

# Rapport Final Projet

LABOURG Priscilla NLATE-DZOU Claude

May 17, 2018

# 1 Introduction

Toutes les cuisines de restaurant même les plus petites disposent aujourd'hui d'équipement informatique permettant de stocker de nombreuses données et de les traiter rapidement mais toutes ne disposent pas de logiciel de gestion de stock.

Sachant que la gestion des stocks implique principalement l'ajout, la suppression et/ou la modification de produit dans les stocks, nous avons décidé de nous approprier le sujet en ajoutant d'autres options implicites à la gestion des stocks tel que la gestion de la carte du restaurant.

Dans le cadre de notre projet nous avons été amenés à implémenter les fonctionnalités que nous avons dégagées lors de notre analyse et ajouter des interfaces graphiques pour que le logiciel soit facilement accessible à tout type d'utilisateurs.

La gestion du temps liée notamment à un certain nombre d'examen et du projet communication aura été à certains moments compliquée et la gestion du stress tout autant.

La répartition du travail s'est faite très naturellement et très vite l'analyse UML avait été séparée en deux parties distinctes puis suivie d'une concertation du binôme pour ajustement.

Le code aura suivi un chemin légèrement différent puisque les fonctions graphiques ont été développée séparément des fonctions base de données menant à certains désagréments en fin de projet qui toutefois auront été instructif.

L'utilisation d'une base de données portable fût un choix important.

En effet si chaque restaurant doit avoir son propre logiciel et gérer ses stocks pouvoir l'avoir sur place évite des blocages en cas de panne réseau ou serveur. Sans empêcher que la maison mère ait accès au stock

## 2 Organisation

### 2.1 Organisation du travail

Notre travail a eu en plusieurs phases :

Implémentation des classes et méthodes déclarées dans notre analyse

Création de la base de données

Connexion de nos classes à la base de données

Implémentation de l'interface graphique

L'analyse fonctionnelle fût un grand challenge pour nous et l'absence de diagramme de base de données venait principalement du fait que sa modélisation sous forme UML apparaissait compliquée d'autant que celle-ci était déjà en proie à de fréquents changements et maintenant que nous sommes à la fin du projet, c'est peut-être un avantage.

La base de données à énormément évoluée au cours du développement et l'analyse UML de celle-ci aurait été en grande partie erronée aujourd'hui pas tant sur la forme des tables que sur celle des colonnes qui ont bougé au gré des besoins du code.

Le GANTT à été lui globalement respecté une fois nos objectifs arrêtés.

La réalisation de la partie graphique fût compliquée bien que la dite interface soit simple, elle nous paraissait toutefois nécessaire au vu de l'utilisation annoncée nous avons donc voulu tenter de la réaliser malgré tout.

La fonction `Commander()` ne fonctionne pas ce qui est dû partiellement à une incompréhension du protocole de connexion à la base de données, une étude sur le sujet sera donc menée...

## 2.2 Gestion des scripts

Du point de vue graphique pour que nos interfaces s'affichent nous avons utilisées les méthodes implémentées dans les différentes classes selon ce que doit afficher la fenêtre mais aussi en redirigeant les boutons choisis vers de nouvelles fenêtres.

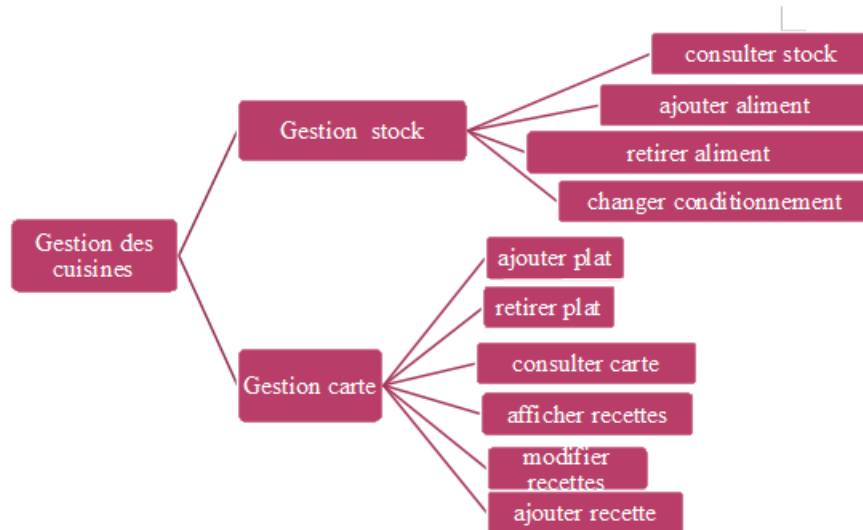


Figure 1: Schéma du Menu.

## 2.3 Améliorations

Nous aurions voulu implémenter plus de fonctions dans certains cas voici leurs histoires.

### 2.3.1 Une interface graphique complète

Il y a bien un début d'interface graphique mais toutes les fonctionnalités ne sont pas disponibles sous cette forme et elle est de plus assez simple. Une interface graphique travaillée avec des icônes notamment, ainsi que des formulaires à la place des icônes pop-up.

L'implémentation de valeurs par défauts dans les formulaires de modification correspondant aux valeurs déjà présentes, une boîte de confirmation pour les suppressions et un affichage plus soigné pour la carte et les stocks. À cela on ajoutera une mise en forme automatique lors de la saisie et un tableau de calorie.

### 2.3.2 Les commandes

Il était initialement prévu de réaliser une fonction commande permettant de simuler la déplétion des stock et qui n'a été ici que partiellement réalisée puisque 'il n'est possible de retirer qu'un seul ingrédient à la fois .

Une vrai interface de commande donc permettant d'être utilisée comme on le voit souvent dans les restaurants aujourd'hui par les serveurs pour savoir ce qui est toujours disponible lors des rush.

Sous sa forme complète il aurait été utile qu'elle face office de ticket permettant au cuisinier de savoir où il en est dans ses commandes et au serveur de savoir quand la préparation est lancée permettant grâce au temps estimé de mieux gérer la salle.

### 2.3.3 Les ajouts et modifications

Cela rentre partiellement dans l'interface graphique mais pour standardiser certaine colonne le choix via un menu déroulant de certains paramètres aurait allié l'utile à l'agréable.

### 2.3.4 Sélection

Enfin, pouvoir sélectionner à partir de la carte ou du stock les éléments que l'on souhaite modifier,et pouvoir réaliser une recherche dans le cas de stocks important un menu déroulant pouvant s'avérer particulièrement contraignant.La commande avec simple click est également une possibilité

### 2.3.5 Et autres...

Identifier les aliments manquants ou en faible quantités,système de recette principale et secondaire avec proposition de recette végane, sans gluten... , Implémentation des allergènes,mise en place du calcul de calorie automatique, menus diététiques,gestion des prix, système de mise en ligne de bases de données pour gestion globale de la chaine de restaurant, avec optimisation des trajets de livraisons...

## 3 Fonctionnement

### 3.1 Description des packages

#### 3.1.1 Proj

Proj contient les classes de gestion de la base de données .Il permet de modifier les différentes tables, de mettre à jour des entrées déjà présente ainsi que savoir ce qui est déjà rentré suivant certaines conditions. Il contient également la méthode disponibilité qui permet de savoir s'il reste assez d'aliment pour réaliser une recette.

#### 3.1.2 Tp

Tp contient la partie graphique du projet soit le début d'interface graphique proposé. Celle-ci se compose de plusieurs fenêtres comportant des boutons et de fenêtre pop-up qui permettent de saisir les valeurs dont nous avons besoin. L'interface n'est pas complète du fait d'un certain manque de temps lié aux différents projets, examens et aléa personnels auxquels nous avons été confrontées.

### 3.2 Execution

L'exécution de la partie graphique à partir d'éclipse se fait en lançant le main de la classe fenêtre 1. Elle ne gère pas les exceptions, c'est à dire que si un String est rentré à la place d'un integer dans une modification par exemple l'interface continuera mais l'aliment ne sera pas ajoutée. L'interface graphique, tout comme les fonctions restent en partie expérimentale, tout a été pensé, mais pas forcément implémenté. Il existe un exécutable mais il est hautement instable, il est disponible dans le fichier projet info. Veiller à ce que le jdbcsqlite soit bien importé par votre IDE.

## 4 Conclusion

Ce projet informatique fût certes chaotique mais instructif. Nous avons appris à créer des fenêtres sous Java avec swing, accéder à des base de données à distance avec le gestionnaire sql lite, gérer un repertoire git, en résoudre les conflits ,développer en équipe et créer des fichier LaTeX parce que c'est plus propre. Des compétences qui nous seront très utiles plus tard. On notera également l'utilisation du plug-in papyrus en début de projet puis de diverses logiciels UML lors de l'analyse. Ce qui fait un panel de compétences très large. Cela nous a également montré qu'un projet très simple peut demander énormément de temps.

## 5 Bibliographie

Nos professeurs.Première référence.

<https://openclassrooms.com/courses/apprenez-a-programmer-en-java>

Pour la connexion à la base de donnée: <http://www.sqlitetutorial.net/sqlite-java/sqlite-jdbc-driver/>

<https://github.com>

Et bien sur un grand merci à StackOverflow pour son existence

## Contents

1	Introduction	2
2	Organisation	3
2.1	Organisation du travail . . . . .	3
2.2	Gestion des scripts . . . . .	4
2.3	Améliorations . . . . .	4
2.3.1	Une interface graphique complète . . . . .	4
2.3.2	Les commandes . . . . .	5
2.3.3	Les ajouts et modifications . . . . .	5
2.3.4	Sélection . . . . .	5
2.3.5	Et autres... . . . .	5
3	Fonctionnement	6
3.1	Description des packages . . . . .	6
3.1.1	Proj . . . . .	6
3.1.2	Tp . . . . .	6
3.2	Execution . . . . .	6
4	Conclusion	6
5	Bibliographie	7