

**Université Hassan Premier**  
*Faculté des sciences et techniques*  
*Settat*

***CRÉATION D'UN SITE WEB POUR LA  
GESTION D'UNE BIBLIOTHÈQUE***

Réalisé par :

**HAMZI Med Amine  
BENLAGHRISSE Youssef  
ELMKEDMI Latifa  
JMOUHI Hajar  
BAGHADI Kaoutar**

**“ JUST BECAUSE YOU  
AREN'T WHERE THEY  
ARE DOES NOT MEAN  
YOU DON'T HAVE  
WHAT IT TAKES”**

# Remerciement

En préambule à ce projet Louanges à Dieu qui est l'origine de toute réussite dans notre vie, puis nous souhaiterions adresser notre remerciement aux personnes qui nous ont apporté leurs aides et qui ont contribué à l'élaboration de ce projet.

Nous souhaitons adresser nos remerciements au corps professoral et administratif de l'université, pour la qualité de l'enseignement offerte et le soutien de l'équipe administrative.

Nous remercions également tous nos enseignants pour leurs efforts qui nous ont guidés et qui ont enrichi notre travail tout au long de nos études universitaires.

Enfin, notre remerciement s'adresse aussi à tous ceux qui ont participé de près ou de loin, à l'élaboration de ce projet de fin d'études et en particulier à nos familles et nos amis.

# Résumé

Le présent document est le fruit de notre projet, qui consiste à réaliser un site web responsive qui permet de faciliter le prêt des livres dans une bibliothèque, en gérant les emprunts et les réservations des livres, de fournir également à l'administrateur un outil de communication avec les clients.

La première étape a été réservée pour définir le contexte général du projet en mettant l'accent sur la problématique à laquelle notre projet doit répondre ensuite nous avons présenté l'étude fonctionnelle où on a défini les besoins fonctionnels et non fonctionnels ainsi que les diagrammes nécessaires à la conception du logiciel.

La deuxième partie s'est consacrée à l'étude des outils techniques que nous avons choisis pour travailler aussi qu'à la réalisation du projet en présentant les différentes interfaces

# Abréviations

CSS : Cascading Style Sheets  
DAO : *Data Access Object*  
HTML : *HyperText MArkup Language*  
IDE : *Integrated Development Environment*  
IOC : *Inversion of control*  
JAR : *Java Archive*  
Java EE : *Java Entreprise Edition*  
JDK : *Java Developpement Kit*  
JPA : *Java Persistence API*  
JRE : *Java Runtime Environement*  
JSON : *JavaScript Objet Notation*  
JSR : *Java Specification requests*  
JWT *JSON Web Token*  
MVC : *Model view Controller*  
Rest : *Representational State Transfer*  
SQL : *Structed Query Language*  
UML : *Unified Modeling Language*  
WAR : *Web Application Resource*  
XML : *Extensible Markup Language*

# Table des matières

Table des matières .....	6
Table des figures .....	9
Introduction générale .....	10
I. Contexte général du projet .....	11
A. Introduction .....	12
B. Contexte et Problématique .....	12
1. Contexte du projet .....	12
2. Problématique : .....	12
3. Solution : .....	12
C. Conduite du Projet .....	13
1. Méthodologie du travail .....	13
a) Méthode SCRUM .....	13
b) Processus de développement .....	16
c) Planification de projet .....	19
D. Objectif : .....	19
E. Conclusion .....	20
II. Cahier de charge .....	21
A. Introduction .....	22
B. Etude de besoins .....	22
1. Besoins fonctionnels : .....	22
a) Les besoins fonctionnels liés aux livres : .....	22
b) Les besoins fonctionnels liés aux catégories : .....	22
c) Les besoins fonctionnels liés aux clients : .....	22
d) Les besoins fonctionnels liés aux emprunts : .....	22
e) Les besoins fonctionnels liés aux réservations : .....	23
2. Besoins fonctionnels : .....	23
C. Conclusion .....	23

III.	Conception du projet .....	24
A.	Introduction : .....	25
B.	Diagramme de cas d'utilisation : .....	25
C.	Diagramme de classes : .....	25
D.	Conclusion : .....	26
IV.	Etudes techniques et technologiques .....	27
A.	Introduction .....	28
B.	Architecture technique .....	28
C.	Technologies et bibliothèques utilisés .....	29
1.	Spring Boot .....	29
2.	Spring Security .....	30
3.	JSON Web Token.....	30
4.	Spring Mail.....	31
5.	Spring Data JPA .....	31
6.	Spring Data Rest.....	31
7.	Hibernate.....	31
8.	Maven.....	32
9.	Bootstrap .....	33
10.	Angular .....	33
11.	PrimeNG .....	34
D.	Langages de programmation.....	34
1.	JAVA.....	34
2.	JAVA EE .....	36
3.	SQL.....	37
4.	Typescript .....	37
E.	Environnement Logiciels .....	38
1.	PHPMYADMIN .....	38
2.	WAMPSEVER.....	39
3.	STARUML .....	40
4.	IntelliJ IDEA .....	40

F. Conclusion .....	41
V. Réalisation du projet .....	42
A. Introduction .....	43
B. Interfaces Homme-Machine .....	43
C. Conclusion .....	54
VI. Test .....	55
A. Introduction .....	56
B. Tests unitaires .....	56
1. Test de l'envoi des emails .....	56
C. Tests d'intégration .....	57



# Table des figures

Figure 1 : Le processus de la méthodologie SCRUM .....	14
Figure 2 : Planification du projet .....	19
Figure 3: Diagramme cas d'utilisation.....	25
Figure 4 : Diagramme de classes .....	26
Figure 5 : L'architecture technique d'application .....	28
Figure 6 : Interface d'accueil .....	43
Figure 7 : Interface de connexion .....	44
Figure 8 : Interface de gestion des clients .....	44
Figure 9 : Interface de modification des informations d'un client .....	45
Figure 10 : Interface de création des informations d'un client .....	45
Figure 11 : Interface de gestion des catégories .....	46
Figure 12 : Interface de modification d'une catégorie.....	46
Figure 13 : Interface de création d'une catégorie.....	47
Figure 14 : Interface de gestion des livres .....	47
Figure 15 : Interface de modification d'un livre.....	48
Figure 16 : interface de création d'un livre.....	48
Figure 17 : Interface de gestion d'un emprunt .....	49
Figure 18 : Interface de chargement pour modifier un emprunt.....	50
Figure 19 : Interface de gestion des emprunts .....	50
Figure 20 : Interface de modification d'un emprunt.....	51
Figure 21 : Interface de gestion des emprunts .....	51
Figure 22 : Interface de création d'un emprunt.....	52
Figure 23 : Interface de gestion des réservations .....	53
Figure 24 : interface de modification d'une réservation.....	53
Figure 25 : Interface de création d'un emprunt.....	54
Figure 26 : Interface de test unitaires.....	57
Figure 27 : Interface de tests d'intégration.....	58

# Introduction générale

Une bibliothèque est le lieu où est conservée et lue une collection organisée de livres. Il existe des bibliothèques privées et des bibliothèques publiques. Les bibliothèques proposent souvent d'autres documents (journaux, périodiques, enregistrements sonores, enregistrements vidéo, cartes et plans, partitions) ainsi que des accès à internet et sont parfois appelées médiathèques ou informathèques.

La majorité des bibliothèques (municipales, universitaires) permettent gratuitement la consultation sur place ainsi que le prêt de documents. D'autres, comme la Bibliothèque publique d'information notamment, n'autorisent que la consultation sur place.

Dans ce document on va vous présenter une solution web pour la gestion de prêt des livres en suivant la démarche suivante :

Le premier chapitre intitulé **présentation du projet** est consacré à une image globale du projet et aux outils utilisés.

Le deuxième chapitre intitulé **cahier de charge** contient une étude détaillée du problème, ainsi que les besoins fonctionnels et non fonctionnels.

Le troisième chapitre intitulé **conception et modélisation du projet**, il est dédié à la conception des patrons utilisés et des modèles de données.

Le quatrième chapitre intitulé **réalisation**, présente des démonstrations du projet.

Le rapport s'achève par une conclusion.

# **I. Contexte général du projet**

## **A. Introduction**

Ce premier chapitre met au clair le projet à réaliser. Il s'organise autour du contexte général et la problématique du projet ainsi que les objectifs fixés et la conduite du projet.

## **B. Contexte et Problématique**

### **1. Contexte du projet**

Ce projet s'inscrit dans le cadre de l'informatisation du système de gestion des emprunts d'une bibliothèque, également dans le but de créer et de gérer les réservations des clients. La problématique est de garantir, par le moyen d'une application, la bonne gestion des livres aussi bien de faciliter la recherche des livres disponibles, elle permettra également de mettre en relation l'administrateur avec les clients de la bibliothèque.

### **2. Problématique :**

Depuis longtemps, les emprunts des livres dans une bibliothèque sont traités de façon manuel, on cherche à les manipuler d'une manière informatique pour faciliter le traitement et améliorer le niveau de prêt dans une bibliothèque par règles à respecter par les clients. En outre le système doit gérer aussi la communication d'administrateur avec les clients. Parfois les livres ne sont pas disponibles dans la bibliothèque, donc le système doit se charger d'informer les clients si le livre est disponible et cela peut être réalisé par des réservations des livres.

### **3. Solution :**

Le développement d'une Application Web Responsive qui gère le prêt des livres, en commençant par chercher la disponibilité du livre demandé, s'il existe l'administrateur enregistre les informations d'emprunteur et associe le livre au client par l'ajout des prêts, sinon l'administrateur peut faire la réservation du livre demandé pour que le système informe le client si jamais le livre est disponible. L'application se charge d'appliquer des règles pour les clients qui se tardent de rendre les livres.

## C. Conduite du Projet

### 1. Méthodologie du travail

Pour atteindre les objectifs du projet, il est indispensable de suivre une méthodologie de conduite. Parce que l'adéquation du projet au processus de développement peut affecter le sort d'un projet informatique. Donc, un mauvais choix peut amener le projet à l'échec. Lors de la réalisation de notre projet nous avons le choix de poursuivre plusieurs méthodologies, cependant nous avons suivi la méthode SCRUM, pour bénéficier de ses avantages qu'on va détailler dans la suite.

#### a) *Méthode SCRUM*

Le Framework Scrum est heuristique : il repose sur l'apprentissage continu et l'adaptation à des facteurs variables. Il reconnaît que l'équipe ne sait pas tout au démarrage d'un projet et évoluera avec l'expérience. La méthodologie Scrum est structurée pour aider les équipes à s'adapter naturellement à l'évolution des conditions et des exigences des utilisateurs. La redéfinition des priorités et les cycles de livraison courts sont intégrés au processus pour permettre à votre équipe d'apprendre et de s'améliorer en permanence.

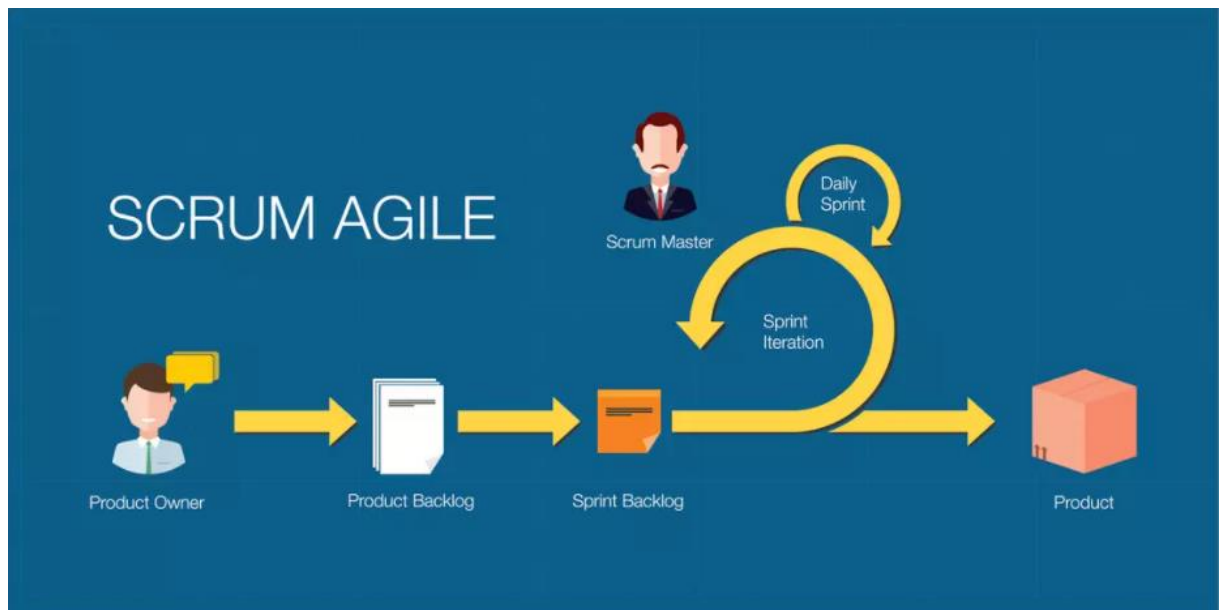


Figure 1 : Le processus de la méthodologie SCRUM

Il consiste à rédiger les besoins du client sous forme des user stories pour construire le backlog produit puis répartir ce dernier en sprints suivant des critères logiques, en affectant à chaque sprint une propriété et une durée d'un mois au maximum. Chaque sprint passe par les étapes suivantes :

- **Planification :**

Le travail à effectuer durant un Sprint est défini durant la réunion de Planification du Sprint (Sprint Planning). Ce plan est créé de manière collaborative par tous les membres de l'équipe Scrum.

La Planification du Sprint répond aux questions suivantes :

- Que peut-on livrer comme incrément résultant du Sprint à venir ?
- Comment sera effectué le travail à livrer et nécessaire pour achever l'Incrément ?

### Thème Un : Que peut-on faire de ce Sprint ?

L'équipe de développement travaille à prévoir les fonctionnalités qui seront développées pendant le Sprint. Le Product Owner discute de l'objectif que le Sprint devrait

concrétiser et des éléments du Backlog produit qui, s'ils sont complétés durant le Sprint, pourraient permettre d'atteindre l'objectif du Sprint. Toute l'équipe Scrum collabore pour comprendre le travail requis durant le Sprint.

### Thème Deux : Comment le travail choisi sera-t-il terminé ?

Une fois l'objectif du sprint fixé et les éléments du Backlog Produit choisis, l'équipe de développement planifie le travail pour transformer cette fonctionnalité en un incrément produit « Fini » durant le Sprint. Les éléments du Backlog produit sélectionnés pour ce sprint plus le plan de travail pour les livrer est appelé le Backlog Sprint (Sprint Backlog).

- **Réalisation :**

App Dans cette étape le sprint passe par les phases suivantes :

- Analyse
- Conception
- Réalisation
- Test

- **Mêlée quotidienne :**

Le Mêlée quotidienne est une réunion intermédiaire qui intervient en cours de sprint. Elle réunit les membres de l'équipe de développement. Objectif : faire le point sur les tâches réalisées la veille, et celle prévue pour la journée.

- **Revue de Sprint :**

Une revue de Sprint (Sprint Review) est tenue à la fin du Sprint pour inspecter l'incrément réalisé et adapter le Backlog Produit si nécessaire. Pendant la revue de Sprint, l'équipe Scrum et les parties prenantes échangent sur ce qui a été fait durant le Sprint.

- **La rétrospective de Sprint :**

Le but de la rétrospective de Sprint est de :

- Inspecter la manière dont le dernier Sprint s'est déroulé en ce qui concerne les personnes, les relations, les processus et les outils ;
- Identifier et ordonner les principaux éléments qui ont bien fonctionné et des améliorations potentielles ; et,
- Créer un plan pour mettre en œuvre des améliorations sur la façon dont l'équipe Scrum fait son travail.

Définissant quelques notions de SCRUM :

- **Product Backlog :**

Liste des fonctionnalités, des fonctions, des exigences, des améliorations et des correctifs qui sont nécessaires à l'évolution du produit, celui-ci est dynamique sur tout le cycle de vie du produit.

- **Sprint Backlog :**

Liste des tâches à accomplir pendant un sprint. Elles correspondent à la réalisation des éléments de carnet de produit affectés au sprint.

- **Sprint :**

Nom d'une itération dans Scrum. Cette itération dure 1 mois maximum en théorie, mais en pratique entre 2 et 4 semaines. Pendant une itération, l'équipe doit développer la liste d'éléments du carnet de produit qui a été définie au début du sprint.

- **Incrément :**

L'incrément est constitué des éléments du Backlog produit « Finis » pendant le sprint ainsi que de la valeur cumulative des incréments livrés dans les sprints précédents. À la fin d'un Sprint, le nouvel incrément doit être « Fini », ce qui implique qu'il doit être dans un état publiable et qu'il correspond à la définition de « Fini » (Définition of Done) de l'équipe de développement.

## *b) Processus de développement*



Dans ce qui suit nous allons réaliser un cycle de vie du notre projet, tout en se basant sur les principes de la méthodologie SCRUM, et en suivant ses phases déjà détaillées.

Pour commencer notre projet nous avons identifié et défini les différents besoins et exigences pour les écrire sous forme des user stories comme suivant :

**User Story 0 :** en tant qu'administrateur de la bibliothèque, je dois pouvoir ajouter un nouveau livre, afin d'augmenter l'effectif des livres du système.

**User Story 1 :** en tant qu'administrateur de la bibliothèque, je dois pouvoir savoir les livres disponibles dans la bibliothèque.

**User Story 2 :** en tant qu'administrateur de la bibliothèque, je dois pouvoir rechercher un livre par son ISBN ou par une partie ou totalité de son titre, afin de visualiser l'ensemble de ses informations.

**User Story 3 :** en tant qu'administrateur de la bibliothèque, suite à une recherche de livres, je dois pouvoir avoir la possibilité de modifier ou de supprimer chacun des livres sur la liste qui s'affiche.

**User Story 4 :** en tant qu'administrateur de la bibliothèque, je dois pouvoir ajouter un nouveau client, afin d'augmenter l'effectif des clients du système.

**User Story 5 :** en tant qu'administrateur de la bibliothèque, je dois pouvoir rechercher un client par son e-mail ou par une partie ou totalité de son nom de famille, afin de visualiser l'ensemble de ses informations.

**User Story 6 :** en tant qu'administrateur de la bibliothèque, suite à une recherche de clients, je dois pouvoir avoir la possibilité de modifier ou de supprimer chacun des clients sur la liste qui s'affiche.

**User Story 7 :** en tant qu'administrateur de la bibliothèque, je dois pouvoir ajouter une nouvelle catégorie, afin d'augmenter l'effectif des catégories du système.

**User Story 8 :** en tant qu'administrateur de la bibliothèque, je dois pouvoir rechercher une catégorie par son label, afin de visualiser l'ensemble de ses informations.

**User Story 9 :** en tant qu'administrateur de la bibliothèque, suite à une recherche de catégories, je dois pouvoir avoir la possibilité de modifier ou de supprimer chacune des catégories sur la liste qui s'affiche.

**User Story 10 :** en tant qu'administrateur de la bibliothèque, je dois pouvoir ajouter un nouveau prêt, afin de réaliser la connexion entre un livre sorti de la bibliothèque et son emprunteur (client enregistré).

**User Story 11 :** en tant qu'administrateur de la bibliothèque, je dois pouvoir rechercher la liste des prêts d'un client via son e-mail et la liste des prêts qui doivent être clôturés dans une certaine date, afin de visualiser l'historique.

**User Story 12 :** en tant qu'administrateur de la bibliothèque, suite à une recherche de prêts, je dois pouvoir avoir la possibilité de modifier ou de clôturer chacun des prêts sur la liste qui s'affiche.

**User Story 13 :** en tant qu'administrateur de la bibliothèque, suite à une recherche de prêts, je dois pouvoir avoir la possibilité d'envoyer un mail à un emprunteur, afin de lui passer une information utile.

**User Story 14 :** un client qui a rendu un livre avec un retard de 3 jours, il ne doit pas pouvoir faire un prêt dans les prochains 3 jours.

**User Story 15 :** en tant qu'administrateur de la bibliothèque, je dois pouvoir ajouter une nouvelle réservation, afin que le client puisse réserver un livre qui n'existe pas à la bibliothèque.

**User Story 16 :** en tant qu'administrateur de la bibliothèque, je dois pouvoir rechercher la liste des réservations d'un client via son e-mail et la liste des réservations liées à un livre via l'ISBN de ce dernier, afin de visualiser l'historique.

**User Story 17 :** en tant qu'administrateur de la bibliothèque, suite à une recherche d'une réservation, je dois pouvoir avoir la possibilité de modifier ou de supprimer chacune des réservations sur la liste qui s'affiche.

**User Story 18 :** en tant qu'administrateur de la bibliothèque, suite à une recherche d'une réservation, je dois pouvoir avoir la possibilité d'envoyer un mail à un client.

**User Story 19 :** le system doit se charger d'envoyer un mail au client s'il le livre qui a réservé est rendu par un autre client.

**User Story 20 :** en tant qu'administrateur de la bibliothèque, je dois pouvoir avoir sur mon application, un menu général qui me permet d'accéder à la page de gestion des livres, celle des réservations, celle des catégories, celle des clients et celle des prêts ; afin d'être plus efficace dans l'utilisation de l'application.

**User Story 21 :** en tant qu'un administrateur de la bibliothèque, je dois m'authentifier pour avoir accès aux différentes possibilités citées dans les US ci-dessus.

Puis dans la planification du sprint, nous avons sélectionné les fonctionnalités qui peuvent être réalisées dans le premier sprint, ensuite nous avons entamer la partie réalisation du sprint (analyse, conception, réalisation, test). Ainsi de suite pour les autres sprints.

Dans la partie suivante on va montrer comment nous avons réparti les différentes fonctionnalités en sprints :

**Sprint 1:** User Story 0, User Story 1, User Story 2, User Story 3

**Sprint 2:** User Story 4, User Story 5, User Story 6

**Sprint 3:** User Story 7, User Story 8, User Story 9

**Sprint 4:** User Story 10, User Story 11, User Story 12, User Story 13, User Story 14

**Sprint 5:** User Story 15, User Story 16, User Story 17, User Story 18, User Story 19

**Sprint 6:** User Story 20

**Sprint 7:** User Story 21

**c) Planification de projet**

La planification est une étape très importante qui permet d'assurer le bon déroulement du projet tout au long des phases constituant le cycle de développement, à savoir l'étude des besoins, l'analyse fonctionnelle, la conception et la modélisation, l'étude technique, le codage et les tests et enfin le déploiement. La figure suivante représente les différentes tâches que nous étions amenés à réaliser durant la période du projet :

Phase du projet	Date de début	Date de fin
analyse de besoins	01/08/2021	01/08/2021
construire le backlog produit	02/08/2021	02/08/2021
repartir le backlog produit en sprints	03/08/2021	03/08/2021
documentation angular, spring data rest	04/08/2021	10/08/2021
sprint1 gestion des livres	11/08/2021	17/08/2021
sprint2 gestion des clients	18/08/2021	25/08/2021
sprint3 gestion des categories	26/08/2021	01/09/2021
sprint4 gestion des emprunts	02/09/2021	08/09/2021
sprint5 gestion des reservations	09/09/2021	15/09/2021
sprint6 menu general	16/09/2021	18/09/2021
sprint7 securité	19/09/2021	21/09/2021

Figure 2 : Planification du projet

**D. Objectif :**

L'objectif principal de notre projet est de faciliter la tâche pour l'administrateur à travers une site web qui lui permet :

- La gestion des emprunts
- La gestion des réservations
- La gestion des catégories
- La gestion des clients
- La gestion des livres
- Le contact avec les clients

## **E. Conclusion**

Dans ce chapitre, nous avons le contexte général du projet : cadre, problématique, solution, conduite du projet, objectifs. Dans le chapitre suivant, nous présenterons en détail le cahier de charge du projet.

## **II. Cahier de charge**

## A. Introduction

Ce chapitre a pour objectif de présenter une étude détaillée des besoins et d'expliquer les différentes fonctionnalités du projet en précisant les différents intervenants.

## B. Etude de besoins

Nous allons définir ici les actions que l'administrateur peut faire à l'aide du site web mise à sa disposition.

### 1. Besoins fonctionnels :

#### a) *Les besoins fonctionnels liés aux livres :*

Notre application permettre à l'administrateur après l'authentification de :

- Visualiser, ajouter, supprimer et modifier les livres
- Rechercher un livre par son ISBN, ou par une partie ou la totalité de son titre

#### b) *Les besoins fonctionnels liés aux catégories :*

Notre application permettre à l'administrateur après l'authentification de :

- Visualiser, ajouter, supprimer et modifier les catégories
- Rechercher une catégorie par une partie ou la totalité de son label

#### c) *Les besoins fonctionnels liés aux clients :*

Notre application permettre à l'administrateur après l'authentification de :

- Visualiser, ajouter, supprimer et modifier les clients
- Rechercher un client par son email, ou par une partie ou la totalité de son nom

#### d) *Les besoins fonctionnels liés aux emprunts :*

Notre application permettre à l'administrateur après l'authentification de :

- Visualiser, ajouter, supprimer et modifier les emprunts
- Rechercher un emprunt par l'email du client, ou la date de fin d'emprunt.

- Empêcher un client qui s'est tardé de rendre un livre, de faire un prêt pour la même durée de son retard.

e) **Les besoins fonctionnels liés aux réservations :**

Notre application permettre à l'administrateur après l'authentification de :

- Visualiser, ajouter, supprimer et modifier les réservations
- Rechercher une réservation par l'email du client, ou l'ISBN du livre.
- Informer un client que livre qui a réservé est disponible

## 2. Besoins fonctionnels :

D'abord nous allons déterminer l'ensemble de contraintes d'implémentation à respecter pour garantir le bon fonctionnement de l'application :

**La sécurité :** Vu que cette application contient des données personnelles des clients ainsi que des informations des livres de la bibliothèque, qui doivent être protégées.

**La performance :** Etant donné le nombre d'utilisateur simultané, l'application doit être en mesure de satisfaire tous les utilisateurs. On distingue pour ça 3 axes de travail :

- La rapidité : l'application doit être conçue pour avoir un temps de réponse minimum.
- La fiabilité : l'application doit être assurer le bon fonctionnement sans erreur.
- L'ergonomie : l'application doit être simple à maintenir et à comprendre.

## C. Conclusion

Après que nous avons pris conscience de tous les besoins fonctionnels et non fonctionnels, dans le prochain chapitre on va élaborer la conception du projet.

# **III. Conception du projet**



### A. Introduction :

Dans le présent chapitre, nous allons entamer une partie importante du développement de l'application. Nous allons présenter dans un premier temps le diagramme de cas d'utilisations qui donne une vue fonctionnelle de notre application, puis le diagramme de classes.

### B. Diagramme de cas d'utilisation :

Avec Unified Modeling Language (UML), on modélise les besoins de chaque acteur dans notre application au moyen de diagramme de cas d'utilisation.

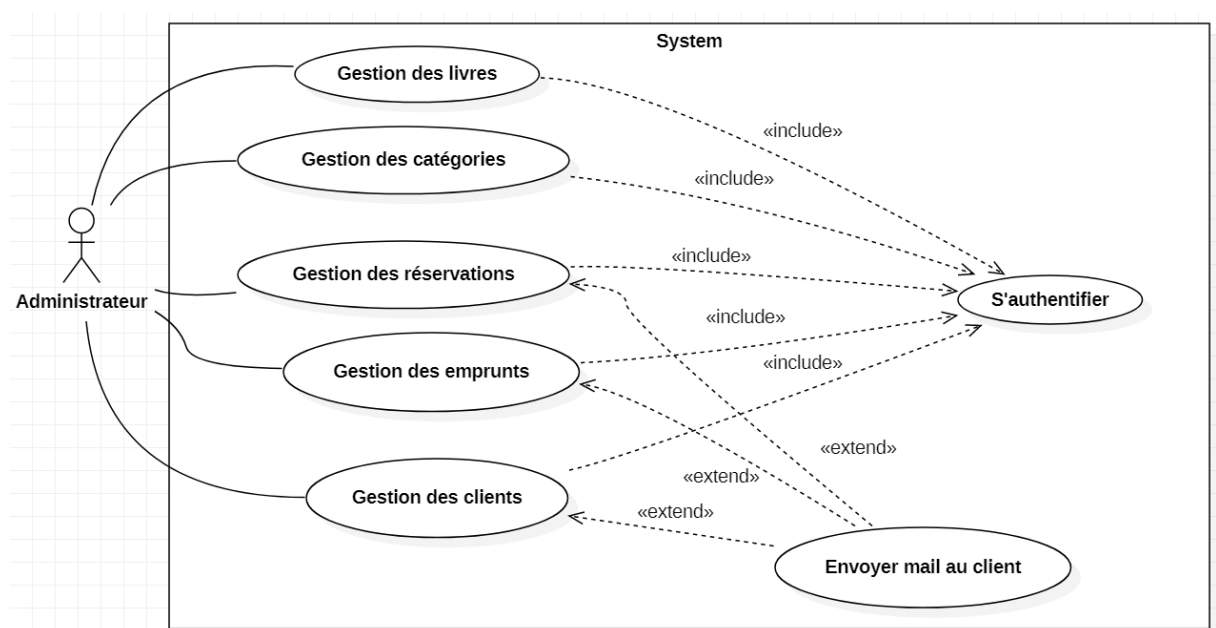


Figure 3: Diagramme cas d'utilisation

### C. Diagramme de classes :

Le diagramme de classes comporte toutes les classes qui vont constituer notre base de données par la suite :

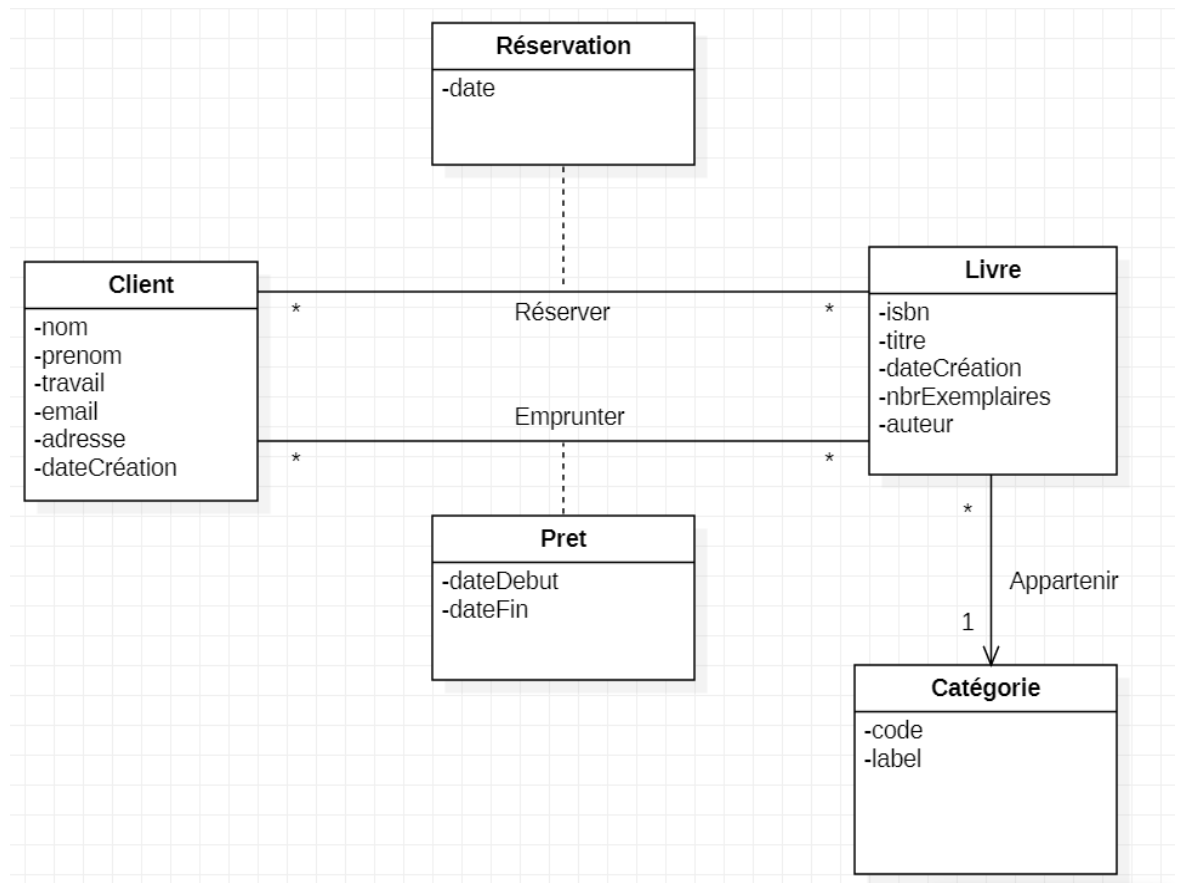


Figure 4 : Diagramme de classes

### D. Conclusion :

Dans ce chapitre, nous avons présenté le diagramme de cas d'utilisation et le diagramme de classes. Le chapitre suivant est consacré aux études techniques et technologiques.

# **IV. Etudes techniques et technologiques**

## A. Introduction

Dans ce chapitre nous avons présenté l'architecture technique, l'environnement dans lequel nous allons travailler, les outils et les choix techniques utilisés pour implémenter notre projet ainsi que les langages de programmation adoptés.

## B. Architecture technique

On présente dans cette partie l'architecture technique adoptée pour notre projet.

La figure ci-dessous est l'architecture Technique que nous avons conçu pour l'application web.

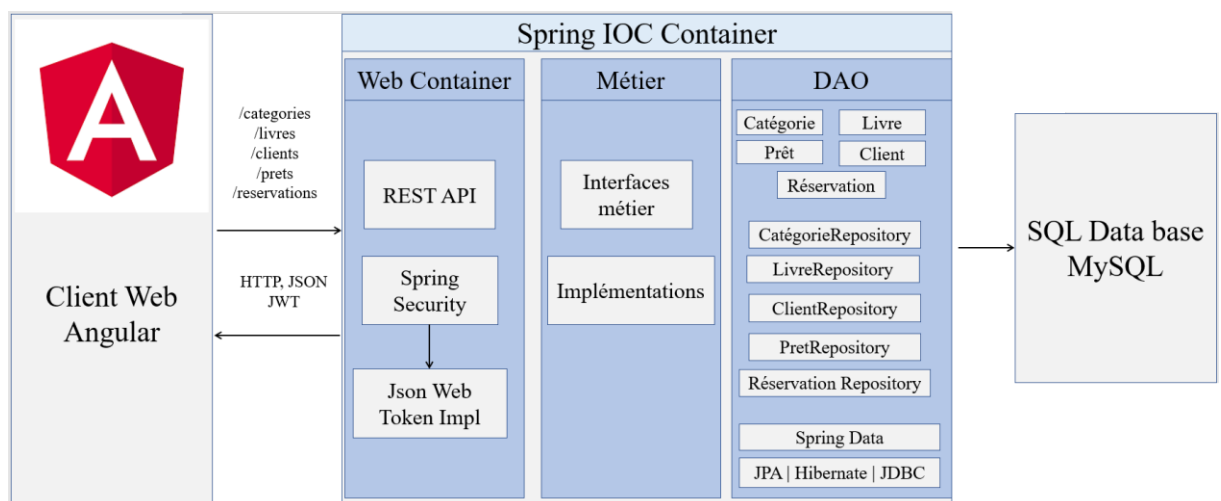


Figure 5 : L'architecture technique d'application

- Inversion of control (IOC) Container : C'est la base la plus importante et également la base, le fondement du Spring. C'est le rôle de la configuration et de la gestion de dépendances et la gestion du cycle de vie des objets Java

Il fait La configuration automatique pour lier les implémentations à leurs interfaces (injection de dépendances)

- WEB : il contient un serveur web Tomcat et l'ensemble des rest contrôleurs qui permet de traiter les requêtes.
- Métier : il contient les interfaces métiers et leurs implémentations.
- DAO (Data Access Object) : Contient toutes les classes Repository permettant l'accès à la base de données.
- Client Web Angular : c'est le client web qui communique avec API (Application Programming Interface) Rest (Representational State Transfer), il se charge d'implémenter les interfaces web en utilisant HTML (HyperText Markup Language), CSS (Cascading Style Sheets), Bootstrap, PrimeNG.

## C. Technologies et bibliothèques utilisés

### 1. Spring Boot



Spring Boot est un framework qui permet la mise en place d'application Spring rapidement et facilement. Il se base sur le Framework Spring et permet de s'affranchir de la plupart des configurations de celui-ci à mettre en place pour créer une application. Les principales fonctions :

- Une gestion des dépendances Spring simplifiée
- -Un déploiement facilité

- Intégrez directement Tomcat, Jetty ou Undertow (inutile de déployer des fichiers WAR (Web Application Resource))
- La configuration automatique de bibliothèques Spring et autres
- La configuration des propriétés externes plus lisible
- Facilités pour créer des repositories - Des possibilités de déclarer des sorties JSON (JavaScript Objet Notation) multiples
- L'exposition des ressources par REST juste avec une annotation
- Aucune génération de code et aucune exigence pour la configuration XML (Extensible Markup Language)

## 2. Spring Security

Spring Security, est un contrôleur d'authentification flexible et puissant pour assurer une application Web Java basé sur Spring.

## 3. JSON Web Token



JSON Web Token (JWT) est un standard ouvert défini dans la RFC 75191. Il permet l'échange sécurisé de jetons (tokens) entre plusieurs parties. Cette sécurité de l'échange se traduit par la vérification de l'intégrité et de l'authenticité des données. Elle s'effectue par l'algorithme HMAC ou RSA.

Un jeton se compose de trois parties :

- Un en-tête (header), utilisé pour décrire le jeton. Il s'agit d'un objet JSON.
- Une charge utile (payload) qui représente les informations embarquées dans le jeton. Il s'agit également d'un objet JSON.
- Une signature numérique.

Il existe des outils en ligne permettant de les déchiffrer.

#### **4. Spring Mail**

Spring Email est une librairie fournie par le framework Spring, elle comprend certaines interfaces et classes pour la gestion et l'envoi des emails.

#### **5. Spring Data JPA**

Spring Data JPA facilite l'implémentation de repositories basés sur JPA (Java Persistence API), il traite aussi les couches d'accès aux données basées sur JPA. En outre il facilite la création d'applications Spring qui utilisent des technologies d'accès aux données.

#### **6. Spring Data Rest**

Données Spring Data Rest fait du projet générique Spring Data et facilite la création de « hypermedia-driven REST web services » au-dessus des repositories Spring Data.

#### **7. Hibernate**



Hibernate Framework Est un Framework open source gérant la persistance des objets en base de données relationnelle. Hibernate est adaptable en termes d'architecture, il peut donc être utilisé aussi bien dans un développement client lourd, que dans un environnement web léger de type Apache Tomcat ou dans un environnement Java EE (Java Entreprise Edition) complet : WebSphere, JBoss Application Server et Oracle WebLogic Server.

## 8. Maven



Maven est un outil permettant d'automatiser la gestion de projets Java. Il offre entre autres les fonctionnalités suivantes :

- Compilation et déploiement des applications Java (JAR, WAR)
- Gestion des librairies requises par l'application
- Exécution des tests unitaires
- Génération des documentations du projet (site web, PDF, Latex)



- Intégration dans différents IDE (Eclipse, JBuilder)

## 9. Bootstrap



Un Framework développé par l'équipe du réseau social Twitter, utilisant les langages HTML, CSS et JavaScript fournit aux développeurs des outils pour créer un site facilement. Il sert à développer des sites avec un design responsive, qui s'adapte à tout type d'écran. Il fournit des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de navigation et bien d'autres encore.

## 10. Angular



Il permet la création d'applications Web et plus particulièrement de ce qu'on appelle des « Single Page Applications » : des applications web accessibles via une page web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action.

Le Framework est basé sur une architecture du type MVC (Model View Controller) et permet donc de séparer les données, le visuel et les actions pour une meilleure gestion des responsabilités.

## 11. PrimeNG



PrimeNG est une collection de composants d'interface utilisateur riches pour Angular. Il est créé par PrimeTek Informatics.

## D. Langages de programmation

### 1. JAVA



Java est un langage de programmation inspiré du langage C++, avec un modèle de programmation orienté objet.

Java permet de créer des applications complètes. Il peut également servir à créer un petit module d'application, dit applet, à intégrer dans une page Web.

Les principales caractéristiques de Java sont les suivantes :

- Les programmes créés sont portables. Le programme source est compilé dans un « code », qui peut être exécuté sur un serveur ou un client doté d'une machine virtuelle Java. Cette dernière traduit le code compilé en code exécutable sur le matériel informatique. Cela signifie que les différences entre les plateformes, comme la longueur des instructions, peuvent être reconnues et gérées en local au fil de l'exécution du programme. Il n'est donc plus nécessaire de créer des versions différentes du programme pour chaque plateforme.
- Le code est robuste. Cela qui signifie que les objets Java ne peuvent contenir aucune référence à des données qui leur sont externes à ou à d'autres objets connus. Ce mécanisme garantit qu'une instruction ne contiendra pas l'adresse de données stockées dans une autre application ou dans le système d'exploitation lui-même, ce qui provoquerait l'arrêt ou le « plantage » du programme, voire du système d'exploitation. La machine virtuelle Java procède à diverses vérifications sur chaque objet pour en assurer l'intégrité.
- Java est orienté objet, ce qui implique, entre autres caractéristiques, qu'un objet tire parti de son appartenance à une classe d'objets pour hériter du code commun à cette classe. Les objets sont considérés comme des « noms » auxquels un utilisateur peut se rapporter, plutôt qu'à des « verbes » traditionnellement utilisés dans les procédures. Ainsi, une méthode peut être considérée comme l'une des fonctionnalités ou l'un des comportements de l'objet.

La machine virtuelle Java comprend un compilateur JIT (Just-In-Time), ou compilateur à la volée qui compile dynamiquement le code source en code exécutable au lieu de l'interpréter instruction par instruction. (5)

## 2. JAVA EE



Jakarta EE (ou Java EE), est une spécification pour la plate-forme Java d'Oracle, destinée aux applications d'entreprise.

22La plate-forme étend Java Platform, Standard Edition (Java SE) en fournissant une API de mapping objet-relationnel, des architectures distribuées et multitiers, et des services web. La plate-forme se fonde principalement sur des composants modulaires exécutés sur un serveur d'applications.

Pour ce faire, Java EE définit les éléments suivants :

- Une plate-forme (Java EE Platform), pour héberger et exécuter les applications, incluant outre Java SE des bibliothèques logicielles additionnelles du Java Development Kit (JDK).
- Une suite de tests (Java EE Compatibility Test Suite) pour vérifier la compatibilité.
- Une réalisation de référence (Java EE Reference Implementation), dénommée GlassFish.
- Un catalogue de bonnes pratiques (Java EE BluePrints).
- Un code script.

À chaque version de Java EE correspond notamment, comme toutes les éditions Java :

- Les Java Specification Requests (JSR), constituant les spécifications de la version considérée ;
- Un Java Development Kit (JDK), contenant les bibliothèques logicielles ;
- Un Java Runtime Environment (JRE), contenant le seul environnement d'exécution (compris de base dans le JDK).

### 3. SQL



Le langage SQL (**Structured Query Language**, en français **langage de requête structurée**) est un langage informatique normalisé servant à exploiter des bases de données relationnelles.

En utilisant dans notre projet l'interface PHPMYADMIN pour gérer notre base de données MySQL.

### 4. Typescript



TypeScript est un langage de programmation libre et open source développé par Microsoft qui a pour but d'améliorer et de sécuriser la production de code JavaScript. Il s'agit d'un sur-ensemble syntaxique strict de JavaScript (c'est-à-dire que tout code JavaScript correct peut être utilisé avec

TypeScript). Le code TypeScript est transcompilé en JavaScript, et peut ainsi être interprété par n'importe quel navigateur web ou moteur JavaScript. TypeScript a été cocréé par Anders Hejlsberg, principal inventeur de C#.

TypeScript permet un typage statique optionnel des variables et des fonctions, la création de classes et d'interfaces, l'import de modules, tout en conservant l'approche non-contrainante de JavaScript. Il supporte la spécification ECMAScript 6.

## E. Environnement Logiciels

### 1. PHPMYADMIN



PhpMyAdmin (PMA) est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée principalement en PHP et distribuée sous licence GNU GPL.

Il s'agit de l'une des plus célèbres interfaces pour gérer une base de données MySQL sur un serveur PHP. De nombreux hébergeurs, gratuits comme payants, le proposent ce qui évite à l'utilisateur d'avoir à l'installer.

Cette interface pratique permet d'exécuter, très facilement et sans grandes connaissances en bases de données, des requêtes comme les créations de table de données, insertions, mises à jour, suppressions et modifications de structure de la base de données, ainsi que l'attribution et la révocation de droits et l'import/export. Ce système permet de sauvegarder commodément une base de données sous forme de fichier .SQL est d'y transférer ses données, même sans connaître SQL. (11)

## 2. WAMPSEVER



Signifie « Windows, Apache, MySQL et PHP ». WAMP est une variante de LAMP pour les systèmes Windows et est souvent installé sous forme de bundle logiciel (Apache, MySQL et PHP). Il est souvent utilisé pour le développement Web et les tests internes, mais peut également être utilisé pour servir des sites Web en direct.

La partie la plus importante du package WAMP est Apache (ou "Apache HTTP Server") qui est utilisé pour exécuter le serveur Web sous Windows. En exécutant un serveur Web Apache local sur une machine Windows, un développeur Web peut tester des pages Web dans un navigateur Web sans les publier en direct sur Internet.

WAMP comprend également MySQL et PHP , qui sont deux des technologies les plus couramment utilisées pour créer des sites Web dynamiques . MySQL est une base de données à haut débit, tandis que PHP est un langage de script qui peut être utilisé pour accéder aux données de la base de données. En installant ces deux composants localement, un développeur peut créer et tester un site Web dynamique avant de le publier sur un serveur Web public.

Alors qu'Apache, MySQL et PHP sont des composants open source qui peuvent être installés individuellement, ils sont généralement installés ensemble. Un package populaire est appelé "WampServer", qui fournit un moyen convivial d'installer et de configurer les composants "AMP" sous Windows. (12)

### 3. STARUML



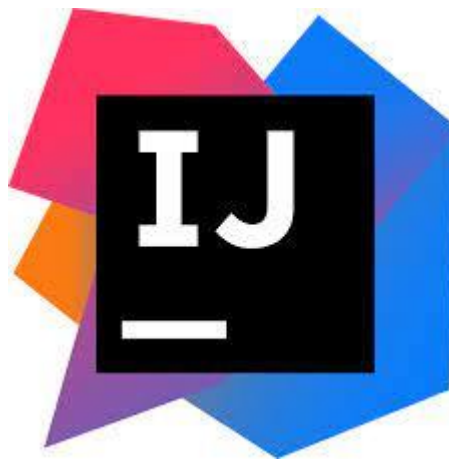
StarUML est un logiciel de modélisation UML, cédé comme open source par son éditeur, à la fin de son exploitation commerciale, sous une licence modifiée de GNU GPL.

L'objectif de la reprise de ce projet était de se substituer à des solutions commerciales comme IBM Rational Rose ou Borland Together.

StarUML gère la plupart des diagrammes spécifiés dans la norme UML 2.0.

StarUML est écrit en Delphi, et dépend de composants Delphi propriétaires (non open-source), ce qui explique peut-être pourquoi il n'est plus mis à jour. (15)

### 4. IntelliJ IDEA



IntelliJ IDEA également appelé « IntelliJ », « IDEA » ou « IDJ » est un environnement de développement intégré (en anglais Integrated Development Environment - IDE) destiné au développement de logiciels informatiques reposant sur la technologie Java.



Il est développé par JetBrains (anciennement « IntelliJ ») et disponible en deux versions, l'une communautaire, open source, sous licence Apache 2 et l'autre propriétaire, protégée par une licence commerciale. Tous deux supportent les langages de programmation Java, Kotlin, Groovy et Scala.

## **F. Conclusion**

Dans ce chapitre, nous avons présenté l'architecture technique et les choix techniques adoptés pour développer cette solution. Le chapitre suivant est consacré à la phase de réalisation.

# **V. Réalisation du projet**

## A. Introduction

Après avoir entamé les phases précédentes qui reposaient sur la spécification, l'analyse, et la conception, nous abordons dans ce chapitre la réalisation. Nous exposerons les interfaces homme-machine du système réalisé.

## B. Interfaces Homme-Machine

Dès que l'administrateur visite le site, il rencontre une interface qui donne une petite description sur le site web.

La figure ci-dessous représente l'interface d'accueil :

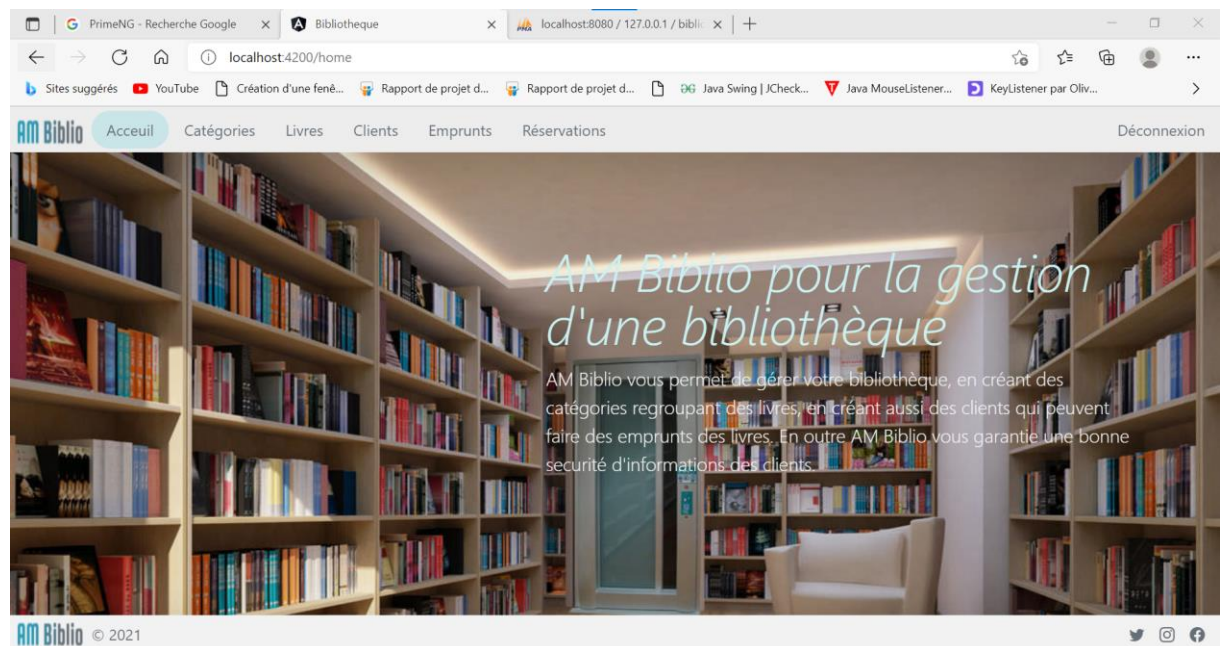


Figure 6 : Interface d'accueil

L'administrateur peut se connecter à travers une fenêtre où, il met son nom d'utilisateur et mot de passe.

La figure ci-dessous représente l'interface de connexion :

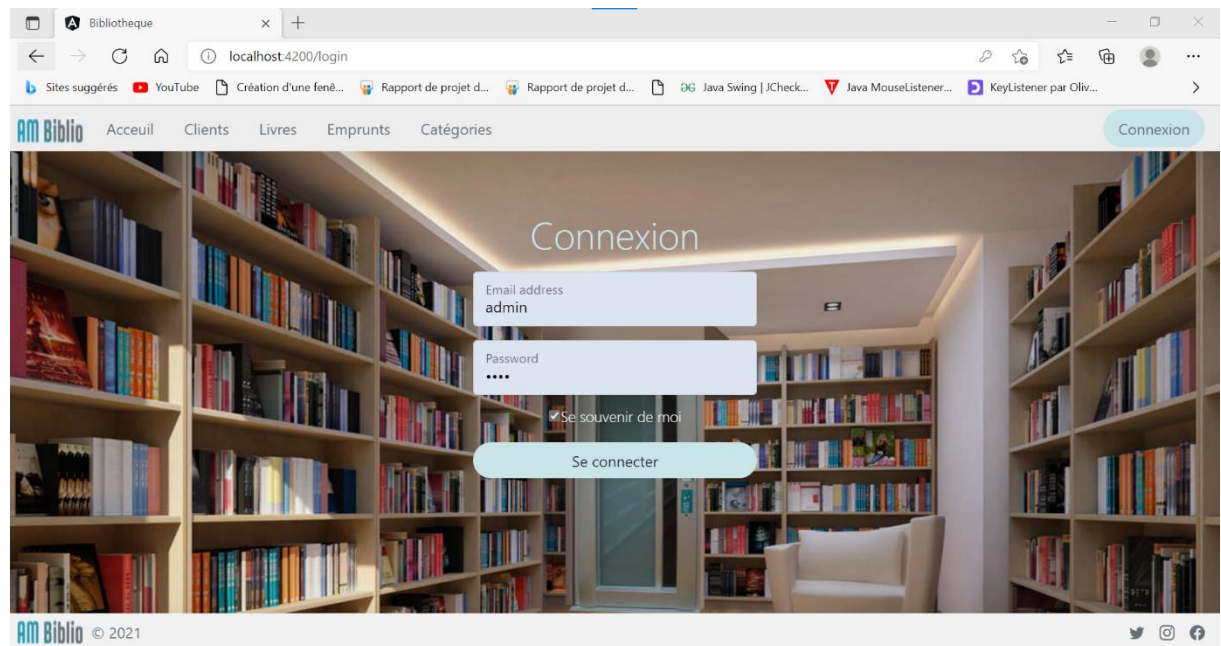


Figure 7 : Interface de connexion

La figure ci-dessous représente la liste des clients où l'administrateur peut supprimer, modifier, ajouter ou rechercher un client par son email ou par la partie ou la totalité de son nom :

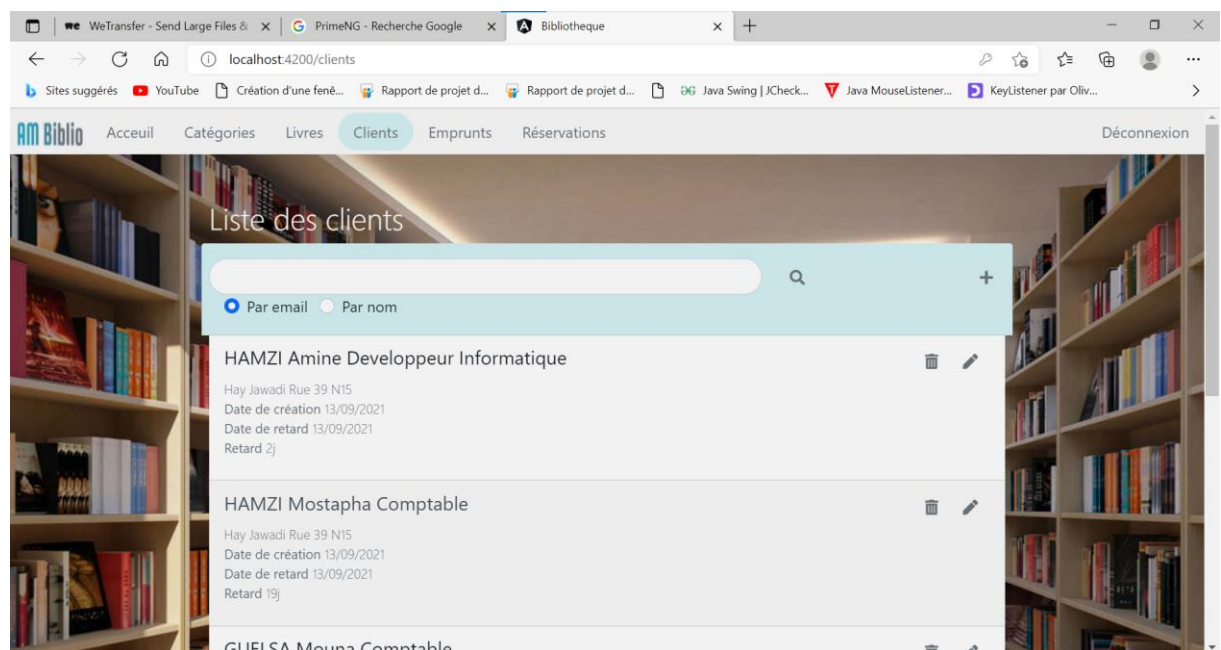


Figure 8 : Interface de gestion des clients

La figure ci-dessous représente l'interface pour modifier les informations d'un client :

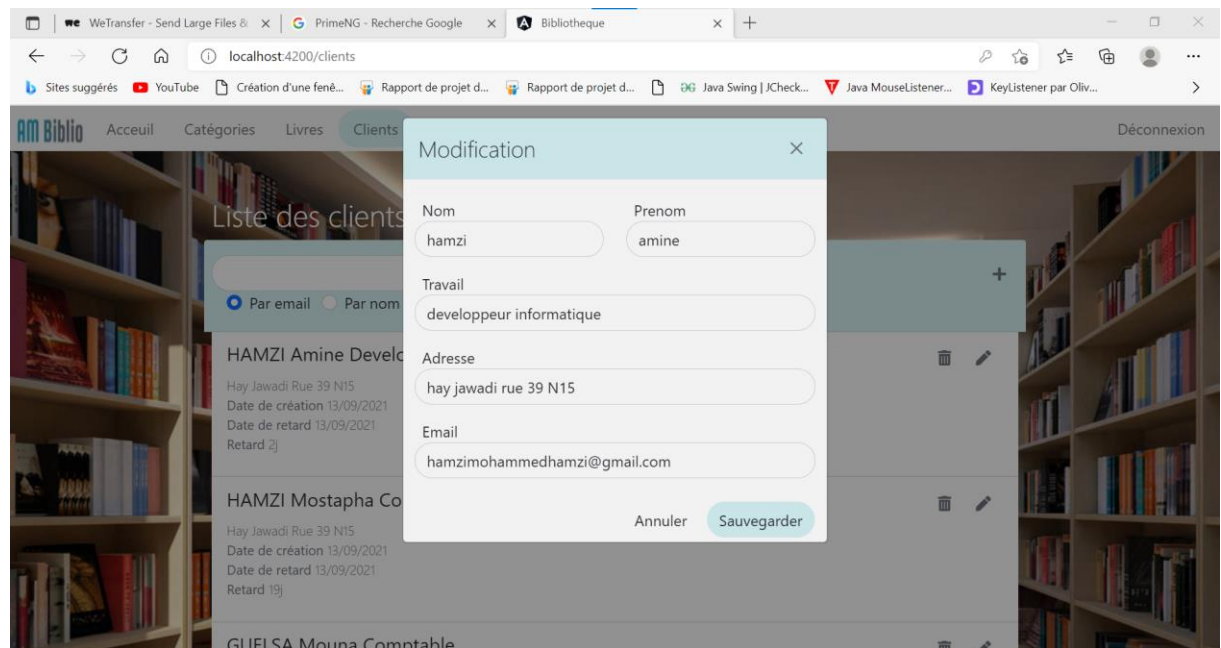


Figure 9 : Interface de modification des informations d'un client

La figure ci-dessous représente l'interface pour ajouter un nouveau client :

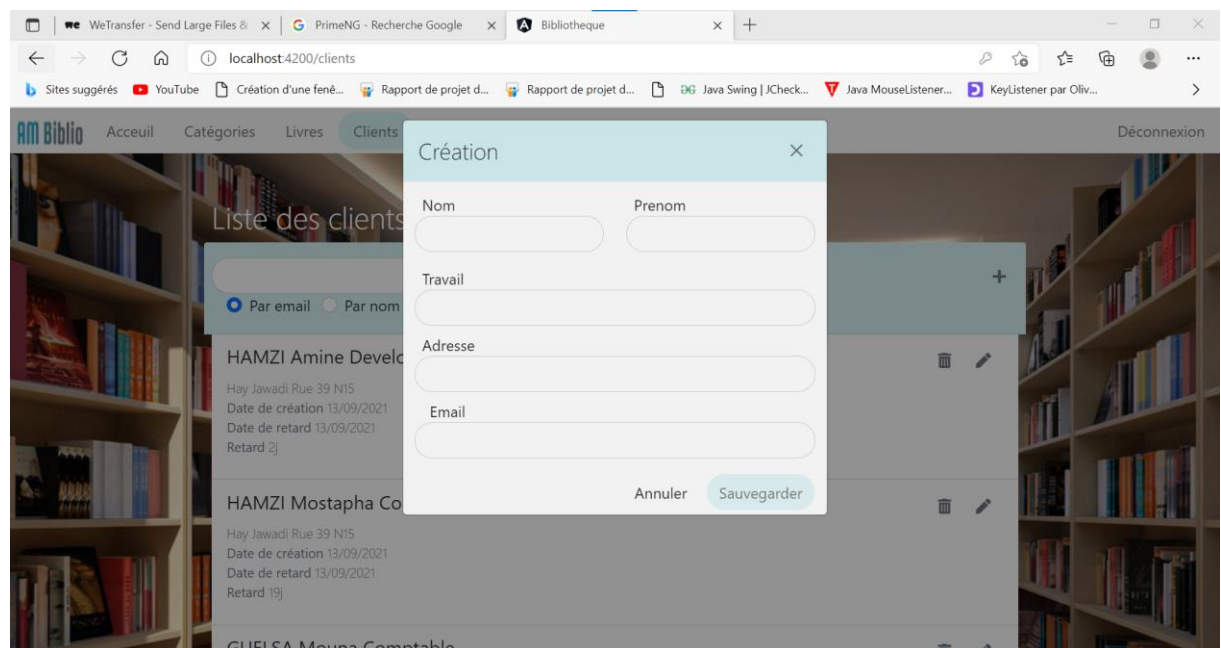


Figure 10 : Interface de création des informations d'un client

La figure ci-dessous représente la liste des catégories où l'administrateur peut supprimer, modifier, ajouter ou rechercher une catégorie par son label :



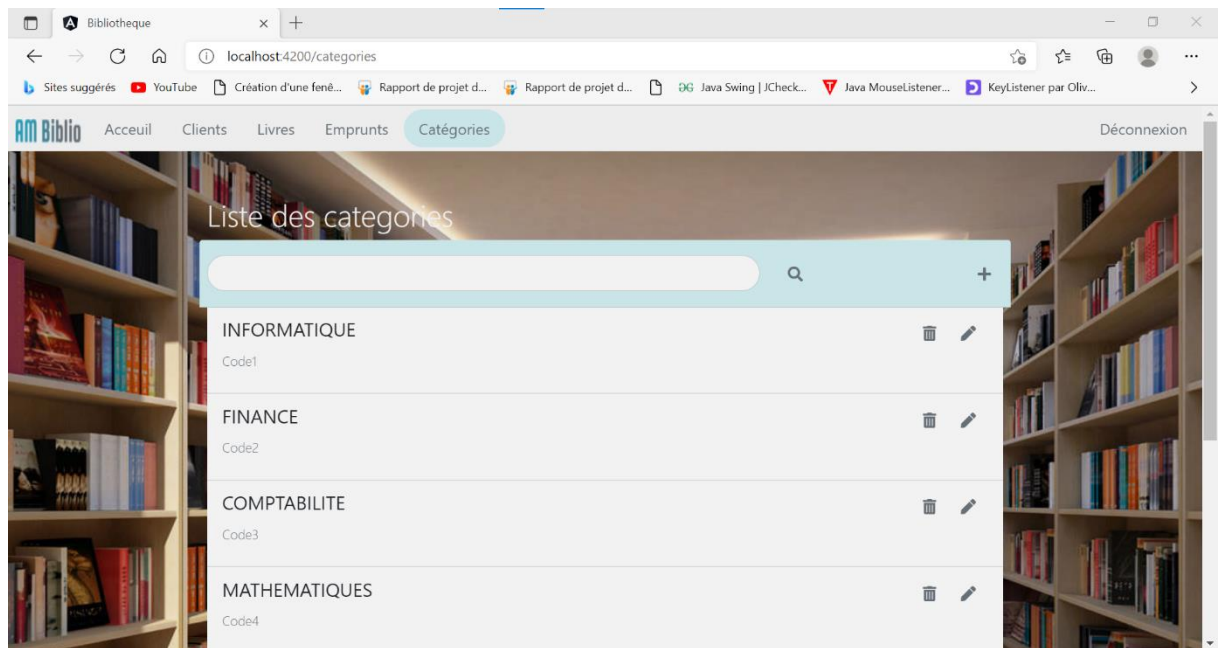


Figure 11 : Interface de gestion des catégories

La figure ci-dessous représente l'interface pour modifier les informations d'une catégorie :

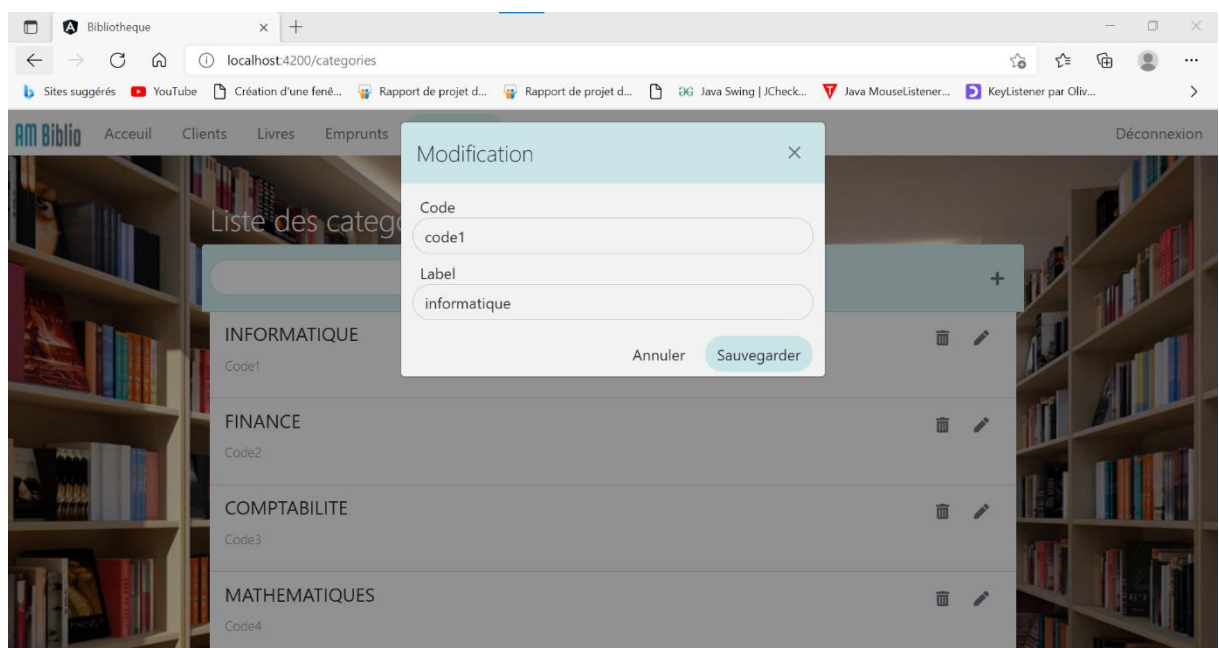


Figure 12 : Interface de modification d'une catégorie

La figure ci-dessous représente l'interface pour ajouter une nouvelle catégorie :

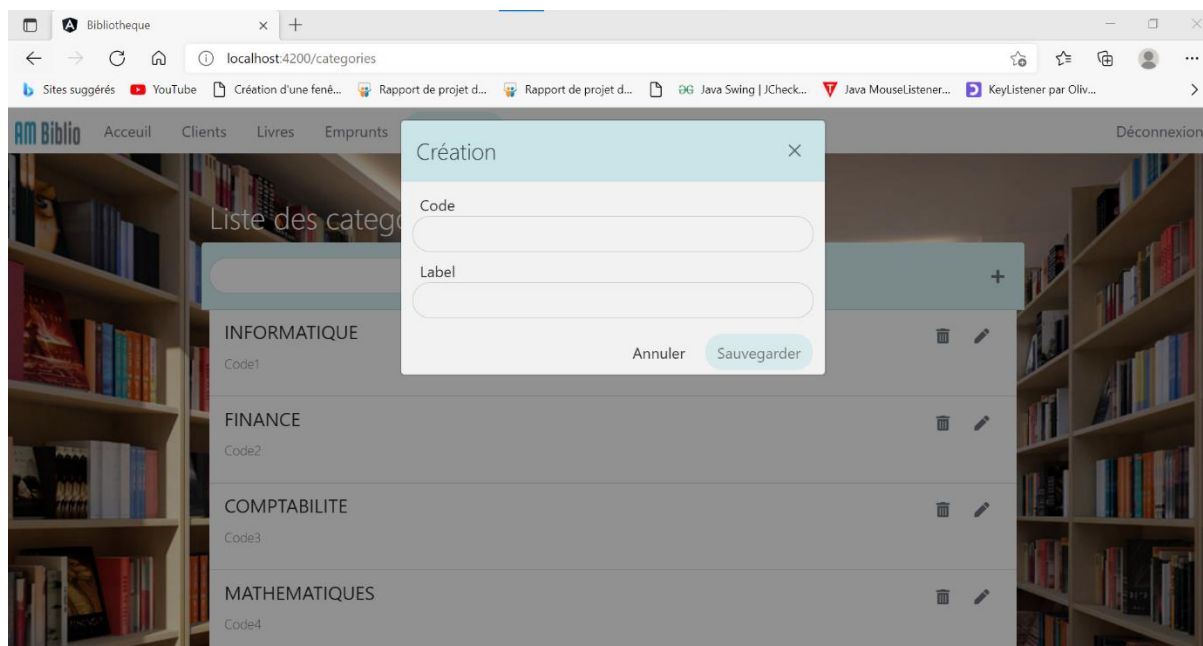


Figure 13 : Interface de création d'une catégorie

La figure ci-dessous représente la liste des livres, dont les livres marqués en rouge représente les livres qui n'existe pas dans la bibliothèque actuellement (nombre d'exemplaires est égale à 0). L'administrateur peut supprimer, modifier, ajouter ou rechercher un livre par son ISBN ou par la partie ou la totalité de son titre :

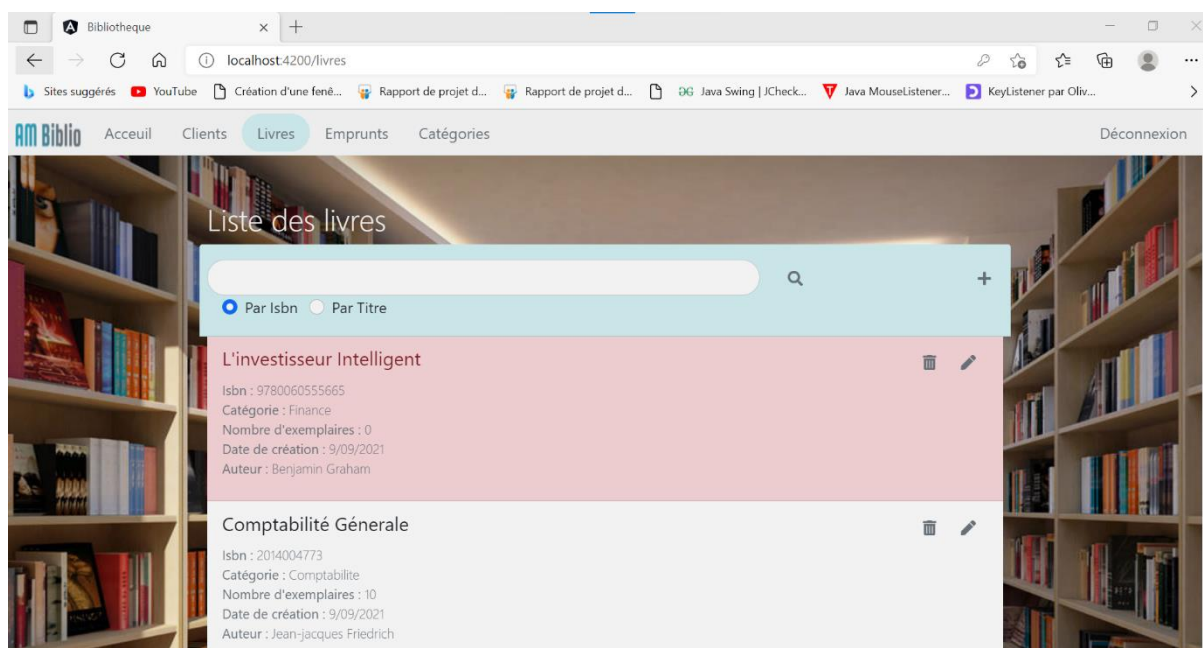


Figure 14 : Interface de gestion des livres

La figure ci-dessous représente l'interface pour modifier les informations d'un livre :

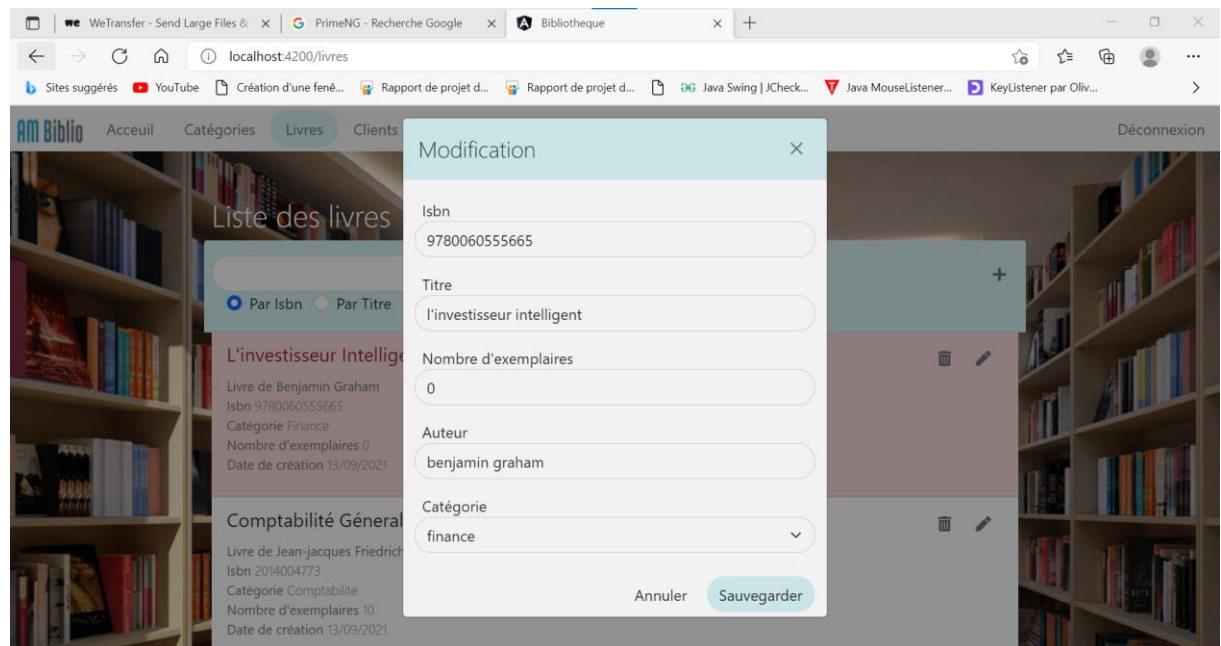


Figure 15 : Interface de modification d'un livre

La figure ci-dessous représente l'interface pour ajouter un nouveau livre :

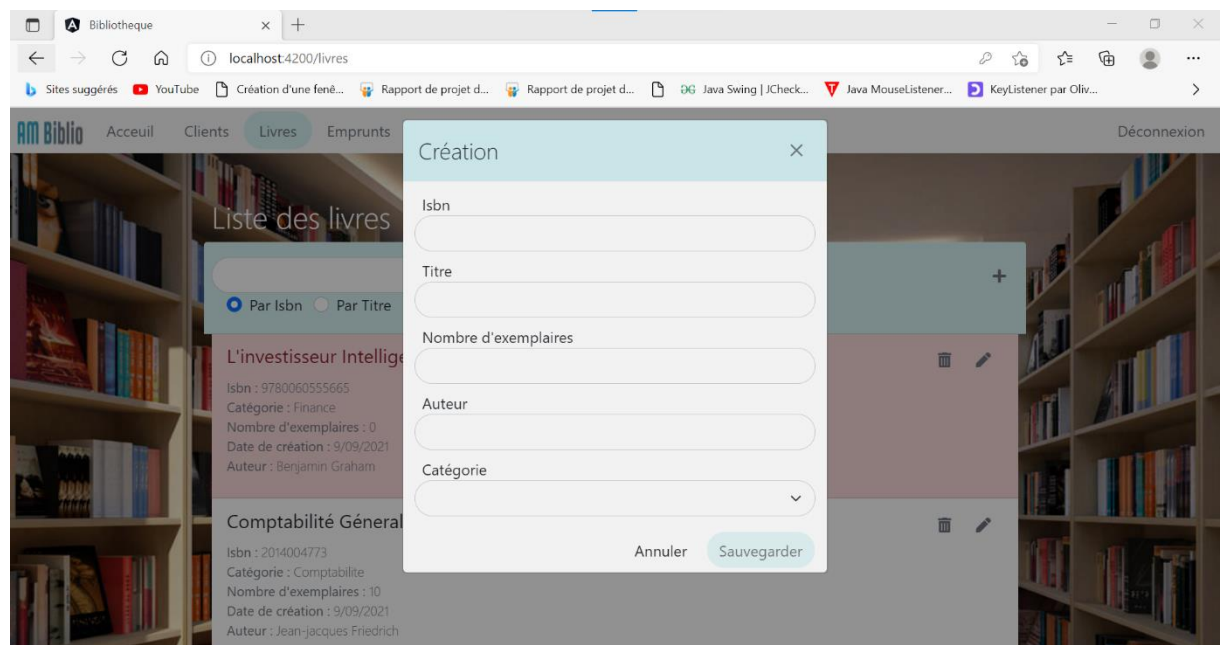


Figure 16 : interface de création d'un livre



Le site web permet à l'administrateur de clôturer, modifier, ajouter ou rechercher un emprunt par l'email d'emprunteur ou par la date de fin d'emprunt, ainsi de passer une information utile au client par Gmail.

La figure ci-dessous représente la liste des emprunts :

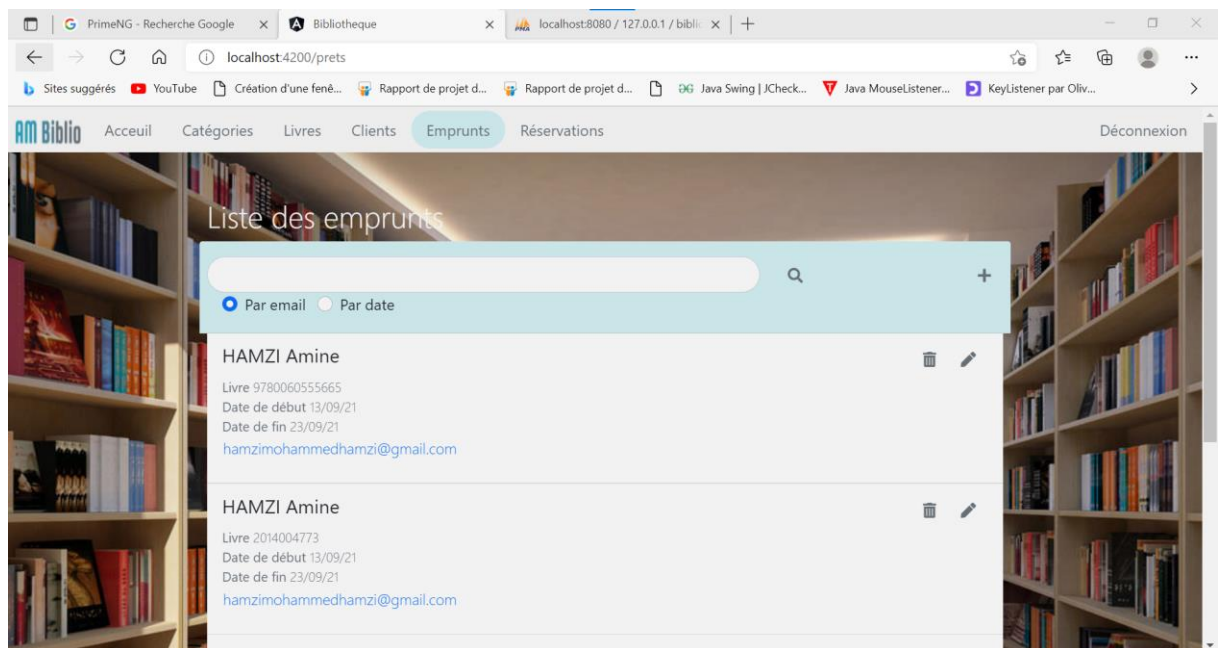


Figure 17 : Interface de gestion d'un emprunt

Lors de la clôture d'un emprunt le système se bloque pour faire une recherche dans les réservations afin de voir si le livre qui vient d'être rendu est réservé par quelqu'un, si était le cas le système envoie un email à tous les clients qui ont réservé ce livre. Sinon l'emprunt est clôturé.

La figure ci-dessous représente le blocage de la page lors de la clôture d'un emprunt :

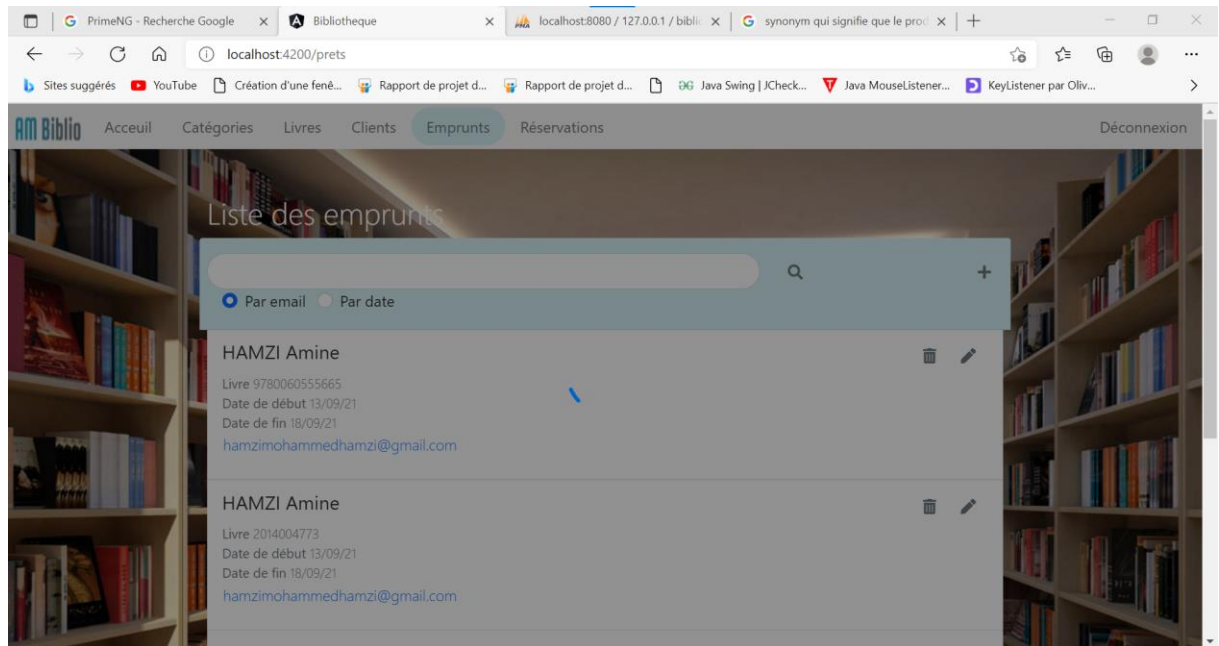


Figure 18 : Interface de chargement pour modifier un emprunt

La figure ci-dessous représente un message d'info au haut de la page pour indiquer à l'administrateur que le système a informé tous les clients qui ont réservé le livre qui vient d'être rendu :

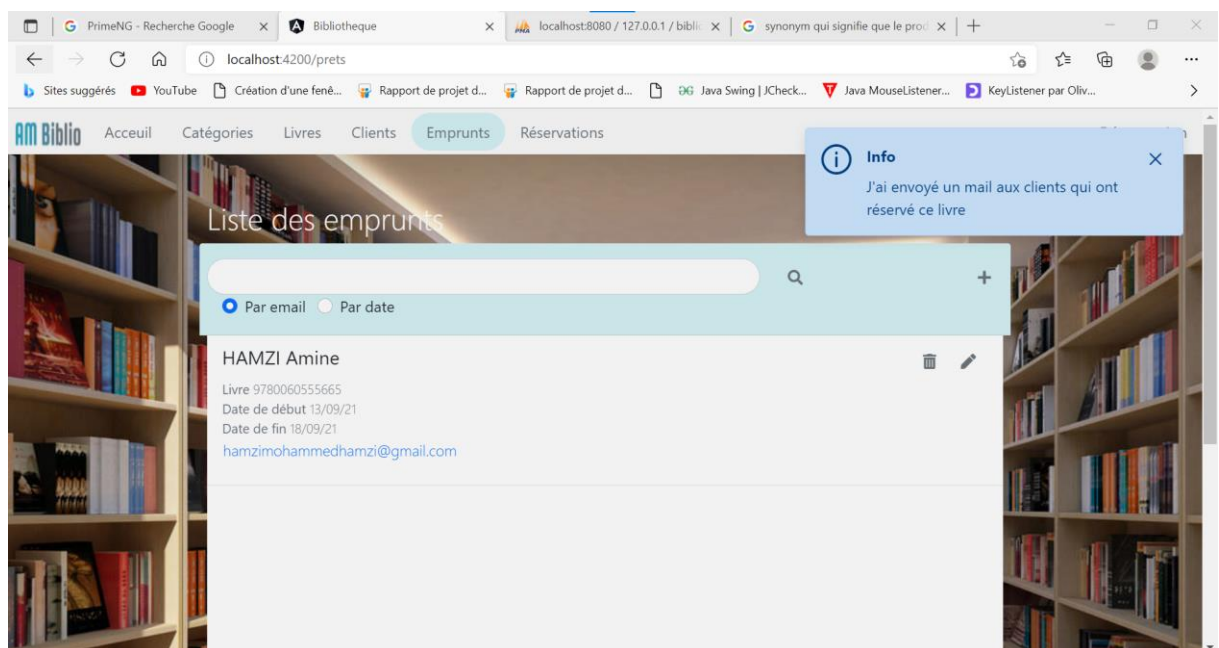


Figure 19 : Interface de gestion des emprunts

La figure ci-dessous représente l'interface pour modifier les informations d'un emprunt :

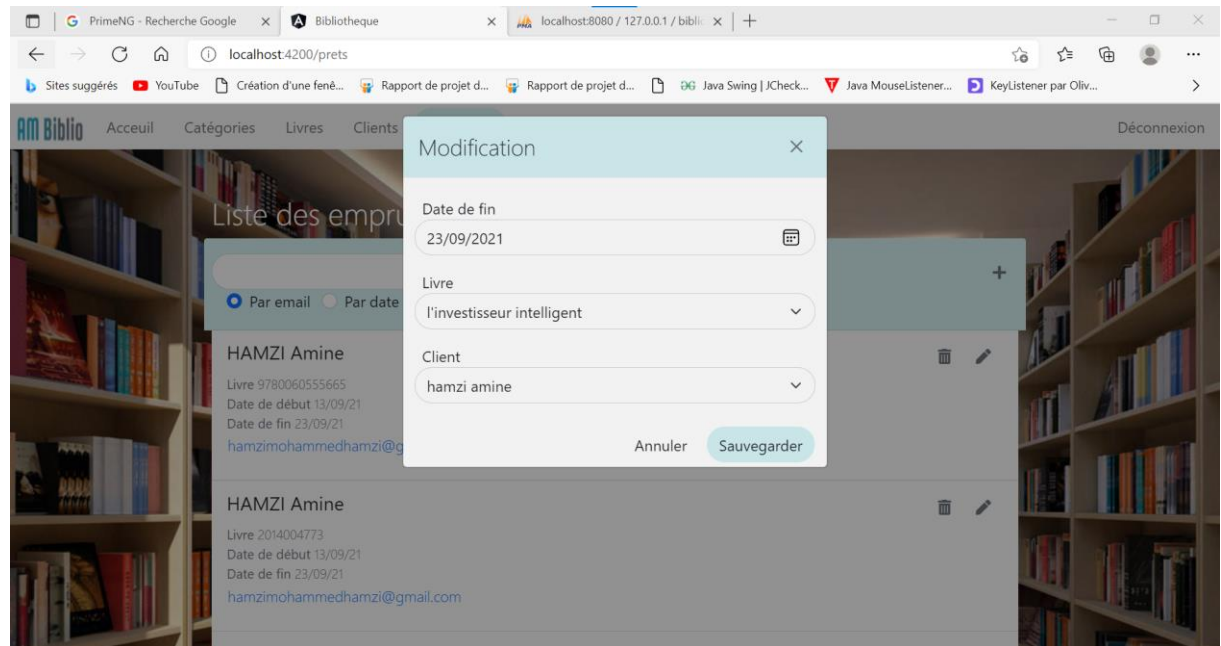


Figure 20 : Interface de modification d'un emprunt

Lors d'ajout d'un nouvel emprunt le système vérifie si le client qui veut emprunter le livre a un retard, comme il est montré dans la figure ci-dessous par message marqué en rouge au haut de la page :

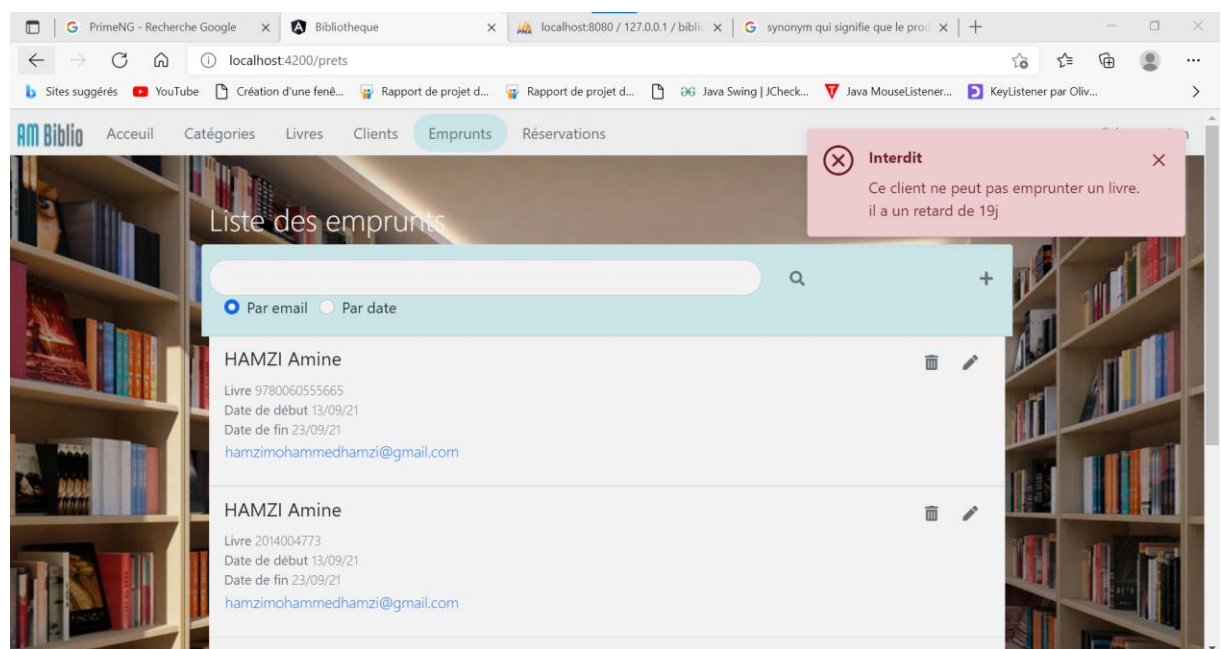


Figure 21 : Interface de gestion des emprunts

La figure ci-dessous représente l'interface pour ajouter un nouvel emprunt :

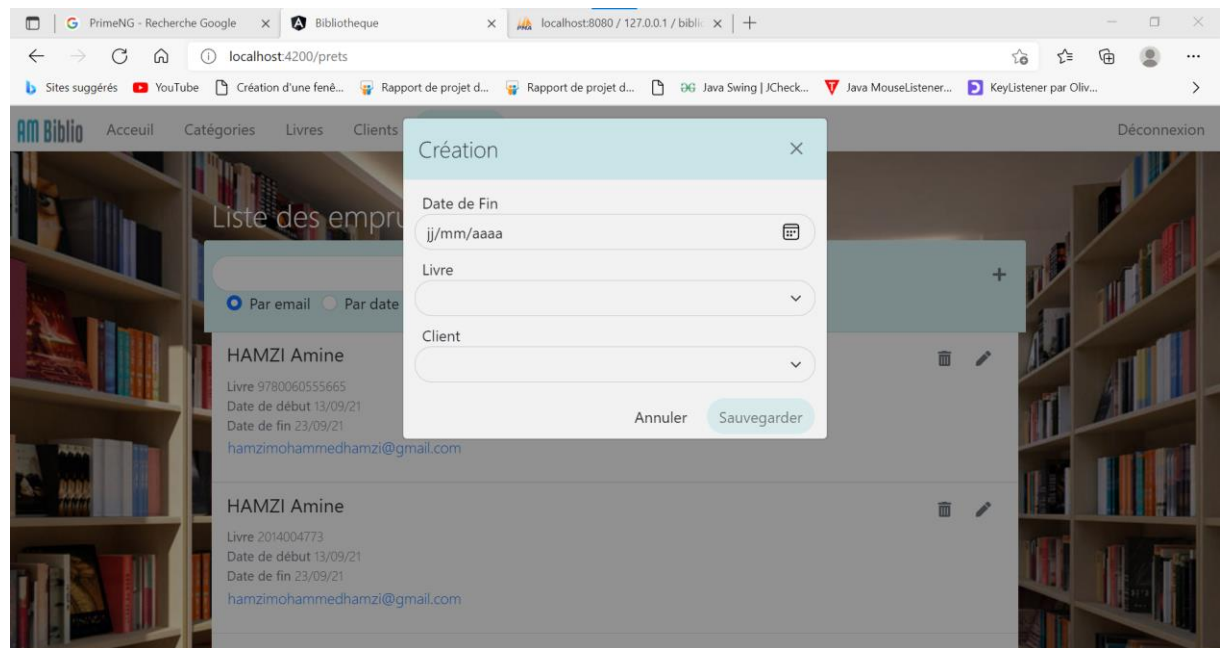


Figure 22 : Interface de création d'un emprunt

La figure ci-dessous représente la liste des réservations où l'administrateur peut supprimer, modifier, ajouter ou rechercher une réservation par l'email du client ou par l'ISBN du livre, ainsi de passer une information utile au client par Gmail :

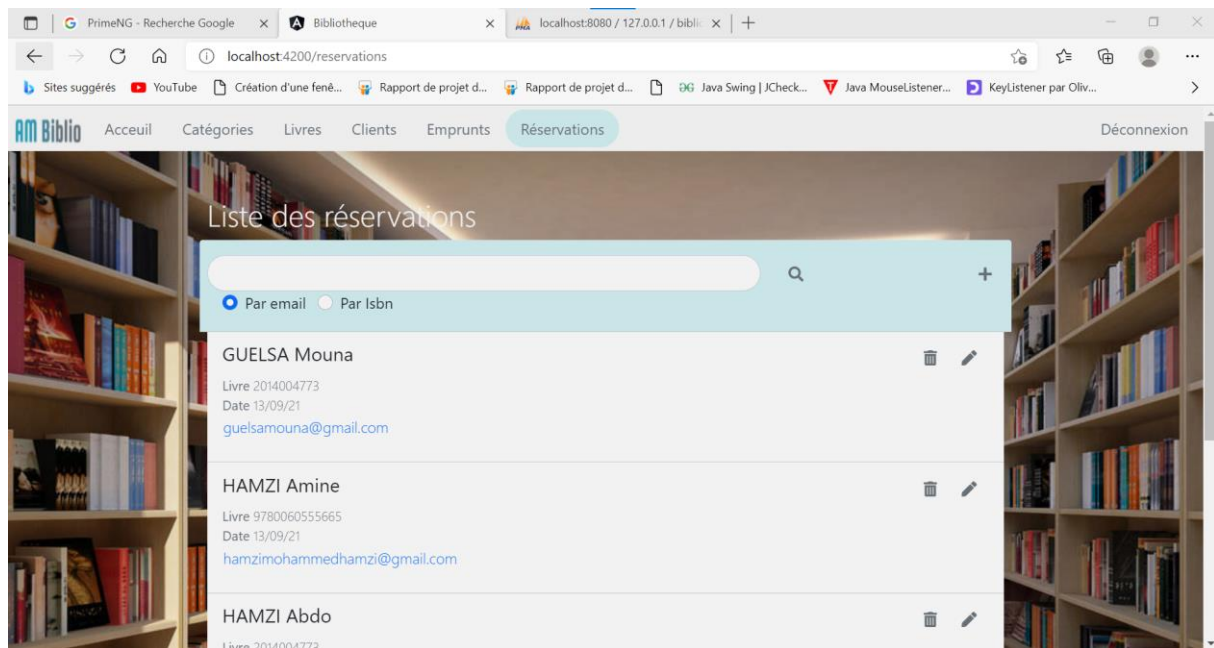


Figure 23 : Interface de gestion des réservations

La figure ci-dessous représente l'interface pour modifier les informations d'une réservation :

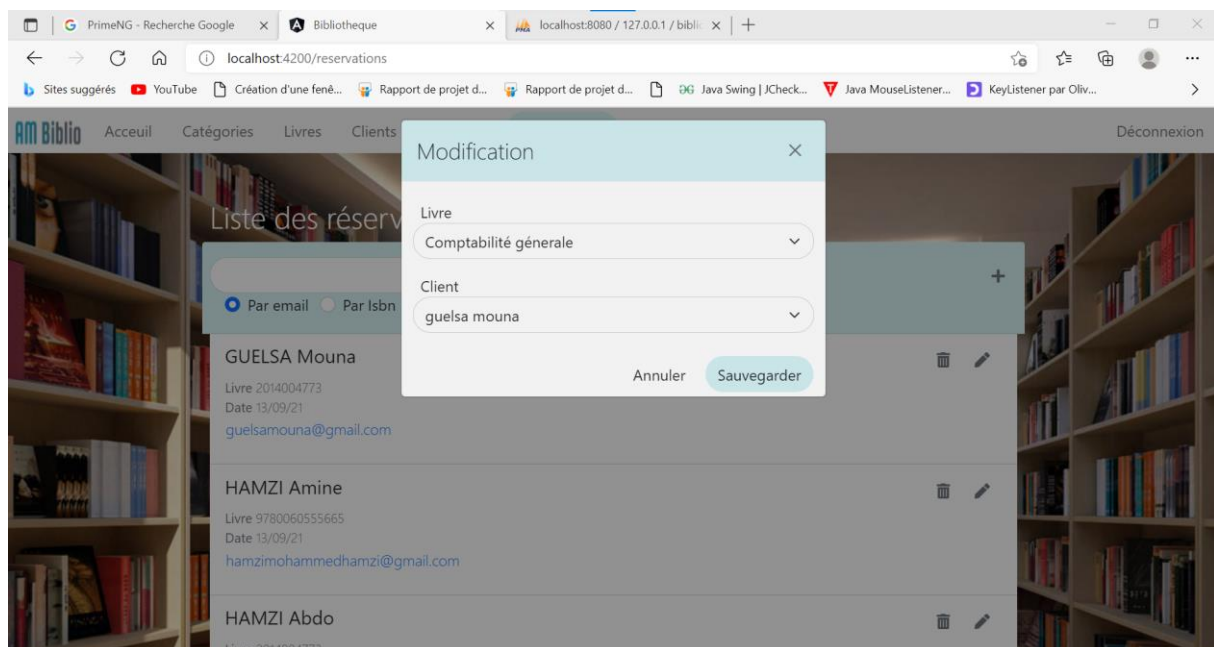


Figure 24 : interface de modification d'une réservation



La figure ci-dessous représente l'interface pour ajouter une nouvelle réservation :

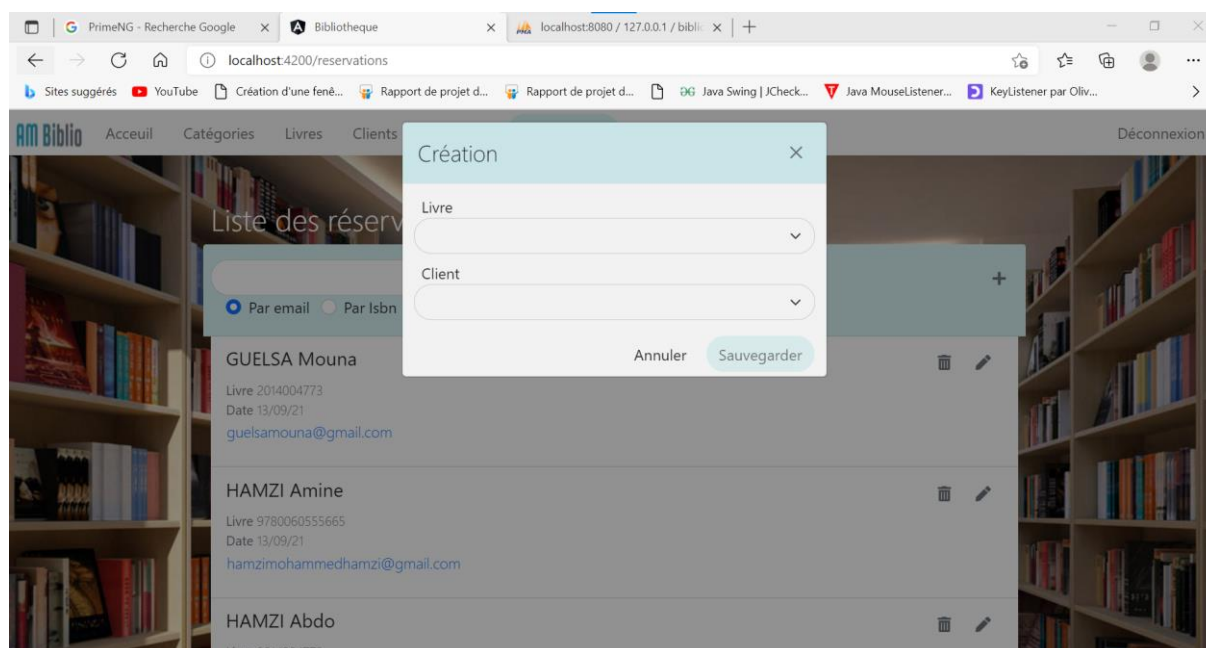


Figure 25 : Interface de création d'un emprunt

### C. Conclusion

Dans ce chapitre, nous avons présenté des démonstrations de notre application.

# VI. Test

## A. Introduction

Les tests constituent une condition indispensable au développement et à l'implémentation réussies de systèmes informatiques. La complexité des logiciels modernes est telle qu'il est presque impossible de les implémenter correctement dès la première fois sans aucune vérification.

## B. Tests unitaires

Le test unitaire est une procédure permettant de vérifier le bon fonctionnement d'une partie précise d'un logiciel ou d'une portion d'un programme (appelée « unité » ou « module »)

Vu que nous avons utilisé Spring Data Rest qui génère les contrôleurs automatiquement, donc nous avons fait des tests unitaires seulement pour le contrôleur que nous avons créé.

### 1. Test de l'envoi des emails

Dans ce test on teste l'envoi d'un email à un client donné à travers les annotations de SpringBoot (WebMvcTest, MockBean), ainsi la classe MockMvc qui permet d'appeler la fonction perform() pour exécuter les requêtes voulues.

La figure ci-dessous représente l'exécution du test :



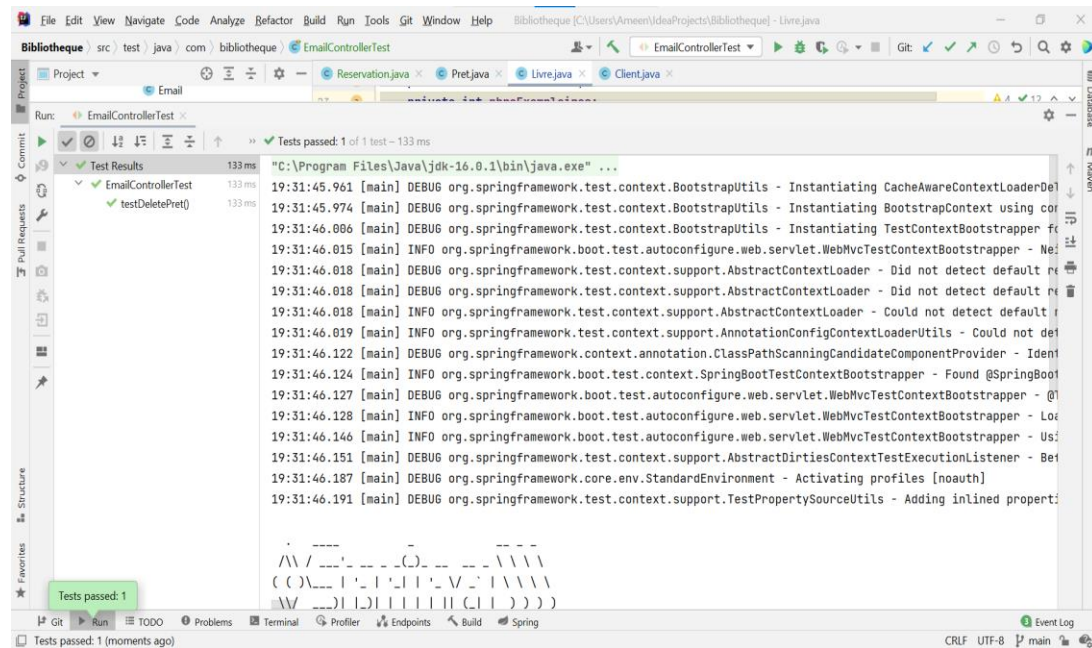


Figure 26 : Interface de test unitaires

## C. Tests d'intégration

Les tests d'intégration consistent à tester le comportement de l'application en intégrant progressivement toutes les briques du logiciel, ils permettent également de vérifier l'aspect fonctionnel, les performances et la fiabilité du logiciel. A chaque finalisation d'une fonctionnalité on réalise un test d'intégration pour vérifier le processus de fonctionnement de notre application et la compatibilité de la nouvelle fonctionnalité ajoutée avec les autres parties de logiciel.

La figure ci-dessous représente tous les tests faits sur tous les Endpoints :

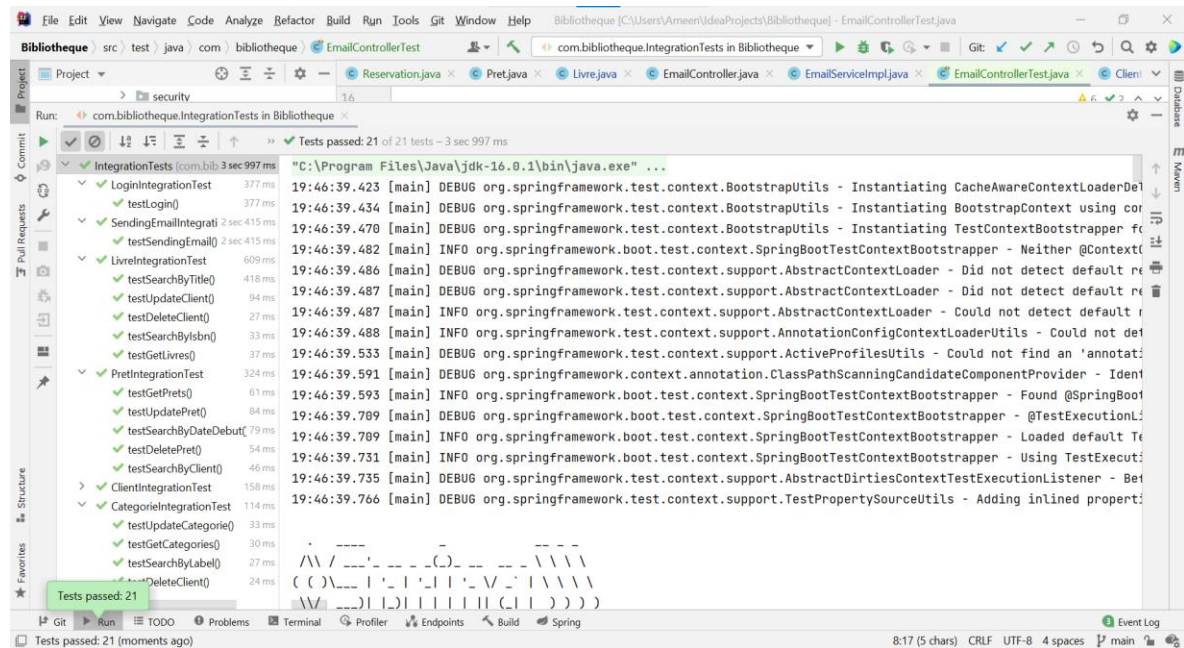


Figure 27 : Interface de tests d'intégration

# Conclusion générale

Dans le cadre de notre projet de fin d'études, nous avons conçu et développé une application web pour la gestion de prêt d'une bibliothèque permettant à son administrateur de bien gérer les emprunts, ainsi aux clients de faire une réservation s'ils n'ont pas trouvé qu'ils veulent emprunter.

Le présent rapport décrit les différentes étapes de la réalisation de ce projet. Allant du contexte général du projet, de l'étude fonctionnelle et technique, ensuite la conception détaillée du projet pour finir avec la réalisation et la mise en œuvre du projet.

Pour réaliser ce projet, nous avons opté pour une démarche de gestion de projet, nous avons commencé par définir le contexte général du projet en introduisant la problématique à laquelle notre projet doit répondre. Par la suite, dans le cadre de l'étude fonctionnelle, nous avons effectué une analyse en spécifiant les besoins fonctionnels et non fonctionnels attendues de l'application. Nous avons ensuite mené une étude technique qui servira à la réalisation de l'application.

L'étape suivante était de concevoir et de modéliser les différents cas d'utilisation, scénarios et modules de l'application à travers le standard UML, pour qu'ensuite réaliser et mettre en œuvre l'application.

Ce travail nous a permis d'approfondir nos connaissances dans la bonne pratique de programmation et la documentation.

Nous avons rencontré de nombreux problèmes au cours de l'élaboration de ce travail. Ces derniers consistent principalement en des difficultés dans la programmation vue que les outils sont nouveaux pour nous.

Nous tenons à exprimer nos remerciements et nos respects à toutes les personnes qui nous ont dirigés tout au long de ce projet et nous espérons que ce

modeste rapport pourra refléter le travail sérieux que nous avons fourni.