Projet 5 : Utiliser la base de données Open Food Facts

Requirement

- **Python 3.x**.
- Utiliser une base de données MySQL

Installation

- Telecharger ou cloner le repository.
- Utiliser un **environnement virtuel** est recommandé.
    - Exécuter la ligne de commande : `python3 -m venv /path/to/new/virtual/environment`
      puis `source <path/to/venv>/bin/activate` depuis MacOS
      ou `<path/to/venv>\Scripts\activate.bat` depuis Windows
- Installer les dépendances : `pip install -r requirements.txt`

Configuration de la base de données

- connectez-vous à la base de données MySQL avec tous les droits
- créer une database MySQL "openfoodfacts"
- exécuter le fichier bdd.sql dans la database

Dans constants.py modifier les élements suivant :

clé	valeur
host	Database host
database	database name (par défaut openfoodfacts)
user	user login
password	user password (si besoin)

1. Conception du programme

J'ai suivi la méthode de conception agile (scrum) pour développer ce projet.
Pour ce faire, j'ai découpé plusieurs fonctionnalités en plusieurs "stories".
Vous pouvez trouver ma "To do List" sur trello.com et le lien github.
On peut décomposer le programme en cinq fichiers.

- bdd.sql
- PrinterManager.py
- DatabseManager.py
- APIManager.py
- constants.py

Tout d'abord, le fichier **bdd.sql** contiendra le code sql permettant de créer les tables ainsi que les procédures et les vues stockées dans la base de données.
On comptera deux procédure et une vue stockées dans **bdd.sql:**

- La procédure **check_if_product_exist_by_bar_code** qui me permettra de vérifier si un produit existe déjà dans la base de données en fonction de son code bar.
- La procédure **get_product_detail** qui me permettra de récupérer toutes les données concernant un produit.
- La vue **V_get_substitutable_products** qui me permettra de récupérer une liste de tous les produits qui ont été substitué dans la base de données.

Le fichier **PrinterManager.py** contiendra la classe PrinterManager, son rôle sera de contenir la logique du déroulement des messages à afficher sur le terminal, puis, d'être le point de jonction entre la classe DatabaseManager dans **DatabaseManager.py** et la classe ApiOperator dans **ApiOperator.py**
PrinterManager va afficher les choix suivants sur le terminal :

1. Quel aliment souhaitez-vous remplacer ?
    a. Parcourir le rayon.
    b. Effectuer une recherche.
2. Retrouver mes aliments substitués.

Le fichier **DatabaseManager.py** contiendra la classe DatabaseManager, son rôle sera de communiquer avec la base de données, avec les paramètres de connexions situées dans **constants.py**.
Cette classe sera capable d'appeler des procédures ou des vues stockées disponibles dans la base de données afin de récupérer des informations concernant un produit. Et permettra aussi d'insérer des données relatives à un produit dans la base de donnée.

Enfin, le fichier **ApiManager.py** contiendra la classe ApiManager qui sera capable de communiquer avec l'API JSON d'Openfoodfacts. Elle permettra de lancer une recherche de produits sur Openfoodfacts et de retourner les substituts de meilleure qualité pour un produit.

En résumé: