

RAPPORT MINI-PROJET PYTHON

SUJET: APPLICATION DE GESTION DE BIBLIOTHEQUE
GI3_S2

Date : 28-06-2025

Cree par :
Amine EL MAHI

Encadre par :
M.HAJA Zakaria

Introduction:

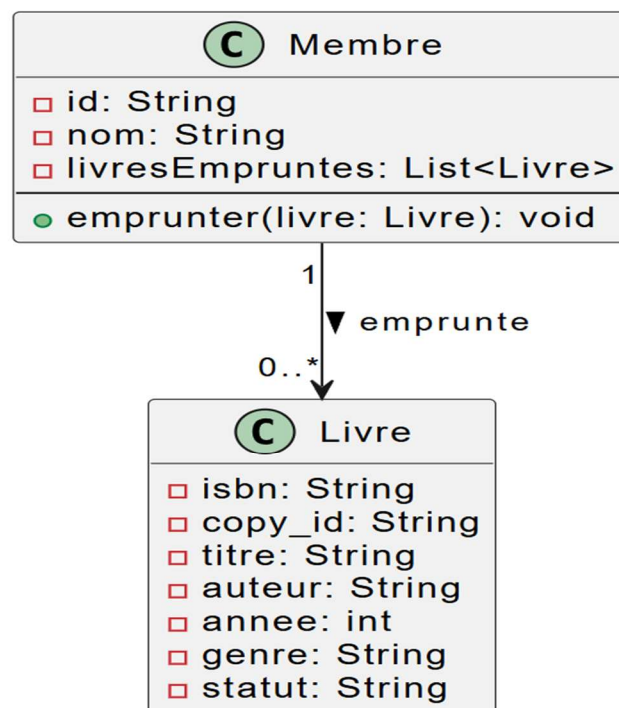
Dans le cadre de ce projet, nous proposons la conception et le développement d'une **application de gestion de bibliothèque** destinée exclusivement au **personnel de la bibliothèque** (et non aux lecteurs ou clients directs). L'objectif principal est de fournir un outil informatique simple, efficace et structuré permettant de gérer les membres inscrits ainsi que les livres disponibles à l'emprunt.

Chaque **membre** de la bibliothèque est identifié par un identifiant unique et possède un nom, ainsi qu'une liste des exemplaires de livres qu'il a empruntés. Le processus d'emprunt consiste à associer à un membre une **copie précise** d'un livre.

Dans notre système, un **livre** est modélisé à travers ses **exemplaires physiques**. Autrement dit, deux exemplaires du même livre (même ISBN) sont considérés comme deux entités distinctes grâce à leur identifiant de copie (`copy_id`). Chaque exemplaire contient également des informations bibliographiques telles que le **titre**, l'**auteur**, l'**année de publication**, le **genre**, et un **statut** indiquant s'il est disponible, emprunté ou réservé.

Voici Le Diagramme de Class simplifiée

Diagramme de classes - Système de gestion de bibliothèque



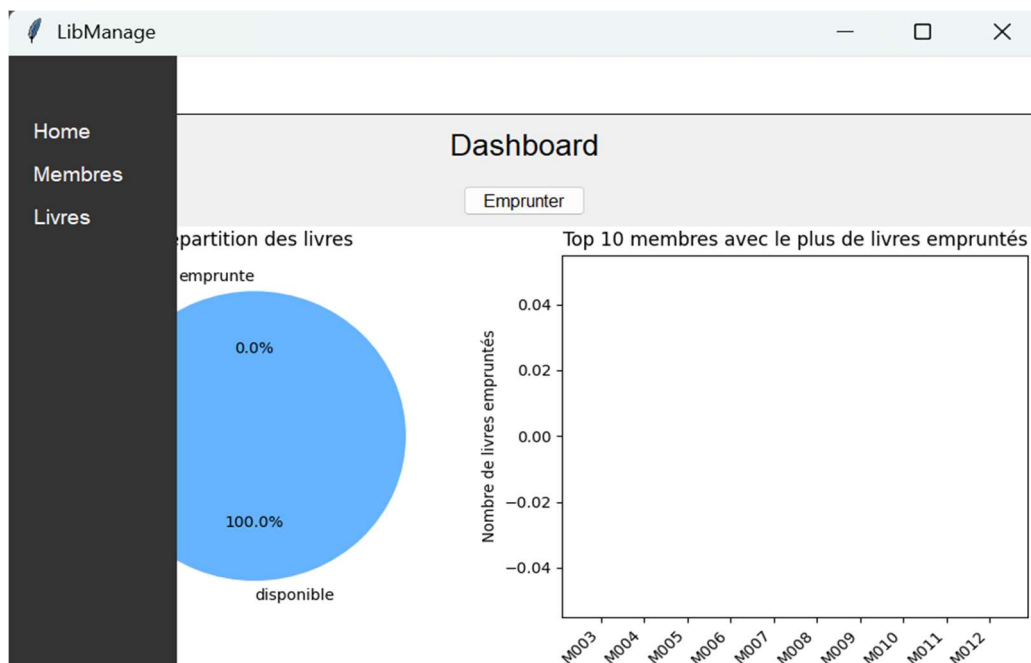
Les différents services de l'application :

L'application permet donc au personnel de :

- **Ajouter des membres,**
- **Enregistrer des livres (par exemplaire),**
- **Gérer les emprunts et retours,**
- **Consulter les livres empruntés par chaque membre.**

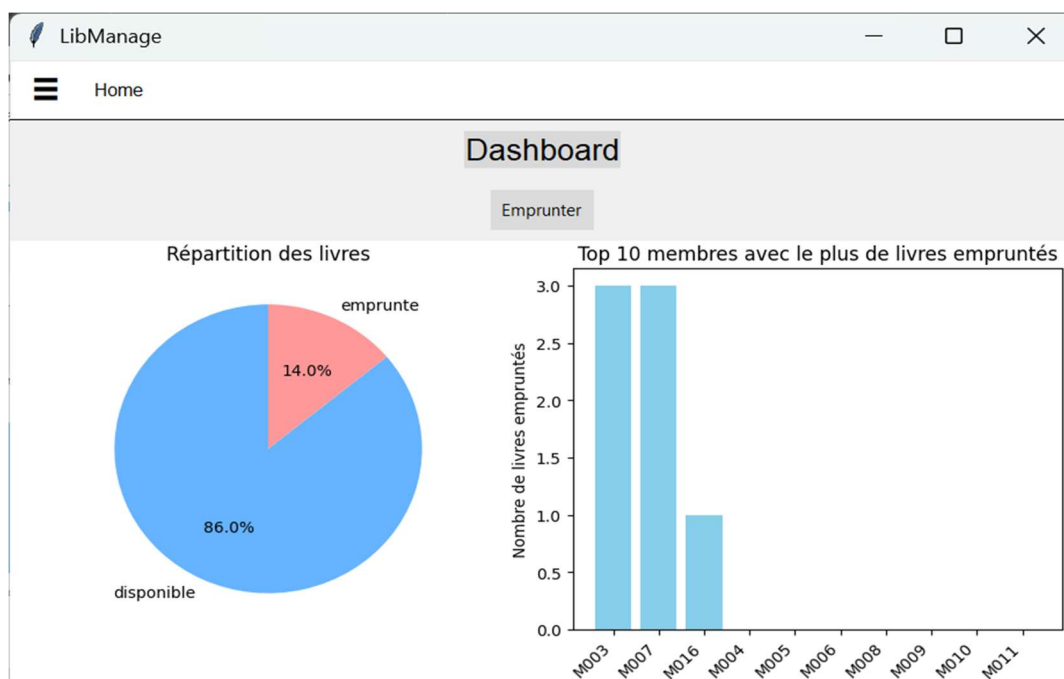
L'interface utilisateur :

Navigation dans l'application :



La navigation dans l'application **LibManage** est conçue pour être fluide et intuitive. L'utilisateur peut parcourir les différentes sections de l'application grâce à un **menu latéral escamotable** accessible via l'icône "**hamburger Menu**" en haut à gauche de l'interface. Ce menu permet de passer rapidement d'un écran à un autre, notamment vers l'accueil, la gestion des livres, des membres ou encore des emprunts. En complément de cette navigation principale, l'application offre une interaction contextuelle avancée grâce au **clic droit** : dans certaines vues comme celle des emprunts ou des membres, un clic droit sur une ligne d'un tableau déclenche un **menu contextuel** permettant d'effectuer des actions rapides (par exemple : retourner un livre, modifier un enregistrement, etc.). Cette approche rend l'interface à la fois ergonomique et efficace pour le personnel de la bibliothèque, en facilitant l'accès aux opérations courantes sans surcharger l'écran principal.

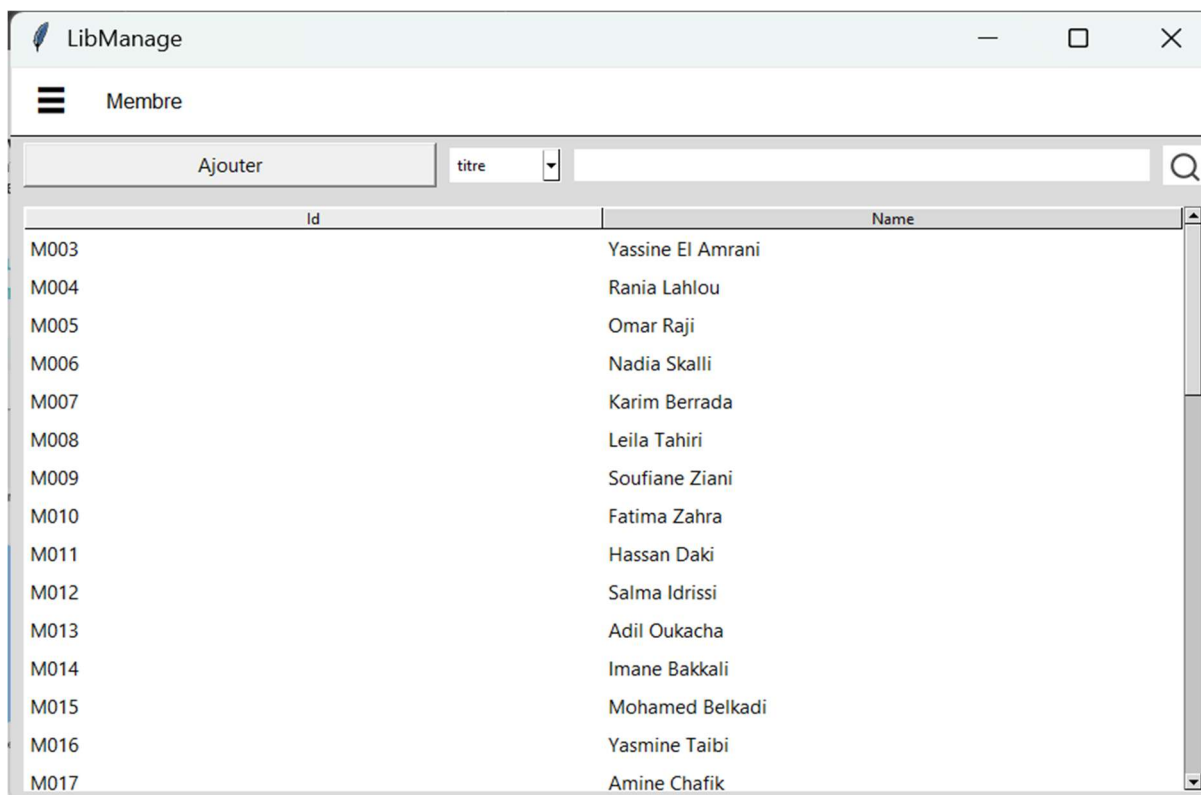
L'accueil :



La **vue d'accueil** est l'écran principal accessible après l'ouverture de l'application. Elle offre un aperçu global de l'activité de la bibliothèque, en affichant des **statistiques visuelles** sous forme de **graphiques**. L'interface est simple

Le **Button Emprunter** peut ouvrir un dialogue afin de réaliser le process d'emprunte

L'interface Membre :



L'interface dédiée aux **membres** de l'application LibManage permet au personnel de consulter, rechercher, ajouter et interagir avec les membres enregistrés dans le système. Par défaut, tous les membres sont affichés dans un **tableau clair et structuré**, avec leurs informations essentielles (identifiant, nom, etc.). Un **champ de recherche** permet de filtrer dynamiquement les membres selon leur identifiant ou leur nom : dès qu'une recherche est effectuée, le tableau se met à jour automatiquement pour n'afficher que les résultats correspondants. L'utilisateur peut également **ajouter un nouveau membre** via un formulaire prévu à cet effet.

ID	<input type="text"/>
Name	<input type="text"/>
<input type="button" value="Ajouter"/>	<input type="button" value="Annuler"/>

L'interface prend en charge des interactions avancées grâce au **clic droit sur une ligne du tableau**. Cela fait apparaître un **menu contextuel** qui propose des actions rapides comme :

- **Afficher les livres empruntés** par ce membre,
- **Supprimer un emprunt** spécifique.



L'interface Livre

LibManage								
Livres								
Ajouter			titre					
Isbn	Copy_id	Titre	Auteur	Annee	Genre	Copie_id	Statut	Copies_coun
978-0001	C1	Le Petit Prince	Antoine de Sa	1943	Fiction	C1	emprunte	1
978-0002	C1	L'Étranger	Albert Camus	1942	Philosophie	C1	emprunte	1
978-0003	C3	1984	George Orwel	1949	Dystopie	C3	emprunte	1
978-0005	C1	Harry Potter à	J.K. Rowling	1997	Fantastique	C1	emprunte	1
978-0006	C1	Le Comte de	Alexandre Du	1844	Aventure	C1	emprunte	1
978-0007	C1	Les Misérable	Victor Hugo	1862	Drame	C1	emprunte	1
978-0008	C1	La Peste	Albert Camus	1947	Roman	C1	disponible	1
978-0009	C1	Le Rouge et le	Stendhal	1830	Roman	C1	disponible	1
978-0010	C1	Notre-Dame de	Victor Hugo	1831	Roman histori	C1	disponible	1
978-0011	C1	Le Nom de la	Umberto Eco	1980	Policier	C1	disponible	1
978-0012	C1	Crime et Châti	Fiodor Dostoï	1866	Psychologie	C1	disponible	1
978-0013	C1	Le Seigneur de	J.R.R. Tolkien	1954	Fantastique	C1	disponible	1
978-0014	C1	Don Quichotte	Miguel de Cer	1605	Aventure	C1	disponible	1
978-0015	C1	Dracula	Bram Stoker	1897	Horreur	C1	disponible	1
978-0016	C1	Frankenstein	Marv Shellerv	1818	Science-fiction	C1	emprunte	1

L'interface **livre** de l'application LibManage permet au personnel de gérer facilement l'ensemble des exemplaires de livres disponibles dans la bibliothèque. Tous les exemplaires sont affichés dans un **tableau complet**, incluant des informations détaillées telles que l'ISBN, le copy_id (identifiant unique de l'exemplaire), le titre, l'auteur, l'année de publication, le genre, ainsi que le **statut** (disponible ou emprunté). L'utilisateur peut **ajouter un nouvel exemplaire** de livre via un formulaire dédié, en renseignant toutes les informations nécessaires.

Chaque ligne du tableau représente un **exemplaire physique distinct**, même si plusieurs exemplaires partagent le même ISBN. Grâce à un **clic droit sur une ligne du tableau**, un **menu contextuel** s'affiche et permet d'effectuer des actions rapides :

- **Supprimer** un exemplaire de livre,
- **Emprunter** directement cet exemplaire à un membre,
- **Retourner** un exemplaire précédemment emprunté.

ISBN	<input type="text"/>
Copy ID	<input type="text"/>
Titre	<input type="text"/>
Auteur	<input type="text"/>
Année	<input type="text"/>
Genre	<input type="text"/>
Statut	<input type="text" value="disponible"/>
<input type="button" value="Ajouter"/>	
<input type="button" value="Annuler"/>	

Isbn	Copy_id	Titre	Auteur	Annee	Genre	Copie_id	Statut	Copies_count
978-0001	C1	Le Petit Prince	Antoine de Sa	1943	Fiction	C1	emprunte	1
978-0002	C1	L'Étranger	Albert Ca	1942	Philosophie	C1	emprunte	1
978-0003	C3	1984	George O	1949	Dystopie	C3	emprunte	1
978-0005	C1	Harry Potter à	J.K. Rowling	1997	Fantastique	C1	emprunte	1
978-0006	C1	Le Comte de t	Alexandre Du	1844	Aventure	C1	emprunte	1

Dans l'application, la fonctionnalité « **Emprunter** » peut être lancée soit via un bouton depuis l'écran d'accueil, soit directement depuis un menu contextuel associé à un membre ou à un livre.

Lorsqu'on déclenche l'action d'emprunt, une nouvelle fenêtre s'ouvre, composée de **deux onglets (ou « slides »)** :

1. **Onglet de sélection du membre** : permet de rechercher et de choisir le membre qui souhaite emprunter un livre.
- 2.
3. **Onglet de sélection du livre** : permet de rechercher et de choisir le livre (ou la copie) à emprunter

Le comportement de cette fenêtre s'adapte automatiquement en fonction de l'origine de l'action :

- Si l'emprunt est lancé depuis l'interface **membre** (par exemple via un clic droit sur un membre puis « Emprunter »), alors le membre concerné est **sélectionné automatiquement** dans l'onglet de sélection du membre. Cela évite à l'utilisateur de devoir le rechercher manuellement.
- À l'inverse, si l'emprunt est déclenché depuis l'interface **livre** (par exemple via un clic droit sur un livre puis « Emprunter »), alors le livre est **recherché et pré-sélectionné automatiquement** dans l'onglet de sélection du livre.

Emprunt

Selectionner le livre a emprunter


ISBN


Search

ISBN	Title	Copy ID	Author	Year	Genre	Status	Count
978-0001	Le Petit Prince	C1	Antoine de Sa	1943	Fiction	emprunte	1
978-0002	L'Étranger	C1	Albert Camus	1942	Philosophie	emprunte	1
978-0003	1984	C3	George Orwel	1949	Dystopie	emprunte	1
978-0005	Harry Potter à	C1	J.K. Rowling	1997	Fantastique	emprunte	1
978-0006	Le Comte de t	C1	Alexandre Du	1844	Aventure	emprunte	1
978-0007	Les Misérables	C1	Victor Hugo	1862	Drame	emprunte	1

Back

☒
Emprunter

 Emprunt

 **Rechercher Membre**

ID

ID	Name
M003	Yassine El Amrani
M004	Rania Lahlou
M005	Omar Raji
M006	Nadia Skalli
M007	Karim Berrada
M008	Leila Tahiri

➡ Next

Technologies utilisées :

XML : Pour la gestion des données, j'ai opté pour le format **XML**, qui permet de stocker les informations des **membres, livres et emprunts** sous une forme structurée, facile à lire et à modifier. Cette structure facilite aussi **la validation** des données via des schémas XSD

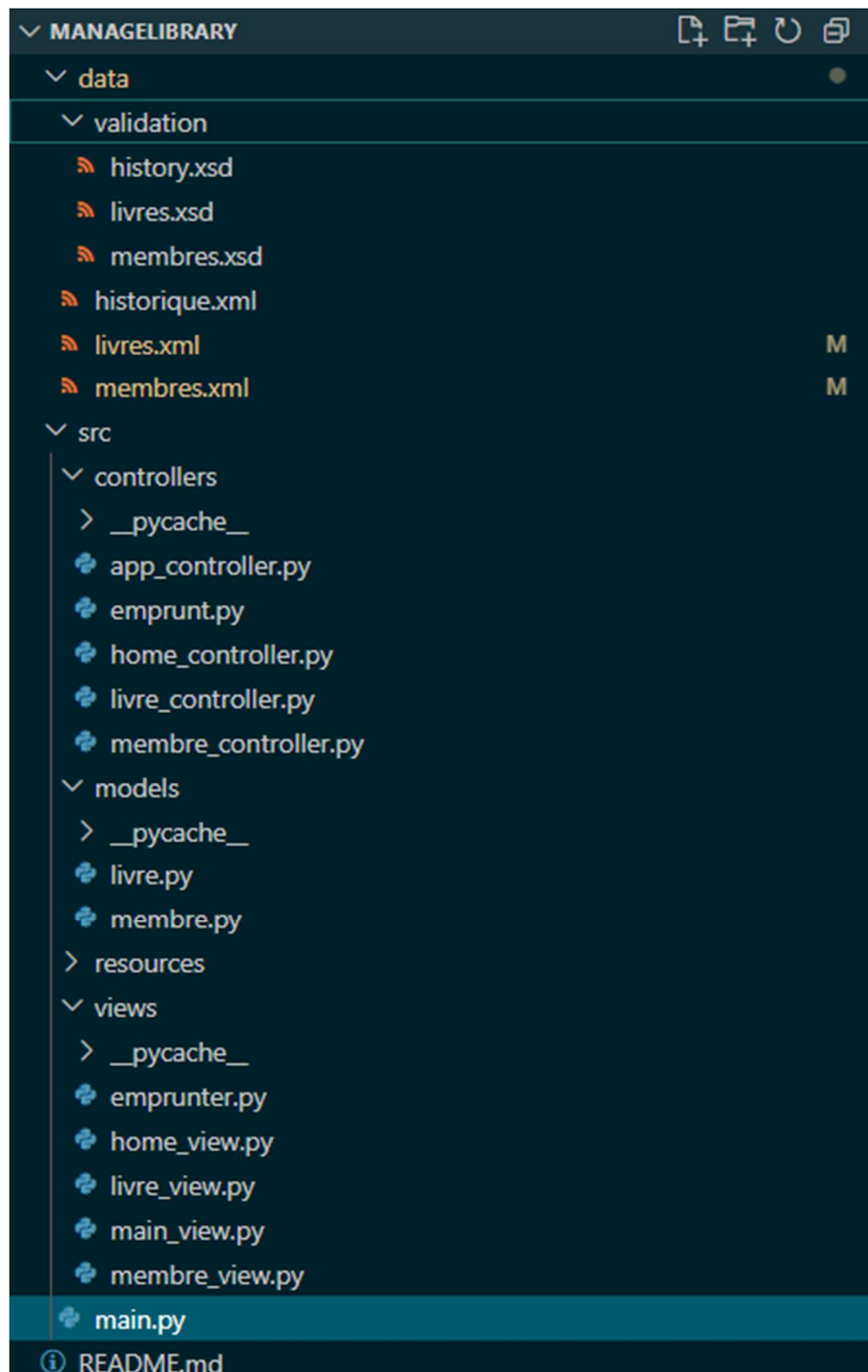
Autres :

Pour développer cette application, j'ai utilisé plusieurs technologies adaptées à mes besoins. Le langage principal est **Python**, qui offre une grande flexibilité et une syntaxe claire. Pour l'interface graphique, j'ai choisi **Tkinter**, la bibliothèque standard de Python pour créer des fenêtres, des boutons, des menus, et gérer les interactions utilisateur de manière simple et efficace. Afin d'intégrer des visualisations graphiques et des statistiques, j'ai utilisé **Matplotlib**, une bibliothèque puissante pour tracer des graphiques directement dans l'application.

Côté développement, j'utilise **Visual Studio Code** comme environnement intégré (IDE), qui propose de nombreuses extensions utiles pour Python, XML, et Git, rendant le codage, le débogage et la gestion des fichiers plus efficaces. Enfin, pour le contrôle de version et la gestion collaborative du code, j'ai recours à **Git**, qui me permet de versionner mon projet, suivre les modifications et partager facilement le code via **Github**.

MVC :

Dans ce projet, nous avons adopté l'architecture **MVC (Modèle-Vue-Contrôleur)** afin d'organiser clairement le code et séparer les responsabilités. Le **Modèle** gère les données et la logique métier, notamment la manipulation des membres, livres et emprunts via la lecture et l'écriture des fichiers XML. La **Vue** correspond à l'interface graphique construite avec Tkinter, qui affiche les informations à l'utilisateur et collecte ses interactions. Le **Contrôleur** fait le lien entre les deux, en orchestrant le flux des données, en répondant aux actions de l'utilisateur (clics, saisies) et en mettant à jour la vue en conséquence. Cette organisation facilite la maintenance, la compréhension du code et permet d'étendre plus facilement l'application.



Gestion des bugs :

Au cours du développement, j'ai rencontré de nombreux bugs et comportements inattendus qui rendaient l'application instable ou provoquaient des erreurs. Pour identifier précisément l'origine de ces problèmes, j'ai utilisé des **logs très simples**, basés uniquement sur des appels à la fonction print. En insérant des impressions stratégiques dans le code, j'ai pu suivre le déroulement des opérations, vérifier les valeurs des variables à différents moments, et comprendre où et pourquoi le programme ne fonctionnait pas comme prévu. Cette méthode rudimentaire mais efficace m'a permis de corriger étape par étape les erreurs sans utiliser d'outils de logging complexes.


```
Borrowing dialog closed.  
[voir_livres_emprnt] selected rows:  
- ('M003', 'Yassine El Amrani')  
[show livres-membre]: id parameter is :M003  
[show livre empruntee membre ] title is : Le Petit Prince  
[show livre empruntee membre ] title is : L'Étranger  
[show livre empruntee membre ] title is : 1984
```

Conclusion :

Pour conclure, ce projet m'a permis de mobiliser et d'appliquer une grande partie des compétences acquises tout au long de l'année en Python, notamment la programmation orientée objet et l'utilisation de bibliothèques comme Matplotlib pour la visualisation de données. J'ai également intégré des notions complémentaires apprises en dehors du cadre principal, comme la gestion des données en XML et l'architecture logicielle MVC, ce qui a structuré et clarifié le développement de l'application.

Cependant, le travail ne s'arrête pas là : il reste encore de nombreuses pistes d'amélioration, notamment en corrigeant les bugs restants et en optimisant l'interface utilisateur pour rendre l'application plus intuitive et agréable à utiliser. De plus, une évolution majeure serait de migrer le stockage des données depuis des fichiers XML vers une base de données. Cette transition permettrait d'améliorer significativement la rapidité et la robustesse de l'application, tout en facilitant la gestion et la sécurité des données sur le long terme.

Ce projet représente donc une étape importante dans mon apprentissage, avec un fort potentiel d'évolution future.

