Dossier Architecture logicielle : BDD & UML

JOUVENÇON Etienne
BERNARD Yoann
HIS Julien
MESCHIN Jean-Marc

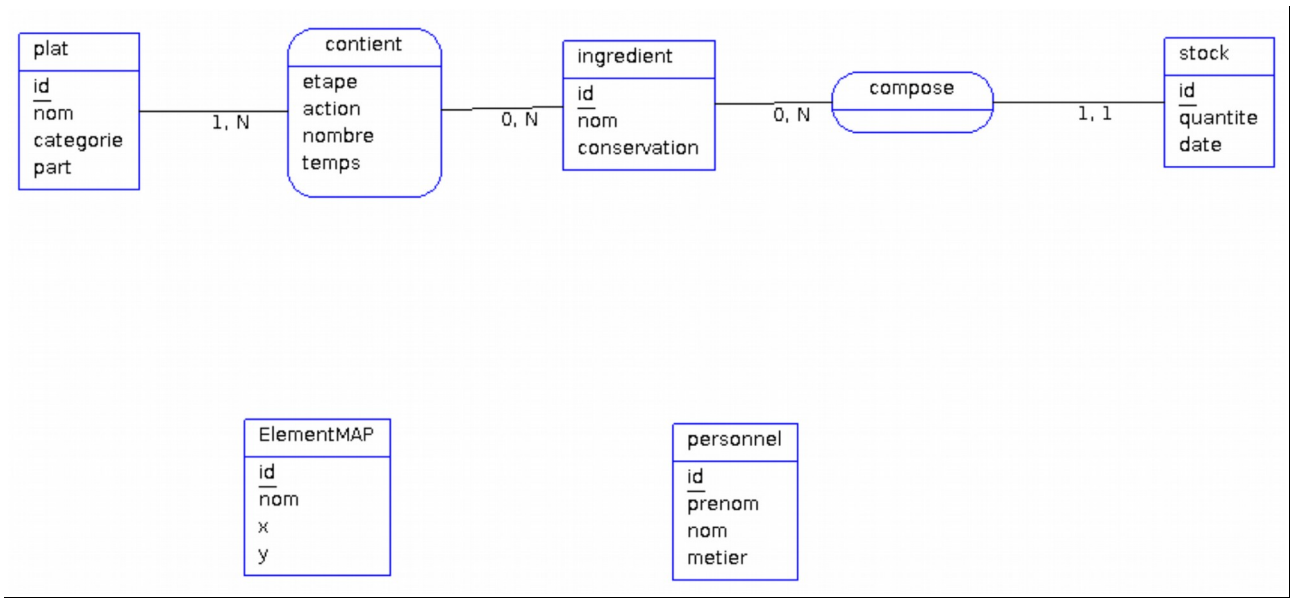
Modèle conceptuel de données

Nous avons choisi de mettre en place une base de données simple et efficace, en partie non relationnelle, censée permettre à notre futur programme de lire et d'écrire les données persistantes :

- Nos recettes ainsi que les ingrédients qui les composent
- Les ustensiles de notre cuisine
- Les éléments et leurs coordonnées (lave-vaisselle, tables, etc . . .)
- Le personnel et leurs métiers.

Deplus, la BDD permettra de modifier facilement le comportement de notre programme sans pour autant modifier notre code source.

(Ajout de personnel, agencement des tables et autres éléments, etc . . .)



Annexe :

MCD.png

MPD.png

script.sql

UML

Vous pouvez retrouver notre diagramme de classes : UML projet.png
Ainsi que le diagramme de cas d'utilisation : Cas utilisation client.png
Ainsi que le diagramme d'activité : Diagramme d'activité.png

Note sur l'utilisation des Design Patterns :

Nous avons implémenté un design pattern Modèle Vue Controlleur.

Dans le controller nous ajoutons une Factory, qui va venir construire les classes que nous allons utiliser. Cela nous permet de créer toutes les classes utilisées pour notre gui avec une simple instruction.

Ces objets possèdent le design pattern Singleton pour limiter le nombre de leurs instances. Cela permet par exemple de limiter le nombre d'objet Restaurant à1.
Le Design Pattern Factory permet aussi notamment d'effectuer un Singleton natif.

Ils communiquent ensuite avec un Builder qui va les composer avec d'autres classes. Notamment des Strategy, pour construire toutes les methodes de nos classes créées par la factory.

Notamment des méthodes grâce à des classes Strategy, des stratégies définissant des méthodes s'attribuant à différentes classes avant même qu'elles soient créées.

Tout cela sera lancé par une classe servant de Main et contenant un Observeur, regardant les clients arrivant dans le restaurant, pour créer les classes adéquates et en lancer les bonnes méthodes.