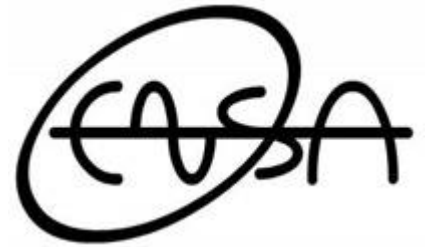




**École Nationale des
Sciences Appliquées
d'Oujda**



RAPPORT DU PROJET

Programmation avancée en Python

Réalisé par :

LAMYAE HAMDAOUI

Filière :

Génie Informatique

Encadré par :

Ms. HAJJA ZAKARIA

Année universitaire : 2024/2025

Table des matières

1 .Introduction :	3
2.Diagramme UML :	3
3. Détails des Algorithmes Clés	4
3.1. Supprimer un livre	4
3.2. Emprunter un livre	4
3.3. Retourner un livre	4
3.4. Trouver un membre / un livre	4
3.5. Sauvegarder les données	4
3.6. Charger les données	4
3.7.Visualisations Statistiques	5
Diagramme circulaire	5
Histogramme	5
4 . Captures d'écran des visualisations	5
5 . Difficultés rencontrées et solutions	8
6. Conclusion	8

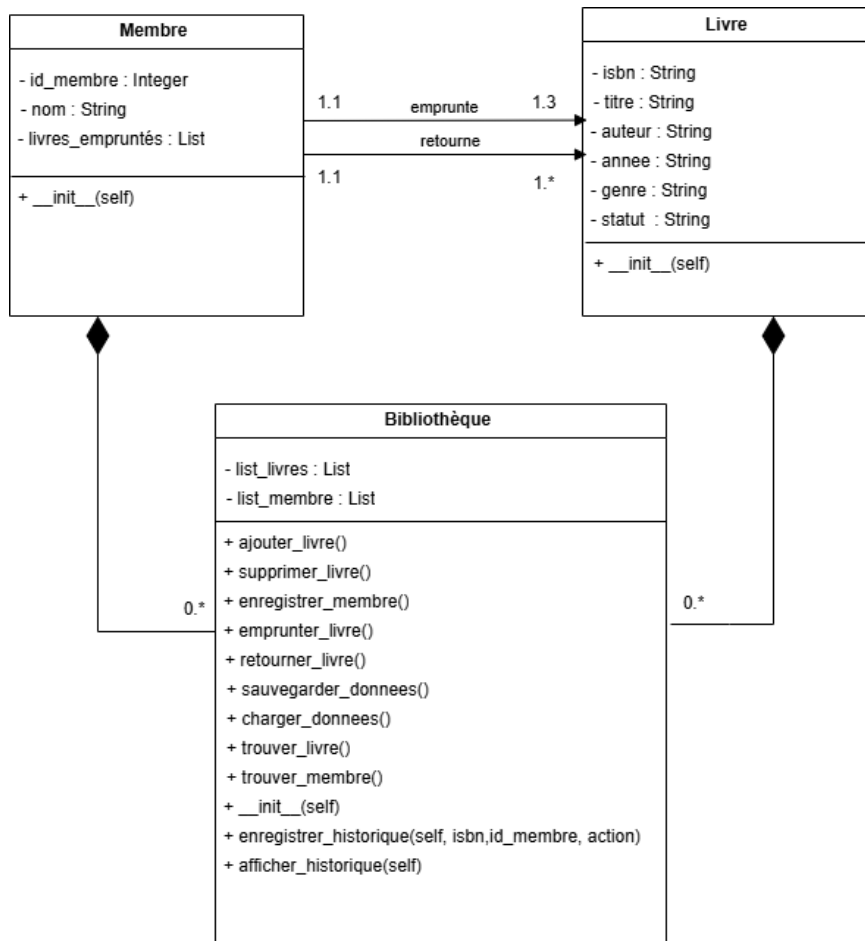
1. Introduction :

Cette application consiste à développer un système de gestion de bibliothèque en Python en utilisant la Programmation Orientée Objet (POO) . Il permet de gérer les livres, les membres, les emprunts et les retours de livres.

Le projet assure :

- La gestion des exceptions pour les erreurs métier
- La persistance des données dans des fichiers
- La génération de rapports statistiques avec Matplotlib
- La persistance des données (fichiers JSON/CSV)

2. Diagramme UML :



3.Détails des Algorithmes Clés

3.1. Supprimer un livre

Pour supprimer un livre : On parcourt la liste des livres de la bibliothèque, en comparant l'ISBN du livre courant avec celui donné en paramètre. Si on le trouve, on le supprime avec la méthode `remove()`, sinon, on lève l'exception `LivreInexistentError`, ce qui signifie que l'ISBN fourni n'existe pas dans la collection.

3.2. Emprunter un livre

Pour emprunter un livre : On commence par rechercher le membre via son ID, puis le livre via son ISBN. Si le livre est disponible (`statut == "disponible"`), on l'ajoute à la liste `livres_empruntes` du membre et on met à jour le statut du livre à "emprunté". Sinon, on lève l'exception `LivreIndisponibleError`. Et si le membre a déjà emprunté 3 livres, on lève l'exception `QuotaEmpruntDepasseError`, indiquant qu'il a atteint la limite maximale d'emprunts autorisés.

3.3. Retourner un livre

C'est presque la même procédure que Emprunter. **Pour retourner un livre :** On recherche le membre et le livre correspondants à partir des identifiants donnés en paramètres. Si le livre existe dans la liste des livres empruntés par ce membre : On le retire de cette liste, et on change son statut à "disponible".

3.4. Trouver un membre / un livre

Pour trouver un membre : On parcourt la liste des membres en comparant chaque identifiant (id) à celui donné en paramètre. Si un membre correspondant est trouvé, on le retourne, sinon, on lève l'exception `MembreInexistentError`.

Même logique pour trouver un livre : on recherche dans la liste des livres selon l'ISBN. Si non trouvé, on lève `LivreInexistentError`.

3.5. Sauvegarder les données

Pour sauvegarder les données : On ouvre `livres.txt` en mode écriture, puis on parcourt la liste des livres. Pour chaque livre, on écrit une ligne contenant ses attributs, séparés par ;.

On fait de même pour les membres dans `membres.txt`, en ajoutant aussi la liste des ISBN des livres qu'ils ont empruntés.

3.6. Charger les données

Pour charger les données : D'abord on initialise la liste des livres vide. Puis on ouvre `livres.txt`, on lit ligne par ligne, on extrait les champs, puis on instancie des objets `Livre` qu'on ajoute à la liste. Même procédure pour `membres.txt`.
Si les fichiers n'existent pas, une exception `FileNotFoundError` est levée

3.7. Visualisations Statistiques

Les statistiques sont générées à l'aide de **matplotlib**, on a :

⇒ Diagramme circulaire : Représente la proportion de livres par genre.

```
plt.pie(counts.values(), labels=counts.keys(), autopct='%1.1f%%')
```

⇒ Histogramme : Affiche le top 10 des auteurs les plus populaires dans la bibliothèque.

```
plt.bar(noms, freqs)
```

4. Captures d'écran des visualisations :

On peut visualiser sur la **ligne de commande** ou bien sur les interfaces **Tkinter** :

- Ligne de commande :

```
C:\Users\pc\Desktop\s6\python\projet>python src/main.py

==== GESTION BIBLIOTHEQUE ====
1. Ajouter un livre
2. Supprimer un livre
3. Enregistrer un membre
4. Emprunter un livre
5. Retourner un livre
6. Sauvegarder
7. Statistiques
8. Afficher l'historique
9. Quitter
Choix : █
```

```
Choix : 1
ISBN: 978
Titre: Le Horla
Auteur: Guy de Maupassant
Année: 1887
Genre: Nouvelle fantastique
Livre ajouté.
```

```
Choix : 4
ID membre: 21
ISBN livre: 978
Livre emprunté.
```

```
Choix : 5
Choix : 9
On a quitté.
```

```
C:\Users\pc\Desktop\s6\python\projet>█
```

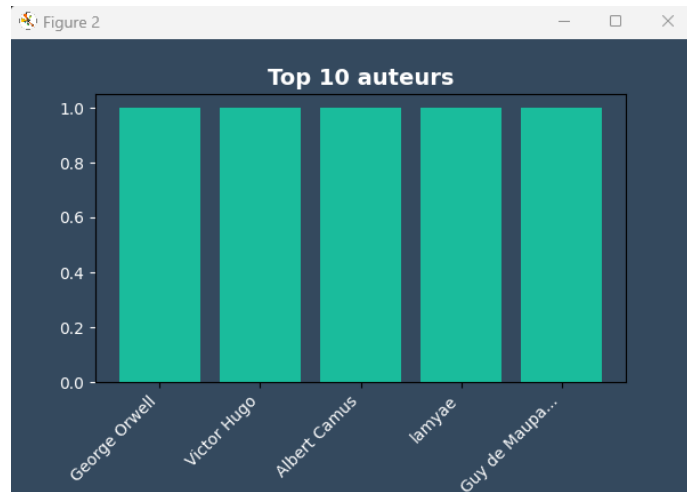
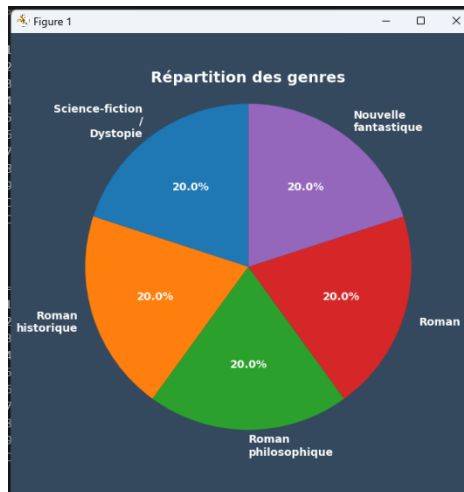
```
Choix : 2
ISBN du livre à supprimer: 978
Livre supprimé.
```

```
Choix : 3
ID membre: 21
Nom membre: Emma Durand
Membre enregistré.
```

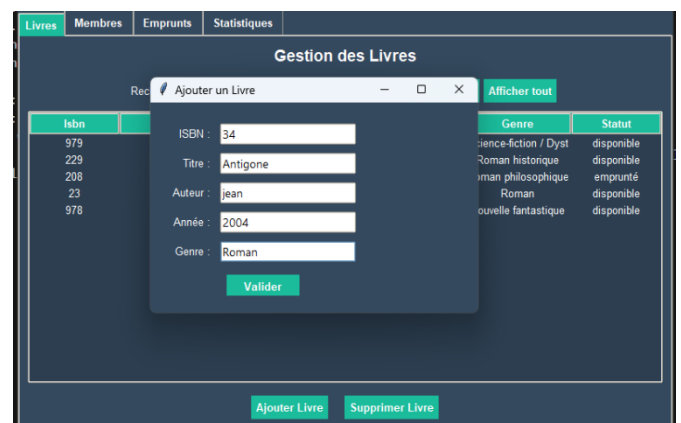
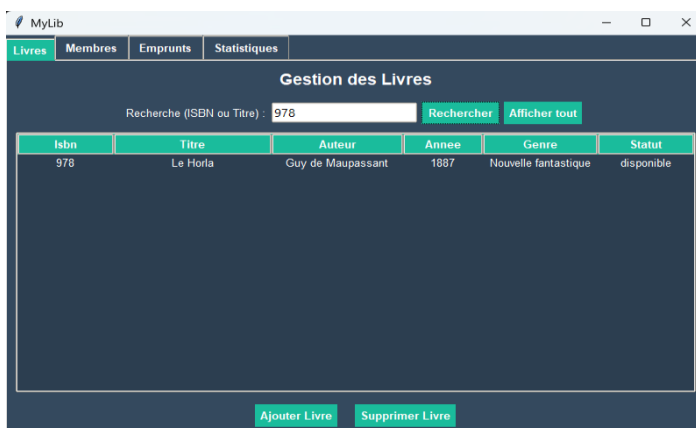
```
Choix : 8
2025-06-25,208,M001,emprunté
2025-06-25,23,23,emprunté
2025-06-25,23,23,retourné
2025-06-26,978,21,emprunt
2025-06-26,978,21,retour
```

Choix : 7

```
C:\Users\pc\Desktop\s6\python\projet\src\visualisation.py:48: UserWarning: set_ticklabels() should only be used with a fixed number of ticks, i.e. after set_ticks() or using a FixedLocator.  
  ax.set_xticklabels(noms_courts, rotation=45, ha='right', fontsize=10, color='white') #appliquer ces noms sur  
l'axe x avec 45°
```



- Interface Tkinter :



LivresMembresEmpruntsStatistiques

Gestion des Livres

Recherche (ISBN ou Titre) : 978RechercherAfficher tout

Isbn	Titre	Auteur	Annee	Genre	Statut
979	"1984"	George Orwell	1949	Science-fiction / Dystopie	disponible
229	Les Misérables	Victor Hugo	1862	Roman historique	disponible
208	L'Étranger	Albert Camus	1942	Roman philosophique	emprunté
23	Le H	lamyae	2009	Roman	disponible
978	Le Horla			Nouvelle fantastique	disponible
34	Antigone			Roman	disponible

Succès
Livre ajouté avec succès.
OK

Ajouter LivreSupprimer Livre

LivresMembresEmpruntsStatistiques

Gestion des Livres

Recherche (ISBN ou Titre) : 978RechercherAfficher tout

Isbn	Titre	Auteur	Annee	Genre	Statut
979	"1984"	George Orwell	1949	Science-fiction / Dystopie	disponible
229	Les Misérables	Victor Hugo	1862	Roman historique	disponible
208	L'Étranger	Albert Camus	1942	Roman philosophique	emprunté
23	Le H	lamyae	2009	Roman	disponible
978	Le Horla			Nouvelle fantastique	disponible
34	Antigone			Roman	disponible

Confirmation
Voulez-vous vraiment supprimer ce livre ?
OuiNon

Ajouter LivreSupprimer Livre

LivresMembresEmpruntsStatistiques

Gestion des Membres

ID Membre : 45
Nom : Issam Hamdaoui
Enregistrer MembreRechercherAfficher tout

Id	Nom	Livres empruntés
M001	Jean Dupont	208
M002	Marie Martin	
M003	Lucie Bernard	
23	oumeyma	
21	Emma Durand	

Supprimer Membre

LivresMembresEmpruntsStatistiques

Gestion des Membres

ID Membre :
Nom :
Enregistrer MembreRechercherAfficher tout

Succès
Membre enregistré.
OK

Id	Nom	Livres empruntés
M001	Jean Dupont	208
M002	Marie Martin	
M003	Lucie Bernard	
23	oumeyma	
21	Emma Durand	
45	Issam Hamdaoui	

Supprimer Membre

LivresMembresEmpruntsStatistiques

Gestion des Emprunts

ID Membre : 21
ISBN Livre : 34
Succès
Livres retourné.
OK

LivresMembresEmpruntsStatistiques

Gestion des Emprunts

ID Membre : 21
ISBN Livre : 34
Succès
Livres emprunté.
OK

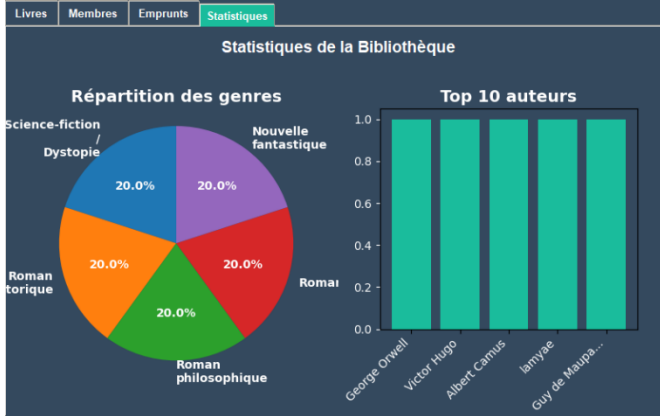
LivresMembresEmpruntsStatistiques

Gestion des Membres

ID Membre : 21
Nom : Emma Durand
Enregistrer MembreRechercherAfficher tout

Id	Nom	Livres empruntés
21	Emma Durand	

Supprimer Membre



5 . Difficultés rencontrées et solutions

Première expérience avec un projet Python complet

La principale difficulté rencontrée durant ce projet est que c'était ma **première fois que je développais une application complète en Python**.

Au début, il n'était pas évident de bien comprendre l'ensemble du processus, notamment la structure modulaire, la logique métier et les interactions entre les classes. Mais en pratiquant régulièrement et en analysant le ligne par ligne, j'ai fini par mieux comprendre la structure et le fonctionnement global du programme.

Compréhension du stockage avec fichiers texte / JSON

J'ai également rencontré des difficultés à comprendre comment stocker les données dans des **fichiers texte (.txt)** et **CSV**, au lieu d'utiliser une base de données classique.

Au départ, cela me semblait complexe, mais après avoir effectué des recherches et regardé des tutoriels, j'ai compris le fonctionnement de ce type de persistance.

Pour les méthodes de sauvegarde et de chargement des données, je me suis partiellement inspirée d'exemples proposés par une intelligence artificielle. Cela m'a permis de mieux comprendre la manière appropriée d'écrire et de stocker dans les fichiers.

Découverte de la bibliothèque Tkinter

Une autre difficulté importante a été l'utilisation de **Tkinter** pour créer une interface graphique.

C'était la première fois que je découvrais cette bibliothèque, et je n'avais jamais travaillé avec des interfaces graphiques en python auparavant.

Pour dépasser ces difficultés : Premièrement, j'ai visionné plusieurs tutoriels, notamment ceux proposés dans l'énoncé du projet. Deuxièmement, j'ai construit les interfaces étape par étape (onglets, formulaires, boutons, etc.). Et J'ai également utilisé des générateurs d'interfaces via l'IA pour m'inspirer et mieux comprendre le rôle de chaque élément.

Même si je ne comprends pas encore tout dans le code, cette méthode m'a aidée à bien commencer et je pourrai mieux comprendre avec le temps.

6. Conclusion

Ce projet de gestion de bibliothèque a été une bonne expérience pour moi. Il m'a permis d'appliquer ce que j'ai appris sur la programmation orientée objet en Python, ainsi que de gérer les erreurs grâce aux exceptions personnalisées. J'ai aussi appris à utiliser des fichiers texte pour sauvegarder et charger les données, ce qui est pratique pour un petit projet sans avoir besoin d'une base de données. J'ai aussi découvert comment afficher des graphiques avec Matplotlib, ce qui m'a aidée à visualiser des statistiques comme les genres les plus lus ou les auteurs les plus populaires.

Créer une interface graphique avec Tkinter a été un défi, car c'était ma première expérience dans ce domaine. Cependant, cela m'a permis d'acquérir de nouvelles compétences et de progresser.

En conclusion, ce projet me fournit une base solide que je pourrai réutiliser et améliorer à l'avenir, par exemple en développant une interface plus enrichissante ou en ajoutant de nouvelles fonctionnalités.