

INFO-F307 : Génie logiciel et gestion de projets

Projet d'année - itération 3

Planification et conception



Groupe 6

mars 2022

Université Libre de Bruxelles

Table des matières

1	Planification globale du projet	2
2	Planification détaillée	3
2.1	Itération 1	3
2.2	itération 2	4
2.3	itération 3	5
2.4	itération 4	6
3	Choix de conception	7
3.1	Annexes	7

1 Planification globale du projet

La planification du projet a été initiée à l'aide 3 diagrammes. 2 diagrammes de classe et 1 diagramme représentant la base de données ont été produits afin de donner un premier aperçu du travail à entamer.

Figure 1: Diagramme de la User Story 1

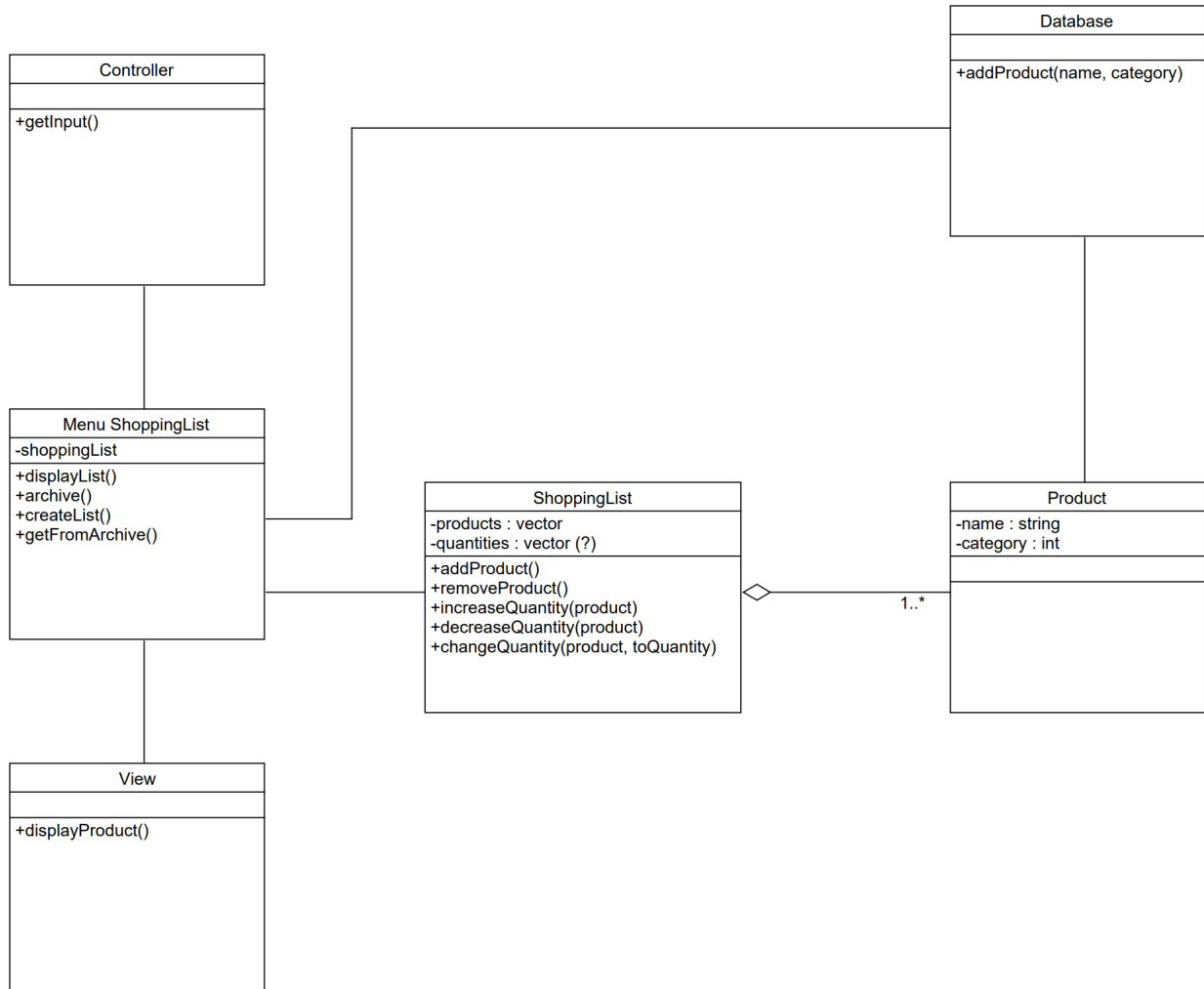


Figure 2: Diagramme de la User Story 2

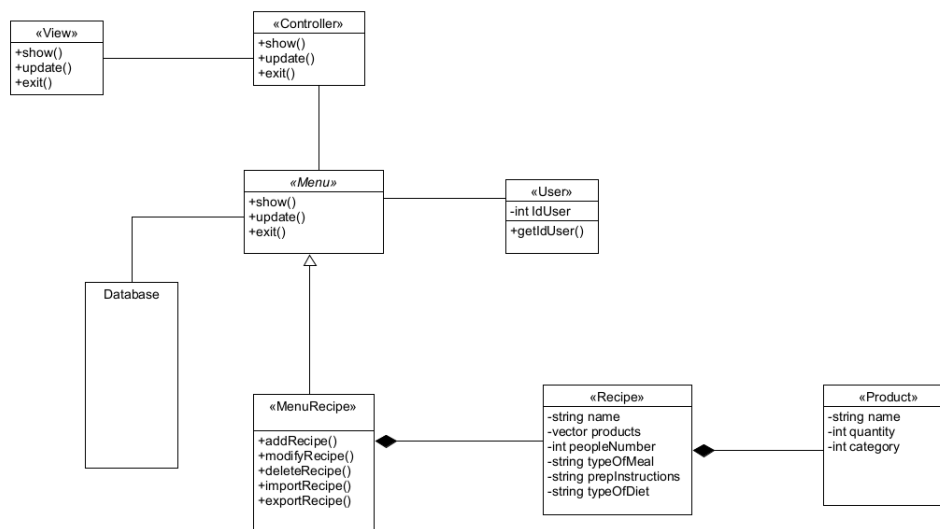
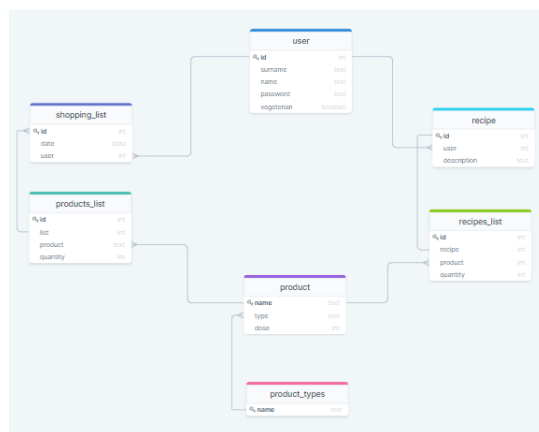


Figure 3: Diagramme de la base de données



2 Planification détaillée

2.1 Itération 1

Une fois ces diagrammes réalisés et la recherche nécessaire pour comprendre les librairies nécessaires (javafx, sql, json,...), le travail a pu être entamé.

Les fonctionnalités convenues pour l'itération 1 étaient les trois premières histoires utilisateur, cependant seules les deux premières ont pu être implémentées.

Le travail fut initialement séparé en 5 parties :

- Le back-end de l'histoire 1
- Le back-end de l'histoire 2
- Le front-end de l'histoire 1
- Le front-end de l'histoire 2
- La base de donnée pour les histoires 1 et 2

La rotation des binômes s'est avérée bien plus compliquée que prévu. En effet, se familiariser avec un nouveau code et son nouveau binôme était trop fastidieux pour réaliser le tout dans un rythme de 11 heures par semaine. Nous avons donc respecté le pair programming avec une rotation plus petite.

Concernant la mise à jour du groupe face à l'évolution du projet, nous avons opté pour des réunions rapides de 15-30 minutes tous les 3 jours. Ceci a permis de remettre au clair les difficultés, les décisions d'implémentation et enfin, ça a permis une communication entre les groupes concernant les liaisons de chaque partie. Malgré cette décision, les connaissances manquantes de javafx, l'évolution du projet ne fut pas assez rythmée, le groupe n'a donc pas réussi à produire l'histoire 3.

Les difficultés supplémentaires se sont retrouvées dans les décisions d'implémentation, il était difficile de décider certaines implémentations sachant que la réponse se retrouvait chez le client, des libertés ont donc été prises basées sur un vote du groupe. Cette difficulté prend sa source dans la difficulté de créer un environnement de réel projet.

2.2 itération 2

Les fonctionnalités convenues pour l'itération 2 étaient les histoires 3,4 et 9. Il nous était également demandé de peaufiner les deux premières histoires et d'utiliser un Scene Builder pour faciliter la création du Front-end. Il a donc aussi fallu retravailler l'aspect graphique des histoires précédentes.

Pour cette itération, à l'exception du passage à Scene Builder, le travail n'a pas été divisé entre le back-end et le front-end mais en fonctionnalités à implémenter.

Le travail fut initialement séparé en 5 parties:

- Passage à Scene Builder
- Fin des histoires de l'itération 1
- Histoire numéro 9
- Histoire numéro 3

- Histoire numéro 4

Certains blocs de travail étant beaucoup plus conséquents que d'autres, les binômes ont du travailler sur plusieurs de ces blocs afin de répartir le travail équitablement.

Après avoir remarqué qu'il était difficile de trouver un moment arrangeant les deux personnes du binôme, nous avons décidé d'expérimenter une nouvelle méthode de travail. Chaque personne du groupe rentre ses disponibilités sur un calendrier prévu à cet effet, afin de trouver plus aisément quelqu'un dont les horaires de travail concordent. Il a parfois été nécessaire pour certaines personnes de travailler seul car aucune disponibilité ne les arrangeait. Nous avons également décider d'utiliser d'avantage le Trello afin de se savoir qui travaille sur quelle tâche mais aussi pour que tout le monde puisse savoir ce qu'il restait à faire.

La fréquence des réunions est restée identique pour cette itération.

Les difficultés supplémentaires sont restées relativement identique. La principale difficulté rencontrée a été la perte de motivation générale du groupe. Cela peut être expliqué par la longueur de l'itération et la présence d'un congé de deux semaines au milieu. Le travail a tout de même été terminé à temps grâce au temps amplement suffisant qui nous était accordé.

2.3 itération 3

Les fonctionnalités convenues pour l'itération 3 étaient les histoires 5 et 6. il était également nécessaire d'effectuer un refactoring conséquent sur le code déjà présent et de créer des tests pour certaines classes. Après discussion avec le client, le rajout d'une manière de récupérer le mot de passe a été demandé.

Lors de cette itération, le travail a été divisé en 4 parties:

- La récupération du mot de passe
- La gestion des erreurs
- L'intégration en JavaScript de la carte
- L'ajout de nouveaux types de données dans la Database et son implémentation DAO
- L'histoire numéro 6

La méthode de recherche de binôme convenait à tout le monde et nous avons donc décidé de continuer comme cela pour la troisième itération

la fréquence des réunions est restée identique

Les problèmes majeurs rencontrés ont été :

- le choix de la librairie à utiliser pour la map: certaines ne convenaient pas à nos besoins ou étaient trop complexes à mettre en place, hésitation leaflet / ArcGIS.
- la récupération du mot de passe: le mot de passe étant crypté dans la base de données, le récupérer et l'envoyer au client par mail ne suffit pas. Nous avons donc opter pour la génération d'un String aléatoire afin de pouvoir facilement l'envoyer par mail à l'utilisateur et ensuite le crypter et le remplacer dans la base de données. Une fois connecté, l'utilisateur a la possibilité de changer ce mot de passe dans le menu "Profil".
- Notre connaissance lacunaire du JavaScript a également été un frein pour l'implémentation de la carte comme fonctionnalité de notre projet.
- Problème de communication entre JavaScript et JavaFX. JavaFX demande l'utilisation du paramètre "Action Event" pour pouvoir changer de fenêtre mais javaScript ne permet pas ce type d'action.

Nous n'avons malheureusement pas eu le temps nécessaire pour finir l'implémentation de l'US6, bien que certaines fonctionnalités sont déjà disponible (recherche du magasin le plus proche sur la carte mise à disposition de l'utilisateur). Nous avons donc choisi de terminer parfaitement l'US5 plutôt que de rendre les deux histoires à moitié finies

2.4 itération 4

Les fonctionnalités convenues pour l'itération 4 étaient les histoires 10 et 12 ainsi que le refactoring du projet.

Le travail a donc été divisé en trois parties:

- L'histoire numéro 10
- L'histoire numéro 12
- Le refactoring

Les problèmes majeurs rencontrés furent,

- L'identification de l'endroit où l'utilisateur se trouve au niveau du logiciel afin de lui proposer le bon endroit pour la section d'aide directement. Cette fonctionnalité n'a pas été implémentée et devrait être une fonctionnalité en plus si le temps le permettait.
- L'import des fichiers json, tant pour les recettes que pour les produits d'un magasin, n'est pas optimisé et la gestion des erreurs au niveau du format du fichier n'est pas approfondie jusqu'au bout.

Les deux tâches ont donc été implémentées correctement et les fonctionnalités sont disponibles.

Le refactoring a permis de construire une javadoc complète ainsi qu'un programme sans aucune erreur de QAPlug ou de warnings.

3 Choix de conception

Le modèle d'implémentation est le modèle MVC, ceci étant demandé par les consignes du projet. Le modèle reprend donc, globalement, chaque partie des histoires (un produit, une liste de course, une recette, ...).

Nous avons choisi d'utiliser Scene Builder afin de faciliter l'implémentation graphique (problème d'alignement en utilisant uniquement JavaFX) . Enfin, chaque partie du projet est aisément accessible via la barre de gauche.

En vue des histoires utilisateurs à venir, des éléments comme User dans la base de données ont déjà été implémentés, que ce soit pour gérer la disposition de la fenêtre ou pour éviter des reformattages de la base de données trop importants.

Nous avons préféré opter pour une fenêtre de gestion du profil plutôt que pour des "Settings", car aucun choix lié à l'application ne doit être fait.

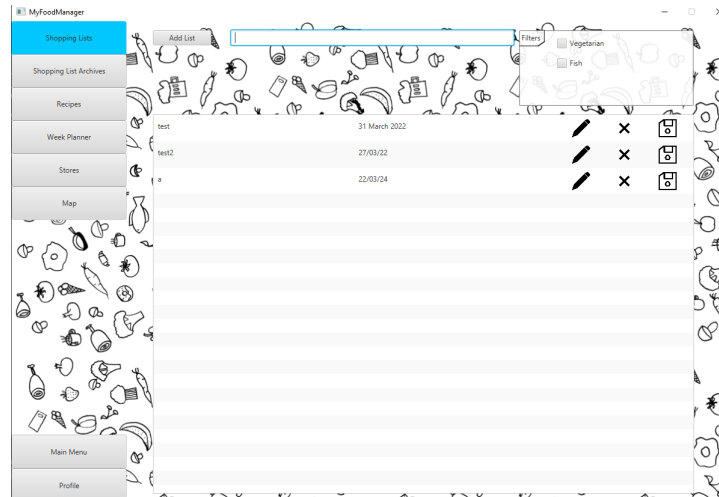
Lors de l'ouverture des détails concernant un magasin sur la carte, nous avons opté pour un Pop-Up plutôt qu'un changement de fenêtre, cela permettant de contourner le problème des "Action Event" expliqué précédemment, mais également de pouvoir rester sur la carte lors de la fermeture de la fenêtre tout juste créé.

Les designs patterns n'ont pas été réfléchis, ceci étant dû aux manques de connaissances et de temps. Malgré ceci, le modèle DAO a été choisi pour la base de donnée et bien que l'implémentation actuelle y ressemble, le design pattern n'est pas encore respecté.

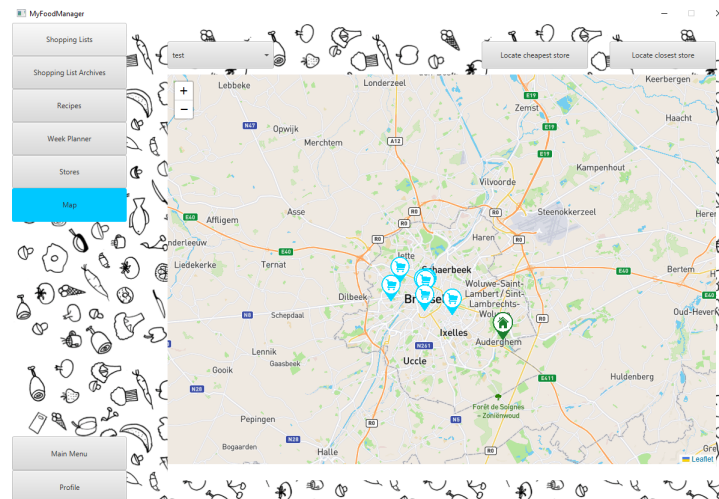
Enfin, dans le but de pouvoir lancer le jar, une classe App a été créée. Ceci a pour impact de déplacer la librairie javafx dans le classpath plutôt que dans le modulepath. Bien que ceci crée un warning de javafx, celui-ci ne put être évité et, selon les ressources vérifiées, ce warning ne fut pas évitable à moins de sacrifier la possibilité de lancer le fichier jar directement.

3.1 Annexes

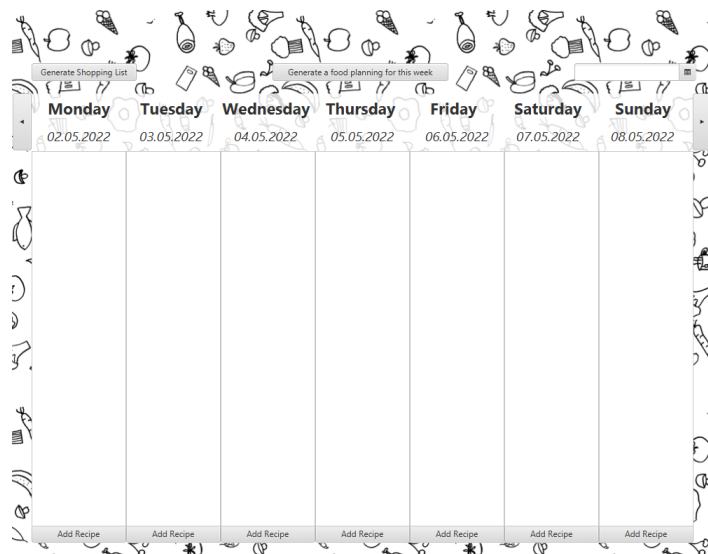
Les listes de course



La carte reprenant les magasins



Le planning auto-généré des repas pour une semaine



Les différents magasins référencés



Les différentes recettes consultables

