

*Documentation du projet  
de gestion de bibliothèque  
avec utilisation de  
Flask-JWT-Extended*

*Groupe 7:  
-GNANSA Essossolam  
-AKOMANYI Victor*

## **Table des Matières**

- 1-Introduction**
- 2-Architecture du projet**
- 3-Diagrammes UML**
  - \*Diagramme de classe**
  - \*Diagramme de séquence**
  - \*Diagramme d'activité**
- 4-Images des différentes fonctionnalités de l'application**
- 5- Fonctionnement de Flask-Security**
- 6-Tests**
- 7-Améliorations possibles**

## **1 - Introduction**

*Ce projet est une application web complète de gestion de bibliothèque développée avec une architecture moderne :*

*Backend: Flask (Python) pour l'API RESTful*

*Frontend: Vue.js pour l'interface utilisateur*

*Base de données: PostgreSQL pour le stockage relationnel*

*L'application permet aux utilisateurs de:*

*\$ S'inscrire et se connecter via un système sécurisé JWT*

*\$ Parcourir et rechercher des livres*

*\$ Emprunter et retourner des livres*

*\$ Gérer le catalogue (pour les administrateurs)*

*\$ Recevoir des notifications par email*

*L'utilisation de JWT (JSON Web Tokens) constitue le cœur du système de sécurité. Cette technologie permet:*

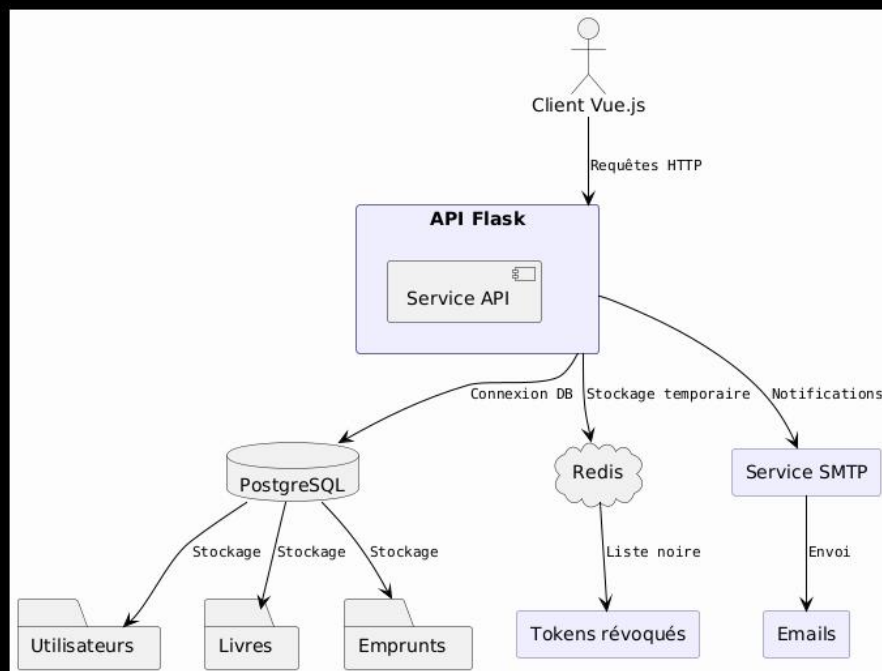
*-Une authentification stateless sans sessions serveur*

*-Des tokens d'accès courte durée (15 min) pour les opérations quotidiennes*

*-Des tokens de rafraîchissement longue durée (30 jours) pour maintenir la session*

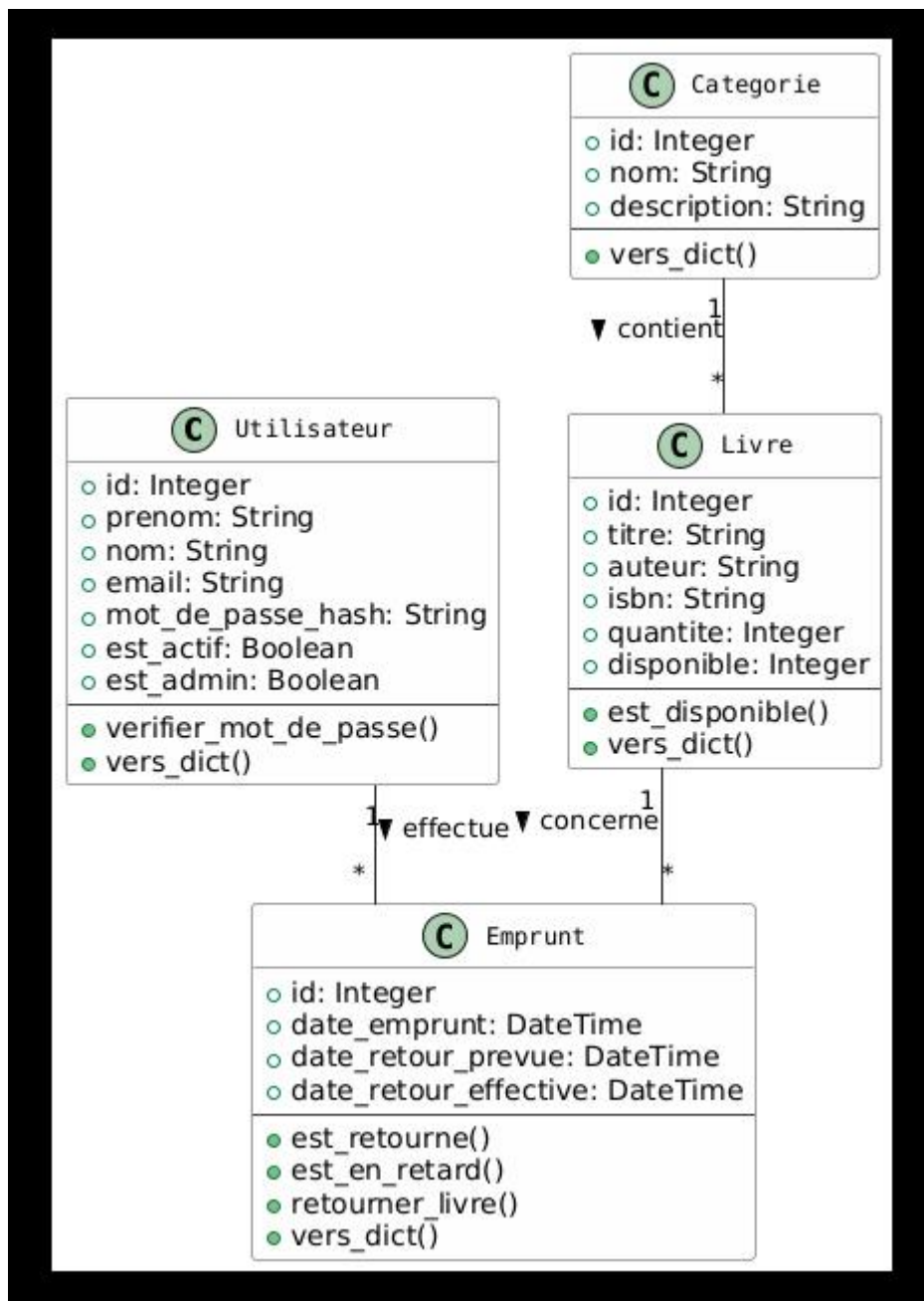
*-Une révocation de tokens pour les déconnexions sécurisées*

## **2- Architecture Technique du projet**

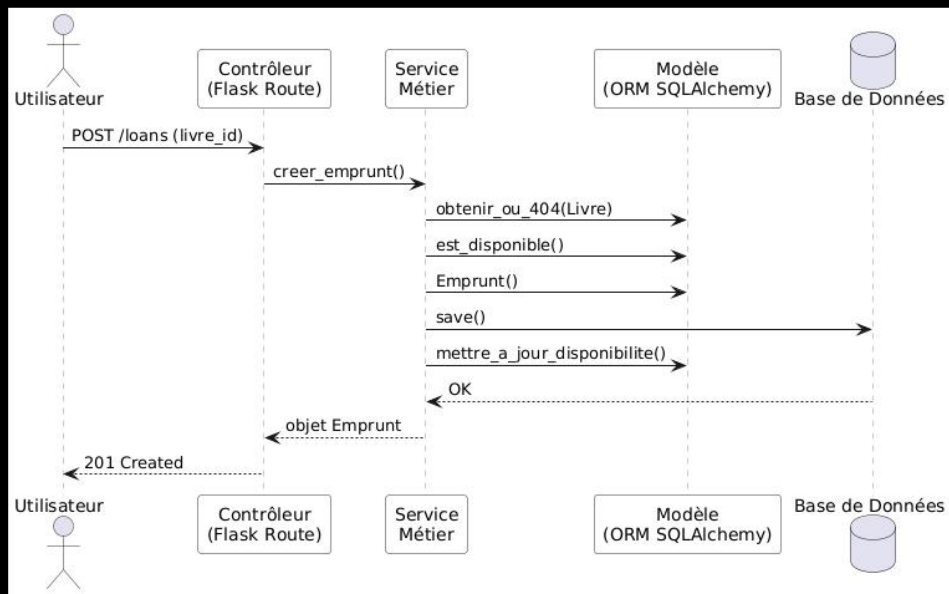


### 3-Diagrammes Uml importants du projet

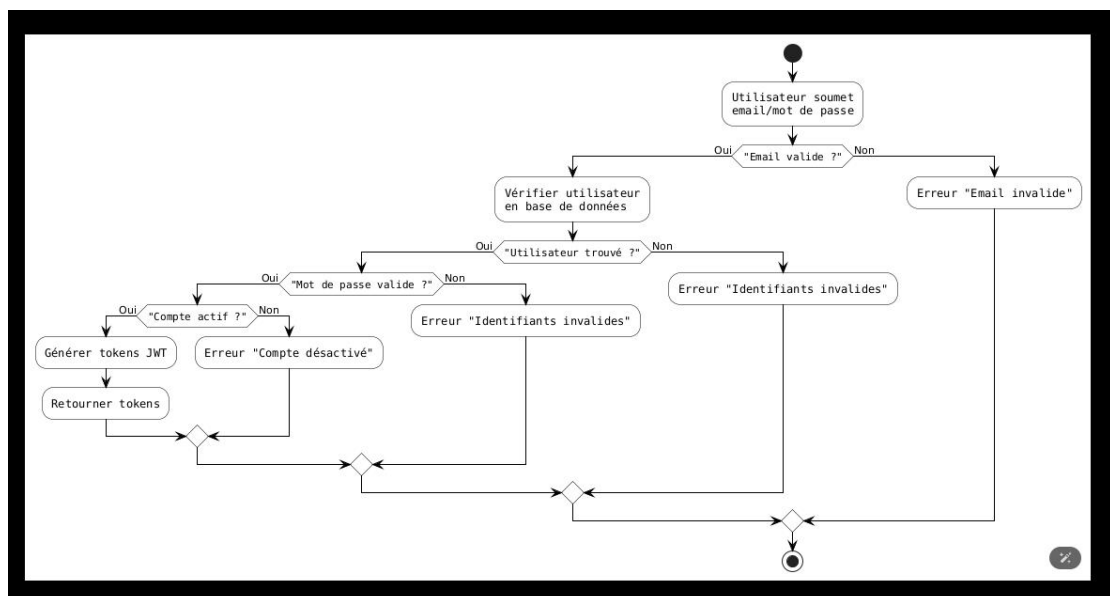
*\*Diagramme de classe*



\* Diagramme de séquence



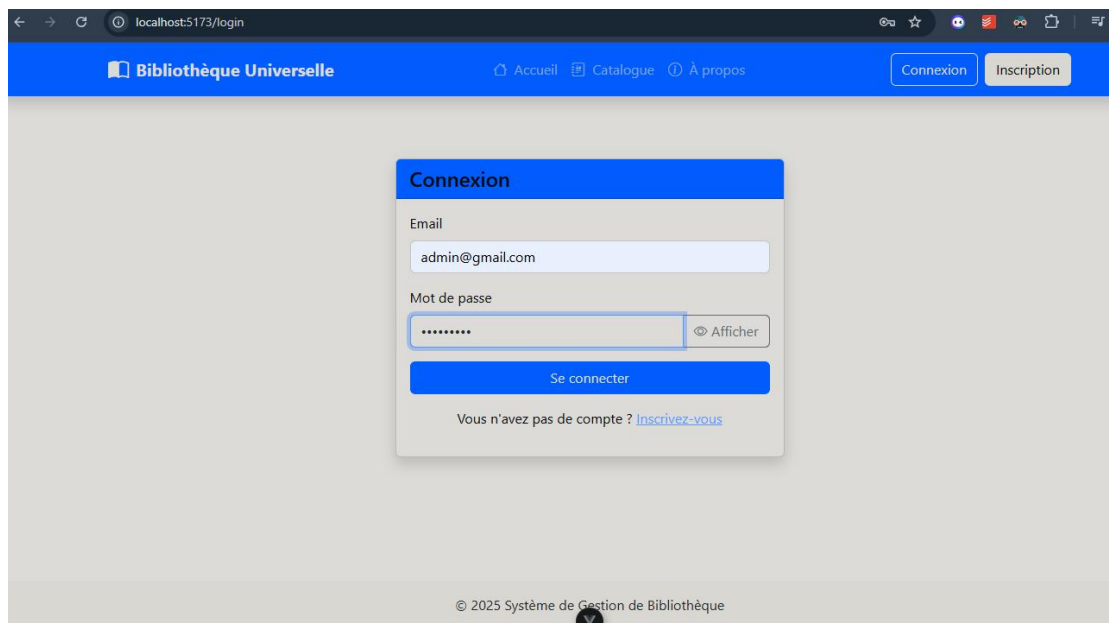
*\*Diagramme d'activité (Authentification)*



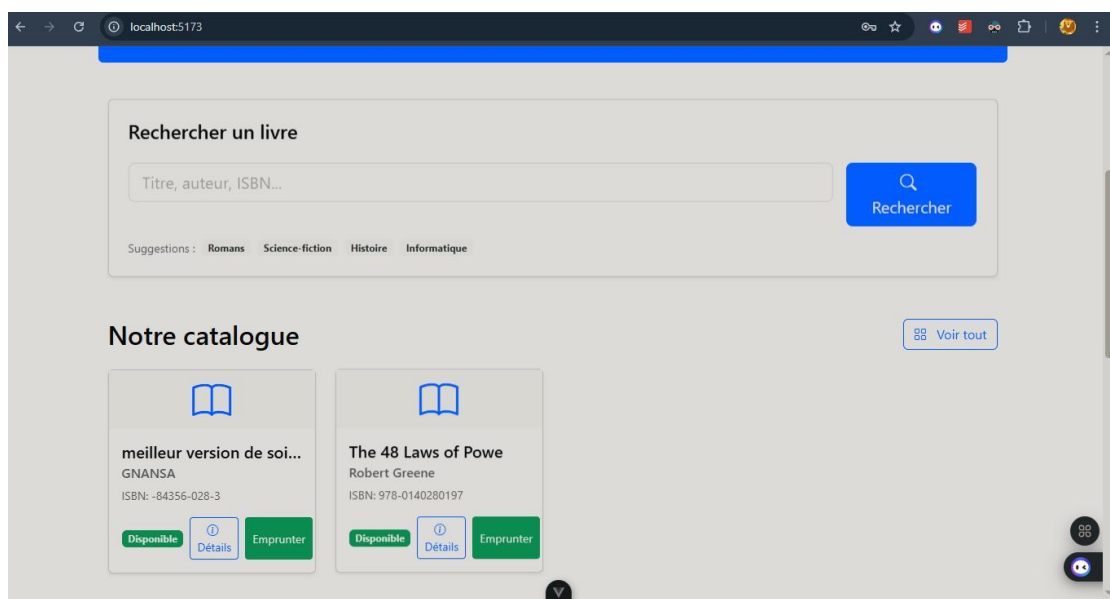
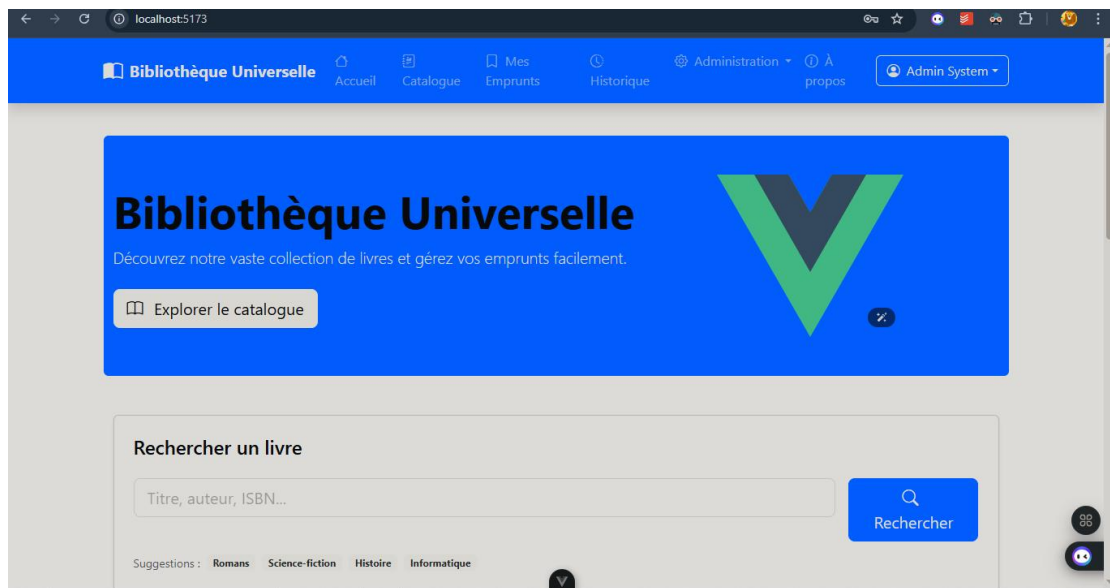
## 4-Images des différentes fonctionnalités de l'application

\* Tout d'abord je commence par la connexion d'un admin de l'app et de ses différents cas d'utilisation

\* Connexion d'un admin



\*Voici le landing page apres la connexion de l'admin



*\*Ajout d'un livre au catalogue par un Admin*

localhost:5173/books/create

**Bibliothèque Universelle** Accueil Catalogue Mes Emprunts Historique Administration A propos Admin System

## Ajouter un Livre

← Retour

**Titre \***  
Presence

**Auteur \***  
Amy Cuddy

**ISBN \***  
978-0316256575  
Format: ISBN-10 ou ISBN-13

**Date de publication**  
22/12/2015  
Format: YYYY-MM-DD

**Quantité \***  
2

Annuler

Ajouter le livre

© 2025 Système de Gestion de Bibliothèque

localhost:5173/books/4

**Bibliothèque Universelle** Accueil Catalogue Mes Emprunts Historique Administration A propos Admin System

## Presence

Amy Cuddy

← Retour

**ISBN:** 978-0316256575  
**Date de publication:** 22/12/2015  
[Modifier](#)

**Disponibilité:** 2/2 disponible(s)  
**Ajouté le:** 27/06/2025  
[Emprunter](#)

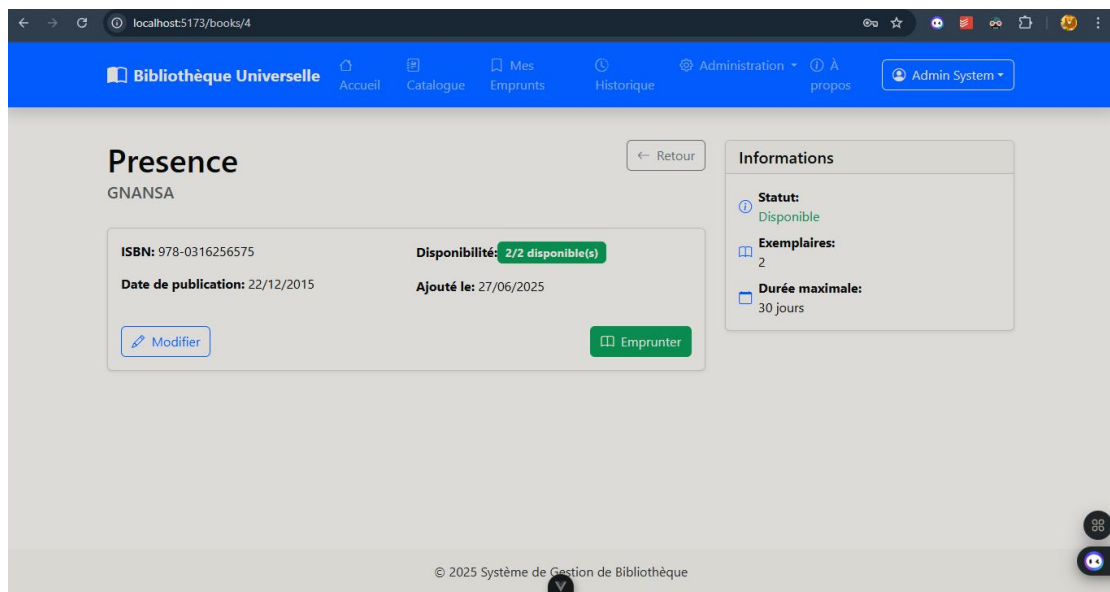
**Informations**

- Statut:** Disponible
- Exemplaires:** 2
- Durée maximale:** 30 jours

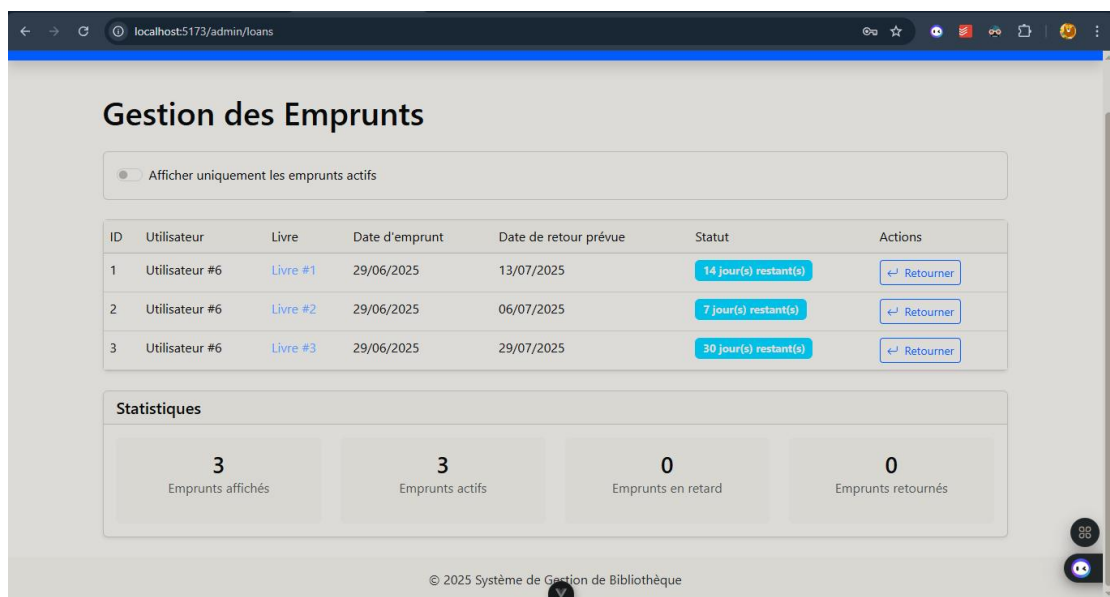
© 2025 Système de Gestion de Bibliothèque

\* La modification d'un livre (Ici j'ai changé le nom de l'auteur)



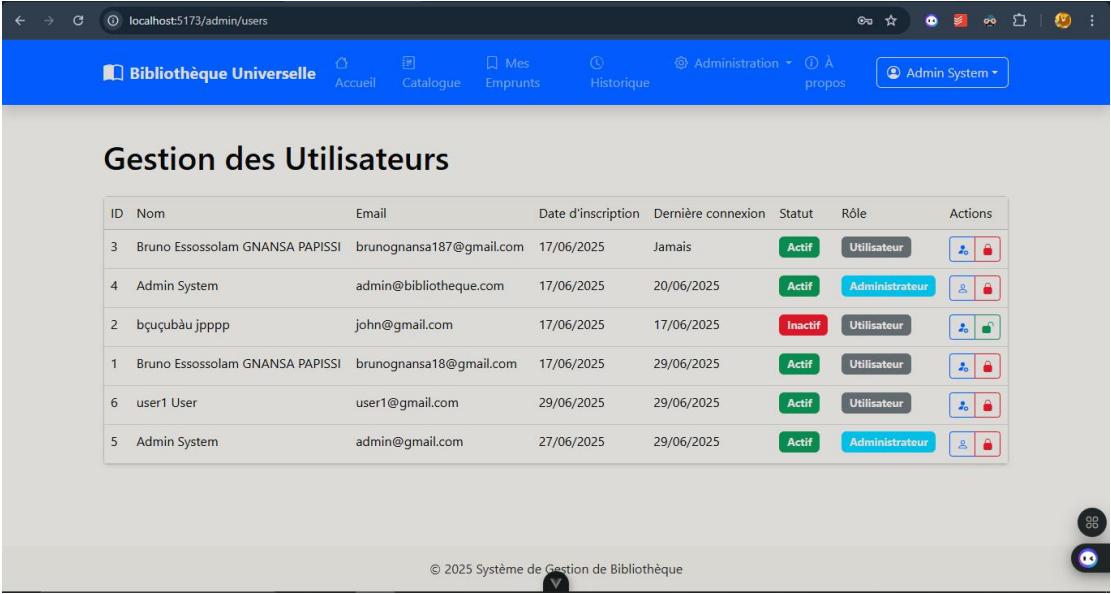














*\*Nous avons une interface qui permet a l'administrateur de visualier les différents emprunts effectuer par les utilisateurs*



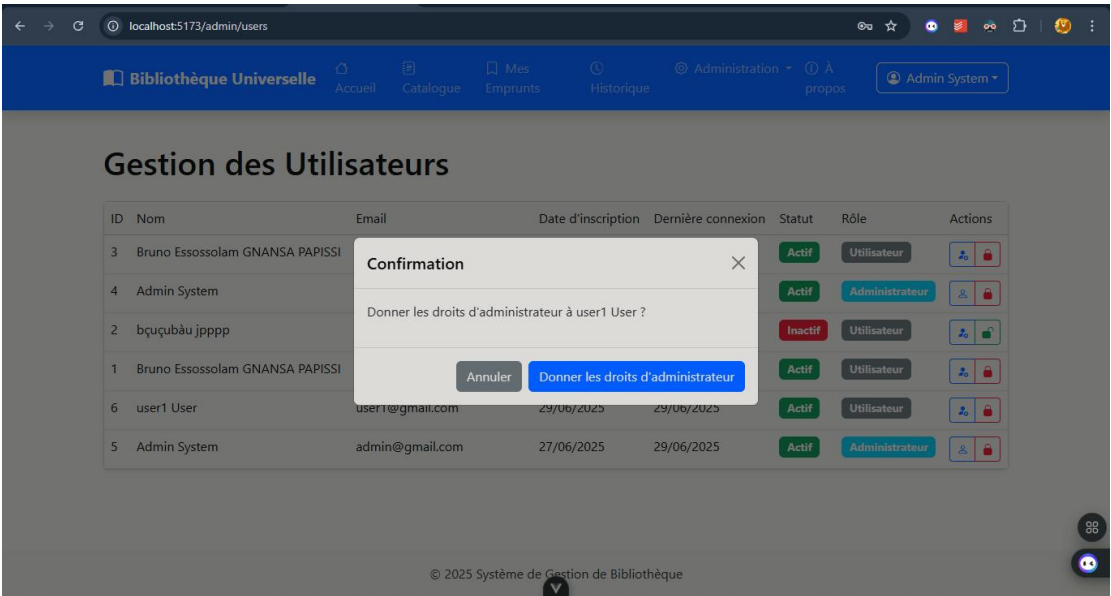
*Cette interface permet a l'administrateur de visualiser les emprunts actifs, ceux qui ont été retournés et ceux qui ne l'ont pas encore été*













\* Une autre fonctionnalité de l'admin qui permet d'obtenir la liste des différents utilisateurs avec différentes informations leurs concernant (leurs dates d'inscription , leurs dernières connexion ,leurs statuts et leurs roles )



ID	Nom	Email	Date d'inscription	Dernière connexion	Statut	Rôle	Actions
3	Bruno Essossolam GNANSA PAPISSI	brunognansa187@gmail.com	17/06/2025	Jamais	Actif	Utilisateur	 
4	Admin System	admin@bibliotheque.com	17/06/2025	20/06/2025	Actif	Administrateur	 
2	bçucubàu jpppp	john@gmail.com	17/06/2025	17/06/2025	Inactif	Utilisateur	 
1	Bruno Essossolam GNANSA PAPISSI	brunognansa18@gmail.com	17/06/2025	29/06/2025	Actif	Utilisateur	 
6	user1 User	user1@gmail.com	29/06/2025	29/06/2025	Actif	Utilisateur	 
5	Admin System	admin@gmail.com	27/06/2025	29/06/2025	Actif	Administrateur	 

\*L'admin peut aussi accorder des droits d'administrateur a un autre utilisateur



ID	Nom	Email	Date d'inscription	Dernière connexion	Statut	Rôle	Actions
3	Bruno Essossolam GNANSA PAPISSI	brunognansa187@gmail.com	17/06/2025	Jamais	Actif	Utilisateur	 
4	Admin System	admin@bibliotheque.com	17/06/2025	20/06/2025	Actif	Administrateur	 
2	bçucubàu jpppp	john@gmail.com	17/06/2025	17/06/2025	Inactif	Utilisateur	 
1	Bruno Essossolam GNANSA PAPISSI	brunognansa18@gmail.com	17/06/2025	29/06/2025	Actif	Utilisateur	 
6	user1 User	user1@gmail.com	29/06/2025	29/06/2025	Actif	Utilisateur	 
5	Admin System	admin@gmail.com	27/06/2025	29/06/2025	Actif	Administrateur	 

*Ou soit retirer des droits d'administrateur*

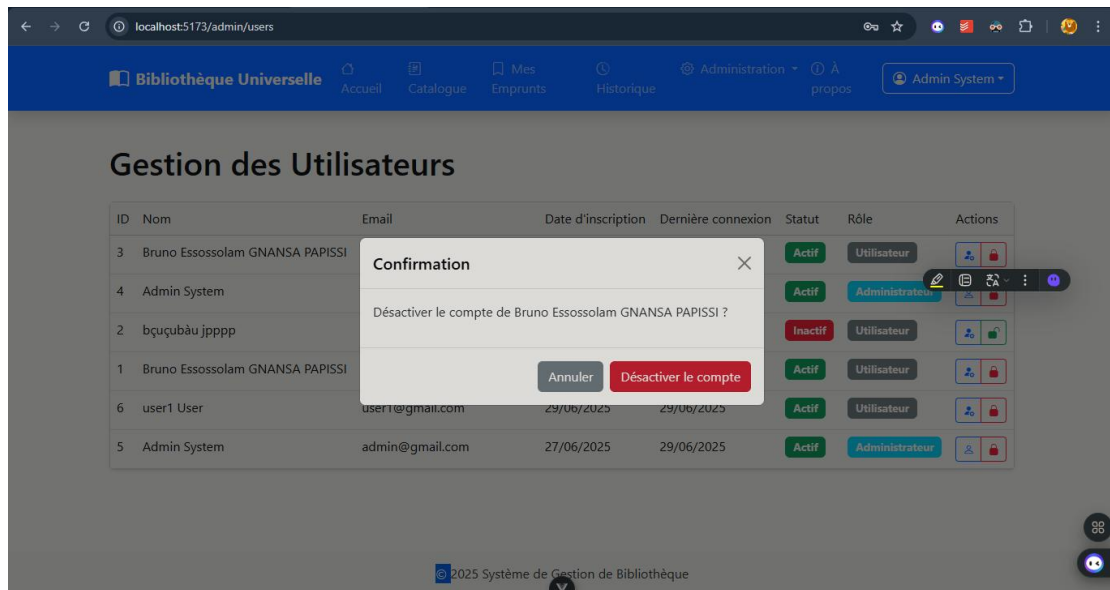
The screenshot shows the 'Gestion des Utilisateurs' page in a web application. A confirmation dialog is displayed over the table, asking 'Retirer les droits d'administrateur de Admin System ?'. The dialog has two buttons: 'Annuler' and 'Retirer les droits d'administrateur'. The table below has columns: ID, Nom, Email, Date d'inscription, Dernière connexion, Statut, Rôle, and Actions. The user 'Admin System' is highlighted in the table.

ID	Nom	Email	Date d'inscription	Dernière connexion	Statut	Rôle	Actions
3	Bruno Essossolam GNANSA PAPISSI				Actif	Utilisateur	[Icons]
4	Admin System				Actif	Administrateur	[Icons]
2	bçuçubâu jpppp				Inactif	Utilisateur	[Icons]
1	Bruno Essossolam GNANSA PAPISSI				Actif	Utilisateur	[Icons]
6	user1 User	user1@gmail.com	29/06/2025	29/06/2025	Actif	Utilisateur	[Icons]
5	Admin System	admin@gmail.com	27/06/2025	29/06/2025	Actif	Administrateur	[Icons]

*\*L'administrateur peut aussi activer et désactiver un compte*

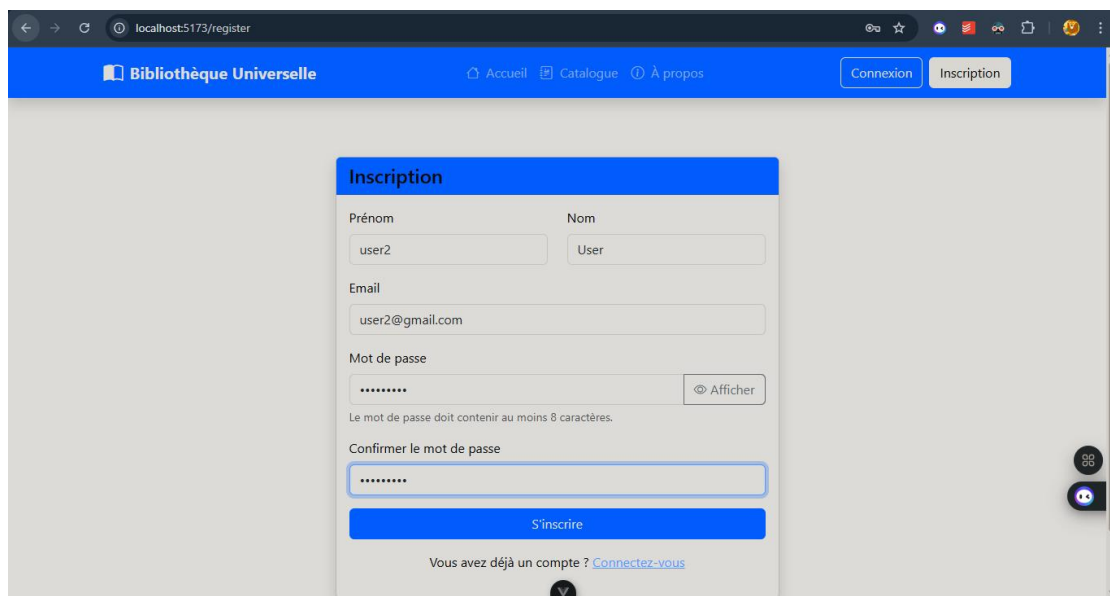
The screenshot shows the 'Gestion des Utilisateurs' page in a web application. A confirmation dialog is displayed over the table, asking 'Activer le compte de bçuçubâu jpppp ?'. The dialog has two buttons: 'Annuler' and 'Activer le compte'. The user 'bçuçubâu jpppp' is highlighted in the table.

ID	Nom	Email	Date d'inscription	Dernière connexion	Statut	Rôle	Actions
3	Bruno Essossolam GNANSA PAPISSI				Actif	Utilisateur	[Icons]
4	Admin System				Actif	Administrateur	[Icons]
2	bçuçubâu jpppp				Inactif	Utilisateur	[Icons]
1	Bruno Essossolam GNANSA PAPISSI				Actif	Utilisateur	[Icons]
6	user1 User	user1@gmail.com	29/06/2025	29/06/2025	Actif	Utilisateur	[Icons]
5	Admin System	admin@gmail.com	27/06/2025	29/06/2025	Actif	Administrateur	[Icons]

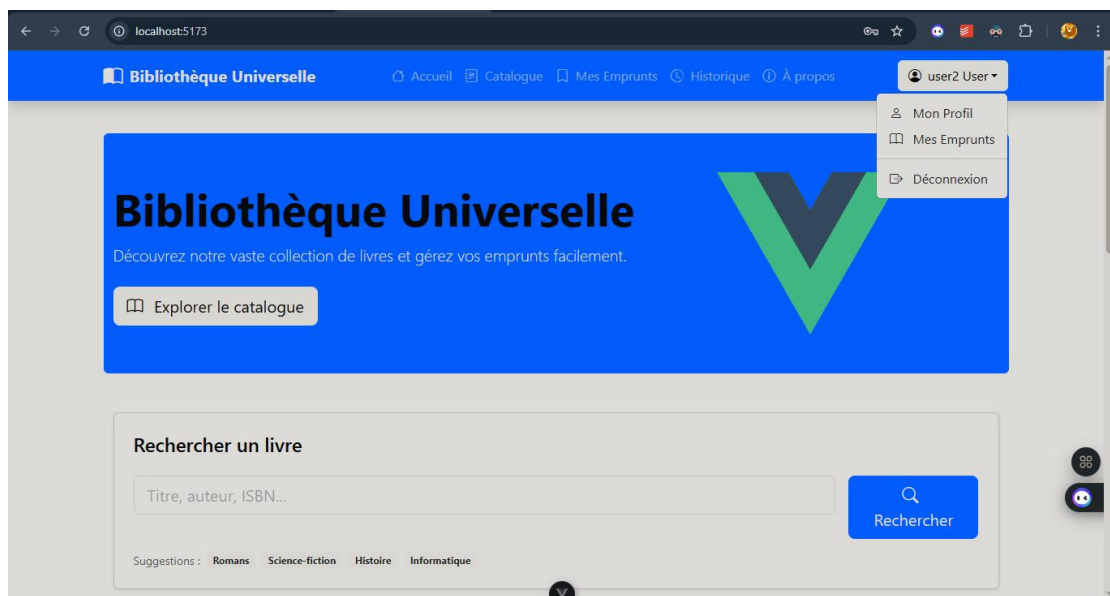
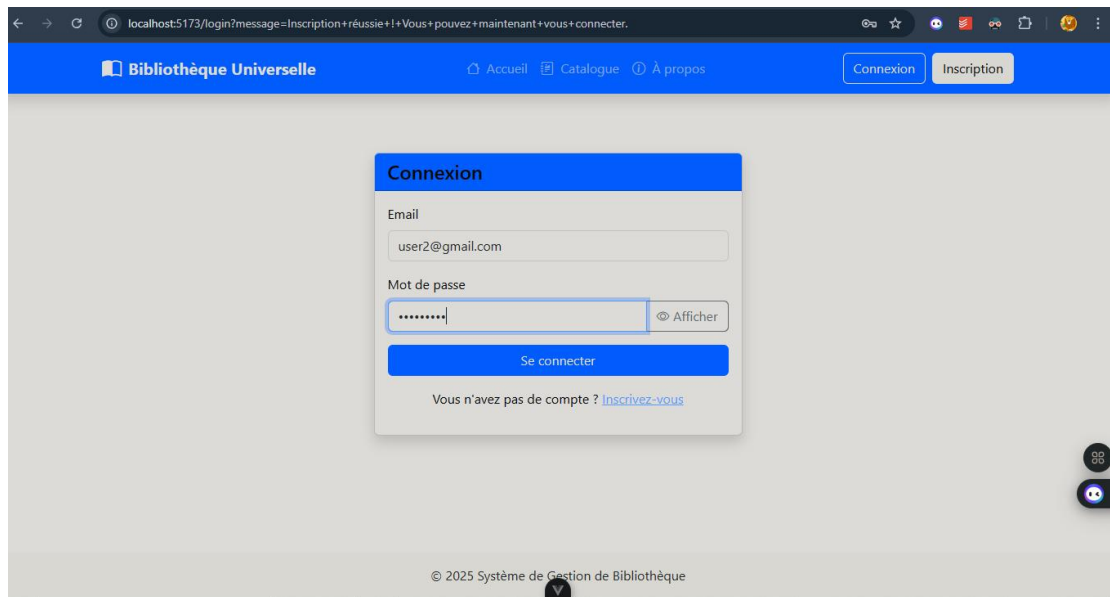


*\* Ici nous parlerons des différents cas d'utilisation d'un utilisateur sur l'app*

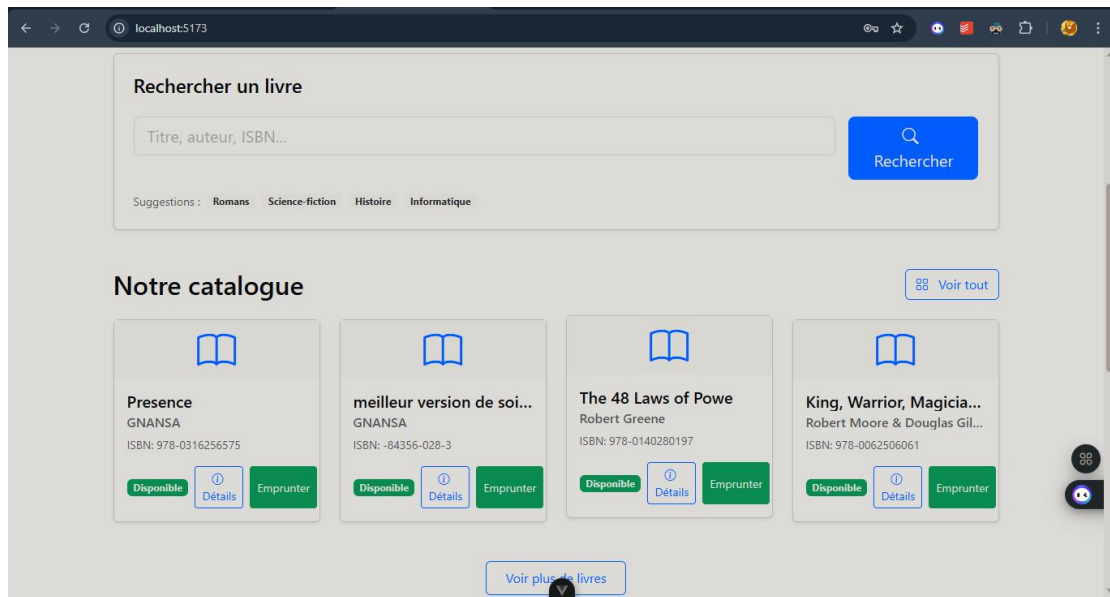
*On commence par l'inscription d'un utilisateur user2*



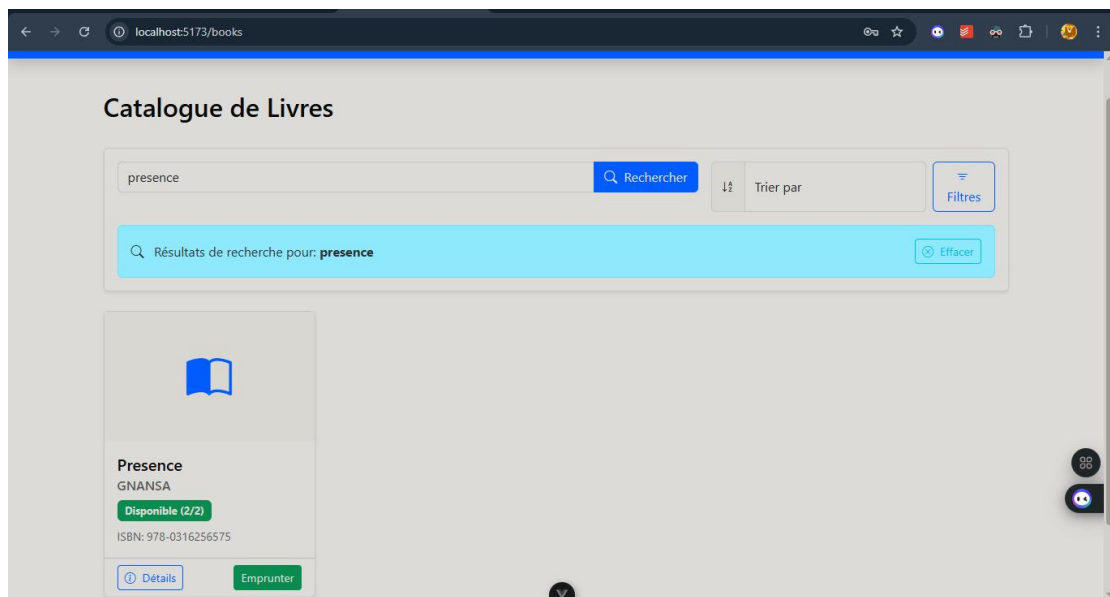
*Après une inscription réussie on redirige l'utilisateur sur la page de connexion*



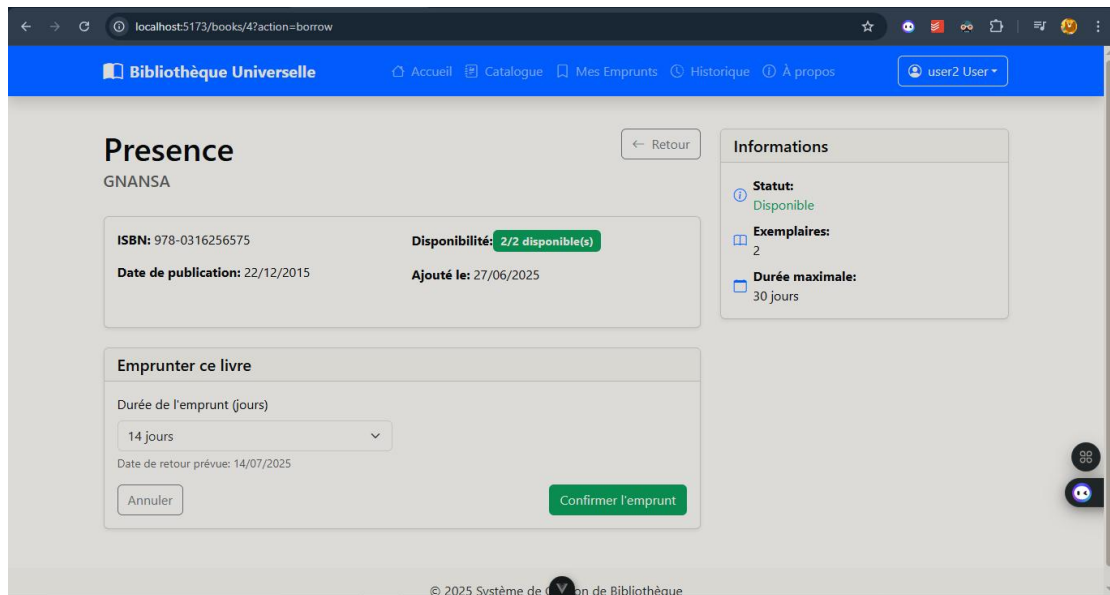
*L'utilisateur user2 est bien connecté et a accès au catalogue*



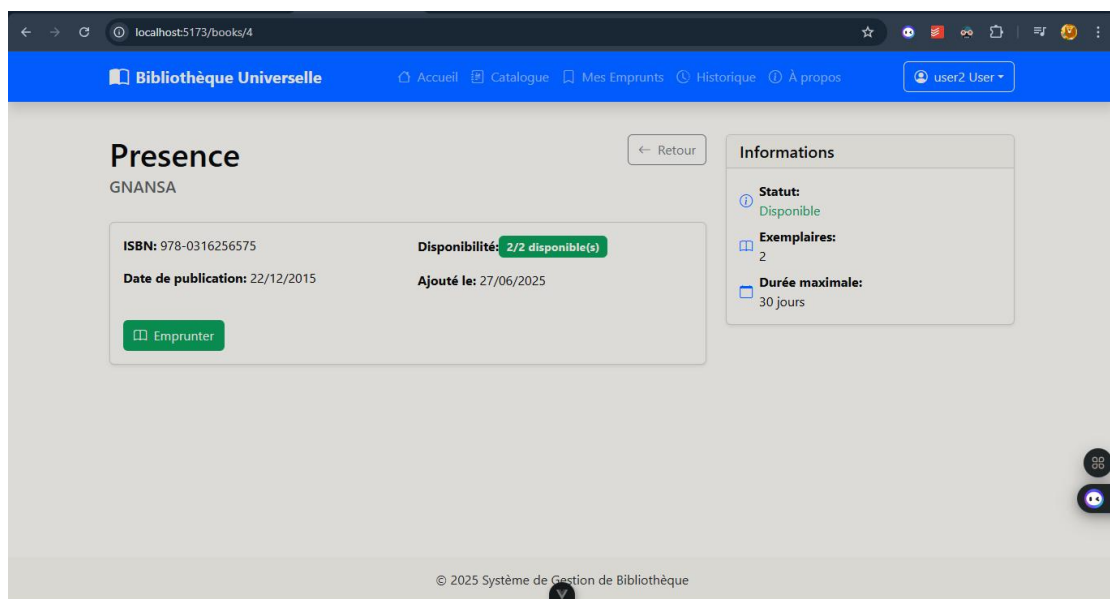
*\*L'utilisateur peut rechercher un livre comme présenter dans l'image ci dessous*



*\* Un utilisateur a la possibilité de d'emprunter un livre comme le montre l'image ci dessous*



*\*L'utilisateur peut aussi visualiser les détails d'un livre .*



## 5- Fonctionnement de flask security dans le projet

### Architecture de Sécurité

*Le système de sécurité repose sur plusieurs composants Flask :*

*\*Flask-JWT-Extended : Gestion des tokens JWT*

*\*Flask-Bcrypt : Hachage des mots de passe*

*\*Decorators personnalisés : @token\_requis et @admin\_requis*

*\*Flux d'Authentification*

*1-Inscription :*

*.Validation des données (email, mot de passe)*

*.Hachage du mot de passe avec bcrypt*

*.Création de l'utilisateur en base*

*2-Connexion :*

```
16 def connecter_utilisateur(donnees): 2 usages  Essossolam
17
18     # Obtenir l'utilisateur par email
19     try:
20         utilisateur = obtenir_par_champ_ou_404(Utilisateur, champ='email', donnees.get('email'), message="Email ou mot de pass
21     except:
22         # Utiliser le même message d'erreur pour la sécurité
23         raise ErreurNonAutorise("Email ou mot de passe invalide")
24
25     # Vérifier le mot de passe
26     if not utilisateur.verifier_mot_de_passe(donnees.get('mot_de_passe')):
27         raise ErreurNonAutorise("Email ou mot de passe invalide")
28
29     # Vérifier si l'utilisateur est actif
30     if not utilisateur.est_actif:
31         raise ErreurNonAutorise("Compte désactivé")
32
33     # Mettre à jour la dernière connexion
34     utilisateur.derniere_connexion = datetime.utcnow()
35     valider_changements()
36
37     # Générer des tokens
38     tokens = generer_tokens(utilisateur.id)
39
40     return {
41         'utilisateur': utilisateur.vers_dict(),
42         'tokens': tokens
43     }
```



### 3-Génération des tokens

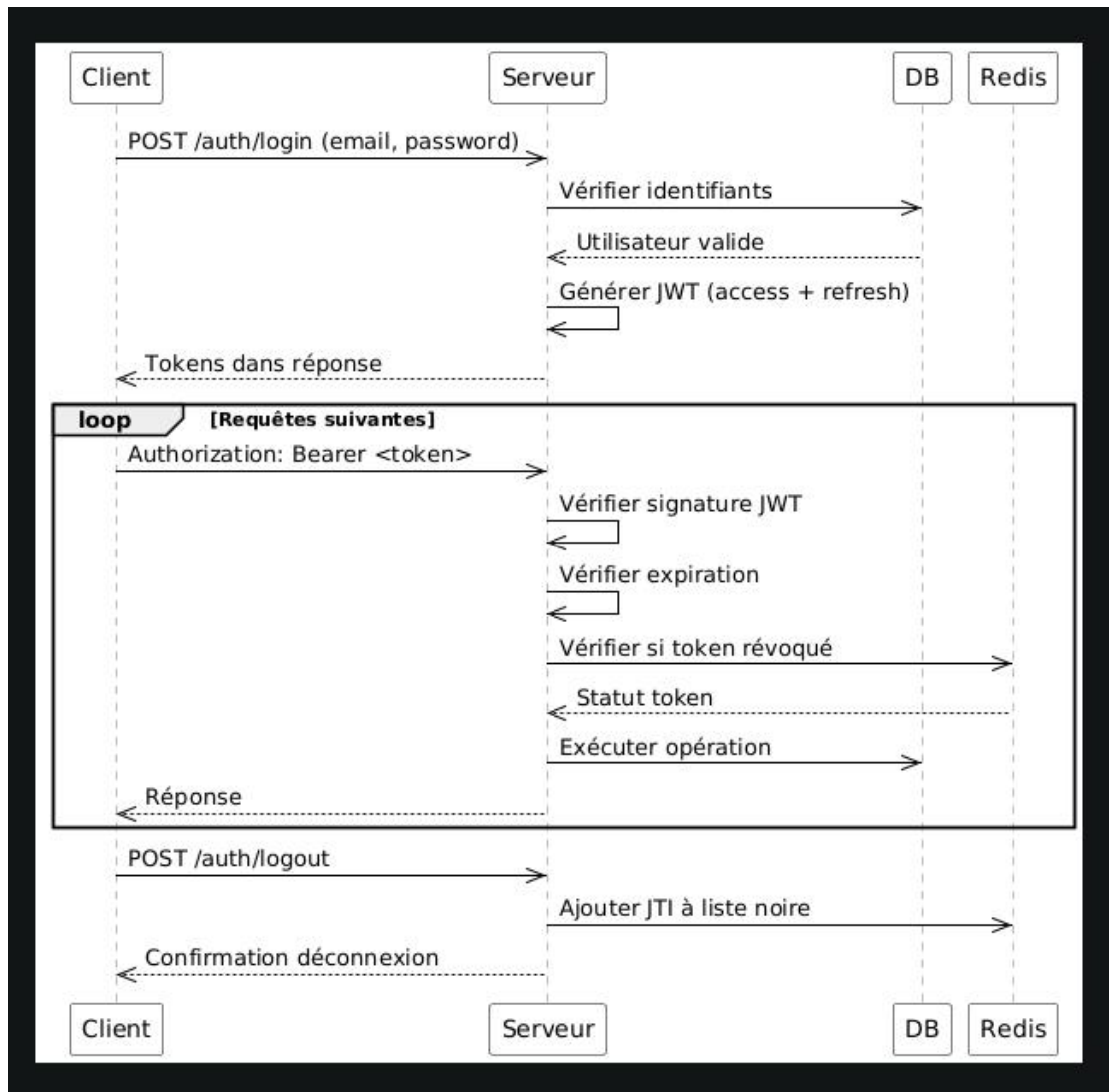
.Token d'accès (15 min de durée)

.Token de rafraîchissement (30 jours)

.Stockage des tokens révoqués

### 4-Validation des tokens

```
43
44 @token_requis new *
   Qodo Gen: Options | Test this function
45 def route_protegee():
46     # Le décorateur vérifie :
47     # 1. Présence du token
48     # 2. Validité du token
49     # 3. Non-révocation
50     # 4. Statut actif de l'utilisateur
51     pass
```



## \*Implémentation technique

```

11 # Ensemble de liste noire de tokens révoqués
12 LISTE_NOIRE = set()
13
14
15 Qodo Gen: Options | Test this function
16 def generer_tokens(utilisateur_id): 2 usages & Essosolam
17 """Générer des tokens d'accès et de rafraichissement pour un utilisateur"""
18 # CORRECTION: Convertir l'ID en string
19 identity_str = str(utilisateur_id)
20 token_acces = create_access_token(identity=identity_str)
21 token_rafraichissement = create_refresh_token(identity=identity_str)
22 return {
23     'token_acces': token_acces,
24     'token_rafraichissement': token_rafraichissement
25 }
26
27 Qodo Gen: Options | Test this function
28 def revoquer_token(jti): 2 usages & Essosolam
  
```

The screenshot shows a code editor with a project named 'ProjetBibliotheque' and a file named 'security.py'. The code defines a set 'LISTE\_NOIRE' for revoked tokens. It includes a function 'generer\_tokens' that takes a user ID, converts it to a string, and generates access and refresh tokens using 'create\_access\_token' and 'create\_refresh\_token'. It returns a dictionary with these tokens. There is also a function 'revoquer\_token' which is partially visible at the bottom.

```
25
26
27 def revoquer_token(jti): 2 usages  ⚡ Essossolam
28     """Ajouter un JWT ID à la liste noire"""
29     LISTE_NOIRE.add(jti)
30
31
32 def est_token_revoque(payload_jwt): 2 usages  ⚡ Essossolam
33     """Vérifier si un token a été révoqué"""
34     jti = payload_jwt['jti']
35     return jti in LISTE_NOIRE
36
37
38 def token_requis(fn): 12 usages  ⚡ Essossolam
39     """Décorateur pour protéger les routes avec JWT"""
40
41     @wraps(fn)  ⚡ Essossolam
```

```
77 def admin_requis(fn): 14 usages  ⚡ Essossolam
78     @wraps(fn)  ⚡ Essossolam
79     def wrapper(*args, **kwargs):
80         try:
81             verify_jwt_in_request()
82             donnees_jwt = get_jwt()
83
84             # Vérifier si le token est dans la liste noire
85             if est_token_revoque(donnees_jwt):
86                 raise ErreurNonAutorise("Le token a été révoqué")
87
88             # Obtenir l'utilisateur actuel
89             utilisateur_actuel_id = get_jwt_identity()
90
91             # CORRECTION: Convertir l'identité string en int
92             try:
93                 utilisateur_actuel_id_int = int(utilisateur_actuel_id)
94             except (ValueError, TypeError):
95
96
```

## 6-Stratégie de test

### 1) Tests unitaires

- Modèles
- Services
- Utilitaires

### 2) Tests d'intégration

- Points d'API
- Flux complets

### 3) Tests de sécurité

- Contrôle d'accès
- Validations des tokens

## 7-Améliorations possibles

Voici les améliorations possibles pour ce projet de bibliothèque :

### 1. Sécurité

- Ajout de validation du mot de passe plus stricte
- Protection contre les injections SQL
- Rate limiting sur les API
- Logs de sécurité détaillés

### 2. Fonctionnalités Utilisateurs

- Système de réservation de livres
- Notifications par email
- Système de rappels automatiques
- Liste de souhaits

### 3. Gestion du Catalogue

- Import/Export de livres en masse (CSV/Excel)
- Gestion des éditions multiples
- Système de tags/étiquettes
- Photos des couvertures
- QR codes pour les livres

### 4. Interface Admin

- Dashboard avec statistiques
- Rapports d'activité
- Gestion des amendes
- Sauvegarde automatique

- Logs d'activité détaillés

#### 5. Optimisations Techniques

- Cache Redis pour les requêtes fréquentes
- Indexation pour la recherche (Elasticsearch)
- Tests unitaires et d'intégration
- Documentation API avec Swagger
- Conteneurisation avec Docker

#### 6. Extensions du Système

- API mobile
- Support multilingue
- Système de commentaires/avis
- Intégration avec des API externes (Google Books)
- Gestion des médias numériques

#### 7. Maintenance

- Logging plus détaillé
- Monitoring des performances
- Backup automatisé
- Scripts de maintenance
- Gestion des versions de base de données

