



REPUBLIQUE DU CAMEROUN

Paix – Travail – Patrie

UNIVERSITE DE DOUALA

ECOLE NATIONALE SUPERIEURE
POLYTECHNIQUE DE DOUALA

B.P. 2701 Douala

Tel. (237) 697 542 240

Site web: www.enspd-udo.cm

REPUBLIC OF CAMEROON

Peace – Work – Fatherland

THE UNIVERSITY OF DOUALA

NATIONAL HIGHER POLYTECHNIC
SCHOOL OF DOUALA

P O Box: 2701 Douala










Phone : (237) 697 542 240

Email: contact@enspd-udo.cm



PROJET DE PROGRAMMATION ORIENTE OBJET
**THEME : CONCEPTION D'UNE APPLICATION DE GESTION D'UNE
BIBLIOTHEQUE**

MEMBRES DU GROUPE 7

-  **DIOM LUCRAINE LETHICIA FIEN 22G00071**
-  **DJESSANG REPELE JOSEPH 22G00074**
-  **DONFACK WAMBA DYLAN OLIVIER 22G00087**
-  **KITIO MATCHOUSSA GERBEAU 22G00196**
-  **MATCHAN ELLA KETSIA ANNAELLE 24G01107**
(chef)
-  **NGOMBOUL NGAMA ARCHIMEDE 22G00312**
-  **NGOUAJIO TEGOFACK DARYL 22G00315**
-  **VOUMIA OUALA DILANE 22G00541**
-  **YAPDJO FOTSO ARTHUR BRYAN 24G01135**

Sous la Coordination de : DR NOULAPEU



INTRODUCTION

Dans un contexte où la gestion efficace des ressources documentaires devient un enjeu majeur pour les bibliothèques modernes, la mise en place d'un système informatisé de gestion des emprunts s'avère indispensable. Ce rapport présente le travail réalisé dans le cadre du développement d'une application web de gestion des emprunts et retours de livres, utilisant les technologies **Java (JSP/Servlets)**.

L'objectif principal de ce projet est de faciliter le suivi des livres empruntés par les membres, d'automatiser les processus de prêt et de retour, et d'optimiser la gestion des délais ainsi que des amendes en cas de retard. Grâce à une interface intuitive et à des fonctionnalités interactives, le système permet de centraliser les informations relatives aux membres, aux livres et aux transactions d'emprunt.

Ce document détaille les différentes étapes de développement, les choix techniques effectués, ainsi que les fonctionnalités implémentées pour répondre aux besoins d'un gestionnaire de bibliothèque.

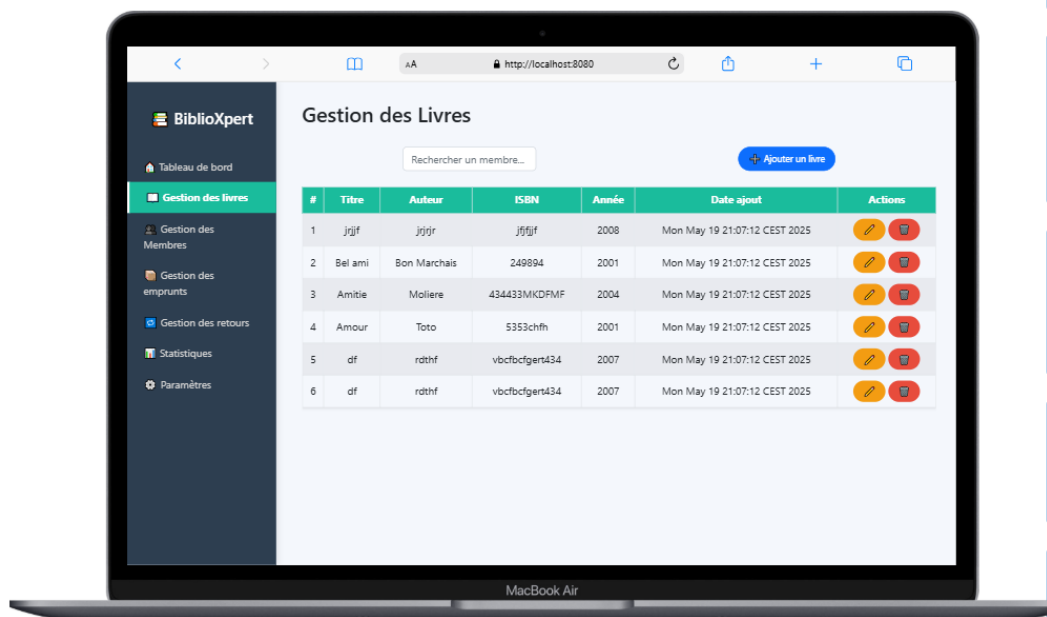
I. OBJECTIFS DU PROJET

L'objectif de ce projet est de concevoir et développer une application Java destinée à faciliter la gestion des opérations courantes au sein d'une petite bibliothèque. Cette application permettra notamment l'enregistrement des ouvrages, la gestion des emprunts et des retours, ainsi que la consultation en temps réel de l'inventaire des livres disponibles.

II- FONCTIONNALITES PRINCIPALE

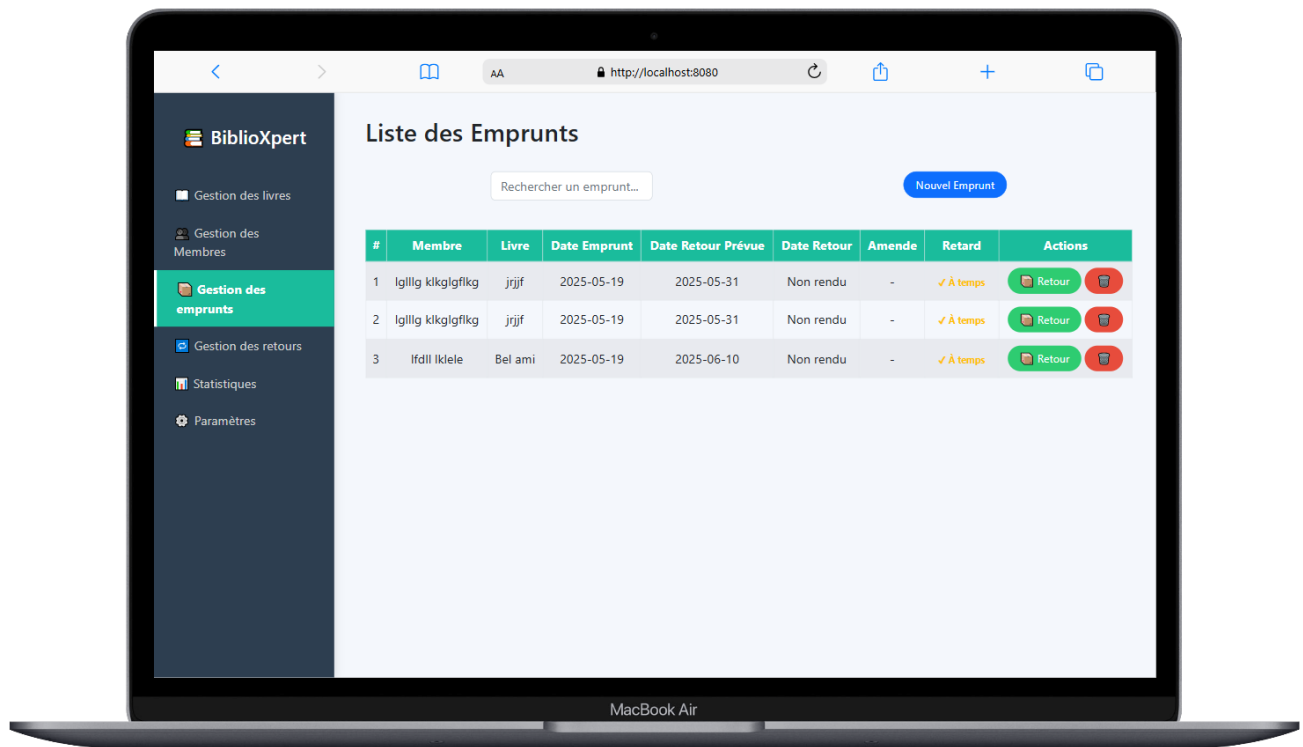
a- Gestion des Livres

- **Ajout d'un livre** : Saisir les informations de base (titre, auteur, ISBN, année de publication).
- **Suppression et modification** : Permettre la mise à jour ou la suppression d'un livre existant.
- **Consultation de l'inventaire** : Afficher la liste complète des livres disponibles.

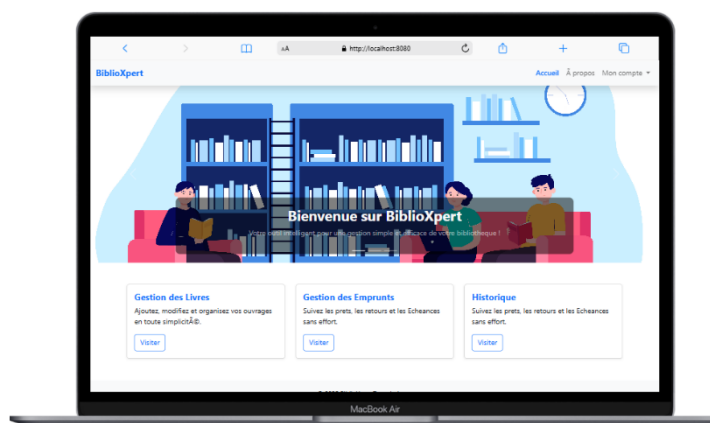


b- Gestion des Emprunts

- **Emprunt d'un livre** : Enregistrer l'emprunt par un utilisateur (nom, date d'emprunt).
- **Retour de livre** : Enregistrer le retour et mettre à jour le statut du livre.
- **Historique** : Consulter l'historique des emprunts pour chaque livre ou utilisateur.



c- Interface Utilisateur Console



- Un menu principal permettant de naviguer entre les fonctionnalités (gestion des livres, gestion des emprunts, consultation).
- Gestion des entrées utilisateur avec des validations de saisi

III- EXIGENCES TECHNIQUES ET OUTILS UTILISES

- ✚ **Langage de programmation** : Le projet est développé en Java, version 11, afin de bénéficier des dernières fonctionnalités du langage.
- ✚ **Méthodologie** : L'application repose sur une approche orientée objet, facilitant la modélisation, la réutilisabilité du code et la maintenabilité du projet.
- ✚ **Environnement de développement** : Le développement a été réalisé à l'aide d'un IDE tel que NetBeans.
- ✚ **Base de données** : Les données sont stockées dans une base MySQL **app_bibliotheque**, avec une communication assurée par le connecteur JDBC pour permettre les échanges entre l'application Java et la base de données.

IV- ARCHITECTURE DE L'APPLICATION

L'architecture de notre application repose sur une structure MVC (Modèle – Vue – Contrôleur), couramment utilisée dans les projets web pour assurer une séparation claire des responsabilités :

- **Modèle (Model)** : Représente les entités de l'application, comme Livre, Utilisateur, ou Emprunt. Ces classes sont directement liées à la base de données via JPA (Jakarta Persistence API).
- **Vue (View)** : Le front-end de l'application a été réalisé avec Jakarta EE (technologies JSP/Servlets), permettant d'afficher dynamiquement les pages web à partir des données du back-end. Des balises JSTL ont été utilisées pour simplifier le rendu dynamique dans les pages JSP.
- **Contrôleur (Controller)** : Les Servlets jouent le rôle de contrôleur. Elles interceptent les requêtes utilisateur, exécutent la logique appropriée (via les services métier), et redirigent vers les pages de résultats.

•

L'application suit donc une architecture en couches :

1. **Présentation** (JSP)
2. **Contrôleur** (Servlets)
3. **Service** (logique métier en Java ou model)
4. **DAO** (accès aux données avec JDBC ou JPA)
5. **Base de données** (MySQL) **app_bibliotheque**

Choix techniques adoptés

Langage de programmation : Java

Framework/Technologies :

- **Jakarta EE** (anciennement Java EE) pour le développement web (Servlets, JSP)
- **JPA** (avec Hibernate) pour la gestion de la persistance des données
- **JDBC** utilisé pour certaines opérations de base
- Base de données : MySQL
- IDE : **NETBEANS**
- Serveur d'applications : **Apache Tomcat**
- Système de gestion de version : Git (avec GitHub pour la collaboration)
- Structure du projet : Projet **dynamique** avec dossier src, **WebContent**, **WEB-INF**, etc.

Ces choix ont été faits pour :

- Utiliser des outils professionnels standards.
- Favoriser la réutilisabilité et la maintenabilité du code.
- Mettre en œuvre une vraie séparation des couches.

Manuel d'utilisation de l'application

Voici un guide pour faciliter la prise en main de l'application :

1. Démarrage de l'application

- Ouvrir le projet dans l'IDE.
- Configurer le serveur Tomcat ou GlassFish.
- Démarrer le serveur.
- Accéder à l'application via un navigateur à l'adresse : **<http://localhost:8080/WebApplication1/page-jsp/Accueil.jsp>**
-

2. Fonctionnalités principales

Gestion des Livres

- Ajouter un livre : Aller dans la section “Livres” puis cliquer sur “Ajouter”. Remplir les champs (titre, auteur, ISBN, année) et valider.
- Modifier un livre : Cliquer sur “Modifier” à côté d'un livre dans la liste.
- Supprimer un livre : Cliquer sur “Supprimer” à côté du livre concerné.
- Voir l'inventaire : Une liste complète s'affiche avec les détails des livres disponibles.

Gestion des Emprunts

- Emprunter un livre : Aller dans “Emprunts”, sélectionner un livre et un utilisateur, puis valider.
- Retourner un livre : Cliquer sur “Retour” dans la liste des emprunts.
- Voir l'historique : Consulter la liste des livres empruntés avec les dates et les utilisateurs.

Interface Utilisateur

- L'interface est simple avec un menu principal permettant d'accéder aux différentes sections :
→ Livres | Emprunts | Historique | Accueil

3. Connexion (si nécessaire)

- Si une connexion est requise, utiliser les identifiants fournis par l'administrateur ou définis en base de données.

V- REPARTITION DES TACHES

Phases	Tâches	Responsables	Durée	Dépendances	Semaines (1 à 8)
Planification	1-Définition des besoins	Toute l'équipe	1 semaines		1
	2-Répartition des rôles	MATCHAN (chef projet)		1	
Architecture technique	3-Cahier des charges	DIOM	1 semaine	1	2
	4-Diagrammes UML	DONFACK & KITIO	1 semaine	1	3
Développement	5-Backend (BD)	MATCHAN, VOUMIA, YAPDJO	3 semaines	3 & 4	4-6
	6-Frontend (web mobile)	NGOUAJIO, DJESSANG, NGOMBOUL	2 semaines	3 & 5	5-6
	7-Intégration backend & frontend	MATCHAN	1 semaine	5 & 6	7
Tests	8-tests fonctionnels	MATCHAN, VOUMIA	2 jours	7	8
	9-correction des bugs	Toute l'équipe	3 jours	8	8
Documentation	10-Redaction du rapport	VOUMIA	5 jours	1 à 9	8

VI- DEROULEMENT DU PROJET

Le projet s'est articulé autour de plusieurs étapes clés, réparties sur une période d'environ **sept semaines**. Chaque phase a contribué progressivement à la mise en œuvre complète de l'application.

1. Analyse des besoins

Cette première étape a consisté à identifier et définir les **fonctionnalités essentielles** attendues d'un système de gestion pour une petite bibliothèque.

2. Conception

Nous avons élaboré des **maquettes fonctionnelles** ainsi que les **modèles de données** (diagrammes de classes, schémas de base de données) afin de structurer le développement.

3. Développement

L'implémentation s'est faite de manière **modulaire**, en construisant progressivement chaque composant de l'application, selon une approche orientée objet.

4. Tests

À cette étape, nous avons procédé à des **vérifications systématiques** pour nous assurer que chaque fonctionnalité répondait bien aux attentes et fonctionnait correctement.

5. Livraison

Une fois le développement finalisé, le **livrable a été partagé** via un **dossier Drive** destiné à l'enseignante, en vue de la présentation finale.

VII- DIAGRAMME UML

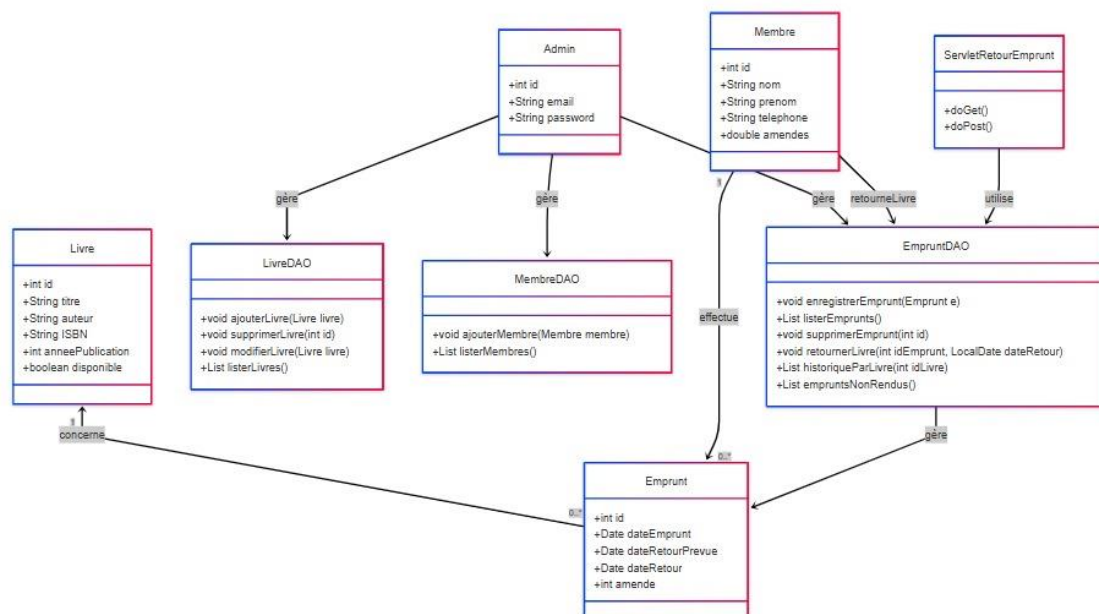


Figure 1 Diagramme de classe

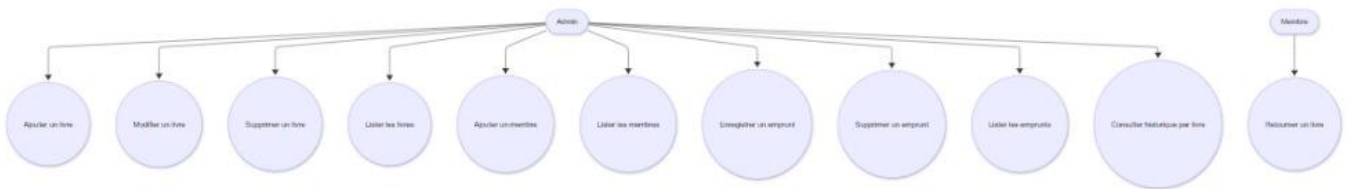


Figure 3 Diagramme ce cas d'utilisation

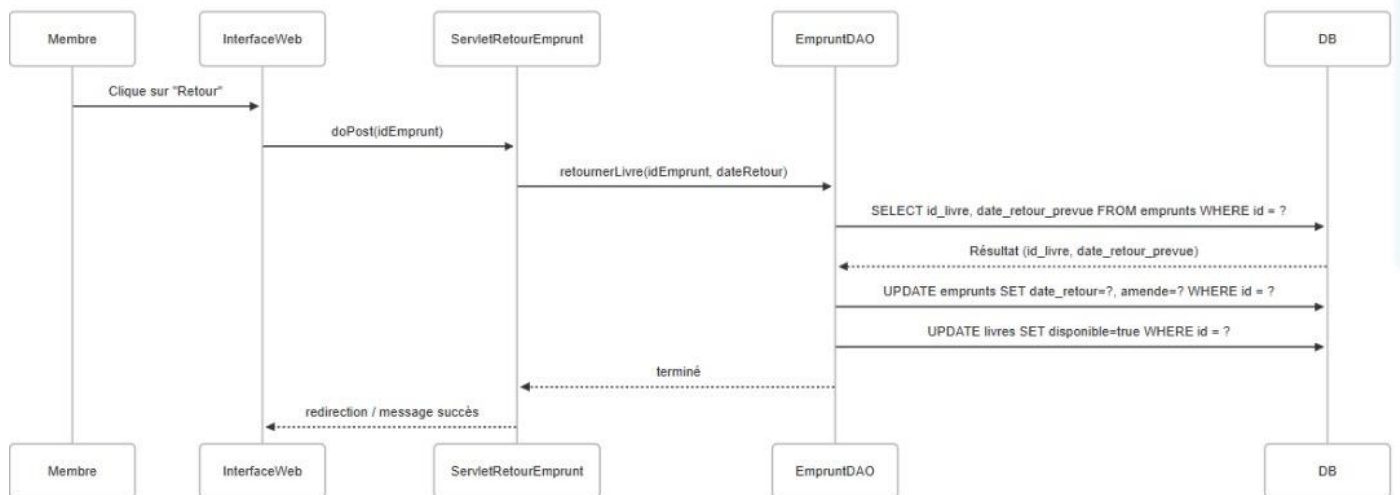


Figure 2 Diagramme de Sequence

VIII- RESULTATS OBTENUS

L'application développée est fonctionnelle et comprend l'ensemble des fonctionnalités de base. Les tests d'utilisateurs ont révélé une bonne ergonomie et une performance satisfaisante. L'interface d'administration reste perfectible, et certaines fonctionnalités comme les notifications de retour ou la messagerie interne pourront être rajouter ultérieurement.

DIFFICULTES RENCONTREES

- Coordination des taches dans le travail à distance
- Problème de synchronisation entre le front-end et la base de données
- Difficulté à réaliser le front-end avec JAVA pour une application mobile, alors nous avons utilisé le web java avec Maven et le front-end avec Jakarta EE
- Erreur 500 lors de l'insertion des livres et de la connexion
- Erreur 404 lors de la configuration des routes vers le servlet.
- Difficultés à exécuter le code JS
- Difficulte a implementer la fonctionnalite retour des livres empruntes.

CONCLUSION

La réalisation de ce projet de gestion d'une bibliothèque en Java nous a permis de mettre en pratique les notions fondamentales de la programmation orientée objet, ainsi que les compétences acquises en conception de base de données, modélisation et structuration d'une application logicielle.

Grâce à une approche progressive – de l'analyse des besoins à la livraison finale – nous avons développé une application fonctionnelle répondant aux attentes de base d'un système de prêt et de gestion d'ouvrages. Ce projet nous a également permis de mieux appréhender le travail en équipe, l'importance de l'organisation du code, et la rigueur nécessaire dans le processus de développement logiciel.

En somme, cette expérience a été formatrice tant sur le plan technique que méthodologique, et représente une étape importante dans notre parcours académique en informatique.

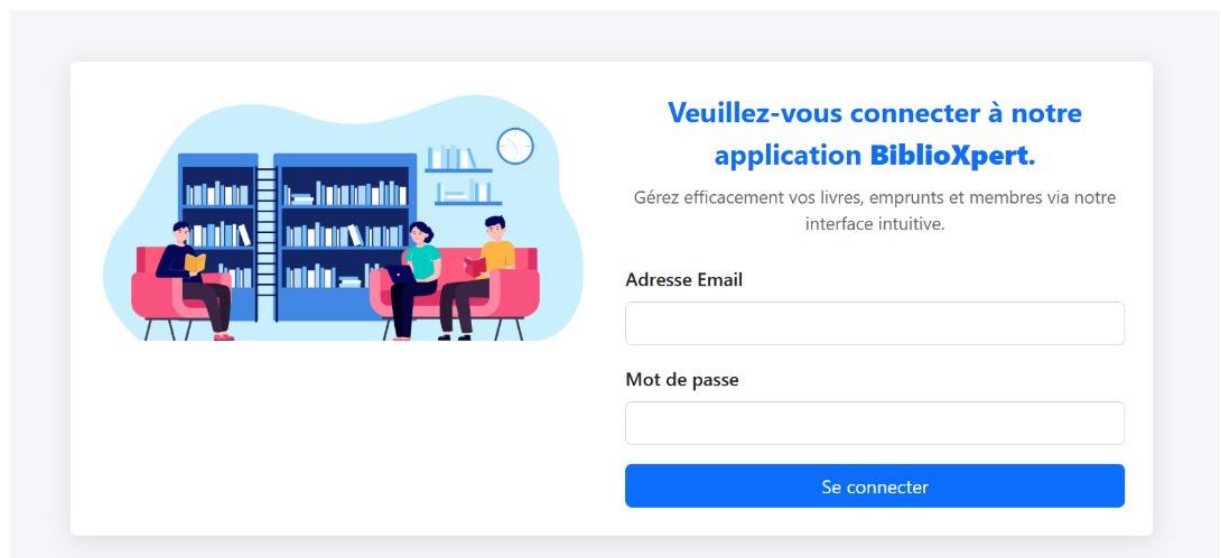
ANNEXES

1- Captures d'écrans de l'application

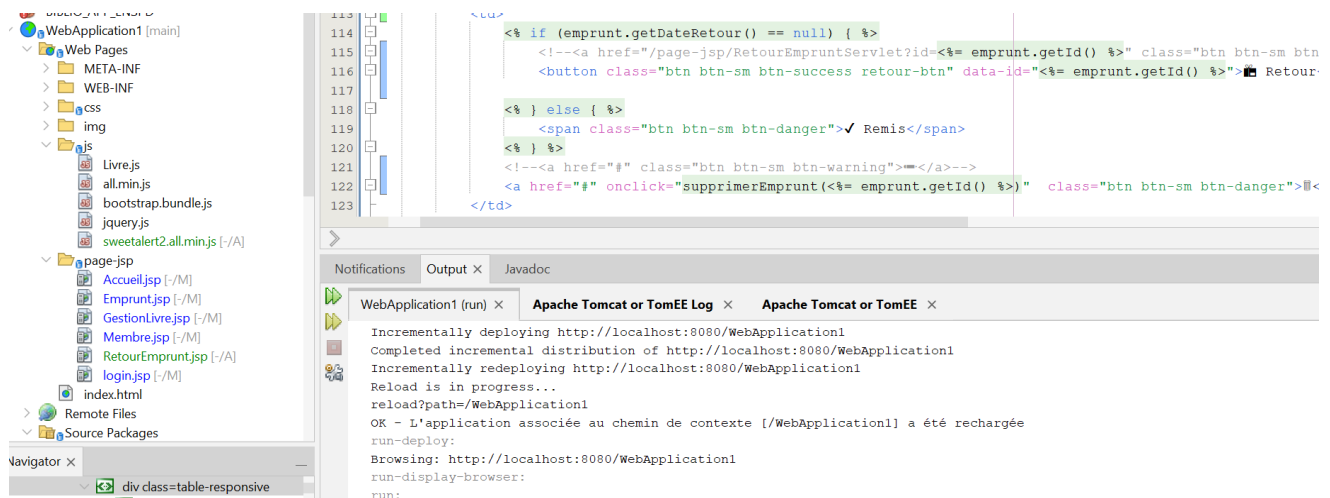
a- Page de connexion

b-

Informations de connexion : email : ellaketsy@gmail.com et Mot de passe : Meka@2004



2- Architecture de l'application



3- Lien du code source GitHub

Lien : <https://github.com/KetsiaAnnaelle/BiblioXpert-GLO3-groupe-7-projet-appllication-de-gestion-de-bibliotheque.git>