République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique
Université Abderrahmane Mira
Faculté des Sciences Exactes
Département Informatique

Projet de Fin de cycle
en vue de l'obtention du diplôme Master professionnel en informatique
Option : Génie Logiciel (GL)

Thème

Plateforme avancée d'extraction de métadonnées et de recherche de doucuments alimentée par LLM

Présenté par :

Melle. BOUDRA Rania
Melle. BENAMOURA Nadira

Dirigé par :

M. BOUDRIES Abdelmalek

Soutenu le 30/06/2025 devant le jury composé de :

Président M. DJEBARI Nabil
Examinateur Mme. AIT ABDELOUHAB Karima
Examinateur Mme. CHIBANI Samia
Examinateur Mme. CHAABANE Sarah

Année universitaire : 2024/2025

❀ *Remerciements* ❀

Avant tout, nous remercions Dieu, le Tout-Puissant, de nous avoir donné la force, la patience et la volonté d'achever ce modeste travail.

Nous tenons à exprimer notre profonde gratitude à Monsieur BOUDRIES Abdelmalek, notre encadrant, pour sa disponibilité, ses conseils précieux et son accompagnement tout au long de ce projet.

Nos sincères remerciements vont également aux membres du jury pour avoir accepté d'évaluer ce travail et pour leurs remarques enrichissantes.

Nous remercions également Monsieur BENSLIMANE Samir, ainsi que tout le personnel du cabinet, pour leur aide, leur écoute et leur soutien pendant cette période.

Enfin, nous adressons un grand merci à notre famille et à nos amis pour leur soutien moral, leur encouragement et leur confiance, qui nous ont permis de mener à bien ce projet.

❀ *Dédicaces* ❀

Je dédie ce mémoire à mes chers parents, pour leur amour inépuisable, leur patience, leur sacrifices et leur confience en moi, meme dans les moments de doute. Votre soutien a été ma plus grande force.

À mon frère, mes soeurs, pour leur affection, leur humour, leurs encouragements et leur présence rassurante à chaque étape de ce parcours.

À toute ma famille et à mes amis, pour leurs mots gentils, leur écoute, leur aide, ou simplement leur présence au bon moment.

Et à toutes les personnes que je n'ai pas mentionnées mais qui, d'une manière ou d'une autre, ont laissé une trace positive dans cette aventure.

Je ne vous oublie pas.

BOUDRA Rania

❀ *Dédicaces* ❀

Je dédie ce modeste travail à tous les membres de ma famille, qui m'ont toujours soutenu avec amour, patience et sacrifices. Leur confiance en moi a été une source de force tout au long de ce parcours.

Je remercie également mes amis pour leur soutien constant, ainsi que toutes les personnes qui m'ont aidé de près ou de loin dans la réalisation de ce projet.

Enfin, Je dédie ce travail à moi-même, pour les efforts constants, la patience et la persévérance qui ont permis de mener à bien ce projet.

BENAMOURA Nadira

Table des matières

Liste des tableaux

Table des figures

Liste des abréviations

LLM	Large Language Model
NLP	Natural Language Processing
OCR	Optical Character Recognition
DRF	Django Rest Framework
UML	Unified Modeling Language
SQL	Structured Query Language
MTV	Model-Template-View
REST	Representational State Transfer
API	Application Programming Interface
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
XML	eXtensible Markup Language
DOM	Document Object Model
BDD	Base de Données
PostgreSQL	Postgre Structured Query Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
CSS	Cascading Style Sheets
OS	Operating System
XP	eXtreme Programming
UP	Unified Process
2UP	Two-Phase Unified Process
SSD	Solid State Drive
ORM	Object-Relational Mapping

Introduction générale

Aujourd'hui, avec les nouvelles technologies connues, l'homme cherche de plus en plus de faciliter le mode de sa vie dans n'importe quel domaine, cela grâce à l'informatique qui est une science étudiant les techniques du traitement automatique de l'information. Elle joue un rôle important dans le développement de l'entreprise et d'autres établissements. Dans notre étude, on s'intéresse aux cabinets d'avocats spécialisés en droit commercial. Ces derniers sont confrontés à une explosion documentaire qui rend cruciale l'adoption de technologies d'intelligence artificielle innovantes comme modèles de langage avancés (LLM : Large Language Model) pour traiter, analyser et extraire des informations pertinentes à partir de documents juridiques complexes pour une meilleure productivité.

Le métier d'avocat, auparavant centré sur l'expertise juridique et la défense des intérêts des clients, se trouve aujourd'hui confronté à la nécessité d'intégrer ces innovations technologiques. Si la fonction traditionnelle de conseil et de représentation demeure fondamentale, la manière dont les avocats accèdent à l'information juridique et l'exploitent évolue radicalement. ces innovations technologiques permettent non seulement d'organiser intelligemment les connaissances juridiques, mais aussi d'accélérer significativement les processus de recherche et d'analyse des documents. Cette approche répond directement aux enjeux contemporains du secteur juridique : optimisation du temps, précision accrue et indexation, elle permet également de faciliter l'accès aux documents juridiques, améliorer la productivité des équipes, garantir la sécurité des données sensibles .

La plateforme proposée s'inscrit dans cette dynamique d'innovation, en exploitant la puissance des modèles de langage pour transformer des bases documentaires hétérogènes en ressources structurées et facilement exploitables. Offrant ainsi aux cabinets d'avocats un outil stratégique pour maintenir l'excellence de leurs services dans un environnement numérique en constante évolution.

Ce mémoire est réparti en 4 chapitres :

Le premier chapitre est consacré à la présentation de l'organisme d'accueil et l'étude de l'existant, ainsi qu'à la méthodologie du développement suivi pour la réalisation du système. Ensuite, dans le second chapitre nous allons aborder la spécification et l'analyse des besoins du projet. Le troisième chapitre est consacré à la conception dans lequel les résultats précédents seront exploités pour détailler le fonctionnement du système. Enfin, le quatrième chapitre est dédié à la réalisation de l'application ou nous allons présenter l'environnement du travail qui inclut tous les outils et les langages de programmation utilisé, plus L'architecture globale suivi de quelques interfaces graphiques de notre application. A la fin nous clôturerons par une conclusion générale qui décrit les résultats obtenus et quelques perspectives.

Chapitre I

Présentation de l'organisme d'accueil et étude de l'existant

I.1 Inroduction

Dans ce chapitre, nous présentons l'organisme d'accueil, un cabinet d'avocats spécialisé en droit de commerce, ainsi que l'étude de l'existant concernant la gestion documentaire. Nous détaillons les problématiques actuelles liées à la recherche et à l'accessibilité des documents, et nous proposons une solution basée sur une plateforme avancée d'extraction de métadonnées et de recherche alimentée par des modèles de langage (LLM).

I.2 Présentation de l'organisme d'accueil

I.2.1 Contexte et objectif

Le cabinet d'avocats, spécialisé en droit de commerce, gère une volumineuse base documentaire composée principalement de fichiers PDF et Word (90 %), ainsi que d'autres formats tels qu'Excel et PowerPoint (10 %). Environ 50 % des documents sont en arabe, tandis que les autres sont en français et en anglais. Une partie importante des documents (60 à 70 %) est déjà OCRisée, mais le reste est sous forme d'images ou non exploitable directement par un moteur de recherche.

L'objectif principal du cabinet est de :

- Faciliter l'accès aux documents juridiques
- Améliorer la productivité des équipes
- Garantir la sécurité des données sensibles

I.2.2 Organisation d'accueil

Le cabinet **Benslimane and Partners** est une structure juridique privée fondée en 2007 par Maître Samir Benslimane, avocat algérien expérimenté. Le cabinet est entièrement détenu et dirigé par des professionnels du droit algérien, ce qui lui confère une parfaite maîtrise du contexte juridique national et une capacité d'adaptation aux réalités du terrain.

Implanté à Béjaïa, le cabinet s'est imposé comme un acteur reconnu dans le domaine du droit des affaires, tout en développant une expertise étendue dans d'autres branches du droit, tant au niveau national qu'international.

Sa mission principale est d'offrir des services juridiques de haut niveau, en alliant rigueur, discrétion et innovation. À ce titre, Benslimane and Partners intervient dans plusieurs domaines juridiques tels que le conseil juridique, le contentieux, l'arbitrage international, et l'accompagnement stratégique de ses clients. Son équipe expérimentée œuvre avec un souci constant de confidentialité, en mettant l'accent sur la qualité du conseil et la réactivité face aux besoins spécifiques de chaque client.

Le cabinet couvre un large éventail de secteurs d'activité : droit bancaire, droit des sociétés, télécommunications, assurances, immobilier, marchés publics, concurrence, aviation, marques commerciales et industrie pharmaceutique. Il propose trois principaux types de services juridiques :

- Il offre un service de **conseil juridique (Advisory)**, en accompagnant ses clients avec des solutions adaptées à leurs besoins, basées sur une bonne maîtrise du droit algérien.
- Il intervient dans le **contentieux (Litigation)**, en défendant ses clients dans des affaires civiles, commerciales, administratives ou pénales.

- Il dispose d'une expertise reconnue en **arbitrage international (Arbitrage)**, notamment devant la Cour Internationale d'Arbitrage et la London Court of International Arbitration.

L'objectif du cabinet est d'être un partenaire juridique fiable et proactif, capable de fournir des solutions innovantes, tout en participant activement au développement et à la promotion du droit en Algérie.

I.3 Étude de l'existant

I.3.1 Cadre d'étude

Le cadre d'étude se concentre sur l'analyse des limites du système actuel de gestion documentaire. Actuellement, les documents sont classés de manière anarchique dans le système d'information (SI), principalement par dossiers, sans structuration efficace, ce qui complique l'accès rapide aux informations. Une plateforme collaborative, basée sur Nextcloud[1], a été mise en place pour gérer les documents à l'aide de workflows et associer les ressources documentaires aux clients. Cependant, cette plateforme reste perfectible en raison de plusieurs lacunes : la recherche est peu performante, l'indexation des documents manque de précision, et l'accessibilité des ressources, notamment des livres électroniques, est limité en raison d'une organisation anarchique. De plus, le système actuel présente des failles de sécurité significatives, notamment en raison de l'exposition potentielle des données sur Nextcloud, qui est moins sécurisé que l'utilisation d'un modèle de langage local (LLM) intégré à notre application. Par ailleurs, le système précédent reposait uniquement sur un enregistrement des documents sous forme de fichiers nommés, sans base documentaire structurée, ce qui amplifie les difficultés actuelles d'organisation et de gestion des connaissances.

I.3.2 Problématique

La problématique principale réside dans la difficulté à retrouver rapidement les documents juridiques et commerciaux en raison d'une organisation limitée et d'un manque d'indexation intelligente. De plus, une partie des documents n'est pas exploitable directement par un moteur de recherche, ce qui réduit l'efficacité du système actuel. Les avocats et les assistants juridiques passent un temps considérable à rechercher des documents spécifiques, ce qui affecte leur productivité. Par ailleurs, le système actuel, basé sur Nextcloud, présente des failles de sécurité significatives dues à l'exposition potentielle des données, rendant la protection des informations sensibles insuffisante.

I.3.3 Solution proposée

Pour résoudre ces problèmes, nous proposons la mise en place d'une plateforme avancée d'extraction de métadonnées et de recherche alimentée par des modèles de langage (LLM). Cette solution complète intégrera les fonctionnalités suivantes :

- Un OCR (Optical Character Recognition) avancé pour traiter les documents non OCRisés
- Une indexation intelligente des ressources documentaires
- Un moteur de recherche multilingue (arabe, français, anglais)
- Un système Dynamic LLM Selection pour adapter automatiquement le modèle de langage

- Une gestion fine des droits d'accès aux documents
- Une interface collaborative pour le travail d'équipe
- Une sécurité renforcée grâce à l'utilisation d'un LLM local, réduisant les risques d'exposition des données par rapport à des solutions cloud comme Nextcloud

I.3.4 Méthodologie de développement

I.3.4.1 Méthodes de développement logiciel

Voici un tableau comparatif des principales méthodologies de développement logiciel : UP, 2UP, XP et Scrum.[2][3][4]

Critère	**UP (Unified Process)**	**2UP (Two-Phase UP)**	**XP (Extreme Programming)**	**Scrum**
Type	Processus itératif et incrémental	Variante simplifiée de UP en 2 phases	Méthode agile de développement logiciel axée sur la qualité	Cadre agile de gestion de projet basé sur des sprints
Phases principales	Inception, Elaboration, Construction, Transition	Inception/ Elaboration et Construction/ Transition	Cycles courts d'itérations (1-2 semaines)	Sprints courts (1-4 semaines)
Type de projet	Grands projets à long terme, complexité moyenne à élevée	Projets moyens à grands, recherche de rapidité et simplicité	Projets complexes, nécessitant haute qualité technique et collaboration étroite	Projets complexes, nécessitant adaptabilité et livraison rapide
Rôles	Plusieurs rôles définis (architecte, ingénieur composant, designer,etc)	Similaire à UP	6 rôles (dont Coach)	3 rôles principaux (Product Owner, Scrum Master, équipe)
Taille d'équipe	Moyenne à grande	Moyenne	Petite à moyenne	Petite à moyenne

TABLE I.1 – Tableau comparatif entre les méthodes de développement logiciel

I.3.4.2 Description de la méthodologie choisie

Dans le cadre de ce projet, nous avons choisi d'adopter la méthodologie Scrum, une approche agile qui repose sur des cycles courts et répétitifs appelés sprints.[5]

FIGURE I.1 – Cycle de développement Scrum

Chaque sprint est considéré comme un mini-projet, avec ses propres objectifs, activités et livrables. L'ensemble du processus Scrum repose sur les étapes suivantes :

- **Préparation :** Avant de commencer un sprint, on définit les objectifs du projet, les grandes étapes à suivre et les tâches à réaliser.
- **Planification du sprint :** L'équipe choisit les tâches les plus importantes à faire pendant le sprint, fixe un objectif clair et estime le temps nécessaire.
- **Réunion quotidienne (Daily Scrum) :** Chaque jour, une courte réunion permet à chacun de dire ce qu'il a fait, ce qu'il va faire, et s'il rencontre des problèmes.
- **Implémentation :** L'équipe travaille sur les tâches prévues, en suivant les priorités définies au début du sprint.
- **Revue de sprint :** À la fin du sprint, le travail réalisé est présenté. Les retours permettent d'ajuster si besoin.
- **Rétrospective :** L'équipe fait le point sur ce qui a bien ou moins bien fonctionné, pour s'améliorer dans les prochains sprints.

Le tableau suivant présente la répartition des tâches effectuées au fil des sprints :

Sprint n°	**Tâches effectuées**	**Durée**
1	Initialisation du backend avec Django REST Framework ; création du frontend avec Vue 3, Vuetify et Pinia ; configuration de la base de données PostgreSQL ; connexion backend-BDD	4 semaine
2	Intégration d'un outil OCR basé sur Mistral ; développement de la logique d'extraction du texte	2 semaine
3	Intégration du modèle LLM Mistral ; développement de l'API interne ; définition des prompts ; stockage des métadonnées dans le modèle Metadata	4 semaine
4	Mise en œuvre de la recherche full-text PostgreSQL ; création des filtres (type de fichier, catégorie, date) ; intégration de la barre de recherche ; affichage dynamique des résultats	2 semaine
5	Mise en place de l'authentification avec JWT (JSON Web Token) ; création des vues Login côté frontend	1 semaine

TABLE I.2 – Répartition des tâches par sprint selon la méthode Scrum

I.3.4.3 Langage UML

Le langage de modélisation unifié (UML) est un outil visuel utilisé pour représenter à la fois le comportement dynamique et la structure statique d'un système. Dans le développement logiciel, il sert à définir, construire, visualiser et documenter les différents éléments d'une application. Son principal objectif est de faciliter la compréhension et la communication autour des fonctionnalités et décisions à prendre. UML repose sur des symboles et des diagrammes pour illustrer les différentes parties d'un système et leurs interactions, offrant ainsi un cadre standardisé et adaptable pour organiser l'information technique. Les caractéristiques principales des diagrammes UML sont les suivantes :

- **Standard** : UML est reconnu comme une norme largement adoptée, permettant aux développeurs travaillant sur différentes plateformes de partager une compréhension commune des systèmes.
- **Visuel** : Le langage repose sur des représentations graphiques simples pour modéliser la structure et le comportement des systèmes, rendant ainsi les architectures complexes plus accessibles.
- **Polyvalent** : UML est capable de modéliser divers types de systèmes, au-delà des applications logicielles, notamment pour des processus métier ou d'autres types d'organisations.[6]

I.3.4.4 Types de diagrammes UML

Les diagrammes UML sont divisés en deux catégories principales :

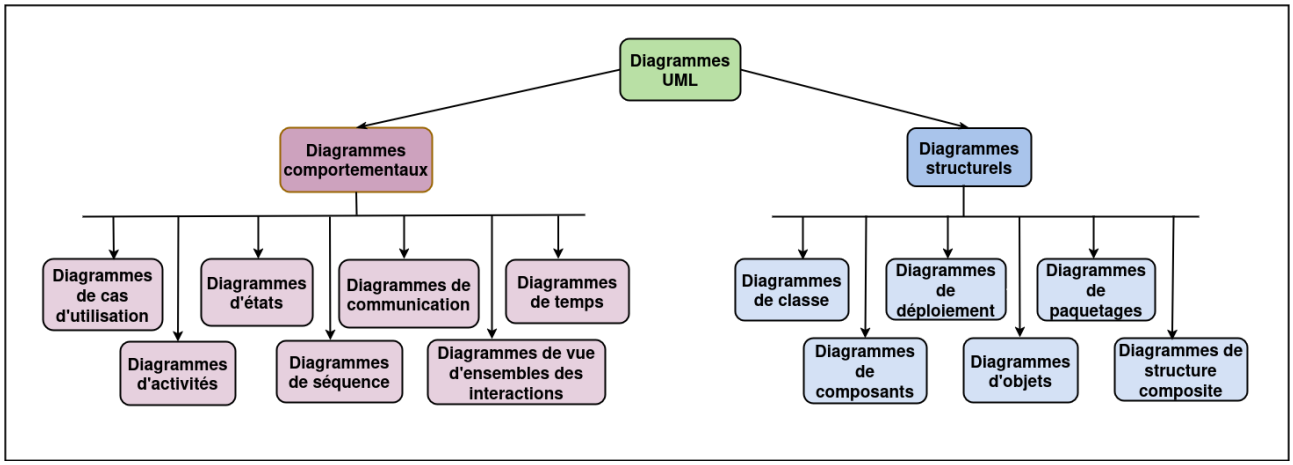


FIGURE I.2 – Types de diagrammes UML

Diagrammes structurels (aspects statiques)

- **Diagramme de classes** : Représente les classes, leurs attributs, méthodes et les relations (héritage, association).
- **Diagramme de composants** : Montre les relations entre les composants logiciels et leurs dépendances.
- **Diagramme de déploiement** : Illustre l'installation physique du système sur des serveurs ou bases de données.
- **Diagramme d'objets** : Affiche un instantané des objets et de leurs liens à un moment donné.
- **Diagramme de paquetages** : Organise les parties du système en paquetages avec leurs dépendances.
- **Diagramme de structure composite** : Détaille la structure interne d'un composant complexe.

Diagrammes comportementaux (aspects dynamiques)

- **Diagramme de cas d'utilisation** : Décrit les interactions entre les acteurs (utilisateurs) et le système.
- **Diagramme d'activités** : Montre le flux d'étapes (séquentielles, parallèles ou conditionnelles).
- **Diagramme d'états** : Visualise les états d'un objet et les transitions déclenchées par des événements.
- **Diagramme de séquence** : Met en évidence l'échange de messages entre objets dans un scénario précis.
- **Diagramme de communication** : Similaire au diagramme de séquence, mais avec une vue plus générale des interactions.
- **Diagramme de vue d'ensemble des interactions** : Offre une vue globale des interactions via des cadres.
- **Diagramme de temps** : Se concentre sur les contraintes temporelles des échanges de messages.

I.4 Conclusion

Ce chapitre a présenté l'organisme d'accueil, un cabinet d'avocats spécialisé en droit de commerce, et les problématiques actuelles liées à la gestion documentaire. Nous avons également proposé une solution basée sur une plateforme avancée d'extraction de métadonnées et de recherche alimentée par des modèles de langage (LLM). La méthodologie de développement choisie, ainsi que l'utilisation du langage UML, seront essentielles pour la réussite du projet, en particulier pour faciliter la recherche et l'accès aux documents juridiques et commerciaux.

Chapitre II

Spécification et Analyse des besoins

II.1 Introduction

Ce chapitre a pour objectif de décrire les fonctionnalités de notre projet selon les exigences fonctionnelles et non fonctionnelles afin d'entamer l'analyse des besoins en identifiant des acteurs et leurs interactions avec le système grâce au diagramme de contexte système, ainsi qu'une modélisation UML globale accompagnée des cas d'utilisation avec leur description détaillée.

II.2 Spécification des besoins

Cet axe offre une vue globale du projet, en spécifiant les différentes fonctionnalités à implémenter. Toutefois, il est essentiel de comprendre et de savoir d'abord que veut dire métadonnées et LLM.

Métadonnées :

Le mot « méta », d'origine grecque, signifie « au-delà ». Dans le contexte numérique, il fait référence aux métadonnées, qui sont des données servant à décrire d'autres données.

Par exemple, la date de création d'un document, le nom de l'auteur, ou encore le type de fichier sont des métadonnées. Elles ne font pas partie du contenu principal, mais apportent des informations précieuses sur ce contenu, facilitant ainsi sa gestion, son organisation et sa recherche.

Les métadonnées sont omniprésentes dans le monde numérique :

- elles se trouvent dans les fichiers textes, images, vidéos, emails, ou encore pages web.
- elles peuvent être créées automatiquement (comme l'horodatage d'un fichier), ou ajoutées manuellement (comme des mots-clés ou une description).
- elles décrivent le contenu sans l'ouvrir.
- elles retrouvent facilement l'information grâce à des mots-clés ou des dates.
- elles organisent les documents dans des systèmes de gestion.
- elles conne À connectent des données entre elles (dans le web sémantique ou la recherche intelligente).

Par exemple : sur une photo, les métadonnées peuvent indiquer l'emplacement, la date, et même le modèle de l'appareil photo utilisé.[7]

LLM :

Un LLM (Large Language Model) est un système d'intelligence artificielle capable de comprendre, générer, reformuler et analyser du langage humain, grâce à l'apprentissage automatique (Machine Learning) et au Deep Learning via des modèles transformateurs. Il est utilisé dans de nombreuses tâches de Traitement Automatique du Langage Naturel (NLP).

En pratique, un LLM est entraîné sur d'immenses quantités de texte, ce qui lui permet de détecter des structures linguistiques, d'anticiper des suites de mots, de répondre à des questions, de résumer des documents, de traduire des textes, ou même de générer des descriptions selon le contexte donné.

Mistral, en tant que LLM open source performante, repose sur cette même architecture, optimisée pour être léger, rapide et facilement déployable. Spécialisé dans la compréhension du langage naturel, il s'inscrit pleinement dans le champ du NLP moderne.[8]

II.2.1 Besoins fonctionnels

Le système doit offrir plusieurs fonctionnalités essentielles :

- **Authentification et gestion des utilisateurs**
  - ⋆ Un utilisateur peut se connecter avec un nom d'utilisateur ou email et un mot de passe.
  - ⋆ L'accès aux fonctionnalités dépend du rôle de l'utilisateur.
- **Recherche avancée de documents**
  - ⋆ Un utilisateur peut rechercher des documents à partir de trois lettres selon différents critères :
    - Nom
    - Contenu
  - ⋆ Les résultats peuvent selon le choix de tri par les métadonnées être filtrables par le système.
- **gestion des documents**
  - ⋆ Les documents peuvent être créés par le système lui-même d'une façon automatique.
  - ⋆ Les extensions supportées incluent (PDF, DOCX, PPT, TXT, Excel, Images...) avec les trois langues français, arabe, anglais.
  - ⋆ Un traitement OCR (Reconnaissance optique de caractères) peut être appliqué aux images, pdf scanné pour extraire du texte.
- **Extraction automatique des métadonnées**
  Grâce à un LLM il est possible d'accomplir des tâches NLP (Natural Language Processing) telles que l'extraction des informations clés à partir d'un texte comme :
  - ⋆ L'auteur du document
  - ⋆ Les mots-clés
  - ⋆ Le titre
  - ⋆ La description
  - ⋆ Type de document
- **Visualisation et téléchargement des documents**
  - ⋆ Cette fonctionnalité permet aux utilisateurs de consulter un aperçu du document traité avant de le télécharger, pour une meilleure expérience et une vérification de contenu.
  - ⋆ Le téléchargement sera possible uniquement pour les documents déjà traités et stockés dans la base de données. Avant cela, le système vérifie si le chemin du fichier est valide (au sens de son existence dans le système os) : si oui, une icône de téléchargement s'affiche sinon, elle reste masquée.
- **Indexation et mise à jour des documents**
  Chaque document déjà traité est indexé dans la base de données pour faciliter la recherche par métadonnées et si un est mis à jour, ses métadonnées seront mises à jour aussi, pour assurer ceci on a opté le hachage du contenu d'un fichier comme suit :
  Pour la création, le système valide le path et vérifie si le fichier a déjà été traité grâce à son empreinte (hash) qui identifier chaque document d'une façon unique. Si le document est nouveau ou a été modifié, il va passer par tout le processus d'extraction et vas être indexé dans la BDD et les métadonnées seront créés ou mises à jour. Sinon, si c'est le même contenu malgré le path, le nom est différent ou même si c'est le document lui-même les données déjà extraites seront directement réutilisées, évitant ainsi les retraitements inutiles et la redondance.

II.2.2 Besoins non fonctionnels

Les besoins non fonctionnels sont des exigences qui influencent l'expérience utilisateur sans être liées directement aux fonctionnalités principales. Ils représentent les contraintes techniques et les caractéristiques qualitatives que le système doit respecter.

- **Performance** : La recherche et l'extraction des métadonnées doivent être rapides.
- **Sécurité** : l'accès à l'application exige une authentification de l'utilisateur.
- **Scalabilité** : Possibilité d'ajouter un grand nombre de documents sans ralentissement.
- **Fiabilité** : Les données stockées doivent être précises et cohérentes.
- **L'ergonomie et l'adaptabilité des interfaces aux différents terminaux** : la plateforme doit offrir une interface responsive , intuitive simple à utiliser.
- **Mode d'emploi** : l'application doit être facile a utiliser et offrir une bonne qualité d'expérience utilisateur.
- **L'extensibilité de l'application de pouvoir introduire d'autres fonctionnalités.**

II.3 Analyse des besoins

L'analyse des besoins permet à partir des besoins spécifiés d'identifier et de décrire les cas d'utilisation du futur système afin de répondre aux attentes du client.

II.3.1 Identification des acteurs

Un acteur est une entité externe qui interagit avec le système (un utilisateur, un matériel externe ou logiciel, même un autre système), on peut donc dire qu'un acteur représente l'utilisateur de notre système, dans ce qui suit, nous présentons les acteurs principaux de notre système.[9][10][11]

Acteur	Description «rôle»
Administrateur	Gère les utilisateurs, les catégories, Rechercher, Envoyer le path et Consulter, Télécharger les documents, Choisir le tri par métadonnées (type de document, date de création, catégorie)
Avocat, Assistant	Rechercher et Consulter, Télécharger les documents, Choisir le tri par métadonnées (type de document, date de création, catégorie)

TABLE II.1 – Identification des acteurs

II.3.2 Diagramme de contexte système

C'est une représentation simplifiée qui décrit le fonctionnement du future système en mode boîte noire sans entrer dans les détails internes, il représente l'interaction des différents acteurs identifiés précédemment avec notre système comme est illustré dans le tableau suivant :

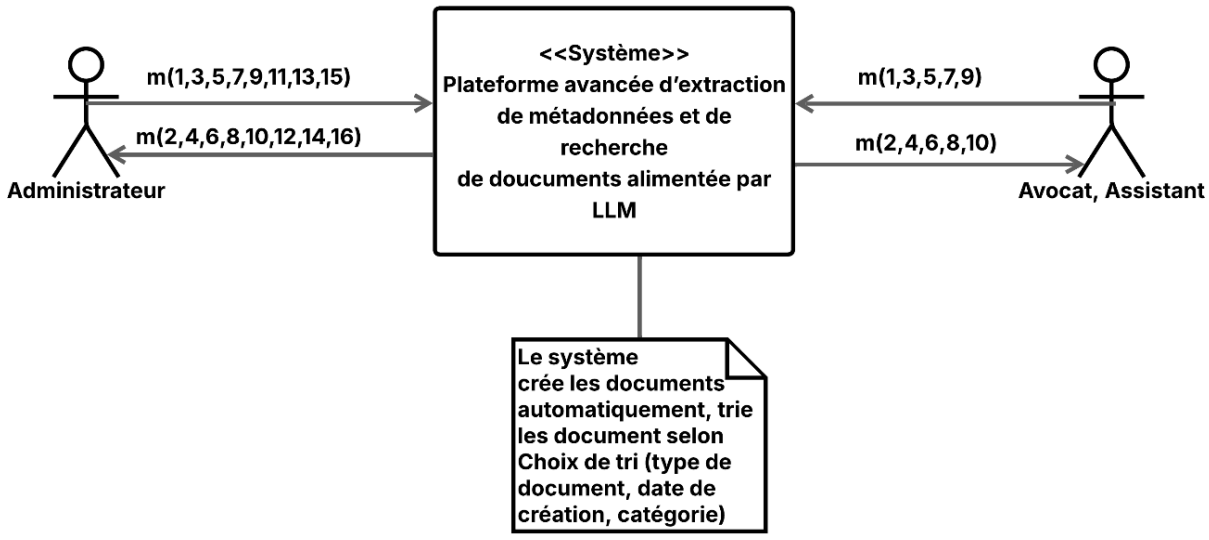


FIGURE II.1 – Diagramme de contexte système

Acteur	**MessageActeur → Système**	**MessageSystème → Acteur**
Administrateur	m1 : Demande d'authentification	m2 : Message d'erreur ou affichage d'espace administrateur, ou interface de recherche
	m3 : Gestion des comptes utilisateur	m4 : Message de succès ou erreur
	m5 : Gestion des catégories	m6 : Message de succès ou erreur
	m7 : Télécharger un document	m8 : Message de succès ou erreur
	m9 : Envoyer le path	m10 : Message d'erreur ou succès
	m11 : Rechercher un document	m12 : m12 : Message d'erreur ou affichage de document
	m13 : Consulter un document	m14 : Message d'erreur ou affichage des information du document
	m15 : Choisir le tri par métadonnées (type de document, date de création, catégorie)	m16 : Message d'erreur ou affichage des documents filtrés
Avocat, Assistant	m1 : Demande d'authentification	m2 : Message d'erreur ou accès direct à l'interface de recherche
	m3 : Consultation d'un document	m4 : Affichage des informations de document ou message d'erreur
	m5 : Choix de tri (type de document, date de création, catégorie)	m6 : Message d'erreur ou affichage des documents filtrés
	m7 : Rechercher un document	m8 : affichages de résultat de la recherche
	m9 : Télécharger un document	m10 : Message de succès ou erreur

TABLE II.2 – Les messages échangés entre les acteurs et le système

II.3.3 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est une modélisation UML qui permet de visualiser l'ensemble du comportement fonctionnel d'un système logiciel pour un acteur particulier, consiste à bien organiser les besoins du système selon les fonctionnalités principales. [9][10][11]

II.3.3.1 Identification des cas d'utilisation

Le tableau suivant récapitule les différents cas d'utilisation associés à notre système :

N°	**Cas d'utilisation**	**Acteur(s)**
1	S'authentifier	Administrateur, Avocat, Assistant
2	Gérer les comptes des utilisateurs	Administrateur
3	Consulter un document	Administrateur, Avocat, Assistant
4	Envoyer le path	Administrateur
5	Rechercher un document	Administrateur, Avocat, Assistant
6	Gérer les catégories	Administrateur
7	Choisir le tri par métadonnées (type de document, date de création, catégorie)	Administrateur, Avocat, Assistant
8	Télécharger un document	Administrateur, Avocat, Assistant

TABLE II.3 – Identification des cas d'utilisation

II.3.3.2 Description des cas d'utilisation les plus important

Dans cette phase nous allons présenter une description textuelle des cas d'utilisation les plus important.

- Description du cas d'utilisation «S'authentifier» :

Sommaire	
Titre	S'authentifier
Résumé	Permet aux utilisateurs d'accéder aux espaces correspondants.
Description des scenarios	
Acteurs	Administrateur, Avocat, Assistant
Préconditions	L'application doit être accessible. L'utilisateur doit disposer d'un compte valide.
scénario	1. L'utilisateur saisit son nom d'utilisateur ou email, mot de passe. 2. Le système vérifie la saisie. 3. Le système vérifie les informations d'identification. 4. Si correct, accès à l'interface de correspondante. 5. Sinon renvoi la page d'authentification.
Enchaînement d'erreurs	- Erreur de saisie (champs vides ou incomplet). - Données erronées (mot de passe incorrect ou utilisateur non inscrit).
Postconditions	L'utilisateur est authentifié et peut accéder à l'interface correspondante.

Table II.4 – Description textuelle du cas d'utilisation «S'authentifier»

- Description du cas d'utilisation «Gérer les comptes» :

Sommaire	
Titre	Gérer les comptes
Résumé	Permet de gérer les comptes utilisateurs.
Description des scenarios	
Acteurs	Administrateur
Préconditions	L'administrateur doit être authentifié.
scénario	1. Le système affiche l'espace correspondant à l'administrateur. 2. Il ajoute/modifie/supprime des comptes. 3. le système affiche le formulaire correspondant. 4. Soit remplis formulaire d'ajout ou modifier formulaire de modification, même peut effectuer la suppression. 5. Le système enregistre les modifications. 6. Sinon le système renvoie le formulaire a nouveau.
Enchaînement d'erreurs	- Erreur de saisie (champs vides, incomplet). - Données erronées (données invalides).
Postconditions	Modification enregistrée et la base de données est mise à jour.

TABLE II.5 – Description textuelle du cas d'utilisation «Gérer les comptes»

- Description du cas d'utilisation «Rechercher un document» :

Sommaire	
Titre	Rechercher un document
Résumé	Permet aux utilisateurs d'effectuer une recherche d'un document.
Description des scenarios	
Acteurs	Avocat, Assistant, Administrateur
Préconditions	L'acteur est authentifié.
scénario	1. Si le document est déjà traité on passe direct à la recherche (2), sinon l'administrateur vas envoyer le path du document ou dossier qui contient ce document pour le rendre traité via le processus d'extraction. 2. Ensuite un acteur saisit nom ou mot clé parmi son contenu dans la barre de recherche, puis valide sa recherche. 3. Le système effectue la recherche et affiche les résultats. 4. L'acteur peut aussi Choisir le tri par métadonnées (type de document, date de création, catégorie).
Enchaînement d'erreurs	Le système affiche un message «aucun document trouvé» si n'existe pas.
Postconditions	Le système affiche les résultats de la recherche et l'utilisateur les visualise et même peut télécharger le document.

TABLE II.6 – Description textuelle du cas d'utilisation «Rechercher un document»

- Description du cas d'utilisation «Envoyer le path» :

Sommaire	
Titre	Envoyer le path
Résumé	Permet à l'administrateur d'envoyer un path d'un fichier ou dossier à l'api backend.
Description des scenarios	
Acteurs	Administrateur
Préconditions	L'acteur est authentifié.
scénario	1. Avant de commencer la recherche, d'abord l'administrateur doit envoyer le path de tous les documents existant à l'api backend. 2. Le système effectue le processus d'extraction de métadonnées et les rend traités et stocké dans la BDD.
Enchaînement d'erreurs	Le système affiche un message d'erreur «path invalide » si n'existe pas.
Postconditions	Stockage dans la base de données avec succès et affichage de message de succès avec les statistiques des résultats.

TABLE II.7 – Description textuelle du cas d'utilisation «Envoyer le path»

II.3.3.3 Diagrammes de cas d'utilisation

Maintenant nous allons modéliser les Diagrammes de cas d'utilisation par acteurs :

- Diagramme de cas d'utilisation «Administrateur» :

Le diagramme de la figure suivante représente les tâches spécifiées à un administrateur :

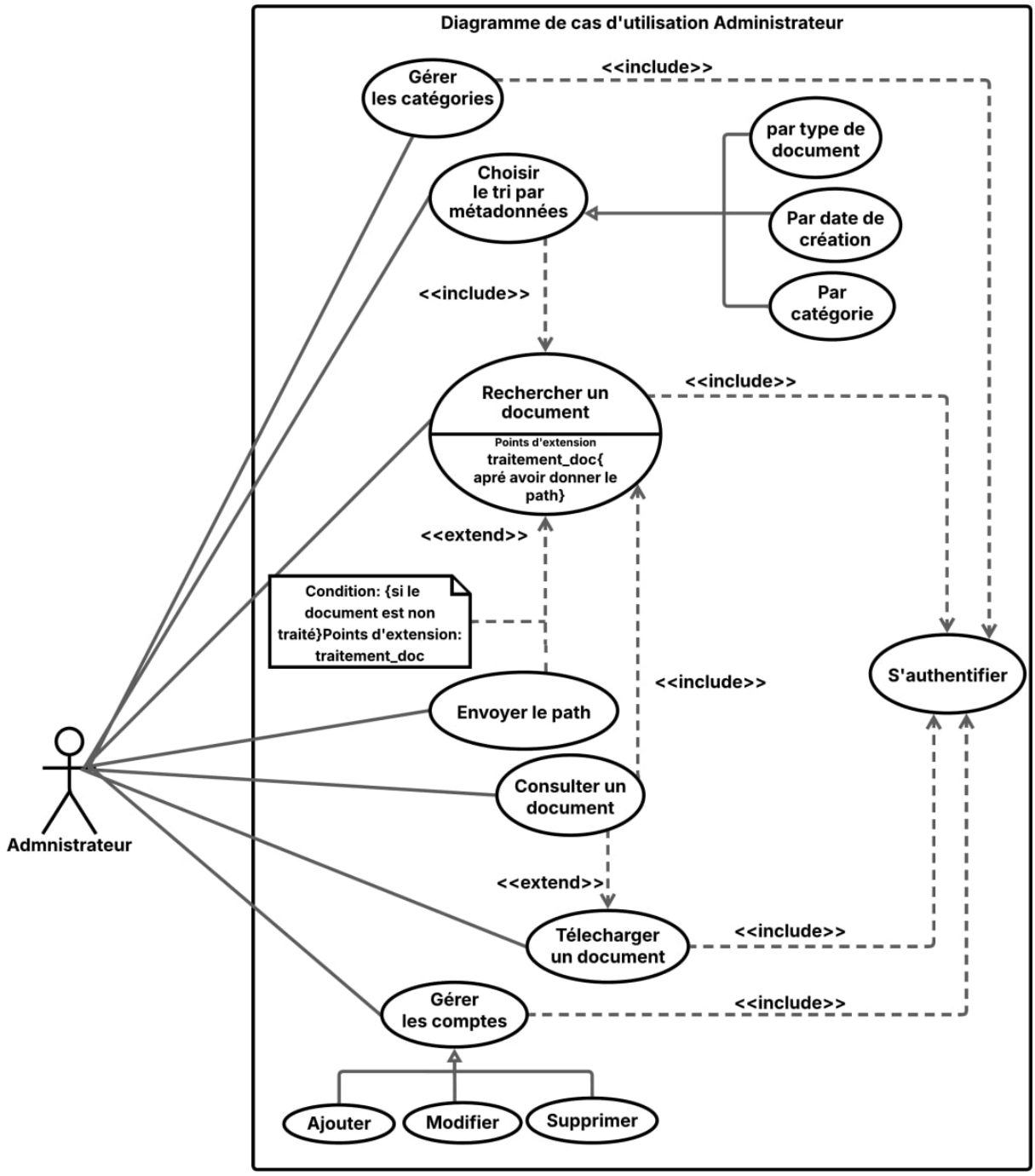


FIGURE II.2 – Diagramme de cas d'utilisation «Administrateur»

- Diagramme de cas d'utilisation «Avocat, Assistant» :

Le diagramme suivant représente les tâches spécifiées à un avocat et un assistant :

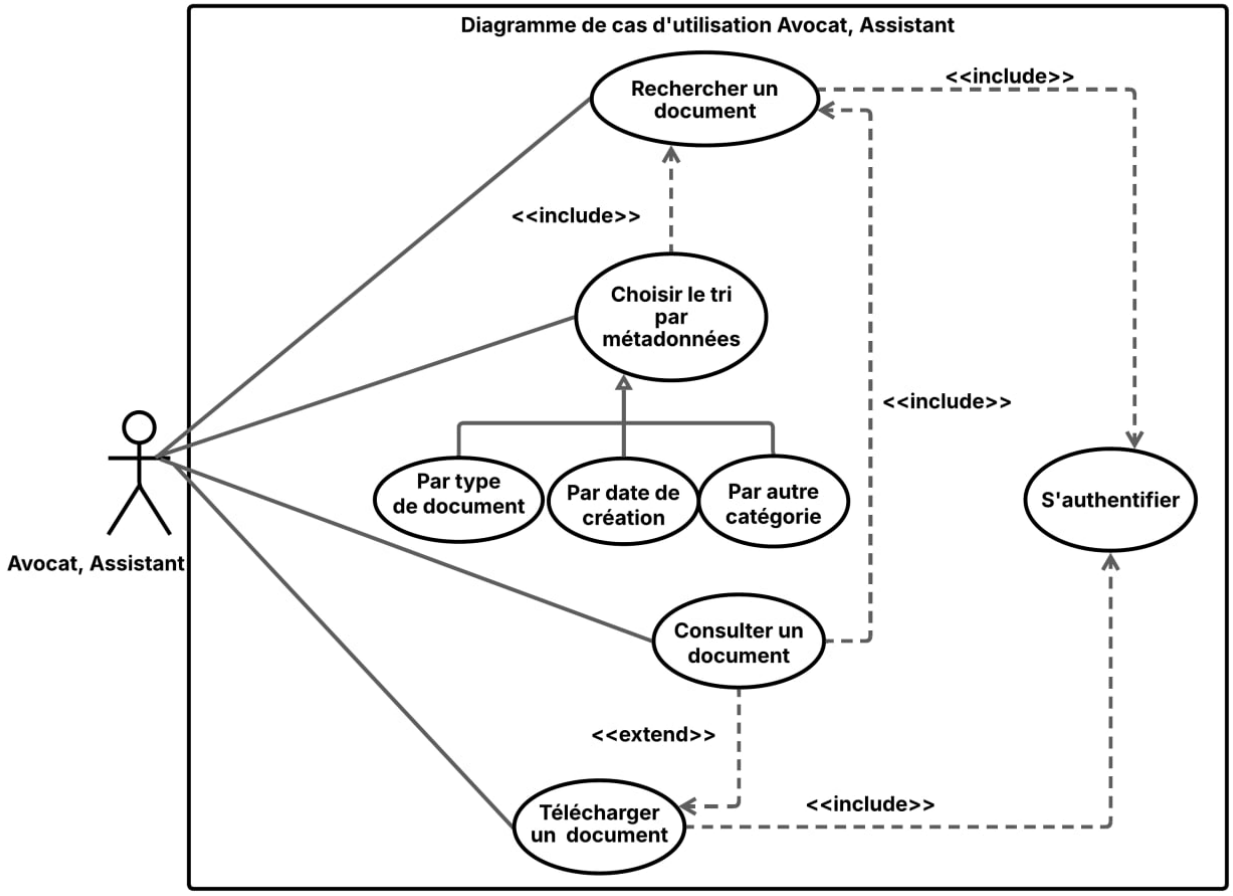


FIGURE II.3 – Diagramme de cas d'utilisation «Avocat, Assistant»

II.3.3.4 Diagramme de cas d'utilisation global

Le diagramme suivant représente les différents cas d'utilisation effectués par les différents acteurs :

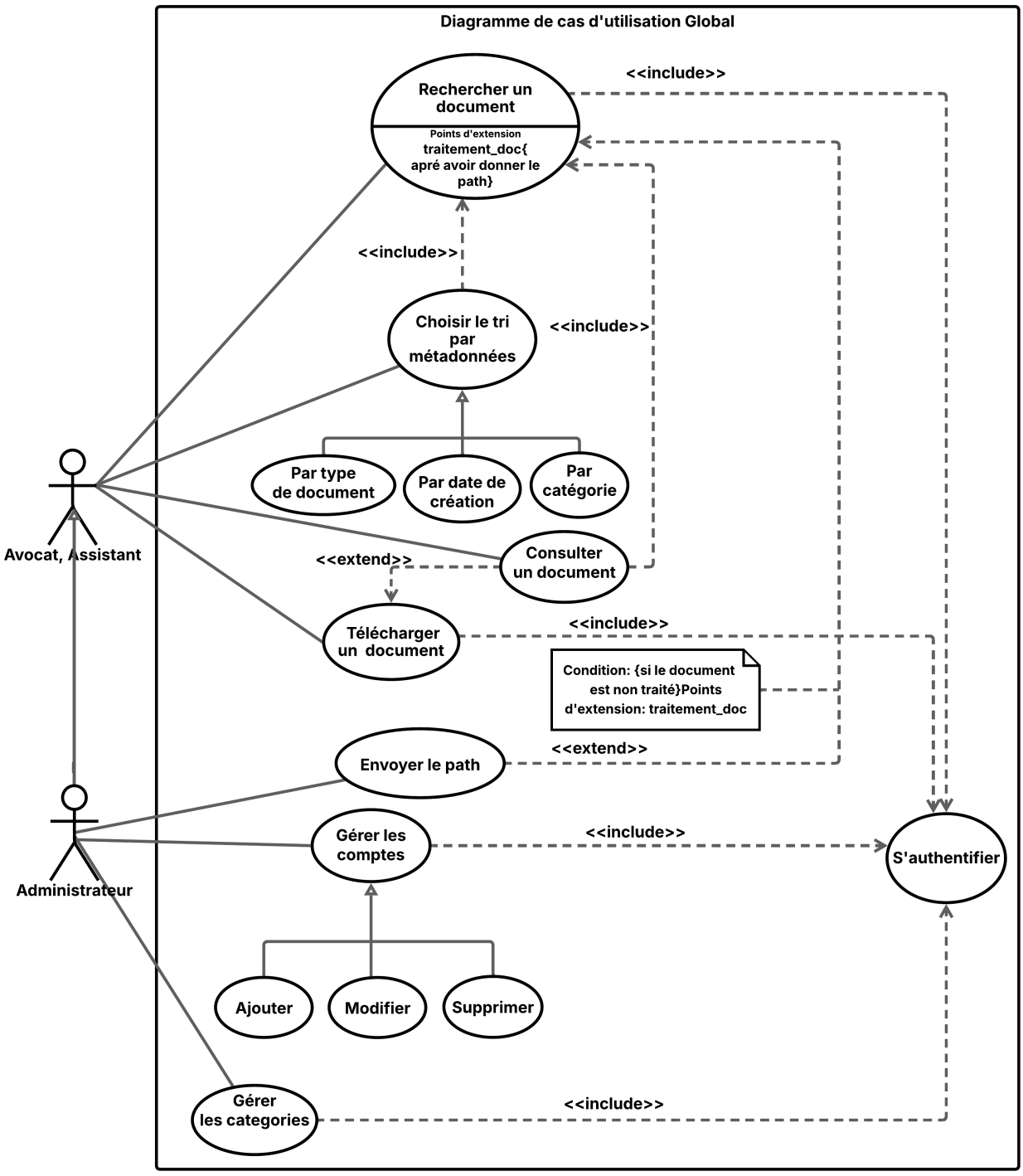


FIGURE II.4 – Diagramme de cas d'utilisation global

II.4 Conclusion

A l'issu de cette étape, nous avons pu décrire les besoins de notre système à travers les diagrammes UML, où nous avons présenté le diagramme de contexte et les diagrammes de cas d'utilisation suivi d'une description textuelle de chaque cas. Dans le prochain chapitre, nous entamons la phase de conception afin de décrire d'une manière détaillée ces besoins.

Chapitre III

Conception

III.1 Introduction

Ce chapitre a pour objectif de présenter la phase de conception, qui permet de structurer et d'organiser le projet. Nous commençons par la création des diagrammes de séquences correspondant aux cas d'utilisation déjà décrits. Ensuite, nous procédons à l'élaboration du diagramme de classes associé à notre application conçue par les différentes classes, leurs attributs et leurs relations. Enfin, nous verrons le modèle relationnel qui définit la structure de la base de données, les relations entre les tables, suivi d'un dictionnaire des données.

III.2 Diagrammes de séquence

Le diagramme de séquence est un diagramme UML, qui fait partie des diagrammes comportementaux (dynamique) et plus précisément des diagrammes d'interactions. Il permet de représenter l'interaction et l'enchaînement temporel entre les différents objets et acteurs du système lors l'exécution d'un scénario.[11][12]

Le diagramme de séquence standard se compose des éléments suivants :[13]

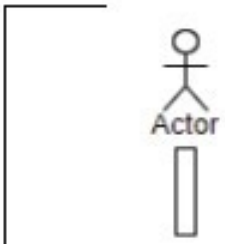
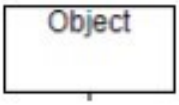
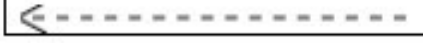
	Acteur	Les acteurs peuvent communiquer avec des objets Un acteur est modélisé en utilisant le symbole habituel : Stick man.
	Objet	Les objets sont des entités appartenant au système (instance d'une classe) ou se trouvant à ses limites (acteurs)
	Ligne de vie	Elle est représentée par une ligne verticale en dessous des Objets, elle représente la période de temps durant laquelle l'objet « existe »
	Message récursif	L'envoi de messages récursifs se représente par un dédoublement de la bande d'activation
	Message	Les objets communiquent en échangeant des messages représentés sous forme de flèches, ils sont étiquetés par le nom de l'opération ou du signal invoqué.
	Message de retour	Représenté par une flèche discontinue, c'est la réponse au message envoyé.

Figure III.1 – Différents composants d'un diagramme de séquence

Diagramme de séquence pour chaque cas d'utilisation :

— Diagramme de séquence du cas d'utilisation «S'authentifier» :

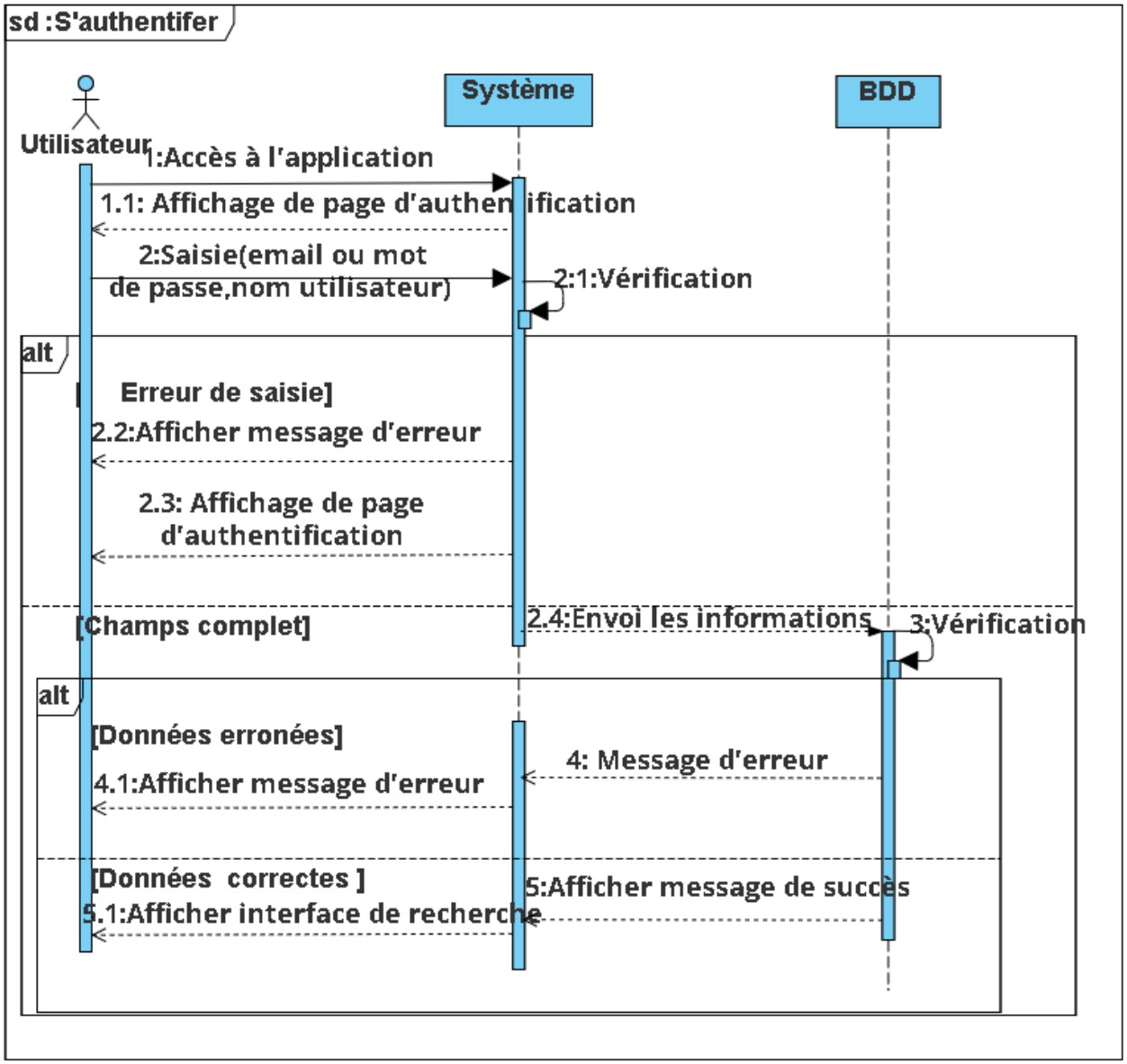


FIGURE III.2 – Diagramme de séquence du cas d'utilisation «S'authentifier»

— Diagramme de séquence du cas d'utilisation «gérer les comptes» :

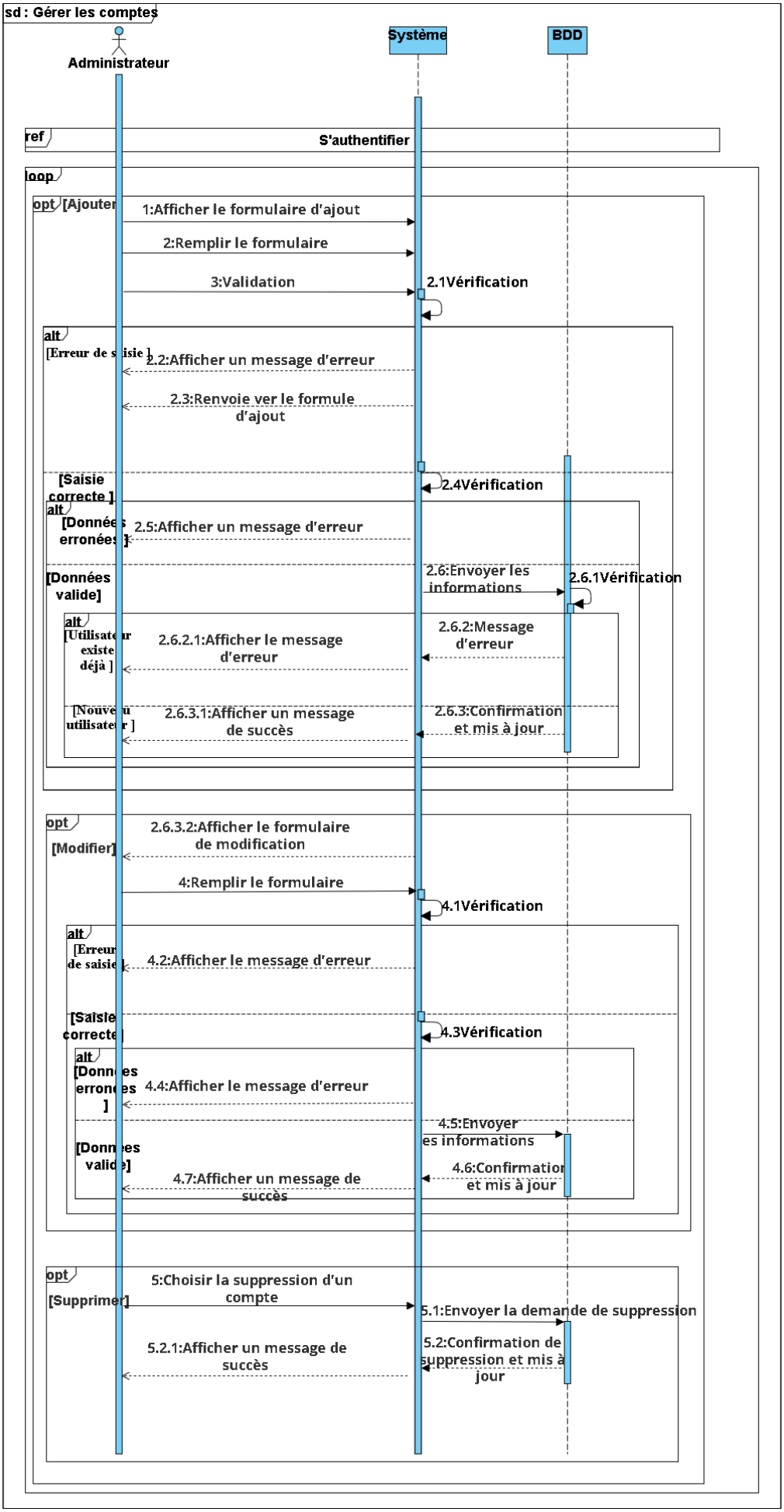


FIGURE III.3 – Diagramme de séquence du cas d'utilisation «Gérer les comptes»

— Diagramme de séquence du cas d'utilisation «Rechercher un document» :

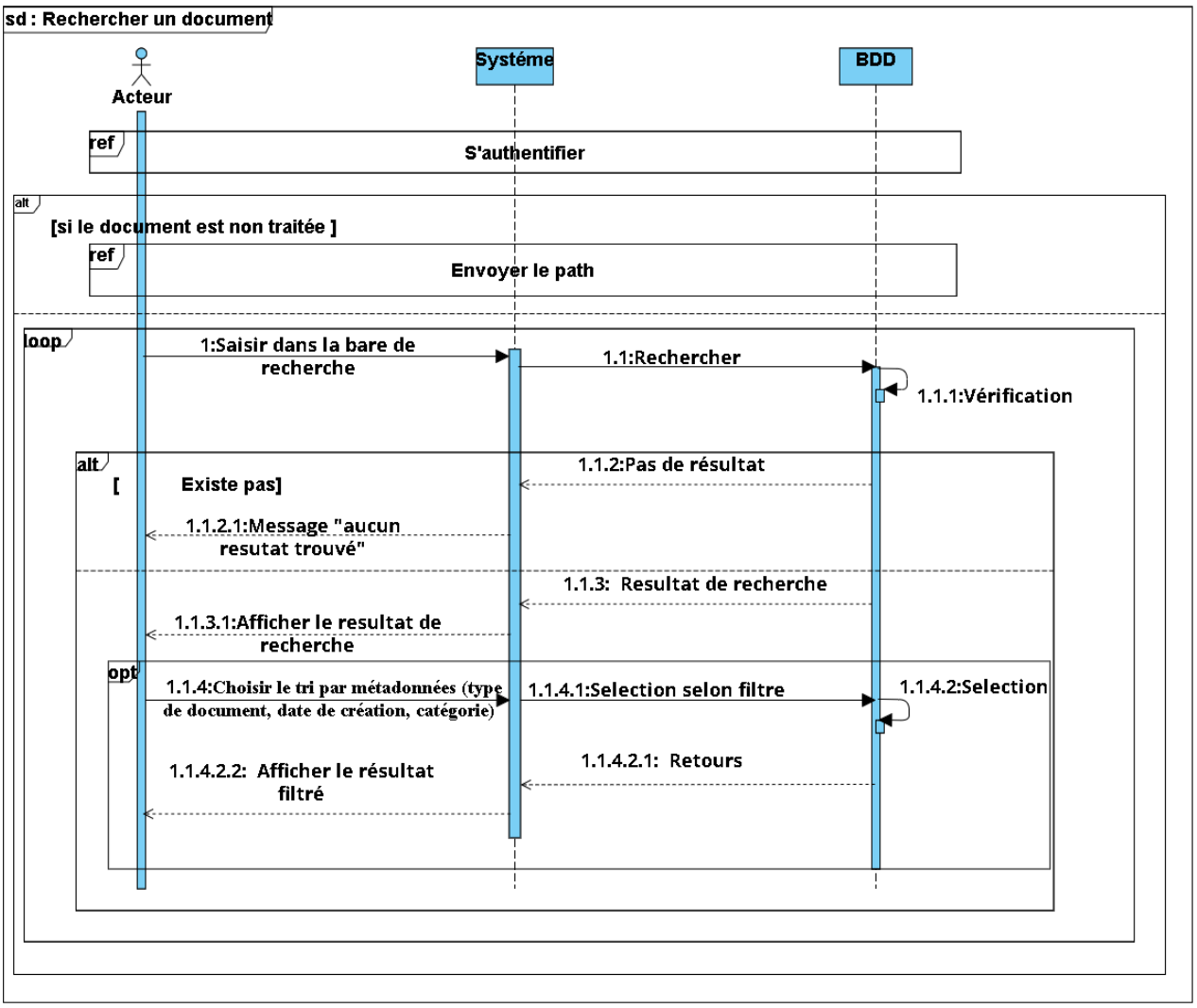


FIGURE III.4 – Diagramme de séquence du cas d'utilisation «Rechercher un document»

— Diagramme de séquence du cas d'utilisation «Envoyer le path» :

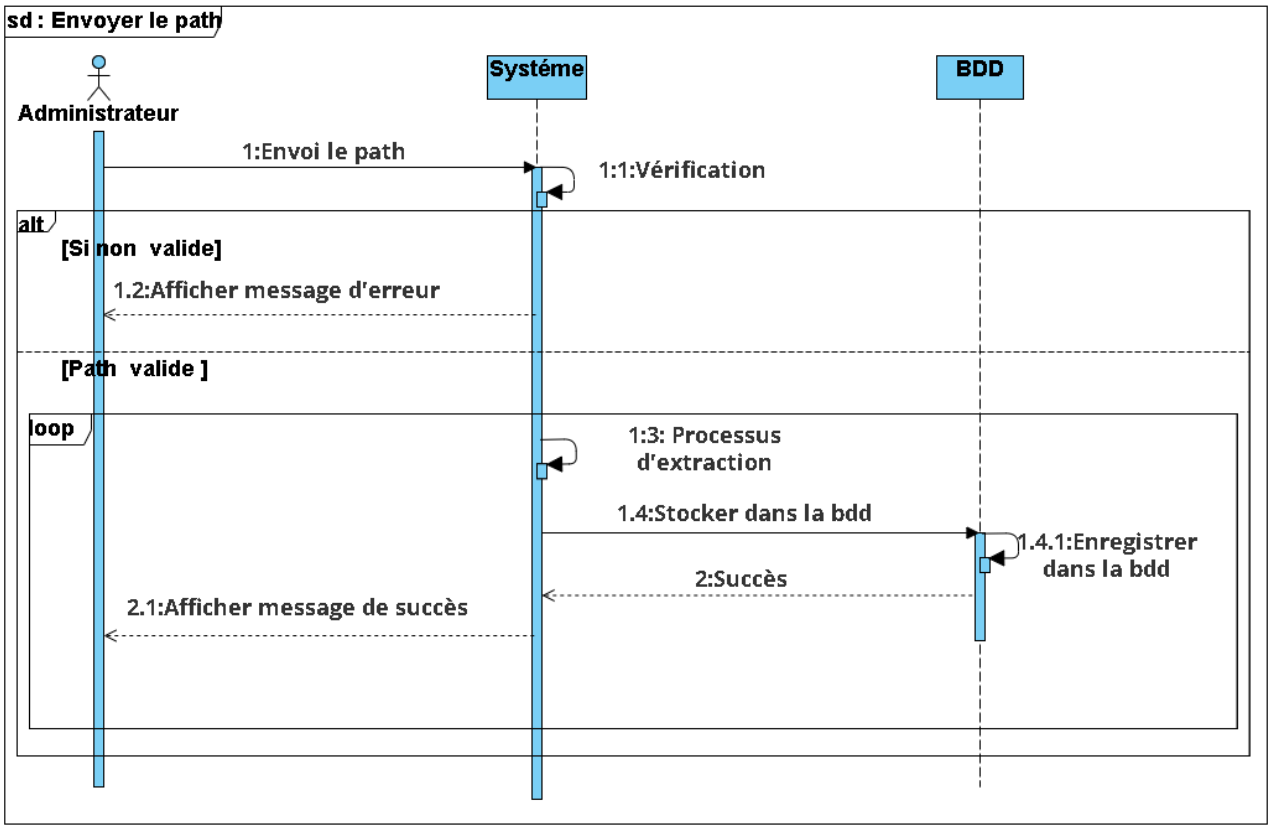


FIGURE III.5 – Diagramme de séquence du cas d'utilisation «Envoyer le path»

III.3 Diagramme de classe

Un diagramme de classe est un type de diagramme de modélisation qui permet de représenter les classes d'un système logiciel, ainsi que les relations qui existent entre elles .[14]

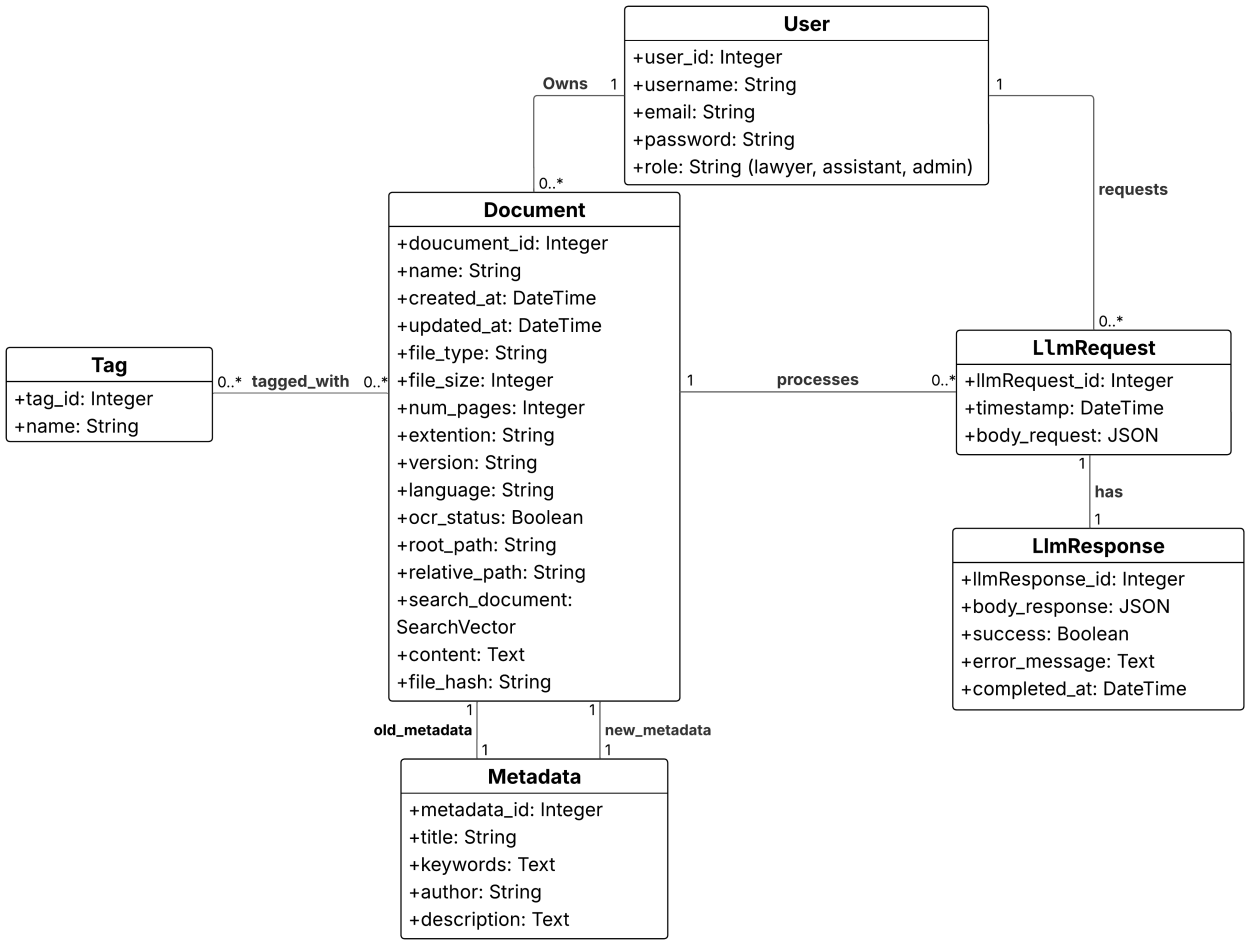


FIGURE III.6 – Diagramme de classe

III.4 Le modèle relationnel

Le modèle relationnel organise les données sous forme de tables. Chaque table représente une structure logique, sans se soucier de la manière dont les données sont stockées physiquement dans le système.[15]

Règles de passage au modèle relationnel

Pour passer d'un modèle conceptuel (comme un diagramme de classes) à un modèle relationnel, on applique les règles suivantes :

- **Règle 1 : Classe → Table**
  Chaque classe devient une table. Ses attributs deviennent les colonnes. Si la classe a un identifiant, il devient la clé primaire. Sinon, on ajoute un identifiant unique.
- **Règle 2 : Association 1 à 1**
  Dans une relation 1..1 entre deux tables, la clé primaire de l'une devient une clé étrangère dans l'autre.
- **Règle 3 : Association 1 à plusieurs (1..*)**
  On applique le même principe que la règle précédente, mais la clé étrangère est toujours placée du côté « plusieurs ».

- **Règle 4 : Association plusieurs à plusieurs (..)**
  On crée une nouvelle table contenant les clés primaires des deux entités concernées. Ces clés deviennent aussi la clé primaire combinée de cette nouvelle table.
- **Règle 5 : Héritage (classe mère et sous-classes)**
  Chaque sous-classe devient une table. La clé primaire de la classe mère est utilisée comme clé primaire dans chaque sous-classe.
- **Règle 6 : Composition**
  La clé primaire de la classe composée est ajoutée comme clé étrangère dans la classe composant.
- **Règle 7 : Agrégation**
  Même principe que pour une association 1 à plusieurs (règle 3).

En appliquant les règles de transformation vers le modèle relationnel, nous avons construit le schéma ci-dessous :

User (user_id, username, email, password, role)

Document (document_id, name, created_at, updated_at, file_type, file_size, num_pages, extention, version, language, ocr_status, root_path, relative_path, search_document, content, file_hash, #owner_id, #new_metadata_id, #old_metadata_id)

Tag (tag_id, name)

Document_Tags (#document_id, #tag_id)

LlmRequest (llmRequest_id, timestamp, body_request, #document_id, #requested_by_id)

LlmResponse (llmResponse_id, body_response, success, error_message, completed_at, #request_id)

Metadata (metadata_id, title, keywords, author, description)

Dictionnaire des données

Dans ce qui suit, nous allons fournir une description détaillée des attributs des différentes tables représentées dans le modèle relationnel.

Table	Attribut	Signification	Type	Taille
User	user_id	Identifiant de l'utilisateur	Integer	11
	username	Nom de l'utilisateur	Varchar	125
	email	Email de l'utilisateur	Varchar	125
	password	Mot de passe de l'utilisateur	Varchar	125
	role	Rôle de l'utilisateur (lawyer - assistant - admin)	Varchar	25
Tag	tag_id	Identifiant du tag	Integer	11
	name	Nom du tag	Varchar	100

Document_Tags	#document_id	Identifiant du document	Integer	11
	#tag_id	Identifiant du tag	Integer	11
Document	document_id	Identifiant du document	Varchar	255
	name	Nom du document	Varchar	255
	created_at	Date de création	DateTime	-
	updated_at	Date de mise à jour	DateTime	-
	file_type	Type de fichier	Varchar	50
	file_size	Taille du fichier	Integer	11
	num_pages	Nombre de pages	Integer	11
	extension	Extension du fichier	Varchar	50
	version	Version du document	Varchar	50
	language	Langue du document	Varchar	50
	ocr_status	Statut OCR	Boolean	1
	root_path	Chemin racine	Varchar	1024
	relative_path	Chemin relatif	Varchar	1024
	search_document	Vecteur de recherche	SearchVector	-
	content	Contenu du document	Text	-
	file_hash	Hachage du fichier	varchar	64
	#owner_id	Identifiant de l'utilisateur (propriétaire)	Integer	11
	#old_metadata_id	Identifiant des anciennes métadonnées	Integer	11
	#new_metadata_id	Identifiant des nouvelles métadonnées	Integer	11
Metadata	metadata_id	Identifiant des métadonnées	Integer	11
	title	Titre du document	Varchar	255
	keywords	Mots-clés	Text	-
	author	Auteur	Varchar	247
	description	Description	Text	-
LlmRequest	llmRequest_id	Identifiant de la requête	Integer	11
	timestamp	la date et l'heure de création de la requête	DateTime	-
	body_request	Corps de la requête (JSON)	JSON	-
	document_id	Identifiant du document	Integer	11
	requested_by_id	Identifiant de l'utilisateur (auteur de la requête)	Integer	11

LlmResponse	llmResponse_id	Identifiant de la réponse	Integer	11
	body_response	Corps de la réponse (JSON)	JSON	-
	success	Succès de la réponse	Boolean	1
	error_message	Message d'erreur	Text	-
	completed_at	Date de complétion	DateTime	-
	request_id	Identifiant de la requête associée	Integer	11

III.5 Conclusion

Dans ce chapitre, nous avons exploré la phase de conception de notre projet en élaborant des diagrammes de séquence précis pour illustrer le fonctionnement du système. Nous avons aussi conçu le diagramme de classes et défini le modèle relationnel, accompagné d'une description complète des attributs des tables de la base de données. Le chapitre suivant, dédié à la réalisation, abordera l'implémentation de l'application, en présentant les outils, l'architecture globale et certaines interfaces.

Chapitre IV

Réalisation

IV.1 Introduction

Après avoir réalisé la conception appropriée à notre application, nous allons dans ce chapitre décrire la mise en œuvre technique de notre plateforme, nous commençons par décrire l'environnement matérielle et logicielle. Ensuite nous procédons à la présentation du produit final de notre application, en mettant en avant l'architecture globale que nous avons utilisés pour le développement, et un aperçu des principales interfaces graphiques.

IV.2 Environnement du travail

IV.2.1 Environnement matériel

Pour la réalisation de ce projet, nous disposons de deux micro-ordinateurs. Le premier est équipé d'un système Intel Core i5-5200U, une mémoire de 8 GB, un disque dur de 256 GB SSD, un écran de résolution 1366×768 pixels, et le système d'exploitation Linux Mint 22.1 Cinnamon. Le second dispose d'un processeur Intel Core i7-6600U, une mémoire de 8 GB, un disque dur de 256 GB SSD, un écran de résolution 1366×768 pixels, et le système d'exploitation Windows 11.

IV.2.2 Environnement logiciel

Nous allons lister au cours de cette partie les différents outils utilisés tout au long de ce projet pour l'étude et la mise en place de notre plateforme.

IV.2.2.1 Outil de modélisation UML

Lucidchart :

Lucidchart est une solution de modélisation en ligne qui permet de représenter visuellement des structures organisationnelles, des processus métiers et des systèmes techniques. Cette solution permet de mettre en images des idées complexes, suivre des projets et facilite la collaboration.

Le logiciel met à la disposition de ses utilisateurs de nombreuses fonctionnalités, dont : la création de diagramme, la collaboration en temps réel et la visualisation de vos données dans leur contexte.[16]

FIGURE IV.1 – lucidchart logo

Visual Paradigm :

Est un outil de conception logicielle professionnel destiné aux développeurs et concepteurs. Il prend en charge la création de diagrammes UML complexes, la génération automatique de

code et la collaboration en équipe. Réputé dans le domaine de l'ingénierie logicielle, cet outil améliore considérablement les processus de conception tout en optimisant la communication entre les différents membres de l'équipe de développement. [17]

FIGURE IV.2 – Visual Paradigm logo

IV.2.2.2 Environnement de développement

Pour le développement de notre application, nous avons utilisé un riche écosystème d'extensions pour la programmation dans divers langages.

Visual Studio Code (VS Code) :

Est un éditeur gratuit et performant de code source multiplateformes, se distingue par sa légèreté, lui permettant de fonctionner de manière fluide sur la plupart des configurations matérielles, offrant une grande compatibilité avec plusieurs langages de programmation grâce à des extensions. Il dispose de fonctionnalités avancées telles que l'auto-complétion, la coloration syntaxique et le débogage, ainsi que l'intégration de commandes git. Idéal pour les développeurs, quel que soit le niveau de compétence. VS Code offre une expérience de codage à la fois accessible et professionnelle.[18]

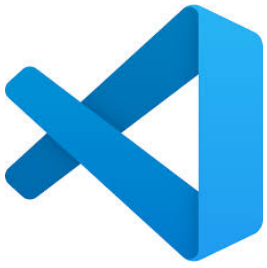


FIGURE IV.3 – Visual Studio Code logo

Docker :

Docker est une plateforme de conteneurisation open-source qui simplifie le déploiement d'applications en encapsulant les applications et leurs dépendances dans des environnements isolés appelés conteneurs. Contrairement aux machines virtuelles traditionnelles, les conteneurs Docker partagent le noyau du système d'exploitation hôte, ce qui les rend plus efficaces et plus légers.

Ils garantissent une cohérence de fonctionnement entre les environnements de développement, test et production. Cela permet de réduire les problèmes de compatibilité et d'améliorer la portabilité sur différentes plateformes. En raison de sa flexibilité et de son évolutivité, Docker est devenu un outil crucial dans les flux de travail modernes de DevOps et de développement cloud-native.[19]

FIGURE IV.4 – Docker logo

PostgreSQL :

Est un système de gestion de base de données open source gratuit qui prend en charge les requêtes relationnelles (SQL) et non relationnelles (JSON). PostgreSQL est largement utilisé pour les applications web, notamment grâce à sa conformité aux standards et ses performances élevées, Fiable et compatible avec de nombreux langages (Python, Java, C/C++, etc.).[20]

FIGURE IV.5 – PostgreSQL logo

IV.2.2.3 Langages et Framework développement

La mise en œuvre du projet a nécessité l'utilisation de plusieurs langages de programmation orientés objet ainsi que des frameworks modernes adaptés au développement que ce soit côté client (frontend) ou côté serveur (backend).

Python :

Python est un langage de programmation orienté objet clair et puissant, utilisé pour le développement du backend. Il est reconnu pour sa lisibilité, sa richesse en bibliothèques et son intégration facile avec les frameworks web comme Django. Utilise une syntaxe élégante, rendant les programmes plus faciles à lire. Cela rend Python idéal pour le développement de prototypes ou d'autres tâches de programmation, sans compromettre la maintenabilité.

Compatible avec tous les systèmes majeurs Windows, macOS, Linux. Pour la réalisation de ce projet, nous avons utilisé la version 3.12.3[21]

FIGURE IV.6 – Python logo

Django REST Framework(drf) :

Extension puissante et flexible du framework Django, conçue pour créer des API Web robustes. Il fournit une boîte à outils complète, permettant de gérer toutes les fonctionnalités nécessaires à l'exposition de données via une interface web, notamment dans le respect de l'architecture REST.

DRF facilite l'interfaçage entre les bases de données (via l'ORM Django) et les clients frontend ou autres services via des formats d'échange standard comme JSON ou XML.

Voici quelques raisons pour lesquelles choisir le REST framework :

- L'API web navigable est un atout majeur en termes d'ergonomie pour les développeurs.
- Des politiques d'authentification sont prises en charge, y compris des packages pour OAuth1a et OAuth2.
- La sérialisation prend en charge à la fois les sources de données ORM (comme les modèles Django) et non-ORM.
- Le Framework est entièrement personnalisable, vous pouvez simplement utiliser des vues basées sur des fonctions classiques si vous n'avez pas besoin des fonctionnalités plus avancées.
- Une documentation complète et une communauté active facilitent l'apprentissage et le développement. Pour la réalisation de notre projet nous avons utilisé la version 3.15.2[22]

FIGURE IV.7 – Django REST Framework logo

Django :

Est un framework Python de haut niveau open source, conçu pour permettre un développement rapide d'applications web sécurisées, fiables et faciles à maintenir. Créé par des développeurs expérimentés, il prend en charge de nombreuses tâches courantes du développement web (authentification, gestion de la base de données, sécurité, administration, etc.), ce qui vous permet de vous concentrer sur la logique métier de votre application sans avoir à "réinventer la roue".

Il est :

- Gratuit et open source

- Doté d'une documentation complète
- Soutenu par une communauté active
- Facilement extensible avec de nombreuses bibliothèques tierces

Django s'appuie sur le modèle MTV (Model-Template-View), qui permet de structurer clairement le code d'une application. Pour la réalisation de notre projet nous avons utilisé la version 5.0.13.[23]

FIGURE IV.8 – Django logo

JavaScript :

Est un langage de script, multi-plateforme et orienté objet. C'est un langage léger qui doit faire partie d'un environnement hôte (un navigateur web par exemple) pour qu'il puisse être utilisé sur les objets de cet environnement. JavaScript côté client étend ces éléments de base en fournissant des objets pour manipuler le navigateur et le Document Object Model (DOM). D'interagir avec l'utilisateur (clics, formulaires, navigation) et d'enrichir l'interface, son utilisation courante dans le développement web vise à améliorer l'expérience utilisateur en rendant les applications web dynamiques plus interactifs et réactifs.[24]

FIGURE IV.9 – JavaScript logo

Vue JS :

Vue est un framework JavaScript progressif utilisé pour construire une interface utilisateur réactive, et moderne et un écosystème qui couvre la plupart des fonctionnalités courantes nécessaires au développement front-end, Vue est basé sur une architecture par composants. Chaque partie de l'interface est encapsulée dans un composant réutilisable et gère efficacement l'état de l'application côté client. Le Web est très diversifié les choses que nous y construisons peuvent varier radicalement. C'est pourquoi Vue a été conçu pour être flexible et adoptable de manière incrémentale. Pour la réalisation de notre projet nous avons utilisé la version 3.5.13[25]

FIGURE IV.10 – Vue JS logo

Vuetify :

Vuetify est une bibliothèque de composants graphiques pour Vue JS. Elle est basée sur Material Design. Elle est disponible gratuitement et open source. Elle est très simple à utiliser et permet de créer des interfaces web élégantes cohérentes et responsives très rapidement, sans avoir à écrire manuellement du CSS complexe. Pour la réalisation de notre projet nous avons utilisé la version 3.8.1[26]

FIGURE IV.11 – Vuetify logo

Pinia :

Pania est une bibliothèque officielle et super pratique pour gérer l'état global dans Vue.js. Elle se présente comme une alternative moderne à Vuex plus légère, plus ergonomique, et entièrement compatible avec Vue 3

Pourquoi Choisir Pinia pour Gérer l'État dans Vue.js ?

Dans les applications Vue.js, gérer un état global partagé entre plusieurs composants peut devenir un vrai casse-tête. Pinia simplifie tout ça en offrant une structure centralisée où tu peux stocker et manipuler toutes tes données de manière fluide et propre. Pour la réalisation de notre projet nous avons utilisé la version 3.0.2[27]

FIGURE IV.12 – Pinia logo

Axios :

Est une bibliothèque JavaScript permettant d'effectuer facilement des requêtes HTTP depuis le navigateur ou un environnement Node.js. Basée sur le système de promesses (Promises) de JavaScript, elle simplifie la gestion du code asynchrone, le rendant plus lisible et maintenable. Très utilisée dans le développement front-end, notamment avec Vue.js, Axios facilite la communication avec des API REST, que ce soit pour récupérer ou envoyer des données.

Dans le cadre de notre projet, nous avons utilisé la version 1.9.0 d'Axios pour assurer les échanges entre l'interface utilisateur et le backend.[28]

FIGURE IV.13 – Axios logo

IV.2.2.4 Plateforme des services Web et API

Un service Web est une application accessible en ligne, conçue pour échanger des données avec d'autres systèmes via des protocoles standards comme HTTP. Parmi les plus courants, on trouve les **API REST**, qui permettent d'interagir facilement entre applications.

Cette partie englobe les outils et technologies utilisés pour communiquer, tester, documenter et interagir avec des services web, ainsi que pour partager et versionner le code source de manière collaborative.

Insomnia :

Est un **client HTTP** open-source très utilisé pour tester les **API REST**. qui vous permet d'envoyer des requêtes HTTP et de visualiser les réponses (JSON, XML, etc.).

Insomnia est un outil intuitif qui simplifie le test et le débogage des API. Il permet aux développeurs de concevoir, envoyer et gérer facilement leurs requêtes HTTP. Grâce à une interface organisée en espaces de travail, il est possible de structurer ses requêtes, de les enregistrer, puis de les partager avec d'autres membres de l'équipe.[29]

FIGURE IV.14 – Insomnia logo

GitHub :

GitHub est une plateforme cloud qui permet de stocker, gérer et collaborer sur du code. En déposant votre code dans un référentiel (repository), vous pouvez :

- Partager facilement vos projets avec d'autres.

- Suivre l'historique des modifications et revenir à des versions antérieures si nécessaire.
- Recevoir des retours ou suggestions via des pull requests.
- Travailler en équipe sur un même projet sans risquer de perturber le travail des autres avant la validation des changements.

GitHub repose sur **Git**, un système de contrôle de version open source, qui rend possible cette collaboration efficace et sécurisée.[30]

FIGURE IV.15 – GitHub logo

GIT :

Git est un système de gestion de version décentralisé qui permet de suivre efficacement les modifications apportées aux fichiers, notamment lors du travail collaboratif. Il est particulièrement utile lorsque plusieurs personnes modifient simultanément le même projet.

Dans un flux de travail typique avec Git :

- Chaque développeur crée une branche à partir de la version principale (souvent appelée main ou master).
- Les modifications sont effectuées de manière isolée sur cette branche, sans impacter les autres.
- Une fois prêtes, les modifications peuvent être fusionnées (merge) dans la branche principale. Git gère intelligemment les différences pour éviter les conflits.
- Git garde une trace précise de chaque modification, permettant ainsi de revenir à une version antérieure ou de suivre l'historique des contributions.[31]

FIGURE IV.16 – GIT logo

API :

Une API (Application Programming Interface) est un ensemble de règles qui définit comment deux logiciels peuvent interagir. Elle agit comme un contrat entre un fournisseur de services et un consommateur : le client envoie une requête selon un format défini, et le serveur lui retourne une réponse correspondante. Elle sert donc d'intermédiaire entre les utilisateurs et les ressources, tout en permettant un accès contrôlé, sécurisé et standardisé aux données.[32]

API LLM Mistral(locale) :

Désigne une interface de programmation d'application (API) permettant d'interagir avec un modèle de langage de grande taille (LLM) open source, comme Mistral, installé directement sur un serveur privé pour une meilleure confidentialité. Cette API peut être utilisée via une clé d'authentification, et permet d'exploiter la puissance du traitement du langage naturel (NLP) pour réaliser des tâches avancées telles que : générer du texte, résumer, répondre à des questions, ou extraire des métadonnées de documents, sans recourir à des services cloud tiers.[33]

FIGURE IV.17 – Mistral logo

API REST :

Une API REST est un type d'API qui suit les principes de l'architecture **REST (REpresentational State Transfer)**. Elle repose sur le protocole **HTTP** pour transmettre des données et interagir avec des ressources (utilisateurs, produits, documents...).

Elle est conçue autour de plusieurs principes fondamentaux :

- Architecture client-serveur : le client envoie des requêtes, le serveur y répond.
- Stateless : chaque requête est indépendante, le serveur ne conserve pas l'état du client.
- Mise en cache : les données peuvent être mises en cache pour plus d'efficacité.
- Interface uniforme : chaque ressource est identifiée par une URL unique (URI), et manipulée via des méthodes HTTP standards comme GET, POST, PUT, DELETE.
- Format de réponse : souvent en JSON, lisible par l'homme et la machine.[34]

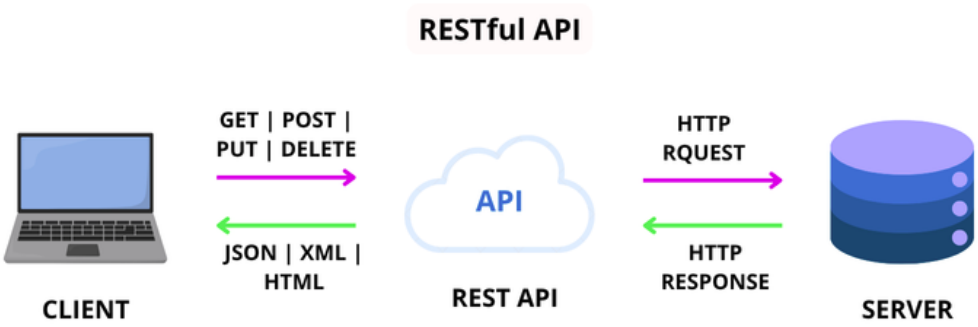


FIGURE IV.18 – Architecture REST API

IV.3 L'architecture globale de l'application

La figure suivante illustre l'architecture globale de notre application de manière visuelle et simplifiée.

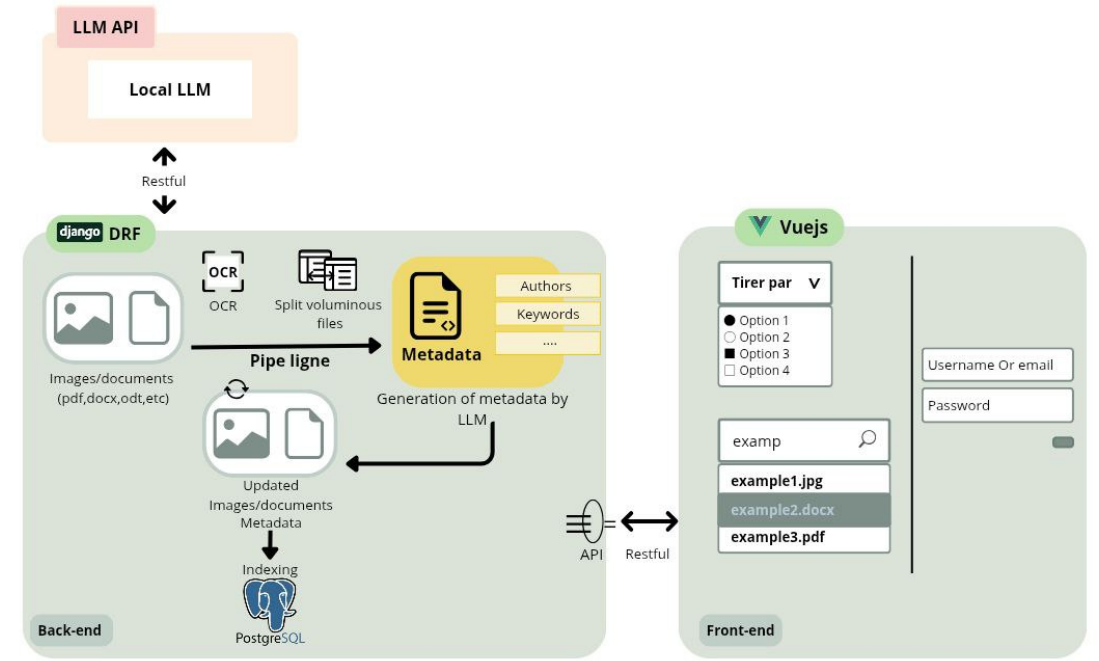


FIGURE IV.19 – Architecture globale de l'application

La plateforme repose sur une architecture modulaire orientée services, répartie en trois grandes couches : le front-end, le back-end, et un service d'intelligence artificielle local (LLM). Ces couches communiquent entre elles via des API RESTful, assurant ainsi un découplage fort, une interopérabilité maximale, et une évolutivité facilitée.

1. **Front-end**
   Les interfaces développées avec **Vue js** permettent à l'utilisateur de :
   - S'authentifier avec un formulaire username ou e-mail / password.
   - Rechercher des documents par nom et contenu.
   - Visualiser les documents disponibles (JPG, DOCX, PDF...).
   - Filtrer les documents à l'aide de plusieurs critères (type de fichier, tags, date).
2. **Back-end**
   Développées avec Django et Django REST Framework , met en œuvre un pipeline de traitement documentaire automatisé qui gère :
   - le traitement des documents variés (PDF, DOCX, ODT...).
   - La reconnaissance de texte via un moteur OCR intégré.
   - La gestion des fichiers volumineux en les découpant si nécessaire.
   - Le stockage et l'indexation des documents et leurs métadonnées dans une base de données PostgreSQL.
   - L'extraction et l'enrichissement des métadonnées (titre, auteur, mots-clés...) via un LLM local.

Ce pipeline suit les étapes suivantes :

- OCR et traitement préliminaire
- Extraction de contenu textuel
- Requête au LLM
- Mise à jour du document avec les métadonnées générées
- Sauvegarde et indexation dans PostgreSQL

3. **Service LLM Local (Intelligence Artificielle)**
   - Il s'agit d'un LLM (Large Language Model) déployé localement.
   - Accessible via une API RESTful.
   - Reçoit le contenu textuel du document, puis retourne des métadonnées générées : auteur, mots-clés, description, etc.
   - Fonctionne indépendamment du back-end, mais intégré dans le pipeline via une requête API.

IV.4 Présentation des interfaces de l'application

Cette section décrit les interfaces utilisateur développées dans le cadre du projet, ainsi que les principales fonctionnalités accessibles en fonction du rôle de chaque utilisateur.

IV.4.1 Interface «Authentification»

Cette interface constitue la porte d'entrée de la plateforme et permet à un utilisateur d'accéder aux fonctionnalités de l'application d'une façon sécurisée. Avec un formulaire de connexion contenu ces champs

- Nom d'utilisateur ou e-mail : permet à l'utilisateur de saisir son identifiant.
- Mot de passe : avec un champ masqué sécuriser.

En cas de succès, il sera redirigé vers l'interface de recherche selon son rôle sinon un message d'erreur sera affiché

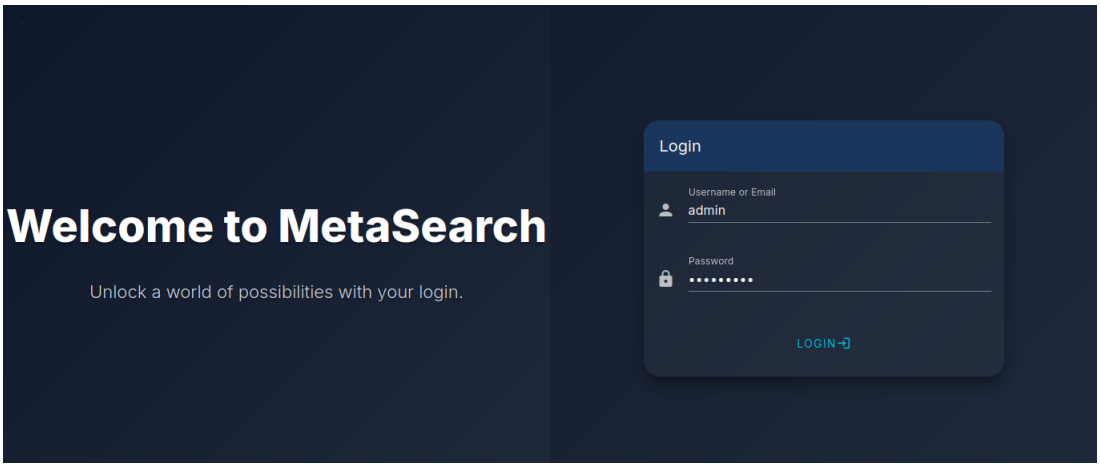


FIGURE IV.20 – Interface «Authentification»

IV.4.2 Interface «Utilisateur»

Après l'authentification en tant qu'utilisateur (assistant, avocat) il sera redirigé vers cette interface où pourra effectuer une recherche d'un document qui est déjà traité.

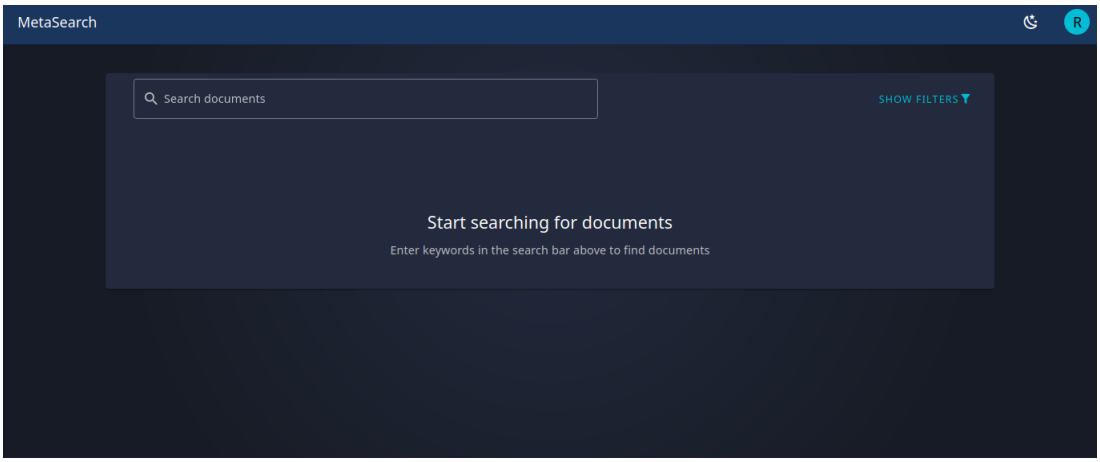


FIGURE IV.21 – Interface «Utilisateur»

IV.4.3 Interface «Administrateur»

Après l'authentification l'administrateur accède à cette interface où il pourra faire les mêmes fonctionnalités en tant qu'utilisateur, aussi que l'ajout d'un path d'un document.

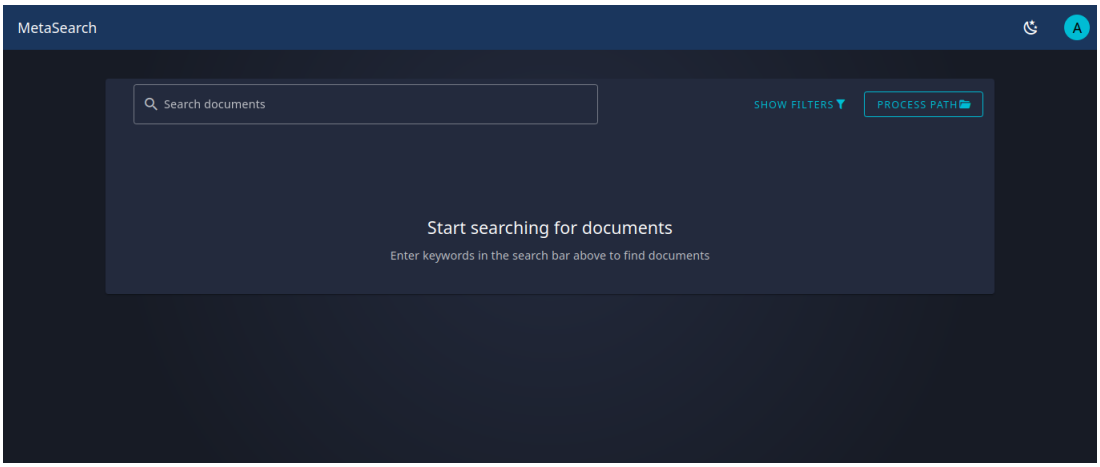


FIGURE IV.22 – Interface «Administrateur»

IV.4.4 Interface «Envoyer d'un path»

Comme on a cité avant s'il n'y a pas de document traité déjà ou dans le cas d'ajout ou de modification l'administrateur vas procéder à saisir et envoyer le path de ces documents pour déclencher le processus d'extraction afin de les rendre traités. Cette action accessible uniquement par l'administrateur.

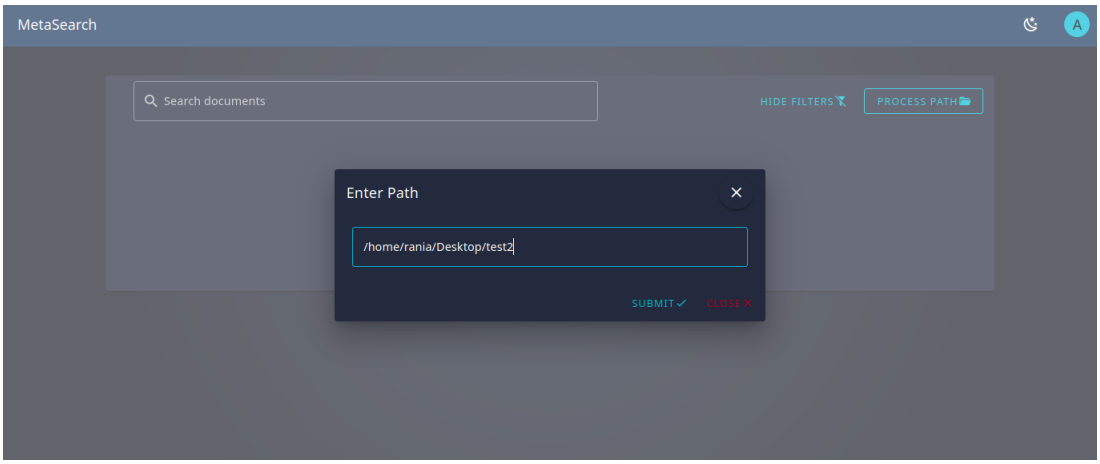


FIGURE IV.23 – Interface «Envoyer d'un path»

IV.4.5 Interface «Processus»

Cette interface est comme un suivi comme elle montre bien ici que le processus d'extraction de métadonnées est en cours d'après le message si dessus, afin de mettre le client au courant.

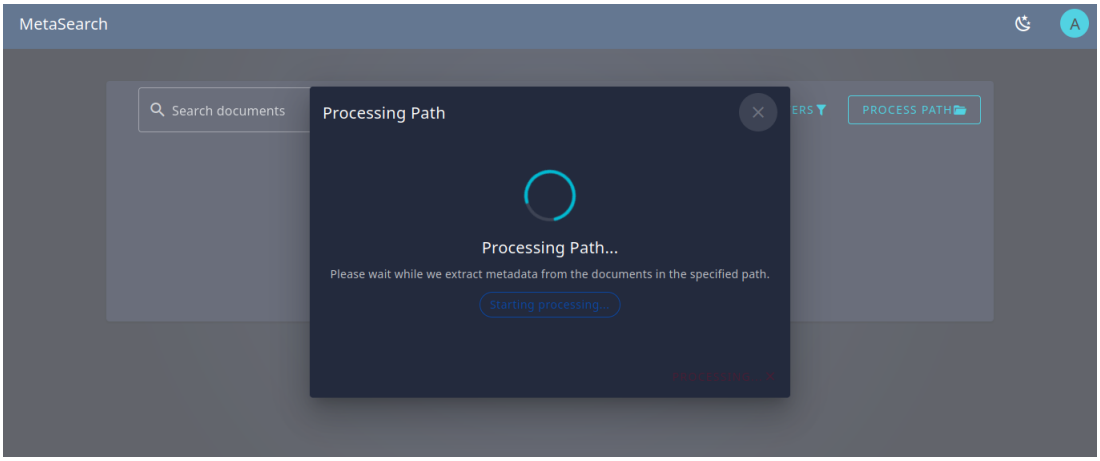


FIGURE IV.24 – Interface «Processus»

IV.4.6 Interface «Processus avec succès»

l'interface suivante présente la fin d'exécution de processus avec le résultat obtenu, d'après le message de succès et statistique si dessus qui preuve bien ça.

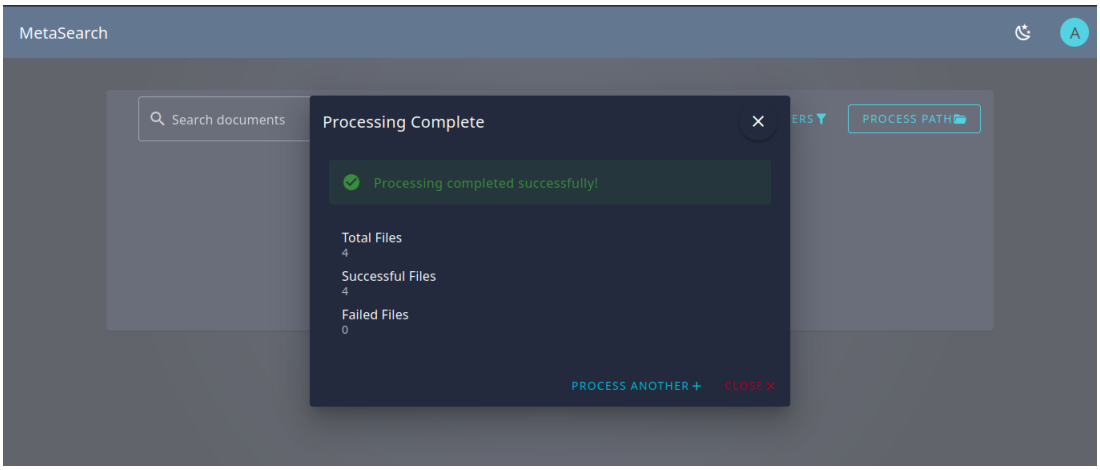


FIGURE IV.25 – Interface «Processus avec succès»

IV.4.7 Interface «Recherche»

C'est l'interface de recherche qui est la principale fonctionnalité, où un utilisateur peut l'effectuer uniquement avec la saisie de trois lettres, qui s'agit du nom de document traité lui-même ou mot clé parmi son contenu, et le système affichera le résultat d'une manière dynamique comme ceci.

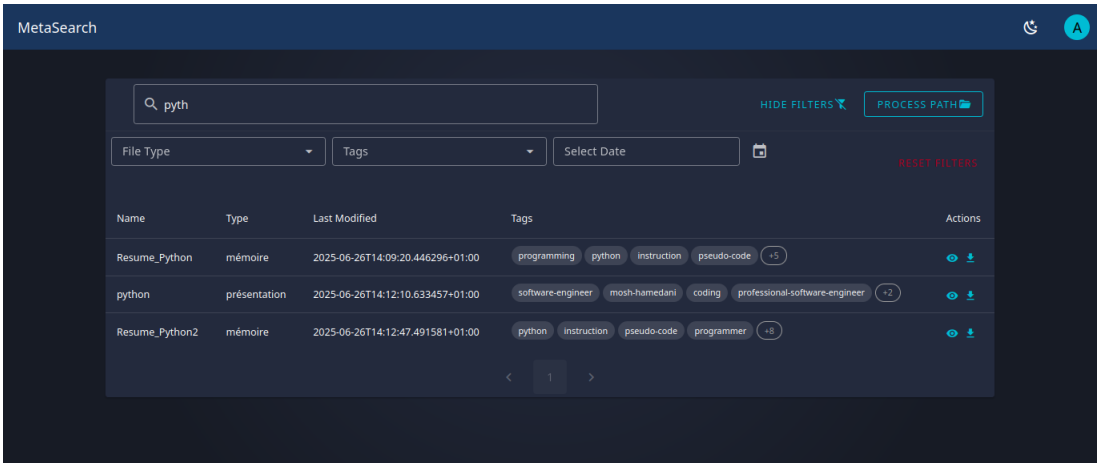


FIGURE IV.26 – Interface «Recherche»

IV.4.8 Interface «Choix de tri»

Cette figure montre bien les filtres possibles pour une recherche, autrement dit après qu'un utilisateur effectue une recherche il aura le choix des filtres selon type de document ou date de création ou même ces catégories, pour une recherche plus optimisée.

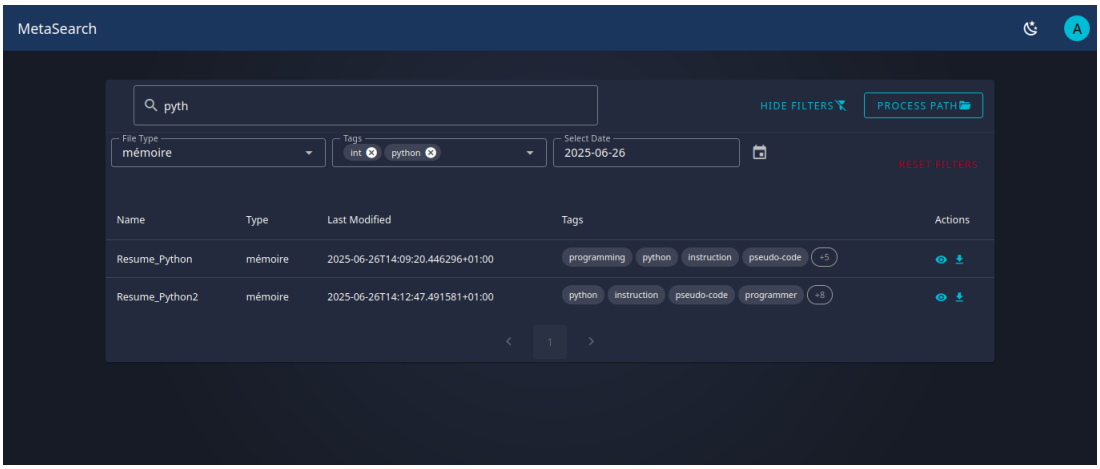


FIGURE IV.27 – Interface «Choix de tri»

IV.4.9 Interface «Consultation»

L'interface de consultation affiche bien un aperçu d'un document avec des métadonnées extraites déjà, et ceci bien sur après la recherche dont il peut visualiser les informations de ce document pour une meilleure confirmation avant de le télécharger.

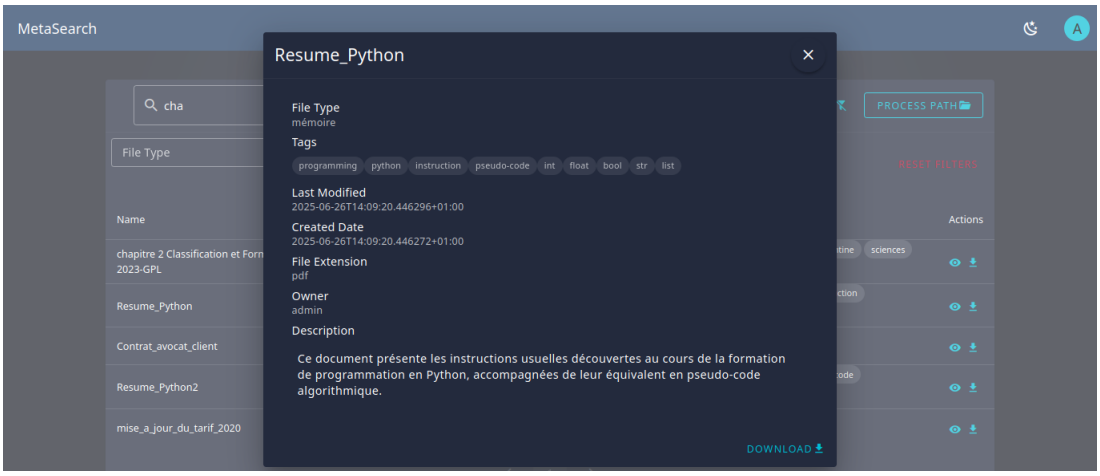


FIGURE IV.28 – Interface «Consultation»

IV.4.10 Interface «Téléchargement»

La figure suivante montre bien la fonctionnalité de téléchargement d'un document où l'utilisateur a le choix de le voir avant d'effectuer cette action ou directement le télécharger.

Et cette option marchera uniquement pour les documents détecter par le OS c'est-à-dire sont pas mal déplacer et existe toujours dans leur emplacement prévu d'après le path.

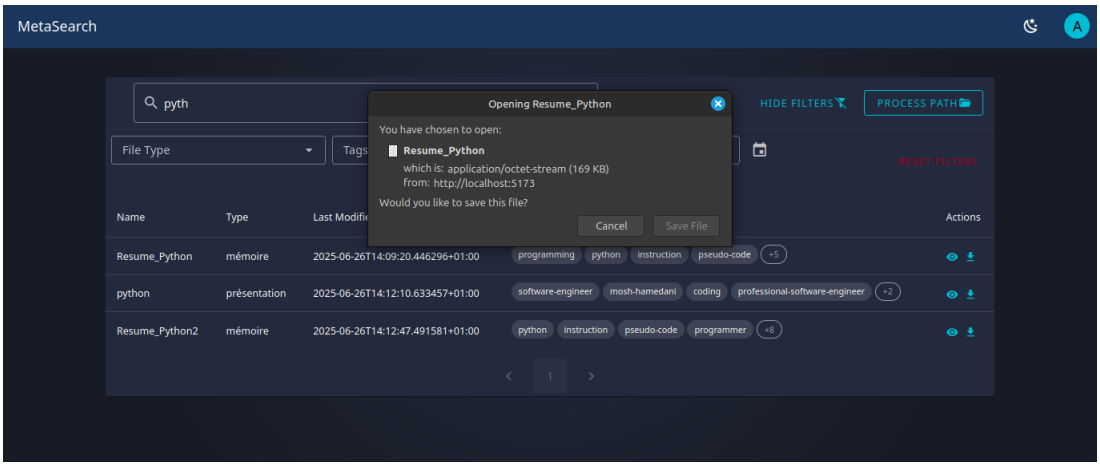


FIGURE IV.29 – Interface «Téléchargement»

IV.5 Conclusion

Dans ce dernier chapitre, nous avons présenté la partie réalisation en décrivant l'environnement du travail aussi bien matériel que logiciel, dont nous avons abordé les différents outils et langages de programmation utilisés, ainsi que l'architecture globale adopté.

En dernier, nous avons illustré quelques interfaces qui souligne les fonctionnalités de base de notre application.

Conclusion générale

Tout au long de ce mémoire, nous avons décrit les différentes étapes de conception et de réalisation d'une plateforme intelligente d'extraction de métadonnées et de recherche documentaire, alimentée par un LLM local. Ce travail a été mené en suivant une démarche méthodique allant de l'étude de l'existant jusqu'à la mise en œuvre technique. L'élaboration de ce projet a suivi plusieurs étapes, dans un premier temps, nous avons présenté l'organisme d'accueil et l'étude de l'existant qui nous a permis d'avoir un aperçu des solutions disponibles pour mieux comprendre les besoins du terrain ce qui nous a aidés à identifier les limites des solutions actuellement utilisées et à proposer une telle plateforme.

Ensuite, pour la méthodologie de développement, nous avons adopté une approche agile basée Scrum. Ce choix nous a permis de structurer notre travail en itérations appelées sprints, assurant ainsi un développement d'un système progressif, de qualité pour chaque itération afin de le rendre plus fidèle au besoin du client, Puis, nous avons procédé à une analyse et spécification des besoins afin d'identifier les besoins fonctionnels et non fonctionnels pour notre application suivie d'une modélisation des déférent cas d'utilisations.

Par la suite, nous avons entamé la phase de conception dont nous avons montré clairement les interactions du système, le fonctionnement de l'application et l'élaboration d'une structure globale de l'application. Enfin, la phase de réalisation a été dédiée au développement de notre plateforme. Nous avons défini l'environnement matériel ou on a effectué le codage et la programmation, aussi bien que logiciel qui liste différents outils, de langages et de frameworks utilisé, cette phase a également inclus l'architecture globale et une démo des interfaces de l'application.

Ce projet nous a offert une réelle opportunité de nous initier à un environnement professionnel concret, de mettre en pratique nos compétences techniques, de travailler en équipe et de gérer les différentes contraintes, notamment celle du temps. Il nous a aussi permis de mieux comprendre l'importance de l'organisation, de la rigueur et de l'autonomie dans la réalisation d'un projet technologique.

Cette expérience a été particulièrement enrichissante, car elle nous a permis d'approfondir nos compétences en programmation, d'explorer des technologies avancées comme les LLM locaux, et de mieux comprendre les enjeux liés à la gestion documentaire intelligente.

Elle nous a également permis de développer notre sens des responsabilités et de nous initier à la gestion de projet, compétences essentielles pour une future insertion dans le monde professionnel.

Enfin, notre travail ne s'arrête pas là. De nombreuses perspectives d'amélioration sont envisageables pour renforcer notre application. Nous prévoyons notamment :

- L'intégration d'un chabot intelligent basé sur le LLM, capable de dialoguer avec les utilisateurs pour les aider à rechercher des documents, poser des questions sur les métadonnées, ou naviguer dans la plateforme de manière fluide.
- L'intégration de Celery comme système de gestion de tâches asynchrones. Afin d'éviter le blocage de l'interface utilisateur ni le serveur principal.

Bibliographie

[1] Nextcloud. URL https://nextcloud.com. Consulté le 15/03/2025.

[2] Comparaison entre processus unifié et scrum. URL https://www.scribd.com/document/500866126/Comparaison-entre-Processus-Unifie-et-Scrum-1. Consulté le 17/03/2025.

[3] Tuleap. Comprendre la méthode agile scrum en 10min, 2025. URL https://www.tuleap.org/fr/agile/comprendre-methode-agile-scrum-10-minutes. Consulté le 17/03/2025.

[4] Scrum-Master·Org. extreme programming (xp) : Guide pour débutants en méthode agile, 2025. URL https://scrum-master.org/lextreme-programming-xp-guide-pour-debutants-en-methode-agile/. Consulté le 17/03/2025.

[5] Vollcom Digital GmbH. The 6 stages of a scrum sprint, December 2022. URL https://vollcomdigital.medium.com/the-6-stages-of-a-scrum-sprint-de0857839573. Article de blog sur Medium, consulté le 17/03/2025.

[6] Kadambini A. Kadam. *The Object-Oriented Approach to Problem Solving and Software Design*. Dreamtech Press, 2012. ISBN 9789380298441. URL https://www.google.dz/books/edition/The_Object_Oriented_Approach_to_Problem/_bVPEQAAQBAJ. Consulté le 17/03/2025.

[7] Métadonnées. URL https://commentouvrir.com/tech/meta-definition-et-explications-sur-les-metadonnees/. Consulté le 16/03/2025.

[8] Llm. URL https://www.cloudflare.com/fr-fr/learning/ai/what-is-large-language-model/. Consulté le 16Mars2025.

[9] Nora Rezaiki. Conception et réalisation d'une application mobile pour la gestion d'un cabinet d'avocat [en ligne], 2019. URL https://theses-algerie.com/2536641431496860/memoire-de-master/centre-universitaire-abdel-hafid-boussouf---mila/conception-et-r%C3%A9alisation-d-une-application-mobile-pour-la-gestion-d-un-cabinet-d-avocat. Mémoire de Master, Centre Universitaire Abdelhafid Boussouf - Mila, consulté le 09/03/2025.

[10] Rouabah Eleulmi Benmahbous, Abid. Réalisation d'une application web de gestion de cabinet d'avocat [en ligne], 2020. URL https://dspace.univ-bba.dz/items/6bf69436-fd1c-43d1-aca2-bdccbdc1b686. Mémoire de Master, Université de Bordj Bou Arréridj, consulté le 10/03/2025.

[11] Lylia HADRI. Conception et realisation d'une application mobile de livraison [en ligne], 2021. URL https://theses-algerie.com/1237797087262469/memoire-de-master/universite-abderrahmane-mira-bejaia/modelisation-et-systemes-dynamiques. Mémoire de Master, Université AbderrahmaneMira – Béjaïa, consulté le 15/03/2025.

[12] KSOURI Ouarda AIT ARAB, Yousra. Conception et réalisation d'une application web pour la gestion des ressources humaines, 2023. URL https://www.univ-bejaia.dz/xmlui/handle/123456789/23479. Mémoire de Master, Université Abderrahmane Mira – Béjaïa, consulté le 25/03/2025.

[13] AIT OUSEKHRI Wissam ASBAI, Lahna. Conception et réalisation d'une plateforme pour le suivi thérapeutique et pédagogique des patients [en ligne], 2023. URL https://www.univ-bejaia.dz/xmlui/handle/123456789/23085. Mémoire de Master, Université Abderrahmane Mira – Béjaïa, consulté le 25/04/2025.

[14] Laurent Audibert. Diagramme de classes – uml 2 : de l'apprentissage à la pratique, October 2006. URL https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-classes. Consulté le 27/04/2025.

[15] Centre Universitaire de Mila. Passage vers le modèle relationnel, 2024. URL https://elearning.centre-univ-mila.dz/a2024/pluginfile.php/100437/mod_resource/content/1/passage%20diag%20classe%20relation.pdf. Support de cours, consulté le 18/03/2025.

[16] Lucidchart. URL https://www.blogdumoderateur.com/tools/lucidchart/. Consulté le 10/05/2025.

[17] Visual paradigm. URL https://online.visual-paradigm.com/fr/diagrams/solutions/free-visual-paradigm-online/. Consulté le 15/05/2025.

[18] Vs code. URL https://code.visualstudio.com/docs/setup/setup-overview. Consulté le 15/05/2025.

[19] Docker ? URL https://docs.docker.com/get-started/docker-overview/. Consulté le 12/06/2025.

[20] Postgresql. URL https://www.w3schools.com/postgresql/postgresql_intro.php. Consulté le 08/06/2025.

[21] Python. URL https://docs.python.org/3/whatsnew/3.12.html. Consulté le 08/06/2025.

[22] Django rest framework. URL https://www.django-rest-framework.org/. Consulté le 08/06/2025.

[23] Django. URL https://developer.mozilla.org/fr/docs/Learn_web_development/Extensions/Server-side/Django/Introduction. Consulté le 15/05/2025.

[24] Javascript. URL https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction. Consulté le 08/06/2025.

[25] Vuejs, . URL https://vuejs.org/guide/introduction.html. Consulté le 08/06/2025.

[26] Vuetify, . URL https://v2.vuetifyjs.com/en/introduction/why-vuetify/. Consulté le 12/06/2025.

[27] Pinia. URL https://pinia.vuejs.org/introduction.html. Consulté le 12/06/2025.

[28] Axios. URL https://axios-http.com/docs/intro. Consulté le 12/06/2025.

[29] Insomnia. URL https://docs.proabono.com/fr/documentation/api-presentation/qu-est-ce-que-insomnia-le-client-api-rest/. Consulté le 08/06/2025.

[30] Github, . URL https://docs.github.com/fr/get-started/start-your-journey/about-github-and-git. Consulté le 08/06/2025.

[31] Git, . URL https://docs.github.com/fr/get-started/start-your-journey/about-github-and-git. Consulté le 08/06/2025.

[32] Api, api-rest, . URL https://www.redhat.com/fr/topics/api/what-is-a-rest-api. Consulté le 20/06/2025.

[33] Api llm mistral, . URL https://www.redhat.com/fr/topics/api/what-is-a-rest-api. Consulté le 08/06/2025.

[34] Architecture api rest. URL https://miro.medium.com/v2/resize:fit:880/1*O5cUaAZp9blbN9RuSeglLQ.png. Consulté le 18/06/2025.

Résumé

Ce mémoire a été réalisé dans le cadre d'un projet de fin d'études en Master Génie Logiciel. Il porte sur la conception et la mise en œuvre d'une plateforme avancée permettant l'extraction automatique de métadonnées et la recherche de documents numériques, en s'appuyant sur un modèle de langage local (LLM).

Pour mettre en œuvre notre solution, nous avons adopté la méthode agile SCRUM comme cadre de gestion de projet, tout en s'appuyant sur le langage UML pour la modélisation. Le backend est développé avec Django et Django REST Framework, assurant le traitement de documents variés (PDF, DOCX, ODT...), l'extraction de texte via OCR, l'enrichissement des métadonnées par un LLM, et le stockage dans une base PostgreSQL. Le frontend est réalisé avec Vue.js, Pinia et Vuetify, permettant une interface utilisateur interactive.

Les tests d'API ont été réalisés à l'aide de l'outil Insomnia, et le code est versionné via Git et hébergé sur GitHub. L'architecture repose également sur l'utilisation de conteneurs Docker pour une meilleure portabilité.

Mots clés : LLM, Django, API REST, Vue.js, OCR, métadonnées, PostgreSQL, Docker, GitHub, Pinia, SCRUM, UML.

Abstract

This thesis was carried out as part of a final year project for a Master's degree in Software Engineering. It focuses on the design and implementation of an advanced platform for automatic extraction of metadata and searching for digital documents, relying on a local language model (LLM).

To implement our solution, we adopted the agile development methodology SCRUM to manage the project, while using UML for modeling. The backend is developed with Django and Django REST Framework, ensuring the processing of various documents (PDF, DOCX, ODT...), text extraction via OCR, metadata enrichment by an LLM, and storage in a PostgreSQL database. The frontend is built with Vue.js, Pinia, and Vuetify, allowing for an interactive user interface.

API tests were conducted using the Insomnia tool, and the code is versioned using Git and hosted on GitHub. The architecture is also based onthe use of Docker containers for better portability.

Keywords : LLM, Django, REST API, Vue.js, OCR, metadata, PostgreSQL, Docker, GitHub, Pinia, SCRUM, UML.