



**ECOLE MAROCAINE DES
SCIENCES DE L'INGENIEUR**

Membre de

HONORIS UNITED UNIVERSITIES



UNITED UNIVERSITIES

Filière : Génie Informatique et Réseaux

**Option : MIAGE (Méthodes Informatiques Appliquées à la
Gestion des Entreprises)**

Projet de Fin d'Année :



Find2Read

**Bibliothèque Digitale
Intelligente avec
assistant AI :**

Réalisé par :

Saad Bendahou
Hamza Elbouanani

Membre de jury :

Mme Intissar Salhi (encadrante)
Mr Zakrani Abdelali (Examineur)

Soutenu le: 27/05/2025

Année universitaire : 2024 – 2025

Dédicaces

Nous dédions ce projet à nos familles respectives, véritables piliers de notre parcours académique. Leur soutien inconditionnel, leur patience et leurs encouragements constants ont été essentiels tout au long de cette aventure. Grâce à leur amour et à leur confiance en nous, nous avons pu franchir chaque étape avec sérénité et détermination. Ce travail leur est dédié en reconnaissance de tous les sacrifices qu'ils ont consentis pour nous permettre de poursuivre nos études dans les meilleures conditions.

Nous adressons également cette dédicace à toutes les personnes qui, de près ou de loin, ont contribué à nourrir notre curiosité intellectuelle, à nous inspirer, et à croire en la valeur de notre projet. À nos amis, qui ont su nous remonter le moral pendant les moments de doute, et à nos enseignants, qui ont semé en nous la passion pour l'informatique et le développement de solutions utiles à la société.

Remerciements

Nous souhaitons exprimer nos plus sincères remerciements à Mme Intissar Salhi, notre encadrante, pour son accompagnement attentif, ses conseils précieux et sa rigueur scientifique qui ont marqué chaque étape de notre projet. Sa disponibilité et son engagement ont été des facteurs déterminants dans l'aboutissement de notre travail. Elle a su nous orienter, nous corriger, et nous motiver avec bienveillance et professionnalisme.

Nos remerciements s'adressent également à l'ensemble du corps enseignant de notre établissement, pour la richesse des enseignements reçus, ainsi qu'à tous ceux qui, de manière directe ou indirecte, ont apporté leur contribution à la réussite de ce projet. Enfin, nous remercions nos familles et notre binôme pour l'esprit de collaboration, la coordination harmonieuse et le respect mutuel qui ont facilité notre travail en équipe.

Table des matières

Dédicaces	2
Remerciements.....	3
Liste des figures	6
Liste des tableaux	6
Introduction générale.....	7
Chapitre 1 : Présentation du cadre de projet	9
Introduction.....	9
Présentation de l'environnement de travail	9
Étude de l'existant	10
Choix du modèle de développement	10
Planning prévisionnel	11
Conclusion	11
Chapitre 2 : Spécification des besoins.....	13
Introduction.....	13
Besoins fonctionnels.....	13
Besoins non fonctionnels	14
Cas d'utilisation et acteurs.....	14
Chapitre 3 : Conception du système.....	19
Introduction.....	19
Modélisation dynamique.....	19
Diagrammes de séquences	19
Modélisation statique.....	21
Diagramme de classes.....	21
Architecture logicielle détaillée	24
Conclusion	26
Chapitre 4 : Réalisation du système	28
Introduction.....	28
Environnement de développement	28
Environnement matériel.....	28
Environnement logiciel.....	28

Principales interfaces de l'application.....	30
Conclusion	34
Conclusion Général :	35

Liste des figures

Figure 3 : Diagramme de cas d'utilisation global	16
Figure 4:Diagramme de séquence (création de nouvelle)	19
Figure 5:Diagramme de séquence (authentification)	20
Figure 6:Diagramme de séquence (Assistant ai).....	20
Figure 7: Diagramme de classes de l'application	23
Figure 8: Architecture logiciel	24
Figure 9:SpringBoot	24
Figure 10:Thymeleaf	24
Figure 11: Mongo DB	25
Figure 12: Phi-1	25
Figure 13:Intellij.....	29
Figure 14: Postman.....	29
Figure 15: GitHub	29
Figure 16:PlantUML	29
Figure 17: page d'accueil.....	30
Figure 18: page Catalogue	31
Figure 19: page détail nouvelle	31
Figure 20: assistant Ai	32
Figure 21: classement par rating	33
Figure 22: gestion profil	33

Liste des tableaux

Table 1:Planning prévisionnel	11
-------------------------------------	----

Introduction générale

Dans un monde où les technologies évoluent rapidement, la lecture ne se limite plus aux supports traditionnels. L'essor du numérique a profondément modifié les comportements des lecteurs, créant un besoin croissant pour des plateformes modernes, interactives et intelligentes. C'est pour répondre à ce besoin que nous avons développé **Find2Read**, une application web de bibliothèque digitale intelligente.

Le projet vise à développer un environnement où chaque utilisateur peut non seulement consulter et lire des œuvres en ligne, mais aussi publier ses propres écrits, interagir avec d'autres lecteurs, et recevoir des recommandations personnalisées. L'intégration d'un assistant conversationnel, basé sur un modèle de langage moderne (Phi de Microsoft), permet de guider le lecteur, répondre à ses questions, et l'orienter dans ses choix littéraires.

La problématique à laquelle nous avons souhaité répondre est claire : **comment rendre la lecture numérique plus accessible, collaborative et intelligente ?** Les solutions existantes se concentrent souvent soit sur la lecture passive, soit sur des fonctionnalités très restreintes. Notre objectif a été de concevoir une application complète, dynamique et évolutive, pensée pour les lecteurs d'aujourd'hui.

Dans ce rapport, nous présenterons d'abord le cadre de réalisation du projet, puis nous détaillerons les spécifications fonctionnelles et techniques. Ensuite, nous aborderons la phase de conception avec les différents diagrammes UML, avant de passer à la réalisation technique, illustrée par les interfaces développées. Nous concluons enfin par une analyse critique et des perspectives d'évolution.



C

hapitre

Présentation du cadre de projet

1

Chapitre 1 : Présentation du cadre de projet

Introduction

Dans ce premier chapitre, nous allons présenter le cadre général dans lequel s'inscrit notre projet de fin d'année intitulé **Find2Read**. Ce projet a été conçu pour répondre à une problématique réelle : aider les lecteurs à trouver des livres adaptés à leurs préférences grâce à l'intelligence artificielle et aux technologies web modernes. L'objectif est de poser les fondations de notre démarche en exposant le contexte académique, la situation actuelle, les limites rencontrées dans les solutions existantes, ainsi que les choix que nous avons opérés pour mener à bien ce travail.

Nous aborderons également la planification prévisionnelle du projet, organisée sur deux mois, et le modèle de développement retenu. Ces éléments sont essentiels pour structurer le processus de réalisation de notre application, garantir une évolution cohérente et organiser efficacement le travail en équipe.

Présentation de l'environnement de travail

Le projet Find2Read a été développé dans le cadre académique de notre formation d'ingénieur en informatique, sous la direction de notre encadrante, Mme **Intissar Salhi**. Aucun environnement professionnel externe (entreprise ou stage) n'a été impliqué. Ce choix nous a permis de bénéficier d'une grande liberté pédagogique, tout en étant encadrés de manière rigoureuse dans le respect des objectifs du module de Projet de Fin d'Année.

Nous avons travaillé en binôme, ce qui a favorisé une répartition claire des tâches, un esprit collaboratif et une efficacité renforcée. Grâce à cette organisation, nous avons pu concentrer nos efforts sur des aspects variés du projet tels que la conception, le développement back-end et front-end, l'intégration du chatbot, la gestion de base de données, et la documentation. L'ensemble de ces tâches ont été menées en autonomie, tout en bénéficiant des retours réguliers de notre encadrante.

.

Étude de l'existant

À l'heure actuelle, les plateformes de lecture en ligne, bien que nombreuses, proposent principalement une expérience figée et standardisée. Elles se contentent souvent de fonctionnalités basiques comme la recherche par titre ou auteur, la classification par genre, ou encore des classements généraux. Peu d'entre elles intègrent de véritables mécanismes de recommandation personnalisée, encore moins via une interface conversationnelle.

Cette limitation nuit à l'expérience utilisateur, surtout pour les lecteurs occasionnels ou ceux qui souhaitent découvrir de nouveaux horizons littéraires sans avoir une idée précise de ce qu'ils recherchent. Il devient alors difficile de parcourir efficacement un vaste catalogue de livres ou d'enrichir ses choix de lecture de manière intuitive et interactive.

Choix du modèle de développement

Nous avons opté pour le modèle en spirale, qui combine les avantages des approches incrémentales et itératives. Ce modèle nous a permis de progresser par cycles successifs, chacun comportant une phase de conception, de développement, de test et de validation. Il favorise la gestion des risques, l'amélioration continue et la flexibilité dans l'intégration de nouvelles idées au fil du projet.

Ce choix s'est avéré pertinent dans un contexte académique, où le retour constant de l'encadrante et la nécessité de démontrer des versions fonctionnelles progressives étaient essentiels. Il nous a également permis de corriger certaines fonctionnalités après test et d'optimiser l'expérience utilisateur au fur et à mesure de l'évolution de l'application.

Planning prévisionnel

Table 1: Planning prévisionnel

	mars	Avril				Mai			
Étape	4	1	2	3	4	1	2	3	4
Étude préalable	X	X							
Conception			X	X					
Réalisation					X	X	X	X	
Tests & Validation								X	X

Conclusion

Ce premier chapitre a permis de poser les bases de notre projet Find2Read, en expliquant son origine, le contexte académique dans lequel il a été réalisé, et les limites des systèmes existants. Nous avons exposé la solution que nous proposons, structurée autour d'un assistant conversationnel intelligent, et détaillé les choix techniques et organisationnels retenus.

Dans les chapitres suivants, nous approfondirons la définition des besoins fonctionnels et non fonctionnels, la conception du système, ainsi que sa mise en œuvre concrète.

A large red square graphic containing the white text 'C hapitre 2'. The 'C' is very large and the word 'hapitre' is smaller, positioned to the right of the 'C'. The number '2' is also large and positioned below the word 'hapitre'.

C hapitre 2

Spécification des besoins :

Chapitre 2 : Spécification des besoins

Introduction

Avant d'entamer toute phase de conception ou de développement, il est essentiel de comprendre et formaliser les besoins auxquels doit répondre le système. Ce chapitre a pour objectif de décrire les exigences fonctionnelles et non fonctionnelles de l'application Find2Read. Ces exigences constituent le socle sur lequel repose la future architecture de notre solution.

La bonne compréhension de ces besoins nous a permis d'assurer une cohérence entre les attentes de l'utilisateur final, les fonctionnalités développées, et les choix technologiques effectués. Nous détaillerons également les différents acteurs impliqués dans l'utilisation du système, ainsi que les cas d'utilisation principaux à travers un diagramme UML.

Besoins fonctionnels

Les besoins fonctionnels définissent les actions que le système doit être capable d'accomplir pour satisfaire l'utilisateur. Dans le cadre de notre projet Find2Read, ces besoins couvrent à la fois les interactions classiques avec une bibliothèque digitale et les fonctions innovantes liées à l'assistant conversationnel. Voici les principales fonctionnalités attendues :

- **Authentification des utilisateurs** : le système doit permettre aux utilisateurs de créer un compte, de se connecter, de modifier leur profil, voire de supprimer leur compte. L'authentification sécurisée constitue la base de l'accès aux services personnalisés.
- **Recherche et consultation de livres** : les utilisateurs peuvent consulter le catalogue, trier les livres par catégorie, auteur, genre ou mot-clé, et visualiser les détails de chaque livre (titre, résumé, auteur, couverture, etc.).
- **Interaction avec le chatbot assistant** : le cœur innovant du système repose sur la possibilité d'interagir avec un assistant conversationnel. L'utilisateur peut exprimer librement ses envies, poser des questions, et recevoir des suggestions de lecture précises basées sur les données de notre bibliothèque.

Besoins non fonctionnels

Les besoins non fonctionnels sont tout aussi importants car ils garantissent la qualité globale du système au-delà de ses simples fonctionnalités. Dans le cas de Find2Read, nous avons identifié plusieurs contraintes techniques et qualitatives :

- **Ergonomie et simplicité d'utilisation** : l'interface doit être claire, intuitive et accessible à tous types d'utilisateurs, même ceux qui ne sont pas à l'aise avec les outils numériques.
- **Réactivité du système** : les temps de réponse doivent être rapides, en particulier lors des requêtes au chatbot, pour ne pas nuire à la fluidité de l'interaction.
- **Sécurité des données** : les informations utilisateurs doivent être protégées. Les mots de passe sont chiffrés, et aucun contenu personnel ne doit être partagé à l'extérieur.
- **Scalabilité** : la base de données doit pouvoir être enrichie facilement avec de nouveaux livres sans altérer les performances.
- **Accessibilité multiplateforme** : l'application web doit être compatible avec les principaux navigateurs (Chrome, Firefox, Edge...) et responsive pour s'adapter aux écrans mobiles et tablettes.

Ces besoins ont été pris en compte dès la phase de conception afin d'orienter nos choix techniques (frameworks, base de données, architecture REST, etc.).

Cas d'utilisation et acteurs

1. Visiteur

Le visiteur est un utilisateur non authentifié de la plateforme Find2Read. Il a un accès limité aux fonctionnalités, principalement en lecture seule. Ce rôle peut :

- Parcourir la bibliothèque de nouvelles disponibles,
- Rechercher une nouvelle spécifique,
- Lire une page ou une nouvelle complète,
- Consulter le classement des nouvelles selon leur popularité,
- Interagir avec l'assistant IA pour obtenir de l'aide, des suggestions ou des résumés.

Ce profil est essentiel pour offrir un accès libre à la plateforme tout en incitant les utilisateurs à s'inscrire pour profiter de fonctionnalités avancées.

2. Auteur

L'auteur est un utilisateur inscrit et connecté, ayant des droits supplémentaires liés à la contribution de contenu. En plus des fonctionnalités offertes au visiteur, l'auteur peut :

- Écrire une nouvelle page ou une œuvre littéraire,
- Publier sa page sur la plateforme,
- Modifier les informations relatives à ses publications,
- Noter et commenter les œuvres d'autres auteurs,
- Consulter les retours des lecteurs sur ses productions,
- Accéder à l'historique de lecture personnel.

L'auteur enrichit la base de données collaborative de la plateforme en y ajoutant du contenu original, ce qui fait de lui un acteur central du projet.

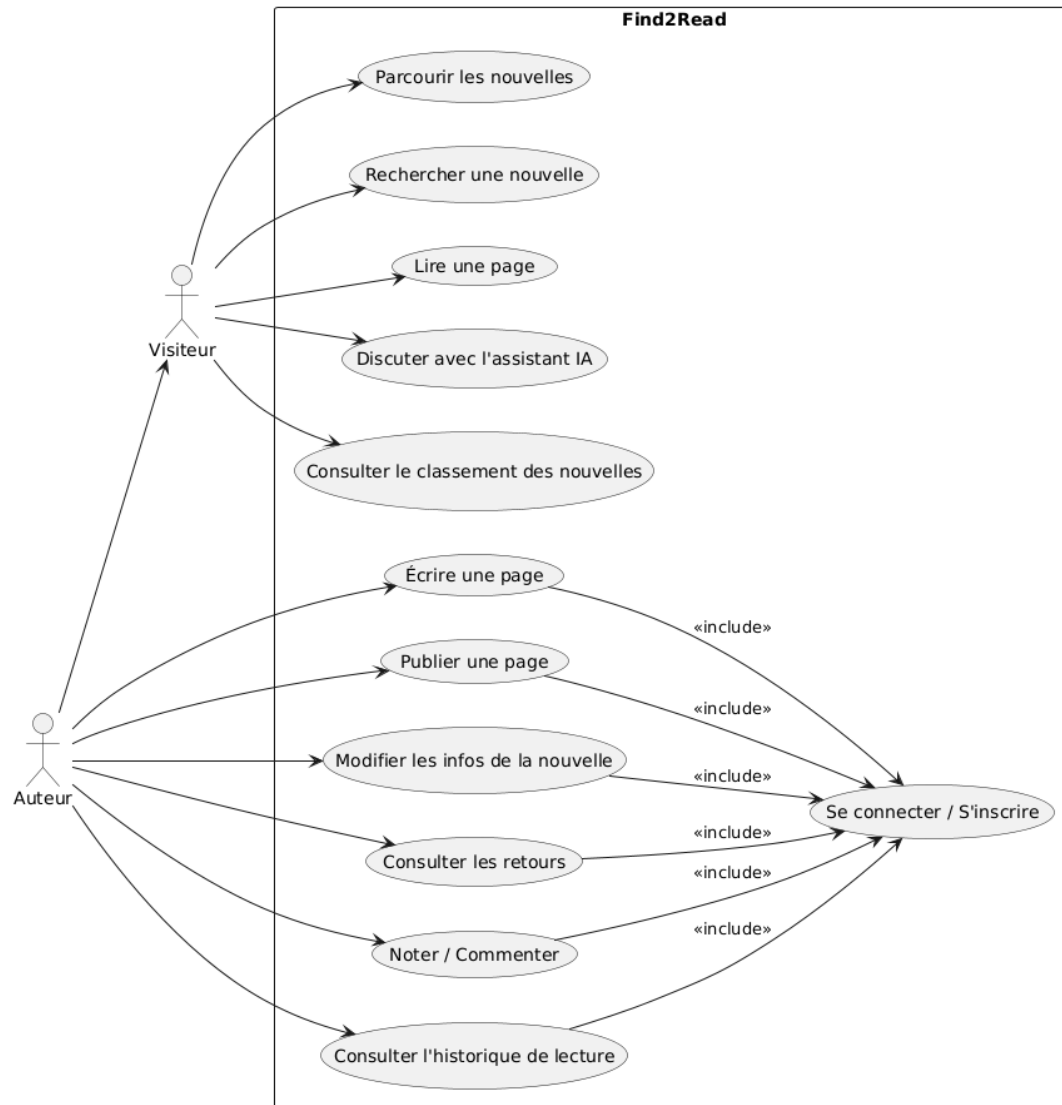


Figure 1 : Diagramme de cas d'utilisation global

Le diagramme suivant représente les cas d'utilisation principaux de l'application Find2Read. Il met en évidence les différentes fonctionnalités offertes à chaque type d'acteur.

Conclusion

Ce deuxième chapitre nous a permis de formaliser les attentes fonctionnelles et non fonctionnelles du système. Cette étape est cruciale dans tout projet informatique, car elle constitue une référence commune pour tous les acteurs du développement. Les besoins exprimés ont été directement traduits en composants logiciels dans les chapitres suivants.

C

hapitre

3

Conception du
système :

Chapitre 3 : Conception du système

Introduction

Après avoir identifié et formalisé les besoins du système, la phase de conception vise à imaginer une solution logicielle cohérente, modulaire et évolutive. Cette étape constitue le pont entre la spécification et la réalisation. Elle permet de visualiser comment les différents composants du système interagissent entre eux, de déterminer les structures de données nécessaires, et de définir les interfaces utilisateur et techniques.

Dans ce chapitre, nous présentons la conception statique du système à travers un **diagramme de classes**, ainsi que la **structure logique** de l'application basée sur l'architecture MVC (Modèle-Vue-Contrôleur). Nous décrivons également de manière détaillée l'organisation des composants techniques, les choix de structuration du backend et les interactions entre les modules front-end et back-end.

Modélisation dynamique

Diagrammes de séquences

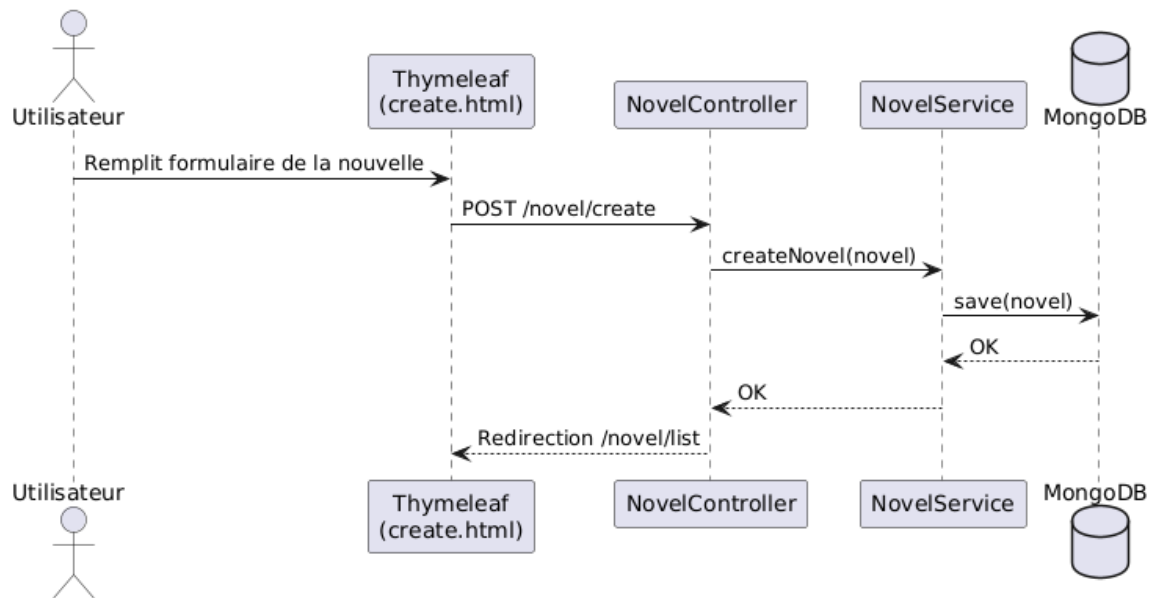


Figure 2: Diagramme de séquence (création de nouvelle)

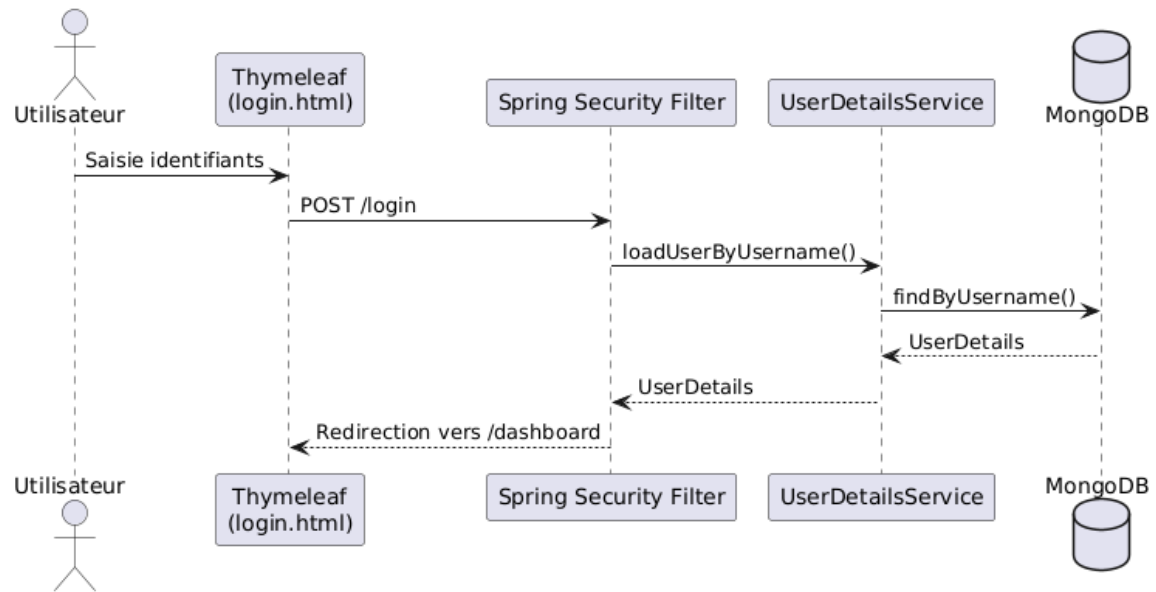


Figure 3:Diagramme de séquence (authentification)

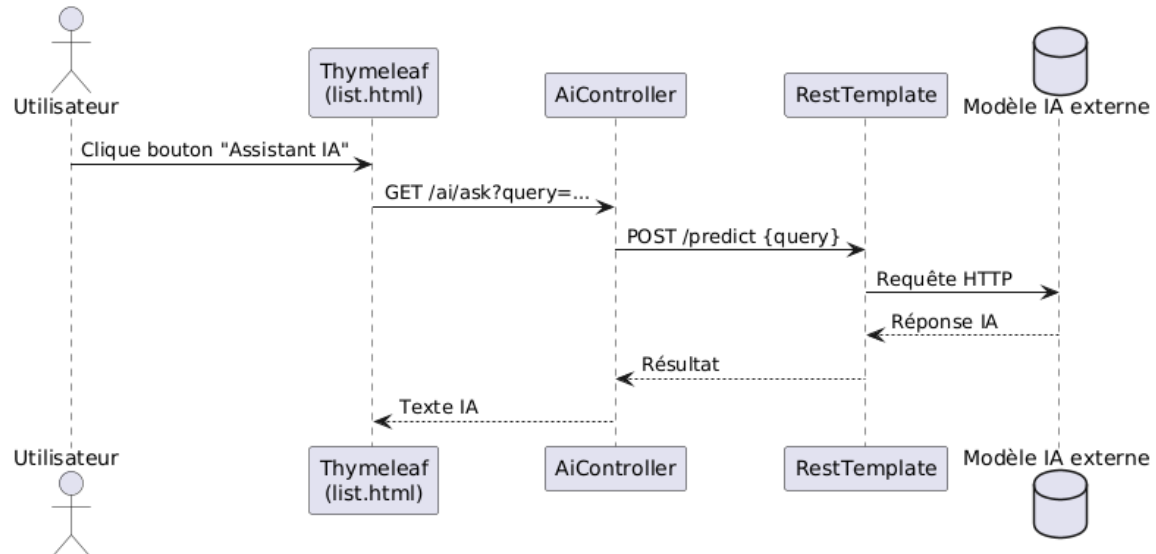


Figure 4:Diagramme de séquence (Assistant ai)

Modélisation statique

Diagramme de classes

Classe User

Cette classe représente les utilisateurs inscrits sur la plateforme. Elle contient des informations personnelles et des méthodes permettant de personnaliser l'expérience de lecture.

Attributs principaux :

- id, username, email, password, role

Méthodes principales :

- incrementViewCount(), addReadingTime()
- updateCategoryPreference(), setProfilePicture()
- addViewedNovel(), addNovelToLibrary(), removeNovelFromLibrary()

Relations :

- Un utilisateur **peut noter** plusieurs nouvelles (Rating)
- Un utilisateur **peut commenter** plusieurs nouvelles (Comment)
- Un utilisateur **peut écrire** plusieurs nouvelles (Novel)

Classe Novel

Elle représente une œuvre littéraire publiée sur la plateforme. Chaque nouvelle contient un ensemble de pages, de commentaires et de notes.

Attributs principaux :

- id, title, author, genre, summary, coverImage

Méthodes principales :

- getAverageRating(), setImageData()

Relations :

- Une nouvelle **est écrite par** un auteur (User)
- Elle **contient** plusieurs pages (Page)
- Elle **reçoit** plusieurs commentaires (Comment)
- Elle **reçoit** plusieurs notes (Rating)

Classe Page

Chaque nouvelle est découpée en pages. Cette classe contient le numéro et le contenu de chaque page.

Attributs principaux :

- id, novelId, number, content

Relations :

- Une page appartient à **une seule nouvelle**

Classe Comment

Elle permet aux utilisateurs de réagir aux nouvelles. Chaque commentaire est lié à un utilisateur et une nouvelle.

Attributs principaux :

- id, novelId, userId, username, content

Méthodes :

- getReplies(), setReplies()

Relations :

- Un commentaire est **reçu par** une nouvelle
- Un utilisateur peut **écrire** plusieurs commentaires

Classe Rating

Elle permet aux utilisateurs de noter les nouvelles à l'aide d'une valeur entière.

Attributs principaux :

- id, novelId, userId, value

Relations :

- Un utilisateur **note** une nouvelle
- Une nouvelle peut **recevoir** plusieurs notes

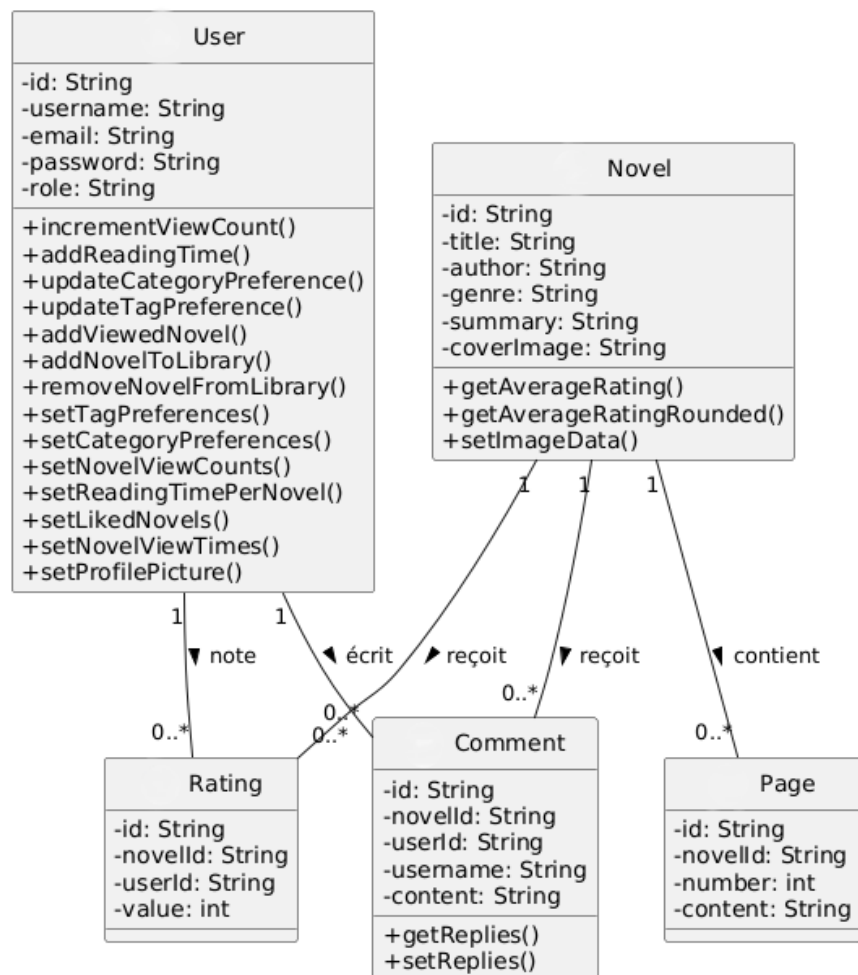


Figure 5: Diagramme de classes de l'application

Architecture logicielle détaillée

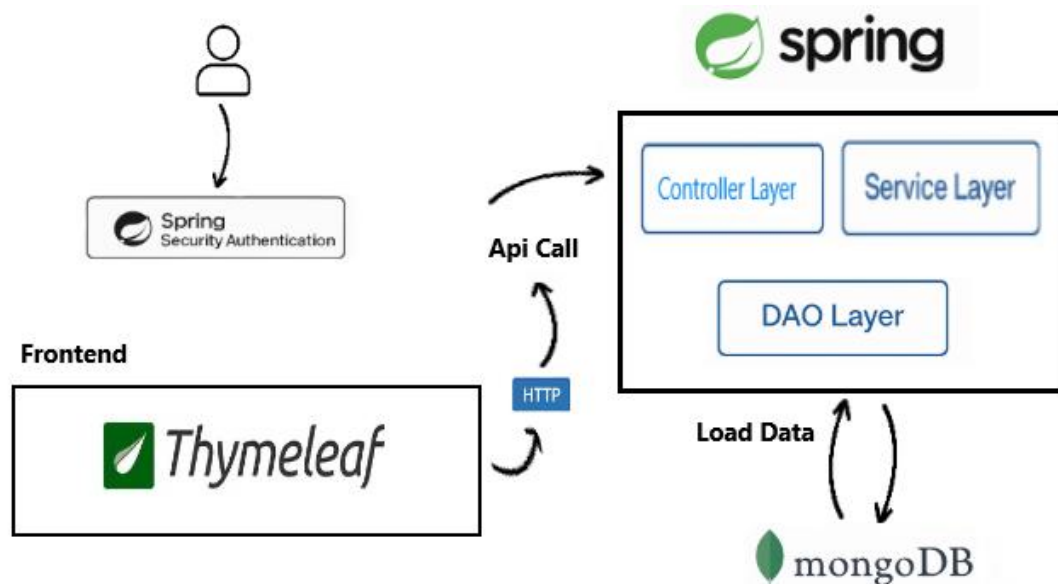


Figure 6: Architecture logiciel

Back-end/Front-End – Spring Boot + Thymeleaf



Le back-end est développé en **Java** à l'aide de **Spring Boot**, un framework robuste qui facilite la création d'applications web évolutives.



Nous avons adopté **Thymeleaf** comme moteur de rendu côté serveur pour générer dynamiquement les pages HTML. Cela nous a permis de créer une interface web fluide tout en gardant une cohérence avec le modèle de données.

- un **contrôleur Spring** (@Controller) pour gérer les requêtes,
- un **service** pour centraliser la logique métier,
- un **DAO** avec **Spring Data MongoDB** pour l'accès aux données.

Spring Boot a aussi permis une intégration rapide avec MongoDB et la mise en place de **tests unitaires** pour les principales fonctionnalités.



Figure 9: Mongo DB

Base de données – MongoDB

Nous avons choisi **MongoDB**, une base de données NoSQL orientée documents, pour sa flexibilité et sa capacité à gérer des structures de données variées. Elle convient parfaitement aux objets complexes tels que les romans composés de pages, commentaires, notes, etc.

La base est structurée autour de collections principales :

- users : pour les données des utilisateurs et leurs préférences,
- novels : contenant les métadonnées et pages des romans,
- comments et ratings : pour les interactions sociales.

La structure sans schéma strict de MongoDB nous a permis d'évoluer facilement pendant le développement et d'intégrer de nouvelles fonctionnalités sans migration complexe.



Figure 10: Phi-1

Assistant IA – Phi (Microsoft) avec système RAG

L'un des éléments différenciateurs de notre application est l'intégration d'un **assistant intelligent**, basé sur **Phi**, un modèle de langage développé par **Microsoft**. Ce modèle est couplé avec une logique **RAG (Retrieval-Augmented**

Generation) pour assurer la pertinence des réponses.

Conclusion

La phase de conception a représenté une étape fondamentale dans notre projet. Elle nous a permis de structurer de manière rigoureuse l'ensemble du système, de définir clairement les relations entre les entités, et de poser les bases techniques du développement.

Les choix technologiques retenus – **Spring Boot** pour le back-end, **Thymeleaf** pour le rendu dynamique des vues, **MongoDB** pour la base de données NoSQL, et **Phi de Microsoft** pour l'intelligence artificielle – ont été guidés par leur souplesse, leur compatibilité mutuelle, et leur capacité à répondre aux besoins fonctionnels de notre application.

Grâce à cette conception bien pensée, la transition vers la phase de développement s'est faite de manière fluide, avec une vision claire de la structure et des objectifs du système.



Réalisation Du Système :

Chapitre 4 : Réalisation du système

Introduction

Ce chapitre présente la concrétisation de notre travail à travers la mise en œuvre technique de l'application Find2Read. Après avoir défini les besoins et établi une conception rigoureuse, nous sommes passés à la phase de développement. Cette étape a nécessité des choix d'outils, une bonne organisation des tâches, et une série de tests continus pour garantir un produit final fonctionnel et stable.

Nous allons donc détailler ici l'environnement matériel et logiciel dans lequel le projet a été développé, puis présenter les principales interfaces de l'application. Cette phase a été marquée par un travail de collaboration soutenu, où les deux membres du groupe se sont répartis les responsabilités de manière efficace.

Environnement de développement

Environnement matériel

Le projet a été entièrement conçu en local, sans recours à des serveurs distants ni à des plateformes cloud, afin de garantir une autonomie totale sur l'infrastructure de test.

Environnement logiciel

Pour mener à bien le développement de notre application **Find2Read**, nous avons mobilisé une variété d'outils logiciels adaptés aux différentes phases du projet :

- **Spring Boot** : Framework Java utilisé pour développer le back-end de l'application. Il facilite la structuration du projet en architecture **MVC** et permet d'exposer des APIs REST robustes.
- **Thymeleaf** : Moteur de template côté serveur, intégré au back-end Spring Boot, pour générer dynamiquement les pages HTML. Il offre une liaison directe entre les données Java et les vues HTML sans JavaScript.
- **MongoDB** (local et **MongoDB Atlas**) : Base de données **NoSQL orientée documents** utilisée pour stocker les informations sur les utilisateurs, les romans, les commentaires, les notes, et les préférences. Sa flexibilité schéma-libre a permis d'adapter facilement les structures aux besoins évolutifs.
- **Phi (Microsoft)** avec **RAG (Retrieval-Augmented Generation)** : Moteur d'intelligence artificielle utilisé pour créer un assistant conversationnel spécialisé dans l'aide à la lecture. Les réponses générées sont basées uniquement sur des documents internes et filtrées pour rester dans le cadre contextuel.



Figure 11: IntelliJ

- **IntelliJ IDEA** : IDE principal utilisé pour le développement du back-end en Java avec Spring Boot.



Figure 12: Postman

- **Postman** : Outil de test des APIs REST pour simuler les échanges entre le front-end et le back-end.



Figure 13: GitHub

- **GitHub** : Plateforme de gestion de version utilisée pour héberger le code source, suivre les modifications, et assurer une collaboration fluide entre les membres de l'équipe.



Figure 14: PlantUML

- **PlantUML** : Outil de modélisation UML utilisé pour générer les **diagrammes de classes, cas d'utilisation, de séquence et d'architecture**. Grâce à sa syntaxe textuelle simple, il a facilité l'intégration rapide des diagrammes dans le rapport et leur modification continue au fil de l'évolution du projet.

Principales interfaces de l'application

1. Page d'accueil

Cette interface présente une introduction à la plateforme et donne accès aux fonctionnalités publiques :

- Champ de recherche de romans par titre ou mot-clé.
- Liste des romans populaires ou récemment consultés.
- Boutons : *S'inscrire*, *Se connecter*, *Catalogue*, *Chatbot*.

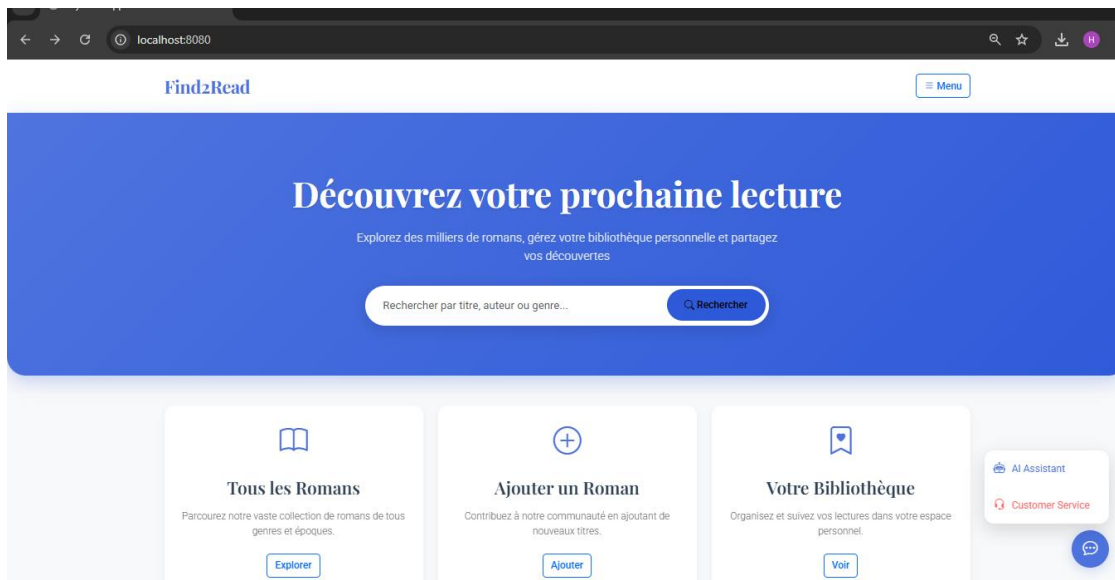


Figure 15: page d'accueil

2. Catalogue des romans

Permet de parcourir l'ensemble des romans enregistrés dans la base :

- Affichage sous forme de cartes avec image, titre, genre.
- Filtres dynamiques par genre, auteur, popularité.
- Pagination des résultats.

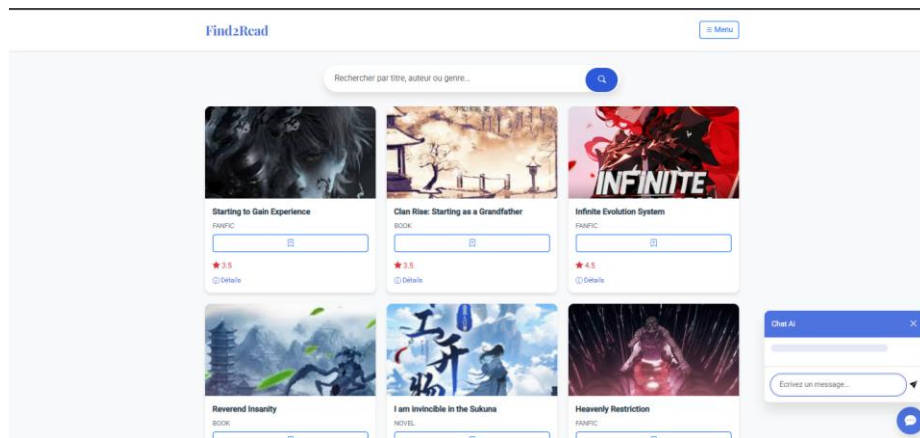


Figure 16: page Catalogue

3. Détail d'un roman (fiche de lecture)

Affiche les détails d'un roman sélectionné :

- Titre, résumé, auteur, date d'ajout.
- Liste des pages avec lecture possible page par page.
- Boutons : *Lire*, *Noter*, *Commenter*, *Ajouter à ma bibliothèque*.
- Notes moyennes et avis d'autres lecteurs.

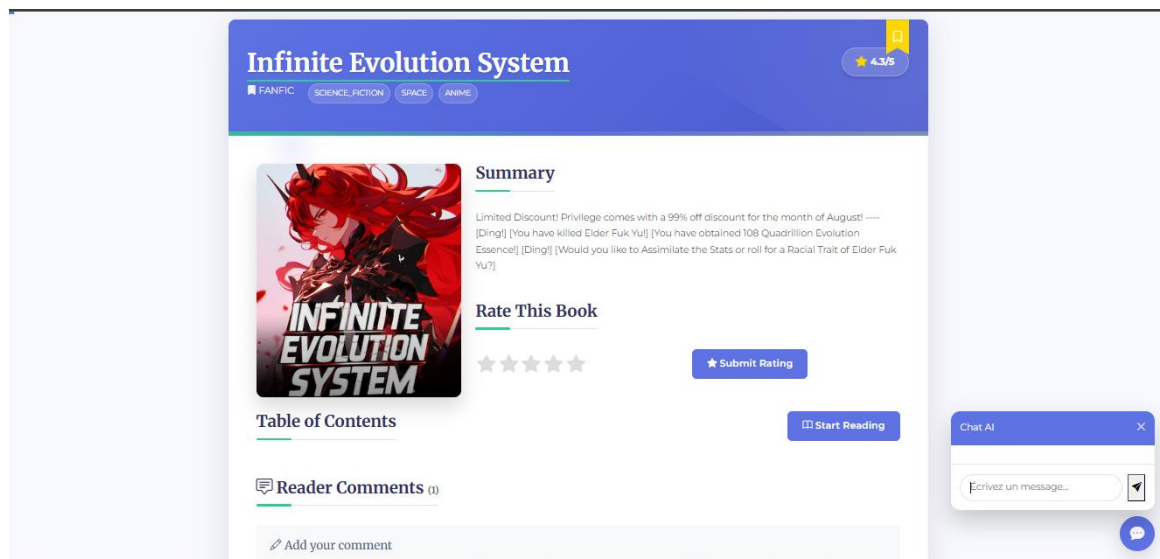


Figure 17: page détail nouvelle

4. Interface de l'assistant IA

Permet aux utilisateurs d'interagir avec l'assistant conversationnel pour :

- Poser des questions sur le contenu d'un roman en cours de lecture.
- Recevoir des définitions ou résumés liés aux passages lus.
- Obtenir une aide pour la navigation ou des recommandations personnalisées.

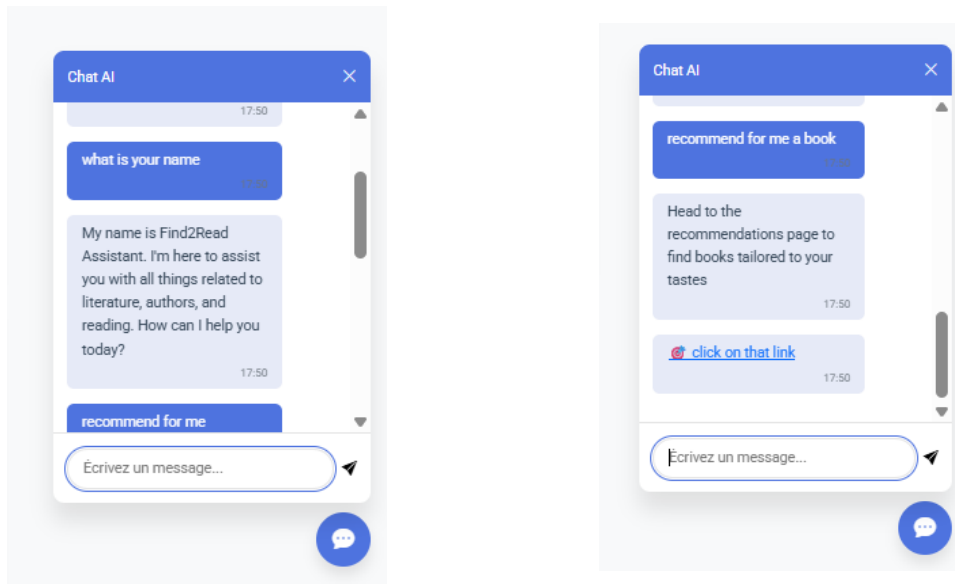


Figure 18: assistant Ai

5. Classement des romans

Permet aux utilisateurs de découvrir les romans les mieux notés et les plus consultés, en se basant sur plusieurs critères :

- Classements dynamiques selon les notes moyennes.
- Affichage sous forme de tableau ou de vignettes.

Figure 19 shows a 'Novel Rankings' page. At the top, there are buttons for 'Back to Home' and 'My Profile'. The main content is a table with 5 columns: Rank, Cover, Title, Rating, and Votes. The table lists 8 novels, with the first one, 'Sign in to the World-Destroying', having the highest rating of 4.5 and 2 votes.

Rank	Cover	Title	Rating	Votes
1		Sign in to the World-Destroying	4.5	2
2		Infinite Evolution System	4.3	3
3		Reverend Insanity	4.0	2
4		I am invincible in the Sukuna	4.0	2
5		Heavenly Restriction	4.0	1
6		I survive	4.0	1
7		Starting to Gain Experience	3.5	2
8		Clan Rise: Starting as a Grandfather	3.5	2

Figure 19: classement par rating

6. Interface du profil utilisateur

Permet à l'utilisateur de gérer ses informations personnelles et d'accéder rapidement à ses contenus :

- Boutons de navigation : Bibliothèque, Historique, Mes Romans, Nouveau Roman.
- Page claire et intuitive facilitant la gestion de l'espace personnel.

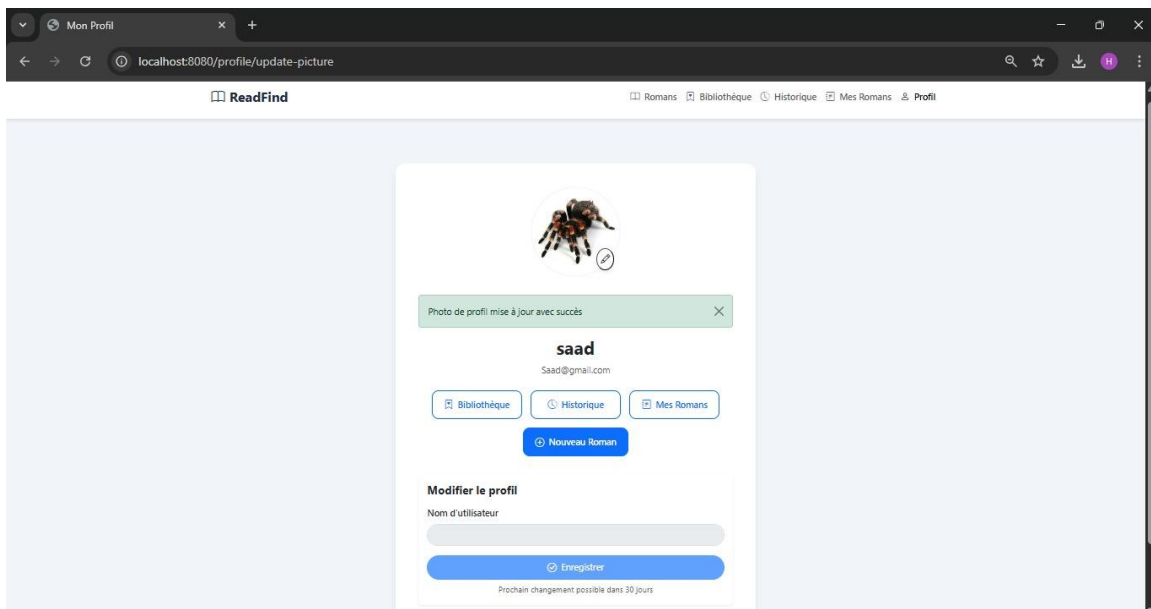


Figure 20: gestion profil

Conclusion

La phase de réalisation a été décisive dans la concrétisation de notre projet Find2Read. Grâce à une préparation rigoureuse, une conception bien pensée et des choix technologiques adaptés, nous avons pu implémenter toutes les fonctionnalités majeures attendues. L'intégration entre le front-end React, le back-end Spring Boot et la base MongoDB s'est déroulée de manière fluide.

Nous avons également réussi à intégrer un assistant intelligent fonctionnel, capable de comprendre les requêtes utilisateur et de proposer des suggestions pertinentes. Cette réussite démontre la faisabilité technique du concept, et ouvre la voie à des évolutions futures plus ambitieuses, comme l'ajout de recommandations croisées ou l'enrichissement de la base par des utilisateurs.

Dans le chapitre suivant, nous concluons notre rapport par une synthèse des résultats, des réflexions personnelles, ainsi que des perspectives d'amélioration.

Conclusion Général :

Ce projet de fin d'année nous a offert l'opportunité de concevoir et développer une application web innovante dans le domaine de la lecture numérique. **Find2Read** se distingue par son approche intelligente qui combine une bibliothèque littéraire collaborative avec un assistant conversationnel contextuel, conçu pour enrichir l'expérience de lecture.

Tout au long de ce projet, nous avons suivi une démarche rigoureuse, en commençant par l'analyse des besoins et la modélisation UML, jusqu'à la mise en œuvre technique et fonctionnelle de la solution. L'adoption de l'architecture **MVC** avec **Spring Boot** et **Thymeleaf**, associée à l'usage de **MongoDB** comme base de données NoSQL, nous a permis de construire une plateforme robuste, modulaire et évolutive.

L'intégration de l'intelligence artificielle, via **le modèle Phi de Microsoft** couplé à un système **RAG (Retrieval-Augmented Generation)**, constitue un apport innovant et différenciateur. Elle offre aux utilisateurs une assistance contextuelle, des réponses personnalisées et des recommandations pertinentes, tout en restant limitée au cadre littéraire.

Ce projet nous a permis de mettre en pratique de nombreuses compétences techniques (programmation, modélisation, intégration IA) et de renforcer notre capacité à travailler en équipe, à gérer un planning agile et à documenter chaque étape de notre travail. Il ouvre également des perspectives d'évolution intéressantes, telles que l'enrichissement communautaire de la base de données, l'amélioration de l'IA ou l'extension vers des applications mobiles.

En définitive, **Find2Read** incarne une solution intelligente, collaborative et accessible, pensée pour transformer la lecture numérique en une expérience interactive et engageante.