



Rapport du Mini-Projet :

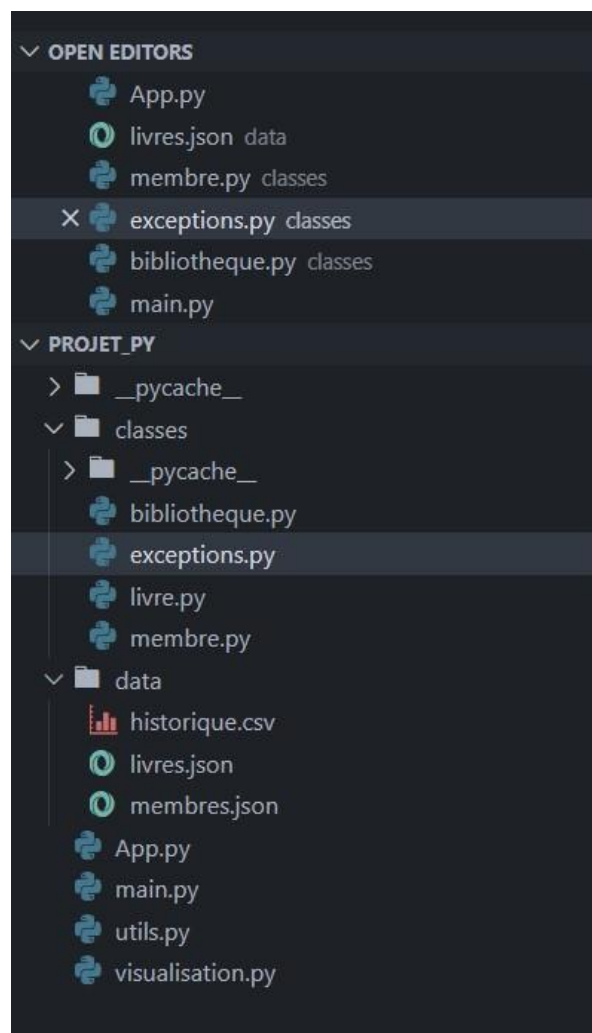
Système de Gestion de Bibliothèque

1. Objectif du Projet

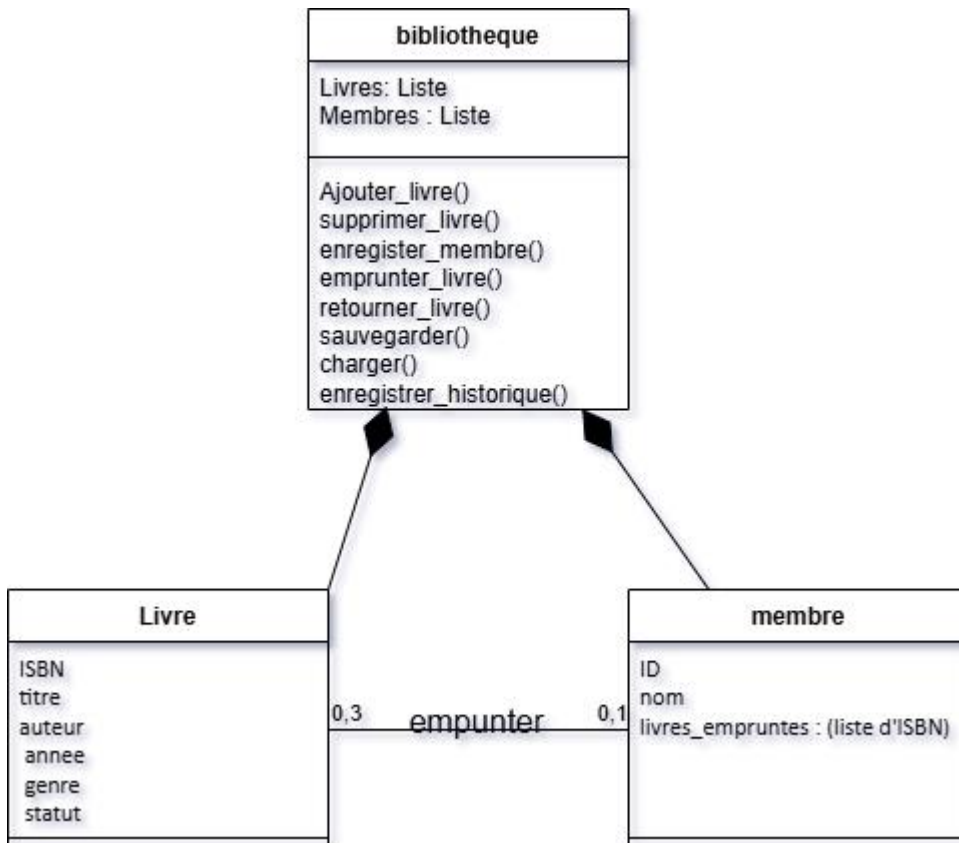
Ce mini-projet consiste à développer un système complet de gestion d'une bibliothèque en Python en respectant les paradigmes de la POO, en intégrant la persistance des données (JSON/CSV), la gestion des erreurs personnalisées, et une interface graphique conviviale avec Tkinter (ttkbootstrap). Un module de statistiques est aussi inclus grâce à Matplotlib.

2. Architecture et Organisation

a. Arborescence du Projet



b. Diagramme de Classe (UML)



3. Fonctionnalités Implémentées

a. Gestion des Livres

- Ajout / mise à jour / suppression
- Validation des champs (vide, doublons)

b. Gestion des Membres

- Ajout de membres
- Emprunt de livre (max 3)
- Retour de livre
- Vérification de disponibilité / erreurs levées

c. Interface Graphique (Tkinter + ttkbootstrap)

- Onglets : Livres / Membres / Statistiques
- Tableaux dynamiques (Treeview)
- Boutons avec feedback utilisateur • Messages d'erreurs stylisés

d. Gestion des Exceptions

- LivreIndisponibleError
- QuotaEmpruntDepasseError
- MembreInexistantError

- LivreInexistantError

-

e. Persistance des Données

- fichiers : livres.json , membres.json , historique.csv
- sauvegarde et chargement automatiques

f. Visualisation Statistique (Matplotlib)

- Diagramme circulaire des genres
- Histogramme Top 10 des auteurs
- Courbe temporelle : emprunts sur 30 jours
- Intégration dans l'interface sans ouvrir de fenêtre externe

4. Difficultés et Solutions

Difficulté	Solution
Encodage UTF-8 mauvais affichage	Ajout encoding='utf-8' lors de la lecture/criture
Rafraîchissement des graphes	Suppression préalable de l'ancien canvas + plt.close()
Interface Tkinter trop chargée	Utilisation de ttk.Labelframe , padding , font, bootstyle
Données dupliquées / vides	Validation stricte avant ajout

5. Perspectives d'Amélioration

quelques idées pour améliorer ce projet à l'avenir :

- Ajout d'un système de connexion avec authentification (admin/utilisateur).
- Possibilité de rechercher des livres ou des membres avec des filtres avancés.
- Intégration d'une base de données SQL (ex. SQLite ou MySQL) pour une meilleure gestion à grande échelle.
- Notification de rappel pour les retards de retour.
- Génération automatique de rapports PDF pour l'historique des emprunts.
- Ajout d'une fonction d'exportation des statistiques sous forme d'image ou de rapport.

6. Conclusion

Ce projet a permis de mettre en pratique les concepts avancés de Python : POO, gestion d'exceptions, manipulation de fichiers, création d'interface graphique et visualisation de données. L'utilisation de bibliothèques modernes comme `ttkbootstrap` et `matplotlib` a enrichi l'interface et l'expérience utilisateur.

7. Remerciements

Je souhaite exprimer ma gratitude à **Monsieur ZAKARIA HAJA** pour son encadrement, ses conseils pertinents et la qualité de son enseignement durant ce module de **Programmation Avancée en Python**, qui ont grandement contribué à la réussite de ce mini-projet.