



## Remerciement

Avant d'ouvrir ce rapport, Nous tenons tout d'abord à remercier ceux qui nous ont assisté au cours de ce projet, et même à ceux qui ont eu l'amabilité de faire de ce projet à un moment peu propice.

Aussi, Nous remercions **Mme HANIFI Majdoulayne** et **GUERMAH Bassma** encadrants de projet qui nous ont formé et accompagné tout au long de cette expérience avec beaucoup de patience et de pédagogie.

Nous profitons de cette tribune pour remercier les encadrants qui de passage, ont pu nous apporter leur contribution, que soit au niveau des idées jusqu'à celui des conceptions.

<b>Introduction générale</b>	<b>5</b>
<b>Chapitre 1 Analyse et conception</b>	<b>6</b>
I. Introduction	6
II. Identification des acteurs	6
III. Diagramme de cas d'utilisation	7
1. Diagramme de cas d'utilisation global	7
2. Description détaillée des cas d'utilisation :	7
3. Scénarios détaillés de chaque cas d'utilisation pour le développement des interfaces graphiques	10
1) Administrateur :	10
a. Description détaillée du cas d'utilisation « Gestion des comptes » :	10
b. Description détaillée du cas d'utilisation « Consultation de la liste des étudiants qui ont dépassé le délai » :	11
c. Description détaillée du cas d'utilisation « Gestion des listes des livres » :	11
2) Etudiant :	12
a. Description détaillée du cas d'utilisation « S'inscrire » :	12
b. Description détaillée du cas d'utilisation « réserver un livre » :	12
c. Description détaillée du cas d'utilisation « emprunter un livre » :	13
d. Description détaillée du cas d'utilisation « Envoyer des réclamations » :	13
IV. Diagramme de classe :	14
V. Diagramme de séquence :	15
1. Administrateur :	15
2. Etudiant :	17
<b>Chapitre 2 Réalisation :</b>	<b>18</b>
I. Modélisation de la base de données :	18
II. Environnement de travail :	21
<b>Chapitre 3 Description de l'application</b>	<b>22</b>
I. Les fonctionnalités d'administrateur dans le système de la bibliothèque :	22
1. Vue principale de connexion et d'inscription :	22
2. Menu principale d'administrateur :	23
3. Menu de gestion des réservations :	23
4. Menu de gestion des livres :	24
5. Menu des plaintes :	26
6. Menu de gestion des comptes :	27
II. Les fonctionnalités de l'utilisateur (étudiant) dans le système de la bibliothèque :	28

1.	Menu de connexion et d'inscription : -----	28
2.	Menu principale de l'utilisateur (étudiant) : -----	29
3.	Envoi une plainte :-----	30
4.	Menu « Réserver un livre » :-----	30
5.	Menu « Emprunter un livre » : -----	31
6.	Menu des livres empruntés : -----	32
<b>Conclusion générale -----</b>		<b>34</b>

## Introduction générale

---

En tant que voies d'accès au savoir et à la culture, les bibliothèques occupent une place déterminante dans la société. Leur ressource et services procurent en effet des possibilités d'apprentissage, favorisent l'alphabétisation et l'éducation et contribuent à former les nouvelles idées et perspectives à la base des sociétés créatives et innovantes.

L'informatique a donné une nouvelle vie à certaines activités humaines, raison pour laquelle on a choisi de réaliser une application de gestion de bibliothèque pour informatiser les tâches quotidiennes d'un bibliothécaire.

« **My library** » est une application desktop, a pour objectif la conception d'un système d'information permettant d'aider à la gestion de la Bibliothèque et plus précisément, aider à la gestion des emprunts, réservations et retours des livres. Aussi, elle permette aux étudiants de consulter et réserver les livres.

### Structure de rapport :

- **Le premier chapitre** est consacré aux différents diagrammes modélisés selon la méthode UML (Unified Modeling Language).
- **Le deuxième chapitre** consiste à la phase de modélisation de base de données et l'environnement de travail.
- **Le troisième chapitre** consiste à la phase de réalisation de l'application.

# Chapitre 1 Analyse et conception

---

## I.Introduction

Comme n'importe quel type de projet, un projet informatique nécessite une phase d'analyse, suivi d'une étape de conception.

Dans la **phase d'analyse**, on cherche d'abord à bien comprendre et à décrire de façon précise les besoins des utilisateurs ou des clients. Que souhaitent-ils faire avec le logiciel ? Quelles fonctionnalités veulent-ils ? Pour quel usage ? Comment l'action devrait-elle fonctionner ? C'est ce qu'on appelle l'**analyse des besoins**. Après validation de notre compréhension du besoin, nous imaginons la solution. C'est la partie **analyse de la solution**. Dans la **phase de conception**, on apporte plus de détails à la solution et on cherche à clarifier des aspects techniques, tels que l'installation des différentes parties logicielles à installer sur du matériel. Pour réaliser ces deux phases dans un projet informatique, nous utilisons des méthodes, des conventions et des notations. UML fait partie des notations les plus utilisées aujourd'hui.

La notation UML est un **langage visuel** constitué d'un ensemble de schémas, appelés des diagrammes, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc.

Dans ce chapitre, nous allons présenter le modèle conceptuel de notre application en basant sur ce modèle. On focalisera sur ce langage et ses outils : les diagrammes. Nous verrons comment ce langage peut contribuer à la phase d'analyse des besoins et du domaine d'un projet informatique.

## II. Identification des acteurs

L'application gère deux types d'utilisateur, étudiant et administrateur. L'administrateur a le droit de gérer la liste des livres, les comptes des étudiants, gérer les emprunts. Les étudiants peuvent réserver un livre et envoyer des réclamations à l'administrateur. Chaque utilisateur est caractérisé par un identifiant, nom, adresse e-mail, login et mot de passe. Les livres sont caractérisés par id, titre, auteur, éditeur, ISBN, nombre d'exemplaire, année.

**Un acteur** représente l'abstraction d'un rôle joué par des entités externes (utilisateur, Dispositif matériel ou autre système) qui interagissent directement avec le système étudié. Les principaux profils qui auront à utiliser le SI sont les suivants :

**-L'administrateur** est l'acteur essentiel du système. Il a pour action de gérer la bibliothèque. Il ajoute et retire les livres. Également, il peut inscrire (ajouter dans la base de données) un étudiant comme le supprimer s'il ne dépend de l'université ou il n'a pas respecté les conditions d'utilisations. Il gère tout aussi, les emprunts comme il gère leurs retours. Dans le cas de l'emprunt, l'administrateur vérifie, dans un premier temps, si l'étudiant est bien inscrit ; s'il est répertorié comme étant « étudiant », il vérifie ses droits d'emprunts sinon il l'inscrit. Ensuite, il effectue une recherche du livre demandé par étudiant dans la base de

données de la bibliothèque. Bien évidemment, il gère le retour des livres et cela implique la mise à jour de l'état des livres.

-L'étudiant peut réserver et consulter les livres présents dans la base de données pour une durée maximale de 7 jours et envoyer des réclamations à l'administration.

### III. Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet. Cette partie montre les différentes étapes du cheminement de notre compréhension du projet.

#### 1. Diagramme de cas d'utilisation global

Gestion de la Bibliothèque

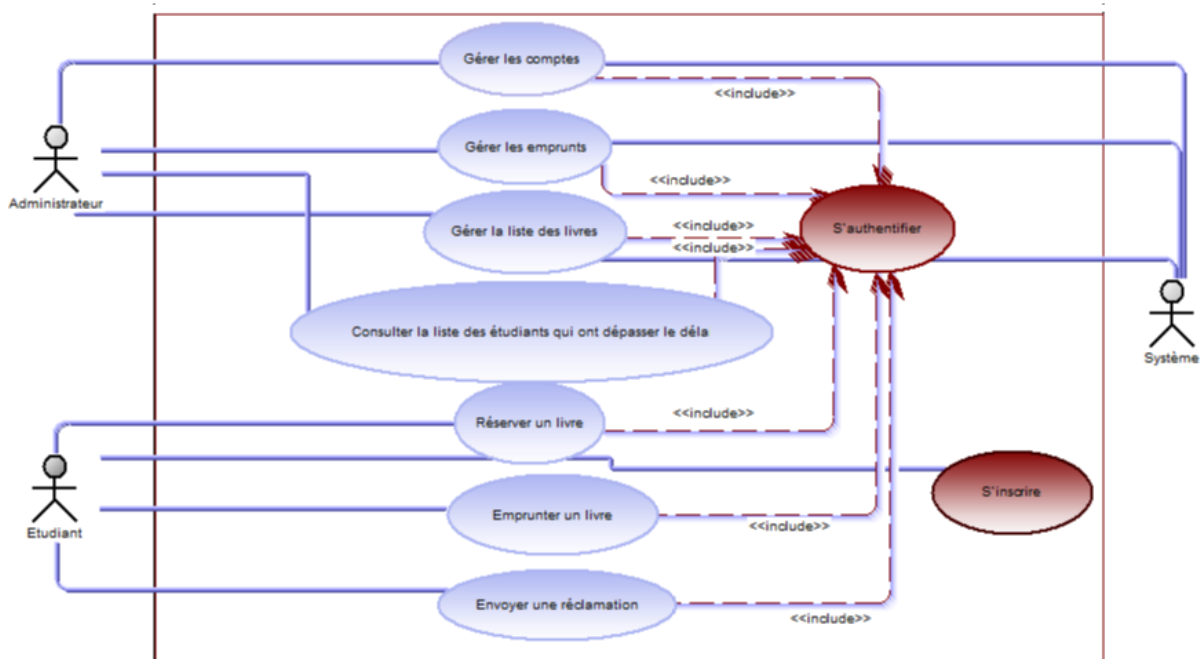


Figure 1: Diagramme d'utilisation

#### Explication :

Ce diagramme est un diagramme d'utilisation globale présente les fonctionnalités dont les différents acteurs concernés par le package auront besoin. Tout d'abord l'administrateur doit s'authentifier pour pouvoir gérer les comptes, les emprunts et la liste des livres. De plus, il peut aussi consulter la liste des étudiants qui ne respectent pas le délai. Ensuite l'étudiant doit s'inscrire et s'authentifier pour réserver, emprunter un livre et envoyer des réclamations. On avait illustré un lien entre le cas « gérer les comptes » et un système externe c'est-à-dire le système de base de données.

#### 2. Description détaillée des cas d'utilisation :

Les cas d'utilisation du système sont organisés en quatre packages :

- Gestion des comptes
- Gestion des emprunts
- Gestion des livres
- Gestion de réservation

• **Administrateur :**

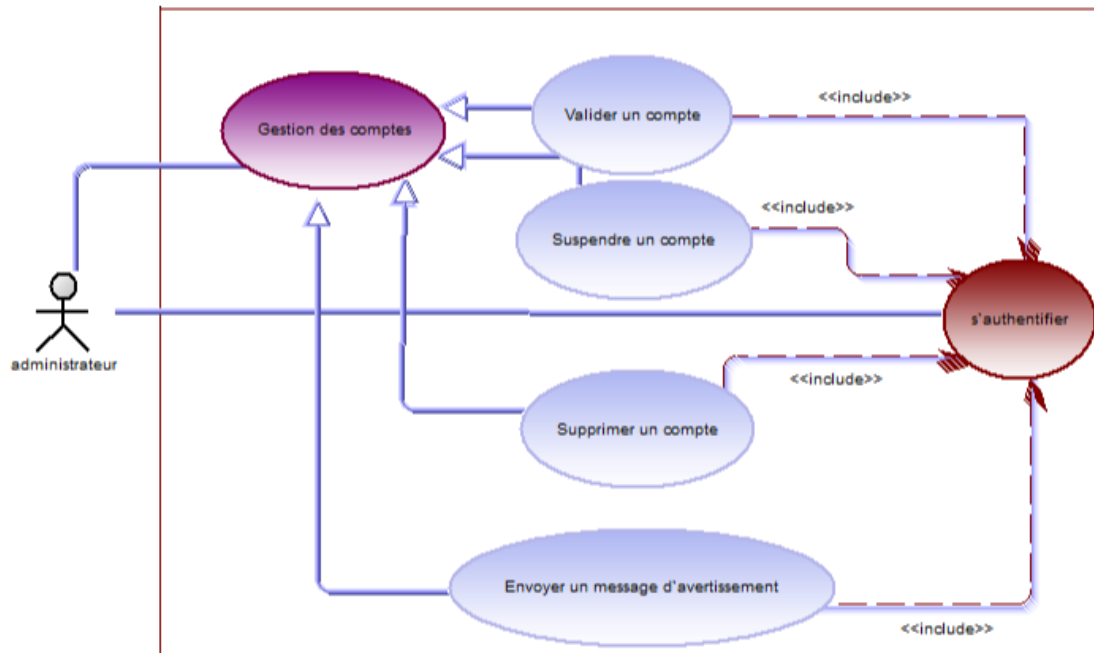


Figure 2:Diagramme de cas d'utilisation, package « Gestion des Comptes »

**Explication :**

Pour le package « Gestion des comptes » l'administrateur peut valider, suspendre, supprimer un compte et envoyer des rappels si l'étudiant oublie de retourner les livres dans un délai précis.

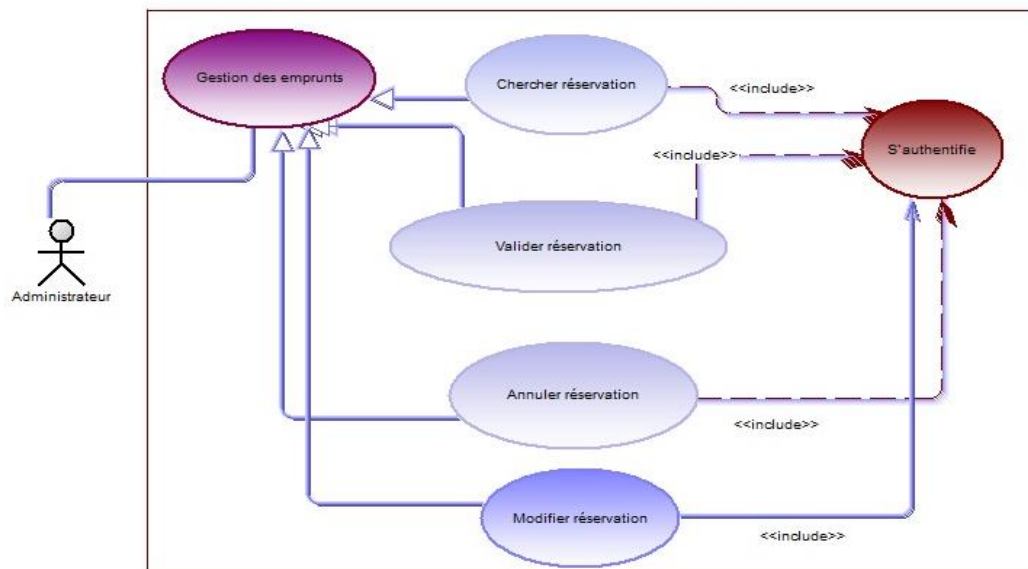


Figure 3: Diagramme de cas d'utilisation, package « Gestion des Emprunts »

**Explication :**

Pour le package « Gestion des emprunts » l'administrateur peut chercher une réservation faite par les étudiants, valider cette dernière, l'annuler ou la modifier.



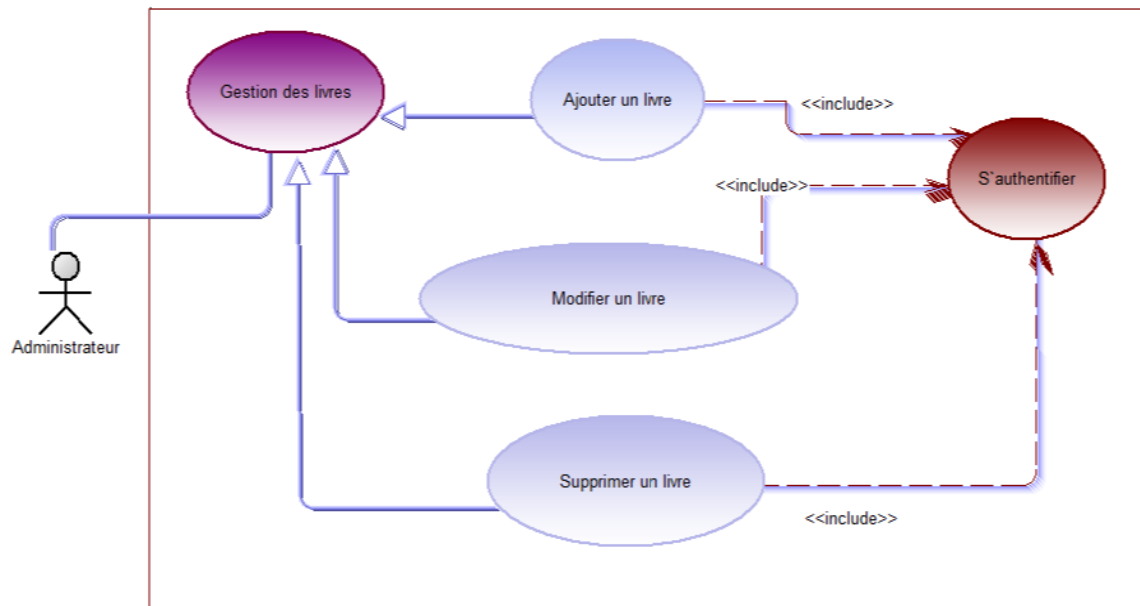
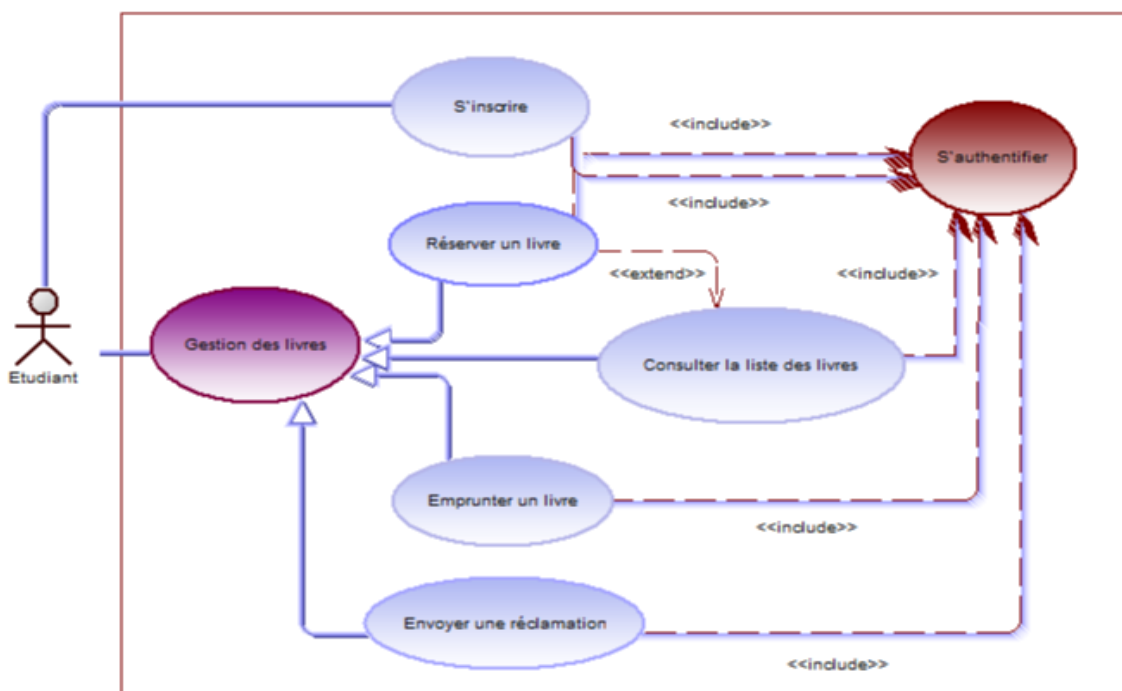


Figure 4: Diagramme de cas d'utilisation, package « Gestion des livres »

**Explication :**

Pour le package « Gestion des livres » l'administrateur peut ajouter, modifier et supprimer un livre.

- **Etudiant :**



**Explication :**

Pour ce package, l'étudiant peut réserver, emprunter un livre et réclamer. De plus il peut consulter la liste des livres qui sont enregistrés dans la base de données.

### 3. Scénarios détaillés de chaque cas d'utilisation pour le développement des interfaces graphiques

1) *Administrateur :*

a. Description détaillée du cas d'utilisation « Gestion des comptes » :

#### Description détaillée du cas d'utilisation « Valider le compte » :

Objectif :	Confirmant les informations fournies par l'utilisateur pour valider la création du compte.
Précondition :	-être authentifié autant qu'admin -l'utilisateur doit être déjà inscrit.
Scénario :	<ol style="list-style-type: none"> <li>1. En cliquant sur le bouton "liste des inscriptions".</li> <li>2. Une page contenant un tableau pour la base de données des inscriptions est ouverte.</li> <li>3. L'administrateur lit les informations fournies par les utilisateurs.</li> <li>4. Si l'administrateur juge un compte valide, il le sélectionne et clique sur le bouton "valider".</li> <li>5. Un pop-up avec le message "compte créé" est ouvert.</li> </ol>

#### Description détaillée du cas d'utilisation « Suspendre le compte » :

Objectif :	Désactiver un compte pendant un certain temps pour avoir enfreint les règles.
Précondition :	-être authentifié autant qu'admin -l'utilisateur tarde à restituer le livre de 7 jours.
Scénario :	<ol style="list-style-type: none"> <li>1. En cliquant sur le bouton "liste des comptes".</li> <li>2. Une page contenant un tableau pour la base de données des comptes est ouverte.</li> <li>3. L'administrateur sélectionne le compte incriminé et clique sur le bouton "suspendre".</li> <li>4. Un pop-up contenant un champ de texte est ouvert demandant le nombre de jours pendant lesquels l'administrateur souhaite suspendre le compte.</li> <li>5. L'administrateur saisit un nombre et clique sur le bouton "confirmer".</li> <li>6. Un pop-up avec le message "compte suspendu" est ouvert.</li> </ol>

#### Description détaillée du cas d'utilisation « Supprimer le compte » :

Objectif :	Supprimer un compte pour avoir enfreint trop souvent les règles ou pour désinscrire du service.
Précondition :	-être authentifié autant qu'admin -Utilisateur violant les règles trop souvent ou se désinscrivant du service.
Scénario :	<ol style="list-style-type: none"> <li>1. En cliquant sur le bouton "liste des comptes".</li> <li>2. Une page contenant un tableau pour la base de données des comptes est ouverte.</li> </ol>

	3. L'administrateur sélectionne le compte et clique sur le bouton "supprimer". 4. Un pop-up avec le message "compte supprimé" est ouvert.
--	--

### Description détaillée du cas d'utilisation « Envoyer un message d'avertissement » :

Objectif :	Envoyer un email d'avertissement
Précondition :	-Être authentifié autant qu'admin -L'utilisateur n'a pas renvoyé le livre après une certaine durée
Scénario :	1.En cliquant sur le bouton "Menu fonctions administrateur". 2.Appuyez sur l'onglet [Gestion des emprunts] de l'écran [Menu fonctions administrateur], puis appuyez successivement < envoyer un message d'avertissement >. 3.Remplir le champ de message par les informations que l'administrateur doit l'envoyer à l'utilisateur qui ont atteint la durée maximale de 7 jours. 4. un pop-up avec le message "le message est envoyé" est ouvert. 7. Appuyez sur [Retour]. 8. L'écran [Menu fonctions administrateur] réapparaît.

### b. Description détaillée du cas d'utilisation « Consultation de la liste des étudiants qui ont dépassé le délai » :

Objectif :	Consulter la liste des étudiants qui ont atteint la durée maximale de 7 jours
Précondition :	-Être authentifié autant qu'admin -L'utilisateur tarde à restituer le livre de 7 jours.
Scénario :	1.En cliquant sur le bouton "Menu fonctions administrateur". 2.Appuyez sur l'onglet [Gestion des emprunts] de l'écran [Menu fonctions administrateur] puis appuyez successivement < Lister la liste des retards >. 3.Une page contenant un tableau pour la base de données des comptes des étudiants qui ont dépassé la durée maximale.

### c. Description détaillée du cas d'utilisation « Gestion des listes des livres » :

### Description détaillée du cas d'utilisation « Ajouter un livre » :

Objectif	Ajouter un livre
Précondition	-Être authentifié autant qu'admin
Scénario	1. En cliquant sur le bouton "Menu fonctions administrateur". 2. Appuyez sur l'onglet [Gestion des listes des livres] de l'écran [Menu fonctions administrateur], puis appuyez successivement < d'ajouter >. 3. Admin clique sur ajouter un livre et une fenêtre pop-up s'affiche. 4. Admin saisit le nom de livre. 5. Le système vérifie le nom un livre saisi 6. Un pop-up avec le message "c'est fait livre Inscription". 7. Appuyez sur [Retour]. 8. L'écran [Menu fonctions administrateur] réapparaît.

### Description détaillée du cas d'utilisation « Modifier un livre » :

Objectif	Modification ou mettre à jour un livre
Précondition	-être authentifié autant qu'admin
Scénario	<ol style="list-style-type: none"> <li>1. en cliquant sur le bouton "Menu fonctions administrateur".</li> <li>2. Appuyez sur l'onglet [Gestion des listes des livres] de l'écran [Menu fonctions administrateur], puis appuyez successivement &lt; modifier ou mettre à jour &gt;.</li> <li>3. Admin clic sur modifier un livre et une fenêtre popup s'affiche et admin effectue ses modifications. <ol style="list-style-type: none"> <li>3-1. Admin confirme et l'événement est modifié.</li> <li>3-2. Admin ne confirme pas et rien ne se change.</li> </ol> </li> <li>4. Appuyez sur [Retour].</li> <li>5. L'écran [Menu fonctions administrateur] réapparaît.</li> </ol>

### Description détaillée du cas d'utilisation « Supprimer un livre » :

Objectif	Supprimer un livre
Précondition	-être authentifié autant qu'admin
Scénario	<ol style="list-style-type: none"> <li>1. en cliquant sur le bouton "Menu fonctions administrateur".</li> <li>2. Appuyez sur l'onglet [Gestion des listes des livres] de l'écran [Menu fonctions administrateur], puis appuyez successivement &lt; supprime &gt;.</li> <li>3. Admin clic sur supprime un livre et une fenêtre popup s'affiche et admin sélectionne le livre et clique sur le bouton "supprimer".</li> <li>4. un pop-up avec le message "Livre supprimé" est ouvert.</li> <li>7. Appuyez sur [Retour].</li> <li>8. L'écran [Menu fonctions administrateur] réapparaît.</li> </ol>

2) *Etudiant :*

a. Description détaillée du cas d'utilisation « S'inscrire » :

Objectif :	S'abonner au système
Précondition :	-L'utilisateur doit être un étudiant
Scénario :	<ol style="list-style-type: none"> <li>1. Cliquez sur le bouton s'inscrire</li> <li>2. Remplir le formulaire d'inscription est affiché</li> <li>3. L'étudiant doit lire et accepter les conditions d'utilisations</li> <li>4. Si l'étudiant clique sur le bouton valider, le compte est créé.</li> </ol>

b. Description détaillée du cas d'utilisation « réserver un livre » :

Objectif :	Effectuer une réservation d'un livre
Précondition :	-L'étudiant doit être déjà inscrit -être authentifié
Scénario :	<ol style="list-style-type: none"> <li>1. Demande d'identification de l'étudiant (login, password)</li> <li>2. Affichage de l'écran d'accueil (Menu)</li> <li>3. Choix de l'opération « réservation »</li> <li>4. L'étudiant demande à connaître la liste des livres présents à la bibliothèque</li> <li>5. Choisir un livre</li> <li>6. Message « Réservation ok »</li> </ol>

	7. Refus : « Trop de livre déjà réserves ! » 8. Demande de déconnexion
--	---

c. Description détaillée du cas d'utilisation « emprunter un livre » :

Objectif :	Effectuer un emprunt
Précondition :	-L'étudiant doit être déjà inscrit -être authentifié -l'étudiant ne doit pas avoir atteint le nombre maximum de livre à emprunter -le livre doit être disponible et déjà réservé
Scénario :	1. Demande d'identification de l'étudiant (login, password) 2. Affichage de l'écran d'accueil (Menu) 3. Choix de l'opération « emprunter un livre » 4. Le système indique que le livre réservé est disponible 5. Message « cette opération s'est terminée avec succès » 6. Refus : « Vous avez atteint le nombre de livre à emprunter ! » 7. Demande de déconnexion

d. Description détaillée du cas d'utilisation « Envoyer des réclamations » :

Objectif :	-Ajouter une réclamation
Précondition :	-être authentifié
Scénario :	6. Demande d'identification de l'étudiant (login, password) 7. Affichage de l'écran d'accueil (Menu) 8. Choix de l'opération « réclamer » 9. Demande d'ajout d'une réclamation 10. Le système ajoute une réclamation à la base de données 11. Message « votre réclamation sera traitée de manière équitable »

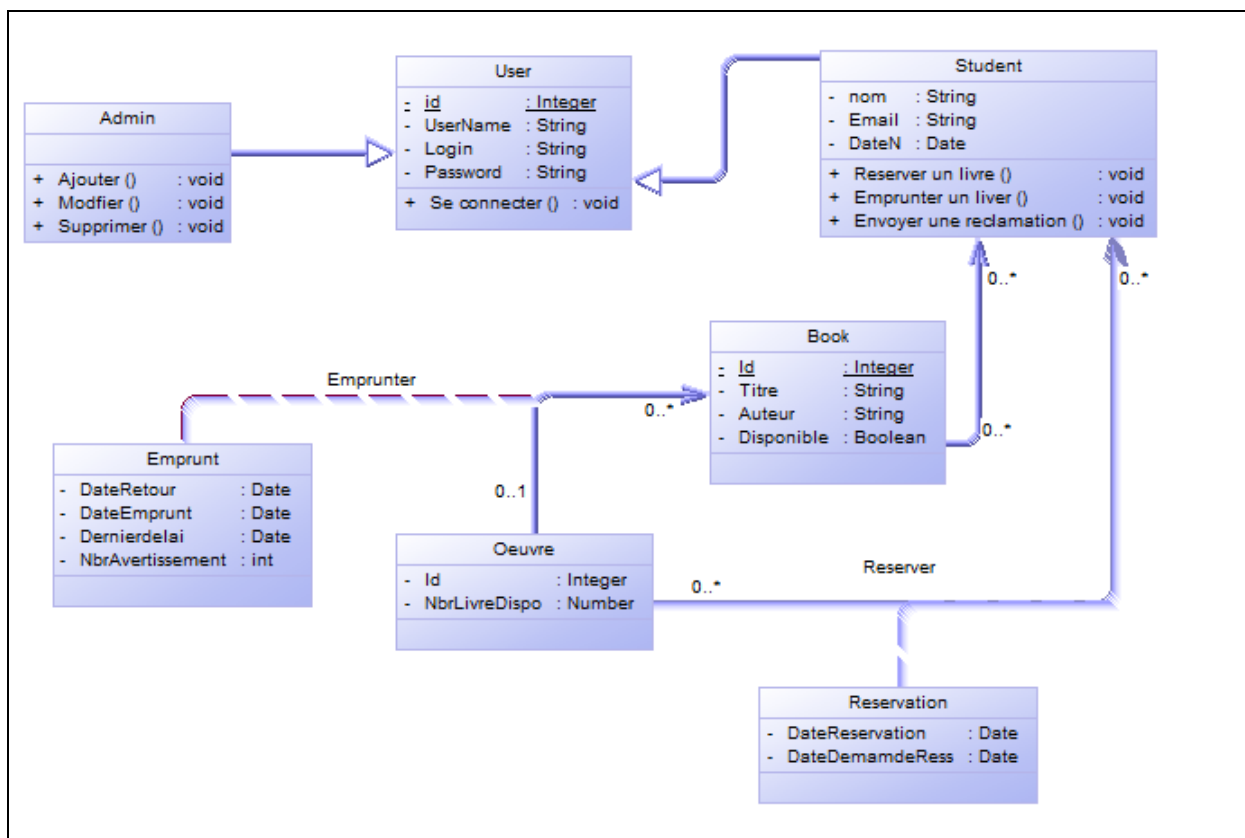
#### IV. Diagramme de classe :

Le diagramme de classes est un schéma utilisé pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

Dans la phase d'analyse, ce diagramme représente les entités (des informations) manipulées par les utilisateurs.

Dans la phase de conception, il représente la structure objet d'un développement orienté objet.

##### Schéma :



##### Explication :

D'après l'énoncé, l'utilisateur est soit un administrateur ou un étudiant, donc les classes Student et Admin qui héritent de la classe User.

Sachant qu'un livre possède un ou plusieurs exemplaires et l'exemplaire (œuvre) est associé à un seul et unique livre, la classe livre est relié à la classe Œuvre avec une association de composition.

Le fait que l'étudiant va réserver des exemplaires de livre, une association réservation relie Student à Exemplaire. Comme l'association réservation à plusieurs propriétés, elle sera alors modélisée comme une classe d'association Reservation. Les réservations vont être sauvegardé dans le système, à partir de là, on parlera de tous les réservations qui ont été effectuées et non des réservations en cours, donc l'étudiant, ne doit pas avoir réserver d'exemplaire ou, avoir réservé plusieurs exemplaires. Ainsi qu'un exemplaire doit être indisponible c'est-à-dire il peut avoir être réserver par un autre étudiant.

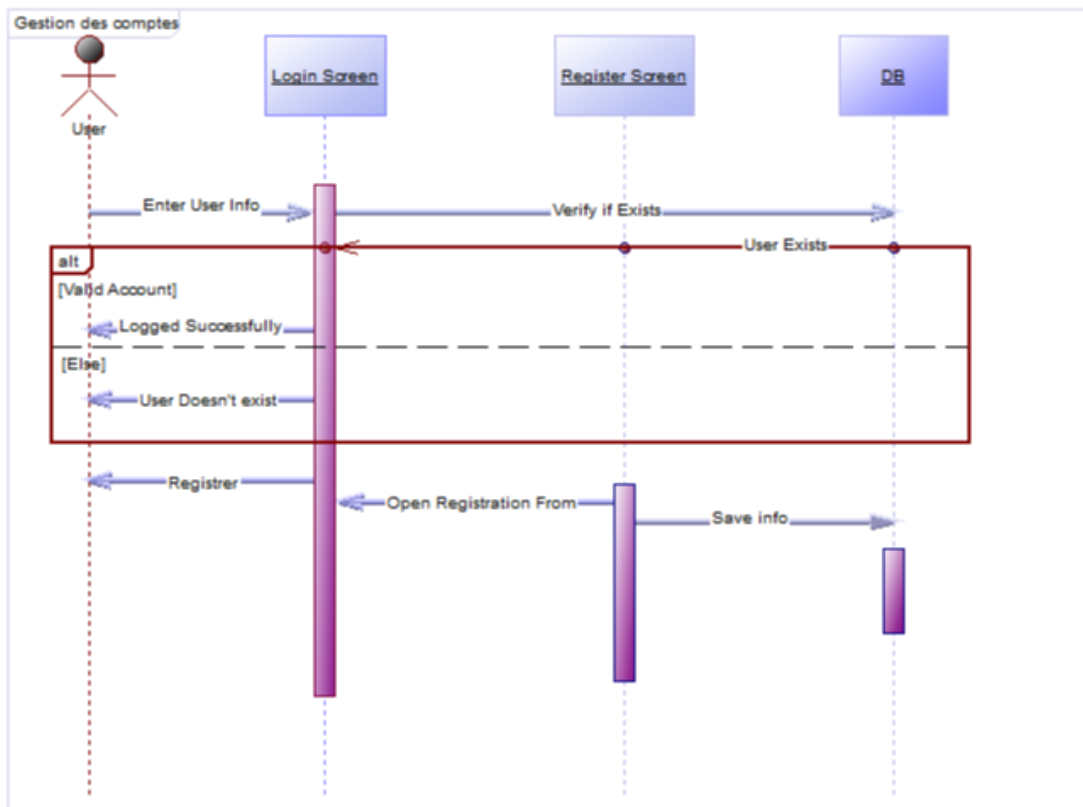
Le fait que l'étudiant va emprunter des exemplaires de livre, une association emprunt relie Student à Exempleire. Comme l'association emprunt à plusieurs propriétés, elle sera alors modélisée comme une classe d'association Emprunt.

### V.Diagramme de séquence :

Le diagramme de séquence permet de décrire les différents scénarios d'utilisation du système. C'est une représentation intuitive qu'on a utilisée pour concrétiser des interactions entre nos entités. Il décrit un scénario qui modélise une exécution particulière d'un cas d'utilisation du début jusqu'à la fin. Il correspond à une sélection d'enchaînements du cas d'utilisation.

#### Schémas :

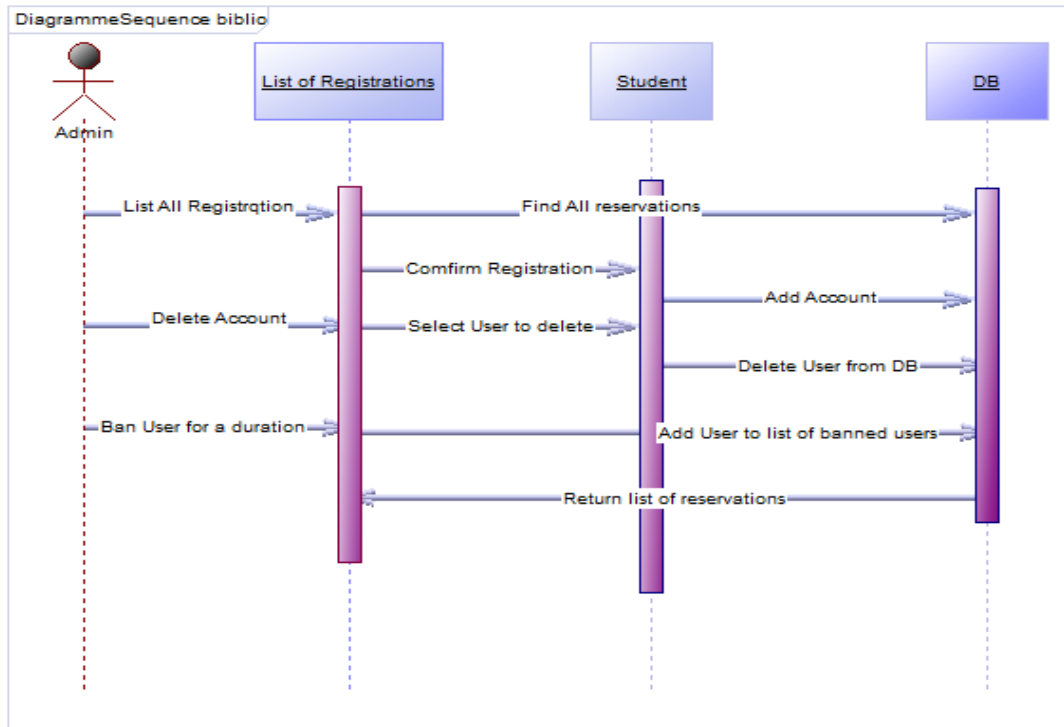
##### 1. Administrateur :



#### Explication :

Tout d'abord l'utilisateur doit s'inscrire si n'a pas un compte. Il accède au formulaire puis entre les informations nécessaires. Ensuite le système vérifie si l'utilisateur est un étudiant dans l'université ou non, si oui l'administrateur confirme l'inscription et l'enregistre dans la base de données. Dans ce cas l'utilisateur peut recevoir une notification de confirmation (le compte a été validé avec succès). Sinon la demande n'est pas acceptée dans ce cas l'utilisateur ne peut pas avoir accès à l'application.

## Gestion des emprunts :

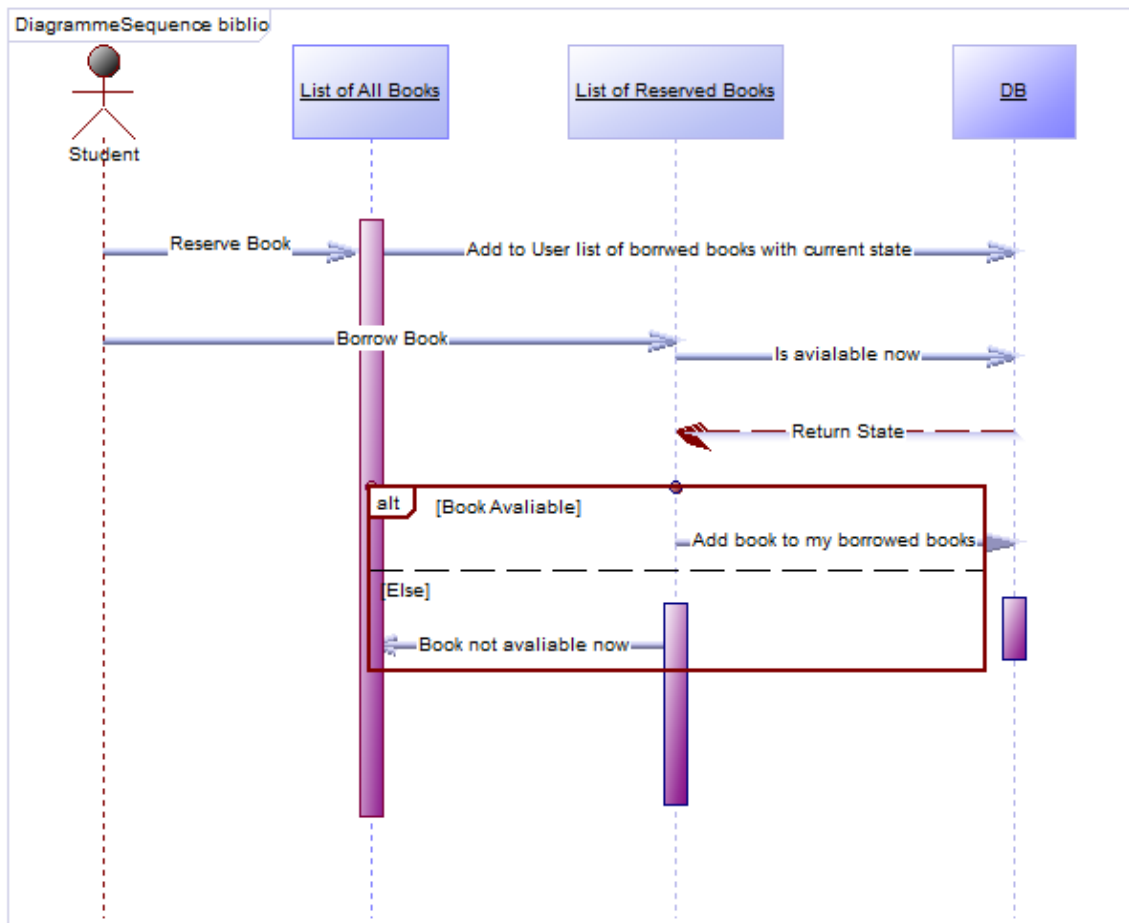
**Explication :**

L'administrateur consulte la liste de toutes les réservations qui seront enregistrés dans la base de données, puis il peut confirmer une réservation.

De plus, il peut consulter la liste des comptes et sélectionne un ou plusieurs comptes pour le supprimer ou désactive son compte puis ça sera ajouter à la liste des comptes suspendus (banned user). Aussi, il peut ajouter un étudiant. Finalement, une mise à jour de la liste.



## 2. Etudiant :

**Explication :**

L'étudiant réserve un livre. Il sera ajouté à la liste des réservations dans la base de données en attendant la confirmation pour l'emprunter. Il peut emprunter le livre s'il est disponible et le livre sera enregistré avec un nouvel état (State=Non Dispo). L'étudiant peut consulter la liste des livre empruntés. Si le livre n'es pas disponible l'étudiant ne peut pas l'emprunter.

## Chapitre 2 Réalisation :

Ce chapitre est consacré à la base de données de notre application et les environnements utilisés.

### I. Modélisation de la base de données :

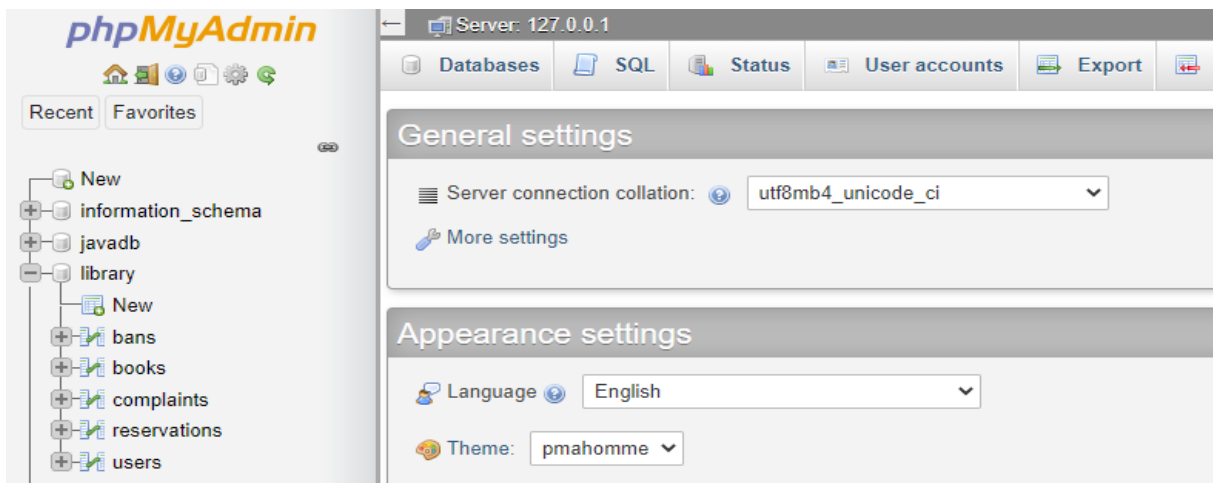
Dans une application desktop on utilise toujours une base de données cette base va stocker toute sorte de donnée le quel on va utiliser pour notre application desktop.

Dans notre projet on a décidé d'utiliser une base de données MySQL afin de se stocker les données qu'on va utiliser pour notre application : les livres, les comptes, etc..

Après l'installation du serveur local XAMPP et à l'aide du programme web PhpMyAdmin, on va créer une nouvelle base de données.

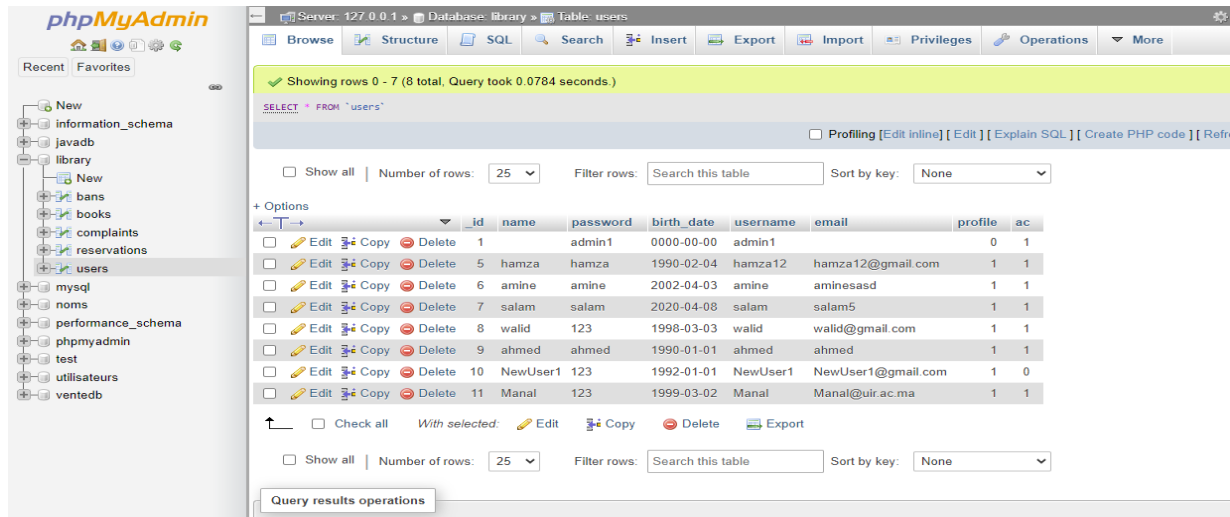
Il s'agit de la base de données « **library** ».

**Xampp** est utilisé pour exécuter un hôte local pour stocker la base de données.



Dans la base de données library on a ajouté 5 tables et dans chaque table on va créer des champs selon nos besoins.

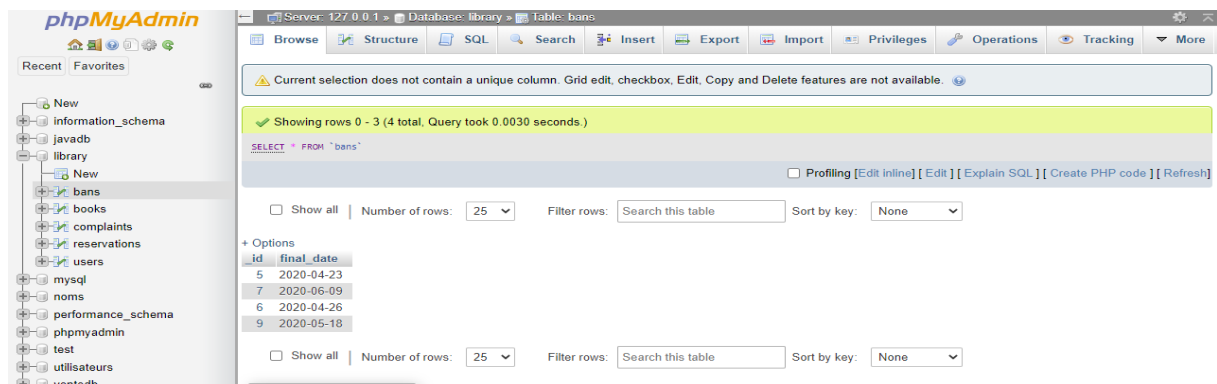
La table « **users** », c'est ici que les utilisateurs sont conservés. Où l'id auto-incrémenté (primary key), le nom d'utilisateur, le mot de passe et d'autres informations sur l'utilisateur sont conservés. A noter le mot de passe n'est pas crypté, il peut donc être clair dans les captures d'écran. La variable **profile**, si c'est 1, c'est un simple utilisateur, 0 si c'est un administrateur. La variable **ac** est pour si l'utilisateur a été accepté.



The screenshot shows the phpMyAdmin interface with the 'users' table selected. The table structure and data are as follows:

	id	name	password	birth_date	username	email	profile	ac
	1	admin1	0000-00-00	admin1			0	1
	5	hamza	hamza	1990-02-04	hamza12	hamza12@gmail.com	1	1
	6	amine	amine	2002-04-03	amine	aminesasd	1	1
	7	salam	salam	2020-04-08	salam	salam5	1	1
	8	walid	123	1998-03-03	walid	walid@gmail.com	1	1
	9	ahmed	ahmed	1990-01-01	ahmed	ahmed	1	1
	10	NewUser1	123	1992-01-01	NewUser1	NewUser1@gmail.com	1	0
	11	Manal	123	1999-03-02	Manal	Manal@uir.ac.ma	1	1

La table « **bans** », c'est ici que les interdictions que l'administrateur a appliquées à certains utilisateurs sont affichées et où la date de fin est conservée et corrélée avec une foreign key "id" tirée de la table « **users** ».



The screenshot shows the phpMyAdmin interface with the 'bans' table selected. The table structure and data are as follows:

	id	final_date
	5	2020-04-23
	7	2020-06-09
	6	2020-04-26
	9	2020-05-18

La table « **book** », c'est ici que les livres sont conservés et où l'id auto-incrémenté (primary key) et d'autres informations sur les livres sont conservés.

The screenshot shows the phpMyAdmin interface with the 'books' table selected in the 'library' database. The table structure is displayed with columns: id, name, author, and quantity. The data is shown in a table with 22 rows.

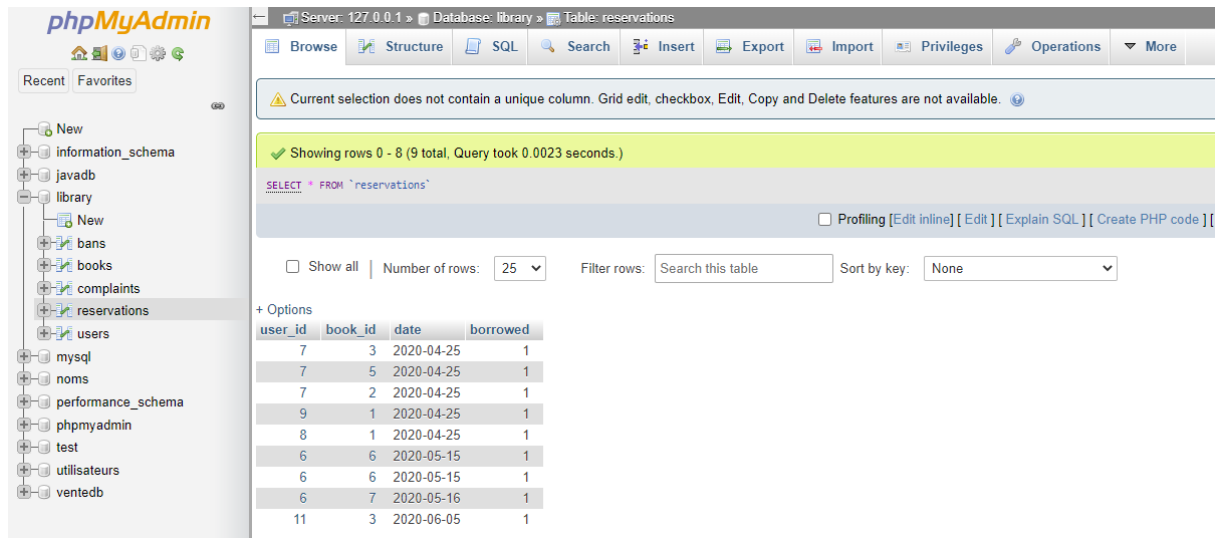
	id	name	author	quantity
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	bookA	walo	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	bookB	bookB	6
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	BOOKER	T	4
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	Harry Potter	Harry Potter	12
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	Harry Potter	Harry Potter	9
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	aaa	a	0
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	sd	asas	1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	test	asas	4
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	dsfsdfds	cx	7
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	11	dsfsdfdsdsfsd	sdds	8
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	13	ttt	dd	8
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	14	ttt	ttt	8
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	15	ttt	ttt	8
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	16	ttt	ttt	8
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	17	ttt	ttt	8
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	19	BookLast	Test3	3
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	20			1
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	21	dsfsdfdsdsfsd	dsfsdfdsdsfsd	8
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	22	NewBook	NewAuthor	11

La table « **complaints** », c'est ici que s'affichent les plaintes émises par les utilisateurs à l'aide de l'application et où l'id auto-incrémenté (primary key) et les messages sont conservés et corrélés avec une foreign key "id" tirée de la table « **users** ».

The screenshot shows the phpMyAdmin interface with the 'complaints' table selected in the 'library' database. The table structure is displayed with columns: id, idUser, and message. The data is shown in a table with 3 rows.

	id	idUser	message
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	7	Salam is declaring a complaint Ahmed here saying that
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	9	you are a fsdfdsf THE END...
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	11	Adding a complaint is easy, fixing it is hard! lol

La table « **réservations** », c'est ici que les réservations faites par les utilisateurs sur des livres spécifiques en utilisant l'application sont affichées et où la date de réservation et la quantité empruntée sont conservées et corrélées avec les 2 clés étrangères "id" tirées respectivement du tables « **users** » et « **books** ».



Server: 127.0.0.1 » Database: library » Table: reservations

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 8 (9 total, Query took 0.0023 seconds.)

SELECT \* FROM `reservations`

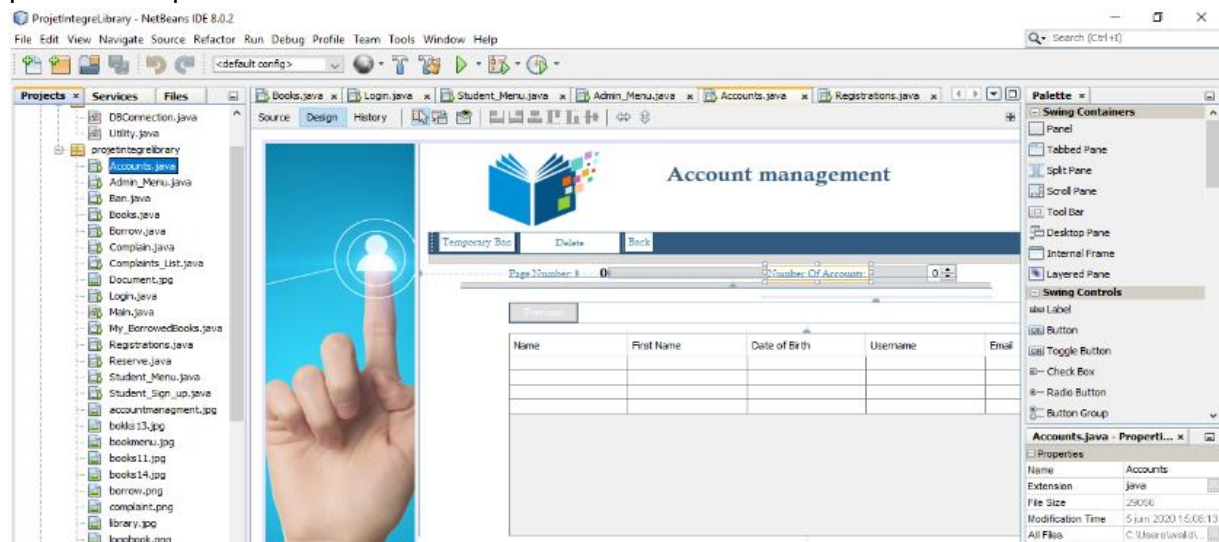
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

user_id	book_id	date	borrowed
7	3	2020-04-25	1
7	5	2020-04-25	1
7	2	2020-04-25	1
9	1	2020-04-25	1
8	1	2020-04-25	1
6	6	2020-05-15	1
6	6	2020-05-15	1
6	7	2020-05-16	1
11	3	2020-06-05	1

## II. Environnement de travail :

Pour l'implémentation nous avons utilisé le langage de programmation JAVA qui possède un certain nombre de caractéristiques qui ont contribué à son énorme succès. Il est interprété, indépendant de toute plate-forme, orienté objet, simple et assure la gestion de la mémoire. Nous avons utilisé NetBeans pour créer un nouvel espace de travail. NetBeans est conçu depuis des composants standard et expose leurs propriétés. Cela met NetBeans à une position unique de flexibilité.



## Chapitre 3 Description de l'application

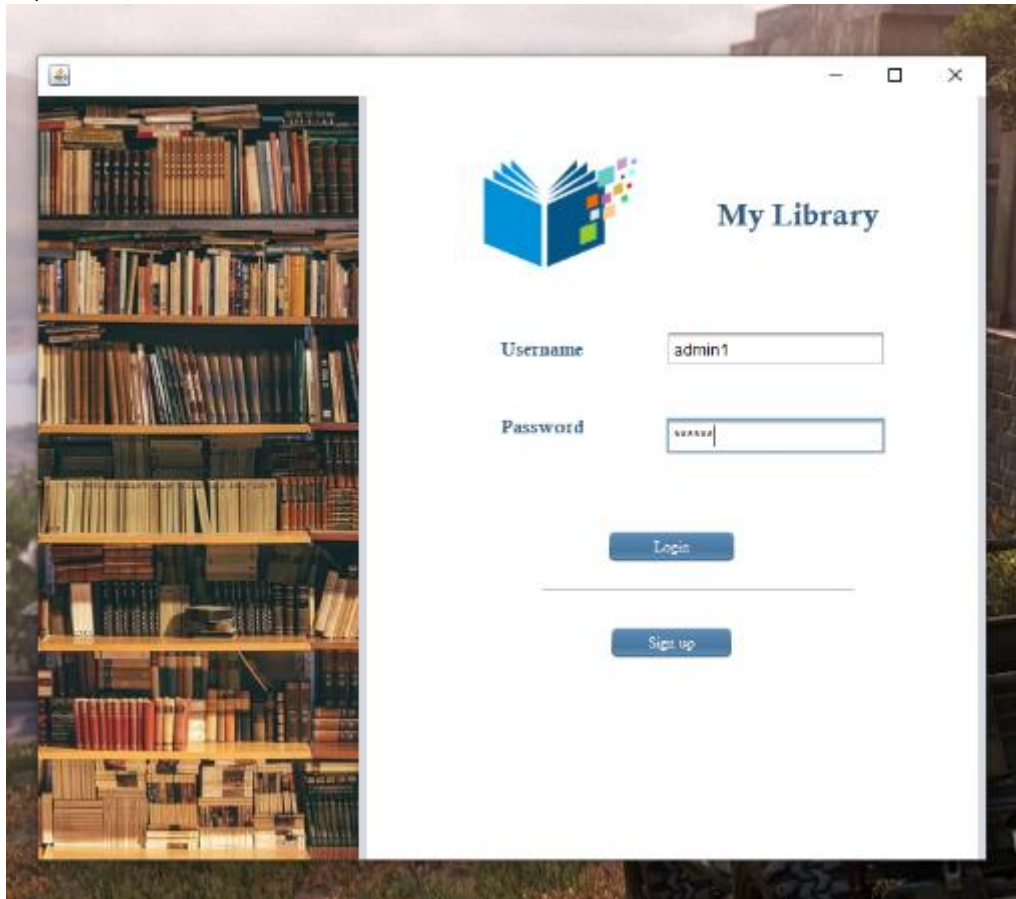
Nous allons présenter dans cette partie la création des interfaces de l'application.

### I. Les fonctionnalités d'administrateur dans le système de la bibliothèque :

#### 1. Vue principale de connexion et d'inscription :

Tout d'abord, l'utilisateur n'a pas un compte enregistré. Il peut créer un à l'aide du bouton d'inscription (**Sign up**).

Ici, un administrateur se connecte.





## 2. Menu principale d'administrateur :

Ceci est le menu principal d'un administrateur. Il peut lister toutes les inscriptions (comptes à confirmer ou non) et les utilisateurs actuels. Il peut aussi lister tous les livres et les plaintes.



## 3. Menu de gestion des réservations :

Voici une liste des inscriptions. L'administrateur peut soit confirmer soit refuser une inscription. Une fois que c'est confirmé, l'utilisateur peut accéder officiellement à son compte récemment créé. Nous pouvons également voir que les données présentées dans le tableau prennent en charge la pagination.

Book reservation management

Previous Number Of Registrations: 2 next Page Number: 1

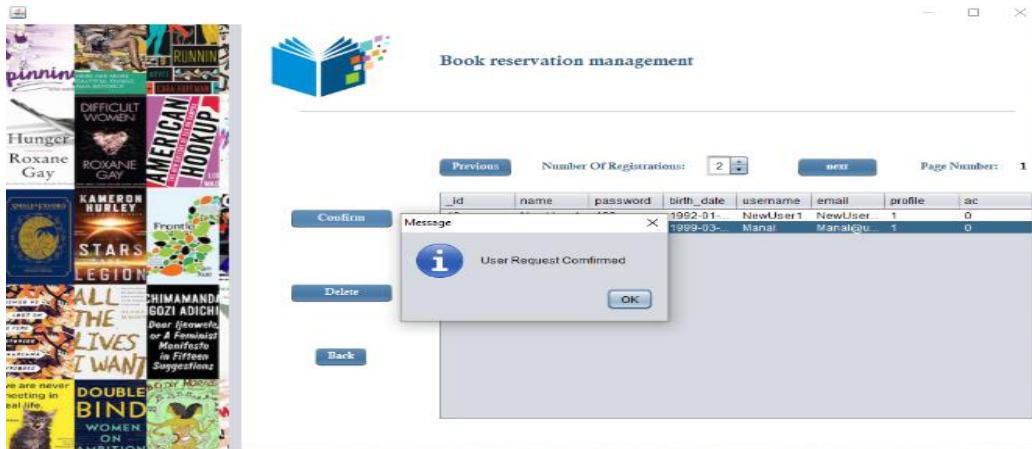
_Id	name	password	birth_date	username	email	profile	ac
10	NewUser1	123	1992-01-	NewUser1	NewUser...	1	0
11	Manal	123	1999-03-	Manal	Manal@u	1	0

Confirm

Delete

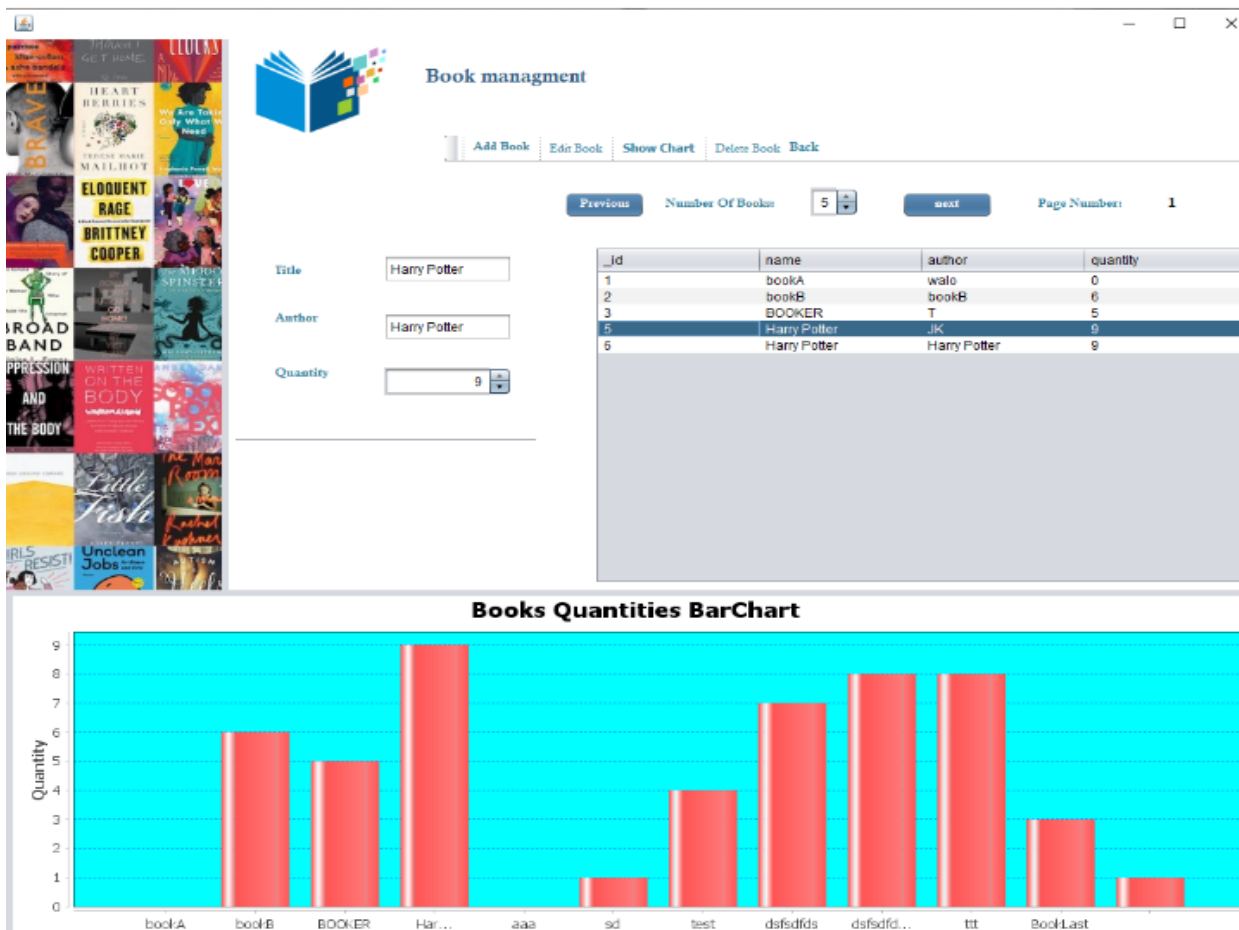
Back

- Ici, l'administrateur confirme l'enregistrement de l'utilisateur sélectionné. Un message apparaît pour montrer que la demande d'un utilisateur a été acceptée.



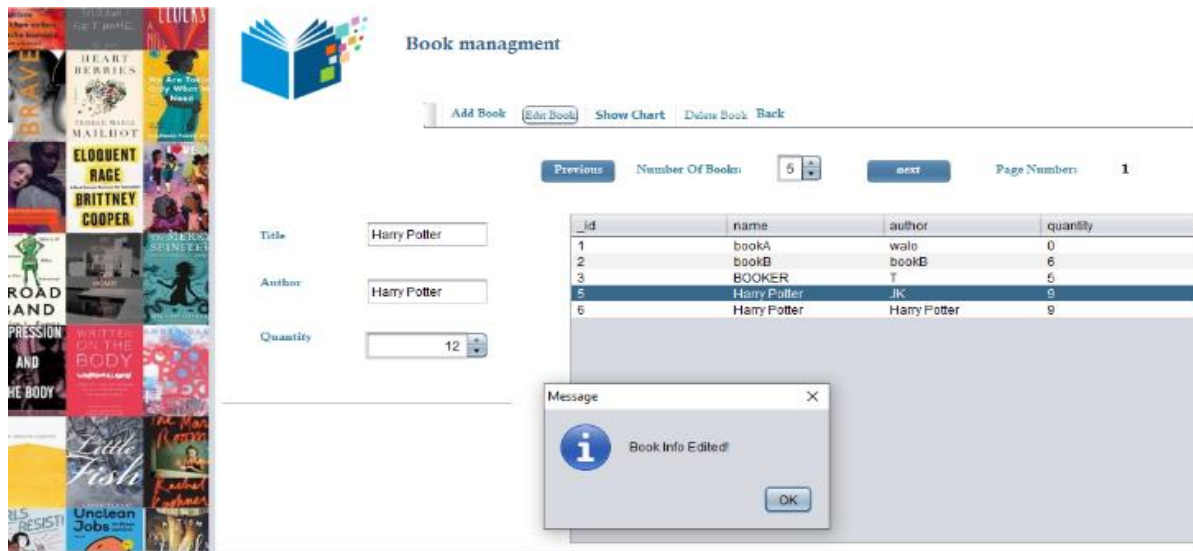
#### 4. Menu de gestion des livres :

Un administrateur a accès à la liste des livres avec toutes leurs informations. Il peut ajouter, supprimer et modifier les informations d'un livre. Il peut également afficher un graphique pour afficher les statistiques de la quantité des livres. Ici, la table prend également en charge la pagination.





- Dans cet exemple, l'administrateur a édité les informations d'un livre sélectionné. Changeant sa quantité de 9 à 12 et le nom de l'auteur aussi.



**Book management**

Add Book Edit Book Show Chart Delete Book Back

Previous Number Of Books: 5 Next Page Number: 1

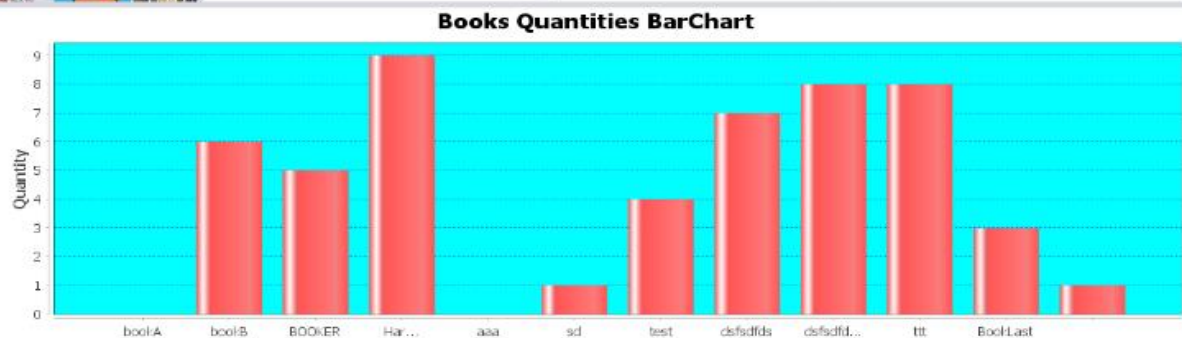
Title: Harry Potter

Author: Harry Potter

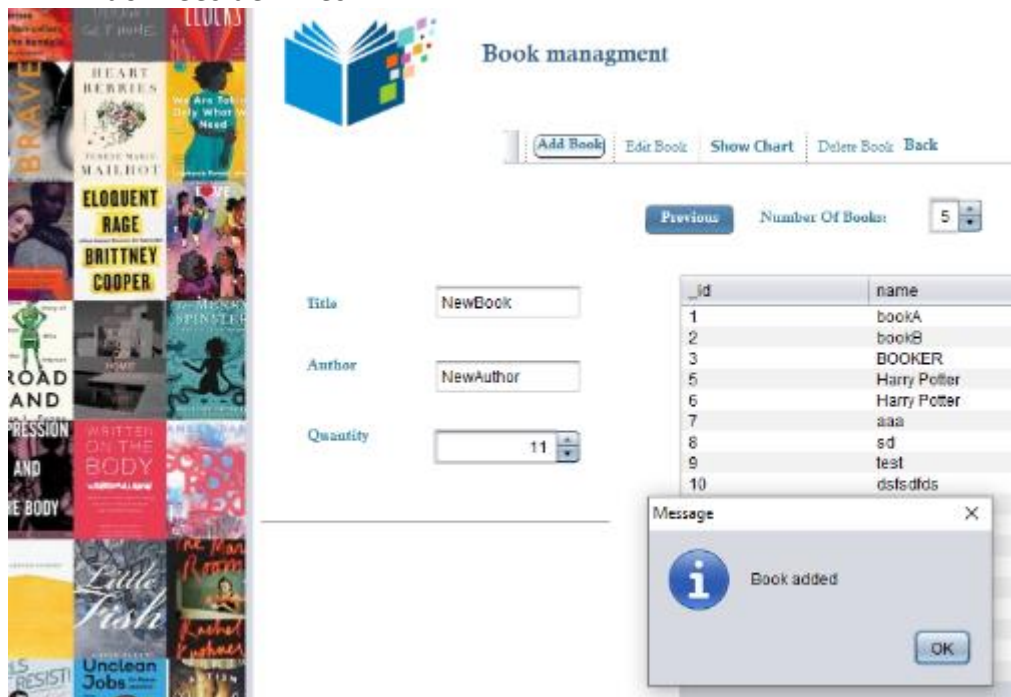
Quantity: 12

_id	name	author	quantity
1	bookA	wale	0
2	bookB	bookB	6
3	BOOKER	T	6
5	Harry Potter	JK	9
6	Harry Potter	Harry Potter	9

Message: Book Info Edited! OK



- L'administrateur ajoute ici un livre qui sera ajouté à la table de base de données de livres.



**Book management**

Add Book Edit Book Show Chart Delete Book Back

Previous Number Of Books: 5 Next

Title: NewBook

Author: NewAuthor

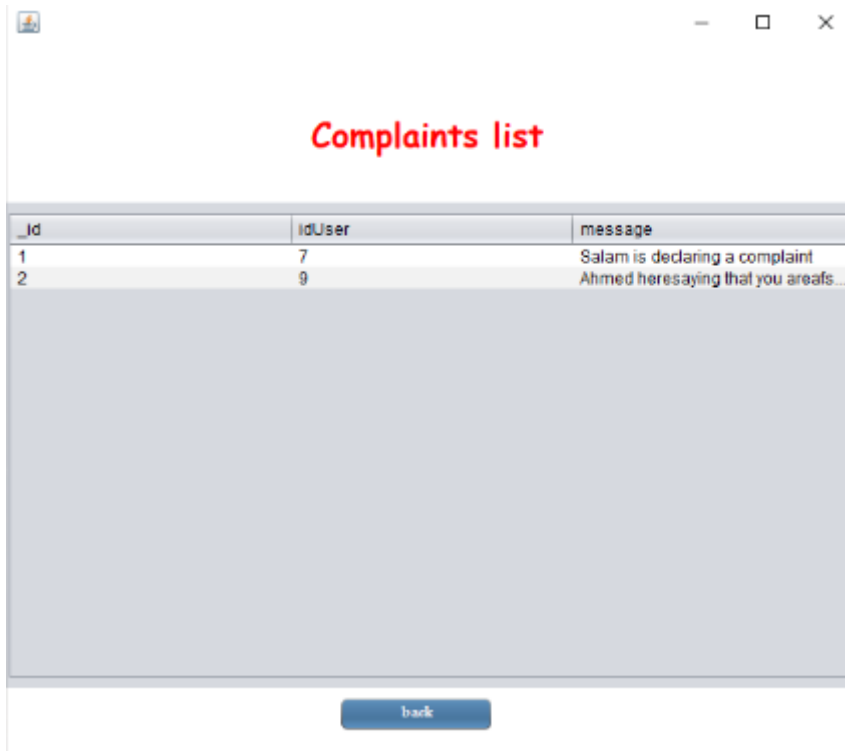
Quantity: 11

_id	name
1	bookA
2	bookB
3	BOOKER
5	Harry Potter
6	Harry Potter
7	aaa
8	sd
9	test
10	dsfsdfds

Message: Book added OK

## 5. Menu des plaintes :

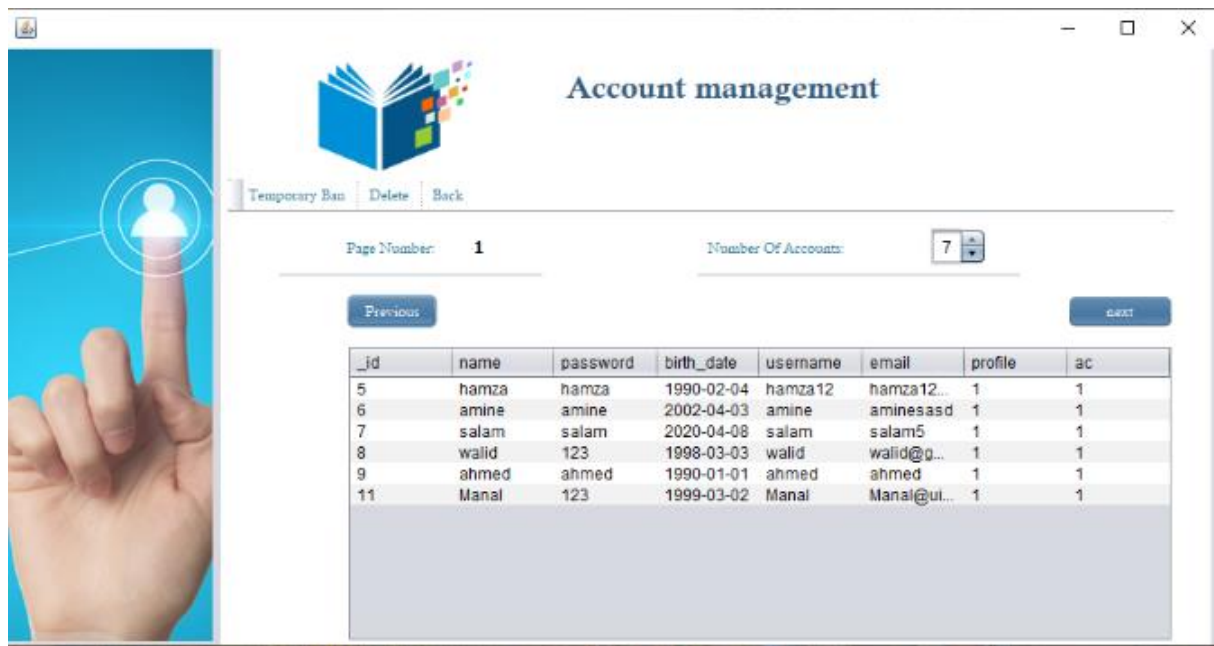
L'administrateur peut également afficher la liste des réclamations envoyées par les utilisateurs.



## 6. Menu de gestion des comptes :

Menu des utilisateurs actifs confirmés :

- L'administrateur peut également afficher la liste des réclamations envoyées par les utilisateurs.
- Ici, l'administrateur répertorie les informations de tous les comptes confirmés (utilisateurs uniquement, pas d'admins).
- Ce menu prend également en charge la pagination.



### ✓ Suspendre un compte d'un utilisateur pour une durée :

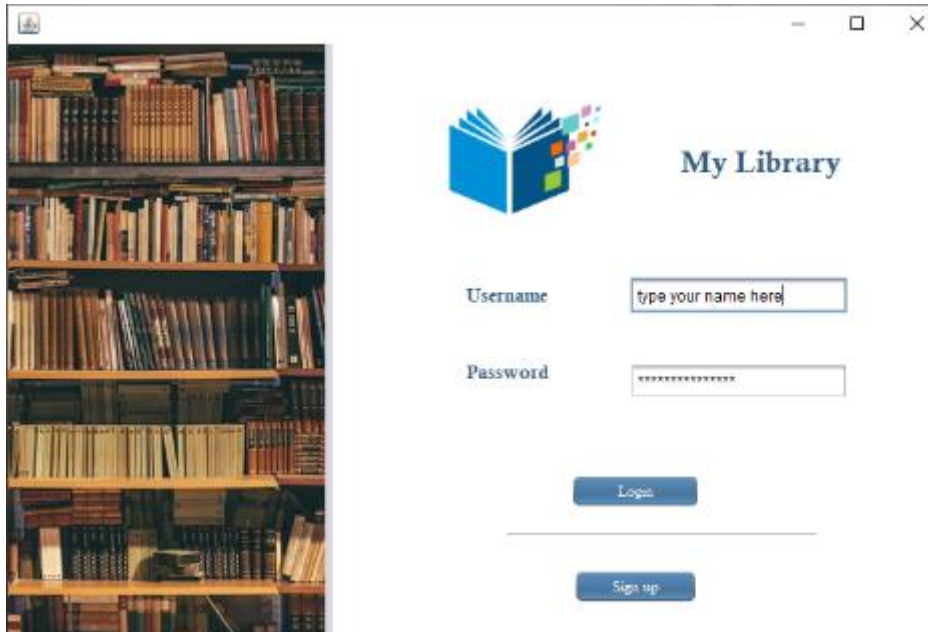
Après avoir sélectionné un utilisateur dans le menu de gestion des comptes, l'administrateur peut également interdire à un utilisateur autant de jours qu'il le souhaite.



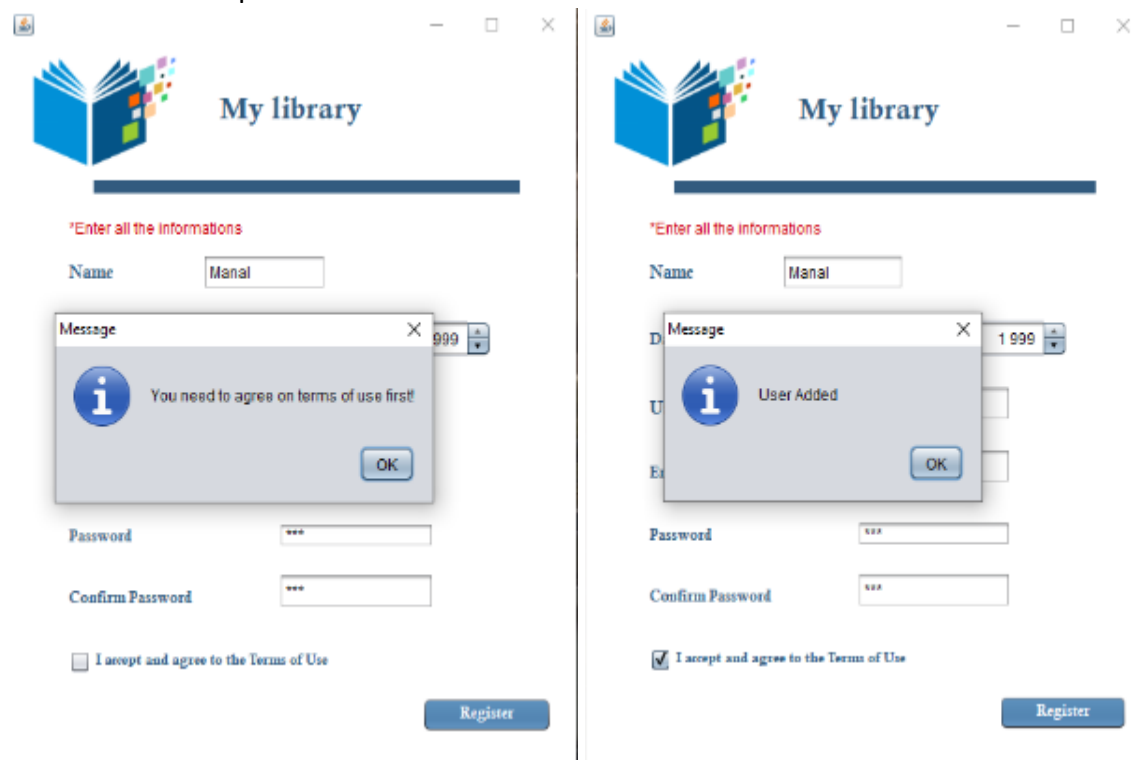
## II. Les fonctionnalités de l'utilisateur (étudiant) dans le système de la bibliothèque :

### 1. Menu de connexion et d'inscription :

Dans le menu principal de connexion, un utilisateur peut utiliser ses informations d'identification pour se connecter à son compte confirmé en tapant sur le bouton **Sign up**.



Une fois terminé, l'enregistrement est fait, les données sont envoyées à l'administrateur pour les confirmer.

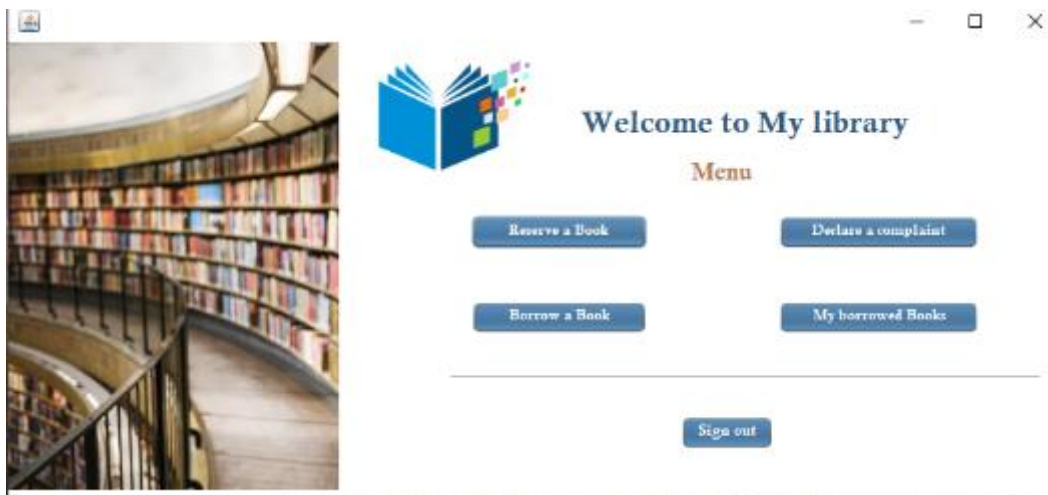


Une fois confirmé, l'utilisateur peut se connecter avec ses nouveaux identifiants.



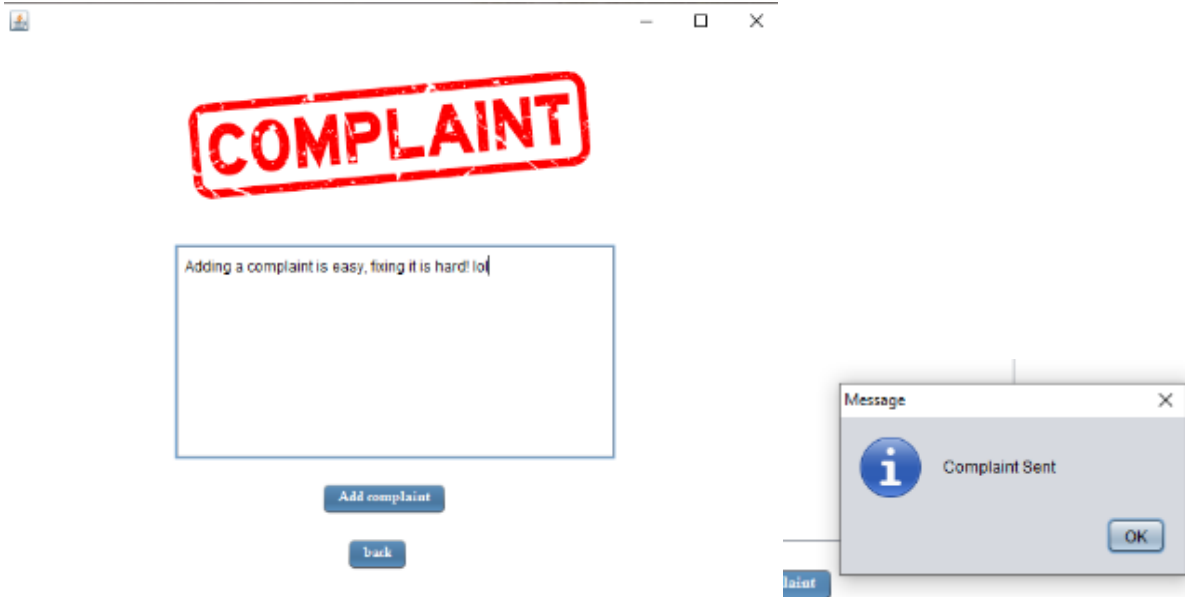
## 2. Menu principale de l'utilisateur (étudiant) :

Il s'agit du menu principal de l'utilisateur. Il peut choisir de réserver, d'emprunter un livre, d'afficher la liste des livres qu'il a emprunté et déposer une réclamation auprès de l'administrateur.



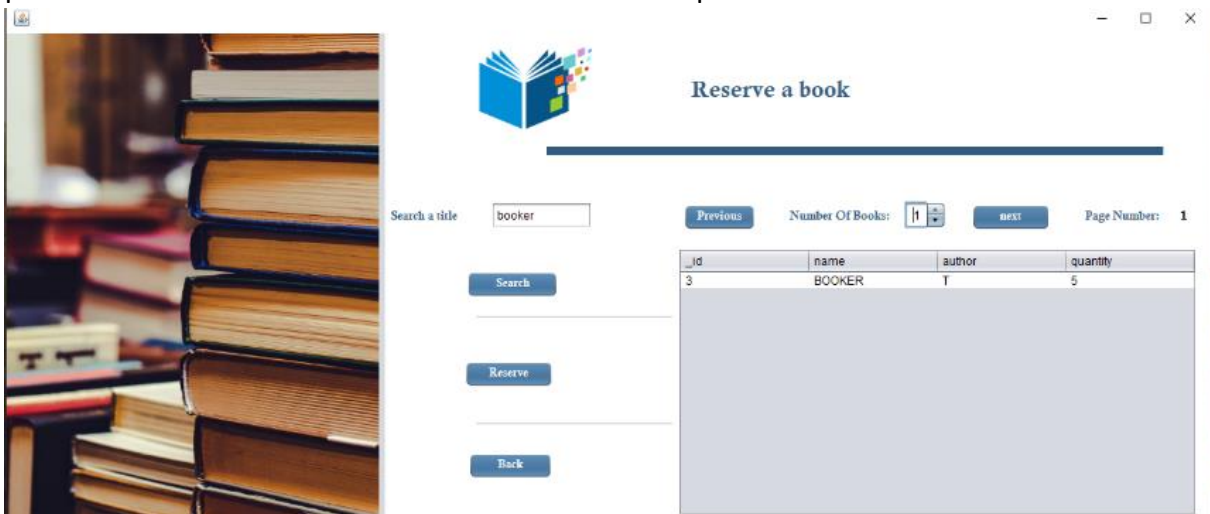
### 3. Envoi une plainte :

Ici, l'utilisateur déclare une réclamation qui sera envoyée à l'administrateur

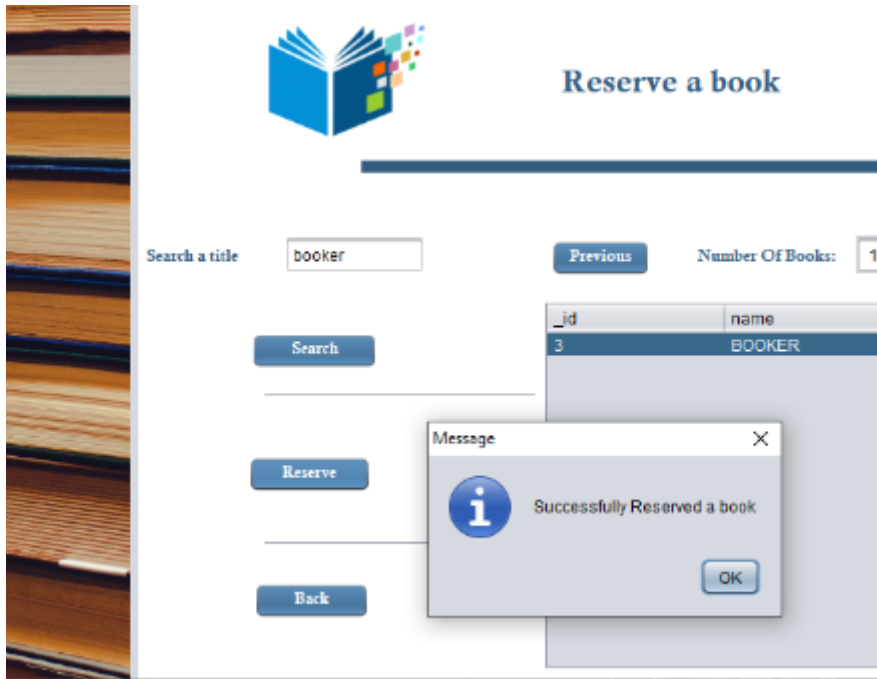


### 4. Menu « Réserver un livre » :

L'utilisateur peut rechercher un livre par son nom, cela utilise une requête SQL, même si l'utilisateur saisit une partie du nom du livre, la base de données sera disponible pour trouver tous les livres de nom similaire ou à proximité.



Ici, l'utilisateur réserve un livre intitulé BOOKER.



#### 5. Menu « Emprunter un livre » :

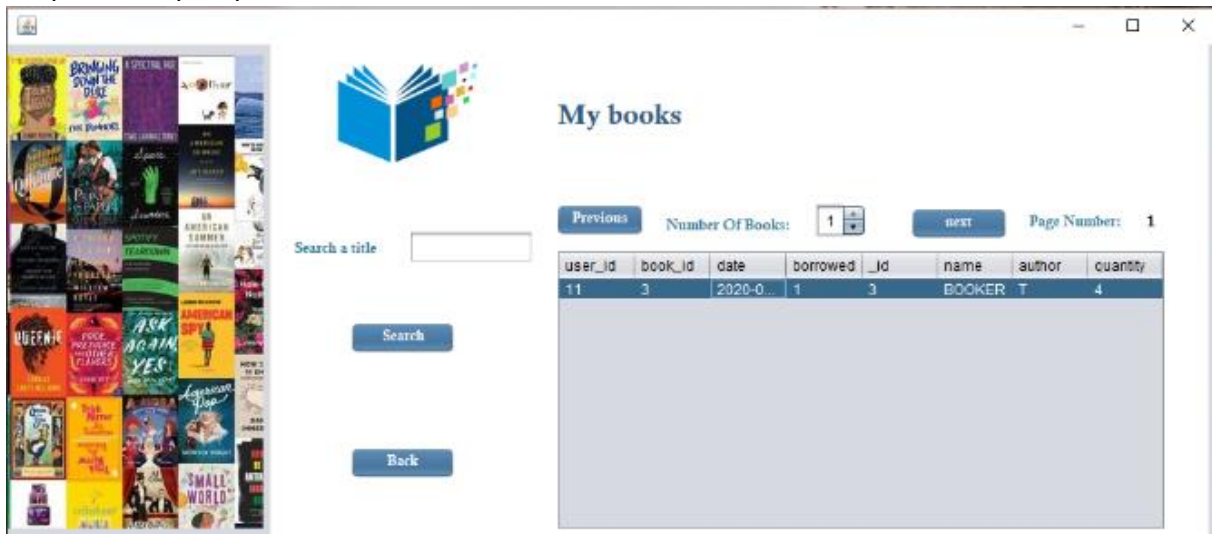
Dans la liste des livres réservés, l'utilisateur peut trouver tous les livres qu'il a réservés, d'ici peut emprunter l'un d'eux.





## 6. Menu des livres empruntés :

Ensuite dans le menu des livres empruntés, l'utilisateur peut trouver les livres empruntés qu'il possède actuellement.



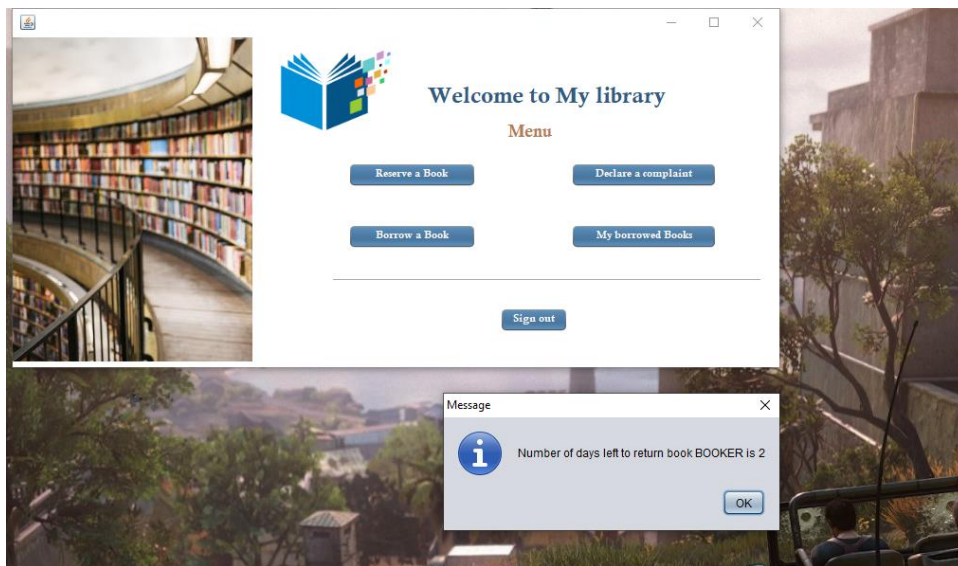
### ✓ Notification pour Rappel :

Ceci est un test de connexion après **1 jour** où le manuel d'utilisation a réservé le livre Booker, Par exemple pour l'étudiante Manal, lorsqu'elle a essayé de se connecter un message de notification (rappel) est reçu.





- Le message de notification avertit l'utilisateur qu'il reste **2 jours** au livre (BOOKER) qu'il a emprunté.



## Conclusion générale

---

La réalisation de ce projet a été une bonne occasion d'une part d'acquérir de nouvelles connaissances et d'autre part, d'assimiler les différents outils acquis durant ce semestre en matière de développement.

La principale mission du projet fut de réaliser la conception d'une application de gestion de bibliothèque qui offre au bibliothécaire un outil pour gérer les informations de la bibliothèque universitaire et aux étudiants un moyen pour faciliter l'utilisation et pour avoir accès à l'information.

La réalisation de l'application s'est déroulée d'une manière itérative. Cela nous a permis d'acquérir les compétences suivantes :

- Modéliser les fonctionnalités avec les cas d'utilisation.
- Représenter les données du système avec un diagramme de classes et séquence
- Utiliser le système de gestion de base de données MySQL pour la définition et la manipulation des données.
- Maîtriser la programmation avec le langage JAVA.