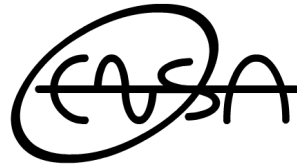


ENSAO - École Nationale des Sciences Appliquées  
Filière Génie Informatique - GI3

# 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

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Architecture du projet</b>	<b>2</b>
<b>3</b>	<b>Modules principaux</b>	<b>3</b>
<b>4</b>	<b>Persistance des données</b>	<b>3</b>
<b>5</b>	<b>Visualisations</b>	<b>4</b>
<b>6</b>	<b>Interface en ligne de commande</b>	<b>4</b>
<b>7</b>	<b>Conclusion</b>	<b>5</b>

# 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

```

1 Gestion/
2   src/
3       main.py
4       bibliotheque.py
5       exceptions.py
6       visualisation.py
7
8   data/
9       livres.json
10      membres.json
11      historique.csv
12
13  assets/
14      genres.png
15      top_auteurs.png
16      activite_emprunts.png
17
18  docs/
19
20  README.md
21
22  requirements.txt
23
24  .gitignore
  
```

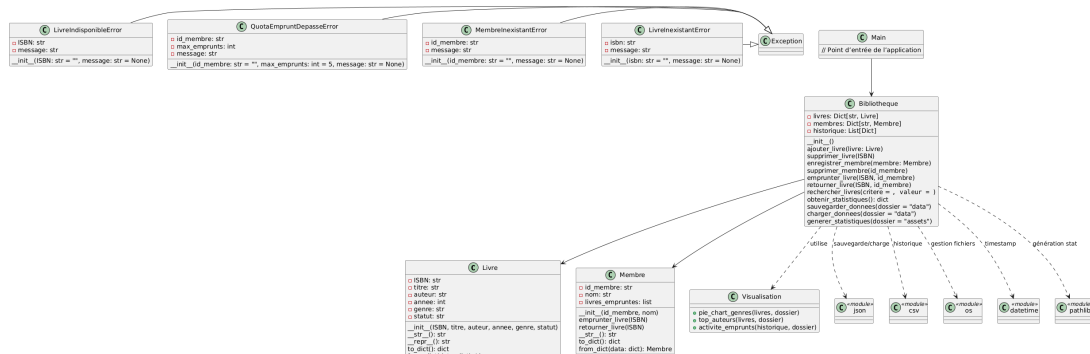


FIGURE 1 – Diagramme de classe- UML

## 3 Modules principaux

### bibliotheque.py

- Livre
- Membre
- Bibliotheque

Listing 2 – Extrait de classe Livre

```
1 class Livre:
2     def __init__(self, ISBN, titre, auteur, annee, genre, statut):
3         self.ISBN=ISBN
4         self.titre=titre
5         self.auteur=auteur
6         self.annee=annee
7         self.genre=genre
8         self.statut=statut
```

### 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 Persistance des données

Les données sont sauvegardées automatiquement :

- livres.txt, membres.txt
- livres.json, historique.csv

La fonction `charger_donnees()` initialise l'application à partir des fichiers.

## 5 Visualisations

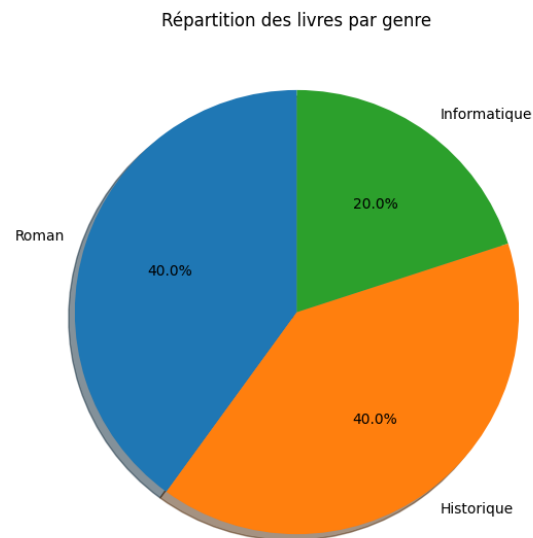


FIGURE 2 – Répartition des livres par genre

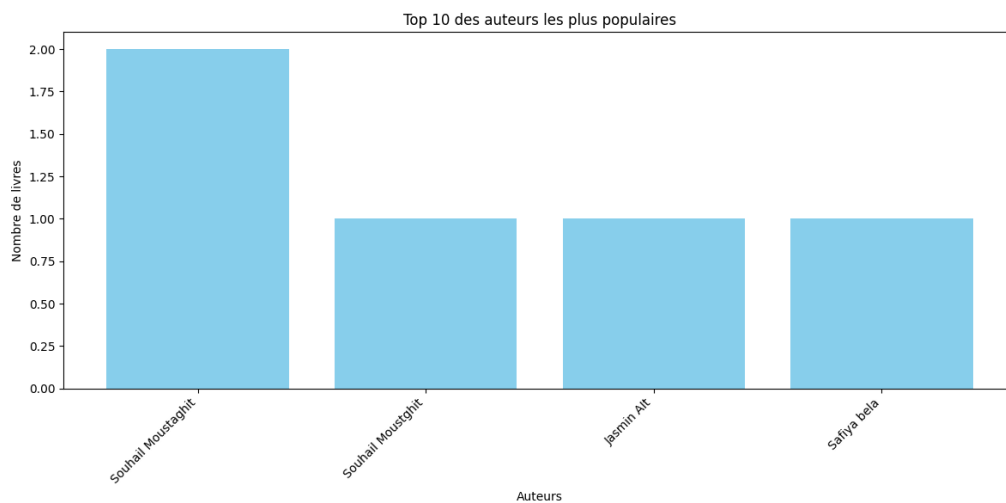


FIGURE 3 – Top 10 des auteurs populaires

## 6 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 3 – Extrait du menu CLI

```
1 def afficher_menu():
2     print("\n===== MENU BIBLIOTHEQUE =====")
3     print("1. Ajouter un livre")
4     print("2. Supprimer un livre")
5     print("3. Enregistrer un membre")
6     print("4. Supprimer un membre")
7     print("5. Emprunter un livre")
8     print("6. Retourner un livre")
9     print("7. Rechercher un livre")
10    print("8. Générer les statistiques visuelles")
11    print("9. Sauvegarder les données")
12    print("10. Charger les données")
13    print("0. Quitter")
```

## Capture d'écran

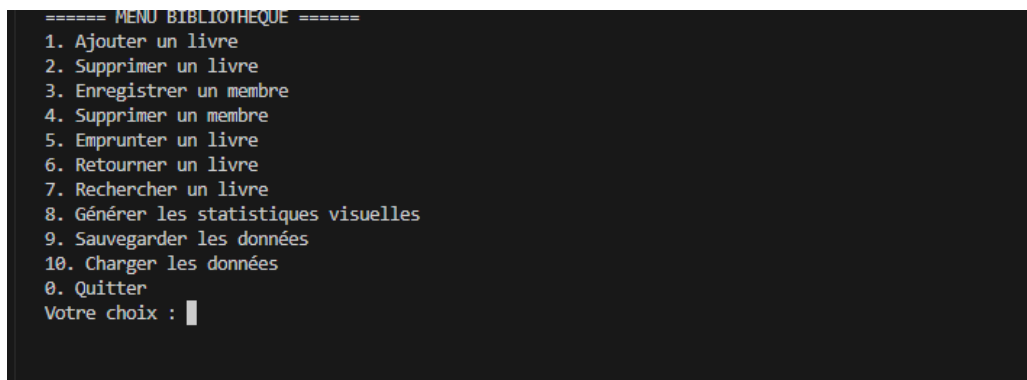


FIGURE 4 – Capture d'écran de l'interface CLI

## 7 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](https://github.com/souhailmstg/Gestion_De_Bibliotheque)