



Projet SE
Analyse du domaine /
modélisation



SIMPLIFY

Par Emma Mathieu, Salim Belkhir, Thomas Fournier et Ayoub Hakemi.

Sommaire

Sommaire	2
1-Contexte	3
2-Organisation	3
3-Problèmes rencontrés	3
4-Solutions apportées	5
5-Remarques	6
5.1 Remarques positives	6
5.2 Remarques négatives	6
6-Résultat	7

1-Contexte

La deuxième étape du projet consiste à réaliser l'analyse du domaine à travers un diagramme de classes globales montrant les relations à travers les différentes classes. Cette étape a duré 6 jours et nous avons dû collaborer ensemble contrairement à l'étape précédente. Le responsable de cette étape était Emma Mathieu.

2-Organisation

Pour cette phase, nous avons travaillé sur draw.io pour réaliser l'analyse du domaine pour modifier simultanément le diagramme en s'aidant des uses-cases précédemment réalisés.

Nous avons réalisé deux réunions pour réaliser le travail ensemble.

De plus, nous avons utilisé Notion pour faire un suivi de notre projet et avoir accès aux différents documents du projet.

3-Problèmes rencontrés

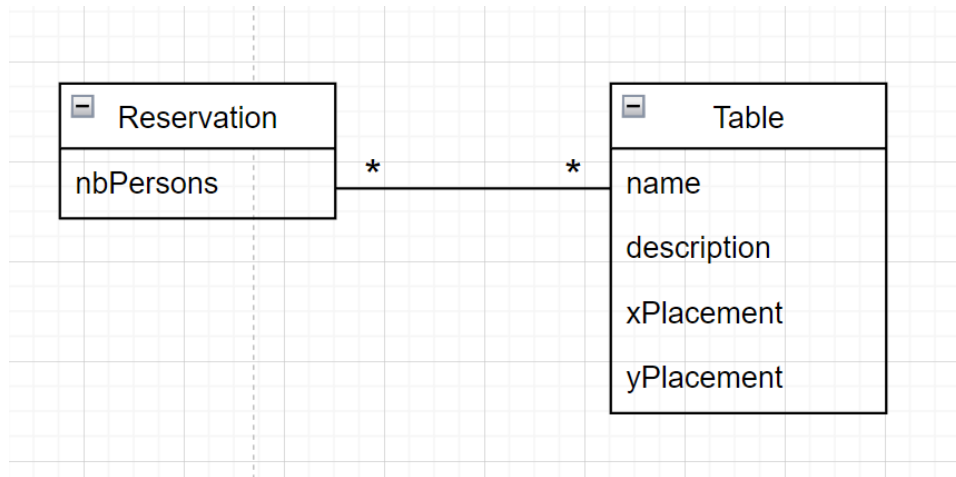
Durant cette phase d'analyse, nous avons rencontré plusieurs difficultés.

Tout d'abord, nous avons eu du mal à fixer clairement les objectifs de cette phase.

Il nous a fallu du temps pour comprendre ce qu'il fallait ou ne fallait pas représenter dans le diagramme de classes.

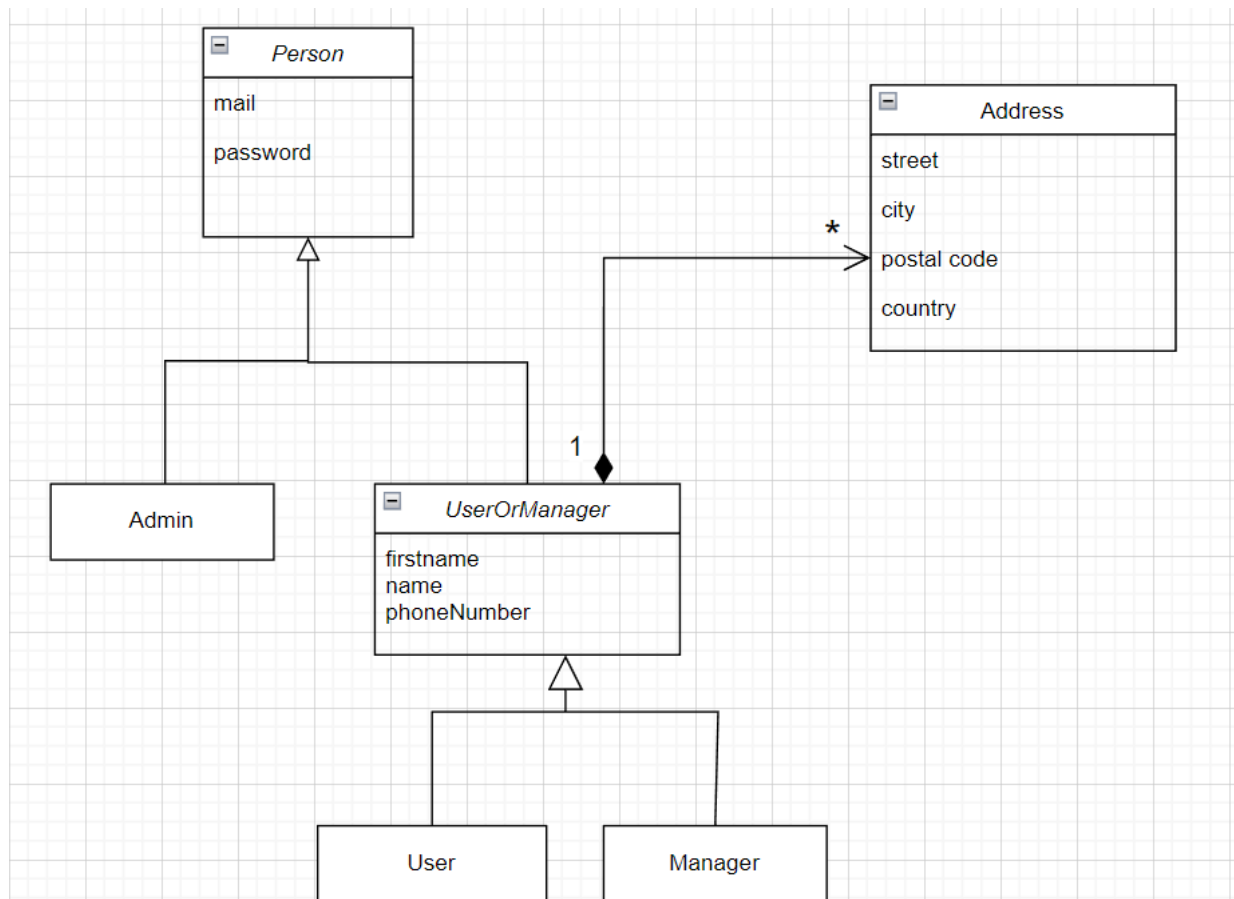
De plus, nous avons eu des problèmes quant à la syntaxe à utiliser. En effet, fallait-il afficher les méthodes des classes, insister sur le type des attributs ou mettre des énumérations ? Nous avons aussi eu des questions au sujet de la multiplicité de chaque association (Cardinalités) et du type de lien entre chaque classe (agrégation ou composition...).

Par exemple, au début de cette étape, nous avons réalisé ce sous-diagramme représentant le lien entre une réservation et une table :



Comme on peut le voir ci-dessus, une réservation peut contenir plusieurs tables. Néanmoins, à partir d'une table, il n'est pas possible de visualiser à quelle réservation elle appartient même si celle-ci peut être contenue dans plusieurs réservations. L'association à double sens est donc invalide.

Par ailleurs, nous avons discuté de la modélisation de certaines classes comme par exemple celle qui concerne les adresses pour un utilisateur. En effet, il est possible qu'un manager ou un utilisateur ait plusieurs adresses mais elle doit être unique à chaque utilisateur. Cela nous permet d'avoir une ouverture sur un design pattern pour représenter cela dans une autre phase du projet.



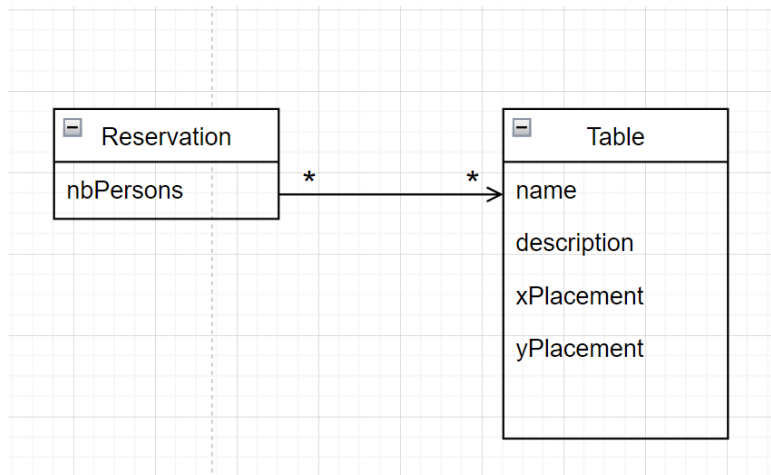
Enfin, nous avons eu plus de mal à nous organiser par rapport à la phase précédente à cause des indisponibilités de plusieurs membres du groupe mais aussi du délai plus court alloué pour cette étape. De plus, notre planning était assez chargé dû aux nombreux cours que nous avons et la semaine internationale.

4-Solutions apportées

Pour résoudre les problèmes cités précédemment, tout d'abord, nous avons décidé de bien revoir le cours (notamment les exemples) sur les différents diagrammes de classes ce qui nous a permis de supprimer des éléments inutiles et de rendre plus lisible. Nous avons essayé de suivre un maximum les étapes du processus même si parfois nous avons commis des erreurs.

Ensuite, nous avons organisé de nombreuses "vérifications" où chacun a expliqué son point de vue au niveau de la syntaxe, de la multiplicité...

Au niveau du problème sur le sens des associations, nous avons fait plusieurs modifications comme montrées ci-dessous :



Sur ce diagramme, on voit qu'il est possible pour une réservation d'avoir accès à une liste de tables mais l'inverse n'est pas possible.

Pour finir, afin d'améliorer notre organisation pour respecter les délais, nous avons pris la décision de commencer l'étape assez tôt. Nous avons aussi dû nous séparer individuellement pour procéder à une première exploration du domaine, pour que chacun puisse exprimer ses idées. Puis nous nous sommes réunis pour assembler les pensées et pour construire le modèle final.

5-Remarques

Au cours de cette analyse, nous avons eu plusieurs remarques positives et négatives.

5.1 Remarques positives

Nous avons pu comprendre l'intérêt de ce diagramme d'analyse, à savoir, ne pas s'intéresser au code (interfaces, méthodes...). Cela nous a permis de rendre plus claire les associations entre les classes pour se concentrer dessus plus tard.

Par ailleurs, cela a facilité la visualisation au niveau des responsabilités des classes pour un futur développement.

5.2 Remarques négatives

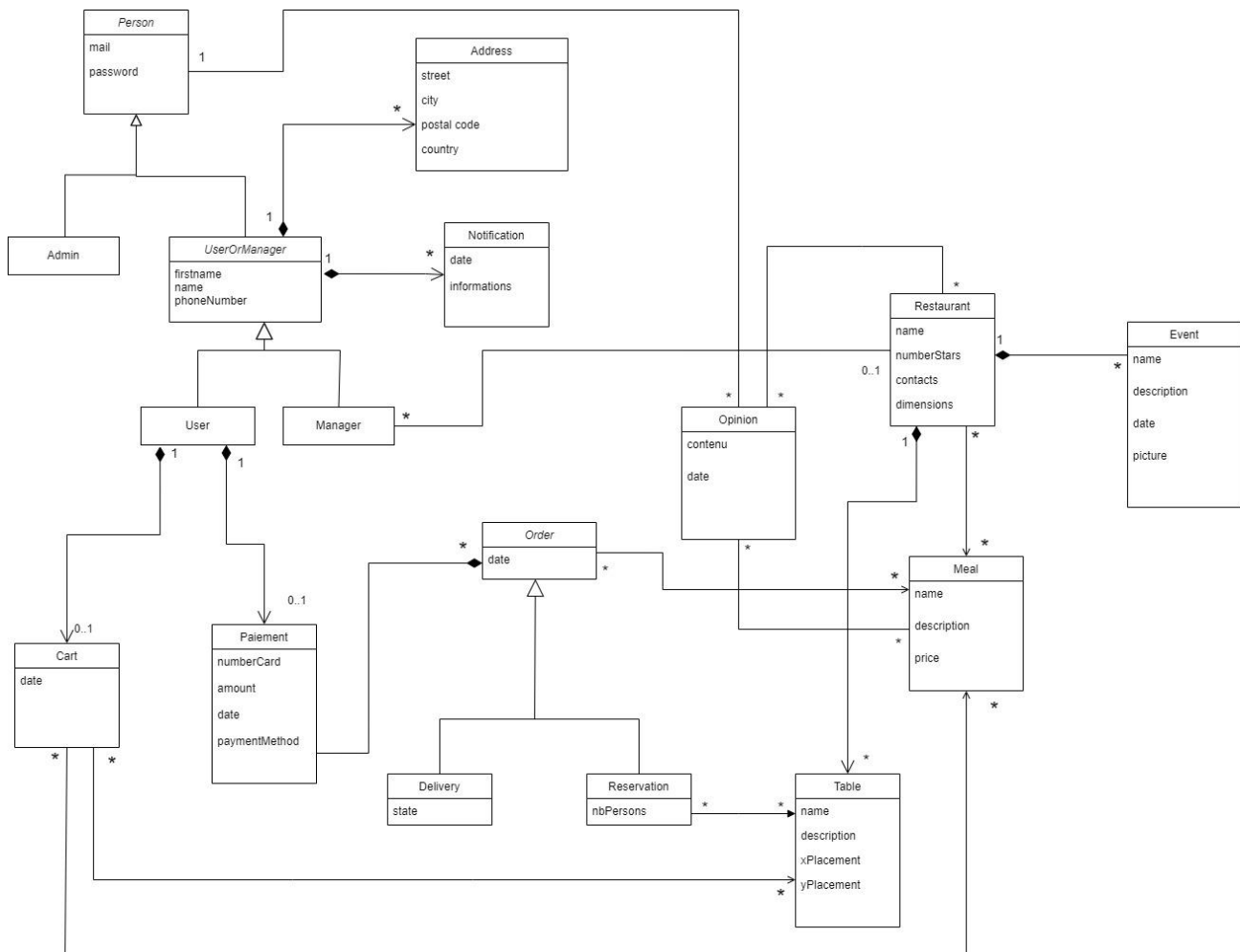
Nous avons, au départ, essayé de prendre un outil différent (Lucid) pour réaliser le diagramme de classes de manière complète (avec les types d'attributs, leurs accessibilités ou encore les méthodes) et cela a rendu plus difficile la modélisation des associations.

Plus tard, nous nous sommes rendu compte que cela était inutile. Notre ancien outil était très bien par rapport aux exigences. Pour les phases futures, nous allons opter

pour un langage de modélisation unifié pour générer automatiquement un diagramme UML sur PlantUML ou un IDE.

6-Résultat

À l'issue de notre travail collaboratif, on a réussi à atteindre notre objectif. Voici ci-dessous, notre diagramme d'analyse du domaine :



Lien du diagramme :

<https://app.diagrams.net/#G1I5hx3eGvh4rJ8j9nEEVU5kcu5dGYMCnn>

Sur ce diagramme on voit bien les comportements des classes vis-à-vis des autres. Nous avons pu clarifier tout ce qui se rapporte à la gestion du panier, du paiement et de la commande.