Système de Gestion de Bibliothèque Universitaire

Projet B3 Développement

Technologies utilisées :
Python • Streamlit • PostgreSQL • SQLAlchemy

Date du rapport : 08/01/2026

1. Introduction

Ce projet consiste en un système complet de gestion de bibliothèque universitaire développé en Python. L'application permet de gérer efficacement les livres, les étudiants, les emprunts et les amendes à travers une interface web intuitive construite avec Streamlit.

2. Objectifs du projet

• **Centraliser la gestion** : Regrouper toutes les opérations de la bibliothèque en un seul système
• **Faciliter les emprunts** : Permettre un suivi en temps réel des livres empruntés
• **Automatiser les amendes** : Calculer automatiquement les pénalités de retard
• **Fournir des statistiques** : Offrir une vue d'ensemble via le tableau de bord

3. Architecture technique

L'application suit une architecture en couches séparant clairement les responsabilités :

Couche	Description	Fichiers
Présentation	Interface utilisateur Streamlit	app.py, modules/*.py
Modèles	Classes SQLAlchemy (ORM)	models.py
Accès données	CRUD et requêtes SQL	crud_orm.py, utils.py
Base de données	PostgreSQL	database.py

4. Fonctionnalités principales

4.1 Page d'accueil
Tableau de bord affichant les statistiques clés :
• Nombre total de livres dans le catalogue
• Nombre d'étudiants inscrits
• Emprunts en cours
• Montant total des amendes

4.2 Gestion des livres
• Affichage du catalogue complet avec ISBN, titre, éditeur, année et disponibilité
• Filtrage des livres disponibles
• Recherche et consultation des informations détaillées

4.3 Gestion des étudiants
• Liste complète des étudiants inscrits
• Informations : ID, nom, prénom, email, date d'inscription, solde d'amende
• Suivi du statut de chaque étudiant

4.4 Gestion des emprunts
• Liste des emprunts en cours et terminés
• Traçabilité complète : dates d'emprunt et de retour
• Calcul automatique des amendes en cas de retard

4.5 Gestion des amendes

• Vue détaillée des amendes par étudiant
• Suivi des paiements et des soldes
• Historique des pénalités

4.6 Module CRUD
• Opérations Create, Read, Update, Delete sur toutes les entités
• Interface d'administration complète
• Gestion avancée des données

5. Modèle de base de données

La base de données PostgreSQL comprend trois tables principales reliées par des clés étrangères :

5.1 Table ETUDIANT

Colonne	Type	Description
id_etud	INTEGER (PK)	Identifiant unique auto-incrémenté
nom	VARCHAR(50)	Nom de l'étudiant
prenom	VARCHAR(50)	Prénom de l'étudiant
email	VARCHAR(100)	Email (unique)
date_inscription	DATE	Date d'inscription
solde_amende	NUMERIC(5,2)	Montant des amendes dues

5.2 Table LIVRE

Colonne	Type	Description
isbn	VARCHAR(13) (PK)	Numéro ISBN unique du livre
titre	VARCHAR(200)	Titre du livre
editeur	VARCHAR(100)	Maison d'édition
annee	INTEGER	Année de publication
exemplaires_dispo	INTEGER	Nombre d'exemplaires disponibles

5.3 Table EMPRUNT

Colonne	Type	Description
id_emprunt	INTEGER (PK)	Identifiant unique de l'emprunt
id_etud	INTEGER (FK)	Référence à l'étudiant
isbn	VARCHAR(13) (FK)	Référence au livre
date_emprunt	DATE	Date de l'emprunt
date_retour	DATE	Date de retour (NULL si en cours)
amende	NUMERIC(5,2)	Montant de l'amende

6. Exemples de code

6.1 Définition du modèle Étudiant (SQLAlchemy)

```
class Etudiant(Base): __tablename__ = 'etudiant' id_etud = Column(Integer, primary_key=True, autoincrement=True) nom = Column(String(50), nullable=False) prenom = Column(String(50), nullable=False) email = Column(String(100), nullable=False, unique=True) date_inscription = Column(Date, default=date.today) solde_amende = Column(Numeric(5, 2), default=0) # Relation avec les emprunts emprunts = relationship("Emprunt", back_populates="etudiant")
```

6.2 Requête SQL pour afficher les livres disponibles

```
requete = """ SELECT isbn, titre, editeur, annee, exemplaires_dispo FROM livre WHERE exemplaires_dispo > 0 ORDER BY titre; """ resultat = executer_requete_sql(requete) tableau_livres = convertir_en_tableau( resultat, ["ISBN", "Titre", "Éditeur", "Année", "Exemplaires disponibles"] )
```

6.3 Structure de navigation Streamlit

```
page_selectionnee = st.sidebar.radio( "Aller à :", [ "■ Accueil", "■ Livres", "■■■ Étudiants", "■ Emprunts", "■ Amendes", "■■ Gestion CRUD" ] ) if page_selectionnee == "■ Accueil": accueil.afficher() elif page_selectionnee == "■ Livres": livres.afficher() # ... autres pages
```

7. Exemples de sorties

7.1 Affichage du catalogue de livres

```
Exemple de sortie pour la page Livres : +---------------+--------------------------------
+------------------+------+----------------------+ | ISBN | Titre | Éditeur | Année|
Exemplaires dispo | +---------------+--------------------------------+------------------+
------+----------------------+ | 9782070360024 | Le Petit Prince | Gallimard | 1943 | 3 |
| 9782253004226 | 1984 | Livre de Poche | 1950 | 2 | | 9782253151623 | Les Misérables |
Livre de Poche | 1862 | 1 | | 9782070413119 | L'Étranger | Gallimard | 1942 | 0 | +------
---------+--------------------------------+------------------+------+--------------------
--+ ■ 4 livre(s) trouvé(s)
```

7.2 Liste des étudiants inscrits

```
Exemple de sortie pour la page Étudiants : +--------+----------+-----------+-------------
--------------+------------------+--------------+ | ID | Nom | Prénom | Email | Date
inscription | Solde amende | +--------+----------+-----------+---------------------------
+------------------+--------------+ | 1 | Dupont | Jean | jean.dupont@univ.fr | 2024-09-01
| 0.00 € | | 2 | Martin | Sophie | sophie.martin@univ.fr | 2024-09-05 | 5.50 € | | 3 |
Bernard | Lucas | lucas.bernard@univ.fr | 2024-09-10 | 0.00 € | | 4 | Petit | Emma |
emma.petit@univ.fr | 2024-09-15 | 12.00 € | +--------+----------+-----------+------------
---------------+------------------+--------------+ ■ 4 étudiant(s) inscrit(s)
```

7.3 Statistiques du tableau de bord

```
Exemple de sortie pour la page d'accueil (Dashboard) : ■ STATISTIQUES DE LA BIBLIOTHÈQUE
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■ ■ Livres au catalogue : 245 ■■■
Étudiants inscrits : 89 ■ Emprunts en cours : 42 ■ Amendes totales : 127.50 € ■ Livres
disponibles : 203 ■ Retours en retard : 8 Dernière mise à jour : 08/01/2026 14:30
```

8. Guide d'installation et d'utilisation

8.1 Installation

```
# 1. Cloner le repository git clone [url-du-projet] cd TP_python # 2. Créer un environnement virtuel python -m venv venv source venv/bin/activate # Sur Windows: venv\Scripts\activate # 3. Installer les dépendances pip install -r requirements.txt # 4. Configurer PostgreSQL # Créer une base de données nommée 'bibliotheque' # Mettre à jour les paramètres de connexion dans database.py
```

8.2 Lancement de l'application

```
# Depuis le répertoire racine du projet streamlit run src/app.py # L'application s'ouvre automatiquement dans le navigateur # URL par défaut : http://localhost:8501
```

8.3 Utilisation

Navigation : Utilisez le menu latéral pour accéder aux différentes sections

Consultation : Les pages Livres, Étudiants, Emprunts et Amendes affichent les données sous forme de tableaux interactifs avec possibilité de tri et de filtre

Gestion CRUD : La page "Gestion CRUD" permet d'effectuer toutes les opérations d'administration : ajout, modification et suppression d'enregistrements

Tableau de bord : La page d'accueil fournit une vue d'ensemble en temps réel des statistiques importantes de la bibliothèque

9. Conclusion

Ce système de gestion de bibliothèque universitaire offre une solution complète et moderne pour gérer efficacement l'ensemble des opérations d'une bibliothèque. L'utilisation de technologies actuelles (Python, Streamlit, PostgreSQL, SQLAlchemy) garantit performance, maintenabilité et évolutivité du système.

L'interface intuitive permet une prise en main rapide par les utilisateurs, tandis que l'architecture modulaire facilite les futures évolutions et l'ajout de nouvelles fonctionnalités.