**UNIVERSITÉ DE YAOUNDÉ I THE UNIVERSITY OF YAOUNDE I**

** ÉCOLE NATIONALE SUPÉRIEURE NATIONAL ADVANCED SCHOOL**

**POLYTECHNIQUE DE YAOUNDÉ OF ENGINEERING OF YAOUNDE**

**DÉPARTEMENT DE MATHÉMATIQUES MATHEMATICS AND PHYSICAL**

**ET SCIENCES PHYSIQUES SCIENCES DEPARTMENT**

**INFORMATIQUE 3 (STI 2022)**

**RAPPORT DE PROJET SUR LA CREATION D’UN GESTIONNAIRE DE BOUTIQUE EN PYTHON**

* **NIVEAU :** MSP2
* **GROUPE :** B
* **MEMBRES DU GROUPE:**

**NOM :** KAMDEM POUOKAM

**PRÉNOM :** IVANN HAROLD

**MATRICULE :** 21P254

**NOM :** KENFACK NOUMEDEM

**PRÉNOM :** FRANCK ULRICH

**MATRICULE :** 21P335

**NOM :** LEKEM NGUEMTE

**PRÉNOM :** NELSON

**MATRICULE :** 21P167

* **COORDONNATEUR :** Dr TALE

1. **INTRODUCTION**

Dans le cadre de l’apprentissage du langage de programmation Python au MSP2 à l’Ecole Nationale Supérieure Polytechnique de Yaoundé, l’Enseignant Dr TALE a confié à chaque groupe de 3 étudiants du groupe B de réaliser un projet : celui de la création d’un gestionnaire de boutique. Il s’agit alors ici, de présenter le rapport de réalisation dudit projet et d’expliciter les différents codes sous-jacents.

**NB :** Il s’agit de la version 2 de ce gestionnaire dont il sera question dans la suite. Il succède à la version 1 dans la mesure où il tient en compte les recommandations et suggestions de l'encadreur Dr TALE.

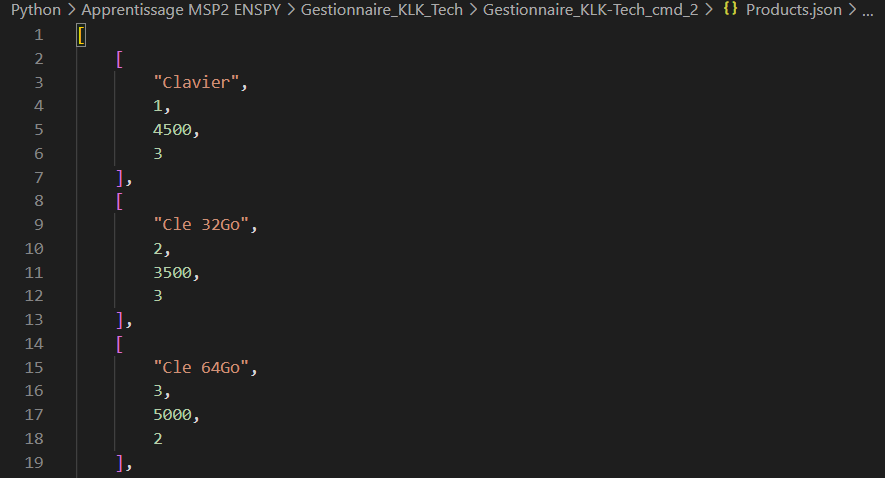
1. **OBJECTIFS**

* La Boutique doit contienir une liste de produits, chacun étant caractérisé par son nom, son code, son prix unitaire et la quantité en stock.
* A l’entrée dans la boutique, le programme doit demander le nom du client, puis lui souhaiter la bienvenue de façon plus ou moins chaleureuse selon qu’il ait déjà été ou non dans la boutique.
* S’il s’agit d’un ancient client, le programme doit lui suggérer ses anciennes commandes dans la mésure où il pourrait être intéressé à nouveau, et ainsi facilité son parcours dans la boutique.
* Si un produit a déjà été acheté à au moins 2 passages du client antérieurement, au passage suivant, le client doit bénéficier d’une réduction de 2% sur ledit produit.
* Après avoir passé ses commandes, le programme doit proposer au client le bilan et en cas de validation, produire et afficher sa facture.
* Le nom de la facture devra porter le nom dudit client.

1. **Fonctionnement de la Boutique**
2. **Gestion des Produits en stock**

* L’ensemble des produits en stock est stocké sous forme de liste dans un fichier « Products.json » (format permettant de visualiser aisément et de manipuler la liste des produits).
* Chaque produit est un élément de la liste caractérisé par : son nom, son code (attribué de façon automatique par le fichier « Gestionnaire\_Archives\_2.py », dont le principe est expliqué dans la suite), son prix unitaire et la quantité en stock.
* Il est déconseillé de modifier les données sur les produits dans le fichier « Products.json ». C’est à cet effet que le fichier « Gestionnaire\_Archives\_2.py » a été créé pour le gérant de la boutique, pour automatiser cette tâche et bien d’autres encore.

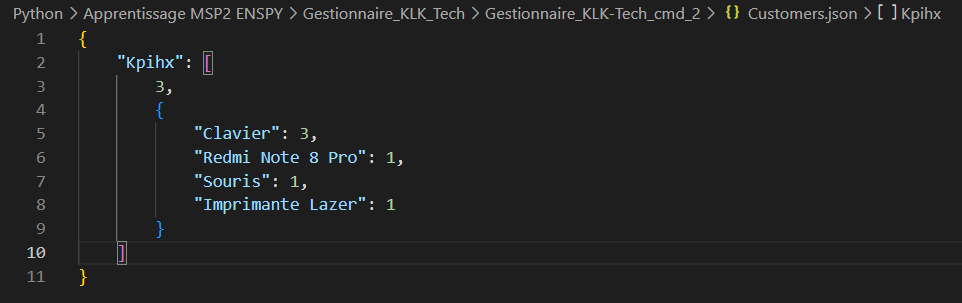
Voici un exemple de remplissage du fichier des produits :



1. **Gestion des Clients**

* Les données sur les clients sont stockées dans un dictionnaire, contenu dans le fichier « Customers.json » (format permettant de visualiser aisément et de manipuler la liste des clients).
* Un client est un élément du dictionnaire dont la clé est son nom, et la valeur est une liste contenant :
* Le nombre de passages du client dans la boutique ;
* Un dictionnaire dont les clés sont les différents produits achetés par le client dans la boutique, et les valeurs, les nombres de passages durant lesquels, il a eu à acheter les produits correspondants.
* Il est déconseillé de modifier les données sur les clients dans le fichier « Customers.json ». C’est à cet effet que le fichier « Gestionnaire\_Archives\_2.py » a été créé pour le gérant de la boutique, pour automatiser cette tâche et bien d’autres encore.
* Le fichier des clients est actualiser automatiquement par le fichier « Gestionnaire\_KLK-Tech\_cmd\_2.py », à chaque passage d’un client dans la boutique.

Voici un exemple de remplissage du fichier « Customers.json » :



**3- Présentation du programme « Gestionnaire\_KLK-Tech\_cmd\_2.py »**

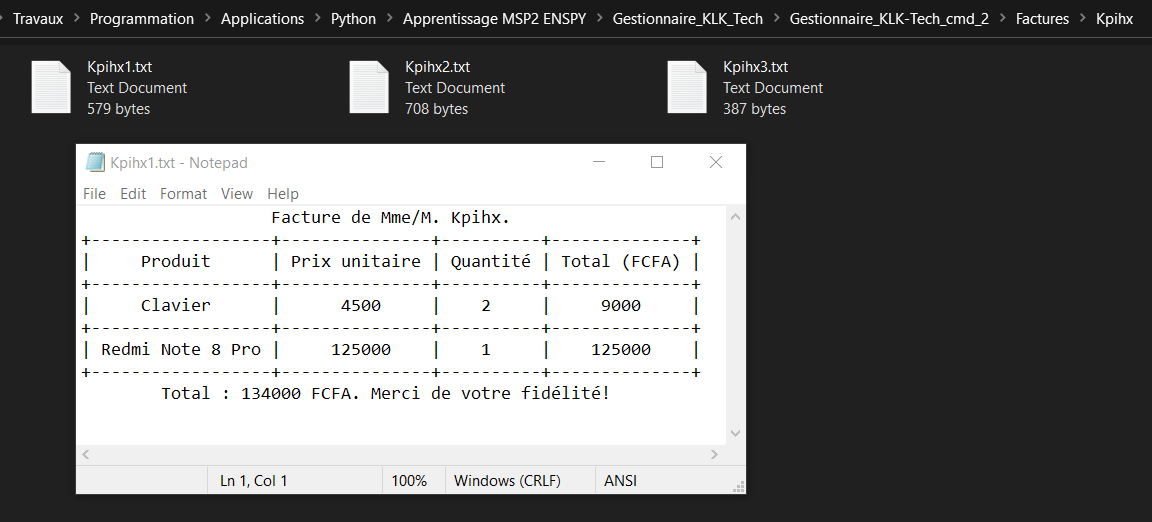
C’est le programme qui s’occupe du client à son arrivée dans la boutique.

* D’emblée, il récupère le nom du client et lui souhaite la bienvenue d’une façon plus ou moins chaleureuse, selon qu’il ait déjà été ou pas dans la boutique.
* Puis, s’il s’agit d’un ancien client, une éventuelle liste de ses précédents achats lui est proposée dans la mésure où cela pourrait l’intérésser à nouveau.
* Si de plus il a déjà eu à commander un produit à au moins de 2 passages dans la boutique, une réduction de 2% sur ce produit lui est proposée sur une éventuelle nouvelle commande dudit produit.
* Ensuite, le programme, suffisamment détaillé, propose les informations sur les produits en stock, et permet au client de passer ses commandes.
* Si une recherche d’un produit n’aboutit à aucun résultat, le programme essaye de trouver les produits en stock qui correspondraient le mieux à la recherche infructueuse. Si la démarche s’avère fructueuse, les produits correspondants sont suggérés au client.
* Après ses commandes, le client voit s’afficher le bilan de ses commandes, et en cas de validation de ce dernier, sa facture est produite et affichée à l’écran.

**NB :** \* La facture d’un client porte son nom, suivi du numéro de passage dans la boutique.

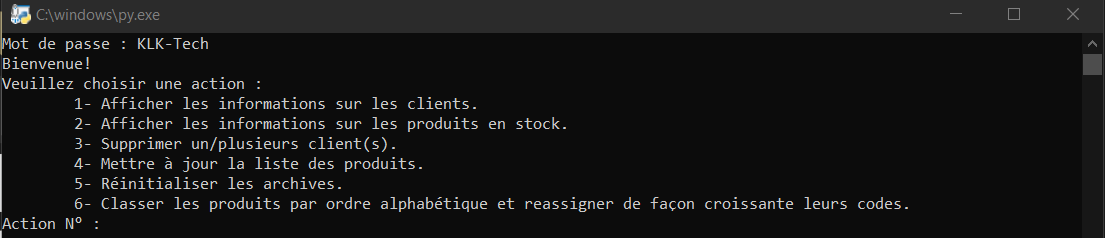
\* Les factures d’un client sont toutes stockées dans un dossier portant son nom, lequel dossier se trouve dans le dossier « Factures », contenant toutes les factures.

Voici un exemple de dossier de facture d’un client, avec un exemple de facture :

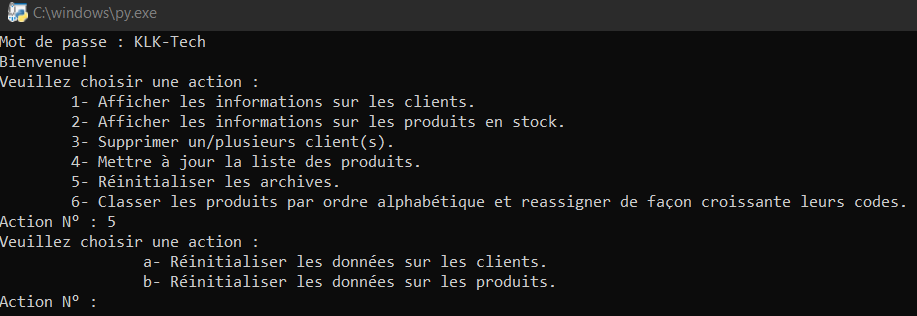


**4- Présentation du programme « Gestionnaire\_Archives\_2.py »**

* Ce programme est destiné au gérant de la boutique et lui permet d’automatiser certaines opérations de manipulation des données en stock.
* Son accès requiert un mot de passe dont seul le gérant à la possession. Dans le cadre de ce projet, le mot de passe est : **« KLK-Tech ».**
* Les différentes opérations pouvant y être effectuées sont listées dans l’image ci-dessous :



* L’interface utilisateur est suffisamment détaillée pour mieux saisir le fonctionnement des différentes possibles opérations.
* Petit éclaircissement sur l’opération « 5 : Réinitialiser les archives » : Elle comporte 2 options :

****

* L’option « a » supprime toutes les données sur les clients ainsi que l’intégralité des factures.
* Quant à l’option « b », vu que le magasin a toujours un stock par défaut de produits qui ne pourront jamais manquer, elle se charge de réinitialiser le stock des produits à ce stock par défaut.

**NB :** Ce stock par défaut est modifiable dans le code source du fichier **« Gestionnaire\_Archives\_2.py »**

* Petit éclaircissement sur l’opération « 6 : Classer les produits par ordre alphabétique et reassigner de façon croissante leurs codes » : Il peut arriver qu’un utilisateur inaverti, malgré les précautions ne recommandant pas la modification directe des données sur les produits dans le fichier « Products.json », s’obstine à le faire. Dans ce cas, il peut s’en suivre qu’il crée un désordre dans l’ordre alphabétique des produits, et même aux niveaux de leurs codes. Dans ce cas, cette option réordonne les produits dans l’ordre alphabétique de leurs noms, et attribue les codes commençant par 1 et évoluant par pas de 1.

1. **Conclusion & Bibliographie**

Avant de conclure, il est à noter que pour une meilleure lisibilité des codes sources, un module « Manager\_Functions\_cmd\_2.py » a été créé, pour héberger l’intégralité des fonctions et modules, dont on a eu besoin pour mener à bien ce projet. Une aide sur le fonctionnement des fonctions de ce module est présentée lors de l’ouverture de ce fichier (double-clic).

Parvenu au terme de la présentation de notre projet, qui portait sur la mise sur pied d’un gestionnaire de Boutique, il en ressort qu’à partir des connaissances en cours d’Informatique 3 (Python), dispensé par le Dr TALE, et de nos propres recherches dans le cadre de ce cours, on a pu mener à bien ledit projet. Pour une implémentation concrète de ce gestionnaire de boutique, on a dû ajouter une touche d’originalité : la mise sur pied d’un gestionnaire de stock pour le gérant de la boutique.

Pour finir, ce projet a été réalisé grâce au cours « Informatique III (Python) » du Dr TALE et au logiciel « Microsoft Visual Studio Code», pour l’écriture des scripts.