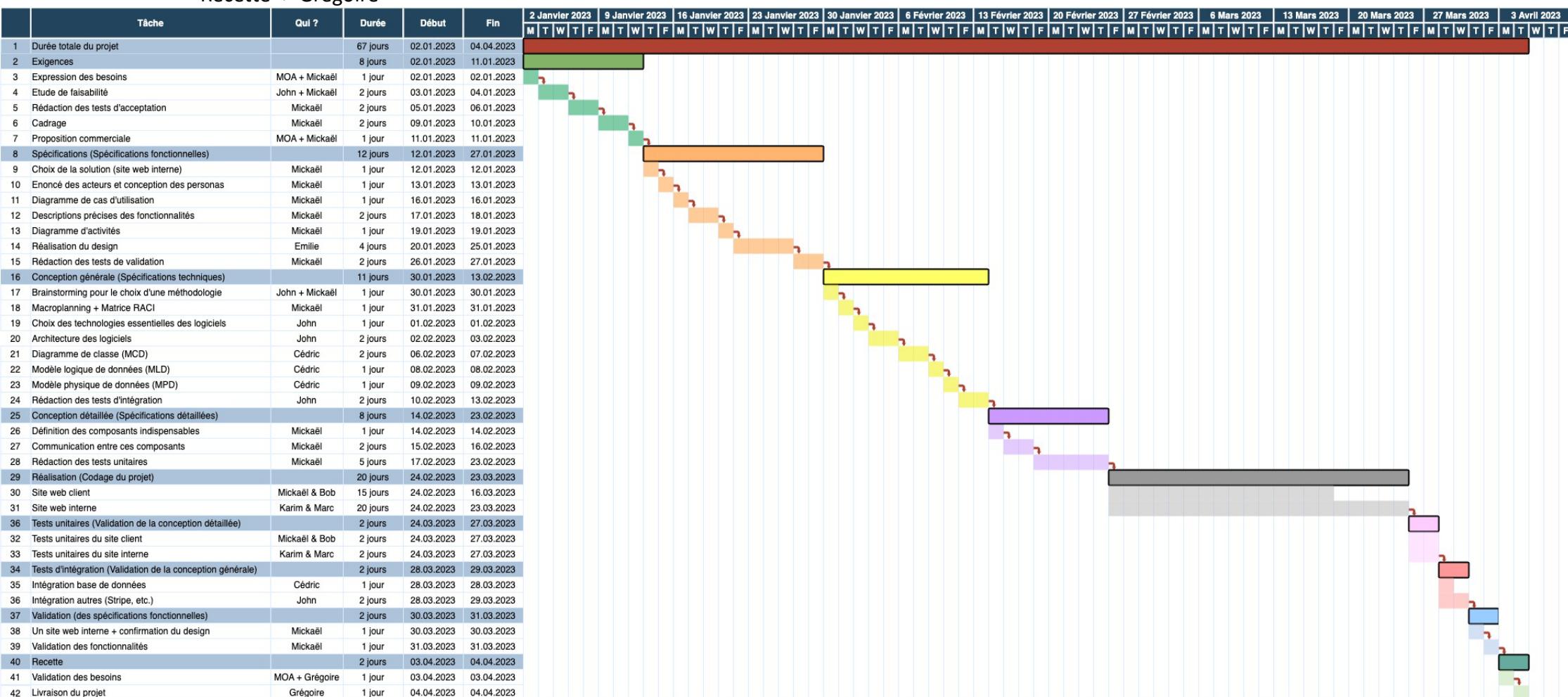
Le client a déjà son idée sur la date de livraison et nous devons lui fournir une vraie date. C'est un des plus gros défauts de la méthodologie agile, ce manque de visibilité sur le temps, on n'a pas d'idée précise du temps et du coût final.

Macroplanning

Voici maintenant un macroplanning sous forme de diagramme de Gantt nous permettant de représenter le projet de manière temporelle.

Équipe :

- Développement -> Mickaël (Lead Developer), Bob, Karim et Marc (Développeurs)
- MOA -> Client
- Équipe Architecturale -> John (Chef De Projet Architecture) et Cédric (Spécialiste base de données)
- Équipe Studio -> Émilie (Design)
- Recette -> Grégoire



Ce diagramme de Gantt reprend de manière logique les différentes étapes de la méthodologie que nous avons choisie.

Comme le cycle en V est une méthodologie de projet séquentielle et linéaire, les étapes s'enchaînent les unes après les autres et non en même temps.

Les tâches, quant à elles, peuvent être réalisées en parallèles, à la condition qu'elles viennent de la même étape.

Par exemple, on constate dans la partie « Réalisation (Codage du projet) », que les deux tâches (le site web côté client et le site web interne) qui la compose sont réalisées en même temps, par deux personnes différentes.

Les week-ends ne sont pas comptés, uniquement les jours ouvrables.

RACI (Responsibility Assignment Matrix)

Intéressons-nous maintenant à la matrice RACI, qui nous offre la possibilité de voir qui fait quoi dans les différentes tâches à accomplir.

R : Responsible

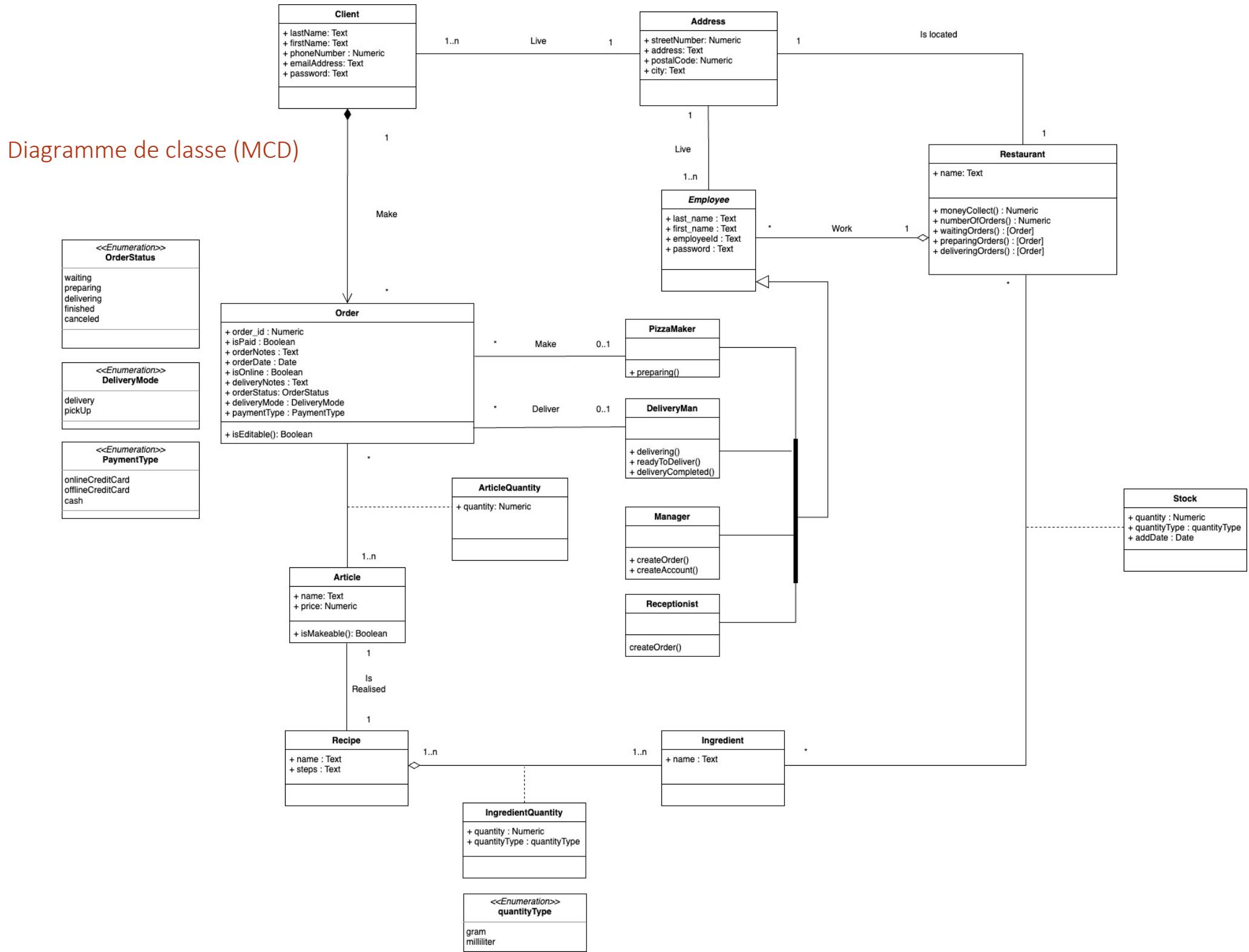
A : Accountable

C : Consulted

I : Informed

	Mickaël	Marc	Bob	Karim	Client	John	Cédric	Emilie	Grégoire
Exigences									
Expression des besoins	A/R	I	I	I	A	I	I	I	I
Etude de faisabilité	A/C/R	-	-	-	-	A/C	-	-	-
Rédaction des tests d'acceptation	A/R	I	I	I	I	I	I	I	I
Cadrage	A/R	I	I	I	-	I	I	I	I
Proposition commerciale	A/R	-	-	-	A	-	-	-	I
Spécifications (Spécifications fonctionnelles)									
Choix de la solution (site web interne)	A/R	C/I	C/I	C/I	-	C/I	C/I	I	I
Enoncé des acteurs et conception des personas	A/R	I	I	I	-	I	I	I	I
Diagramme de cas d'utilisation	A/R	I	I	I	-	I	I	I	I
Descriptions précises des fonctionnalités	A/R	C/I	C/I	C/I	-	C/I	C/I	I	I
Diagramme d'activités	A/R	I	I	I	-	I	I	I	I
Réalisation du design	I	I	I	I	-	I	I	A/R	I
Rédaction des tests de validation	A/R	I	I	I	-	I	I	-	I
Conception générale (Spécifications techniques)									
Brainstorming pour le choix d'une méthodologie	A/R	I	I	I	-	A	I	-	I
Macroplanning + Matrice RACI	A/R	I	I	I	-	I	I	-	I
Choix des technologies essentielles des logiciels	I	I	I	I	-	A/R	I	-	I
Architecture des logiciels	I	I	I	I	-	A/R	I	-	I
Diagramme de classe (MCD)	I	I	I	I	-	I	A/R	-	I
Modèle logique de données (MLD)	I	I	I	I	-	I	A/R	-	I
Modèle physique de données (MPD)	I	I	I	I	-	I	A/R	-	I
Rédaction des tests d'intégration	I	I	I	I	-	A/R	A	-	I

	Mickaël	Marc	Bob	Karim	Client	John	Cédric	Emilie	Grégoire
Conception détaillée (Spécifications détaillées)									
Définition des composants indispensables	A/R	C/I	C/I	C/I	-	-	-	-	I
Communication entre ces composants	A/R	C/I	C/I	C/I	-	-	-	-	I
Rédaction des tests unitaires	A/R	C/I	C/I	C/I	-	-	-	-	I
Réalisation (Codage du projet)									
Site web client	A/R	C	A	C	-	-	-	-	I
Site web interne	C/I/R	A	C	A	-	-	-	-	I
Tests unitaires (Validation de la conception détaillée)									
Tests unitaires du site client	A/R	C	A	C	-	-	-	-	I
Tests unitaires du site interne	C/I/R	A	C	A	-	-	I	-	I
Tests d'intégration (Validation de la conception générale)									
Intégration base de données	I	-	-	-	-	I	A/R	-	I
Intégration autres (Stripe, etc.)	I	-	-	-	-	A/R	-	-	I
Validation (des spécifications fonctionnelles)									
Un site web interne + confirmation du design	A/R	I	-	I	-	-	-	-	I
Validation des fonctionnalités	A/R	I	I	I	-	-	-	-	I
Recette									
Validation des besoins	I	-	-	-	A	-	-	-	A/R
Livraison du projet	I	-	-	-	I	-	-	-	A/R



Dé cortiquons un peu les spécificités de ce diagramme de classe.

Client et Order :

Nous remarquons qu'il y a une relation de composition entre la table Client et Order. En effet, on considère qu'une commande ne peut exister sans un client, d'où la composition côté Client.

Employee et Restaurant :

Ici, nous avons une agrégation qui se justifie par le fait qu'un restaurant est constitué d'un ensemble d'employés.

Restaurant, Ingredient et Stock :

La table Stock est née du fait de la classe d'association entre Restaurant et Ingredient. De cette façon, chaque ingrédient aura son restaurant associé ainsi que ses informations telles que la quantité, le type de quantité et la date d'ajout.

Recipe, Ingredient et IngredientQuantity :

Même logique que pour le stock, nous avons de nouveau une classe d'association plusieurs à plusieurs mais cette fois ci, pour savoir la quantité d'ingrédient de nos recettes.

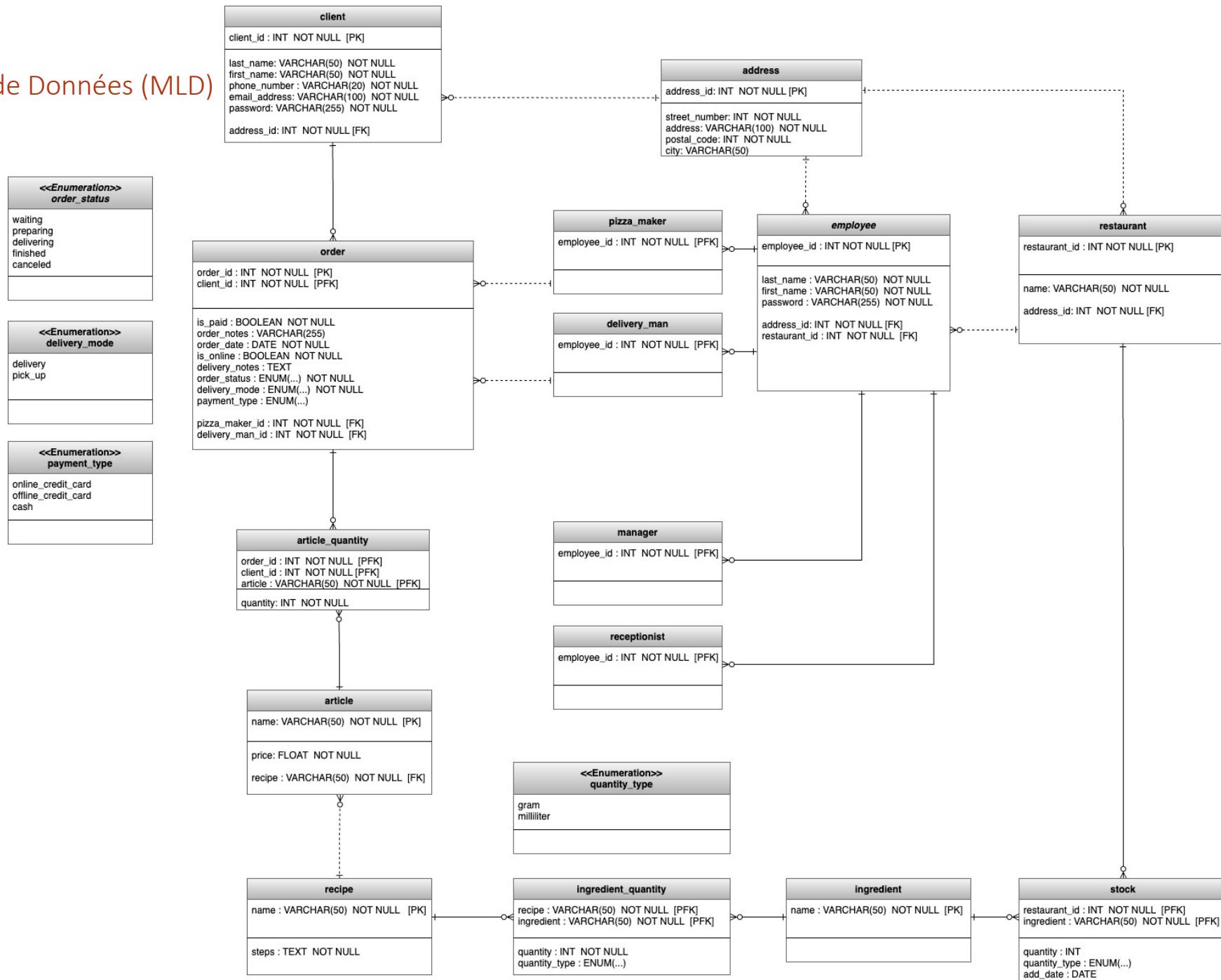
Order, Article et ArticleQuantity :

Un client peut commander plusieurs fois la même pizza et on aura cette information en ayant la commande associée, ainsi que l'article concerné.
D'où la classe d'association ArticleQuantity.

DeliveryMode et PaymentType :

Grace à ces deux énumérations, on va pouvoir connaître tous les détails concernant le mode de livraison ainsi que du type de paiement.
On couvre tous les différents cas, ce qui nous permettra d'avoir un historique de commandes précis.

Modèle Logique de Données (MLD)



Modèle Physique de Données (MPD)

Pour terminer, intéressons-nous maintenant au Modèle Physique de Données. La base de données a été créée avec l'aide de l'outil MySQLWorkbench.

[Vous trouverez trois fichiers dans l'arborescence à l'intérieur du dossier MPD :](#)

1. **horn_mickael_3_mpd_062022.mwb** : correspond à la représentation de la base de données sur l'outil MySQLWorkbench
2. **horn_mickael_5_create_database_062022.sql** : le fichier à lancer afin de créer la base de données
3. **horn_mickael_6_jeu_de donnees_062022.sql** : enfin, le fichier contenant notre jeu de données