1. **Démarche de conception**

Pour bien débuter la conception du projet, nous avons commencé à lister les différentes fonctionnalités demandées. Nous avons écarté huit propriétés demandées :

|  |  |
| --- | --- |
| **1** | Création du schéma relationnel de la base de données |
| **2** | Etude des réponses de l’API Open Food Facts |
| **3** | Création du découpage du programme |
| **4** | Création d’un jeu de donnée à importer dans la base de données local |
| **5** | Utilisation de configparser |
| **6** | Création de la base de données |
| **7** | Créer les différentes requêtes sql |
| **8** | Remplissage de la base avec le jeu de donnée créé |
| **9** | Créer l’affichage pour que l’utilisateur puisse interagir avec le programme |

Par la suite, nous avons ouvert le projet dans un répository git pour y mettre les différentes ressources du projet et effectuer les différentes sauvegardes dont nous aurons besoin. Ce repository se trouve via [ce lien](https://github.com/ZasshuNeko/-OC--Projet-n-3).

1. **Etude de l’algorithme**

Nous avons scindé le projet en 6 parties.

Deux parties reposant sur des fichiers extérieurs, l’un servant à la configuration du programme et permettre de garder des informations importantes au sein d’un même fichier sans encombrer le code, que l’on a nommé  « config.ini ». L’autre fichier étant le script de création des bases de données pour permettre à chaque utilisateur de partir d’une base commune lors de l’utilisation du programme que l’on nomme « script\_sql.sql ».

Le fichier à lancer qui contient le squelette du programme est appelé « OFF\_main.py » il est constitué d’une classe « main » qui est interpellé lors du lancement pour générer l’objet programme, c’est à partir de ce fichier que les autres classes vont être interpellé pour former la vie du programme et son interaction avec l’utilisateur. Lors de ce lancement plusieurs options sont misent à disposition de l’utilisateur et il important d’étudier l’algorythme selon ces différentes options :

La première option proposée est la création de la base de données et l’intégration d’un jeu de donnée dans la base de données local. Cela fait appel à trois classes : Classe affichage, Classe requete, Classe bdd\_mysql

La classe affichage est initialisé et crée une instance d’affichage, ainsi nous appellerons cette classe à chaque fois qu’une interaction avec l’utilisateur est demandé ou à chaque fois qu’une information doit apparaitre à l’écran. Chaque choix est ramené à une fonction vérifiant l’intégrité des données pour éviter les erreurs (tel que mettre une lettre ou un choix n’étant pas dans la liste)

La classe requête contenu dans le fichier « class\_requete.py » contient l’ensemble des demandes vers l’API d’Open Food Facts par ces différents module : req\_catégorie, req\_produit,req\_store, il permet de faire créer une réponse en jsons utilisable par la suite pour former un jeu de donnée que nous allons fournir à notre base de donnée local. Cette classe permet aussi de gérer le lot de donnée obtenu en lui retirant les données inutiles venant de l’API et le rendant plus malléable pour le projet. Pour cela, chaque catégorie choisis est envoyé vers la classe requête pour créer un dictionnaire formé de chaque produit provenant de l’API. Lors de la création de ce dictionnaire dans le module « crea\_dictionnary » les produits sont liée à leurs catégories et vendeurs. Une fois chaque produits des catégories extraites dans le dictionnaire, les doublons son retiré avec le module « dbl\_listing ».

La classe « bdd\_mysql » est initialisé pour instancier la connexion entre le programme et la base de données, cette dernière est vérifiée par une fonction permettant de faire remonter les soucis de connexion. Une fois la connexion instancié, la class va permettre la création de l’architecture de la base de donnée puis prendre le jeu de donnée généré par l’API de Open Food Facts retravaillé par la class requête pour l’intégrer dans la base de donnée local.

La seconde interaction avec le programme permet d’interroger la base de données locale pour permettre de rechercher un substitut alimentaire. Cette interaction fait appel à la classe « bdd\_mysql » et à la classe affichage.

Lors de ce choix, la classe « bdd\_mysql » suit alors une simple logique, il cherche les catégories dans la base de donnée local dans la table « categories » puis les donnent à afficher à la classe affichage. Une fois la réponse effectuer par l’utilisateur la réponse est donnée à la classe « bdd\_mysql » il va alors interroger la base de jointure entre les catégories et les produits, la réponse de cette requête est fournis à la classe affichage pour le choix utilisateur. Quand le produit a substitué est choisis, on compare le « nutri\_score », si celui-là n’est pas renseigné, on compare alors les le rapport calorifique de chaque produit pour extraire un substitut au produit sélectionné par l’utilisateur.

Une fois le substitut sélectionné, les données sont envoyés dans la classe affichage ou l’utilisateur à la capacité d’enregistrer son substitut dans la base de données.

Le dernier choix est celui de charger les recherches faites par l’utilisateur, lors de cette sélection, la classe « bdd\_mysql » est interpellé pour interroger la table « substitut\_save » puis afficher par la classe affichage.

Nous listons alors les étapes que nous allons devoir exécuter pour réaliser ce projet, nous étudierons ces étapes celons les interactions entre l’utilisateur et le programme :

1. **Etablir un plan de travail**
   1. Méthode agile
   2. Déterminer les demandes et les contraintes du projet
   3. Etude de l’API Open Food Facts et des résultats des jsons obtenus
   4. Etablir un schéma relationnel de base de données
2. **Création du programme**
   1. Etablir les différentes classes dont aura besoin le programme
   2. Etablir le fichier de configuration
   3. Installer, créer la base de données
3. **Création des requêtes API et sql**
4. **Finaliser la logique utilisateur**
   1. Lier les différentes classes
   2. Permettre une interaction utilisateurs dynamique
   3. Enregistrement du substitut
5. **Difficultés & solutions**

Les premières difficultés furent de s’approprier l’API de Open Food Facts. Bien comprendre chaque requête, faire la part des données inutiles pour le projet et trouver la meilleure solution de présentation pour le programme. Puis travailler au mieux le format json pour parvenir à créer un contenant de donnée utile pour réinjecter dans la base de données local. Une fois ce format de donnée établit, nous avons installé et configurer Mysql. Une des réflexions était de bien déterminer le nombre de table, le type de donnée dont nous allons avoir besoin. Une fois cela fait réfléchir à la liaison entre les tables, déterminé les jointures pour avoir une idée plus précise du type de requête dont on aura besoin. Nous avons alors établit quatre tables gérant chacune quatre objets différents : Les produits, les catégories, les vendeurs et les sauvegardes. Comme nos jointures produits catégories et produit vendeur sont des liaisons un vers plusieurs, nous avons créé deux tables de jointure supplémentaire. Chaque table est constitué d’un champ « id » utilisé comme clé primaire. Nous avons par la suite généré des clés étrangères pour garantir une cohérence de donnée ainsi il semblait primordiale que ces clés soit présentes sur les tables de jointures ainsi que sur la table de sauvegarde. Nous remplîmes les tables catégories, vendeurs en utilisant les listes créées puis chaque produit, effectuant dans un même temps les liens dans les tables de jointure. Une fois réalisé, nous structurâmes les différentes requêtes sql pour répondre aux demandes du projet.