



Rapport projet semestriel  
*Bases de données et interopérabilité*

## Table des matières

Structure du programme.....	1
Partie Core .....	1
Structure générale.....	1
Gestion d'une commande .....	1
Partie GUI .....	2
Interfaces .....	2
Les conteneurs et les listes.....	2
Améliorations Apportées .....	3
Modifications apportées au Schéma E/A : .....	3

## Structure du programme

La solution se divise en 2 projets, le projet Core qui contient le code avec les requêtes SQL et toutes les interactions avec nos objets, et le projet GUI qui contient toutes les fonctions WPF.

### Partie Core

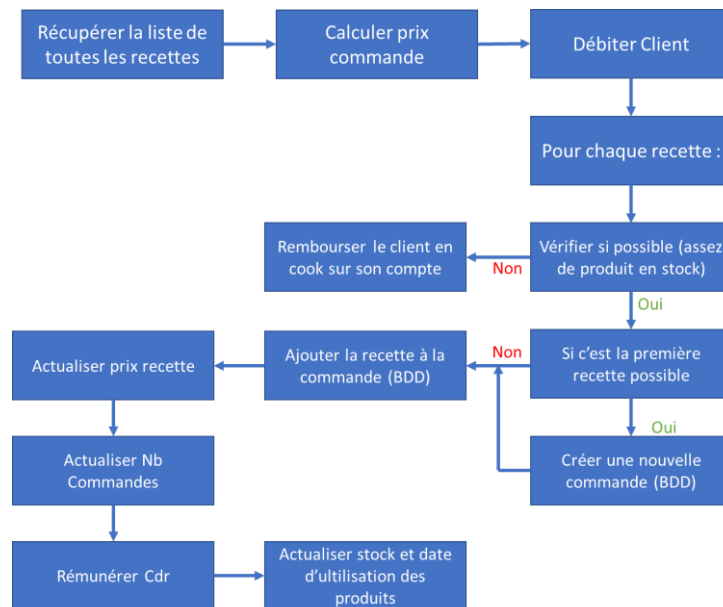
#### Structure générale

Dans le projet Core, toutes nos classes contiennent des fonctions statiques afin de trier les actions en fonction de leur Utilisateur ou des données avec lesquelles elles interagissent (Produit, Recette, Commande ...). La seule classe contenant des attributs est la classe Produit afin de pouvoir utiliser un Sérialiser.

#### Gestion d'une commande

Du point de vue algorithmique, la partie la plus dure était la partie gérant le paiement et la validation des commandes.

Voici l'organigramme de la solution retenue :



## Partie GUI

### Interfaces


Les interfaces n'ayant pas toutes été réalisées en même temps la qualité des codes (ré-usabilité, compréhension structure) peut varier. Si vous voulez voir notre progression, la première fenêtre à avoir été réalisée est la fenêtre Client et la dernière la fenêtre d'administration. Afin de faciliter l'implémentation des fenêtres et de rendre leurs codes plus lisibles, nous avons créé des objets en charge d'afficher des parties de l'interface. Par exemple, l'interface d'un CDR est créée par un objet interface CDR qui prend en paramètre un ID client et qui génère l'interface CDR correspondante.

### Les conteneurs et les listes

Afin de faciliter l'intégration de certaines interfaces, nous avons créé un système de listes contenant un ou plusieurs objets dans leur conteneur.



Ainsi, il suffit de rajouter dans la grille de la fenêtre, la grille ou le Scrollviewer de la liste créée. Avec 4 conteneurs et listes il est possible de réaliser tous les affichages de notre solution. Enfin, en fonction du contexte ceux-ci ont des constructeurs différents (le conteneur à recette d'un administrateur est différent de celui d'un client).




Hamburger

Plat

Un bon hamburger à base de tomme fermière de steak et de bacon


Numéro Recette : 1

Total commandes : 18

Prix : 17 

Supprimer

*Vue administrateur*




Hamburger

Plat

Un bon hamburger à base de tomme fermière de steak et de bacon


Numéro Recette : 1

Total commandes : 18

Prix : 17 

- 0 +

*Vue commande en cours*



Hamburger

Plat

Un bon hamburger à base de tomme fermière de steak et de bacon

Numéro Recette : 1

Total commandes : 18

4

*Vue commande passée*

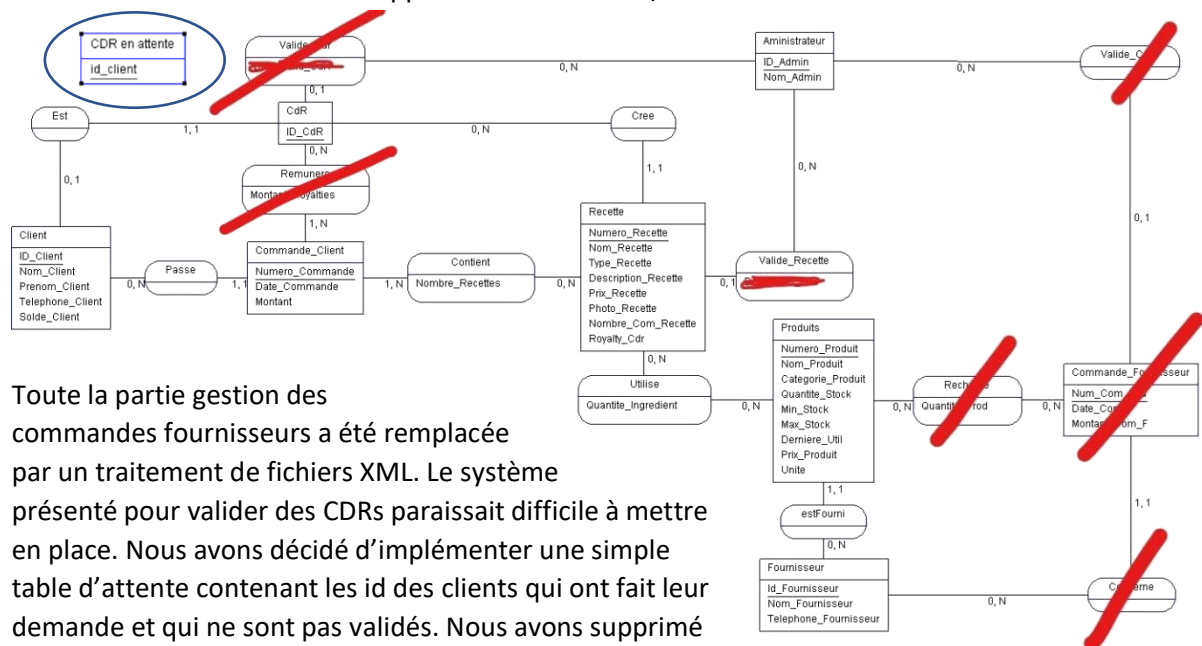
Tout comme pour les fenêtres, les conteneurs n'ont pas été réalisés au même moment et donc là aussi la qualité du code peut varier. Le premier réalisé est le Recipe Container et le dernier le User Container.

## Améliorations Apportées

- Hachage des mots de passe
- Système de validation des CDR
- Système de validation des commandes
- Gestion des commandes produits : pour recharger des produits il suffit d'importer un fichier XML de commande

## Modifications apportées au Schéma E/A :

Voici la liste des modifications apportées au schéma E/A :



Toute la partie gestion des commandes fournisseurs a été remplacée par un traitement de fichiers XML. Le système présenté pour valider des CDRs paraissait difficile à mettre en place. Nous avons décidé d'implémenter une simple table d'attente contenant les id des clients qui ont fait leur demande et qui ne sont pas validés. Nous avons supprimé la date de validation des recettes. A part cela nous n'avons rien changé dans le schéma E/A présenté au TD6.