

Département Mathématiques Informatique

Cycle d'Ingénieur

« Ingénierie Informatique : Big Data & Cloud Computing »

RAPPORT DE PROJET D'INNOVATION

Sous thème :

Bibliothèque en ligne



MYLIBRARY

Elaboré par :

EL ASSIMI Ahmed

EL BADAoui Mohammed

YASSINE Faïq

Encadré par :

Mme Souad AHRIZ

Mme Nezha BENMOUSSA

Année universitaire : 2019/2020

Remerciement

Les mots ne peuvent pas exprimer notre gratitude pour nos encadrantes qui nous ont accompagnées pendant tout le chemin vers l'accomplissement de ce projet d'innovation, elles ont donné beaucoup de propositions tout au long de ce projet, leurs conseils et commentaires précieux nous ont permis de surmonter nos difficultés et de progresser dans notre projet.

Ainsi, nous adressons nos extrêmes gratitudee au Staff pédagogique de l'Ecole Normale Supérieure d'Enseignement Technique d'avoir nous donner cette opportunité de tester et pratiquer nos compétences dans ce projet d'innovation. Enfin nous tenons à remercier toute personne qui a contribué de près ou du loin à la réalisation de ce projet.

Sommaire

Introduction Générale	6
Chapitre 1 : Contexte générale	7
I. Description de projet.....	8
1. Nom de projet.....	8
2. Logo de projet.....	8
3. Equipe de projet	8
II. Idée & Contexte de projet.....	9
III. Objectifs.....	9
IV. Répartition des taches.....	9
Chapitre 2 : Etude fonctionnelle	11
Introduction	12
I. Captures des besoins fonctionnels.....	12
1. Etude d'existant.....	12
2. Cahier De charges	12
II. Analyse des besoins	13
1. Identification des acteurs	13
2. <i>Identification des modules</i>	13
3. Diagramme de cas d'utilisation	14
III. Exigences non fonctionnelles	19
1. Les besoins non fonctionnels :	19
Conclusion	20
Chapitre 3 : Etude Conceptuelle et technique du projet	21

Introduction	22
I. Etude Conceptuelle	22
1. Diagrammes des séquences	22
2. Diagrammes de classe	25
3. Diagrammes d'activité	31
II. Etude technique de projet	33
Conclusion	33
Chapitre 4 : Réalisation et outils de développement.....	34
Introduction	35
I. Outils de développement.....	35
1. Langages utilisés	35
2. Technologies utilisées.....	36
3. Outils utilisés.....	38
II. Réalisation	40
1. Espace administrateur	40
2. Espace Lecteur	46
Conclusion générale	60
Tableau de figure.....	61
Bibliographie	64

Introduction Générale

Dans le cadre de notre formation à l'École Normale Supérieure d'Enseignement Technique nous avons eu l'occasion de traiter et confronter des méthodes et techniques enseignées avec les pratiques en vigueur dans un projet d'innovation.

Dans le monde de la digitalisation, la lecture des livres devient une habitude traditionnelle, et les livres deviennent isolés du monde actuel, avec l'apparition des nouveaux smartphones qui offrent des ressources en ligne accessibles dans tout le temps, les étudiants préfèrent beaucoup plus l'utilisation des smartphones au lieu des papiers tangibles.

Les bibliothèques universitaires posent maintenant beaucoup de problèmes de gestion et de rendre les étudiants satisfaits par leurs services, les étudiants ne ressent pas la facilité d'accès et la disponibilité des services.

L'informatique donne aux utilisateurs de nouveaux outils de travail et leur permet d'améliorer leur rentabilité et leur productivité. En effet les applications informatiques et le stockage des informations dans des bases de données offrent de nombreux avantages grâce à des processus administratifs plus rapides et presque sans erreurs, ce qui contribue à améliorer et faciliter la qualité des services, aussi les utilisateurs bénéficient d'une meilleure sécurité et d'une plus grande autonomie.

Dans ce sens les universités et les écoles se lancent pour produire des applications mobiles et web pour fournir les ressources électroniques pour faciliter le travail.

C'est dans cette perspective que s'inscrit notre projet d'innovation sous thème : « Développement d'une bibliothèque en ligne ».

Dans ce rapport nous allons aborder premièrement le contexte général du projet Ensuite l'analyse fonctionnelle et après l'analyse conceptuelle, et on va terminer par la réalisation du projet.

Chapitre 1 : Contexte générale

I. Description de projet

1. Nom de projet

Nous avons choisi **MYLIBRARY** comme nom de notre projet.

2. Logo de projet



3. Equipe de projet

Trois élèves Ingénieurs, en 2ème année de la Filière Ingénierie Informatique : Big Data et Cloud Computing (II-BDCC), à l'École Normale Supérieure de l'Enseignement Technique de Mohammed VI (ENSET-M). Motivés, dynamiques et ambitieux.



Mohammed EL BADAOUI

Elève Ingénieur



Yassine FAIQ

Elève Ingénieur



Ahmed EL ASSIMI

Elève Ingénieur

II. Idée & Contexte de projet

Nous, comme étant des élèves ingénieurs à l'Ecole Normale Supérieure d'enseignement technique nous sommes amenés à réaliser un projet d'innovation.

Dans ce sens notre idée vient après une étude sur les services offerts par notre école et nous avons sortis par l'idée de créer une plateforme en ligne qui sera dédiés aux ressources électroniques spécifiquement des livres, parce qu'au niveau de notre université nous disposons de plusieurs services mais il n'y a pas de satisfaction de notre part, nous les étudiants.

Alors nous devons innover une solution intelligente qui vas satisfaire nos besoins et au même temps qui va encourager les étudiants et les professeurs à mieux exploiter et utiliser ces ressources.

III. Objectifs

Dès la fixation de notre projet, nos objectifs ont été très clairs :

La plateforme doit respecter les normes suivantes :

- Facilité d'utilisation
- Utilité des ressources
- Accessibilité dans tout le temps
- Offrir des nouvelles fonctionnalités
- Compatibilité avec tous les appareils
- Maintenabilité
- Expérience utilisateur (USER EXPERIENCE) simple
- Design convenable et confortable

IV. Répartition des taches

Membre

Taches

Ahmed EL ASSIMI

Développement de la partie Web :

- Espace utilisateur
- Espace Administrateur

*Mohammed EL
BADAoui*

Développement de la partie Mobile :

- Espace utilisateur

Yassine FAIQ

Développement des outils de lecture des livres au niveau Web et Mobile

Chapitre 2 : Etude fonctionnelle

Introduction

Dans ce chapitre on va aborder une analyse fonctionnelle sur le projet afin de dégager les points de faiblesse de l'existant, et ensuite nous allons définir notre cahier de charges qui vas décrire un projet dans une approche synthétique et on va faire une étude sur les besoins fonctionnels et non fonctionnels et les acteurs qui vont intervenir dans le projet.

I. Captures des besoins fonctionnels

1. Etude d'existant

Les bibliothèques traditionnelles ont vécu après la révolution technologique une baisse au niveau des Utilisateurs dans la plupart des universités et écoles.

Ces bibliothèques posent beaucoup de points de faiblesse :

- La bibliothèque n'est pas ouverte 24h/24, 7j/7.
- Certains étudiants ne préfèrent pas se rendre à chaque fois jusqu'à l'école pour emprunter un livre.
- Il n'y a pas toujours la disponibilité de quelques livres
- Difficulté d'accès et d'emprunter (procédure prend beaucoup de temps)
- La qualité des livres s'affecte avec le temps s'elle n'était pas prise en charge

Pourtant Il y'avait des solutions numériques offertes par des universités comme :

- ScholarVox
- Bibliotheque Mohammed Sekkat

Mais apparemment ces solutions n'étaient pas en faveur des étudiants à cause des procédures d'inscription et aussi ils ne sont pas adaptés à toutes les appareils.

Alors nous devons trouver des solutions améliorées et qui vas offrir plusieurs avantages aux écoles :

- Réduire les coûts d'exploitation
- Suivre les étudiants
- Améliorer l'expérience utilisateur
- Disponibilité des ressources dans n'importe quel moment.
- Assurer une traçabilité sur un système d'information
- Avoir des feedbacks de la part des étudiants.

2. Cahier De charges

- **Définition de projet** : une bibliothèque en ligne qui permet la lecture des livres, gestion des utilisateurs, ressources électroniques
- **Objectifs du projet** :
 - Offrir un espace de lecture
 - Diversité des ressources électroniques

- Gestion des utilisateurs
- Gestion des livres
- Gestion des interactions des utilisateurs
- Espace d'inscription
- Recommandation des livres
- Evaluation des livres

II. Analyse des besoins

1. Identification des acteurs

Un acteur représente l'abstraction d'un rôle joué par des entités externes. Dans notre projet on distingue principalement deux acteurs qui sont les suivants :

- Administrateur
- Utilisateur

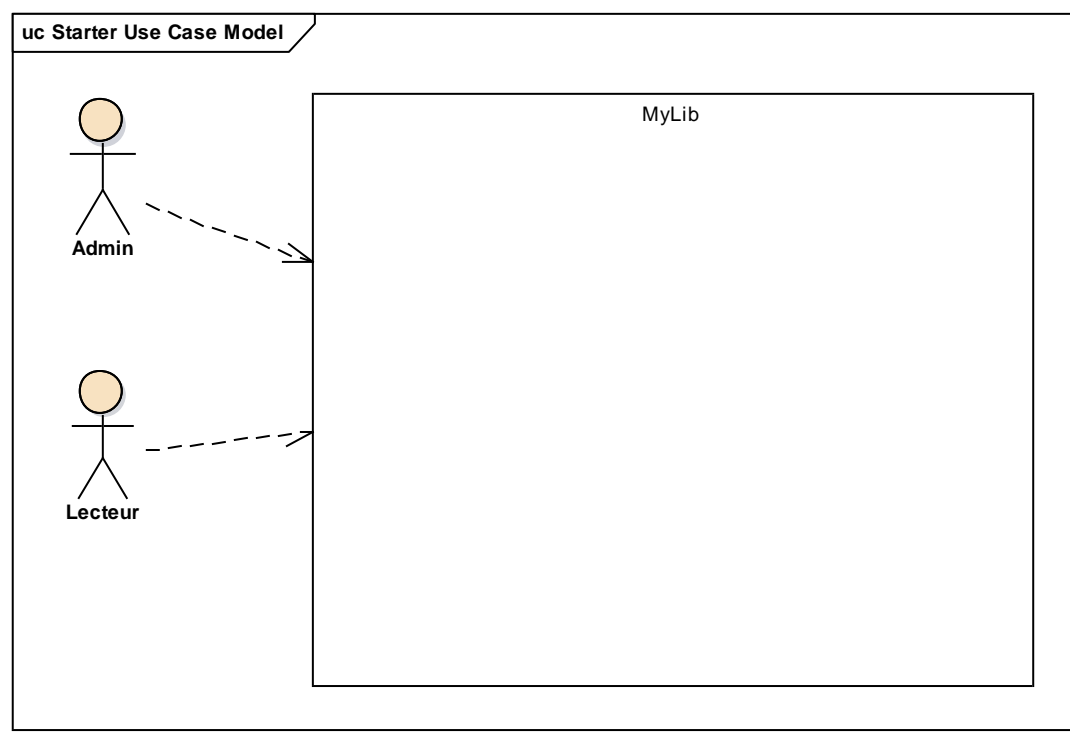


Figure 1:diagramme de contexte

2. Identification des modules

Pour bien reprendre au besoin de notre projet, nous avons diviser notre application a des diffèrent modules :

- Gestion des livres.

- Gestion des utilisateurs
- Gestion des catégories
- Gestion des profils
- Gestion des abonnements
- Gestion des réclamations

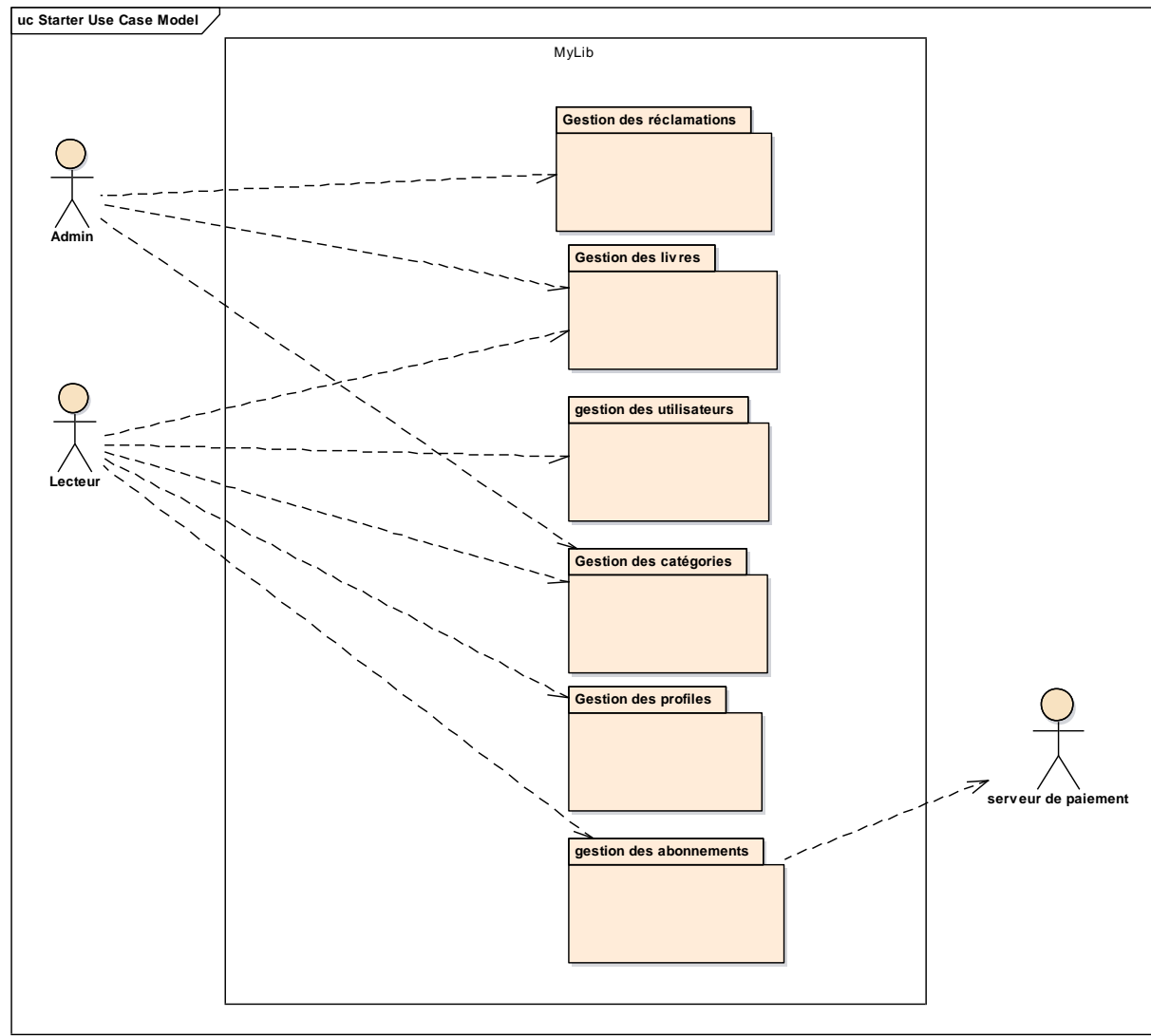


Figure 2 : diagramme de package

3. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation décrit l'interdépendance entre le système et l'acteur en déterminant les besoins de l'utilisateur et tout ce que doit faire le système pour l'acteur

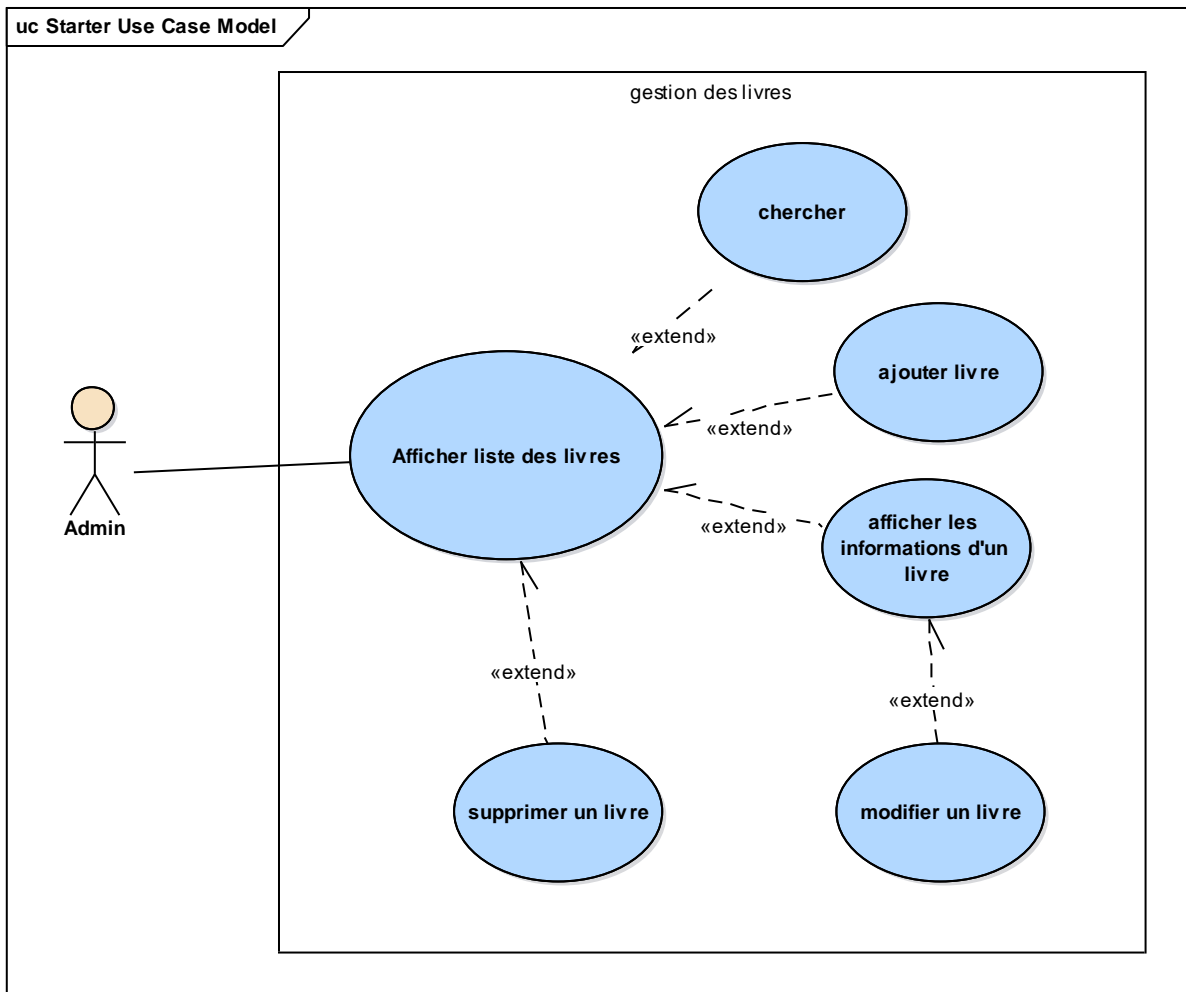


Figure 3 : diagramme de cas d'utilisation "gestion des livres " (admin)

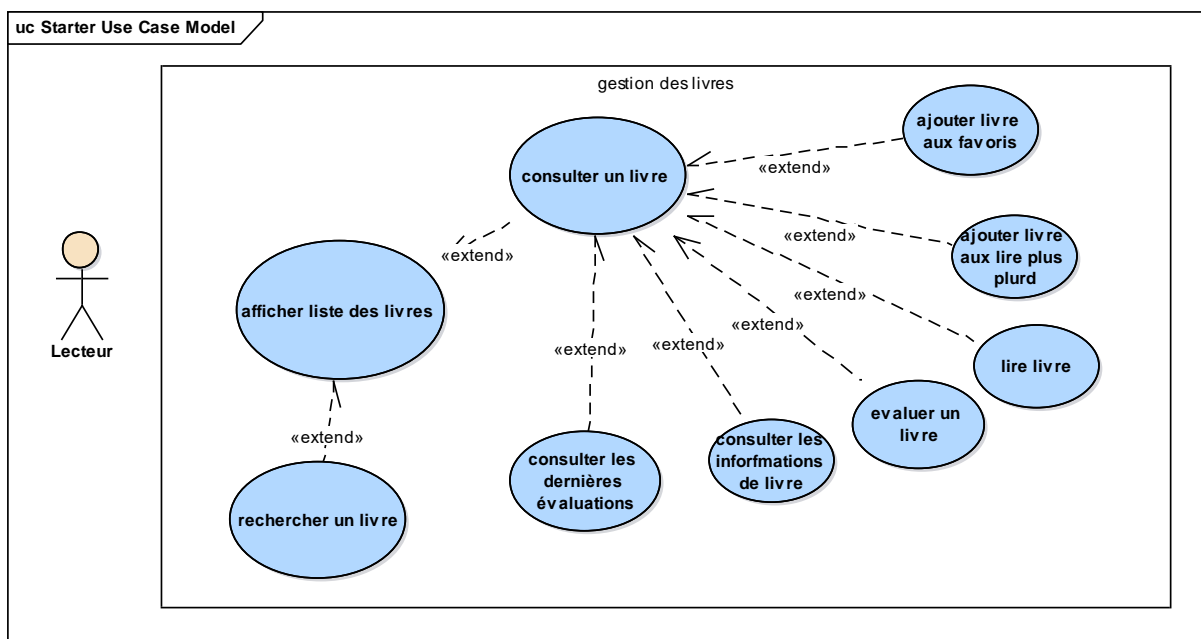


Figure 4 : diagramme de cas d'utilisation "gestion des livres " (lecteur)

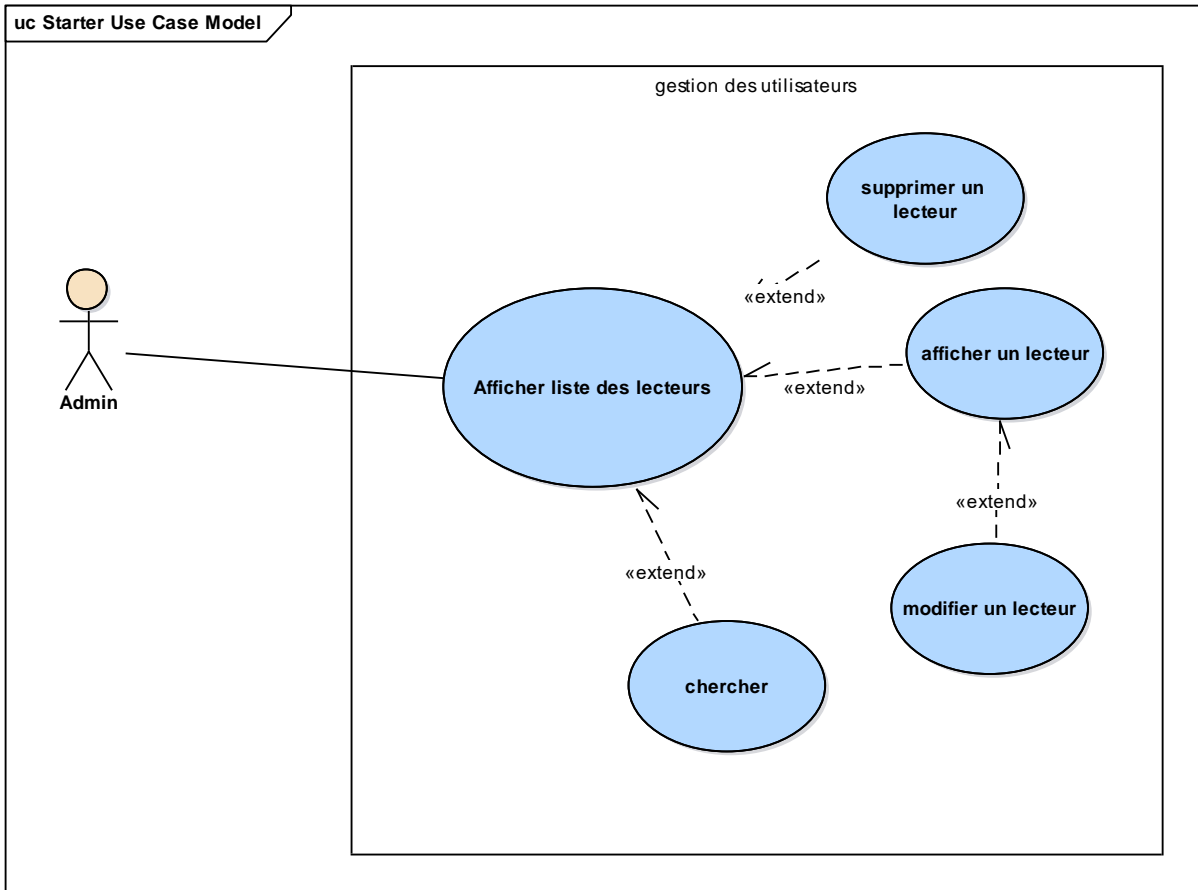


Figure 5 : diagramme de cas d'utilisation "gestion des utilisateurs " (admin)

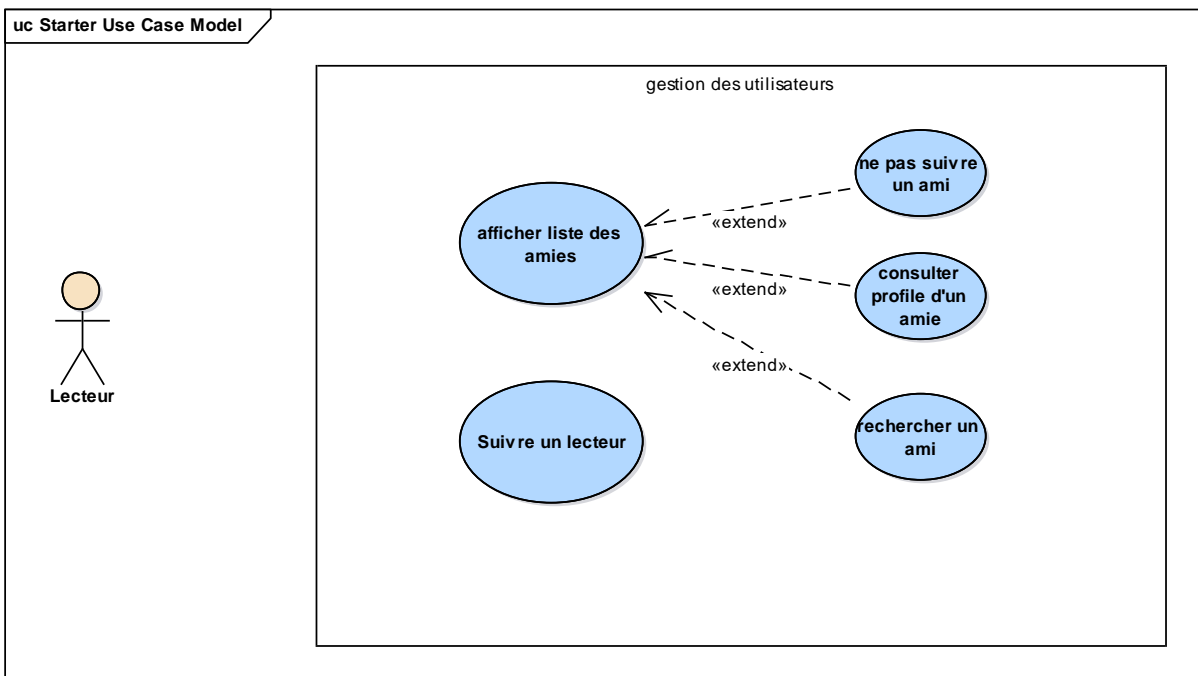


Figure 6: diagramme de cas d'utilisation "gestion des utilisateurs " (lecteur)

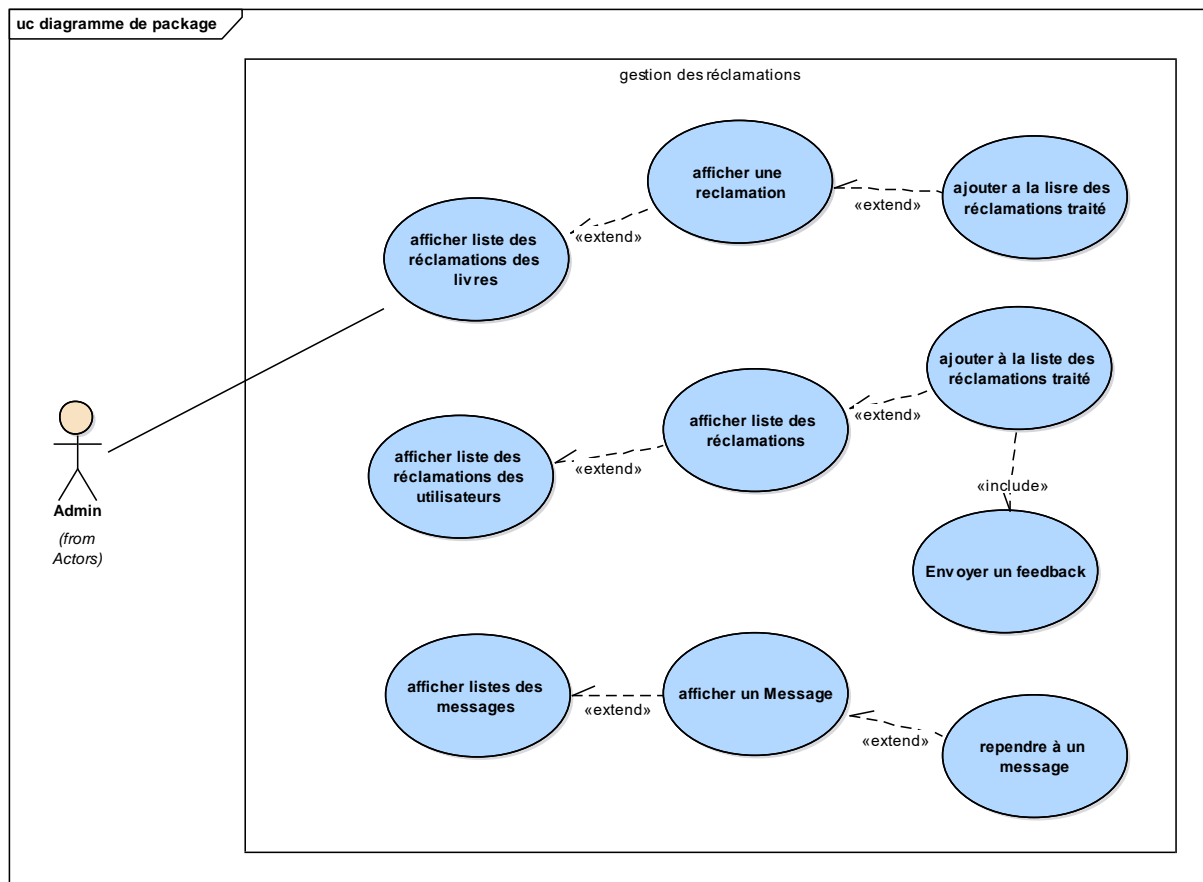


Figure 7 : diagramme de cas d'utilisation "gestion des réclamations " (admin)

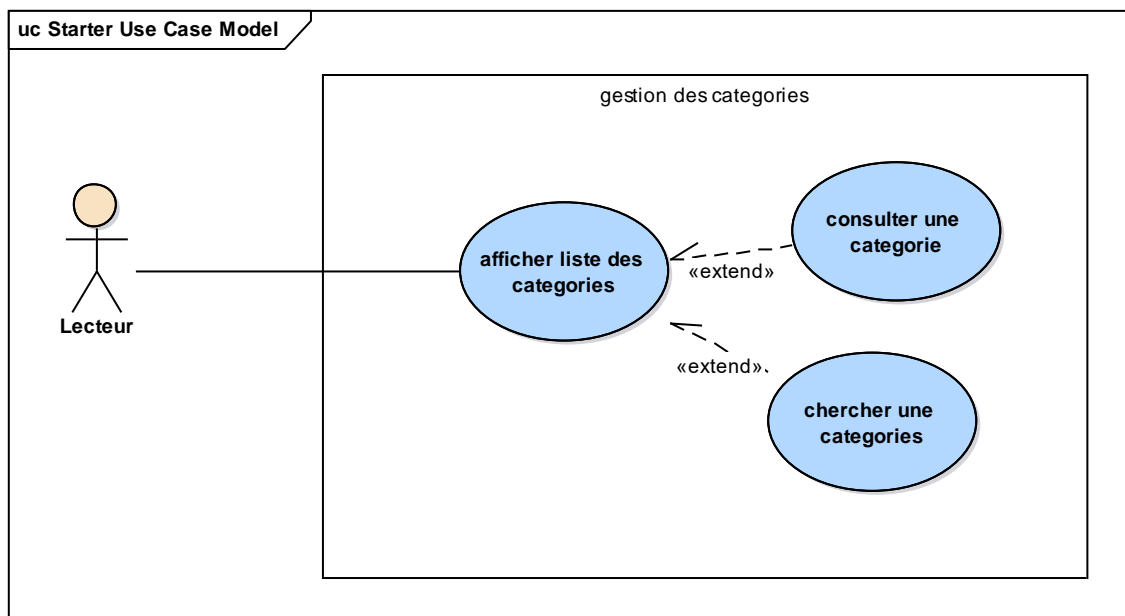


Figure 8 : diagramme de cas d'utilisation "gestion des catégories " (lecteur)

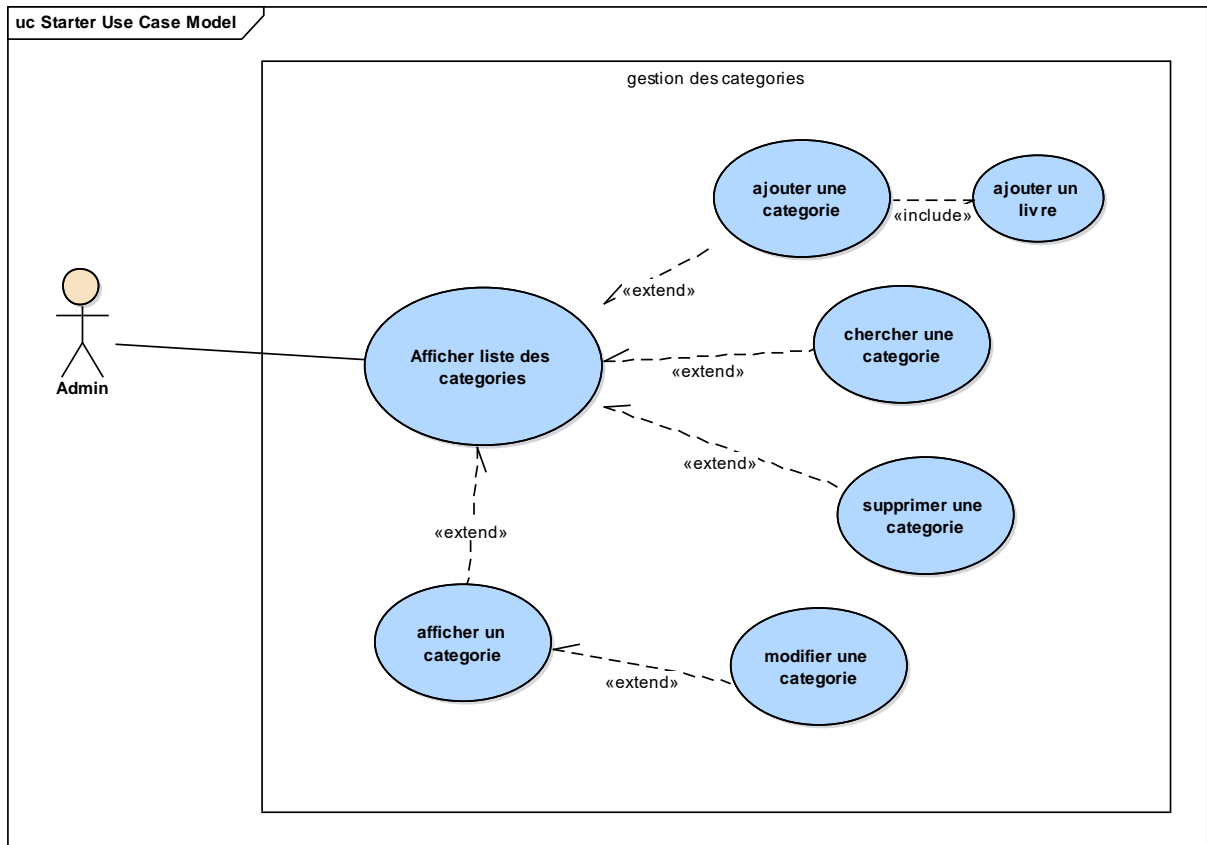


Figure 9 : diagramme de cas d'utilisation "gestion des catégories " (admin)

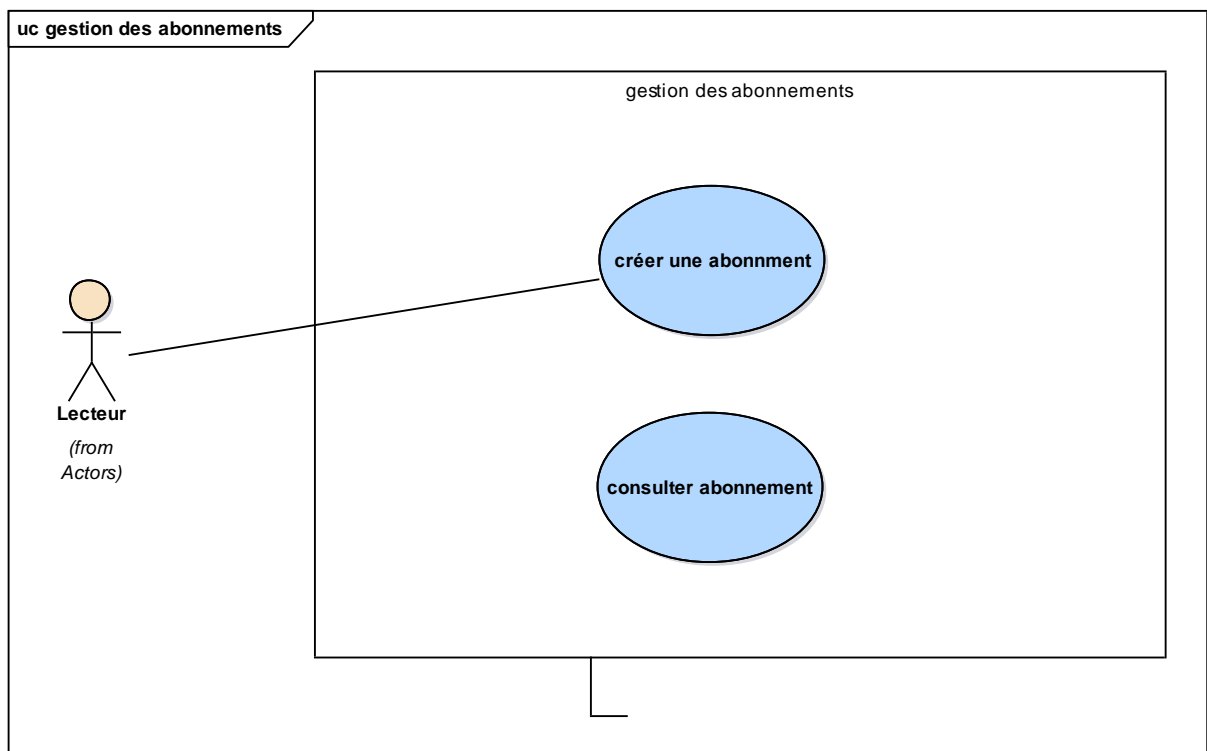


Figure 10 : diagramme de cas d'utilisation "gestion des abonnements "

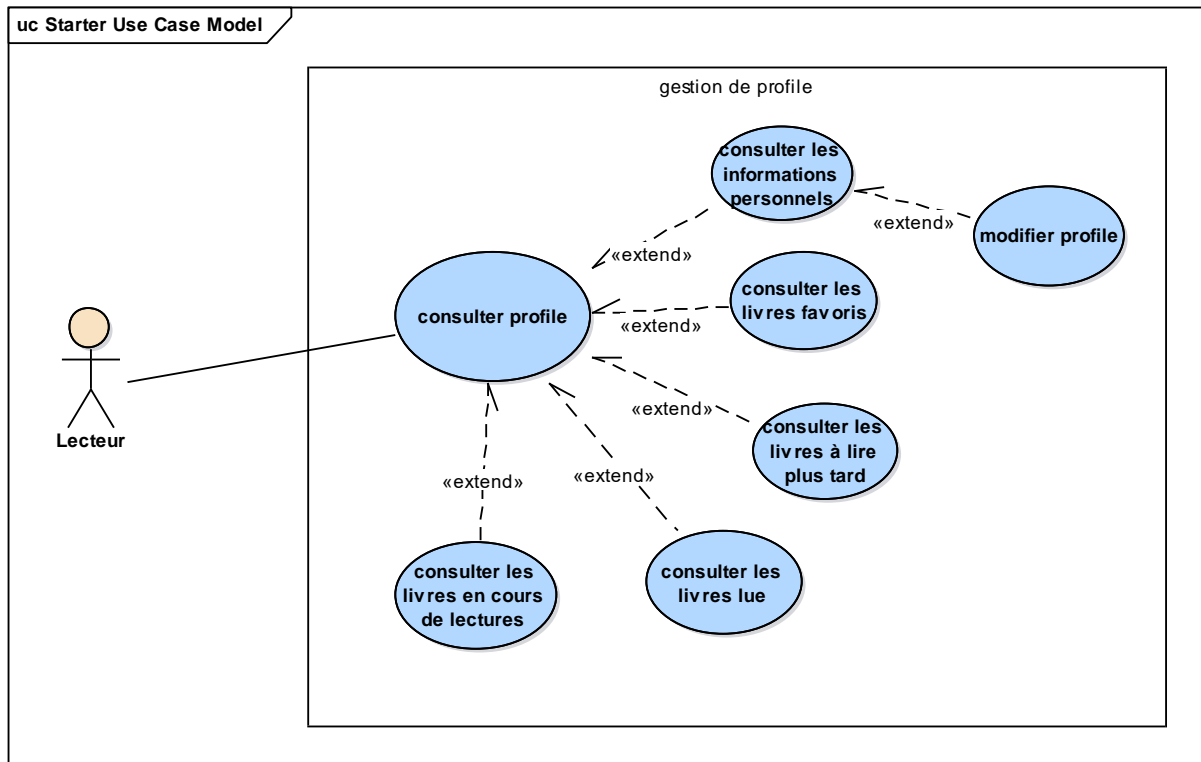


Figure 11 : diagramme de cas d'utilisation "gestion de profile "

III. Exigences non fonctionnelles

1. Les besoins non fonctionnels :

Les besoins non fonctionnels sont importants car ils agissent de façon indirecte sur le résultat et sur le rendement de l'utilisateur, ce qui fait qu'ils ne doivent pas être négligés, pour cela il faut répondre aux exigences suivantes :

Fiabilité : L'application doit fonctionner d'une façon cohérente sans erreurs et doit être satisfaisante.

Ergonomie et bonne Interface : L'application doit être adaptée à l'utilisateur sans qu'il ne fournisse aucun effort (utilisation claire et facile) de point de vue navigation entre les différentes pages, couleurs et mise en textes utilisés.

Sécurité : Notre solution doit respecter surtout la confidentialité des données personnelles des clients qui reste l'une des contraintes les plus importantes dans les sites web.

Aptitude à la maintenance et la réutilisation : Le système doit être conforme à une architecture standard et claire permettant sa maintenance et sa réutilisation.

Compatibilité et portabilité : Un site web quel que soit son domaine, son éditeur et son langage de programmation ne peut être fiable qu'avec une compatibilité avec tous les navigateurs web et tous les moyens que ce soit PC, IPAD ou Mobiles.

Conclusion

Après avoir fait l'analyse des besoins fonctionnels et non fonctionnels et l'étude de l'existant et ensuite les maquettes qu'on doit suivre, on passe au chapitre suivant qui va représenter l'étude conceptuelle du projet.

Chapitre 3 : Etude Conceptuelle et technique du projet

Introduction

Après avoir cité les différents besoins techniques et fonctionnels de notre projet, on va commencer maintenant une phase très importante et indispensable dans le cycle de vie d'une application, Cette phase est la conception qui a pour but d'expliquer le déroulement de notre application ainsi qu'assurer une bonne compréhension des besoins des utilisateurs.

I. Etude Conceptuelle

1. Diagrammes des séquences

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation Unified Modeling Language.

Puisque nous avons plusieurs cas d'utilisation on va s'intéresser dans le diagramme de séquence sur trois cas d'utilisation :

- « S'authentifier »
- « Ajouter une catégorie »
- « Ajouter un livre »

a. Diagramme de séquence : cas d'utilisation « S'authentifier »

Un utilisateur doit entrer les données d'authentification, après que ces données soient vérifiées et validées, le système redirige l'utilisateur vers l'espace de lecture, s'il est un lecteur, sinon redirection vers l'espace admin (Figure 12)

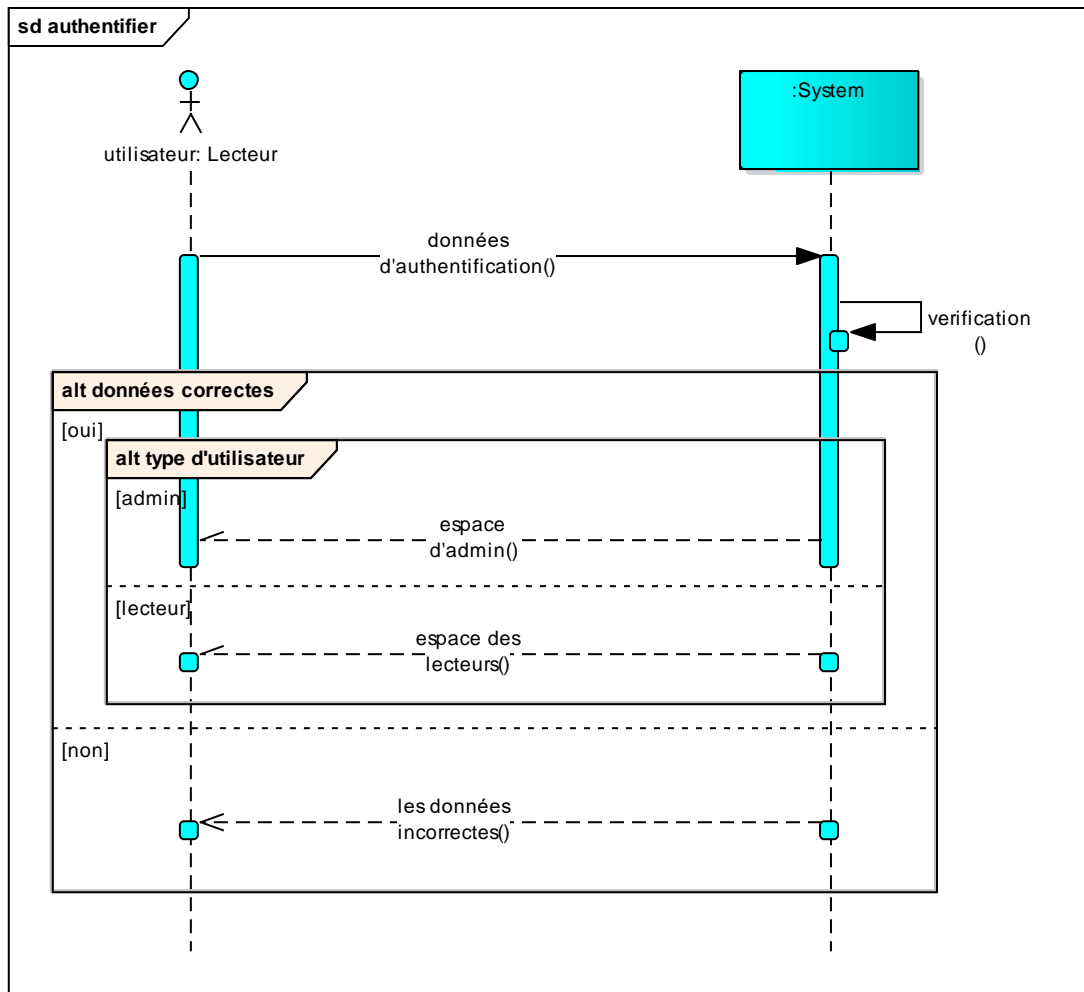


Figure 12 : diagramme de séquence de cas d'utilisation « « authentifier »

b. Diagramme de séquence : cas d'utilisation « Ajouter un livre »

Un utilisateur doit être authentifié au tant qu'un administrateur pour qu'il puisse ajouter un livre. Pour ajouter un livre, l'administrateur doit entrer les données d'un livre. Après que ces données soient vérifiées et validées au niveau de système, le livre est ajouté. (Figure 13)

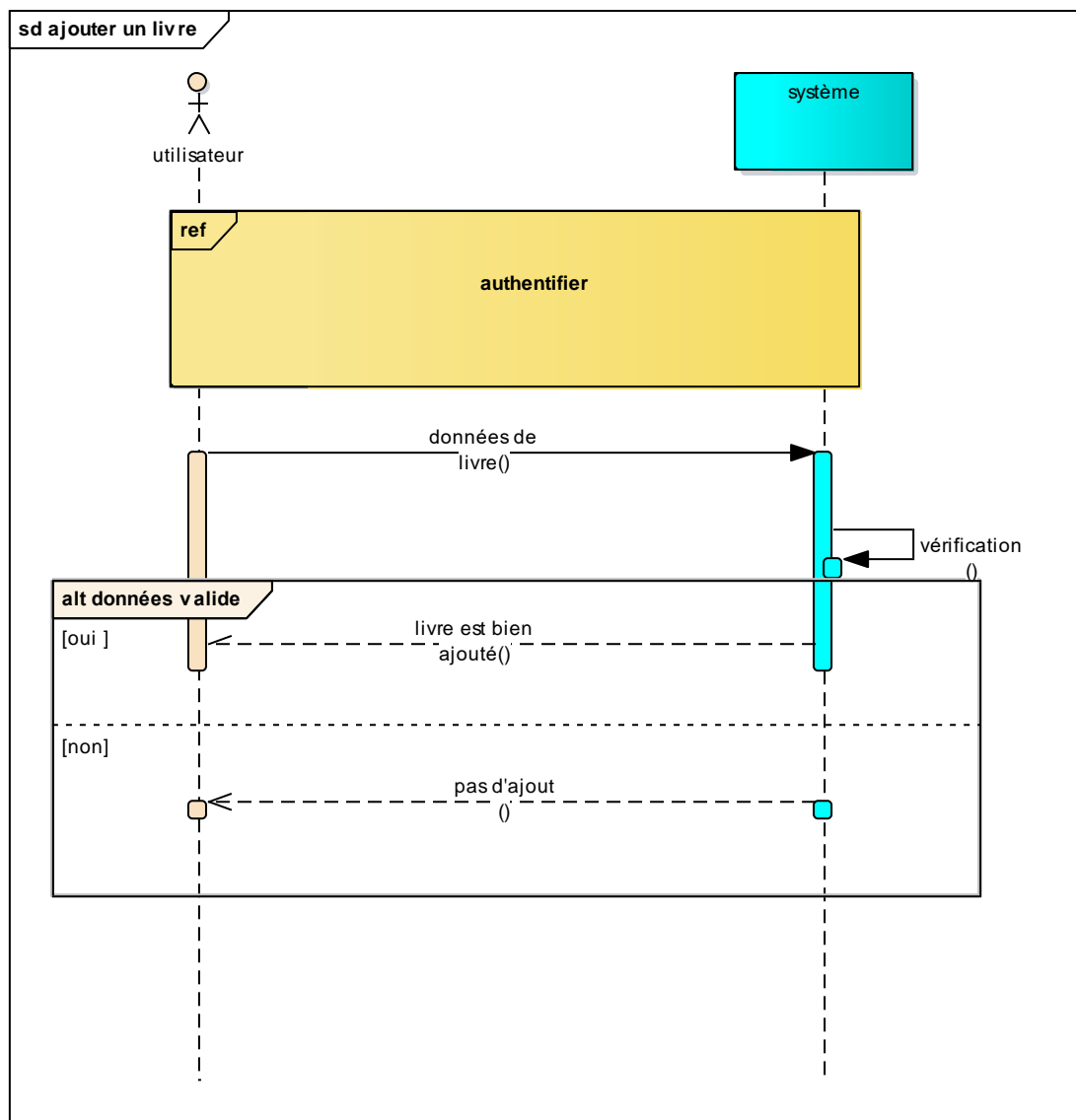


Figure 13 : Diagramme de séquence : cas d'utilisation « Ajouter un livre »

c. Diagramme de séquence : cas d'utilisation « Ajouter une catégorie »

Un utilisateur doit être authentifié au tant qu'un administrateur pour qu'il puisse ajouter une catégorie. Pour ajouter une catégorie, l'administrateur doit entrer les données de catégorie, le système vérifie que la catégorie n'existe pas. Par la suite, tant que la catégorie n'existe pas l'utilisateur doit ajouter un livre de cette catégorie. En fin, l'ajout de catégorie et de livre est effectué.

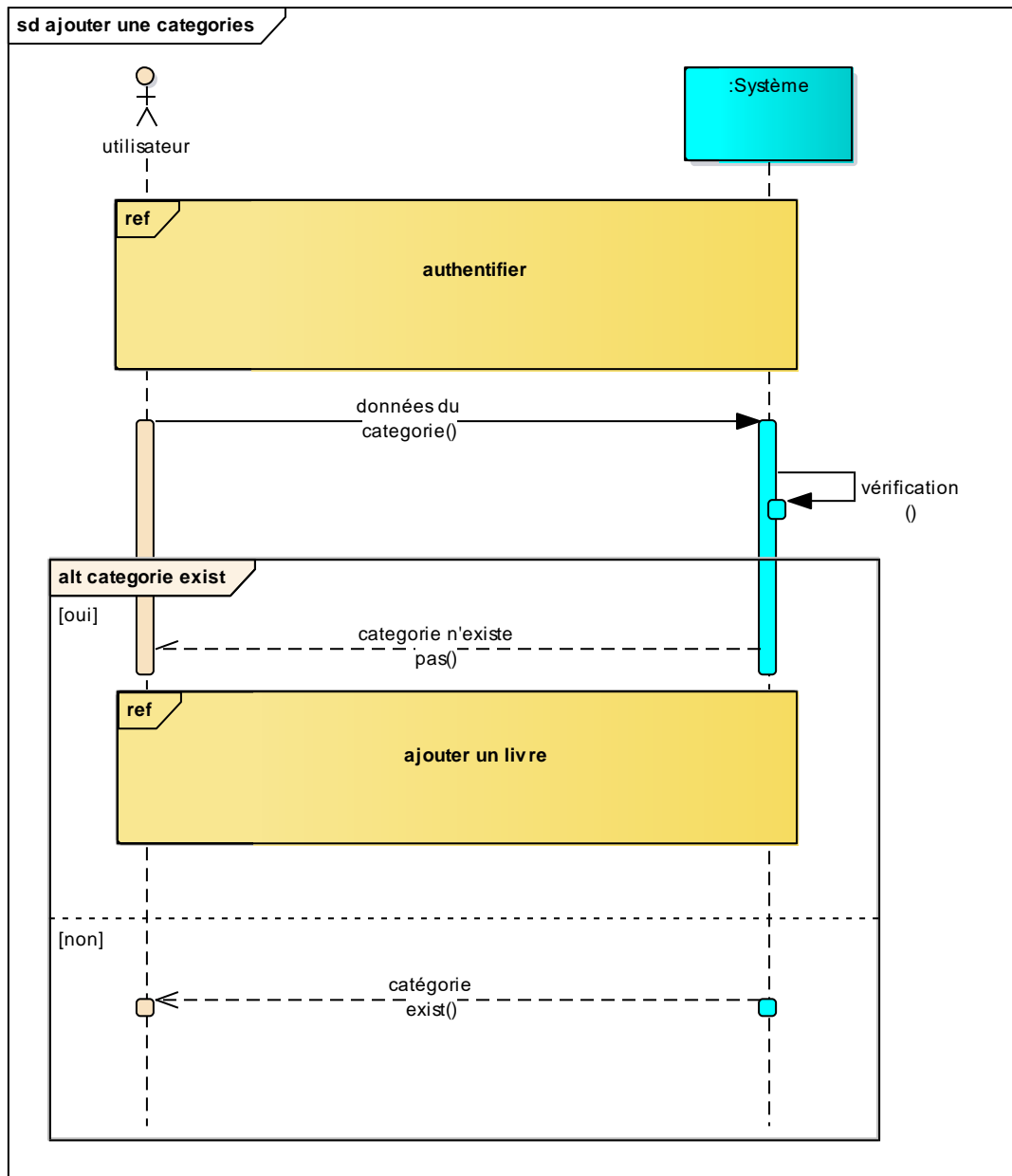


Figure 14 : Diagramme de séquence : cas d'utilisation « Ajouter une catégorie »

2. Diagrammes de classe

Le diagramme de classes est un schéma utilisé en génie logiciel 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.

Notre projet est composé des différents modules (figure diagramme de paquetage), alors Nous avons divisé le diagramme de classe générale à plusieurs sous diagrammes de classe liés

et continues, pour avoir une proche vision et une lisibilité à comprendre les différents classes qui interagissent dans chaque module de notre projet.

a. Diagramme de classe : le module de gestion des utilisateurs

Un utilisateur de notre application est une personne qui à un nom, prénom, email, age, date de naissance, téléphone

Un utilisateur peut être administrateur ou un lecteur

Un lecteur peut être un lecteur invité, un lecteur Enfant ou un lecteur Ultimate(accès globale)

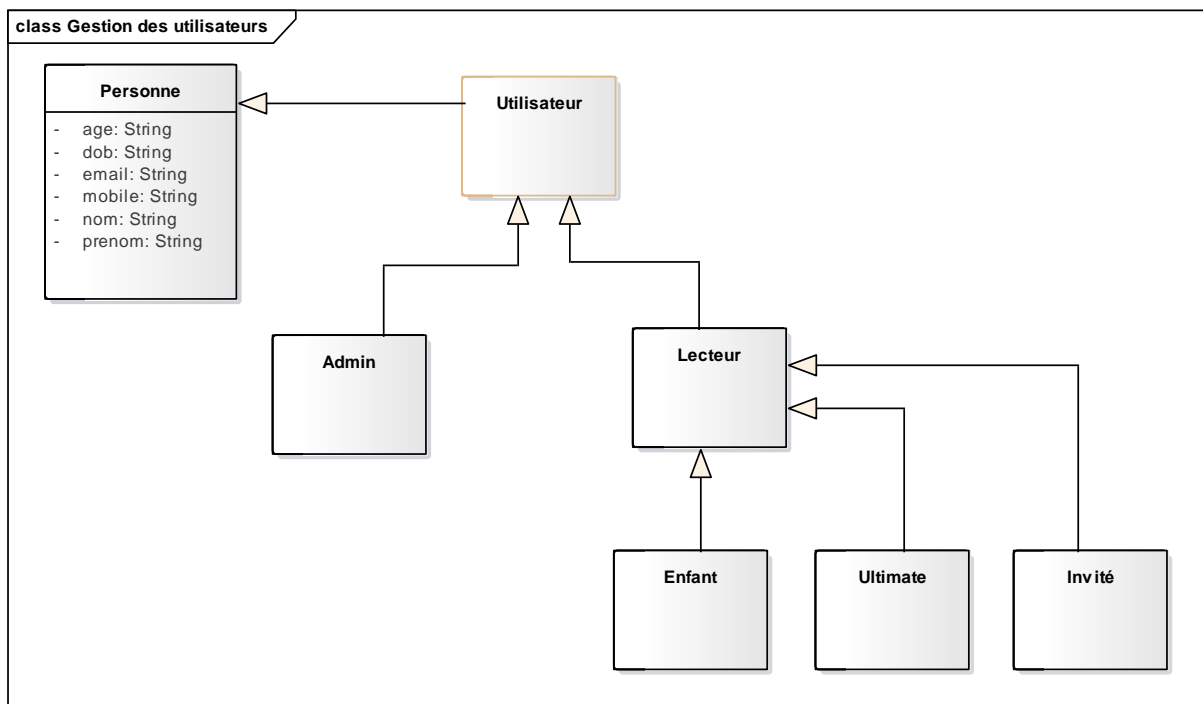


Figure 15 : Diagramme de classe : le module de gestion des utilisateurs

b. Diagramme de classe : le module de gestion de profil

Un utilisateur à un profil qui contient une image, adresse, genre, et login

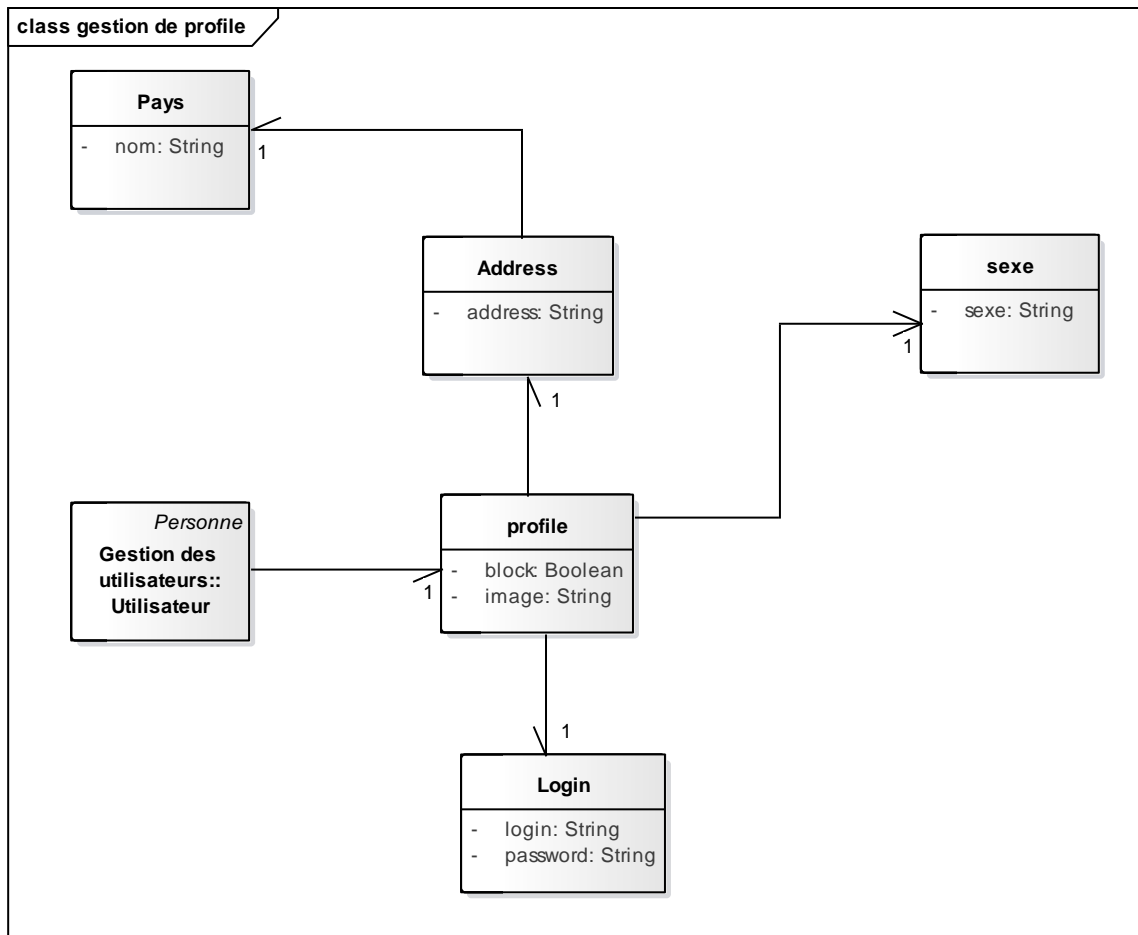


Figure 16: Diagramme de classe : le module de gestion de profil

c. Diagramme de classe : le module de gestion des Livres

Un livre a un titre, description, image, ISBN, nombre des pages, langue et une évaluation.

Un livre a au moins un auteur qui un nom prénom et date de naissance.

Un livre a au moins publication qui a une date de publication et publieur.

Un lecteur peut ajouter un livre à un panier spécifique (livre favoris, livre à lire plus tard, les livres lus, les livres en cours de lecture)

Un lecteur peut évaluer un livre.

Chaque évaluation a une date, l'évaluation et un commentaire

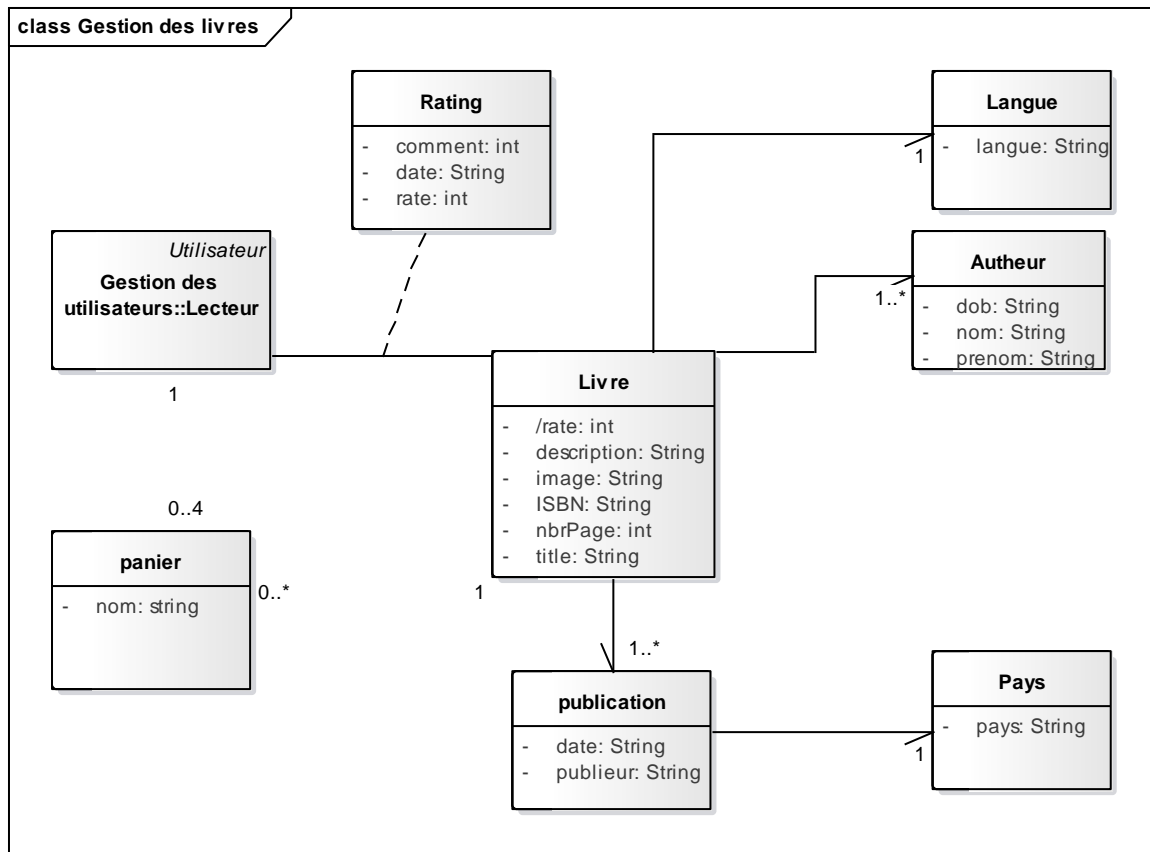


Figure 17 : Diagramme de classe : le module de gestion des Livres

d. Diagramme de classe : le module de gestion des catégories

Une catégorie a son nom et se compose de plusieurs livres.

Une catégorie peut appartenir à une ou plusieurs cible (dans notre application nous avons spécifié pour chaque type de lecteur les types des livres que nous devons lui afficher « Enfant », « Ultimate », « invité »).

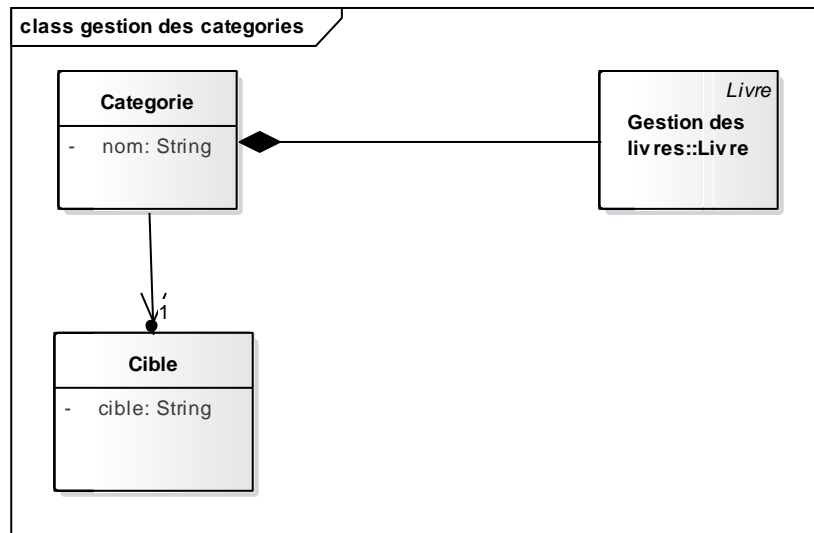


Figure 18 : Diagramme de classe : le module de gestion des catégories

e. Diagramme de classe : le module de gestion des réclamations

Une réclamation a une date et un message

Une réclamation peut être une réclamation sur un livre, une réclamation sur un utilisateur ou un message

Un lecteur fait une réclamation

Un admin traite une réclamation

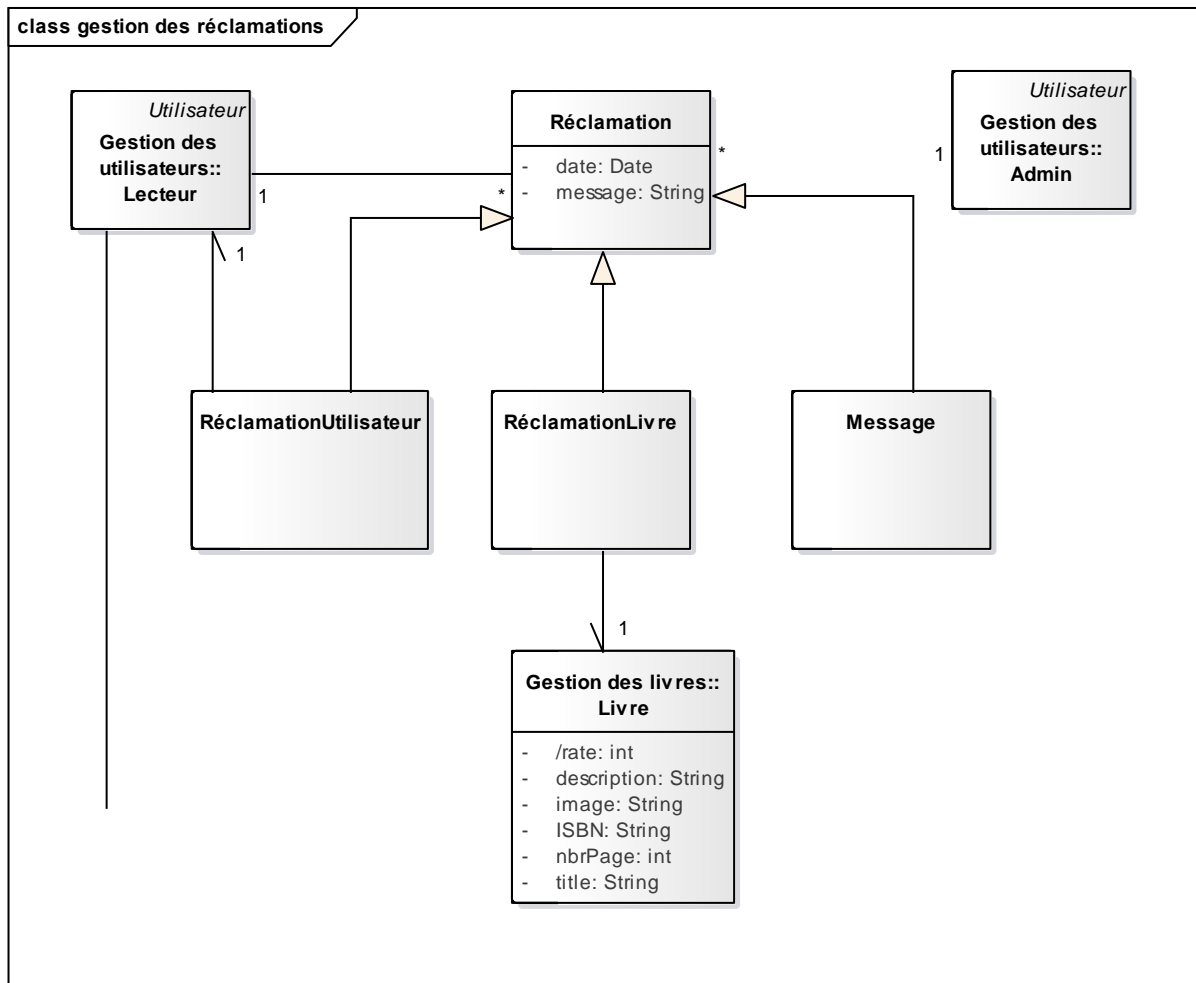


Figure 19 : Diagramme de classe : le module de gestion des réclamations

f. Diagramme de classe : le module de gestion des abonnements

Un lecteur effectue plusieurs abonnements.

Un abonnement a un type, date de début, date de fin, durée et un paiement.

Un paiement a une date de paiement et tarif

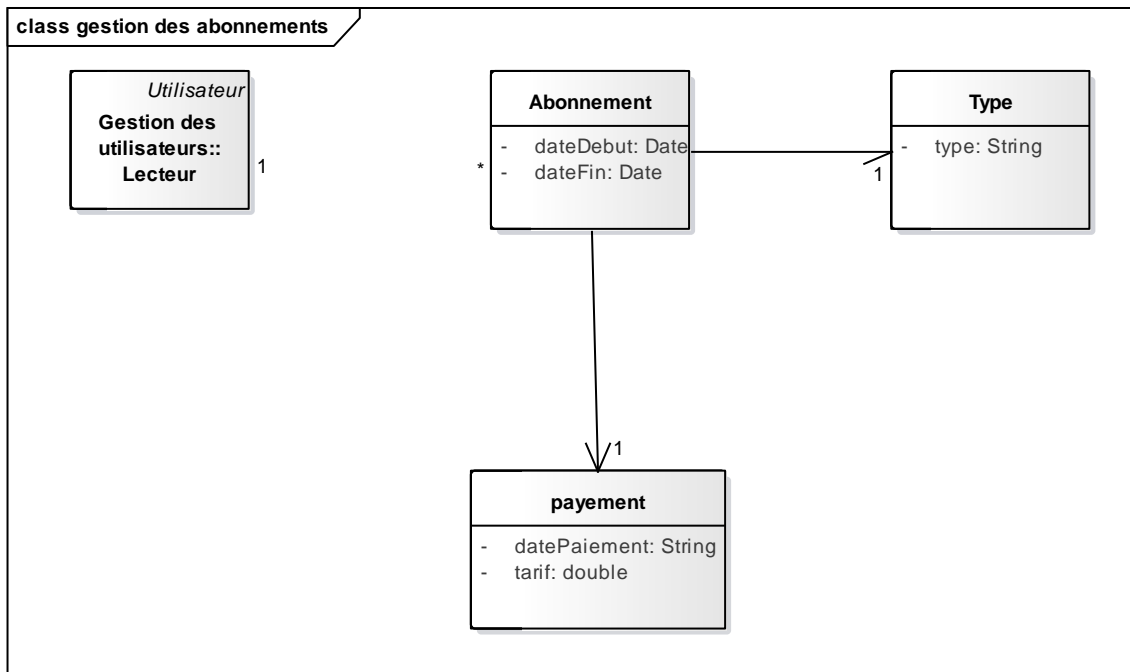


Figure 20 : Diagramme de classe : le module de gestion des abonnements

3. Diagrammes d'activité

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

Puisque nous avons plusieurs cas d'utilisation on va s'intéresser dans le diagramme d'activité sur deux cas « s'authentifier » et « ajouter un livre ».

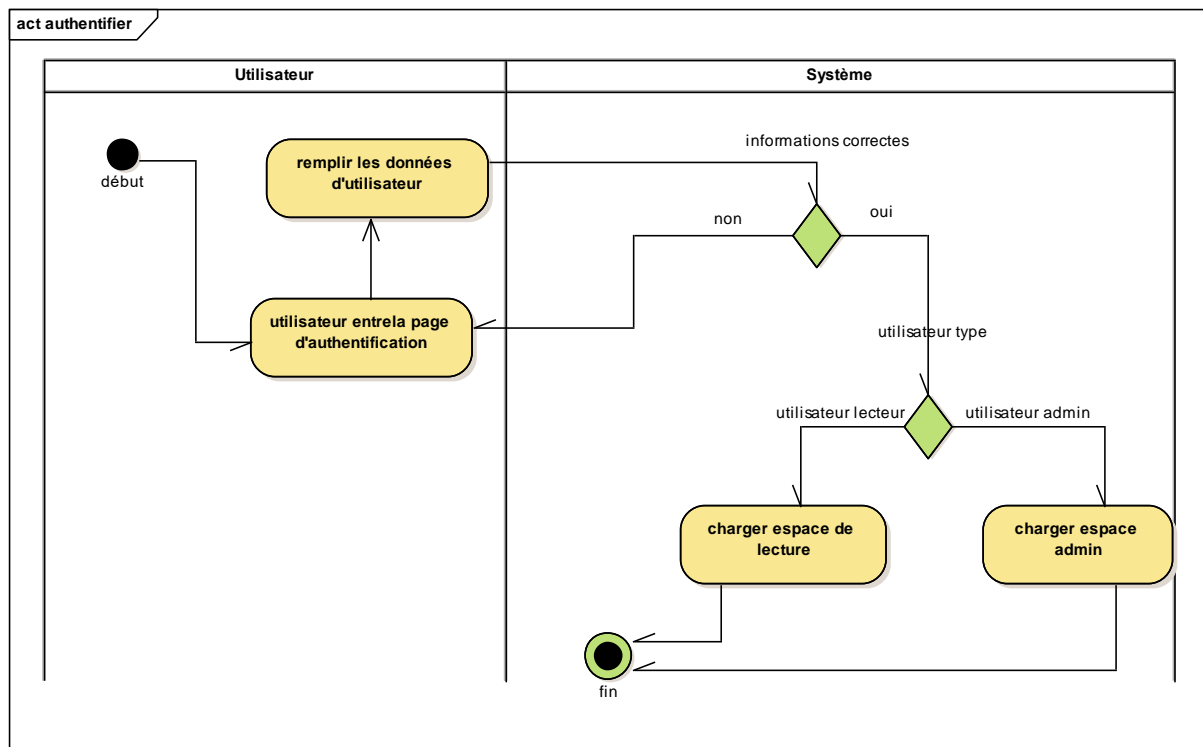


Figure 21 : Diagramme d'activité "authentifier"

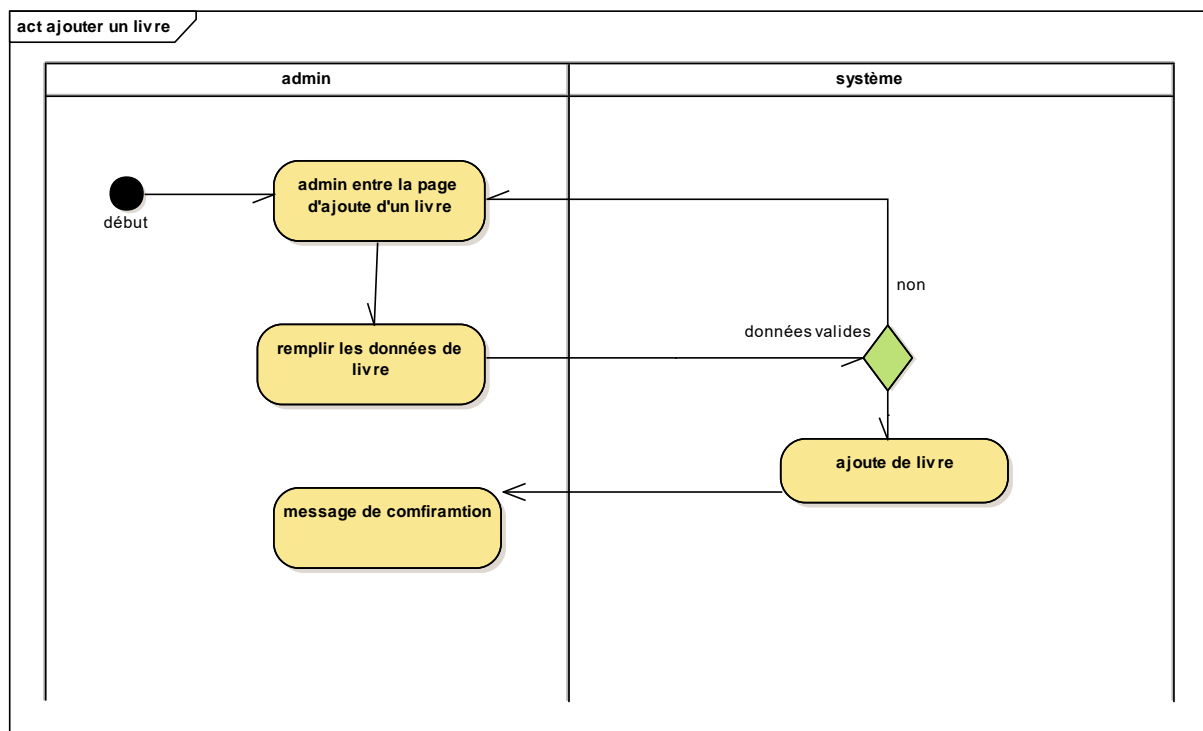


Figure 22 : Diagramme d'activité "ajouter livre"

II. Etude technique de projet

Architecture dispose de deux clients, l'un est un utilisateur, il peut être connecté par mobile ou web. Le deuxième est un administrateur, les deux clients envoient des requêtes (POST, GET, PUT, DELETE) et il reçoivent des réponses JSON.

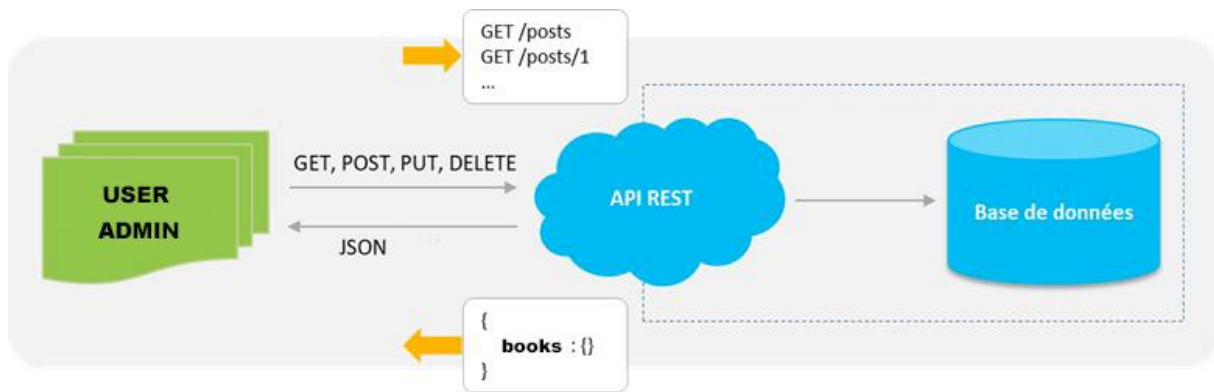


Figure 23 : architecture de projet

Conclusion

Ce chapitre a été consacré à la présentation de la partie conceptuelle de notre projet, et il nous a permis de décrire la partie fonctionnelle de l'application ainsi que les diagrammes de séquences et le diagramme de classe pour clarifier les cas d'utilisations, cela nous permet de passer vers la phase de réalisation.

Chapitre 4 : Réalisation et outils de développement

Introduction

Après avoir présenté la partie conception dans le chapitre précédent. Ce chapitre a pour but de vous présenter notre interface graphique et le fonctionnement de ses composantes, et les outils de développements.

I. Outils de développement

1. Langages utilisés

a. HTML

HTML est un langage informatique utilisé sur l'internet. Ce langage est utilisé pour créer des pages web. L'acronyme signifie HyperText Markup Language, ce qui signifie en français "langage de balisage d'hypertexte". Cette signification porte bien son nom puisqu'effectivement ce langage permet de réaliser de l'hypertexte à base d'une structure de balisage, sa dernière version est HTML5.



b. CSS

CSS Littéralement Cascadant Style Sheets est un langage déclaratif simple pour mettre en forme des pages HTML ou des documents XML. Le langage CSS permet de préciser les caractéristiques visuelles et sonores de présentation d'une page Web : les polices de caractères, les marges et bordures, les couleurs, le positionnement des différents éléments, etc.



c. JavaScript

JavaScript (quelques fois abrégé JS) est un langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, 38 et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés.



d. Dart

Dart est un langage de programmation optimisé par le client pour les applications sur plusieurs plates-formes. Il est développé par Google et est utilisé pour créer des applications mobiles, de bureau, de serveur et Web. Dart est un langage orienté objet, basé sur les classes et récupéré avec la syntaxe de style C. Dart peut compiler en code natif ou en JavaScript. Il prend en charge les interfaces, les mixins, les classes abstraites, les génériques réifiés et l'inférence de type.



e. UML

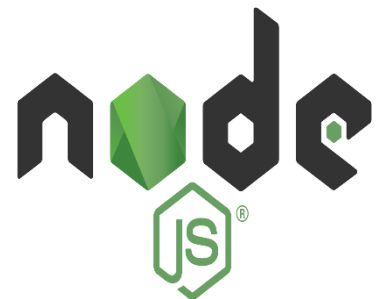
UML Langage de Modélisation Unifié, est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.



2. Technologies utilisées

a. Node.js

Node.js est une plateforme logicielle libre en JavaScript orientée vers les applications réseau événementielles hautement concurrentes qui doivent pouvoir monter en charge. Elle utilise la machine virtuelle, la librairie libuv pour sa boucle d'évènements, et implémente sous licence MIT les spécifications Commons. Parmi les modules natifs de Node.js, on retrouve http qui permet le développement de serveur HTTP. Il est donc possible de se passer de serveurs web tels que Nginx ou Apache lors du déploiement de sites et d'applications web développés avec Node.js.



Concrètement, Node.js est un environnement bas niveau permettant l'exécution de JavaScript côté serveur.

Node.js est utilisé notamment comme plateforme de serveur Web, elle est utilisée par Groupon, Vivaldi, SAP, LinkedIn, Microsoft, Yahoo!, Walmart, Rakuten, Sage et PayPal

b. Express.js

Express.js est un Framework pour construire des applications web basées sur Node.js. C'est de fait le Framework standard pour le développement de serveur en Node.js. L'auteur original, TJ Holowaychuk, le décrit comme un serveur inspiré de Sinatra dans le sens qu'il est relativement minimaliste tout en permettant d'étendre ses fonctionnalités via des plugins.



c. React

React (aussi appelé React.js ou ReactJS) est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état.



React est une bibliothèque qui ne gère que l'interface de l'application, considéré comme la vue dans le modèle MVC. Elle peut ainsi être utilisée avec une autre bibliothèque ou un framework MVC comme AngularJS. La bibliothèque se démarque de ses concurrents par sa flexibilité et ses performances, en travaillant avec un DOM virtuel et en ne mettant à jour le rendu dans le navigateur qu'en cas de nécessité.

d. Redux

Redux est une bibliothèque open-source JavaScript de gestion d'état pour applications web. Elle est plus couramment utilisée avec des bibliothèques comme React ou Angular pour la construction d'interfaces utilisateur. Semblable à (et inspirée de) l'architecture Flux, elle a été créée par Dan Abramov et Andrew Clark.



e. mongoDB

mongoDB est un système de gestion de base de données orienté documents, répartissable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédéfini des données. Il est écrit en C++. Le serveur et les outils sont distribués sous licence SSPL, les pilotes sous licence Apache et la documentation sous licence Creative Commons. Il fait partie de la mouvance NoSQL.



f. Flutter

Flutter est un framework de développement mobile multiplateformes réactif utilisant le langage Dart. Il est utilisé pour développer des applications pour Android, iOS, Windows, Mac, Linux, Google Fuchsia. Flutter est un kit de



développement logiciel open source créé par Google. La première version de Flutter était connue sous le nom de code "Sky" et fonctionnait sur le système d'exploitation Android. Il a été dévoilé lors du sommet des développeurs Dart 2015. Sa version alpha initiale était de retour en mai 2017, elle est donc beaucoup plus jeune que React Native. Le 11 décembre 2019, Flutter 1.12 est sorti lors de l'événement Flutter Interactive

g. Ant Design :

Ant Financial propose un grand nombre de produits de niveau entreprise. Avec des scénarios complexes, les concepteurs et les développeurs doivent souvent réagir rapidement en raison des changements fréquents dans les demandes de produits et du flux de travail R & D simultané. De nombreux contenus similaires existent dans le processus. Grâce à l'abstraction, nous avons pu obtenir des composants et des pages stables et hautement réutilisables.



3. Outils utilisés

a. Enterprise Architect

Enterprise Architect est un logiciel de modélisation et de conception UML, édité par la société australienne Sparx Systems. Couvrant, par ses fonctionnalités, l'ensemble des étapes du cycle de conception d'application, il est l'un des logiciels de conception et de modélisation les plus reconnus.



b. Android studio

Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux



c. Visual code

Visual Studio Code est présenté lors de la conférence des développeurs Build d'avril 2015 comme un éditeur de code multi-plateforme, open source et gratuit, supportant une dizaine de langages.

Il est basé sur Electron, une structure utilisée pour déployer des applications Node.js pour le bureau exécuté sur le moteur Blink. Bien qu'il utilise le framework Electron, le logiciel n'utilise pas Atom mais utilise le même composant éditeur (nommé "Monaco") utilisé dans Azure DevOps (anciennement appelé Visual Studio Online et Visual Studio Team Services).



Le code source est fourni sous la licence libre MIT sur le site du projet sur Github. En revanche, l'exécutable est proposé sur le site officiel de Microsoft sous une licence propriétaire.

d. Postman

Postman est un type de test de logiciel qui implique de tester les interfaces de programmation d'applications (API) directement et dans le cadre des tests d'intégration pour déterminer si elles répondent aux attentes en termes de fonctionnalité, de fiabilité, de performances et de sécurité. Étant donné que les API n'ont pas d'interface graphique, les tests d'API sont effectués à la couche de message



e. MongoDB Compass

MongoDB Compass : 3.4 est excellente. Vous obtenez un aperçu rapide du comportement de vos données et vous pouvez facilement identifier un goulot d'étranglement o

u une requête lente qui pourrait être à l'origine de problèmes de performances, ce qui signifie que vous pouvez obtenir une solution encore plus rapide pour votre problème. Compass est le couteau suisse qui complète notre travail avec MongoDB



mongoDB
Compass

II. Réalisation

Notre application fonctionne sur différents appareils(mobile/Desktop)

1. Espace administrateur

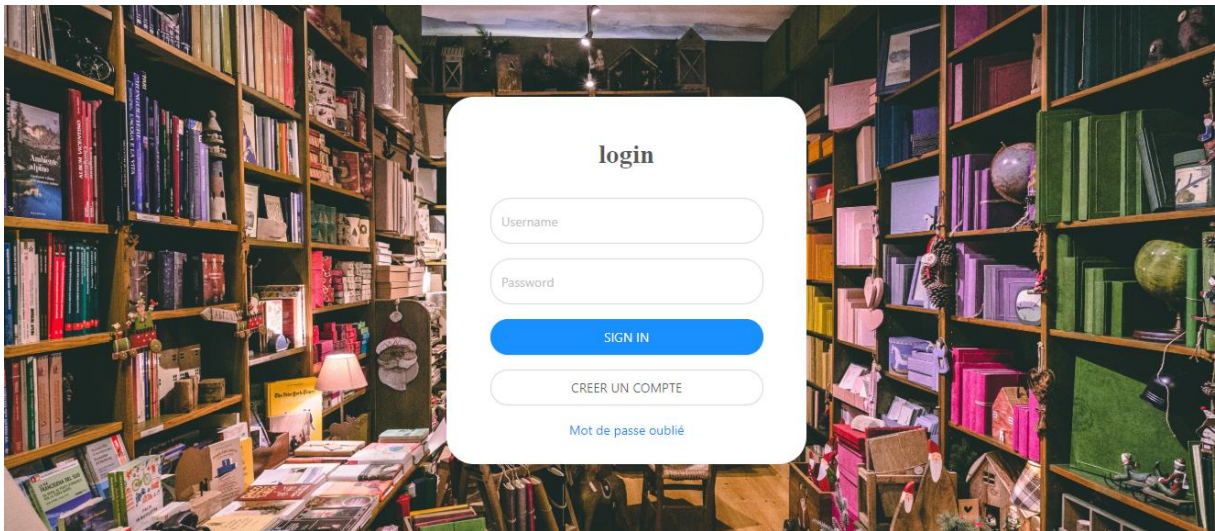


Figure 24 : Interface d'authentification

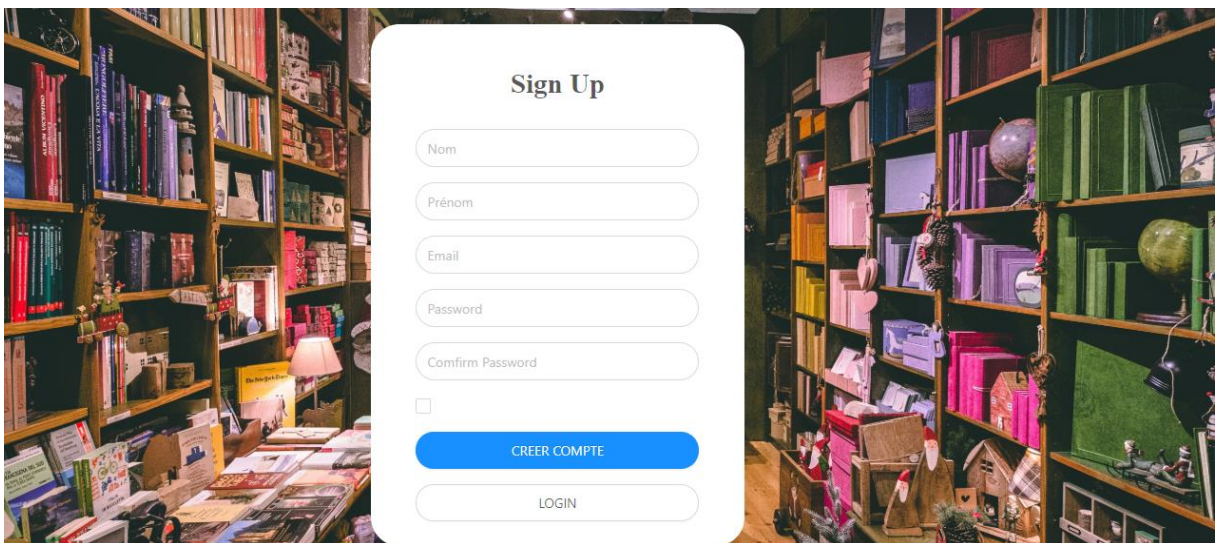


Figure 25 : Interface d'enregistrement

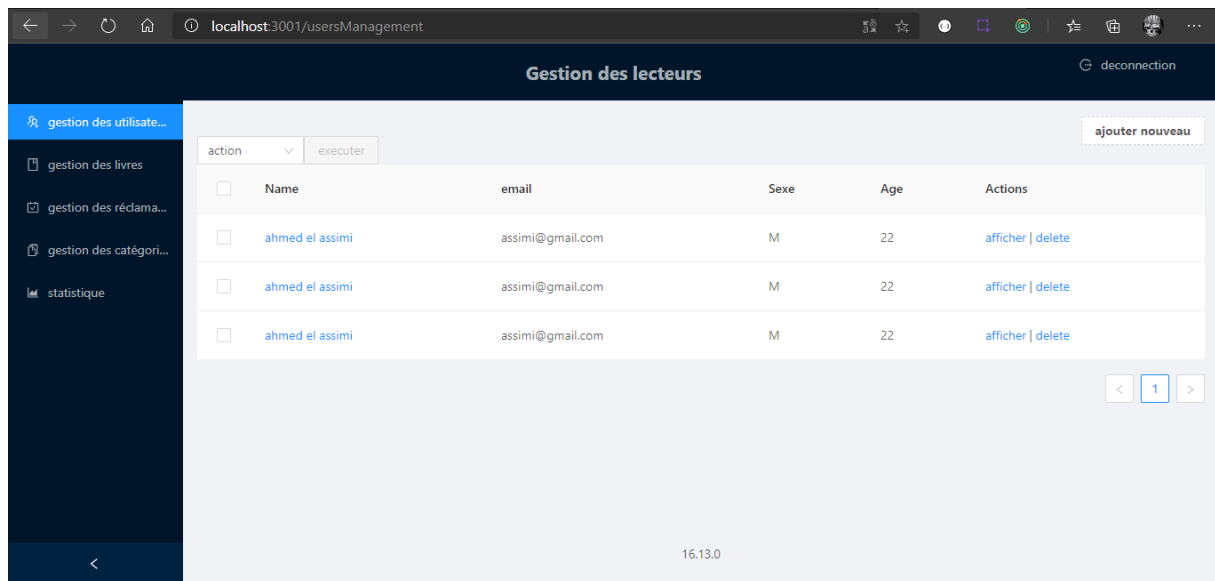


Figure 26: interface de gestion des utilisateurs

The screenshot shows the "Ajouter un Livre" form within the "Gestion des lecteurs" interface. The sidebar is the same as in Figure 26. The main area has a title "Ajouter un Livre" and a "deconnection" link. The form contains the following fields: "Nom Complet:" with sub-fields for "prénom" and "nom"; "Date de naissance:" with a text input showing "Invalid date" and a calendar icon; "Sexe:" with a dropdown menu; "email:" with a text input; "Login:" with sub-fields for "user name" and "password"; "Téléphone:" with a text input showing "Mobile"; and "show state:" with a "show state" button. At the bottom is a "Submit" button. The browser address bar shows "localhost:3001/usersManagement/add".

Figure 27 : interface d'ajout d'un utilisateur

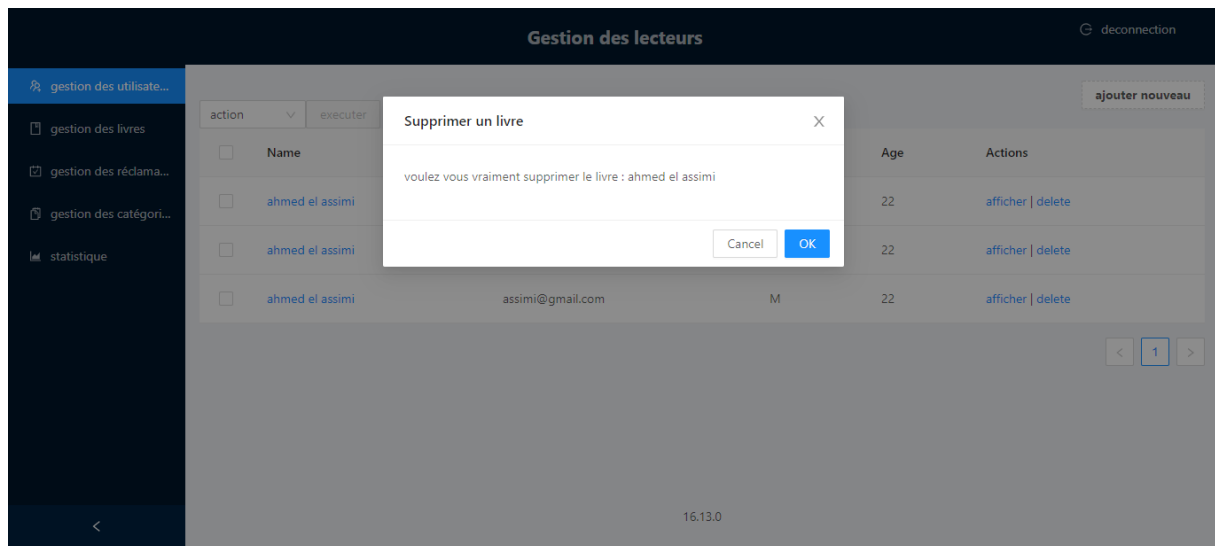


Figure 28 : suppression d'un utilisateur

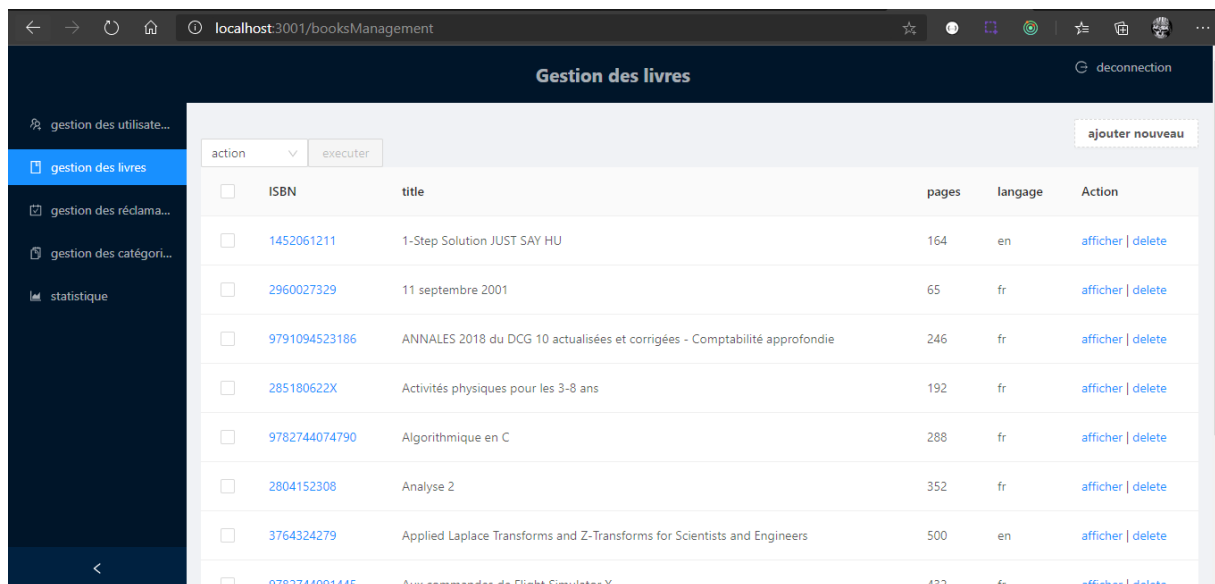


Figure 29 : interface de gestion des livres

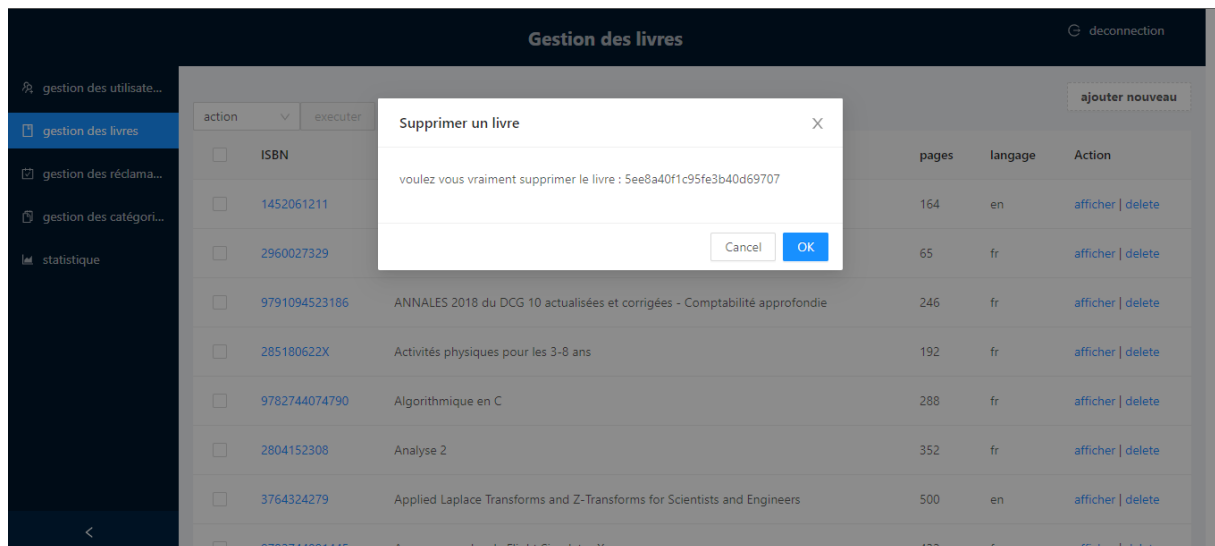


Figure 30 : suppression d'un livre

The screenshot shows the 'Gestion des livres' (Book Management) interface with the 'Ajouter un Livre' (Add a Book) form. The form includes fields for 'Titre', 'ISBN', 'Description', 'Nombre des pages', 'Langage', 'Auteurs', 'Pays', and 'Sujets'. A 'Submit' button is at the bottom right.

Ajouter un Livre

Titre:

ISBN:

Description:

Nombre des pages:

Langage:

Auteurs:

Pays:

Sujets:

:

Figure 31 : interface d'ajout d'un livre

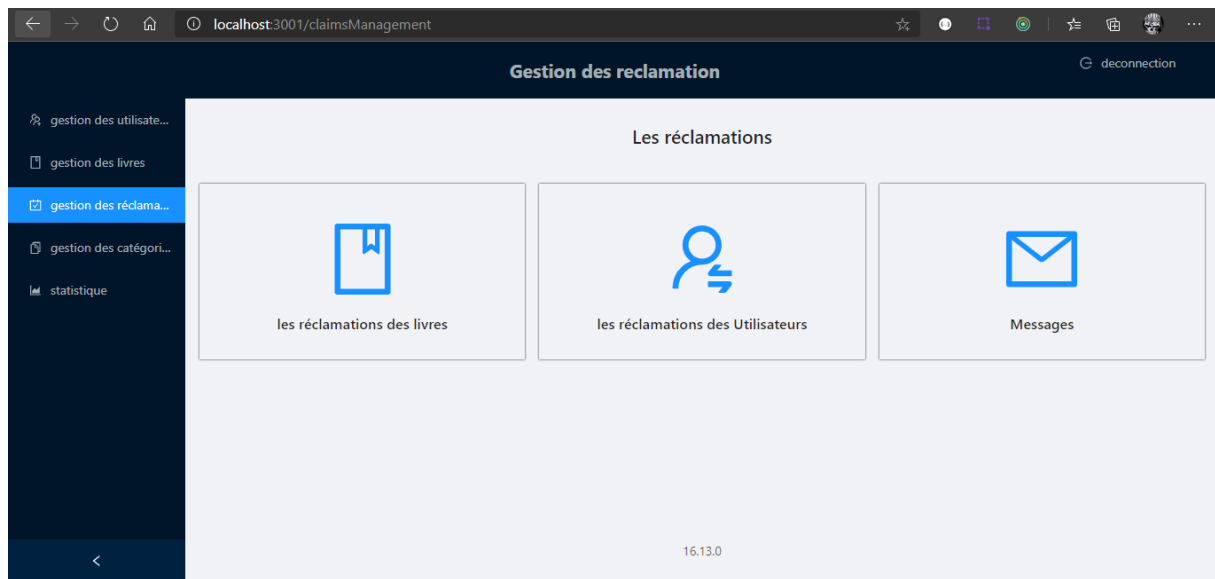


Figure 32: interface de gestion des réclamations

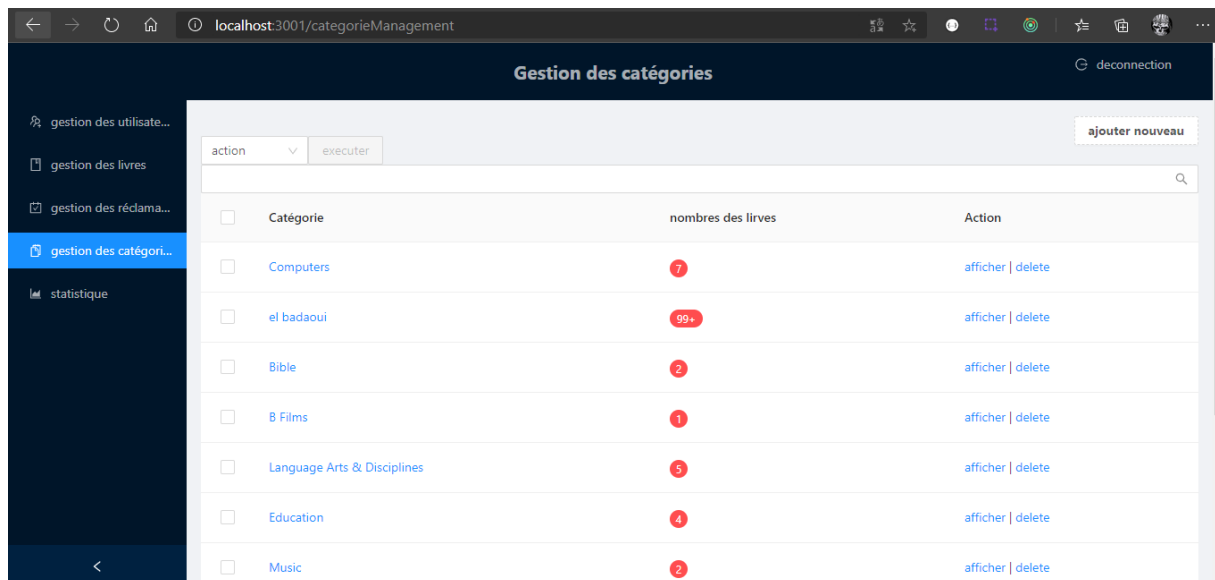


Figure 33 : interface de gestion des catégories

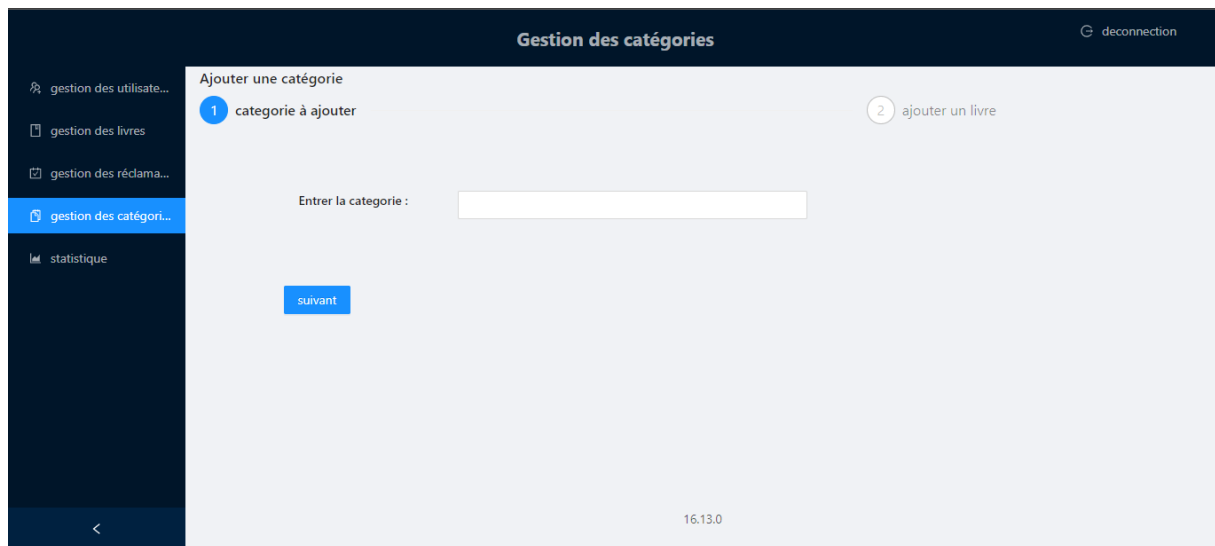


Figure 34 : interface d'ajout d'une catégorie

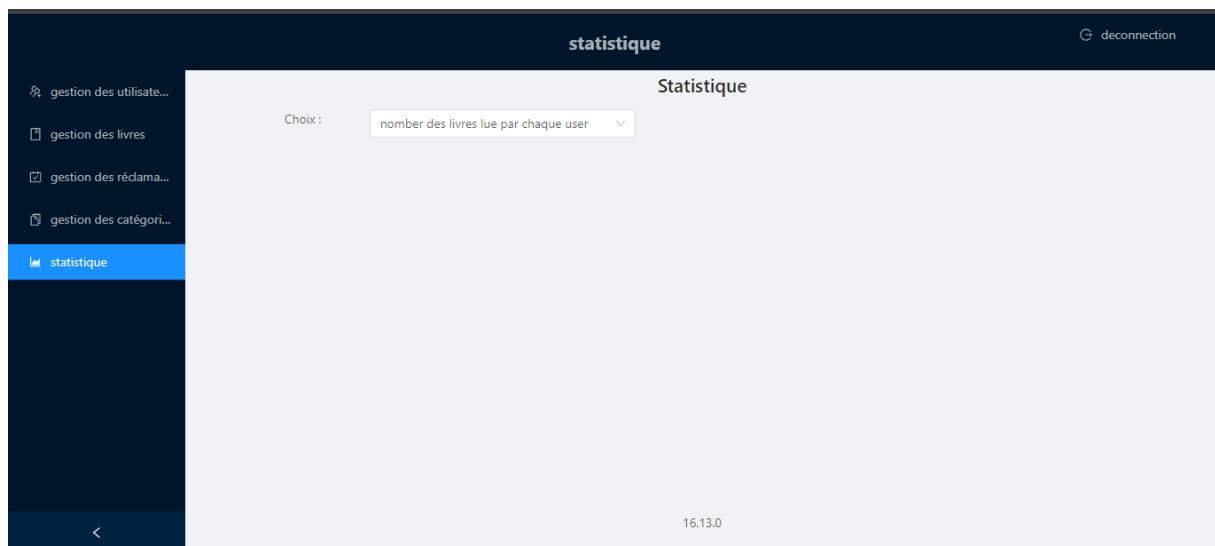


Figure 35 : interface de statistiques

2. Espace Lecteur

a. Web

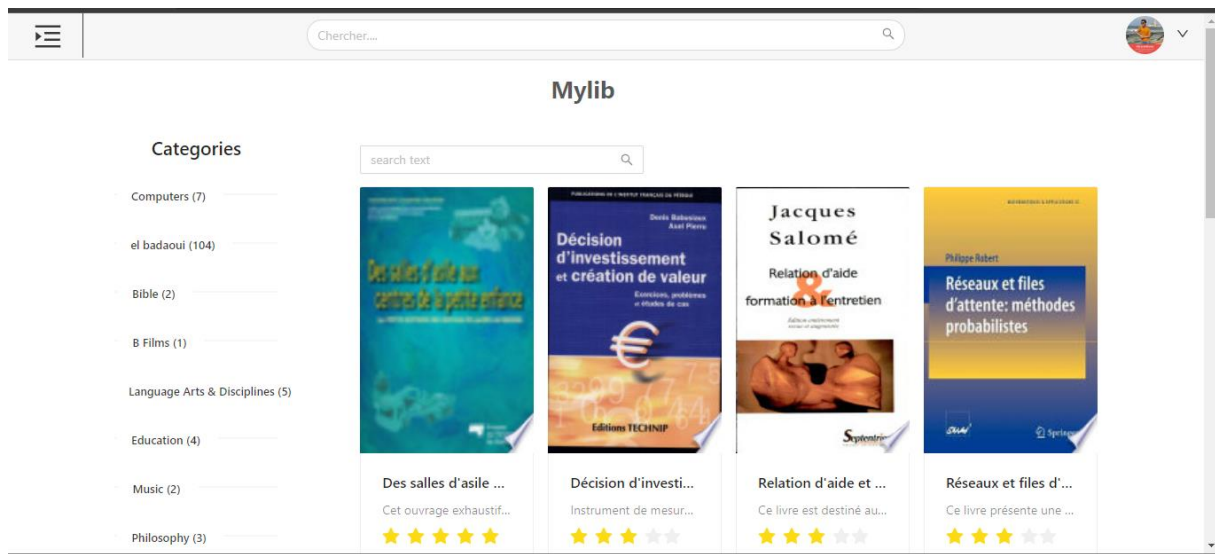


Figure 36 : interface de tous les livres

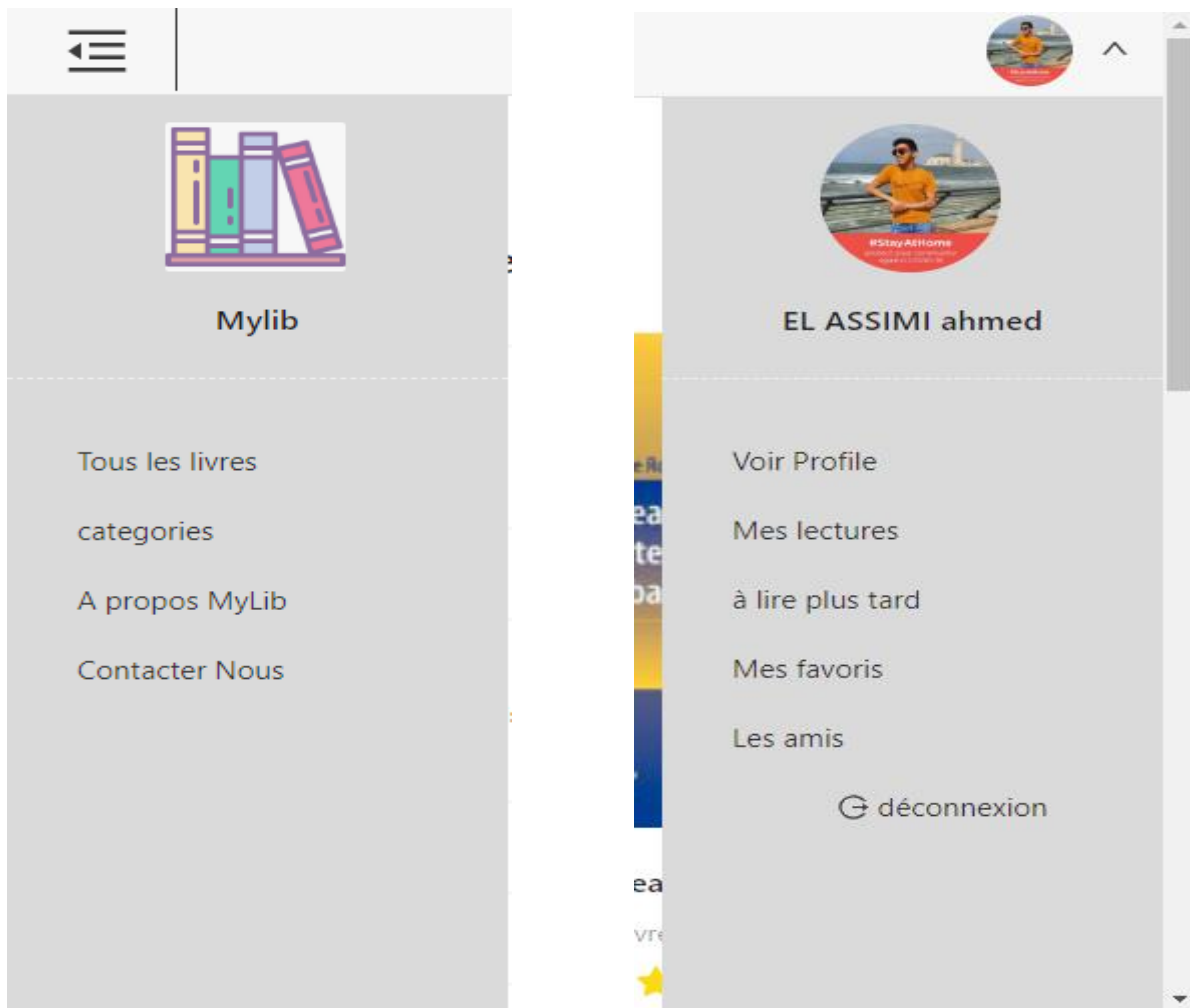


Figure 37 : menu de navigation et d'utilisateur

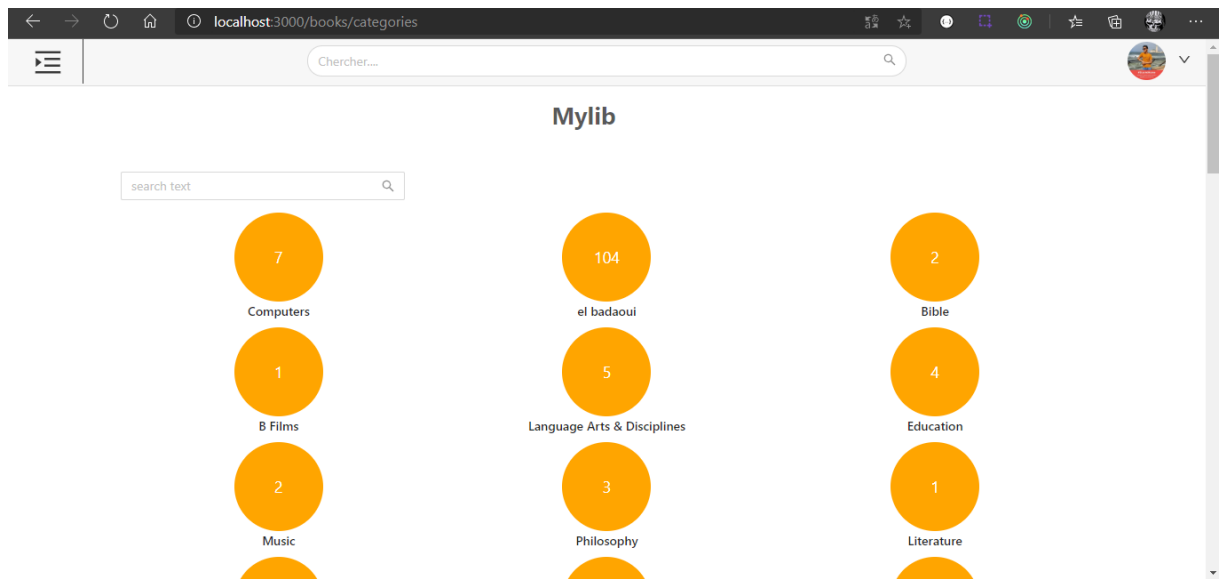


Figure 38 : interface des catégories

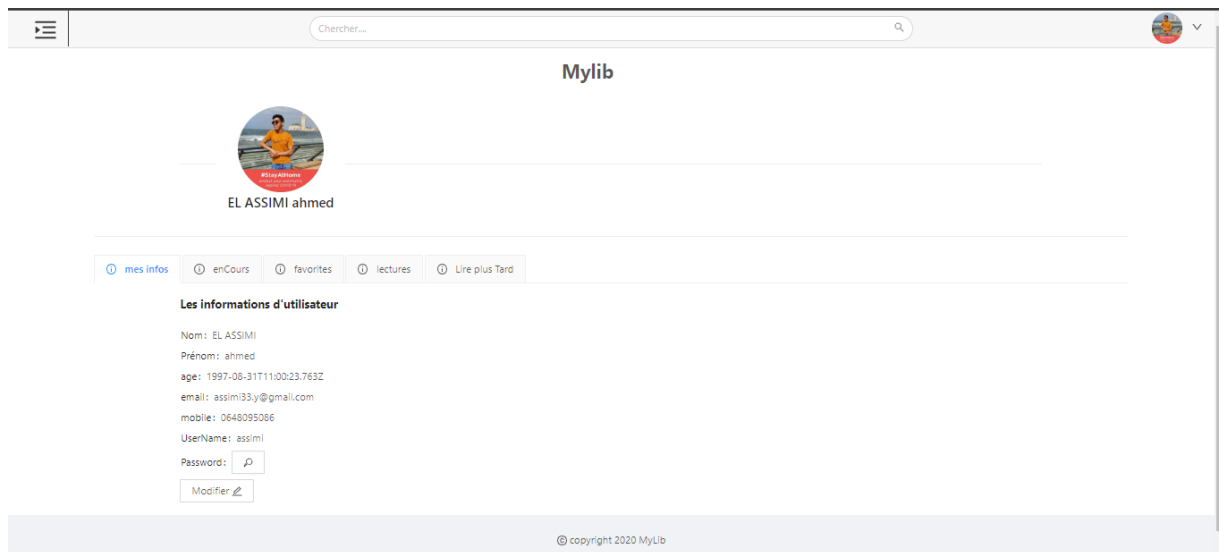


Figure 39 : profile de lecteur

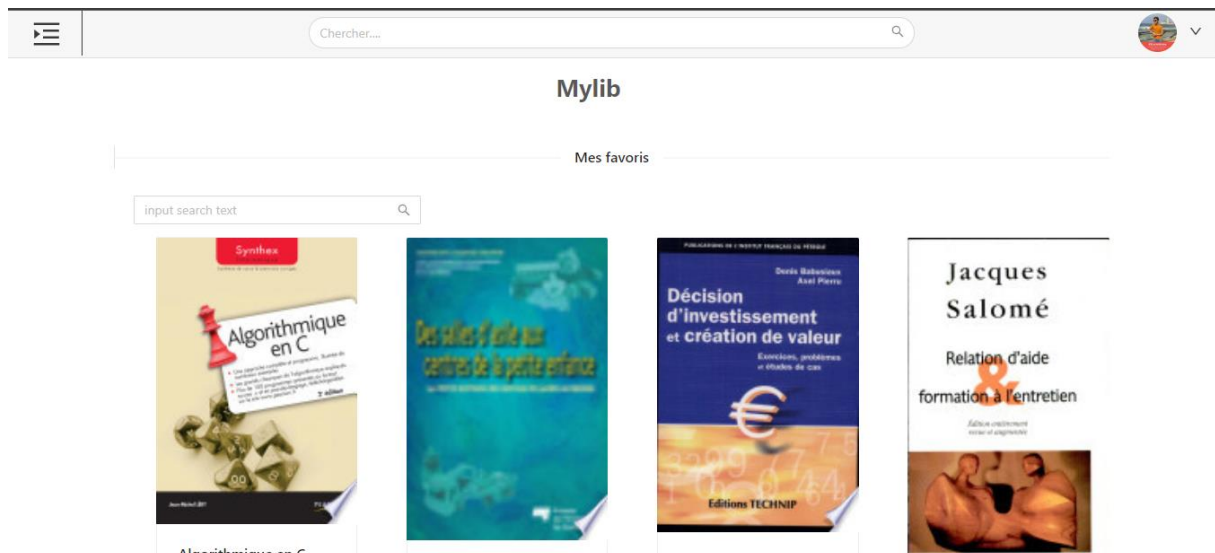


Figure 40 : les livres favoris

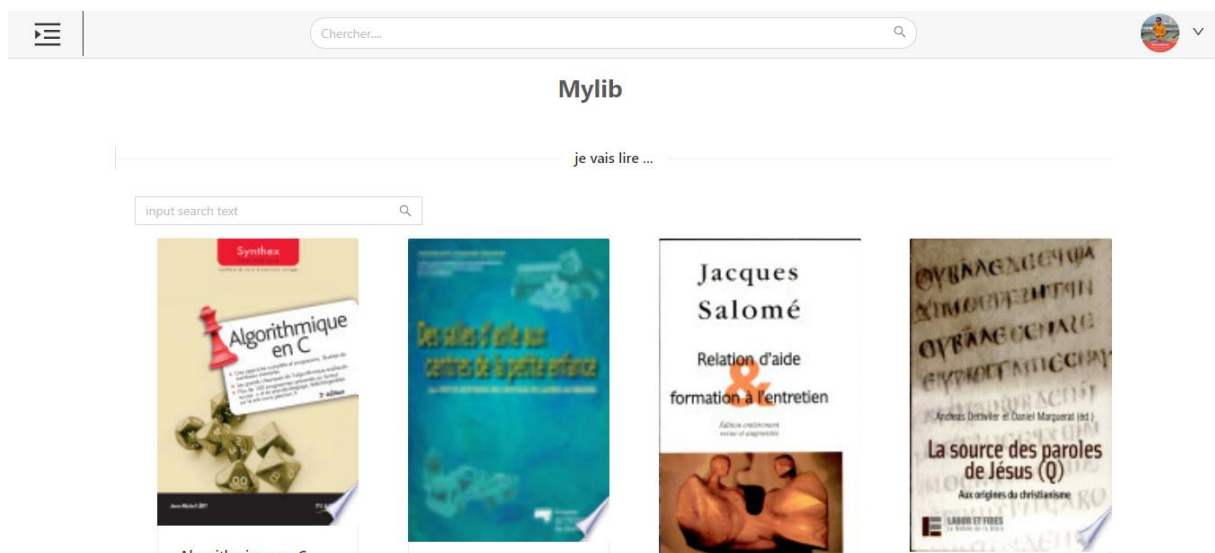


Figure 41 : les livres à lire plus tard

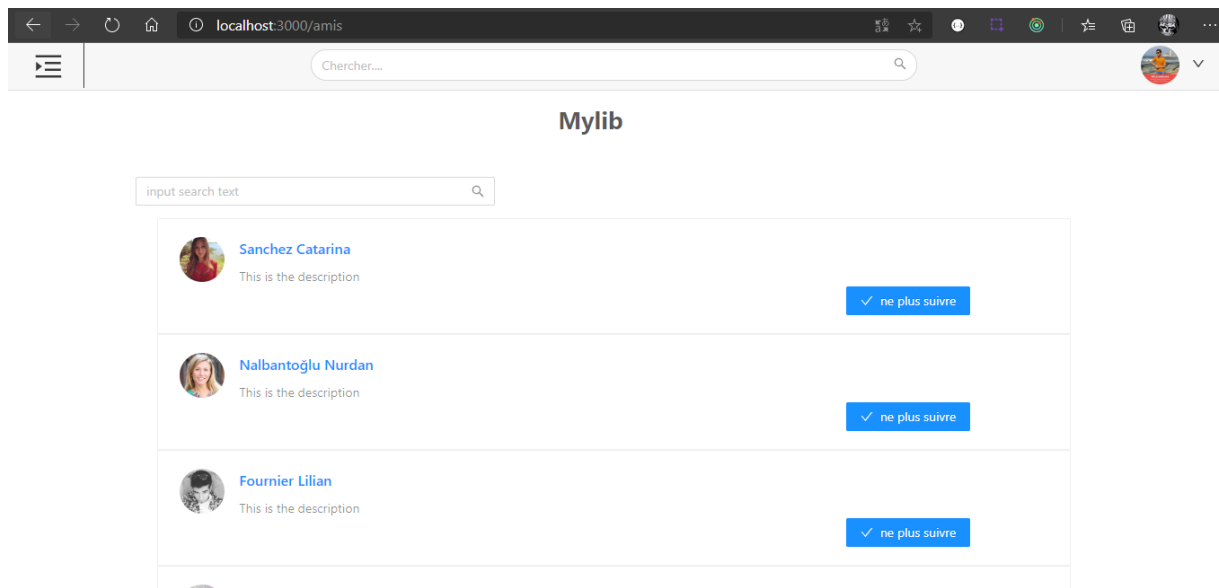


Figure 42 : interface d'affichage des amis

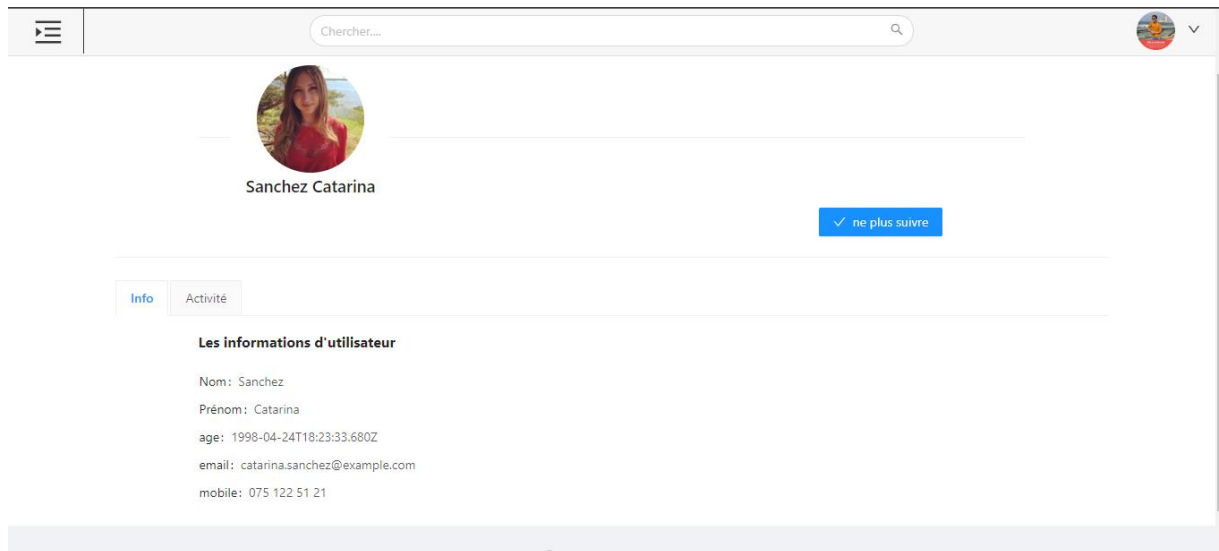


Figure 43 : interface de profile d'un ami

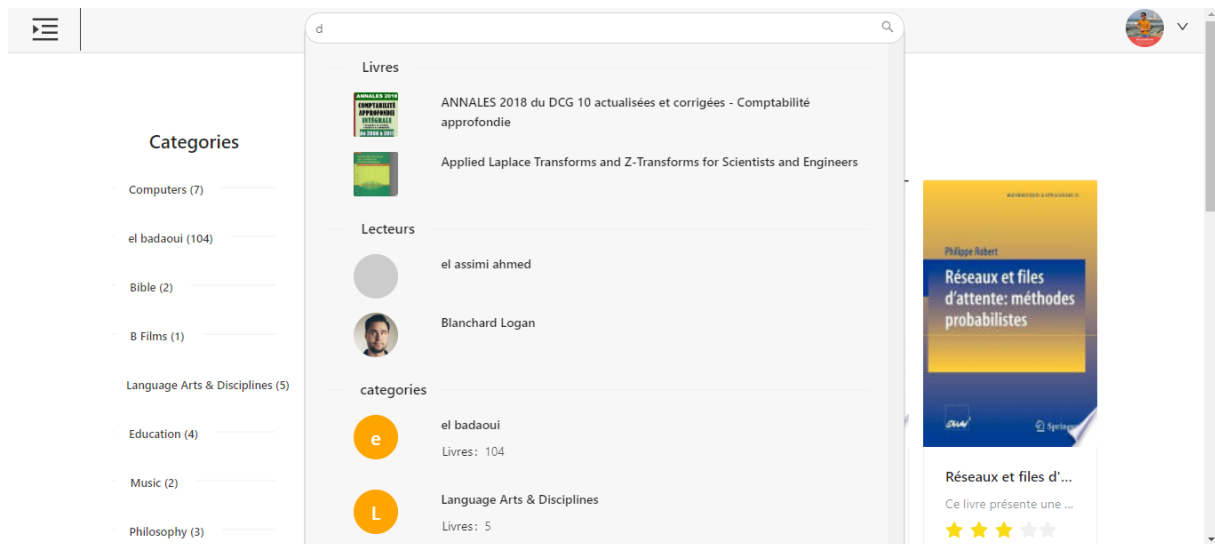


Figure 44 : barre de recherche

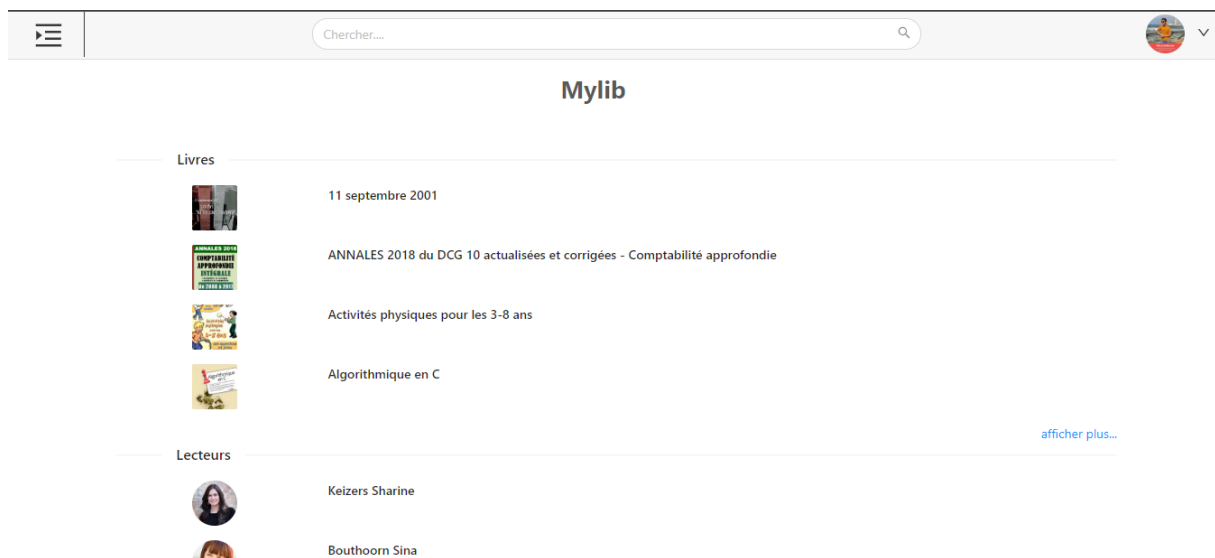


Figure 45 : interface de recherche

The image shows a web browser interface for a support contact form. At the top, there is a navigation bar with a hamburger menu icon on the left, a search bar with the placeholder text "Chercher..." in the center, and a user profile icon with a dropdown arrow on the right. Below the navigation bar, the page title "Mylib" is centered. The main content area is divided into two sections. On the left, there is a vertical line and the word "left" below it. On the right, there is a form with two input fields: "subject:" with the placeholder text "entrer subject" and "Message:" with the placeholder text "Message". Below these fields is a button labeled "Envoyer". At the bottom of the page, there is a footer with the text "© copyright 2020 MyLib".

Figure 46 : interface pour contacter le support

b. Mobile



Figure 48:écran de chargement

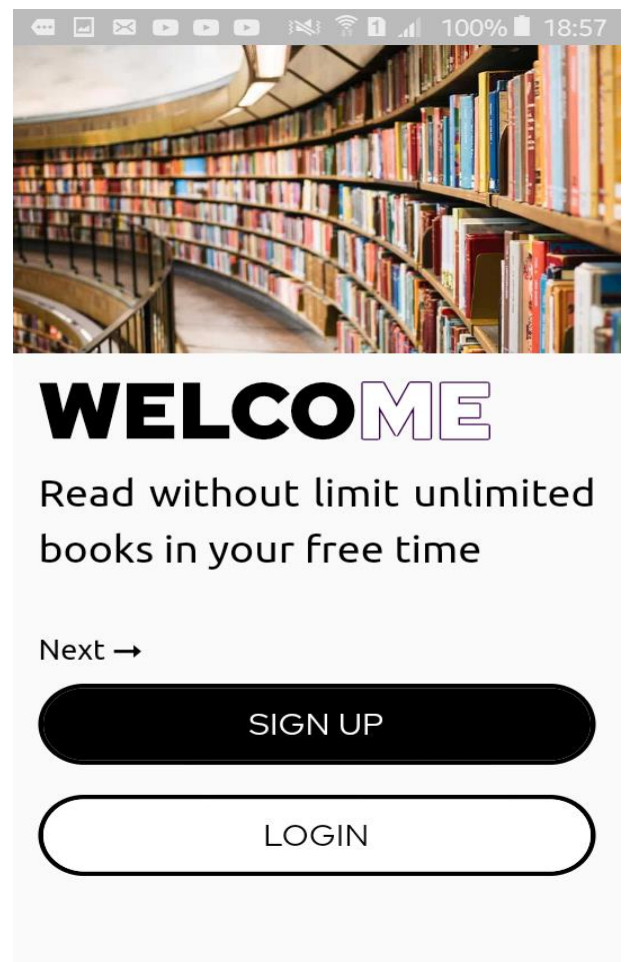


Figure 47:écran de bienvenue

Description :

L'utilisateur dès le lancement d'application reçoit un écran de chargement et après quelques secondes un message de bienvenue est affiché pour indiquer aux utilisateurs de se connecter ou bien de créer un compte.



SIGN UP

Full Name

Number Phone

Email

Password

CREATE ACCOUNT

Figure 49:écran de création de compte



LOG IN

Email

Password

LOGIN

Forget password ?

CREATE ACCOUNT

Figure 50:écran de connexion

Description :

L'utilisateur dès le lancement d'application reçoit un écran de chargement et après quelques secondes un message de bienvenue est affiché pour indiquer aux utilisateurs de se connecter ou bien de créer un compte.

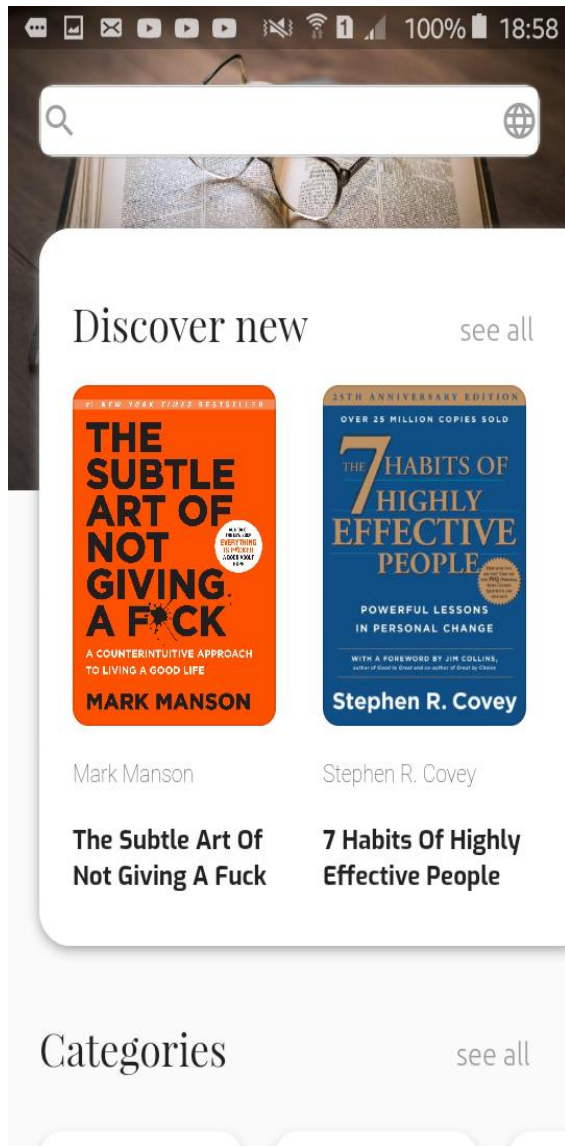


Figure 52:écran d'accueil 1

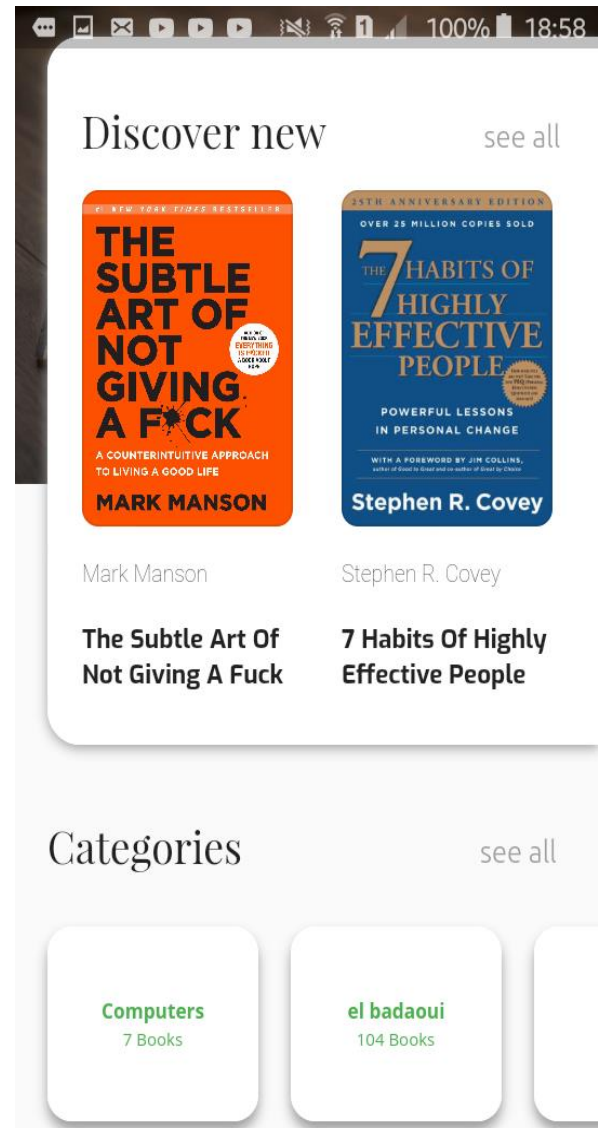


Figure 51:écran d'accueil 2

Description :

Après la validation des données de connexion, l'utilisateur est redirigé vers l'écran d'accueil qui comporte les livres nouveaux, ainsi les différentes catégories

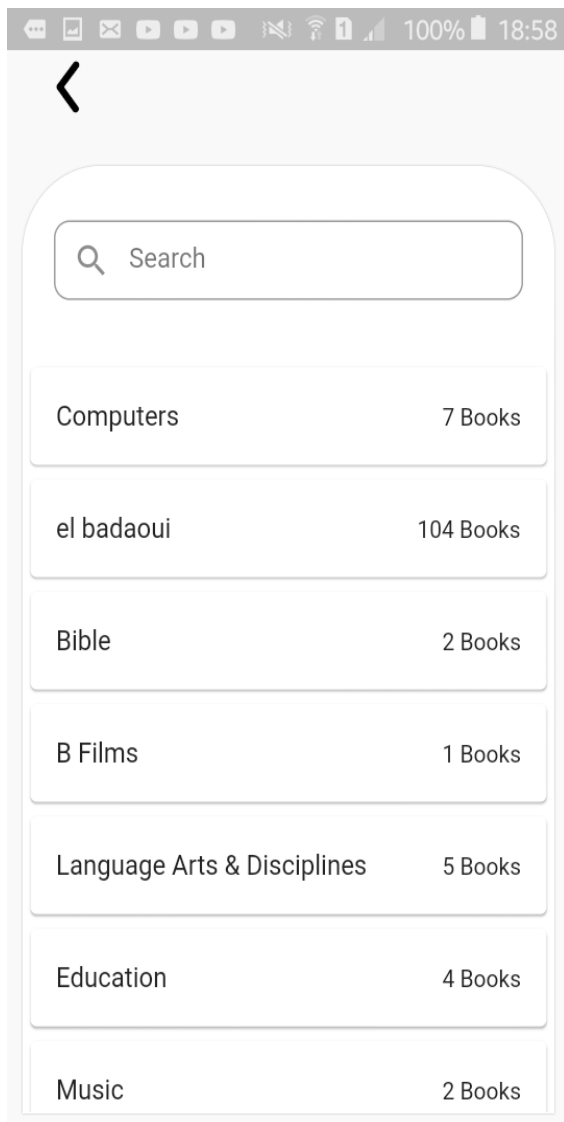


Figure 53: écran des catégories

Description :

L'utilisateur après qu'il tape sur une catégorie, l'écran des livres s'affiche et il peut choisir et chercher un livre d'une catégorie spécifique.

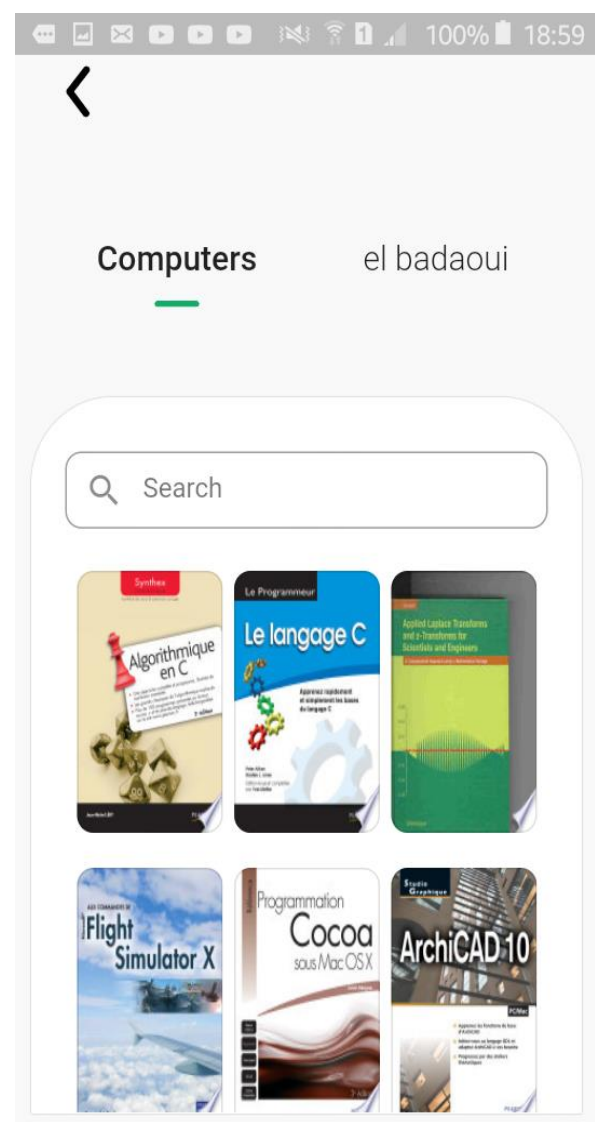


Figure 54:écran des livres par catégorie

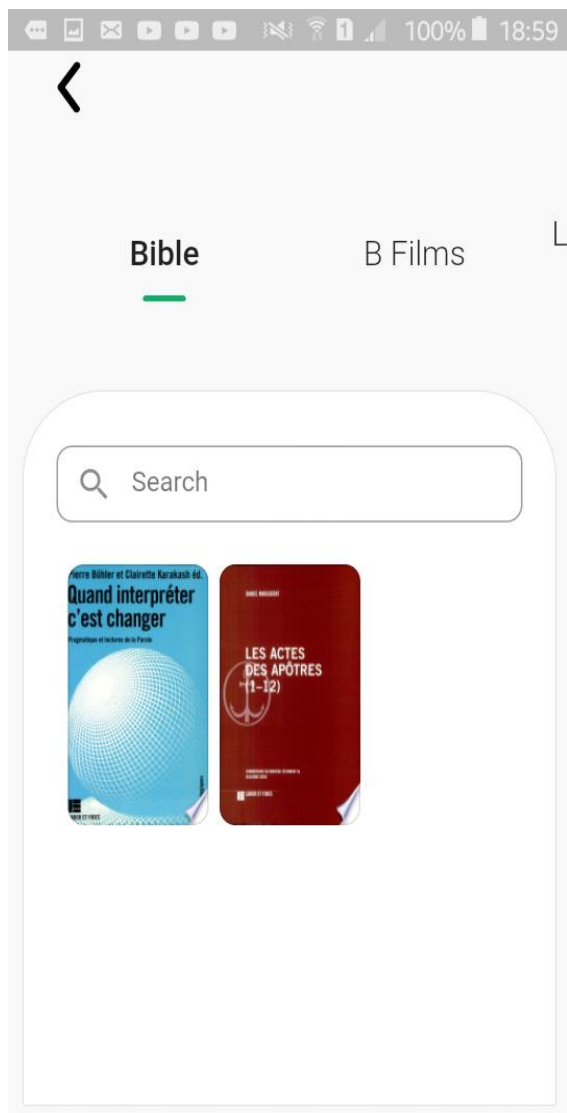


Figure 56:écran des livres d'une autre catégorie

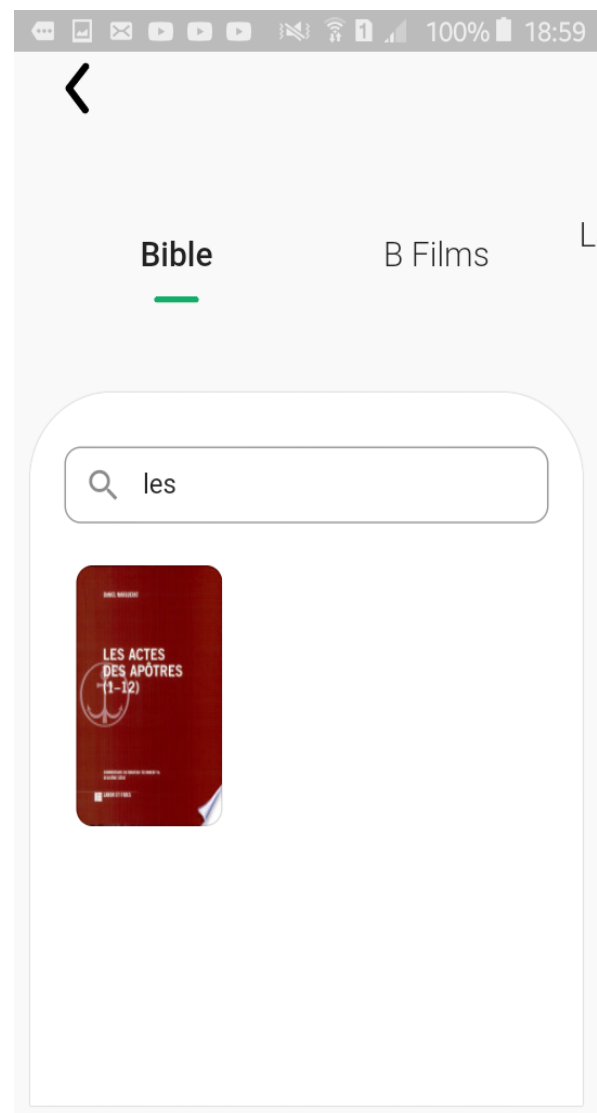
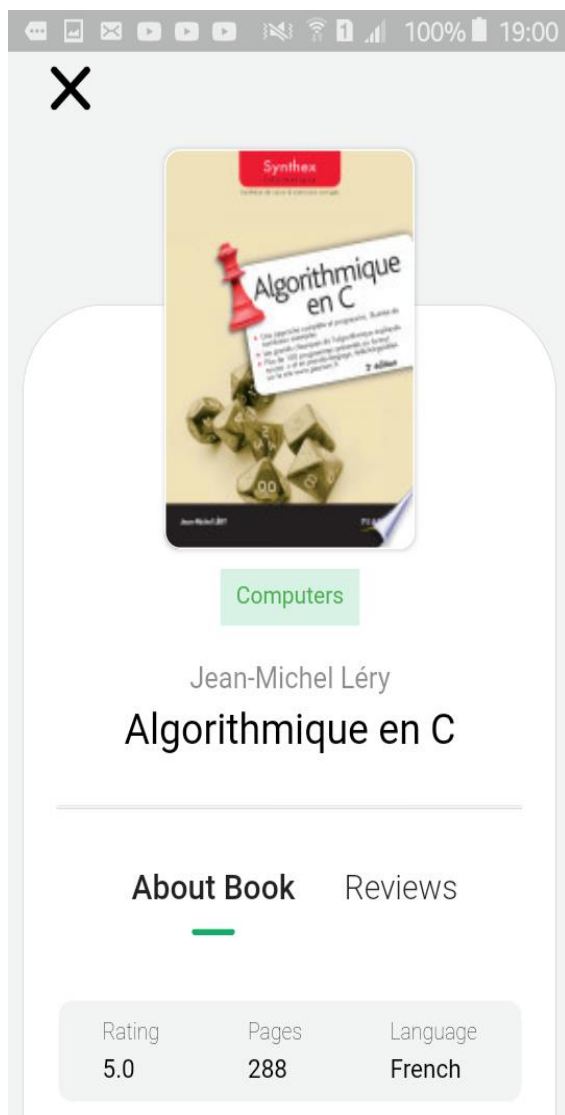


Figure 55:écran des livres après la recherche

Description :

L'utilisateur après qu'il tape sur une catégorie, l'écran des livres s'affiche et il peut choisir et chercher un livre d'une catégorie spécifique.



Description :

L'utilisateur apr s qu'il tape sur un livre, l' cran de livre s'affiche en disposant plusieurs informations sur le livre (pages,langage,description...)

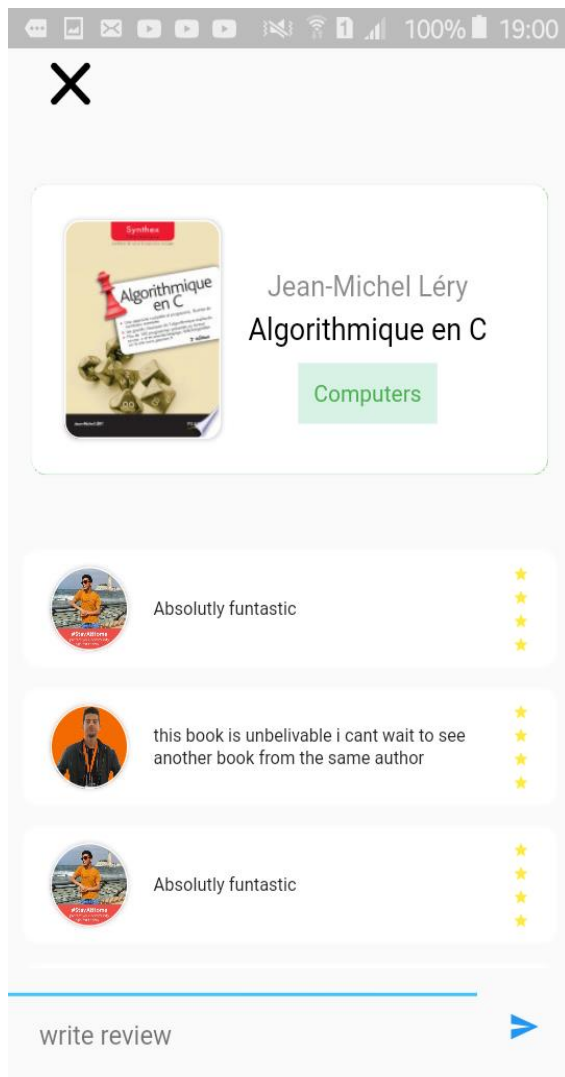


Figure 60:écran des évaluations des utilisateur sur un livre

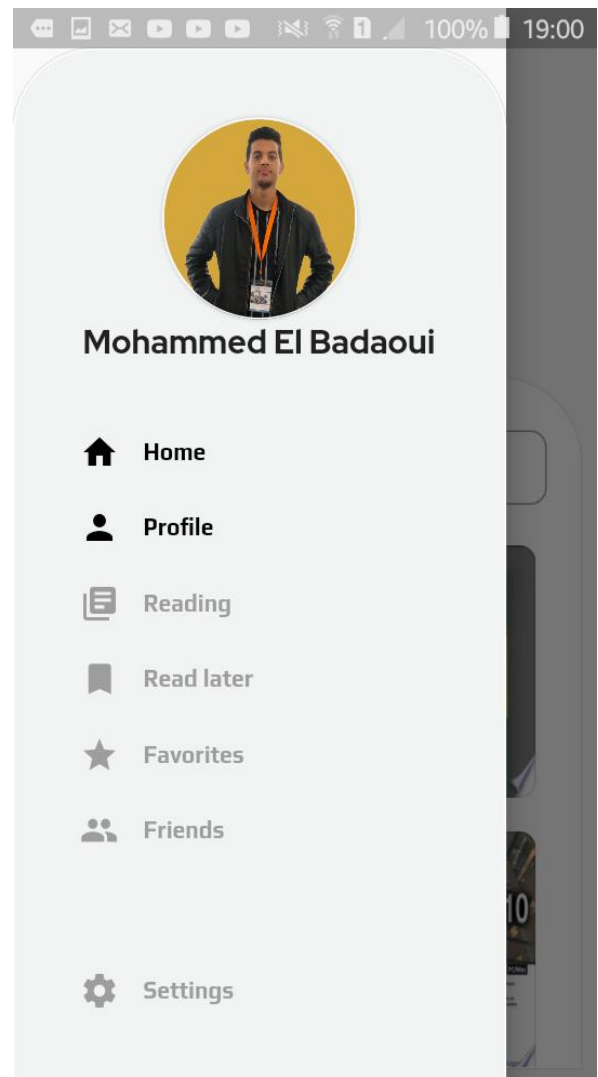


Figure 59:menu de l'application

Description :

L'utilisateur peut voir les évaluations et évaluer lui-même un livre.

Le lecteur peut consulter un ensemble des barres :

Page principale, profile d'utilisateur, les livres à lire plus tard, les livres favoris.

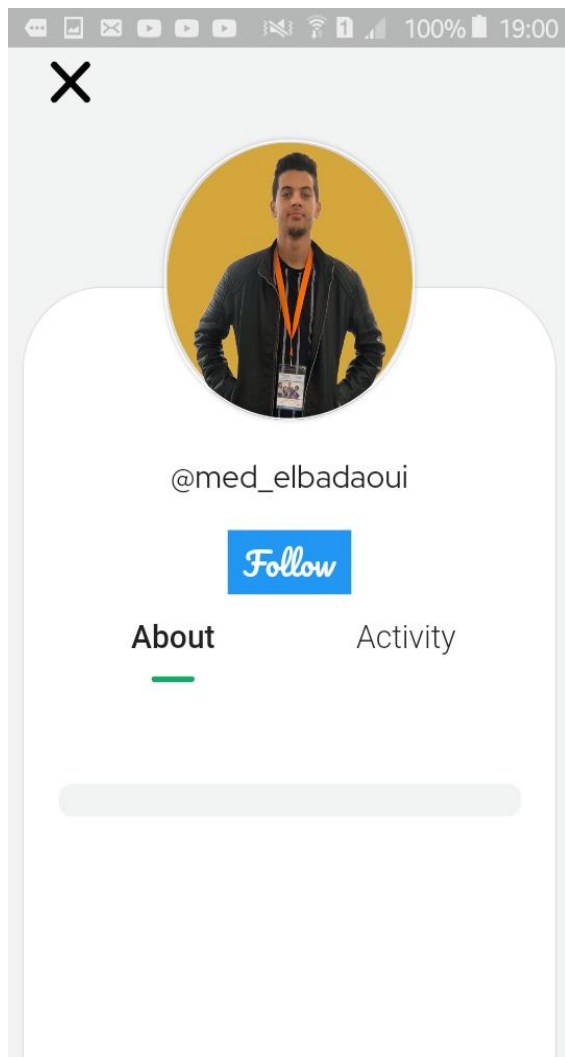


Figure 62:écran de profile d'un utilisateur

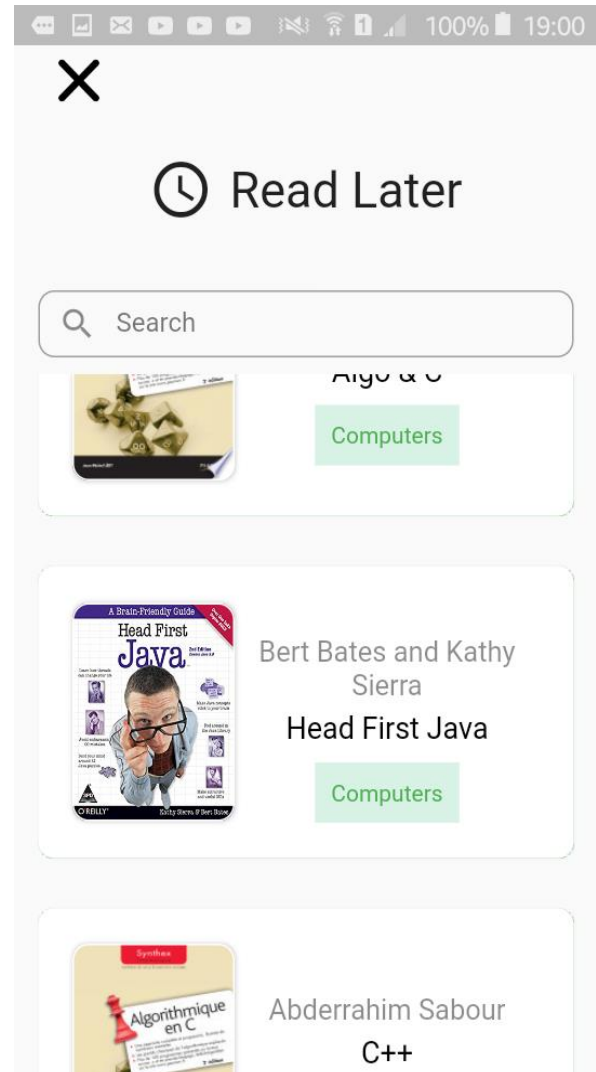


Figure 61:écran des livres à lire plus tard

Description :

L'utilisateur peut afficher un profile et savoir ses activités et ses informations.

Le lecteur peut ajouter des livres dans la liste des livres à lire plus tard et les consulter régulièrement

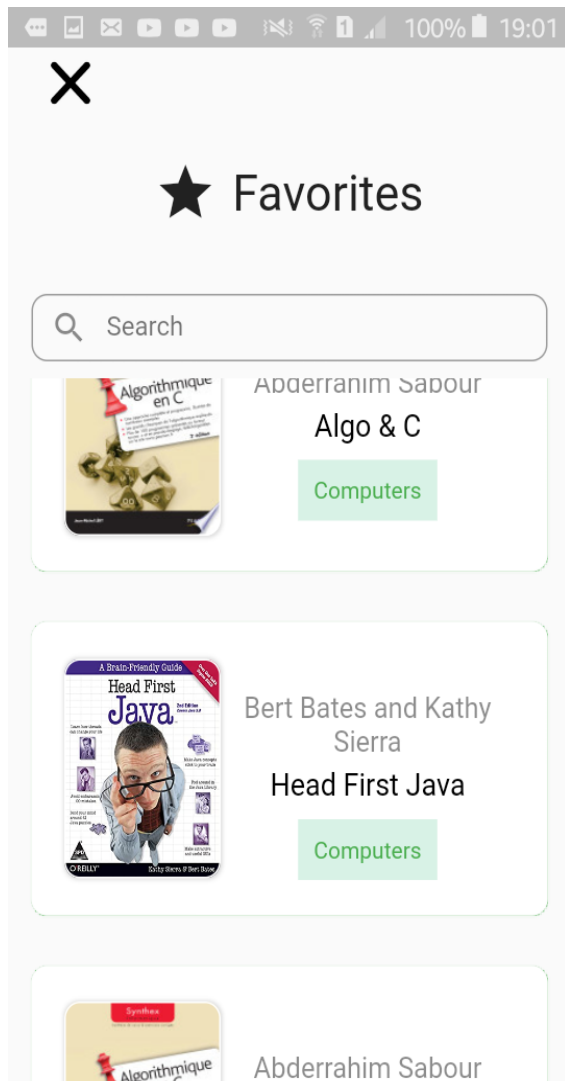


Figure 64:écran des livres favoris

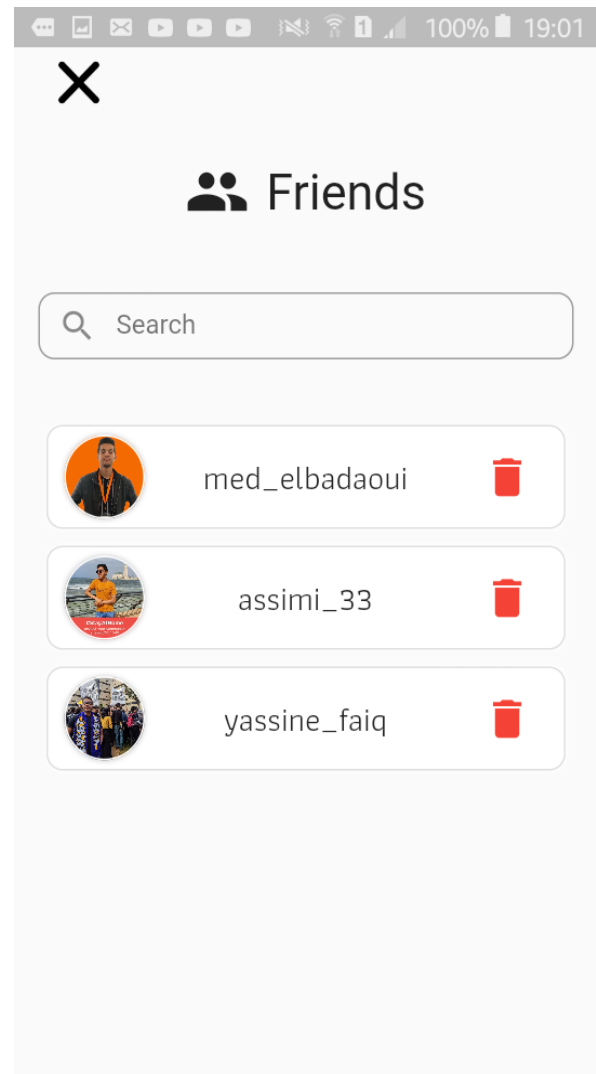


Figure 63:écran de la liste des amis

Description :

Le lecteur peut ajouter des livres dans la liste des livres à lire plus tard et les consulter régulièrement

L'utilisateur peut voir sa liste des amis et il peut à n'importe quel moment éliminer un ami.

Conclusion générale

Dans notre projet, nous étions amenés à concevoir et réaliser une plateforme en ligne qui est une bibliothèque dédiée aux livres électroniques. Dans le premier chapitre nous avons abordé le contexte général du projet ainsi l'équipe de projet, puis nous avons mis l'accent sur l'étude fonctionnelle et conceptuelle de notre projet, et en terminant avec la dernière phase qui est la réalisation et l'implémentation.

Du coup, Notre projet ne se termine pas dans les figures mentionnées, nous souhaitons améliorer notre travail, en ajoutant d'autres méthodes et d'autres fonctionnalités qui sont rarement rencontrés. Également, nous espérons travailler avec d'autres outils de développement pour enrichir notre compétence.

Pendant la durée que nous avons préparé ce projet, nous avons appris beaucoup de chose comme l'esprit collectif et le travail en groupe qui va nous aider à s'engager dans le domaine professionnel, et ce travail nous a donné l'envie de commencer d'autres projets identiques à notre thème afin de plonger dans l'innovation et la créativité.

Tableau de figure

Figure 1:diagramme de contexte	13
Figure 2 : diagramme de package	14
Figure 3 : diagramme de cas d'utilisation "gestion des livres " (admin).....	15
Figure 4 : diagramme de cas d'utilisation "gestion des livres " (lecteur).....	15
Figure 5 : diagramme de cas d'utilisation "gestion des utilisateurs " (admin)	16
Figure 6: diagramme de cas d'utilisation "gestion des utilisateurs " (lecteur)	16
Figure 7 : diagramme de cas d'utilisation "gestion des réclamations " (admin)	17
Figure 8 : diagramme de cas d'utilisation "gestion des catégories " (admin).....	17
Figure 9 : diagramme de cas d'utilisation "gestion des catégories " (lecteur).....	18
Figure 10 : diagramme de cas d'utilisation "gestion des abonnements ".....	18
Figure 11 : diagramme de cas d'utilisation "gestion de profile ".....	19
Figure 12 : diagramme de séquence de cas d'utilisation « authentifier ».....	23
Figure 13 : 2. Diagramme de séquence : cas d'utilisation « Ajouter un livre ».....	24
Figure 14 : Diagramme de séquence : cas d'utilisation « Ajouter une catégorie »	25
Figure 15 : Diagramme de classe : le module de gestion des utilisateurs	26
Figure 16: Diagramme de classe : le module de gestion de profil	27
Figure 17 : Diagramme de classe : le module de gestion des Livres.....	28
Figure 18 : Diagramme de classe : le module de gestion des catégories	29
Figure 19 : Diagramme de classe : le module de gestion des réclamations	30
Figure 20 : Diagramme de classe : le module de gestion des abonnements.....	31
Figure 21 : Diagramme d'activité "authentifier"	32
Figure 22 : Diagramme d'activité "ajouter livre"	32

Figure 23 : architecture de projet	33
Figure 24 : Interface d'authentification	40
Figure 25 : Interface d'enregistrement	40
Figure 26:interface de gestion des utilisateurs	41
Figure 27 : interface d'ajoute d'un utilisateur	41
Figure 28 : suppression d'un utilisateur	42
Figure 29 : interface de gestion des livres.....	42
Figure 30 : suppression d'un livre	43
Figure 31 : interface d'ajoute d'un livre	43
Figure 32:interface de gestion des réclamations	44
Figure 33 : interface de gestion des catégories	44
Figure 34 : inteface d'ajoute d'une catégorie	45
Figure 35 : interface de statistique	45
Figure 36 : interface de tous les livres.....	46
Figure 37 : menu de navigation et d'utilisateur	46
Figure 38 : interface des categories	47
Figure 39 : profile de lecteur	47
Figure 40 : les livres favoris	48
Figure 41 : les livres à lire plus tard	48
Figure 42 : intreface d'affichage des amis.....	49
Figure 43 : interface de profile d'un ami	49
Figure 44 : barre de recherche	50
Figure 45 : interface de recherche	50
Figure 46 : interface pour contacter le supports	51
Figure 47:écran de bienvenue.....	51

Figure 48:écran de chargement	51
Figure 49:écran de création de compte	52
Figure 50:écran de connexion	52
Figure 51:écran d'accueil 2.....	53
Figure 52:écran d'accueil 1.....	53
Figure 53: écran des catégories	54
Figure 54:écran des livres par catégorie	54
Figure 55:écran des livres après la recherche.....	55
Figure 56:écran des livres d'une autre catégorie.....	55
Figure 57:écran d'un livre sélectionné détaillé	56
Figure 58:écran d'un livre sélectionné	56
Figure 59:menu de l'application.....	57
Figure 60:écran des évaluations des utilisateur sur un livre.....	57
Figure 61:écran des livres à lire plus tard	58
Figure 62:écran de profile d'un utilisateur.....	58
Figure 63:écran de la liste des amis	59
Figure 64:écran des livres favoris.....	59

Bibliographie

NodeJS :

<https://nodejs.org/>

ReactJS :

<https://reactjs.org/>

Redux :

<https://redux.js.org/>

Flutter :

<https://flutter.dev/>

Dart :

<https://dart.dev/tutorials>