

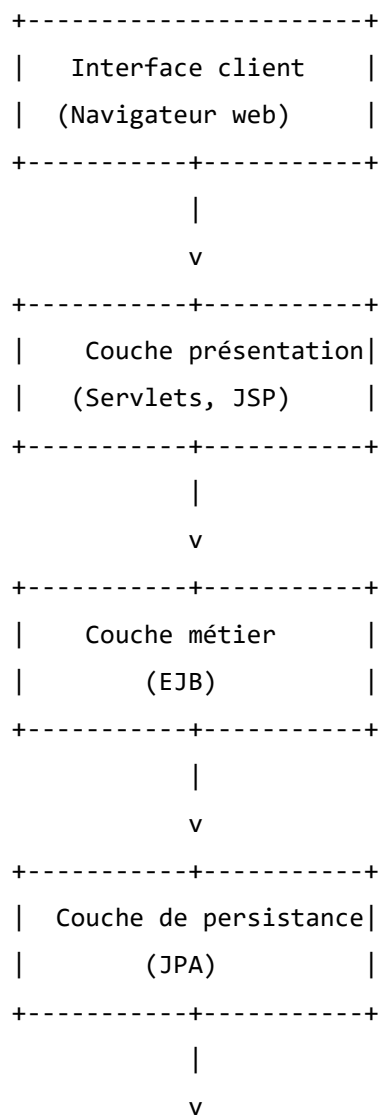
# Introduction

Ce document présente la documentation technique du système de gestion de bibliothèque, une application web développée avec les technologies Java Enterprise Edition (JEE). Cette application permet la gestion complète des ressources d'une bibliothèque incluant le catalogue des livres, les emprunts, les retours et la gestion des utilisateurs.

## Architecture technique

L'application est basée sur une architecture multi-tiers suivant le modèle MVC (Modèle-Vue-Contrôleur), garantissant une séparation claire entre les différentes couches fonctionnelles et facilitant ainsi la maintenance et l'évolution du système

### Diagramme d'architecture



```
+-----+-----+
|   Base de données   |
|   (SQLite)         |
+-----+-----+
```

## Technologies utilisées

### Front-end

- **JSP (JavaServer Pages)** : Génération dynamique des pages HTML
- **JSTL (JSP Standard Tag Library)** : Simplification du code dans les pages JSP
- **CSS** : Mise en forme et présentation des interfaces utilisateur
- **JavaScript** : Interactivité côté client

### Back-end

- **Servlets Java** : Traitement des requêtes HTTP et gestion du flux de l'application
- **EJB (Enterprise JavaBeans)** : Encapsulation de la logique métier
- **JPA (Java Persistence API)** : Framework ORM pour la persistance des données
- **JPQL (Java Persistence Query Language)** : Langage de requête orienté objet

### Base de données

- **SQLite** : Système de gestion de base de données relationnelle léger

### Environnement de développement

- **IDE** : Eclipse / IntelliJ IDEA
- **Build** : Maven
- **Versioning** : Git

## Structure de l'application

L'application est structurée selon le modèle MVC (Modèle-Vue-Contrôleur) :

### 1. Modèle (Model)

Le modèle représente les données et la logique métier de l'application :

- **Entités JPA** :
  - `Livre.java` : Représentation d'un livre dans le catalogue
  - `Lecteur.java` : Informations sur les utilisateurs de la bibliothèque
  - `Emprunt.java` : Gestion des emprunts et des retours

- Utilisateur.java : Comptes des administrateurs et employés du système

- **Services :**

- LivreService.java : Gestion du catalogue
- EmpruntService.java : Logique des prêts et retours
- LecteurService.java : Gestion des comptes lecteurs
- AuthentificationService.java : Vérification des identifiants

- **DAO (Data Access Objects) :**

- LivreDAO.java : Accès aux données des livres
- EmpruntDAO.java : Accès aux données des emprunts
- LecteurDAO.java : Accès aux données des lecteurs
- UtilisateurDAO.java : Accès aux données des utilisateurs du système

## 2. Vue (View)

La vue gère l'interface utilisateur et la présentation des données :

- **Pages JSP :**

- /login.jsp : Page d'authentification
- /dashboard.jsp : Tableau de bord principal
- /livres/liste.jsp : Affichage du catalogue
- /livres/details.jsp : Détails d'un livre
- /emprunts/gestion.jsp : Gestion des emprunts
- /lecteurs/liste.jsp : Liste des lecteurs
- /lecteurs/profil.jsp : Profil détaillé d'un lecteur

- **Ressources statiques :**

- /css/style.css : Feuilles de style pour l'interface
- /js/scripts.js : Scripts côté client
- /images/ : Ressources graphiques

## 3. Contrôleur (Controller)

Le contrôleur intercepte les requêtes HTTP et coordonne le traitement :

- **Servlets :**

- AuthentificationServlet.java : Gestion des connexions
- LivreServlet.java : Traitement des opérations sur les livres

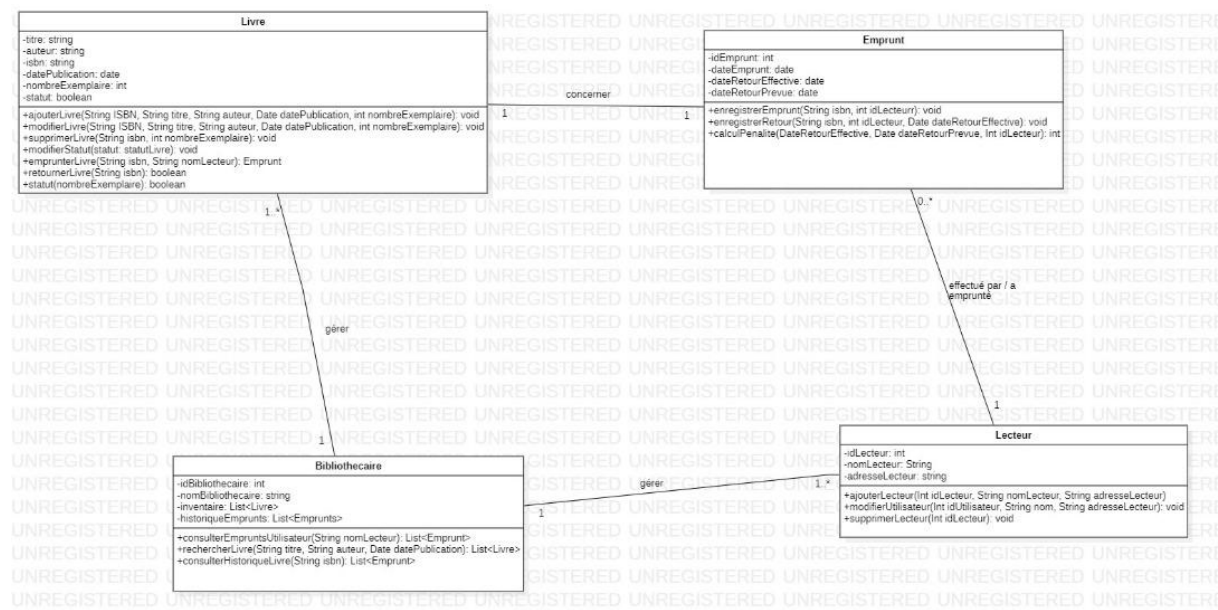
- o EmpruntServlet.java : Traitement des emprunts et retours
- o LecteurServlet.java : Gestion des lecteurs
- o DashboardServlet.java : Affichage du tableau de bord

- **Filtres :**

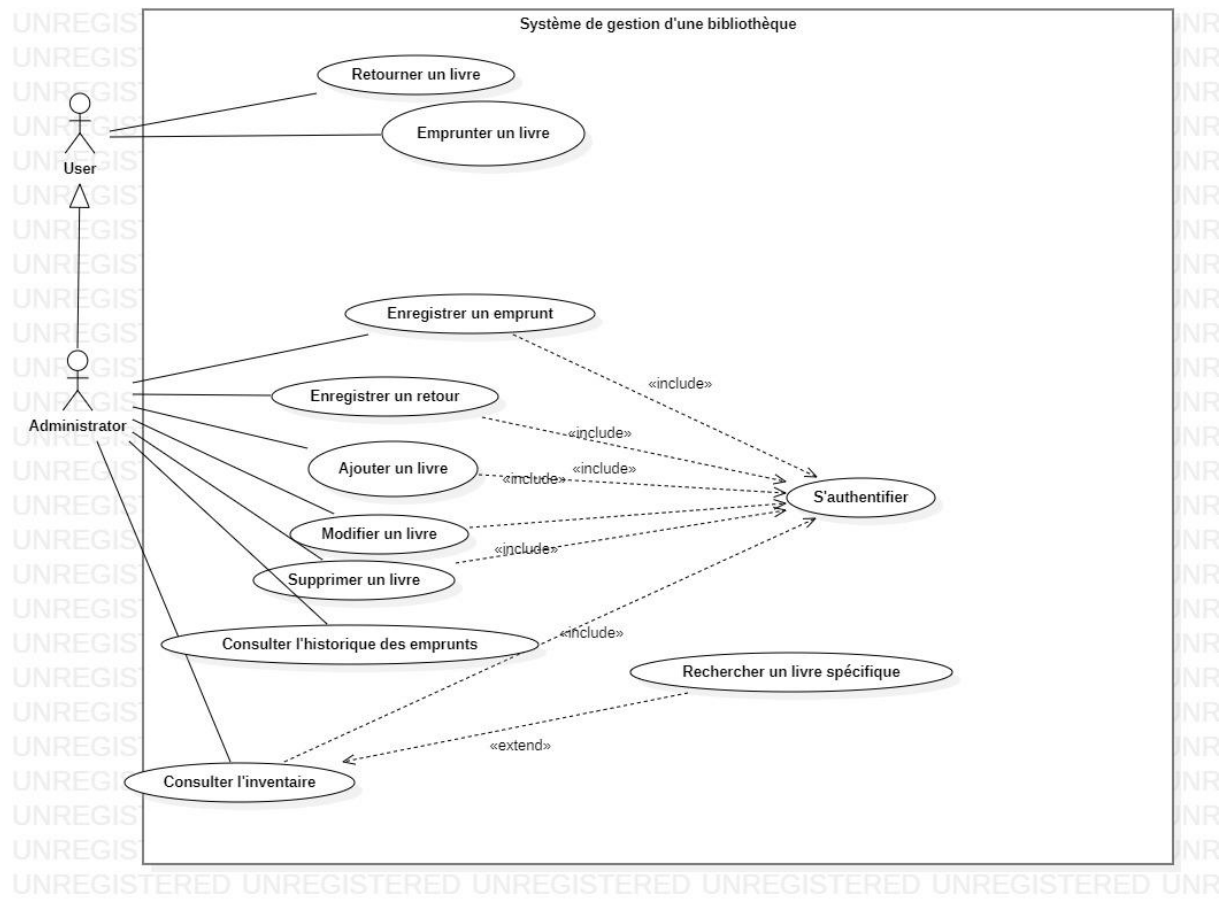
- o AuthenticationFilter.java : Vérification des sessions
- o SecurityFilter.java : Contrôle des accès par rôle

## Modélisation de l'application

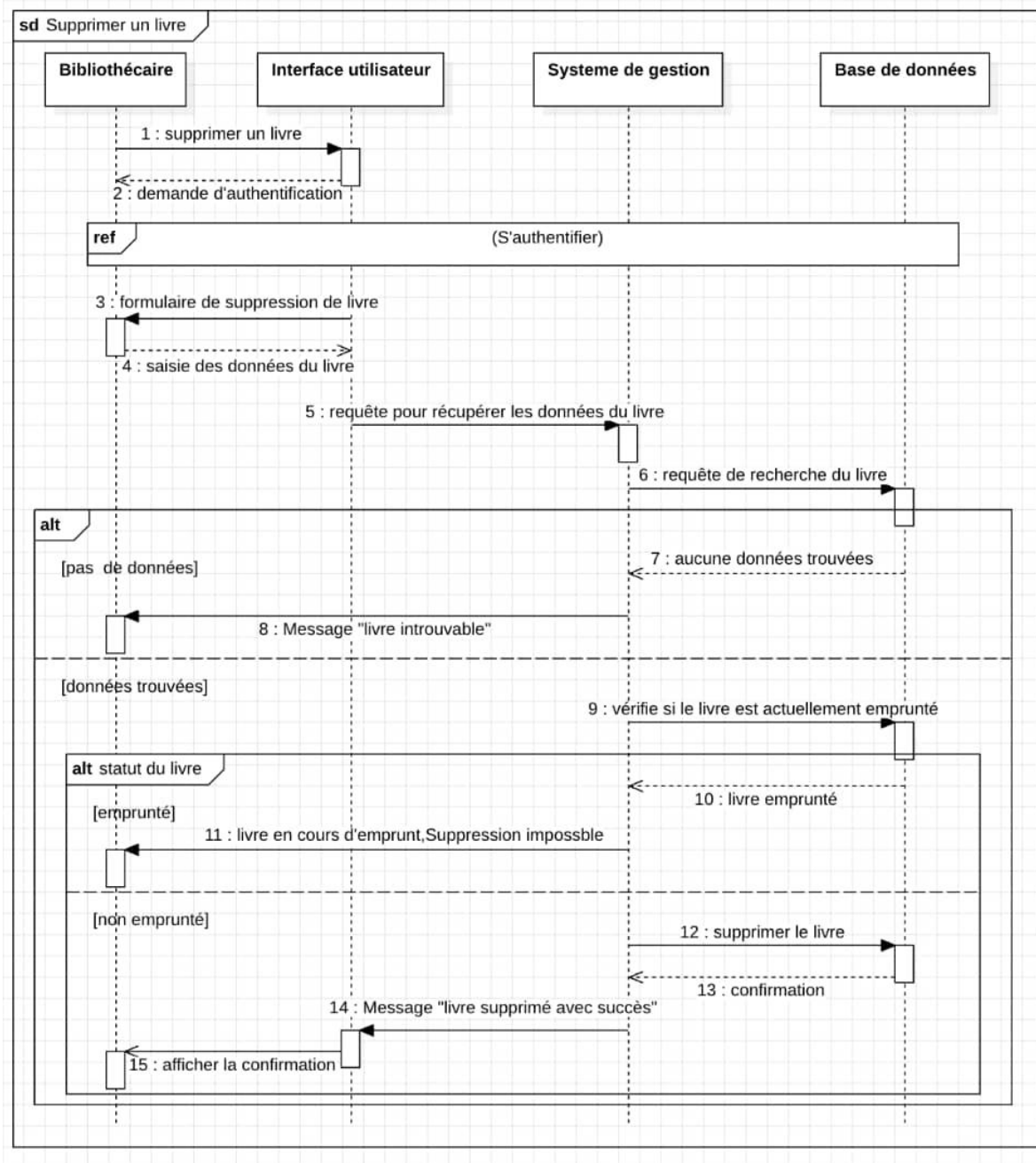
### Diagramme de classes



## Diagrammes de cas d'utilisation



## Diagrammes de sequences



# Description des interfaces utilisateur

## 1. Page d'authentification

- Formulaire de connexion avec champs pour identifiant et mot de passe
- Option de récupération de mot de passe
- Validation des données saisies avec messages d'erreur explicites

## 2. Tableau de bord principal

- Menu de navigation principal avec accès à toutes les fonctionnalités
- Synthèse des activités récentes (derniers emprunts, retours)
- Statistiques sur l'utilisation de la bibliothèque
- Alertes pour les retards de retour et autres événements importants

## 3. Gestion des livres

- Liste paginée des livres avec options avancées de recherche
- Filtrage par catégorie, auteur, disponibilité
- Fonctionnalités de tri par différents critères
- Formulaires modaux pour l'ajout et la modification
- Confirmation avant suppression
- Affichage détaillé des informations d'un livre

## 4. Gestion des emprunts

- Interface intuitive pour l'enregistrement des emprunts
- Scan de code-barres ou recherche par ID
- Affichage des emprunts en cours avec statuts (normal, en retard)
- Historique complet des emprunts avec options de filtrage
- Système de notification pour les rappels

## 5. Gestion des lecteurs

- Liste complète des lecteurs avec recherche instantanée
- Création et modification des profils lecteurs
- Affichage de l'historique des emprunts par lecteur
- Système de gestion des pénalités et blocages

# Sécurité

La sécurité de l'application repose sur plusieurs mécanismes :

## Authentification

- Système d'authentification par formulaire avec création de session
- Stockage sécurisé des mots de passe (hachage avec sel)
- Protection contre les attaques par force brute
- Déconnexion automatique après une période d'inactivité

## Contrôle d'accès

- Gestion des droits basée sur les rôles :
  - **Administrateur** : Accès complet à toutes les fonctionnalités
  - **Bibliothécaire** : Gestion des emprunts, retours et catalogue
  - **Lecteur** : Consultation du catalogue et de son compte personnel

## Protection des données

- Validation des entrées utilisateur pour prévenir les injections SQL
- Protection contre les attaques XSS
- Utilisation de tokens CSRF pour les formulaires

# Déploiement

## Environnement de production

- **Serveur d'applications** : Apache Tomcat 9.0
- **JDK** : Version 11 ou supérieure
- **Système d'exploitation** : Compatible avec Windows, Linux et macOS
- **Base de données** : SQLite 3

## Processus de déploiement

1. Compilation et packaging de l'application en fichier WAR
2. Déploiement du WAR sur le serveur Tomcat
3. Configuration des paramètres d'environnement
4. Initialisation de la base de données
5. Vérification du bon fonctionnement

## Configuration requise

- Mémoire RAM minimale : 4 Go
- Espace disque : 500 Mo minimum pour l'application
- Espace disque supplémentaire pour la base de données selon le volume de données



- Connexion réseau pour les accès multi-utilisateurs

## Maintenance et évolution

### Procédures de sauvegarde

- Sauvegarde quotidienne automatisée de la base de données
- Conservation des logs d'activité pendant 6 mois
- Procédure de restauration documentée

### Évolutivité

Le système a été conçu pour permettre :

- L'ajout de nouveaux types de ressources (DVD, magazines, etc.)
- L'intégration avec d'autres systèmes via API
- La mise en place d'un portail public pour les lecteurs
- L'implémentation de fonctionnalités avancées (réservations, suggestions)

## Annexes

### Structure de la base de données

Détail des tables et relations :

- Table LIVRE
- Table LECTEUR
- Table EMPRUNT
- Table UTILISATEUR
- Table CATEGORIE
- Contraintes et index

### Procédures d'installation détaillées

1. Installation du JDK
2. Configuration de Tomcat
3. Déploiement de l'application
4. Initialisation de la base de données
5. Premier démarrage et configuration