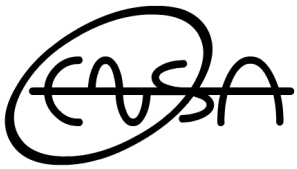
ENSAO - École Nationale des Sciences Appliquées
Filière Génie Informatique - GI3

Système de Gestion de Bibliothèque

Mini-projet de Programmation Avancée en Python



Réalisé par :
Souhail Moustaghit

Encadré par :
Pr. ZAKARIA HAJA

Programmation Avancée en Python

Année Universitaire 2024/2025

Table des matières

1 Introduction

Ce projet a pour but de créer une application de gestion de bibliothèque en Python, mettant en œuvre la POO, la persistance des données, la gestion des erreurs, les visualisations statistiques, et une interface utilisateur via ligne de commande.

2 Architecture du projet

L'organisation du projet suit une structure modulaire :

Listing 1 – Structure complète du projet Gestion de Bibliothèque

```
Gestion/
            src/
                  main.py
                  bibliotheque.py
                  exceptions.py
                  visualisation.py

            data/
                  livres.json
                  membres.json
                  historique.csv

            assets/
                  genres.png
                  top_auteurs.png
                  activite_emprunts.png

            docs/

            README.md

            requirements.txt

            .gitignore
```

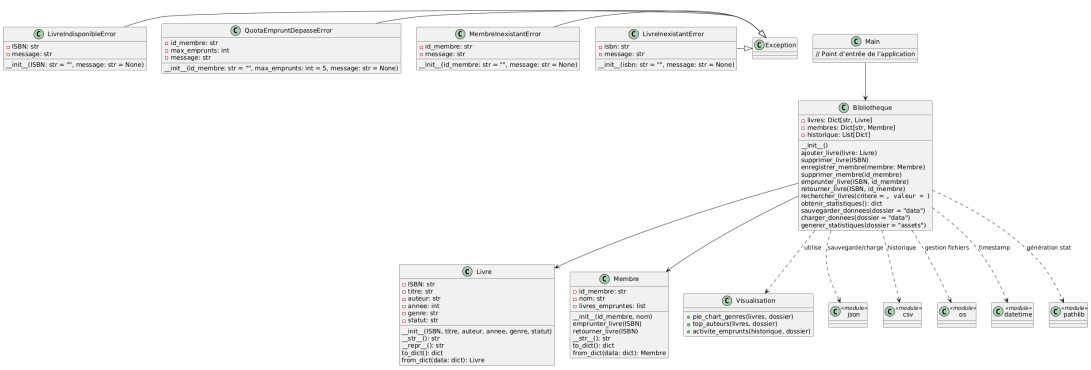


FIGURE 1 – Diagramme de classe- UML

3 Modules principaux

bibliotheque.py

— `Livre`
— `Membre`
— `Bibliotheque`

exceptions.py

Exceptions personnalisées :
— `LivreIndisponibleError`
— `QuotaEmpruntDepasseError`
— `MembreInexistantError`
— `LivreInexistantError`

visualisations.py

Utilise Matplotlib pour :
— Affichage des genres (camembert)
— Top auteurs (barres)
— Emprunts récents (courbe temporelle)

4 Gestion des fonctionnalités avancées dans la classe `Bibliotheque`

4.1 Gestion des emprunts et des retours

La méthode `emprunter_livre` permet à un membre d'emprunter un livre en respectant plusieurs règles de gestion :
— Vérification de l'existence du livre et du membre.
— Vérification de la disponibilité du livre.
— Vérification du quota d'emprunts (maximum 5 livres).
— Mise à jour du statut du livre (`disponible` → `emprunté`).
— Enregistrement de l'action dans l'historique (date, ISBN, membre, action).

La méthode `retourner_livre` fonctionne de manière similaire mais inverse les opérations : le livre devient à nouveau `disponible` et l'action `retour` est ajoutée à l'historique.

4.2 Système d'historique

L'attribut `historique` est une liste contenant des dictionnaires représentant chaque opération effectuée (emprunt ou retour), stockant :
— la date de l'opération (format ISO),
— l'ISBN du livre,
— l'identifiant du membre,
— le type d'action (`emprunt` ou `retour`).

Cette structure permet de tracer toutes les interactions avec les livres, facilitant l'analyse de l'activité.

4.3 Sauvegarde et chargement des données

La méthode `sauvegarder_donnees` utilise des fichiers au format JSON pour sauvegarder les livres et les membres, et le format CSV pour l'historique. Elle vérifie la présence du dossier cible et le crée si nécessaire.

La méthode `charger_donnees` lit ces fichiers et reconstruit les objets `Livre` et `Membre` grâce aux méthodes `from_dict`. Cette approche permet une persistance simple des données de la bibliothèque entre les exécutions du programme.

4.4 Génération de statistiques visuelles

La méthode `generer_statistiques` exploite la classe `Visualisation` pour produire des graphiques informatifs :

— Un diagramme circulaire des genres de livres.
— Un classement des auteurs les plus fréquents.
— Une courbe d'activité des emprunts dans le temps.

Chaque appel est encapsulé dans un bloc `try/except` afin d'éviter qu'une erreur dans un graphique n'interrompe la génération complète. Le tout est exporté dans un dossier (par défaut `assets`).

5 Persistance des données

Les données sont sauvegardées automatiquement :
— `membres.json`
— `livres.json`, `historique.csv`
La fonction `charger_donnees()` initialise l'application à partir des fichiers.

6 Visualisations

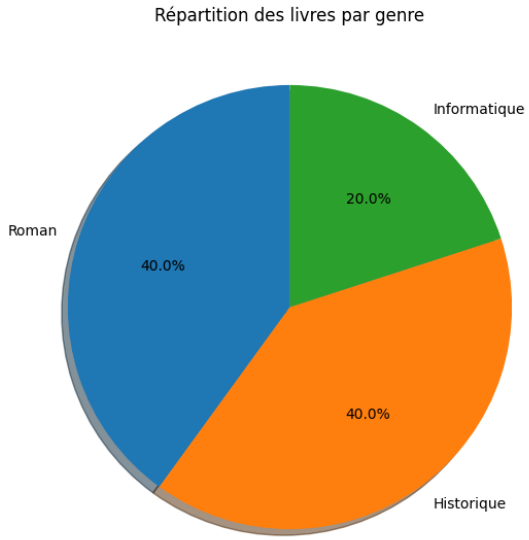


FIGURE 2 – Répartition des livres par genre

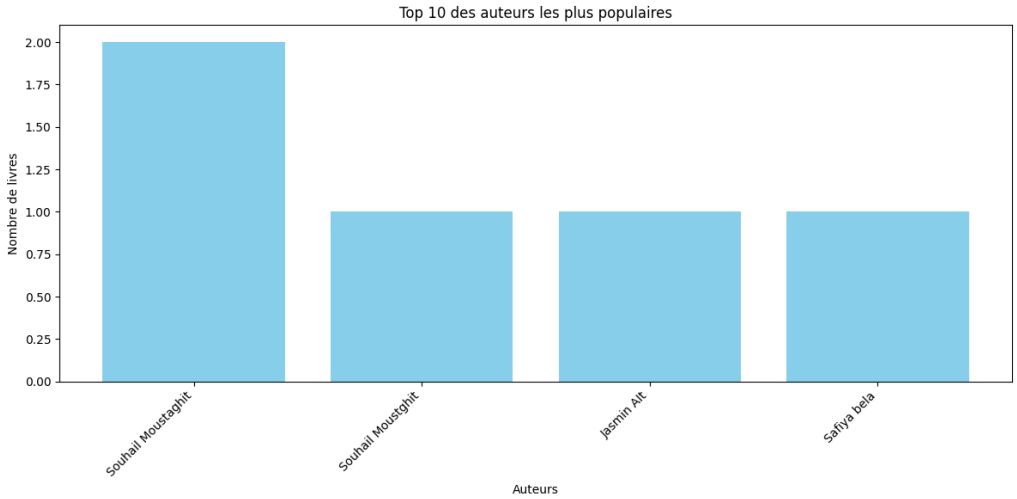


FIGURE 3 – Top 10 des auteurs populaires

7 Interface en ligne de commande

Fonctionnement

Menu principal :

— Ajouter/supprimer un livre

— Enregistrer/supprimer un membre
— Emprunter ou retourner ou rechercher un livre
— Voir les statistiques

Listing 2 – Extrait du menu CLI

```
def afficher_menu():
    print("\n====== MENU BIBLIOTH QUE ======")
    print("1. Ajouter un livre")
    print("2. Supprimer un livre")
    print("3. Enregistrer un membre")
    print("4. Supprimer un membre")
    print("5. Emprunter un livre")
    print("6. Retourner un livre")
    print("7. Rechercher un livre")
    print("8. G n rer les statistiques visuelles")
    print("9. Sauvegarder les donn es")
    print("10. Charger les donn es")
    print("0. Quitter")
```

Capture d'écran

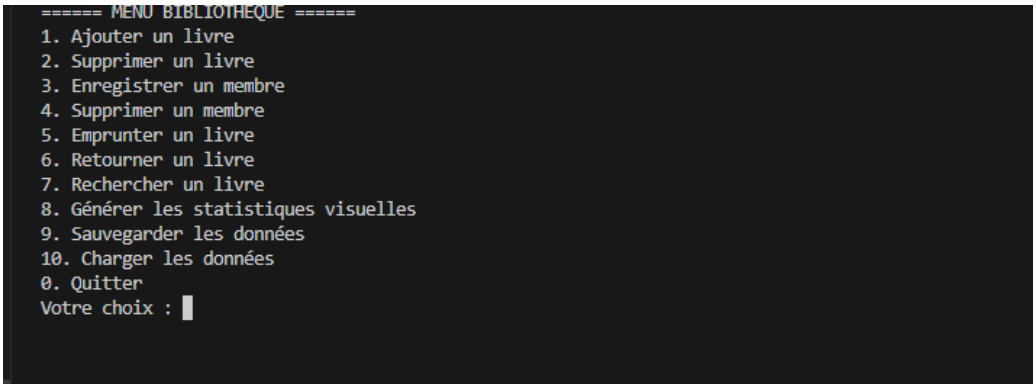


FIGURE 4 – Capture d'écran de l'interface CLI

8 Conclusion

Projet respectant toutes les exigences : POO, erreurs, visualisations, persistance, et CLI. Pistes d'amélioration :

— Interface graphique complète
— Recommandation intelligente avec IA
— Application desktop avec exécutable

Lien GitHub : https://github.com/souhailmstg/Gestion_De_Bibliotheque