

UNIVERSITÉ LIBRE DE BRUXELLES



INFO-F-307-0-201718

GÉNIE LOGICIEL ET GESTION DE PROJETS - 201718

Planification du projet

Titulaire:

F. PLUQUET

Auteur:

Mouad OUTMANE

Prateeba RUGGOO

Akilleas PAPPAS

Nikita BOZGA

Gilles BONKAIN

Raphaël REYNOUARD

Louis CLAESSENS

Hamza IDELCAID

Pierre-Alexandre BLANC

Contents

1	Fonctionnalités Implémentées	2
2	Difficultés techniques	4
3	Motivation des choix concernant la conception	6

Chapter 1

Fonctionnalités Implémentées

Les fonctionnalités implémentées:

- Mise en place d'une base de données générique contenant :
 - Des magasins
 - Des produits
 - Des utilisateurs
 - Des recettes
 - L'historique de recherche
- Mise en place d'une interface graphique qui permet à l'utilisateur de :
 - Créer un compte avec un pseudo et un mot de passe (sécurisé)
 - Se connecter avec le compte créé
 - Se déconnecter
 - Trouver tous les magasins qui vendent un produit
 - Rajouter un produit qui ne se trouve pas dans la base de données
 - Entrer une liste de courses et la modifier :
 - * Suppression d'un produit
 - * Ajout d'un produit
 - * Ajout/changement de la quantité d'un produit
 - Calculer le prix total d'une liste de courses par magasin vendant tous les produits de cette liste de courses
 - Recevoir des informations sur les valeurs nutritionnelles de produits
 - Créer des recettes contenant des ingrédients (=produits) en certaines quantités et des instructions
 - Choisir un nombre de personnes par recette
 - Planifier les recettes en repas, à raison de 3 repas par jour, sur une semaine

- Générer une liste de courses à partir d'une recette ou de plusieurs recettes planifiées
 - Consulter une carte qui contient des épingles pour signaler la présence de magasins
 - Consulter le nom et l'horaire d'un magasin en sélectionnant une épingle
 - Rajouter un magasin sur la carte et y changer les informations qui y sont associées
 - De zoomer et dézoomer sur la carte
 - Effectuer des recherches en fonction des critères choisis.
 - Sauvegarder des recherches pour créer des filtres réutilisables
 - Ajouter ou modifier des produits d'un magasin. Pour ce faire le client peut encoder à la main les changements ou importer un ensemble de données.
 - Pouvoir faire sa liste de courses en fonction du magasin où elle revient le moins cher.
 - Pouvoir faire sa liste de courses en fonction du magasin le plus proche de sa position avec 3 "modes" disponibles : à pied, en vélo ou en voiture.
- A-t-on implémenté toutes les fonctionnalités ?

Oui, on a implémenté toutes les fonctionnalités concernant les histoires. De plus, le client avait demandé une fonctionnalité qui était d'afficher les logos des magasins. Celle-ci a aussi été implémentée.

Chapter 2

Difficultés techniques

Les difficultés techniques rencontrées :

- Gestion de Gitlab
- Choix concernant les librairies à utiliser pour la base de données
- Affichage de logos dans l'interface
- Intégrité et cohérence des entrées dans la base de données
- Sécurisation
- Séparation modèle vue contrôleur
- Gestion des erreurs
- Gestion des exceptions
- Rassemblement des ressources externes dans leurs dossiers respectifs
- Création de la carte et le passage des données de la base de données à la carte.
- Magasin le plus proche
- Executable .jar

Comment les a-t-on contournées ?

- Gestion de Gitlab : Travail de recherche sur le fonctionnement de cet outil, ainsi que des tests.
- Choix concernant les librairies à utiliser pour la base de données : On avait tous des avis différents concernant la base de données à utiliser, par exemple : Jasondb, fichier .txt ou mysql. Après une heure de réflexion on a opté pour mysql.
- Affichage de logos dans l'interface : Changement de type de données dans la base de données (de VARCHAR vers BLOB)

- Intégrité et cohérence des entrées dans la base de données : On a revu la réalité physique des objets et redéfinit les méthodes en fonction de cela.
- Sécurisation : Vu que la sécurisation était un concept nouveau pour tout le monde, il a fallu beaucoup plus se renseigner.
- Séparation modèle vue contrôleur : Application du modèle vue contrôleur
- Gestion des erreurs : Classes spéciales qui gèrent la gestion des erreurs
- Gestion des exceptions : Création d'une classe d'exception qui est plus spécifique.
- Rassemblement des ressources externes dans leurs dossiers respectifs Restructuration de tous les dossiers du projet de manière organisée
- Création de la carte et le passage des données de la base de données à la carte: Utilisation d'une librairie plus adaptée et création d'une classe qui lie la carte au reste.
- Recherche sur les librairies et sur les fonctionnalités pouvant calculer un chemin entre deux positions.
- Changement d'API pour gérer la map afin que le javascript soit compatible avec l'exécutable.

Chapter 3

Motivation des choix concernant la conception

Design pattern principaux :

- Mysql et jdbc driver car les membres du groupe étaient plus à l'aise avec ce genre de gestion de base de données grâce à un précédent projet. Cela est aussi plus flexible.
- Choix d'implémenter une interface
 - User friendly
 - Ergonomique
- Ajout d'une librairie permettant au menu déroulant compris dans notre application d'autocompléter l'input.
- Implémentation du design pattern Singleton afin de restreindre l'instanciation de la classe qui lie la carte au reste du code. Il est aussi utilisé pour l'unique connexion à la base de données et pour l'initialisation de chaque controller.