

Démarche

Comme conseillé dans les spécifications du Projet 5, j'ai commencé par écrire les user stories afin de dégager les fonctionnalités du programme. Je me suis servi de Trello pour les ordonner, le prioriser. Voici les user stories retenues :

- A/ En tant qu'utilisateur, je veux lancer le programme afin d'accéder à ses fonctionnalités
- B/ En tant qu'utilisateur, je veux sélectionner une catégorie afin d'afficher les aliments associés.
- C/ En tant qu'utilisateur, je veux sélectionner un aliment que j'aime afin de connaître un aliment de substitution.
- D/ En tant qu'utilisateur, je veux enregistrer le résultat de ma recherche afin de pouvoir y revenir rapidement.

La 2^e étape a consisté à réaliser le schéma de base de données relationnelle (cf github). Il a fallu pour cela bien étudier la base de données d'Openfoodfacts et déterminer les éléments à récupérer et exploiter dans le programme.

Une fois le schéma bien en tête, j'ai cherché un moyen de récupérer les données directement depuis le site openfoodfacts. Le wiki de l'API d'OFF indique que des fichiers .json sont directement accessibles depuis Internet qui contiennent tous les produits par pays. J'ai donc cherché le moyen de faire des requêtes en HTTP sur Internet en python. C'est ainsi que j'ai découvert le module 'requests'. L'inconvénient de ce module est qu'il renvoie des objets peu exploitables qu'il faut traduire en liste. Une fois le module 'requests' pris en main, il a fallu acquérir un module permettant la lecture/écriture de BDD en mysql. Mon choix s'est porté sur 'mysqlclient' (le MySQLdb version python3).

La principale difficulté lors de la récupération des données a été de vérifier que les produits enregistrés avaient bien tous les champs critiques (name, nutrition grade, catégorie) et optionnels (purchase places, stores, url). Les données sont en effet rentrées à la main sur le site et de nombreux champs sont omis.

Une fois le check d'intégrité codé et les tables remplies, j'ai commencé à coder l'interface (en console). J'ai ainsi pu coder une à une les user stories, en n'omettant pas les contraintes liées à l'interface (check d'intégrité sur les saisies utilisateurs, possibilité de retour au menu principal).

Liens :

Trello : <https://trello.com/b/9z0B1qnR/oc-p5>

Github : <https://github.com/FatSin/oc-le-meme-en-mieux>