

# Rapport Technique

---

## BiblioApp

« Concevoir, réaliser et documenter un système de gestion de bibliothèque moderne, extensible et pédagogique »

ENCADRÉE PAR : ZAKARIA HAJA

BADIOUI ASMAA

ENSA OUJDA – GÉNIE INFORMATIQUE – 1RE  
ANNÉE CYCLE INGÉNIEUR

# **SOMMAIRE**

<b><u>Introduction</u></b>	1
<b><u>Objectifs du projet</u></b>	2
<b><u>Structure et fonctionnement de l'app</u></b>	3
<b><u>Fonctionnalités principales</u></b>	4
<b><u>Interface utilisateur</u></b>	5
<b><u>Sécurité et robustesse</u></b>	6
<b><u>Difficultés rencontrées</u></b>	7
<b><u>Améliorations futures</u></b>	8
<b><u>Conclusion</u></b>	9

# 1. Introduction

---

Les bibliothèques de petite ou moyenne taille rencontrent souvent des difficultés pour gérer efficacement leur catalogue et leurs lecteurs. Le projet BiblioApp vise à répondre à ce besoin en proposant une application simple, intuitive et multiplateforme. Elle permet d'ajouter des livres, de gérer les membres, d'enregistrer les emprunts et les retours, et de visualiser des statistiques utiles.

## 2. Objectifs du projet

---

L'objectif principal de BiblioApp est d'offrir une solution pédagogique et extensible de gestion de bibliothèque. Le projet repose sur les principes suivants:

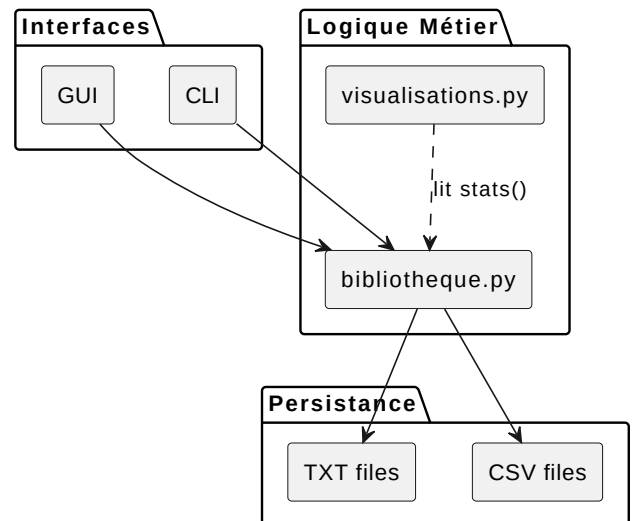
- Une interface graphique conviviale pour le personnel, développée avec Tkinter.
- Une interface en ligne de commande pour les opérations techniques ou automatisées.
- Une structure de code bien organisée (architecture inspirée du modèle MVC).

### 3. Structure et fonctionnement de l'application

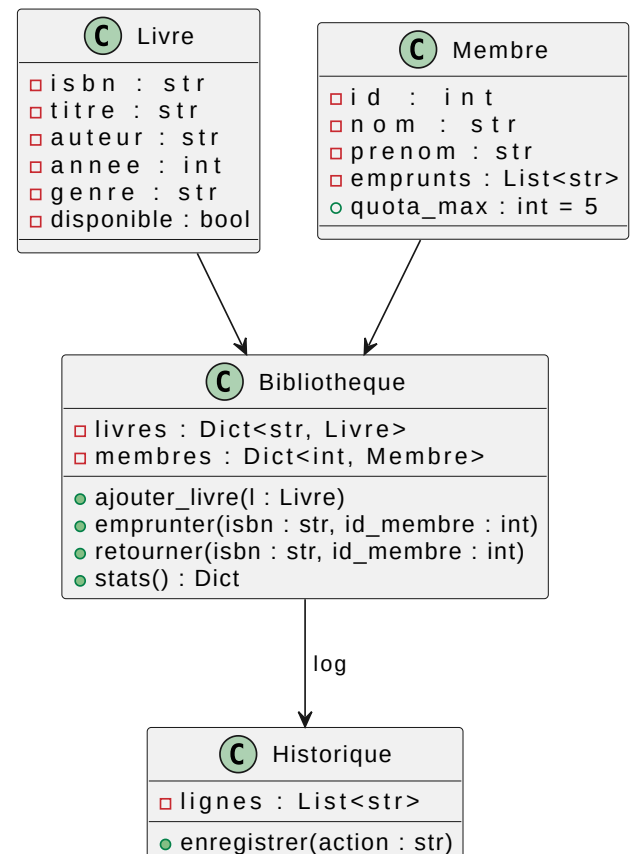
L'application est divisée en plusieurs couches logiques :

- L'interface utilisateur permet la navigation dans les livres, les membres et les statistiques.
- La couche métier applique les règles comme les limites de prêts ou la vérification de la disponibilité.
- La couche d'accès aux données gère la lecture et l'écriture dans les fichiers (livres, membres, historique).
- La couche technique inclut les exceptions, les tests et les outils de journalisation.

Chaque partie est conçue pour être indépendante, afin de faciliter la maintenance et l'évolution.



Pour guider le développement et structurer le code, un diagramme de classes UML a été conçu. Il montre les principales entités du projet (Livre, Membre, Emprunt, Bibliothèque), leurs attributs et leurs relations. Le modèle suit une organisation orientée objet claire : chaque entité possède ses propres responsabilités, tandis que la classe Bibliothèque centralise la logique métier (emprunter, retourner, rechercher, sauvegarder...). Cette modélisation a permis de garder une architecture cohérente et évolutive tout au long du développement.



## 4. Fonctionnalités principales

L'application permet de :

Ajouter, modifier ou supprimer un livre ou un membre.

Effectuer des emprunts et des retours sécurisés.

Consulter un historique horodaté des actions réalisées.

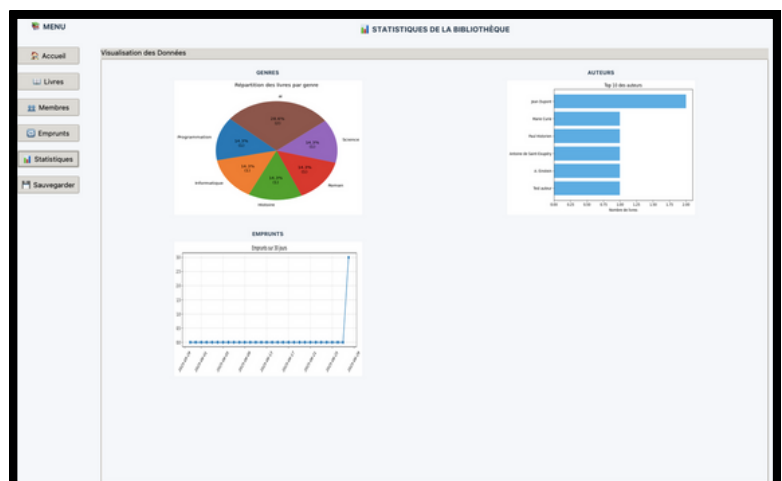
Générer automatiquement des graphiques de statistiques (popularité des livres, emprunts par genre, etc.).

Lors d'un emprunt, l'application vérifie si le livre est disponible et si le membre n'a pas dépassé sa limite. En cas de problème, un message clair est affiché à l'utilisateur.

## 5. Interface utilisateur

L'interface graphique utilise des onglets pour accéder facilement aux différentes parties de l'application : Livres, Membres, Emprunts, Statistiques. Les formulaires sont simples à utiliser et des messages d'erreur guident l'utilisateur en cas de saisie incorrecte.

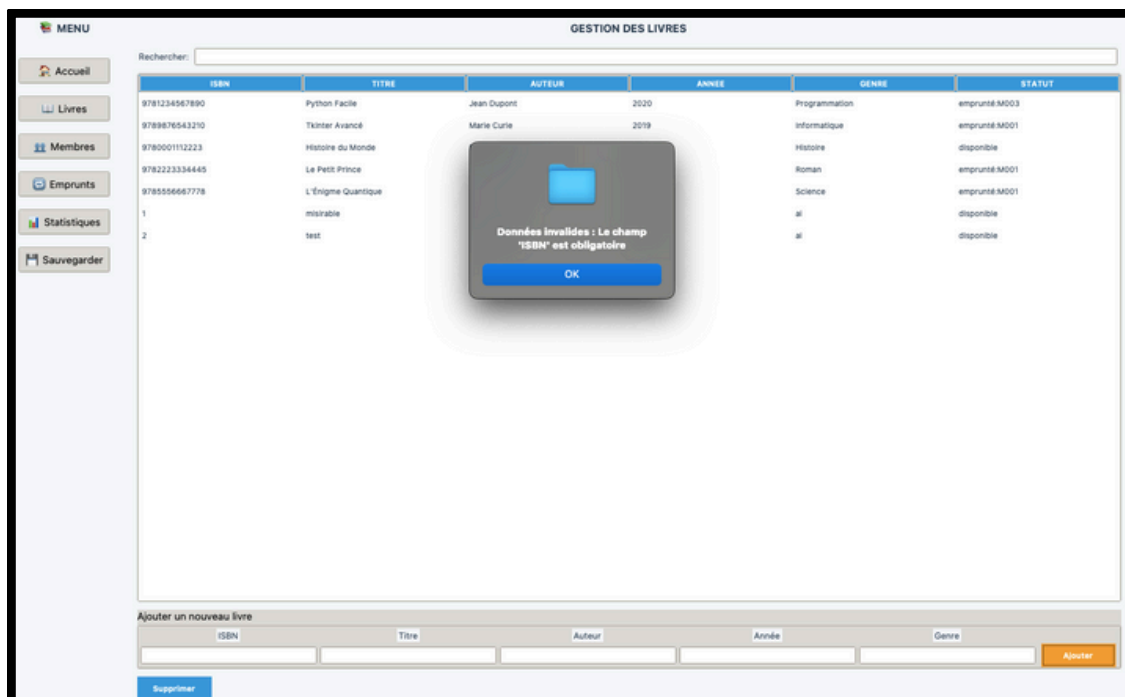
L'interface en ligne de commande est utile pour des scripts automatisés ou une utilisation avancée. Elle affiche un menu clair, avec une aide disponible pour chaque action.



## 6. Sécurité et robustesse

Pour garantir la fiabilité des données :

- Toutes les écritures sont d'abord enregistrées dans un fichier temporaire, puis transférées vers le fichier final.
- Les actions importantes sont enregistrées dans un historique.
- Des exceptions personnalisées sont utilisées pour gérer les erreurs (livre déjà emprunté, membre introuvable, etc.).



## 7. Difficultés rencontrées

Plusieurs problèmes ont été résolus au cours du projet :

- Des conflits d'accès aux fichiers ont été évités grâce à une technique de sauvegarde sécurisée.
- La mise à jour de l'interface graphique a été améliorée en utilisant un système de notification.
- Les performances ont été optimisées pour gérer plusieurs milliers de livres rapidement.

## 8. Améliorations futures

---

Le projet peut facilement évoluer vers une version plus avancée :

- Connexion à une base de données comme SQLite pour une meilleure gestion des relations.
- Création d'une API web avec FastAPI pour des applications mobiles.
- Ajout d'un système de réservation avec envoi de notifications par email.
- Mise en place d'un tableau de bord web en React.

## 10. Conclusion

---

BiblioApp est une application complète, bien structurée et facile à utiliser. Elle remplit pleinement son objectif pédagogique en combinant la programmation orientée objet, les bonnes pratiques Python et les notions de qualité logicielle. Grâce à sa modularité, elle peut facilement être enrichie pour répondre à d'autres besoins dans le futur.