



RAPPORT DU PROJET DE PYTHON

Développement d'une application de Gestion de Bibliothèque

Réalisé par :

- AZIZI Nadia

Encadré par :

- HAJA Zakaria

Table des matières

1. Introduction :	3
2. Analyse et conception :	4
3. Structure du projet :	4
4. Fonctionnalités développées :	5
b. Interface d'ajout d'un livre :	6
c. Interface : Inscrire un membre :	7
d. Interface : Emprunter un livre :	8
e. Interface : Rendre un livre :	9
f. Interface : Visualisation des données :	10
i. Diagramme circulaire : % de livres par genre :	10
ii. Diagramme circulaire : % de livres disponibles :	10
iii. Histogramme : Répartition des auteurs :	10
iv. Courbe d'activité : Historique des emprunts :	11

1. Introduction :

- **Contexte :**

La gestion des bibliothèques nécessite aujourd'hui des outils informatisés capables de simplifier et d'automatiser les opérations courantes telles que l'enregistrement des livres, le suivi des emprunts et la gestion des utilisateurs. Dans un environnement éducatif ou institutionnel, la centralisation de ces informations est essentielle pour assurer une organisation efficace, réduire les erreurs humaines et faciliter l'accès aux données. C'est dans ce cadre que s'inscrit ce projet, développé dans le contexte d'un module de programmation avancée en Python.

- **Objectifs :**

L'objectif principal de ce projet est de concevoir une application complète de gestion de bibliothèque, qui couvre l'ensemble des fonctionnalités nécessaires à la manipulation d'un catalogue de livres et à la gestion des utilisateurs. L'application permet notamment d'ajouter ou de supprimer des livres, d'inscrire des membres, de gérer les emprunts et retours, de sauvegarder les données et de visualiser des statistiques d'utilisation. Une attention particulière a été portée à la robustesse du système grâce à une gestion rigoureuse des erreurs métier et à la modularité du code.

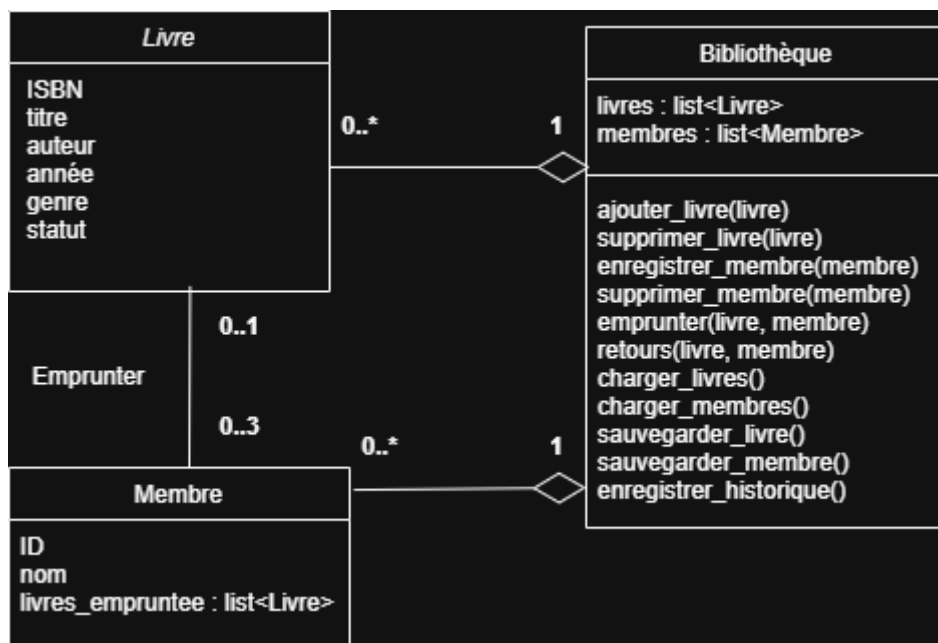
- **Technologies utilisées :**

- **Python** : langage principal utilisé pour la logique du programme.
- **Programmation Orientée Objet (POO)** : pour organiser le code en classes claires et réutilisables (Livre, Membre, Bibliothèque, etc.).
- **Gestion d'exceptions** : pour améliorer la robustesse du code et éviter les plantages lors des erreurs utilisateur ou des lectures de fichiers.

- **Matplotlib** : pour générer des graphiques de statistiques (nombre d'emprunts, livres les plus lus...).
- **Tkinter** : pour une interface graphique conviviale avec des formulaires, des boutons, et des affichages organisés.

2. Analyse et conception :

Avant de passer à l'implémentation, il est important de modéliser l'architecture de l'application à l'aide d'un diagramme de classes UML. Cette représentation visuelle permet de définir clairement les entités principales, leurs attributs, ainsi que les relations entre elles. Elle sert de base à une conception orientée objet cohérente et structurée.

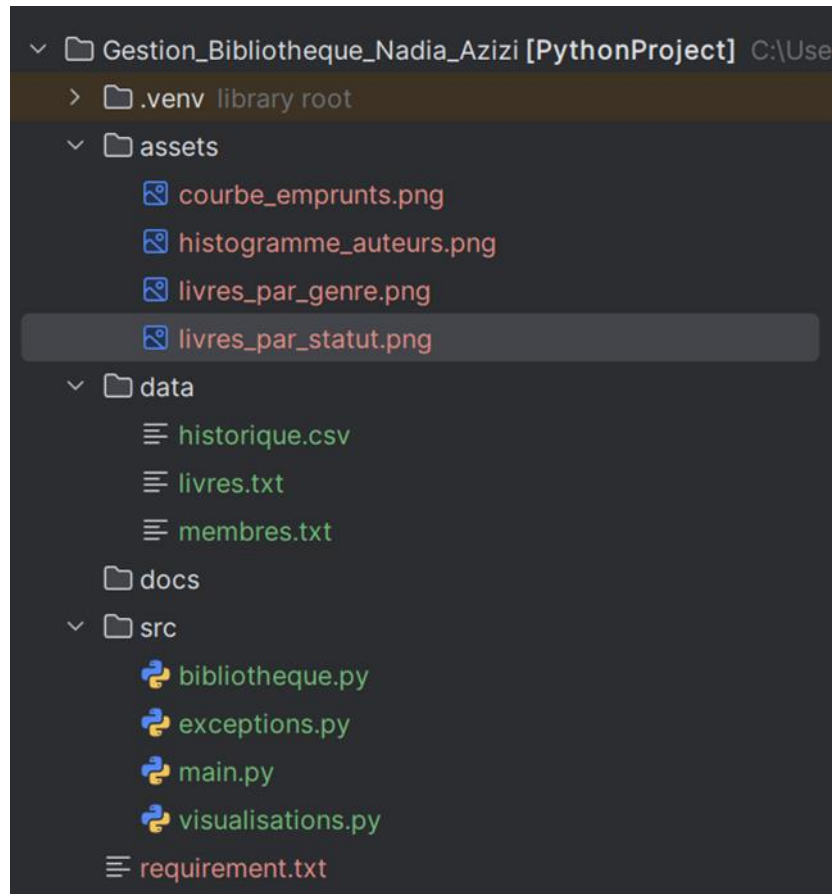


3. Structure du projet :

Le projet est organisé ainsi :

- **bibliotheque.py** : contient toutes les classes principales (Livre, Membre, Bibliothèque) et les méthodes pour gérer les livres, membres, emprunts et retours.

- **exception.py** : exceptions personnalisées.
- **main.py** : pour une interface graphique (**Tkinter**) et aussi le point d'entrée du programme .

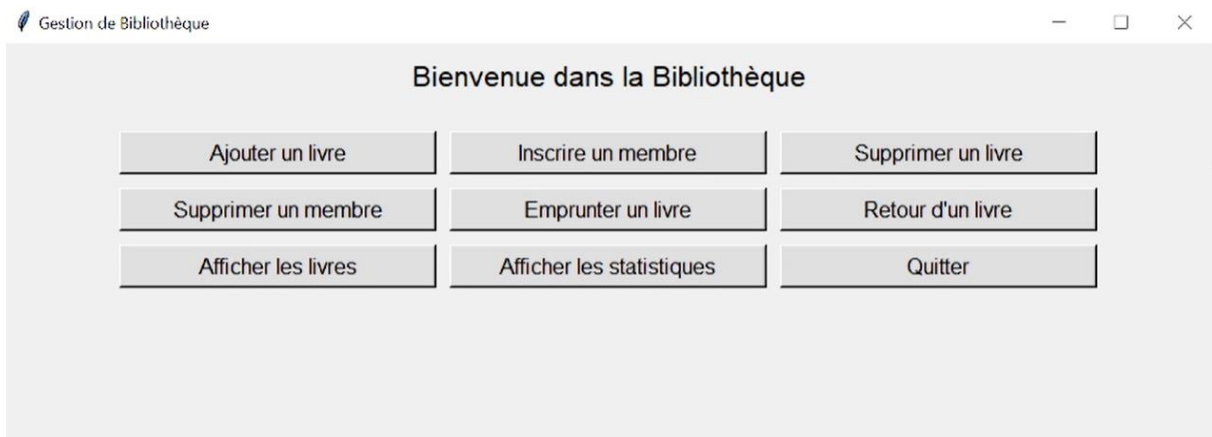


4. Fonctionnalités développées :

- o Ajouter/supprimer des livres.
- o Enregistrer des membres.
- o Enregistrer les emprunts/retours.
- o Sauvegarder/charger les données .
- o Afficher les statistiques (nombre de livres, emprunts, etc.) avec Matplotlib .
- o Interface graphique avec boutons et formulaires

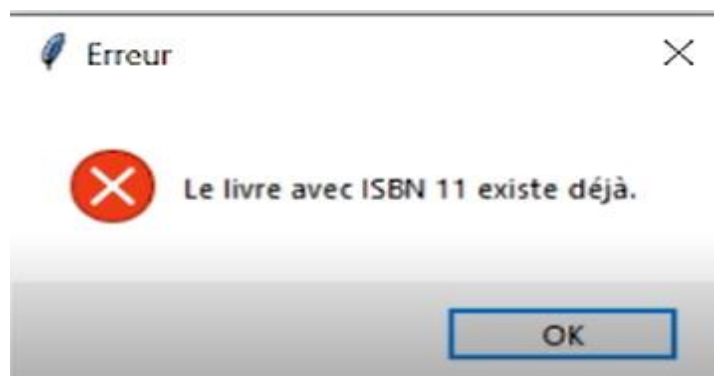
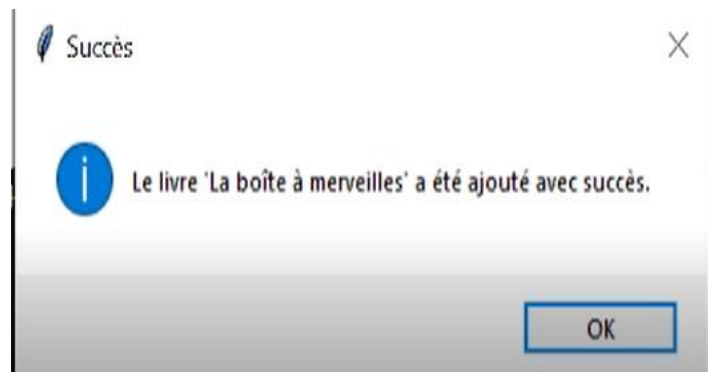
5. Description du projet :

a. Interface de début :



La première page de l'application est un menu interactif qui permet à l'utilisateur de naviguer facilement entre les différentes fonctionnalités de la bibliothèque.

b. Interface d'ajout d'un livre :



Cette interface offre à l'utilisateur la possibilité d'ajouter un nouveau livre à la bibliothèque en renseignant les informations suivantes :

- ISBN
- Titre du livre
- Auteur

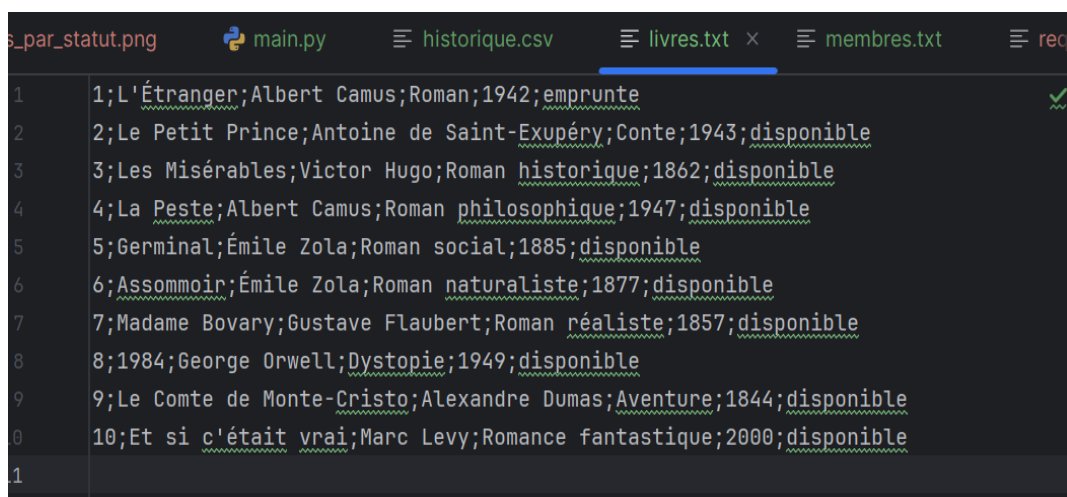
- Année de publication
- Genre

Une fois tous les champs saisis, l'utilisateur clique sur le bouton « **valider** ». Si l'ISBN est nouveau, un message de confirmation s'affiche pour indiquer que le livre a été enregistré avec succès. En revanche, si l'ISBN existe déjà dans le système, une alerte informe l'utilisateur que ce livre est déjà répertorié.

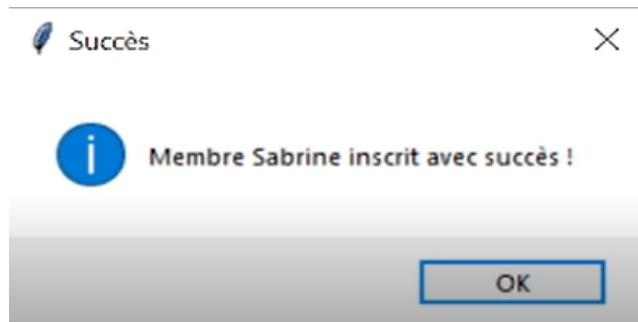
Un bouton "**Afficher les livres**" est également disponible : il permet de visualiser directement la liste des livres enregistrés après l'ajout.



Les données du livre sont **enregistrées dans le fichier livres.txt** :



c. Interface : Inscrire un membre :



Cette interface permet d'inscrire un nouvel utilisateur dans la bibliothèque. Elle contient des champs pour saisir :

- Un identifiant unique (ID)
- Le nom du membre

Une fois les champs remplis, l'utilisateur clique sur le bouton « Inscrire ». Si l'identifiant (ID) du membre est unique, un message de confirmation s'affiche pour indiquer que l'inscription a été effectuée avec succès. En revanche, si cet identifiant existe déjà dans le système, une alerte signale que ce membre est déjà enregistré.

Les données du membre sont **enregistrées dans le fichier membres.txt** :

```

bibliotheque.py exceptions.py visualisations.py main.py historique.csv livres.txt membres.txt x
1;AZIZI Nadia ;.
2;MAZRHAB Hanane ;.
3;NAIMI Zineb ;.
4;MOUMNI Fatima ;.
5;Sabrine;.

```

d. Interface : Emprunter un livre :

Cette interface permet à un membre d'emprunter un livre disponible dans la bibliothèque. L'utilisateur doit fournir deux informations :

- L'identifiant (ID) du membre déjà inscrit
- L'ISBN du livre qu'il souhaite emprunter

Une fois ces champs renseignés, un clic sur le bouton "Valider" lance une série de vérifications :

- Le membre existe dans la liste des membres
- Le livre existe dans la bibliothèque
- Le livre n'est pas déjà emprunté

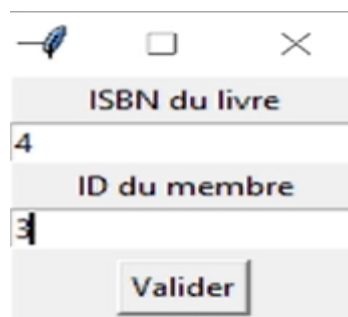
Si toutes les conditions sont remplies, un message de confirmation s'affiche.



e. Interface : Rendre un livre :

Cette interface permet à un membre de retourner un livre précédemment emprunté. L'utilisateur doit entrer :

- L'ID du membre
- L'ISBN du livre à rendre

Une interface utilisateur pour rendre un livre. Elle contient deux champs de saisie : "ISBN du livre" avec la valeur "4" et "ID du membre" avec la valeur "3". En bas, il y a un bouton "Valider".

Lorsque le bouton "Rendre" est cliqué, le programme vérifie :

- Que le membre existe
- Que le livre fait bien partie des livres empruntés par ce membre

Si les conditions sont respectées, le livre est marqué comme disponible à nouveau, et un message de confirmation apparaît :

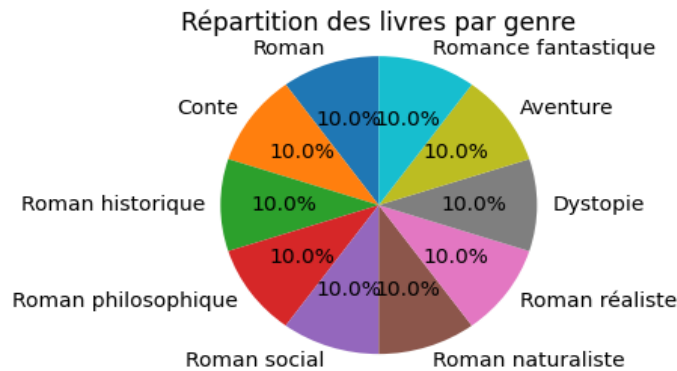


f. Interface : Visualisation des données :

L'application propose une interface dédiée à l'analyse visuelle des données de la bibliothèque. L'utilisateur peut y générer différents graphiques grâce à la bibliothèque Matplotlib.

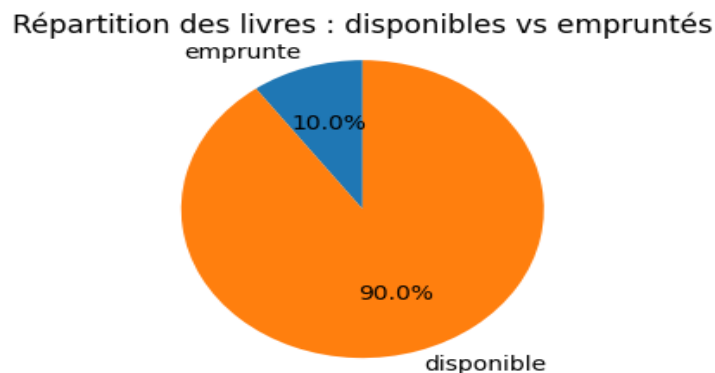
i. Diagramme circulaire : % de livres par genre :

Ce graphique montre la répartition des livres selon leur genre :



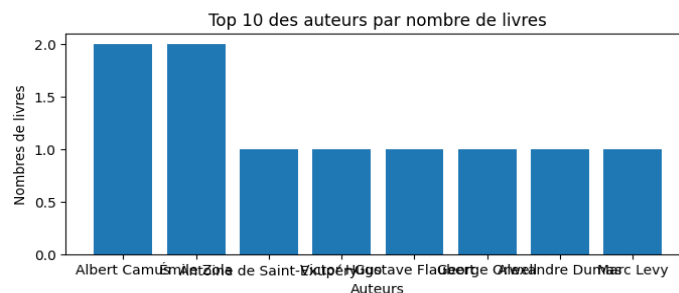
ii. Diagramme circulaire : % de livres disponibles :

Ce graphique montre le pourcentage des livres disponibles dans la bibliothèque :



iii. Histogramme : Répartition des auteurs :

Ce graphique permet de repérer les auteurs les plus présents dans la base de données :



- iv. Courbe d'activité : Historique des emprunts :
Cette courbe montre l'évolution des emprunts dans le temps :

