République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderahmane Mira -Béjaïa-
Faculté des Sciences Exactes
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention d'un master professionnel

Option : Génie logiciel

Conception et réalisation d'un outil de détection de plagiat

Réalisé par :
Mlle. Haddad Sara
Mlle. Charikh Zineb

Encadré par :
Dr. Ouyahia Samira

Soutenu le 17 octobre 2021 devant le jury composé de :

Président : Pr. AMROUN Kamal
Examinatrice : Dr. EL-BOUHISSI Houda

Promotion : 2020/2021

Dédicace

“

Je dédie ce travail :

À mes chers ***parents****, pour l'amour qu'ils m'ont donné, leur soutien et leurs encouragements tout au long de mes études. Que Dieu leur accorde santé, bonheur, prospérité et longue vie afin que je puisse un jour les combler de bonheur.*

À mes chères sœurs ***AMINA*** *et* ***IKRAM****,*

À mon cher petit frère ***ISLEM****,*

À mes cousines et cousins,

À mes proches et tous ceux qui m'estiment,

Et à la famille ***HADDAD*** *et* ***BELKACEMI****.*

”

- Sarah

Dédicace

“

Je dédie ce travail :

À mes chers ***parents*** *pour leur amour, leur soutien et leur patience tout au long de mes études, qu'ils trouvent ici l'expression de ma gratitude et de mon profond amour.*

À mes soeurs et mon frère.

À mes cousins et ma famille.

À mes amis proches, où qu'ils soient.

”

- Zina

Remerciements

Nous tenons tout d'abords à remercier ALLAH le tout-puissant d'avoir nous donner le courage, la volonté, la force et la patience de mener a terme ce présent travail.

Nous adressons nos remerciement à notre encadrante Madame OUYAHIA Samira, pour avoir accepté de diriger ce travail. Son soutien, sa clairvoyance ainsi que sa disponibilité nous ont été d'une aide inestimable.

Nous remercions également Monsieur AKILAL Abdellah, pour nous avoir consacré de son temps, afin de nous aider dans la réalisation de notre projet.

Nos vifs remerciement aux membres du jury, Monsieur AMROUN Kamal et Madame EL-BOUHISSI Houda qui nous font l'honneur d'évaluer notre travail ainsi l'enrichir avec leurs propositions et idées.

Nous tenons également à remercier nos familles et amis pour leur soutien permanent qui nous a été bien utile.

Enfin, nous remercions, de tout cœur, tous ceux qui ont contribué de près ou de loin à la réalisation de ce mémoire.

Table des matières

Liste des figures

Liste des tableaux

Liste des abréviations

BDD *Base De Données*

CRUD *Create, Read, Update, Delete*

CSS *Cascading Style Sheets*

HTML *HyperText Markup Language*

NLP *Natural Language Processing*

NLTK *Natural Language Toolkit*

ORM *Object-Relational Mapping*

PDF *Portable Document Format*

PHP *Hypertext Preprocessor*

SGBD *Système De Gestion de Base de Données*

SGBDR *Système De Gestion de Base de Données Relationnelles*

SQL *Structured Query Language*

UML *Unified Modeling Language*

UP *Unified Process*

VS Code *Visual Studio Code*

WEB *World Wide Web*

WSGI *Web Server Gateway Interface*

Introduction générale

De nos jours, les étudiants ont souvent recours au plagiat pour leurs travaux au cours de leur cursus, particulièrement pour leurs mémoires : ces derniers exploitent des travaux déjà réalisés sans citer de source. En effet, le phénomène de triche par plagiat est un problème qui s'est rapidement développé au cours de ces dernières années : au lieu de produire un travail original, certains préfèrent copier directement le contenu trouvé dans des livres, des articles de journaux ou des travaux antérieurs rédigés par d'autres personnes.

La détection de plagiat est donc primordiale dans une université. C'est, d'ailleurs, pour cela que la plupart d'entre elles utilisent un outil de détection de plagiat afin de détecter les copier-coller et les paraphrases, ainsi que tout ce qui relève du plagiat, dans les travaux des étudiants. Ainsi, ces derniers sont acceptés ou refusés selon le taux de plagiat détecté, et dont la limite diffère d'une université à une autre.

C'est dans ce contexte que nous avons pensé à créer une application web de détection de plagiat pour l'Université de Bejaia. Cela permettra de donner plus de crédibilité aux différents travaux émis par ses étudiants. En effet, un travail copié perd de sa valeur ; en outre, on n'y accorde pas beaucoup de crédit. L'application aura donc pour but d'aider les étudiants à prévenir le plagiat, mais aussi de permettre aux universités de les détecter.

Dans ce rapport, quatre principaux chapitres permettent d'englober le travail réalisé :

Le premier chapitre présente le contexte général du projet en décrivant la problématique rencontrée, et la solution proposée pour pallier aux problèmes survenus.

Le deuxième chapitre décrit les besoins fonctionnels et non fonctionnels de notre application en présentant les cas d'utilisation sous différents diagrammes.

Le troisième chapitre a pour but de détailler la phase de conception de notre système.

Le quatrième chapitre évoque tous les outils et langages de développement utilisés pour la réalisation de notre application, puis nous présenterons les interfaces de cette dernière.

Et enfin, nous terminons par une conclusion générale, en évoquant d'éventuelles perspectives.

Chapitre 1

Présentation du projet

1 Introduction

Avec le développement d'Internet et des nouvelles technologies de l'informatique, les étudiants trouvent beaucoup de documentations pour leurs travaux et projets de fin de cycle, mais beaucoup d'entre eux utilisent ces outils pour voler les travaux des autres et les faire passer pour les leurs. Ce problème de vol, nommé "plagiat", peut se présenter aussi au sein de notre université, l'Université de Bejaia ; et de ce fait, le besoin d'une solution se fait sentir.

Dans ce chapitre, nous détaillerons le contexte de notre projet ainsi que la problématique à résoudre pour ensuite présenter la solution retenue qui sera détaillée plus loin dans le rapport. Enfin, nous terminerons le chapitre par la définition du processus de développement entrepris afin de faciliter l'élaboration du projet.

2 Contexte et problématique

L'Université de Bejaia, de par son ancienneté regorge de travaux scientifiques, en tous genres : mémoires rédigés par des étudiants en fin de cycle, projets de recherche, cours, etc. Mais lesdits travaux peuvent être le produit d'un vol, nommé, dans ce cas, le plagiat. La triche par plagiat est un problème qui s'est rapidement développé au cours de ces dernières années, particulièrement avec le développement d'Internet et des nouvelles technologies. Au lieu de produire un travail original, certains étudiants préfèrent copier directement le contenu trouvé dans des livres, encyclopédies, sites internet, ou travaux rédigés par

d'autres étudiants. Un phénomène qui porte préjudice à l'Université vu les dégâts qu'il peut engendrer, d'où l'importance de le combattre.

Cependant, il s'agit d'une lourde tâche étant donné que l'Université de Bejaia ne dispose pas de système informatisé qui s'occupe de le détecter. En effet, tout se passe de façon classique c'est-à-dire manuellement : beaucoup d'enseignants, pour détecter le plagiat, procèdent à une simple recherche dans un moteur tel que Google en utilisant des mots-clés du texte à analyser. Cette méthode s'avère être très coûteuse en termes de temps et de personnel, et n'aboutit pas toujours à des résultats satisfaisants. C'est dans ce contexte que s'inscrit notre projet afin de remédier à ce problème et mettre en place un système qui gère la problématique relevée.

3 Présentation du plagiat

3.1 Qu'est-ce que le plagiat ?

Le dictionnaire Larousse définit le plagiat comme « l'acte de quelqu'un qui, dans le domaine artistique ou littéraire, donne pour sien ce qu'il a pris à l'œuvre d'un autre » ou « ce qui est emprunté, copié, démarqué ».

Le plagiat est un phénomène largement répandu dans la société, en particulier les universités, et c'est le fait de coller une idée, une phrase, un paragraphe ou même tout un chapitre sans citer l'auteur. En d'autres termes, ça consiste à présenter le travail de quelqu'un d'autre comme étant le sien [37].

L'expression « plagiat universitaire », quant à elle, désigne le plagiat étudiant et le plagiat dans la recherche scientifique. Elle permet aussi d'englober dans la réflexion les pratiques des enseignants dans leurs formations [54].

3.2 Comment le détecter ?

Il existe plusieurs outils permettant la détection du plagiat qui sont des logiciels qui analysent un texte, mémoire ou thèse et qui les comparent à leur base de données pour trouver les éventuels passages plagiés.

Les logiciels anti-plagiat permettent aux étudiants de vérifier leurs travaux avant de les déposer ou les publier au sein de leurs universités afin d'éviter les risques de ce phénomène.

3.3 Types de plagiat

Le plagiat prend diverses formes. Il va de la réutilisation d'un document entier à la réécriture d'un seul paragraphe. En fin de compte, tous les types de plagiat se résument à faire passer les idées ou les mots de quelqu'un d'autre pour les nôtres. Les différents types se présentent comme suit [37] :

1. **Copier-coller ou le pagiat direct :**

   Ça consiste à utiliser un texte provenant d'une autre source sans la citer. Si on veut vraiment inclure mot pour mot un passage d'une autre source, on doit apprendre à le citer (utiliser les « » comme citation puis citer la source).

2. **La paraphrase :**

   Alternative à la citation, paraphraser signifie traduire l'idée d'autrui avec nos propres mots. Dans une université, il est préférable de paraphraser plutôt que de faire une citation, ça rend le travail plus original, mais il faut toujours citer la source.

3. **Le plagiat en mosaïque :**

   C'est une autre forme de copier-coller, ça consiste à collecter un ensemble de différents morceaux de texte pour créer une sorte de « mosaïque » des idées d'autres chercheurs et former un paragraphe ou un texte.

   Bien que le résultat soit un morceau de texte complètement nouveau, les mots et les idées ne sont pas nouveaux.

4. **L'auto-plgiat :**

   Lorsqu'on utilise des parties de nos travaux antérieurs (par exemple un article, une analyse documentaire ou un ensemble de données) sans les citer correctement, on commet ce que l'on appelle de l'auto-plagiat.

5. **La traduction :**

   Lorsqu'on traduit un paragraphe, cela ne signifie pas qu'on en est l'auteur.

   Copier-coller le travail de quelqu'un d'autre en langue étrangère et en faire la traduction sans mentionner la source reste du plagiat. Elle est comparable à la paraphrase.

6. **L'achat de document :**

   Ce type de plagiat est explicite. Si on paye quelqu'un pour écrire un mémoire, une thèse ou une dissertation, c'est du plagiat.

3.4 Les causes du plagiat

Bien que rien ne puisse légitimer le plagiat, il est intéressant de connaître les raisons qui mènent un étudiant à plagier. En voici les principales causes [46] :

1. **Gestion de temps :**

   Pendant les périodes où la charge de travail est excessive, le recours à des méthodes illégales peut être perçu par l'étudiant comme la solution qui lui permettra d'obtenir un maximum de points.

2. **Copier/Coller :**

   L'étudiant qui manque d'organisation trouve des informations sur Internet et les copie dans un document, mais omet de noter la source d'où elles proviennent. Par conséquent, lorsqu'il veut les citer, il lui est impossible de les retrouver. En utilisant des informations sans les citer, il plagie.

3. **Savoir-faire et créativité :**

   Ne faisant pas appel à sa créativité et à son savoir-faire pour rédiger son travail, l'étudiant, qui s'estime incapable d'égaler la qualité de rédaction d'une source trouvée sur Internet, décide de plagier.

4. **Estime de soi :**

   L'étudiant utilise le travail d'un collègue qu'il estime, car il obtient de meilleurs résultats académiques que lui, et le fait passer pour sien.

5. **Pertinence :**

   L'étudiant peut considérer que le travail demandé dans un cours n'est pas pertinent et ne contribue pas à son apprentissage. Peu motivé et désirant éviter une perte de temps, il peut être tenté de bâcler ce devoir.

6. **Abondance des sources d'information :**

   Les sources d'information sur Internet sont nombreuses. L'étudiant pense qu'il ne pourra jamais se faire prendre, car il considère impossible que son professeur retrouve l'information plagiée.

4 Les solutions de détection de plagiat

Il est primordial de vaincre le plagiat dans les institutions d'enseignement. En effet, la citation présente des avantages irrévocables. En voici une liste [54] :

— Elle permet d'appuyer un argument en lui donnant de la pertinence et une valeur scientifique ;

— Le lecteur pourra facilement se référer aux sources ;

— Elle accorde une reconnaissance à l'auteur/créateur d'une idée ;

— Elle permet d'illustrer qu'un travail de recherche a été fait [59] ;

— Elle invoque le fait de préserver la valeur des diplômes pour les étudiants.

En vue de cela, plusieurs logiciels sont apparus pour lutter contre ce phénomène qui augmente ces derniers temps. Un tel logiciel peut être installé sur la machine ou utilisé en ligne, il existe des versions gratuites, d'autres payantes. Ces outils présentent généralement les fonctionnalités suivantes :

- Indiquer le pourcentage ou le taux de plagiat en citant les sources ;
- Offrir la possibilité de vérifier le travail avant de pouvoir le remettre en mains propres ;
- Détecter automatiquement le plagiat en comparant le document source avec le document suspect à l'aide d'un corpus de référence ou des sources Internet.

Cependant, ils ne sont pas capables de prendre en charge les différents types de plagiat qui existent, comme la paraphrase : un tel cas est difficile à caractériser. En outre, l'efficacité à traiter le plagiat intelligent lorsque les idées sont présentées dans des formes différentes est souvent remise en cause.

On peut répartir les logiciels de détection de plagiat selon plusieurs critères : le type de fonctionnement, le coût, la langue, le schéma de fonctionnement et les fonctionnalités offertes.

Pour cette étude, on se base sur des détecteurs de plagiat de langue française.

En termes de coût, l'ensemble des logiciels gratuits, ainsi que plusieurs logiciels payants, mais peu chers, sont incapables de détecter de façon fiable les plagiats.

En termes d'efficacité, il s'agit sans doute du critère le plus important, et donc la capacité d'un logiciel à détecter dans un document supposé original les parties qui ont été

« empruntées » à d'autres sources d'une façon fiable : tous les plagiats mais rien que les plagiats. Sur ce point, les logiciels ont encore beaucoup de progrès à faire et la recherche de l'exhaustivité amène souvent à la multitude de sources, et donc à l'augmentation des faux plagiats.

Autre point important, la capacité du logiciel à identifier les zones plagiées dans lesquelles les étudiants ont introduit soit des fautes d'orthographes ou des synonymes, dans le but de rendre plus difficile la détection des plagiats. Sur ce point, peu de logiciels sont capables de contourner cette difficulté.

Dans le tableau 1.1, nous présentons une comparaison des différents logiciels existants :

Logiciel	**Coût**	**Installation**	**Schéma de fonctionnement**
CopyTracker	Gratuit	Installé en interne	— L'utilisateur importe le document à analyser, avec la possibilité d'entrer des mots clés dans des champs dédiés. — Compare différents textes afin d'afficher leur similarité. — Recherche sur Internet des plagiats éventuels.
Quetext	Gratuit	Installé sur serveur externe	Le logiciel compare le documents copié par l'utilisateur aux sources internet, et fournit la liste des sources sans le taux de similarité ou le rapport détaillé. L'outil prend en charge plusieurs langues et est basé sur des algorithmes de traitement du langage naturel et d'apprentissage automatique.

PlagScan	Payant	Installé sur serveur externe	— Il utilise un algorithme offrant multiples fonctionnalités basées sur les dernières recherches en linguistique informatique et détecte ainsi la plupart des types de plagiat. — Prend en charge toutes les langues utilisant le codage UTF-8 et toutes les langues avec des caractères latins ou arabes.
CopyScape	Payant	Installé sur serveur externe	— Service pour la protection des propriétaires de contenu en vérifiant les textes et les contenus des sites par leur URL. — Il dispose d'un ensemble d'algorithmes robustes et utilise l'API Web de Google pour alimenter ses recherches.
Scribbr	Payant	Installé sur serveur externe	L'utilisateur dépose un fichier et lance l'analyse, et à la fin de l'analyse, un rapport détaillé est affiché accompagné du taux de similarité et des sources correspondantes.

Compilatio	Payant	Installé sur serveur externe	— L'utilisateur sélectionne les documents qu'il souhaite analyser, et lance l'analyse. — Le logiciel génère un rapport indiquant : le coefficient de plagiat du document, les passages identifiés comme «copiés sur Internet» et l'ensemble des sources de plagiats possibles. — L'analyse est assez longue.
Turnitin	Payant	Installé sur serveur externe	— Chaque devoir envoyé est retourné sous la forme d'un rapport d'originalité personnalisé. — Les résultats sont basés sur des recherches complètes de milliards de pages d'exemples actuels et archivés de l'Internet, sur les millions de devoirs d'étudiants auparavant envoyés à Turnitin, et sur les bases de données commerciales d'articles de journaux et de périodiques.

Tableau 1.1 – Tableau comparatif des solutions existantes

5 Méthodes de détection de plagiat

Nous allons présenter, dans cette section, quelques méthodes existantes de recherche de similitudes.

5.1 Fingerprinting

C'est une approche dite "n-grammes", c'est-à-dire qu'elle représente le document comme un ensemble de sous-chaînes (n-gram). L'empreinte du document suspect est construite à partir des n-grammes. Les méthodes "fingerprint" divisent la plupart du temps le document en grammes de longueur n, ainsi les empreintes de deux documents peuvent être comparées et les points (grammes) concordants, identifiés comme étant des passages identiques dans les textes. Certaines de ces méthodes vont au-delà de la recherche de similitudes exactes et introduisent la notion de « similarités proches » pouvant ainsi détecter les paraphrases [17].

5.2 String matching

C'est l'approche la plus répandue en informatique. Lorsqu'elle est appliquée au problème de détection de plagiat, les documents sont comparés mot par mot.
On trouve plusieurs algorithmes de « String matching » qui existent aujourd'hui, entre autres, les algorithmes « Brute Force ». Dans le « brute force » on vérifie tous les caractères du texte avec le premier caractère du motif (c'est-à-dire sous-chaîne). Une fois qu'on a une correspondance entre eux, on décale la comparaison entre le deuxième caractère du motif avec le caractère suivant du texte [16].

5.3 Bag of words

Dans cette méthode, les documents sont représentés sous forme d'un vecteur et le calcul de similarité peut alors compter sur la mesure traditionnelle de similarité cosinus. Vector space model est un modèle algébrique pour représenter les documents, en tant que vecteurs d'identifiants par exemple, des termes d'indexation [16].

6 Notre solution pour la détection de plagiat dans notre Université

Dans ce qui précède, nous avons vu les raisons qui poussent les étudiants au plagiat, ainsi que la difficulté de détecter ce dernier en se basant uniquement sur les moyens utilisés, particulièrement au sein de l'Université de Bejaïa. C'est pour cette raison que nous avons décidé de mettre en œuvre un système qui permet de lutter contre ce type de vol. Il s'agit d'une application web permettant la détection du plagiat dans les mémoires de master Informatique. Notre application va répondre aux besoins de nos enseignants qui cherchent un moyen efficace et rapide pour analyser les mémoires de master qu'ils encadrent et apprendre ainsi aux étudiants la manière correcte et éthique de mener un projet de fin de cycle.

Pour la conception de notre application, nous allons utiliser l'UML et le processus UP qui seront présentés dans la suite de ce chapitre.

Quant à la méthode adoptée, nous avons opté pour **Cosine Similarity**, en suivant le processus ci-dessous que nous détaillerons plus loin dans le rapport.

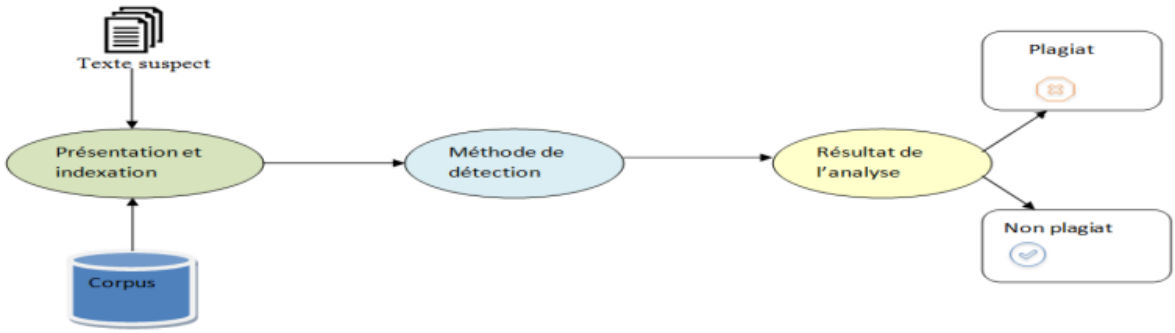

FIGURE 1.1 – Processus de détection de similarité [9]

6.1 Cosine Similarity

La similarité cosinus est une métrique utilisée pour déterminer à quel point les documents sont similaires, quelle que soit leur taille, tel qu'un mot est representé sous forme vectorielle.

Elle varie entre 0 et 1, 1 indique que deux vecteurs sont similaires, tandis que les

valeurs approchant de 0 indiquent qu'il y a moins de similitudes.
Ainsi, en mesurant l'angle entre les vecteurs, nous pouvons avoir une bonne idée de leur similitude, et pour rendre les choses encore plus simples, en prenant le cosinus de cet angle, nous avons un indice de 0 à 1, valeur qui indique cette similarité. Plus l'angle est petit, plus la valeur du cosinus est grande (plus proche de 1), et plus la similarité est grande. [6].
La similarité en cosinus est calculée à l'aide de la formule suivante :

$$\cos(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{AB}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} \mathbf{A}_i \mathbf{B}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{A}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{B}_i)^2}} \tag{1.1}$$

Notre choix s'est porté pour cette méthode en raison des différents avantages qu'elle offre, tel que sa faible complexité et sa rapidité de traitement. En outre, la similarité cosinus est généralement utilisée comme métrique pour mesurer la similarité lorsque la magnitude des vecteurs n'a pas d'importance. Cela se produit par exemple lorsqu'on travaille avec des données textuelles.

En d'autres termes, la méthode suivie est avantageuse car même si deux documents similaires sont éloignés l'un de l'autre en terme du nombre de mots qu'ils contiennent, ils pourraient quand même avoir un angle plus petit entre eux. C'est-à-dire, Cosine similarity se base sur le contenu global plutôt que le nombre de mots.

7 Le processus de développement unifié UP

Un processus définit une séquence d'étapes, partiellement ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant. L'objectif d'un processus de développement est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans des temps et des coûts prévisibles [22].

Il existe plusieurs méthodes de développement logiciel construites sur l'UML : UP, RUP, TTUP, agile UP, XP, 2TUP. Pour ce projet, notre choix s'est porté sur le processus unifié « UP » pour sa simplicité et sa compatibilité avec notre projet.

7.1 Définition du processus UP

Le processus unifié est un processus de développement logiciel « itératif et incrémental, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques » [49] :

- **Itératif et incrémental :** le projet est découpé en itérations de courte durée (environ 1 mois) qui aident à mieux suivre l'avancement global. À la fin de chaque itération, une partie exécutable du système final est produite, de façon incrémentale.

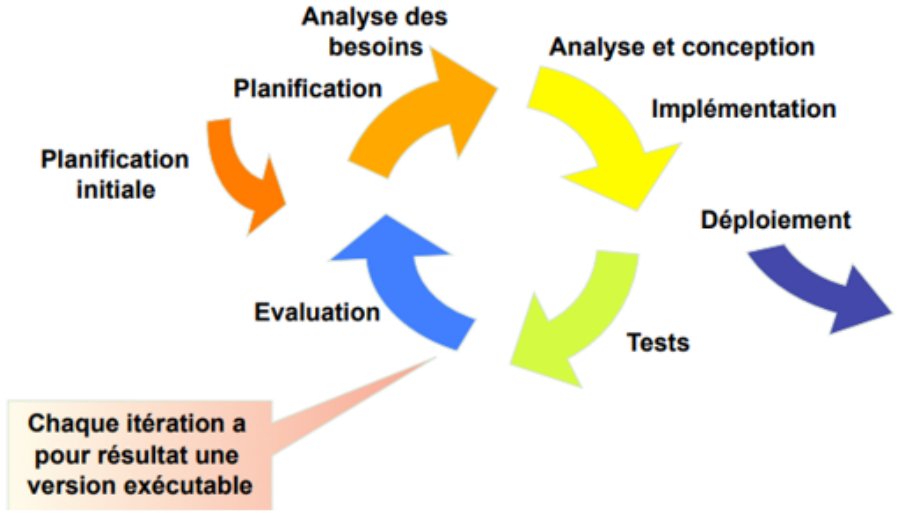

FIGURE 1.2 – Schéma descriptif du cycle itératif et incrémental

- **Centré sur l'architecture :** tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML et pas seulement documentée en texte.
- **Piloté par les risques :** les risques majeurs du projet doivent être identifiés au plus tôt, mais surtout levés le plus rapidement possible. Les mesures à prendre dans ce cadre déterminent l'ordre des itérations.
- **Conduit par les cas d'utilisation :** le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d'utilisation du futur système sont identifiés, décrits avec précision et priorisés.

7.2 Cycle de vie du processus UP

L'objectif du processus unifié est de maîtriser la complexité des projets informatiques en diminuant les risques. UP est un ensemble de principes génériques adapté en fonctions des spécificités des projets.

L'architecture bidirectionnelle : UP gère le processus de développement par deux axes [30] :

— **L'axe vertical :** représente les principaux enchaînements d'activités, qui regroupent les activités selon leur nature. Cette dimension rend compte l'aspect statique du processus qui s'exprime en termes de composants, de processus, d'activités, d'enchaînements, d'artefacts et de travailleurs.

L'enchainement des activités dans le processus UP est présenté en vertical dans la figure 1.3 [1] et expliqué dans ce qui suit [30].

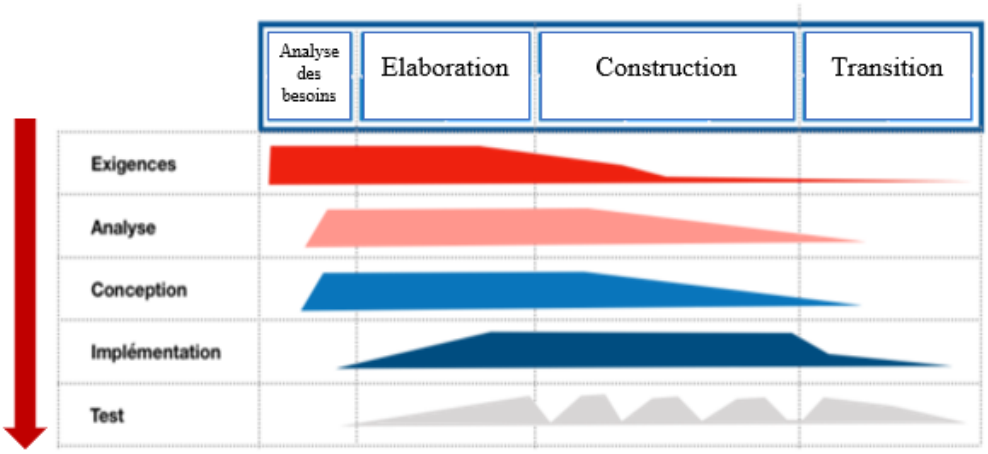

FIGURE 1.3 – Enchaînement d'activités

1. **Expression des besoins :**

   L'expression des besoins comme son nom l'indique, permet de définir les différents besoins :

   - Inventorier les besoins principaux et fournir une liste de leurs fonctions ;
   - Recenser les besoins fonctionnels (du point de vue de l'utilisateur) qui conduisent à l'élaboration des modèles de cas d'utilisation ;
   - Appréhender les besoins non fonctionnels (techniques) et livrer une liste des exigences.

Le modèle de cas d'utilisation présente le système du point de vue de l'utilisateur et représente sous forme de cas d'utilisation et d'acteur, les besoins du client.

2. **Analyse :**

   L'objectif de l'analyse est d'accéder à une compréhension des besoins et des exigences du client. Il s'agit de livrer des spécifications pour permettre de choisir la conception de la solution.

   Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structure sous une forme qui facilite la compréhension (scénarios), la préparation (définition de l'architecture), la modification et la maintenance du futur système. Il s'écrit dans le langage des développeurs et peut être considéré comme une première ébauche du modèle de conception.

3. **Conception :**

   La conception permet d'acquérir une compréhension approfondie des contraintes liées au langage de programmation, à l'utilisation des composants et au système d'exploitation. Elle détermine les principales interfaces et les transcrit à l'aide d'une notation commune.

   Elle constitue un point de départ à l'implémentation :

   - Elle décompose le travail d'implémentation en sous-système ;
   - Elle crée une abstraction transparente de l'implémentation.

4. **Implémentation :**

   L'implémentation est le résultat de la conception pour implémenter le système sous formes de composants, c'est-à-dire, de code source, de scripts, de binaires, d'exécutable et d'autres éléments du même type. Les objectifs principaux de l'implémentation sont de planifier les intégrations des composants pour chaque itération, et de produire les classes et les sous-systèmes sous formes de codes sources.

5. **Test :**

   Les tests permettent de vérifier les résultats de l'implémentation en testant la construction. Pour mener à bien ces tests, il faut les planifier pour chaque

itération, les implémenter en créant des cas de tests, effectuer ces tests et prendre en compte le résultat de chacun.

— **L'axe horizontal :** représente le temps et montre le déroulement du cycle de vie du processus ; cette dimension rend compte de l'aspect dynamique du processus qui s'exprime en termes de cycles, de phases, d'itérations et de jalons [1].

L'enchainement de phases est présenté en horizontal sur la figure 1.4[1] et leurs déroulements sont expliqués comme suit [30] :

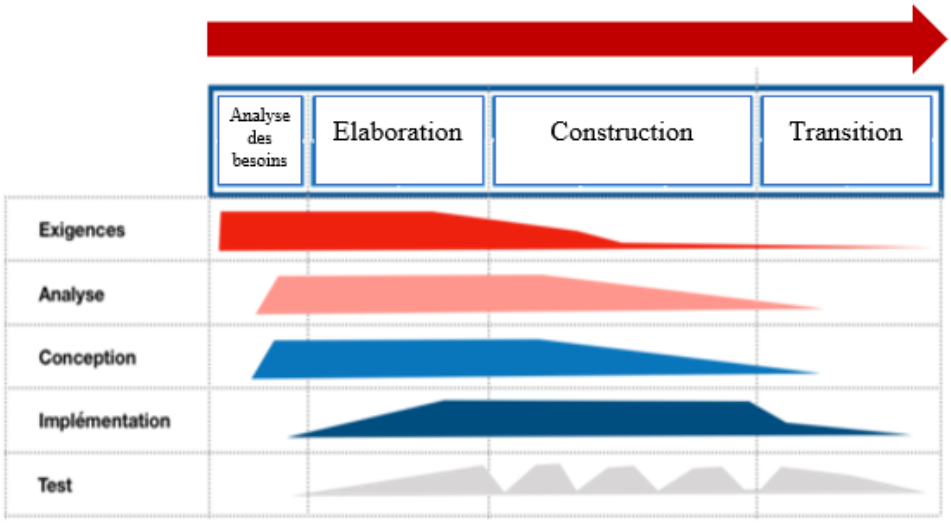

FIGURE 1.4 – Enchainement de phase.

1. **Analyse des besoins :**

   L'analyse des besoins donne une vue du projet sous forme de produit fini. Cette phase porte essentiellement sur les besoins principaux (du point de vue de l'utilisateur), l'architecture générale du système, les risques majeurs, les délais et les coûts.

2. **Elaboration :**

   L'élaboration reprend les éléments de la phase d'analyse des besoins et les précise pour arriver à une spécification détaillée de la solution à mettre en œuvre. L'élaboration permet de préciser la plupart des cas d'utilisation, de

1. Etapes d'évaluation de la phase terminée, et de lancement de la phase suivante.

concevoir l'architecture du système et surtout de déterminer l'architecture de référence.

3. **Construction :**

   La construction est le moment où l'on construit le produit. L'architecture de référence se métamorphose en produit complet. Le produit contient tous les cas d'utilisation que les chefs de projet, en accord avec les utilisateurs ont décidé de mettre au point pour cette version.

4. **Transition :**

   Le produit est en version bêta. Un groupe d'utilisateurs essaye le produit et détecte les anomalies et défauts.La phase de transition permet de faire passer le système informatique des mains des développeurs à celles des utilisateurs finaux.

8 Le langage de modélisation UML

Afin d'optimiser la compréhension ainsi que la réalisation du projet, on a choisi comme langage de modélisation l'UML (Unified Modeling Langage).

UML est un langage de modélisation graphique destiné à visualiser, analyser, spécifier, construire des logiciels orientés objets. UML est aujourd'hui considéré comme un standard autant dans le milieu industriel qu'académique. Il propose un ensemble de diagrammes afin de couvrir l'ensemble des besoins de modélisation potentiellement nécessaires à la conception des logiciels, ce qui le rend relativement complet et générique.

Il permet de modéliser les aspects statiques et dynamiques des systèmes complexes et de couvrir la plupart des phases du développement logiciel (analyse, conception, implantation, déploiement, etc.) [24].

9 Conclusion

Après avoir présenté la problématique et étudié les différents logiciels existants, nous avons une idée globale sur le futur système à concevoir et nous allons commencer dans le chapitre suivant avec la première étape de développement de l'application qui est la sépcification et l'analyse des besoin.

Chapitre 2

Spécification et analyse des besoins

1 Introduction

Après avoir présenté le contexte et la problématique du projet et défini la démarche adoptée, nous allons procéder à l'élaboration de notre projet dans ce chapitre en commençant par la première phase qui est l'expression des besoins, suivie de la phase d'analyse.

2 Cahier des charges

Avant de commencer un projet, un cahier des charges doit être rédigé pour décrire précisément les besoins de la maîtrise d'ouvrage et lister les fonctionnalités du projet.

2.1 Présentation du projet

2.1.1 Contexte

La plupart des universités utilisent des outils spécifiques afin de détecter le plagiat dans les mémoires, c'est dans ce contexte que nous proposons, pour notre université, cette application pour accompagner les étudiants et les enseignants.

2.1.2 Mission

- Sécurité : Le travail de chaque utilisateur devra rester confidentiel.

- Rapidité du traitement : Il est nécessaire que la durée d'exécution des traitements s'approche le plus possible du temps réel.

2.1.3 Objectifs

L'objectif de notre étude est de créer un outil qui sera capable de :

- Analyser les mémoires des étudiants en fin de cycle ;
- Indiquer le taux de similarité et les sources ;
- Comparer deux fichiers et avoir un rapport détaillé.

2.2 Besoins fonctionnels et non fonctionnels :

2.2.1 Besoins fonctionnels

Les besoins fonctionnels représentent les principales fonctionnalités du système [8].

Besoins	**Fonctionnalités**
Espace d'inscription	Créer une interface avec : • Des champs de saisie (formulaire) ; • Un bouton pour valider.
Espace de connexion	Créer une interface qui aura : • Deux champs de saisie (login et mot de passe) ; • Un bouton : Connexion.
Deux types d'utilisateurs	Un compte pour chacun : • Administrateur ; • Utilisateur avancé.

Espace administrateur	Créer le dashboard administateur. Créer une interface pour la gestion des fichiers pour : • L'ajouter des fichiers ; • L'affichage de la liste des fichiers. Créer une interface pour la gestion des utilisateurs pour : • L'ajout des utilisateurs ; • Suppression des utilisateurs.
Espace utilisateur	Créer l'interface d'accueil qui aura un ensemble de choix : • Effectuer une comparaison de deux fichiers ; • Analyser un fichier. Créer l'interface post-analyse qui aura : • Un rapport d'analyse détaillé ; • Un champ pour afficher le taux de similarité ; • La liste des sources.

Tableau 2.1 – Besoins fonctionnels

2.2.2 Besoins non fonctionnels

Les besoins non fonctionnels sont des indicateurs de qualité de l'exécution des besoins fonctionnels [8].

Ergonomie : l'application devra viser la facilité de l'utilisation.

Graphisme :

- Les couleurs principales sont le bleu et le blanc.
- Le choix de la police : "Open Sans", sans-serif.

2.3 Diagramme de contexte

Un diagramme de contexte communique une vue d'ensemble du flux de données d'un système [32]. La figure 2.1 représente le diagramme de contexte de notre système.

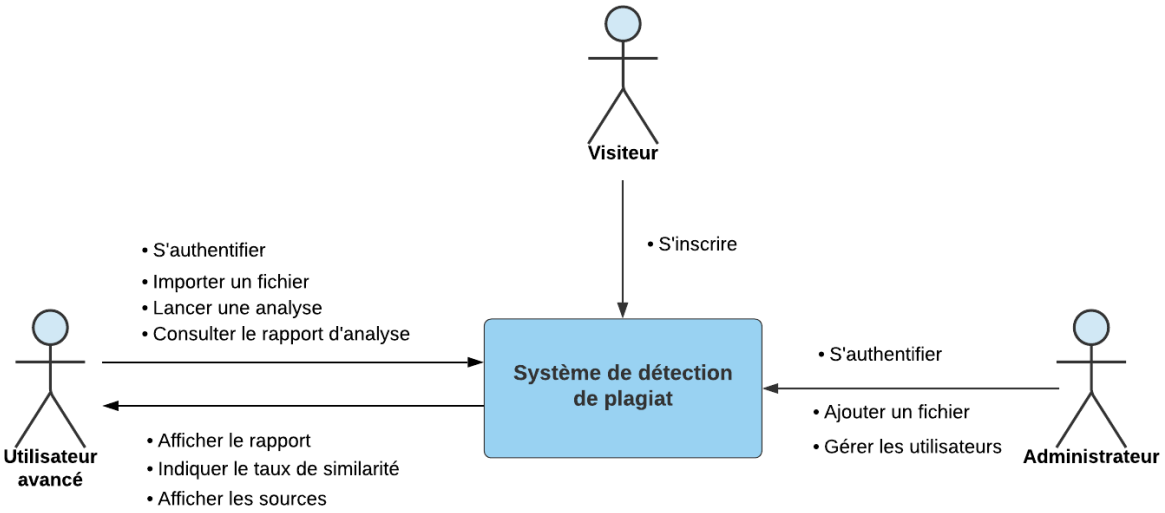

FIGURE 2.1 – Diagramme de contexte

3 Capture des besoins fonctionnels

Cette phase consiste à comprendre le contexte du système. Il s'agit de déterminer les fonctionnalités et les acteurs les plus pertinents, de préciser les risques et d'identifier les cas d'utilisation initiaux [35].

3.1 Identification des acteurs

Un acteur est une personne morale ou physique, interne ou externe intervenant dans le système d'information. Un acteur représente l'abstraction d'un rôle joué par des entités externes qui interagissent directement avec le système étudié [45], [58]. Dans notre cas, il y a quatre acteurs qui peuvent interagir avec notre système, et qui sont :

- **Visiteur** : Utilise l'application pour visiter le site, contacter et demander l'inscription pour devenir utilisateur avancé.
- **Utilisateur** : Tout internaute ayant un compte sur l'application web.

- **Utilisateur avancé** : L'ensemble des étudiants et des enseignants de l'Université de Béjaia inscrits au site web.
- **Administrateur** : C'est l'entité chargée de manipuler la base de données et de gérer les utilisateurs.

3.2 Identification des cas d'utilisation

L'ensemble des cas d'utilisation décrit toutes les exigences fonctionnelles du système et donc fait partie des spécifications fonctionnelles. Chaque cas correspond à une fonction métier du système du point de vue de ses utilisateurs [34]. Dans le système à développer, nous avons identifié les cas d'utilisation cités dans le tableau suivant :

Cas d'utilisations	**Acteur**
Inscription	Visiteur
Traitement des fichiers Analyser un fichier Comparer deux fichiers Importer un fichier Consulter le rapport détaillé Consulter le taux de similarité Consulter les sources	Utilisateur avancé
Ajouter des fichiers à la base de données Gérer les utilisateurs	Administrateur
Authentification	Utilisateur

Tableau 2.2 – Identification des cas d'utilisation

3.3 Diagramme de cas d'utilisation

Le diagramme de la figure 2.2 représente les différentes façons dont un utilisateur peut interagir avec le système.

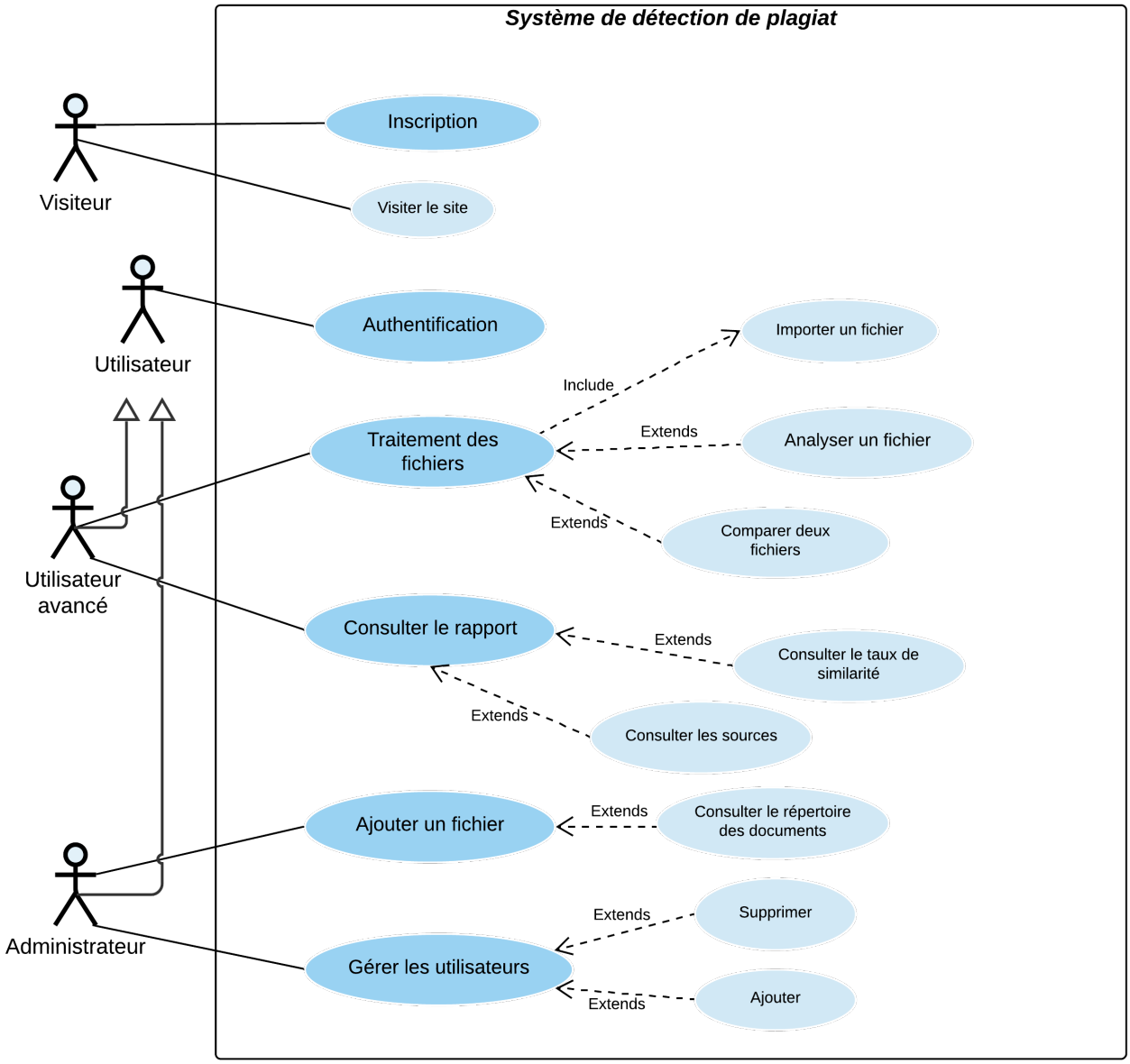

FIGURE 2.2 – Diagramme de cas d'utilisation

4 Description textuelle des cas d'utilisation

La description textuelle d'un cas d'utilisation décrit les objectifs de ce dernier et les interactions entre le système et ses acteurs [10].

Les tableaux 2.3, 2.4, 2.5, 2.6, 2.7, 2.8 représentent respectivement la description textuelle des cas d'utilisation "inscription", "authentification", "traitement d'un fichier", "consulter le rapport", "ajouter un fichier" et "gérer les utilisateurs".

4.1 Inscription

Cas d'utilisation N 1	**Inscription**
Acteur	Visiteur
Objectif	Pouvoir s'authentifier sur le site
Précondition	Avoir une connexion
Scénario nominal	- L'utilisateur atteint la page d'inscription et remplit le formulaire. - Le système vérifie la validité des informations, et ajoute l'utilisateur à la BDD. - Le système affiche ensuite l'espace de connexion
Scénario alternatif	Les informations saisies sont incorrectes, le système lui envoie un message d'erreur.

Tableau 2.3 – Description du cas d'utilisation « Inscription ».

4.2 Authentification

Cas d'utilisation N 2	**Authentification**
Acteur	Utilisateur
Objectif	Vérifier l'identité
Précondition	Avoir une connexion + être inscrit
Scénario nominal	- L'utilisateur atteint la page de connexion. - L'utilisateur saisit le nom et le mot de passe. - Le système vérifie la validité des informations saisies. - Les informations sont envoyées à la BDD. - Une recherche est effectuée sur la BDD pour vérifier l'existence des infos saisies - Le système affiche ensuite l'espace correspondant selon l'acteur.
Scénario alternatif	Le nom ou le MDP sont incorrects, le système envoie un message d'erreur.

Tableau 2.4 – Description du cas id'utilisation « Authentification ».

4.3 Traitement des fichiers

Cas d'utilisation N 3	**Traitement des fichiers**
Acteur	Utilisateur avancé
Objectif	Détecter le plagiat et obtenir le rapport détailé
Précondition	Authentification
Scénario nominal	- L'utilisateur veut faire analyser un fichier. - L'utilisateur importe le fichier. - Une recherche est effectuée dans la BDD. - Un résultat s'affiche à l'utilisateur. - L'utilisateur peut aussi comparer deux fichiers.
Scénario alternatif	Pas de connexion => Pas d'accès à l'application.

Tableau 2.5 – Description du cas d'utilisation « Analyser un fichier ».

4.4 Consulter le rapport

Cas d'utilisation N 4	**Consulter le rapport**
Acteur	Utilisateur avancé
Objectif	L'accès au résultat obtenu après l'analyse
Précondition	Authentification
Scénario nominal	L'utilisateur accède à un rapport détaillé de l'analyse.
Scénario alternatif	Pas de connexion => Pas d'accès au rapport.

Tableau 2.6 – Description du cas d'utilisation « Consulter le rapport ».

4.5 Ajouter un fichier

Cas d'utilisation N 5	**Ajouter un fichier**
Acteur	Administrateur
Objectif	Mettre à jour la BDD.
Précondition	Authentifcation
Scénario nominal	- L'admin importe un fichier à ajouter au répertoire. - Valider avec un bouton.
Scénario alternatif	Le fichier existe déjà => Message d'erreur.

Tableau 2.7 – Description du cas d'utilisation « Ajouter un fichier ».

4.6 Gérer les utilisateurs

Cas d'utilisation N 6	**Gérer les utilisateurs**
Acteur	Administrateur
Objectif	Gestion des utilisateurs
Précondition	Authentification
Scénario nominal	L'admin a le choix entre : - Ajouter un utilisateur à la BDD. - Supprimer un utilisateur. - Consulter la liste des utilisateurs.
Scénario alternatif	Néant.

TABLEAU 2.8 – Description du cas d'utilisation « Gérer les utilisateurs ».

5 Diagrammes de séquences système

Le diagramme de séquence représente les règles d'enchaînement des activités et actions dans le système. Il permet d'une part de consolider la spécification d'un cas d'utilisation, d'autre part de concevoir une méthode [50].

Nous utilisons le terme de diagramme de séquence « système » pour souligner le fait que nous considérons le système informatique comme une boîte noire. Le comportement du système est décrit vu de l'extérieur, sans préjuger de comment il le réalisera. Nous ouvrirons la boîte noire seulement en conception [48].

Les figures 2.3, 2.4, 2.5, 2.6, 2.7, 2.8 représentent respectivement les diagrammes de séquences système des cas d'utilisation ; "inscription", "authentification", "traitement d'un fichier", "consulter le rapport", "ajouter un fichier", et "gérer les utilisateurs".

5.1 Inscription

Ça permet au visiteur de pouvoir s'authentifier et d'avoir ainsi recours aux fonctionnalités de l'application.

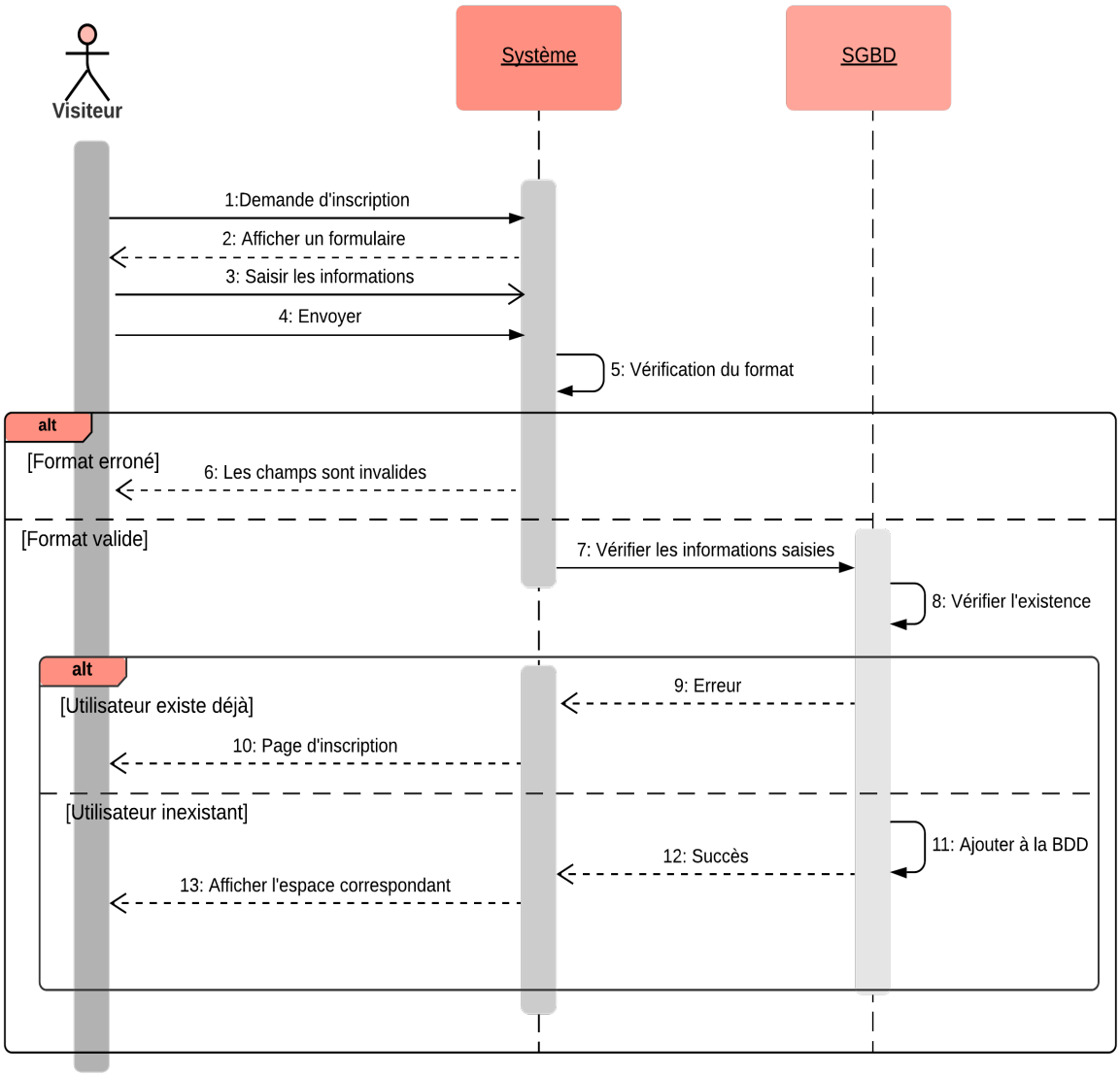

FIGURE 2.3 – Diagramme de séquence du cas d'utilisation "Inscription"

5.2 Authentification

Ça permet à l'utilisateur de prouver son identité afin d'avoir recours aux différents privilèges.

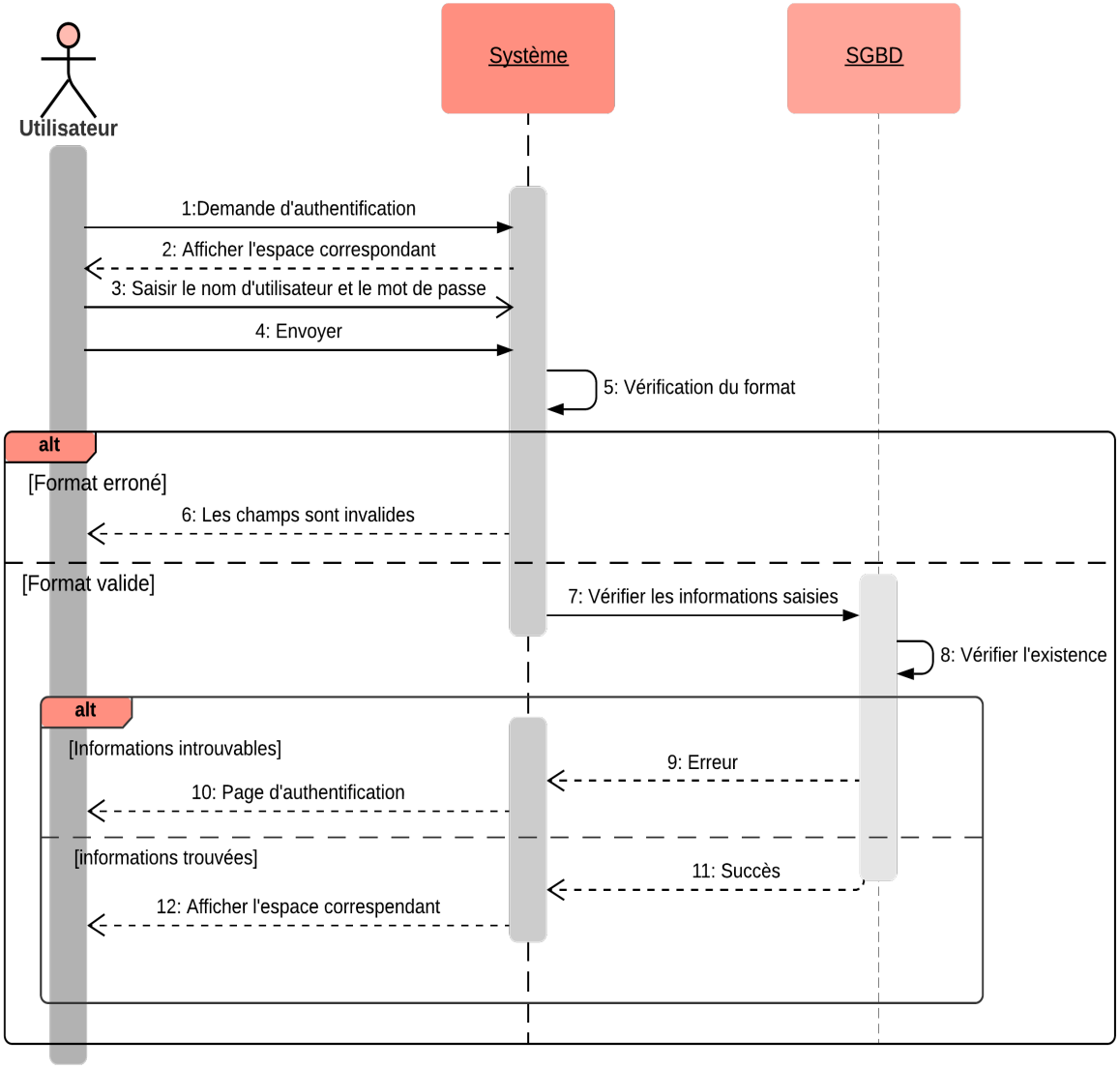

FIGURE 2.4 – Diagramme de séquence du cas d'utilisation "Authentification"

5.3 Traitement d'un fichier

C'est la partie la plus importante de notre projet : l'utilisateur importe le fichier à analyser, le système se charge ensuite de le comparer aux différents fichiers déjà existants afin de détecter d'éventuelles similarités.

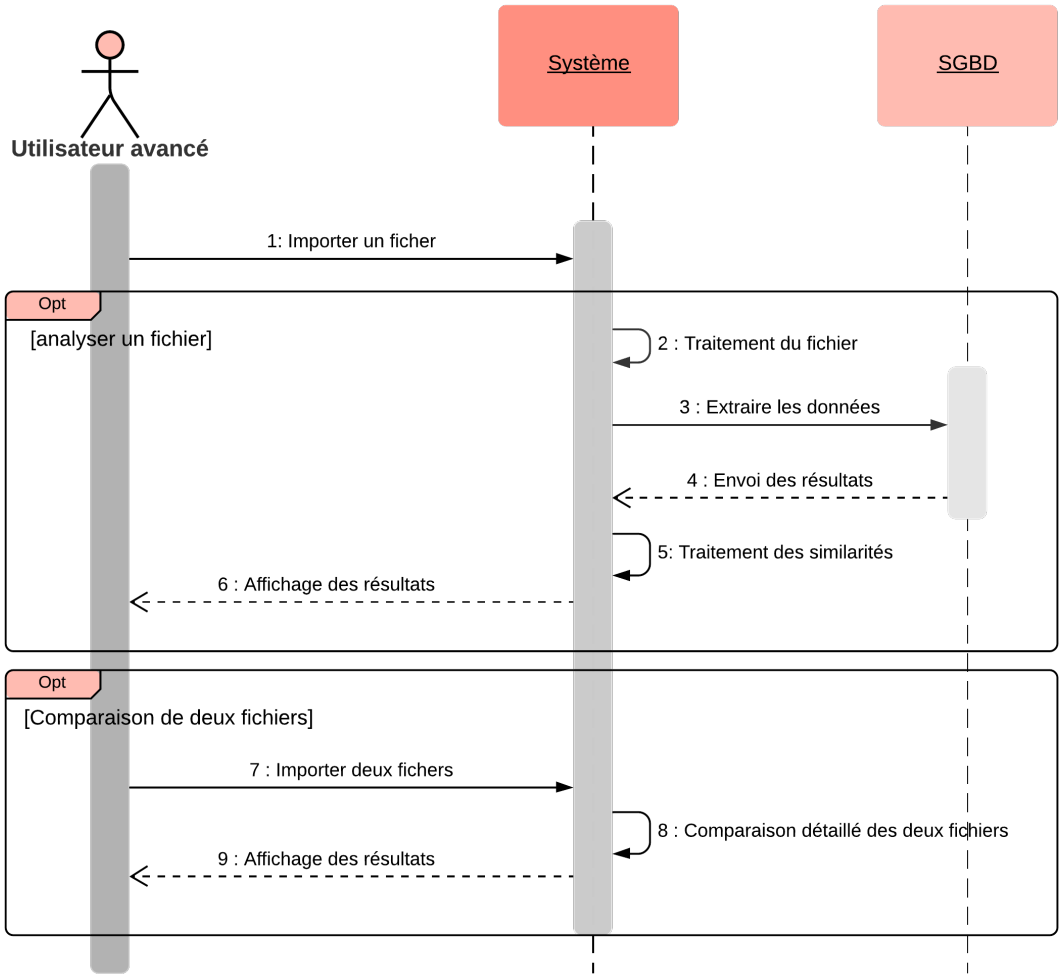

FIGURE 2.5 – Diagramme de séquence du cas d'utilisation "Analyser un fichier"

5.4 Consulter le rapport

L'utilisateur, après avoir validé son choix d'analyse, a accès à un rapport récapitulant cette dernière. Le rapport comporte le pourcentage de similarité ainsi que les passages plagiés et leurs sources.

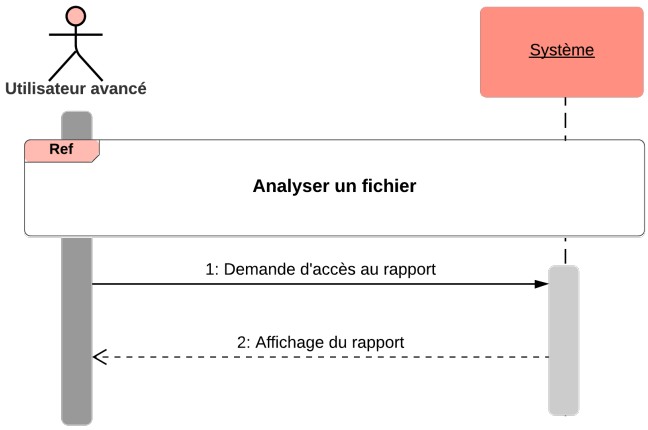

FIGURE 2.6 – Diagramme de séquence du cas d'utilisation "Consulter le rapport"

5.5 Ajouter un fichier

L'administrateur peut ainsi ajouter des fichiers à la base de données, après s'être authentifié.

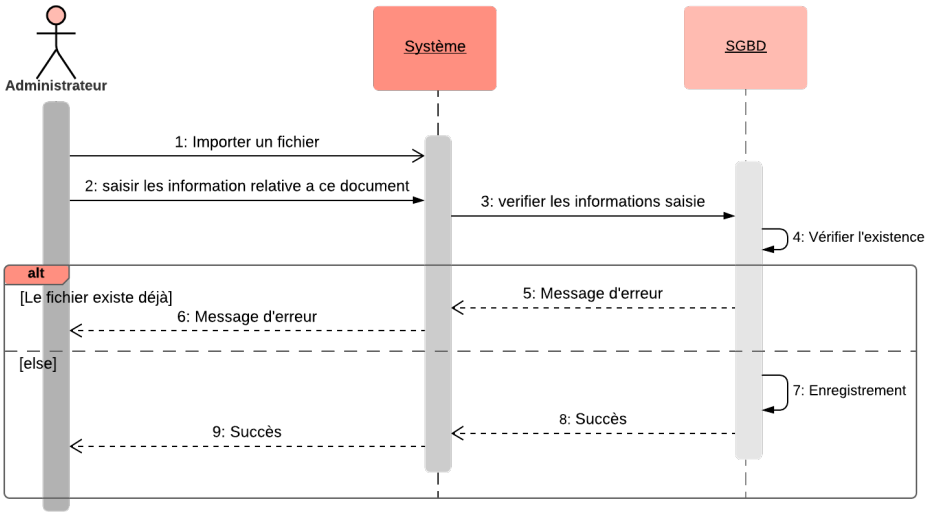

FIGURE 2.7 – Diagramme de séquence du cas d'utilisation "Ajouter un fichier"

5.6 Gérer les utilisateurs

Ça permet à l'administrateur d'ajouter ou de supprimer des utilisateurs.

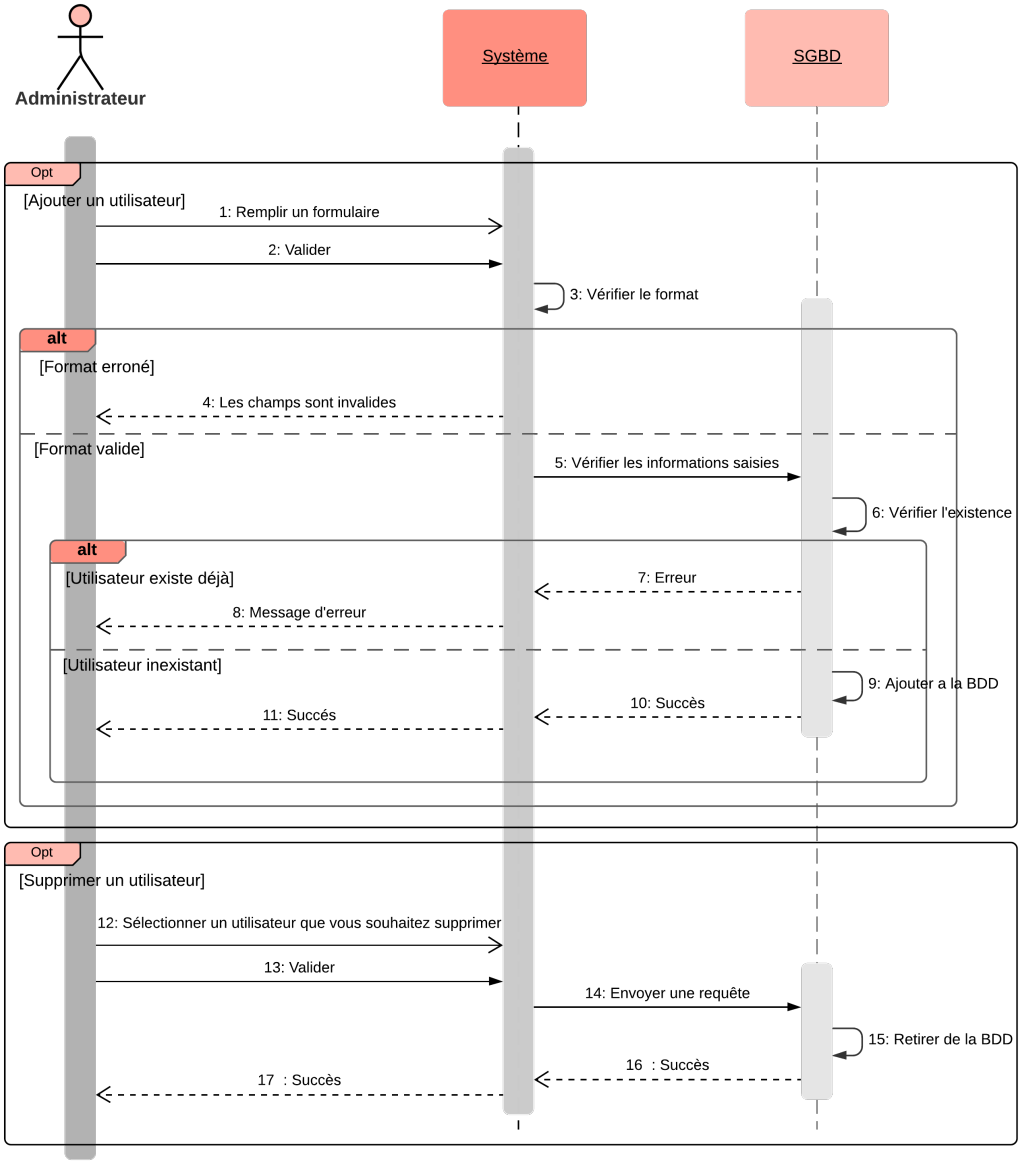

FIGURE 2.8 – Diagramme de séquence du cas d'utilisation "Gérer les utilisateurs"

6 Conclusion

Ce chapitre nous a permis d'exprimer et d'analyser les besoins permettant de décrire les fonctionnalités du système de manière globale. Cela nous permettra, dans le chapitre suivant, de réaliser les différents modèles de la phase de conception.

Chapitre 3

Conception

1 Introduction

La phase de conception est une étape importante dans le cycle de développement logiciel. Elle permet de structurer, organiser et planifier le projet. Dans ce chapitre, nous allons présenter en détails la conception du projet à travers les diagrammes d'interaction, le diagramme de classes ainsi que le modèle relationnel.

2 Diagrammes d'interactions

L'objectif du diagramme d'interaction est de représenter les interactions entre objets en indiquant la chronologie des échanges [21].

Pour chaque diagramme de séquence système défini dans le chapitre 2, nous établirons un diagramme d'interaction, en remplaçant le système vu comme une boîte noire par un ensemble d'objets de classes différentes [12].

Les figures 3.1, 3.2, 3.3, 3.4 et 3.5 représentent respectivement les diagrammes d'interaction des cas d'utilisation "inscription", "authentification", "traitement d'un fichier", "ajouter un fichier" et "gérer les utilisateurs".

2.1 Inscription

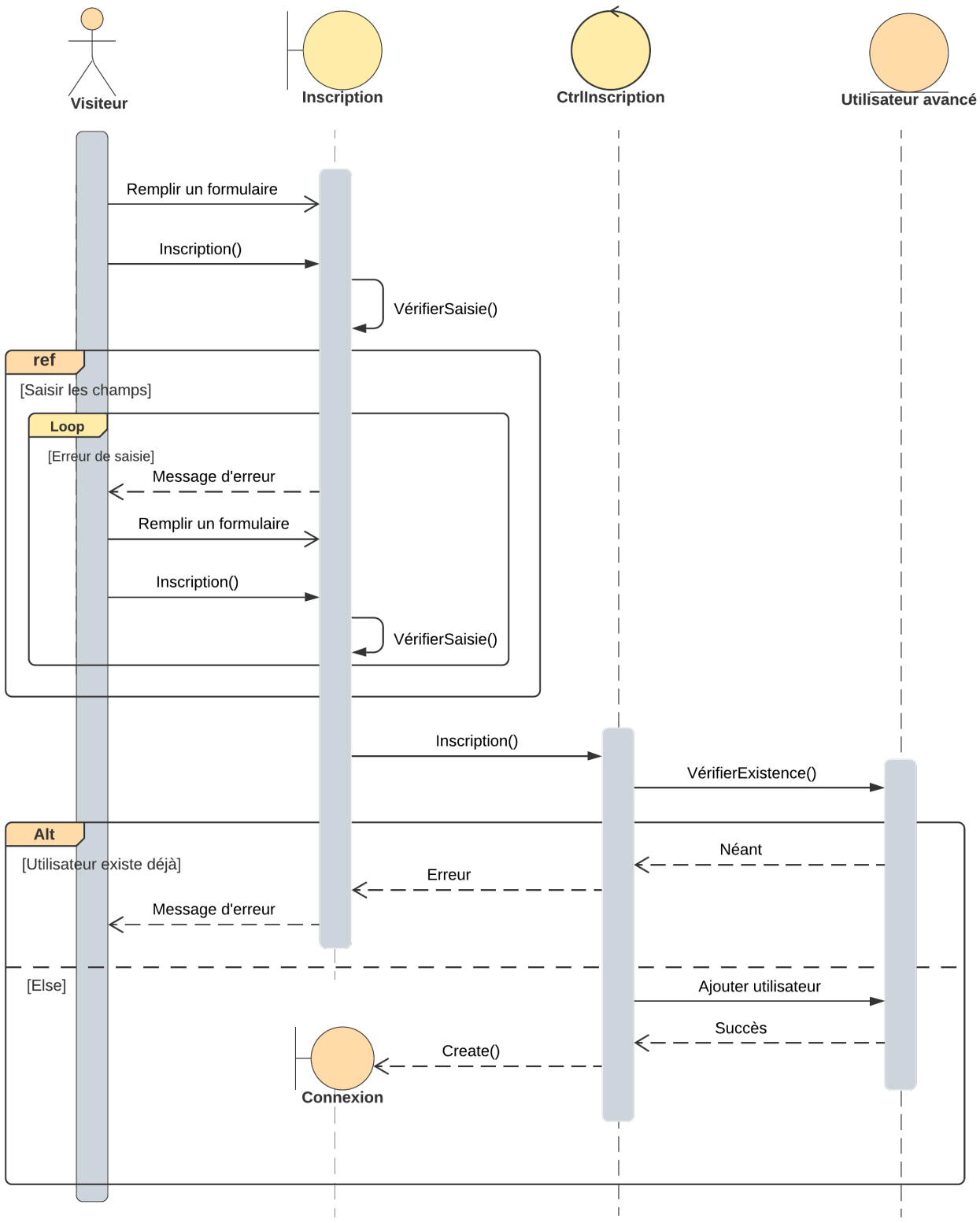

FIGURE 3.1 – Diagramme d'interaction "Inscription"

2.2 Authentification

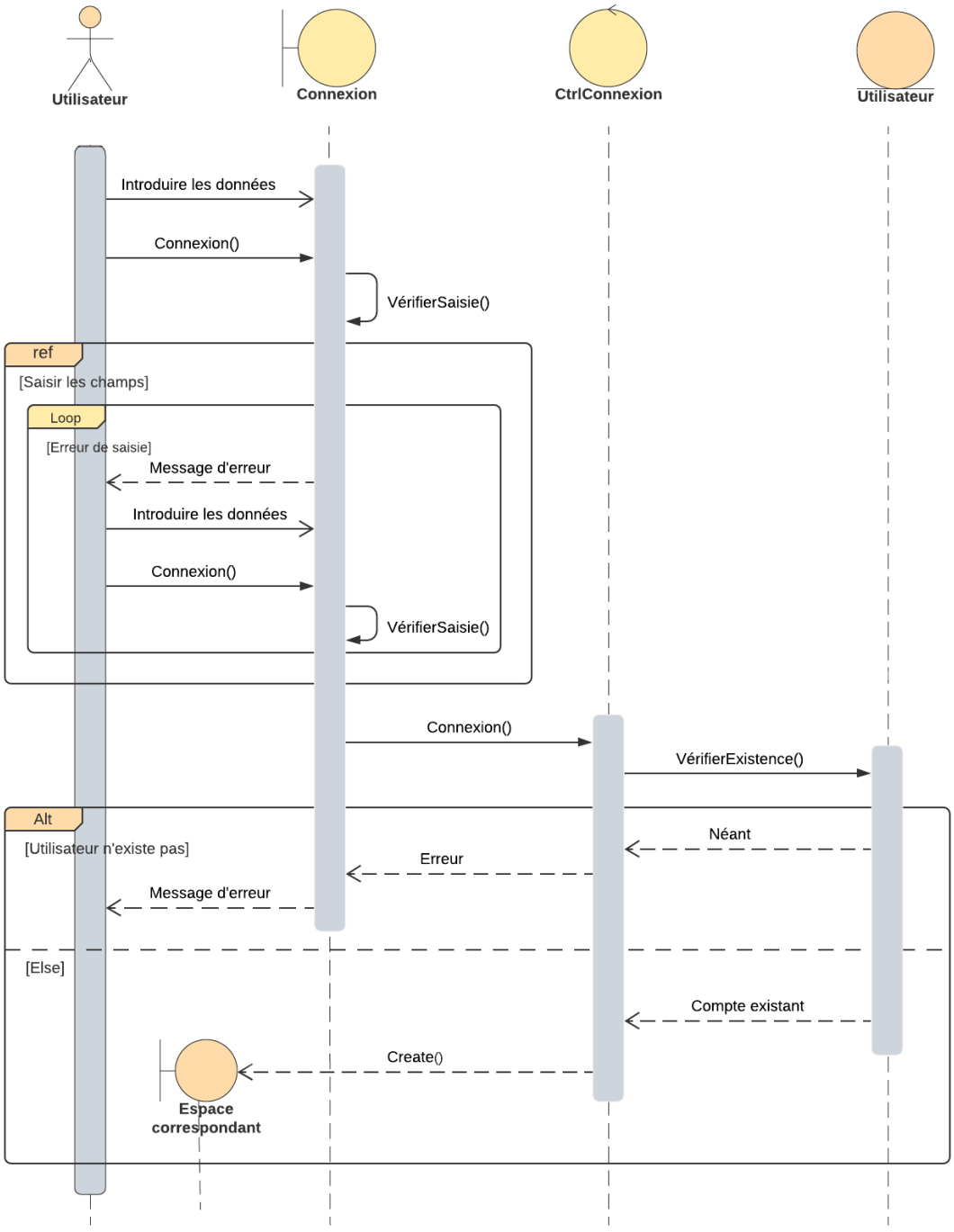

FIGURE 3.2 – Diagramme d'interaction "Authentification"

2.3 Traitement d'un fichier

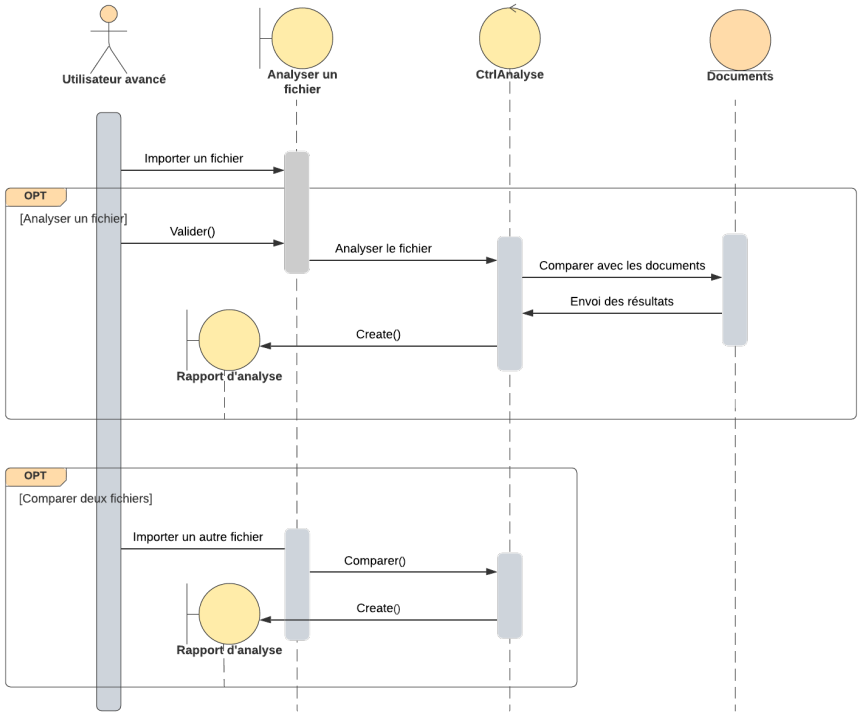

FIGURE 3.3 – Diagramme d'interaction "Analyser un fichier"

2.4 Ajouter un fichier

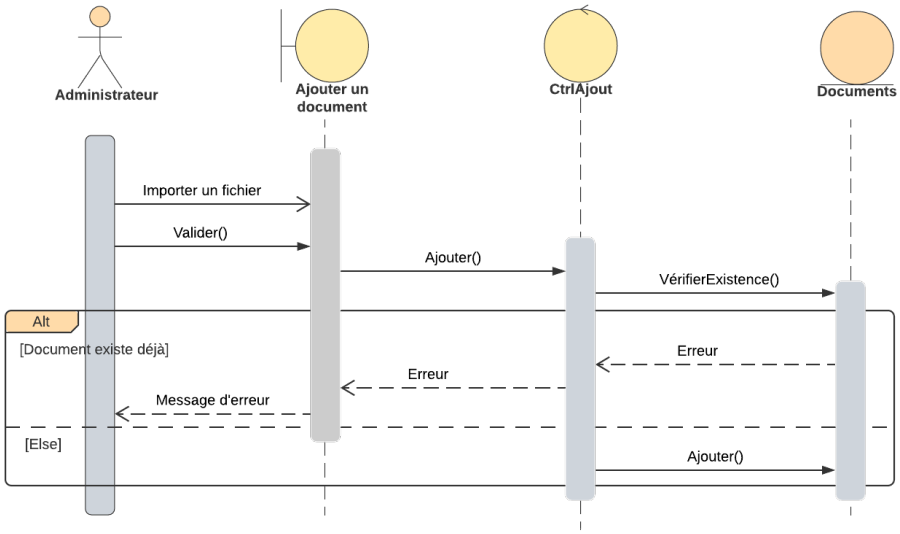

FIGURE 3.4 – Diagramme d'interaction "Ajouter un fichier"

2.5 Gérer les utilisateurs

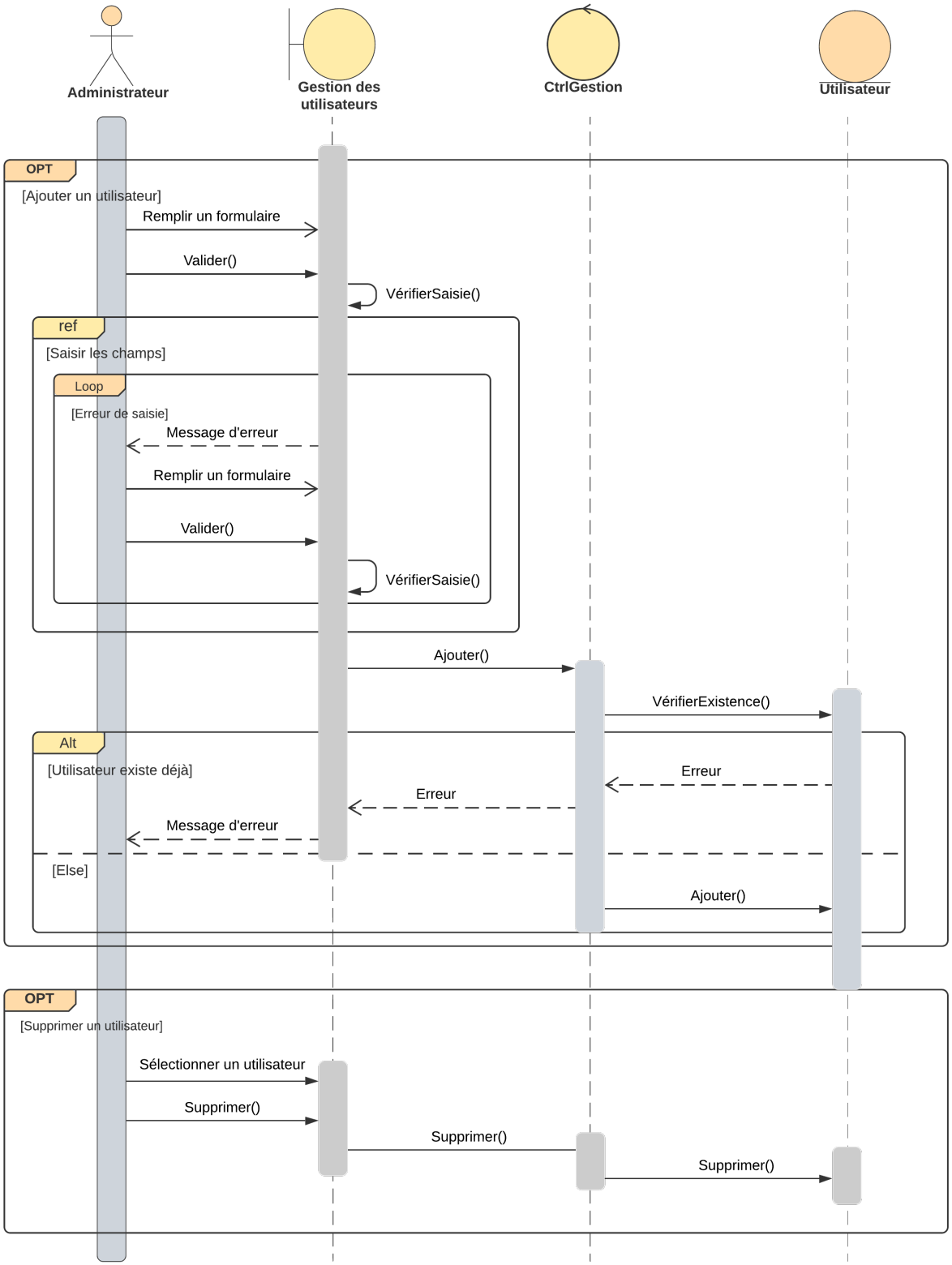

FIGURE 3.5 – Diagramme d'interaction "Gérer les utilisateurs"

3 Dictionnaire de données

Dans le tableau 3.1 sont décrites et expliquées toutes les données qui sont relatives aux classes de notre système.

classe	Résposabilité	Attributs		
		Nom	**Définition**	**Type**
Utilisateur	L'utilisateur	Id	Identifiant d'un utilisateur	Entier
		Username	Le nom d'utilisateur	Chaîne de caractères
		Email	L'adresse couriel	Chaîne de caractères
		Password	Le mot de passe d'un utilisateur	Chaîne de caractères
		Date	Date d'inscription	Datetime
User	Utilisateur avancé			
Admin	Administrateur			
Document	Document ajouté par l'admin	Id	Identifiant	Entier
		Nom	Nom du document	Chaîne de caractères
		Url	Chemin du document	Chaîne de caractères
		Domain	Le domaine du document	Chaîne de caractères
		Filière	La filière du document	Chaîne de caractères
		Option	Option du document	Chaîne de caractères
		Topic	Le titre du document (le thème)	Chaîne de caractères
OtherDoc	Document importé par l'utilisateur			
Rapport	Rapport obtenu après l'analyse	filename	Le nom du fichier	Chaîne de caractère
		Dateh	La date et l'heur d'analyse du fichier	Datetime
		Similarité	Le taux de similarité obtenu	Chaîne de caractères
Engin	Notre système	Version	La version de l'application	Chaîne de caractères
		Lien	Le lien de l'application	Chaîne de caractères

Tableau 3.1 – Dictionnaire de données.

4 Diagramme de classes

Les diagrammes de classes décrivent la structure ou plutôt l'architecture d'un système et sont donc la base de presque toutes les autres techniques de description. En conséquence, les diagrammes de classes, et en particulier les classes, sont un concept qui est utilisé universellement en modélisation et en programmation. Ils permettent l'encapsulation des attributs et des méthodes et la représentation des instances sous forme d'objets [42].
La figure 3.6 représente notre diagramme de classes.

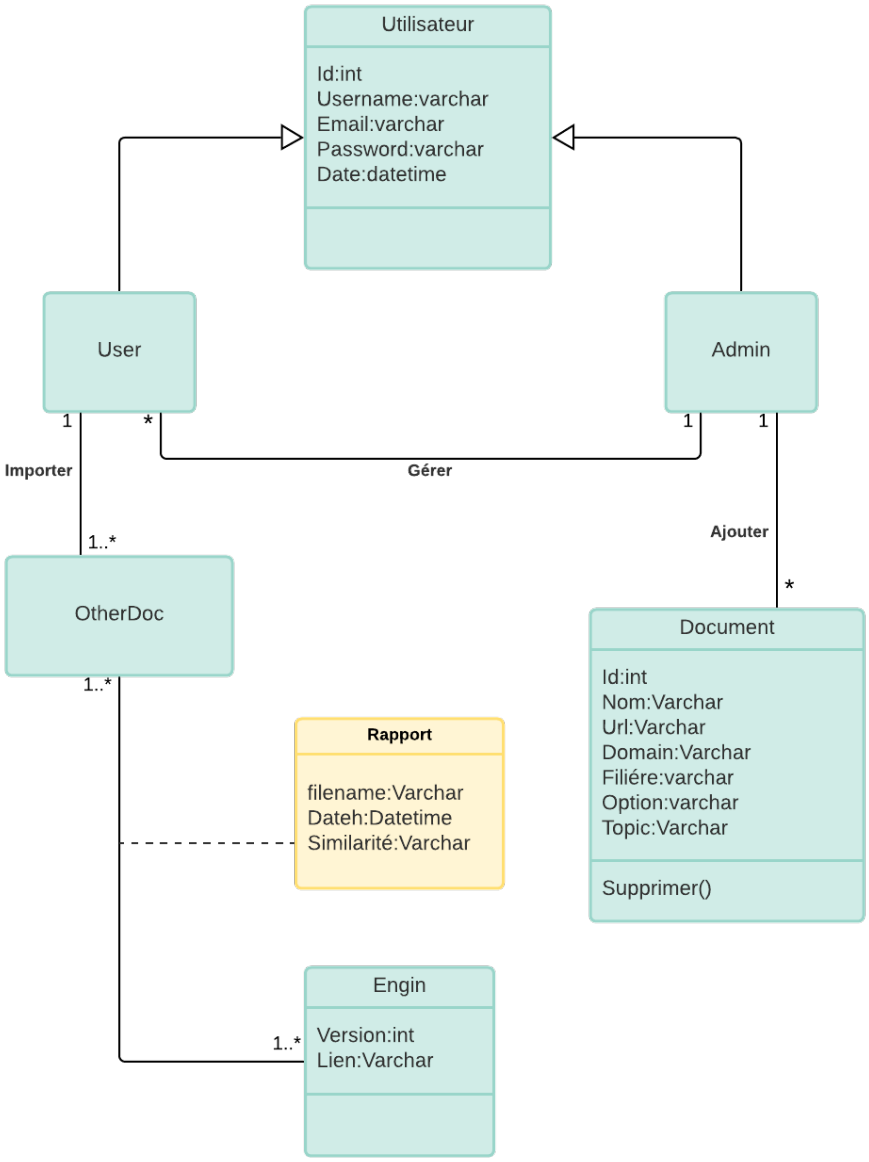

FIGURE 3.6 – Diagramme de classe

5 Schéma relationnel

A partir du diagramme de classes, nous allons réaliser le modèle relationnel qui est le modèle logique de données, ce modèle décrit de façon abstraite comment sont représentées les données dans une base de données.

5.1 Règles de passage vers le modèle relationnel

Les règles utilisées pour le passage du modèle du domaine vers le modèle relationnel sont : [55]

— **Règle 1 (Transformation des classes) :** Chaque classe du diagramme de classe devient une relation, il faut choisir un attribut de la classe pouvant jouer le rôle de clé (le rôle de l'identifiant).

Transformation des associations : Nous distinguons trois familles d'associations :

- **Règle 2 (Association un-à-plusieurs) :** Il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.
- **Règle 3 (Association plusieurs-à-plusieurs) :** La classe-association devient une relation. La clé primaire de cette relation est la concaténation des identifiants des classes connectées à l'association, chaque attribut devient clé étrangère si la classe connectée dont il provient devient une relation en vertu de la règle 1. Les attributs de l'association (classe-association) doivent être ajoutés à la nouvelle relation. Ces attributs ne sont ni clé primaire, ni clé étrangère.
- **Règle 4 (Association un-à-un) :** Il faut ajouter un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de la classe connectée à l'association. Si les deux multiplicités minimales sont à un, il est préférable de fusionner les deux classes en une seule
- **Règle 5 (Transformation de l'héritage) :** Trois décompositions sont possibles pour traduire une association d'héritage en fonction des contraintes existantes :
  - **Décomposition par distinction :** Il faut transformer chaque sous-classe en une relation, la clé primaire de la surclasse, migre dans la relation issue de la sous-classe(s) et devient à la fois clé primaire et clé étrangère.

- **Décomposition descendante :** S'il existe une contrainte de totalité ou de partition sur l'association d'héritage, il est possible de ne pas traduire la relation issue de la surclasse. Il faut alors faire migrer tous ses attributs dans la(les) relation(s) issue(s) de la(des) sous-classe(s).
- **Décomposition ascendante :** Il faut supprimer la relation issue de la sous-classe et faire migrer les attributs dans la relation issue de la surclasse.

5.2 Passage vers le modèle relationnel

Après avoir appliqué les règles de passage au modèle relationnel, nous avons obtenu le schéma suivant :

Utilisateur(Id, Username, Email, Password, Date)
User(User_id#)
Admin(Admin_id#)
Document(Id, Nom, Url, Domain, Filiere, Option, Topic, admin_id#)
OtherDoc(Id, User_id#)
Rapport(Id#, Version#, filename, DateH, Similarité)
Engin(Version, Lien)

Pour l'héritage, on a appliqué la décomposition par distinction.

6 Conclusion

Tout au long de ce chapitre, nous avons proposé une modélisation avec UML à travers les diagrammes d'interactions et diagramme de classe qu'on a traduit en modèle relationnel afin de concevoir le schéma de la base de données. Le prochain chapitre fera l'objet de l'implémentation et réalisation de notre application.

Chapitre 4

Implémentation et réalisation

1 Introduction

Ce dernier chapitre est consacré à la partie pratique de notre projet. Nous y énumérerons les différents langages de programmation, ainsi que les outils de développement et les frameworks utilisés. Ensuite, nous présenterons les interfaces de note application ainsi que le fonctionnement de quelques unes.

2 Le web et l'application web

Le World Wide Web est un système de documentation hypertexte créé en 1993 qui a permis aux utilisateurs de se partager des documents et des images plus rapidement que via le courrier électronique et plus facilement que via le partage de fichiers.

En informatique, une application Web (aussi appelée site Web dynamique ou Web App) est un logiciel applicatif manipulable grâce à un navigateur Web. De la même manière que les sites Web, une application Web est généralement placée sur un serveur et se manipule en actionnant des widgets à l'aide d'un navigateur web via un réseau informatique (Internet, intranet, réseau local, etc.) [4]

3 Vue globale de notre application

Notre application permet le développement d'un espace pour l'utilisateur pour pouvoir traiter des fichiers et un tableau de bord pour l'administrateur afin de gérer les utilisateurs

et les fichiers, et cela nécessite une connexion pour les deux acteurs. La figure suivante nous montre une vue globale de notre application :

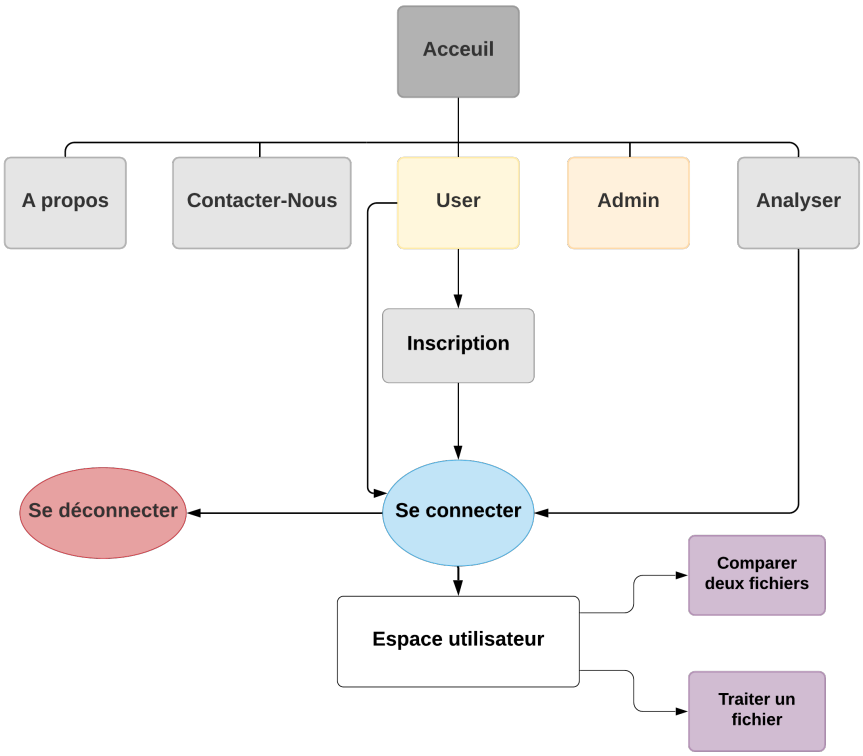

FIGURE 4.1 – Vue globale de l'application

4 Langages de programmation utilisés

4.1 Python

Python est un langage de programmation (au même titre que le C, C++, fortran, Java, etc.), développé en 1989 par **Guido van Rossum**. [25]

Parmi ses caractéristiques, on cite les principales [25] :

- «open-source» : son utilisation est gratuite et les fichiers sources sont disponibles et modifiables ;
- simple et très lisible ;
- doté d'une bibliothèque de base très fournie ;

- importante quantité de bibliothèques disponibles : pour le calcul scientifique, les statistiques, les bases de données... ;
- grande portabilité : indépendant vis à vis du système d'exploitation (linux, windows, MacOS) ;
- orienté objet : possible de concevoir en Python des entités qui miment celles du monde réel (une cellule, une protéine, un atome, etc.)avec un certain nombre de règles de fonctionnement et d'interactions
- typage dynamique : le typage (association à une variable de son type et allocation zone mémoire en conséquence) est fait automatiquement lors de l'exécution du programme, ce qui permet une grande flexibilité et rapidité de programmation, mais qui se paye par une surconsommation de mémoire et une perte de performance ;
- présente un support pour l'intégration d'autres langages. [19].

4.2 SQL

Le langage **SQL** signifie langage de requête structuré (*Structured Query Language*) l'un des plus anciens langages de programmation informatiques pour bases de données relationnelles. Il s'agit aussi du plus populaire [29]. Utilisé pour exploiter des bases de données. Il permet de façon générale la définition, la manipulation et le contrôle de sécurité de données. Il est bien supporté par la très grande majorité des systèmes de gestion de base de données (**SGBD**) [56].

4.3 HTML5

HTML (*Hypertext markup langage*) est un langage informatique descriptif, inventé en 1989 par **Tim Berners lee**, il s'agit d'un format de données qui permet la mise en forme des pages web [26].

L'un des principaux avantages du HTML est notamment son universalité : il peut être consulté sur n'importe quel terminal ou navigateur web [20].

HTML a vu de nombreuses mises à jour au fil du temps, et actuellement, la dernière version HTML est HTML5. HTML5 est bien sûr encore principalement un langage de balisage, mais il a ajouté une pléthore de fonctionnalités au HTML original et a éradiqué

une partie de la rigueur qui était présente dans le XHTML. Chaque jour, de nouvelles fonctionnalités sont ajoutées à **HTML5** [44].

L'un des objectifs qui sous-tendent la création de HTML5 est la prise en charge des documents multimédia sur les terminaux mobiles. Pour cela, des fonctions syntaxiques ont été créées, comme les balises video, audio et canvas [43].

4.4 CSS3

Les **CSS** (*Cascading Style Sheets*), permettent de mettre en forme des pages web de type HTML à l'aide des propriétés d'affichages (couleur, polices, bordure, etc.) et de positionnement (hauteur, largeur, etc.) Le résultat d'affichage d'une page web peut être entièrement changé sans ajouter un code additionnel dans la page web. D'ailleurs, l'objectif principal des feuilles de style est de séparer le contenu de la page de son aspect visuel [47]. CSS a vu de nombreuses versions dont CSS3 qui est la dernière.

4.5 JavaScript

Créé à l'origine par **Netscape**, ce langage de programmation est conçu pour traiter localement des événements provoqués par le lecteur (par exemple, lorsque le lecteur fait glisser la souris sur une zone de texte, cette dernière change de couleur). C'est un langage interprété, c'est-à-dire que le texte contenant le programme est analysé au fur et à mesure par l'interprète, partie intégrante du browser, qui va exécuter les instructions [13].

5 Environnement et outils de développement

5.1 VS Code

Visual Studio Code est un éditeur de code open-source développé par Microsoft qui peut être utilisé avec une variété de langages de programmation, notamment Java, JavaScript, Node.js et Python, grâce à des extensions. Il supporte l'auto-complétion, la coloration syntaxique, le débogage, et les commandes git [60].

5.2 PyCharm

PyCharm est un environnement de développement intégré (IDE) multiplateforme développé par **JetBrains** (entreprise tchèque) pour le langage Python, qui offre une expérience cohérente sur les systèmes d'exploitation Windows, macOS et Linux, et est publié à la fois en tant que logiciel open source et propriétaire payant [15] [39] .

Ce logiciel offre une excellente prise en charge spécifique aux frameworks de développement Web modernes telles que Django, Flask, Google App Engine, Pyramid et Web2py [15].

5.3 Le module venv

Venv est un paquet livré avec Python 3, c'est un outil qui permet de créer et gérer des environnements virtuels isolés. **Venv** crée un dossier qui contient tous les exécutables nécessaires pour utiliser les paquets qu'un projet Python pourrait nécessiter. [40]

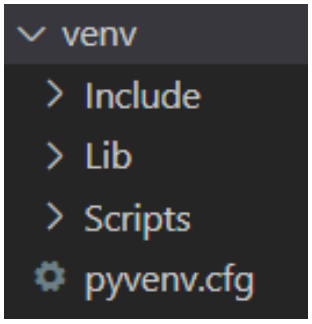

FIGURE 4.2 – Le module venv

- Le répertoire **Include** contient des en-têtes C qui compilent les packages Python.
- Le répertoire **Lib** contient une copie de la version Python. Il inclut également le sous-répertoire **site-packages.**
- Le répertoire **Scripts** contient tous les packages installés sur Python.
- **Pyvenv.cfg** est un fichier de configuration qui stocke des informations sur l'environnement virtuel, telles que la version originale de Python à partir de laquelle l'environnement a été cloné, ainsi que la version de l'interpréteur.

5.4 Wampserver

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données **MySQL**. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données [61].

5.5 MySQL

MySQL est donc un Système de Gestion de Bases de Données Relationnelles, qui utilise le langage **SQL**. C'est un des **SGBDR** les plus utilisés. Sa popularité est due en grande partie au fait qu'il s'agit d'un logiciel Open Source, ce qui signifie que son code source est librement disponible et que quiconque qui en ressent l'envie et/ou le besoin peut modifier MySQL pour l'améliorer ou l'adapter à ses besoins. Une version gratuite de MySQL est par conséquent disponible. À noter qu'une version commerciale payante existe également [23].

5.6 Adobe Photoshop

Photoshop est un logiciel de retouche, de traitement et de dessin assisté par ordinateur. Il a été créé à l'origine en 1988 par Thomas et John Knoll, édité par Adobe. Il est principalement utilisé pour le traitement de photographies numériques, mais sert également à la création d'images [3].

6 Frameworks et ORM

Un **framework** Signifie « **Cadre de travail** », Contient des composants autonomes qui permettent de facilité le développement d'un site web ou d'une application, ces composant résolvent des problèmes souvent rencontré par le développeur. Ils permettent donc de gagner du temps et d'être plus efficace lors du développement du site [15] [42]. Les principaux avantages sont [18] :

- la réutilisation des codes,
- la standardisation de la programmation,
- la formalisation d'une architecture adaptée aux besoins de chaque entreprise.

ORM (*Object-Relational Mapping*) est une technique qui permet d'interroger et manipuler des données à partir d'une base de données à l'aide d'un paradigme orienté objet[14].

Un ORM fournit généralement les fonctionnalités suivantes [31] :

- génération à la volée des requêtes SQL les plus simples (CRUD),
- prise en charge des dépendances entre objets pour la mise en jour en cascade de la base de données,
- support pour la construction de requêtes complexes par programmation.

Dans ce projet, nous avons utilisé deux frameworks (Bootstrap et Flask) et un ORM (SQLAlchemy).

6.1 Bootstrap

Le framework **Bootstrap** est un ensemble de fichiers CSS et JavaScript fonctionnant ensemble qui contiennent des règles prédéfinies et qui définissent des composants qu'on va pouvoir utiliser pour créer des design complexes de manière relativement simple. Ces ensembles de règles sont enfermés dans des classes et nous n'aurons donc qu'à utiliser les classes qui nous intéressent afin d'appliquer un ensemble de styles à tel ou tel élément HTML.

De plus, Bootstrap utilise également des librairies JavaScript externes comme jQuery ou Popper pour définir des composants entiers comme des barres de navigation, des fenêtres modales, etc. qu'on va pouvoir également directement implémenter [38].

Argon Dashboard est un template d'administration dans Bootstrap 4, HTML5. Il est open source, gratuit et est doté de nombreuses fonctionnalités. Ce modèle dispose d'une vaste gamme de composants, et chacun des éléments a des couleurs, des styles et des survols différents. De plus, **Argon Dashboard** a un design qui suscite l'intérêt avec une documentation détaillée [5].

6.2 Flask

Flask est un microframework Python facile et simple qui permet de faire des applications web évolutives. Flask dépend de la boite à outils WSGI de Werkzeug et du moteur de templates Jinja.

Le « micro» dans le micro-framework signifie que Flask vise à garder le code de base simple mais extensible [27].

6.3 SQLAlchemy

SQLAlchemy est le Python SQL toolkit et Object Relational Mapper qui donne aux développeurs d'applications la pleine puissance et la flexibilité de SQL.

Il fournit une suite complète de modèles de persistance au niveau de l'entreprise bien connus, conçus pour un accès efficace et performant à la base de données, adaptés dans un langage de domaine simple et pythonique [57].

7 Principales notions de développement

7.1 Jinja2

Jinja2 est une bibliothèque pour Python conçue pour être flexible, rapide et sécurisée. **Jinja2** est un langage de modélisation moderne et convivial pour Python. Il est rapide, largement utilisé et sécurisé avec l'environnement d'exécution de modèle en bac à sable facultatif. Jinja2 est plus lisible car sa syntaxe est facile à visualiser et à distinguer du code HTML [7].

En outre, étant donné que notre projet tourne autour de l'extraction de données, et donc le domaine des data sciences (sciences de données), nous devons définir certaines notions relatives à ce dernier :

7.2 Data mining

L'ensemble des méthodes, techniques et outils qui permettent de mettre à jour une connaissance à partir d'un grand volume de données, dans le but de dégager des corrélations et de nouvelles informations inconnues [33].

7.3 Web scraping

Technique permettant de récupérer des informations d'un site web, grâce à un programme ou un logiciel et de les réutiliser ensuite. En automatisant ce procédé, on évite ainsi de devoir récolter les données manuellement, on gagne du temps et on accède à un fichier unique et structuré [2].

7.4 NLP

Le traitement du langage naturel (NLP) est le traitement automatique ou semi-automatique du langage humain. Dans le domaine informatique, le NLP est lié aux techniques de compilation, à la théorie du langage formel, à l'interaction homme-machine, à l'apprentissage automatique et à la démonstration de théorèmes [38].

7.4.1 NLTK

Le NLTK, ou Natural Language Toolkit, est une suite de bibliothèques logicielles et de programmes. Elle est conçue pour le traitement naturel symbolique et statistique du langage en langage Python. C'est l'une des bibliothèques de traitement naturel du langage les plus puissantes.

Cette suite d'outils rassemble les algorithmes les plus communs du traitement naturel du langage comme le tokenizing, le part-of-speech tagging, le stemming, l'analyse de sentiments, la segmentation de topic ou la reconnaissance d'entité nommée [36].

7.4.2 L'indexation

L'indexation fait partie des fonctions qu'offre la **librairie NLTK**. C'est l'opération qui vise à construire une structure d'indexe qui permet de retrouver très rapidement les documents incluant les mots demandés. Cette étape consiste à analyser un document afin de créer un ensemble de mots-clés. Son objectif est de trouver les concepts les plus importants de ce dernier qui formeront le descripteur du document [51].
La figure ci-dessous présente les étapes suivies durant la réalisation de l'application.

- **Tokénisation des documents :** Cette étape consiste à transformer un texte en un ensemble de termes, en supprimant la majuscule, ainsi que les séparateurs (ponctuation, parenthèses, chiffres, etc.) [51].

FIGURE 4.3 – Etapes suivies lors de la phase d'indexation

Exemple :

1- Le processus unifié (Unified Process) se caractérise par une démarche itérative et incrémentale, pilotée par les cas d'utilisation, et centrée sur l'architecture et les modèles UML.

le	processus	unifié	unified	process	se	caractérise	par	une	démarche	itérative	et
incrémentale	pilotée	par	les	cas	d	utilisation	et	centrée	sur	l	architecture
et	les	modèles	uml								

FIGURE 4.4 – Exemple d'une tokénisation

- **Elimination des « stop words » :** Les stop words ou mots vides sont les mots non significatifs trouvés dans les documents. En effet, ces mots ne traitent pas le sujet du document mais ils permettent de lier entre les mots d'une phrase pour la structurer comme les articles, les conjonctions de coordination, les verbes auxiliaires, etc. Chaque langue a sa propre liste de stop words [51].

processus	unifié	unified	process	caractérise	démarche	itérative	incrémentale
pilotée	cas	utilisation	centrée	architecture	modèles	uml	

FIGURE 4.5 – Elimination des stop words

- **Stemmatisation :** La stemmatisation consiste à regrouper les mots ayant la même racine synthaxique [11] .

  Exemple : ["traiter", "traitement", "traite", "traitera"]

8 Diagramme de déploiement

La figure suivante illustre les modules logiciels de notre application, et leur répartition sur différentes machines physiques :

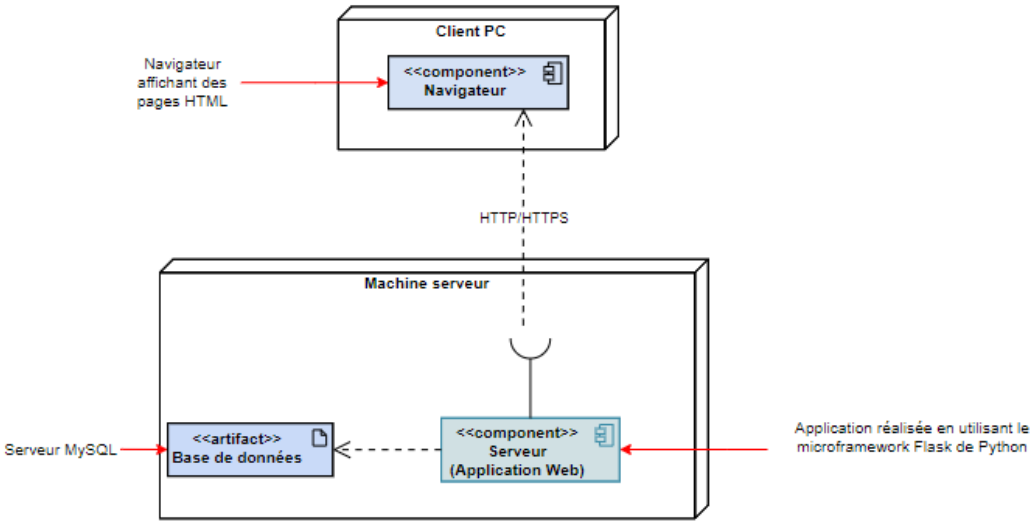

FIGURE 4.6 – Diagramme de déploiement

9 Logo

L'identité visuelle de notre application est représentée dans la figure suivante, dont :

— le choix des couleurs s'est porté en tenant compte des couleurs de l'Université : le bleu et le blanc ;

— **"NoPlag"** est le nom donné à l'application, il est très significatif, étant donné que le but de cette dernière est de combatte le plagiat ;

— Quant au graphisme, il fait référence à l'analyse faite sur un document.

FIGURE 4.7 – Logo de l'application

10 Quelques interfaces de notre système

Dans les figures qui suivent, nous présentons quelques interfaces de notre application.

10.1 La page d'acceuil

La page d'accueil est la première page à laquelle le visiteur de l'application web accède.

FIGURE 4.8 – L'interface "Acceuil"

10.2 Connexion

Les utilisateurs doivent s'authentifier pour pouvoir accéder à l'application.

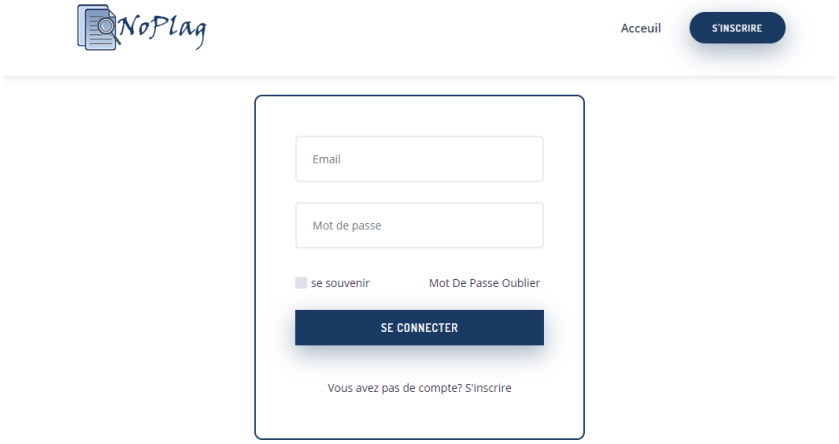

FIGURE 4.9 – L'interface "Connexion"

10.3 Inscription

Afin de pouvoir s'authentifier, certains utilisateurs doivent d'abord s'inscrire.

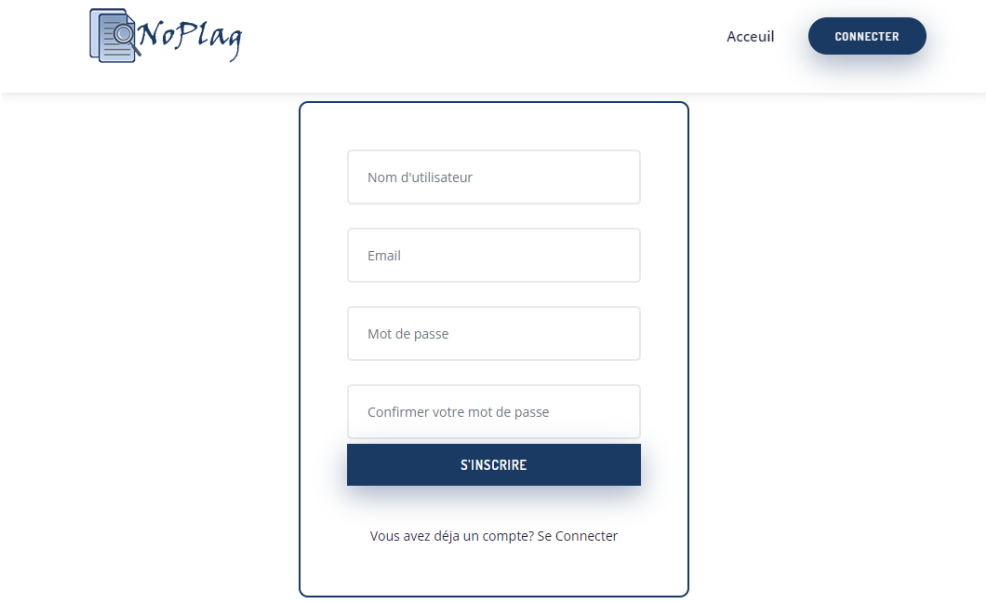

FIGURE 4.10 – L'interface "Inscription"

10.4 Espace utilisateur

L'utilisateur a le choix entre l'analyse d'un fichier et une comparaison entre deux fichers.

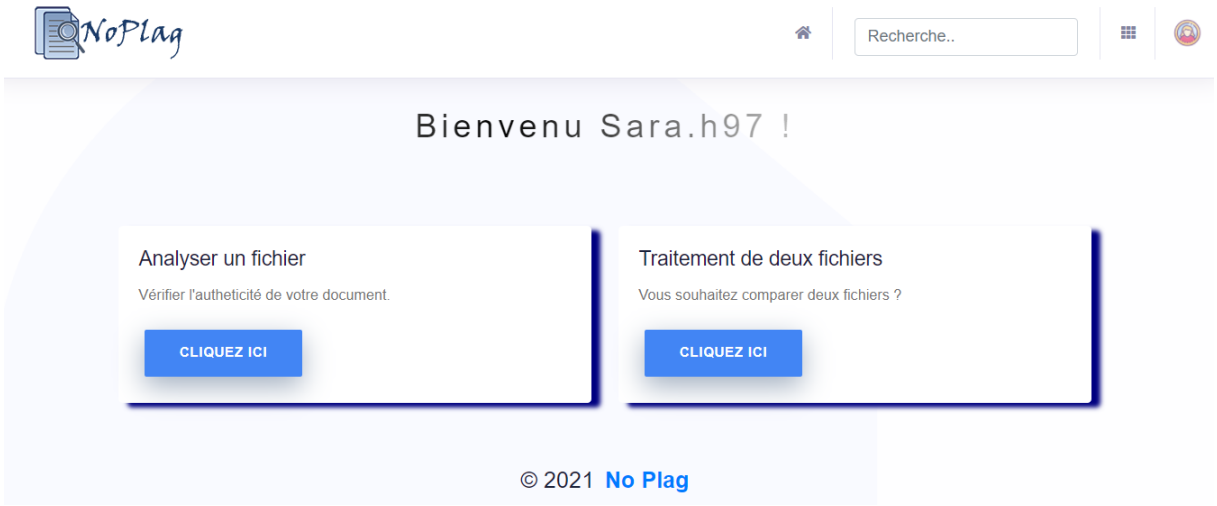

FIGURE 4.11 – L'interface "Accueil utilisateur"

10.4.1 Analyser un fichier

L'utilisateur doit importer un fichier pour le faire analyser.

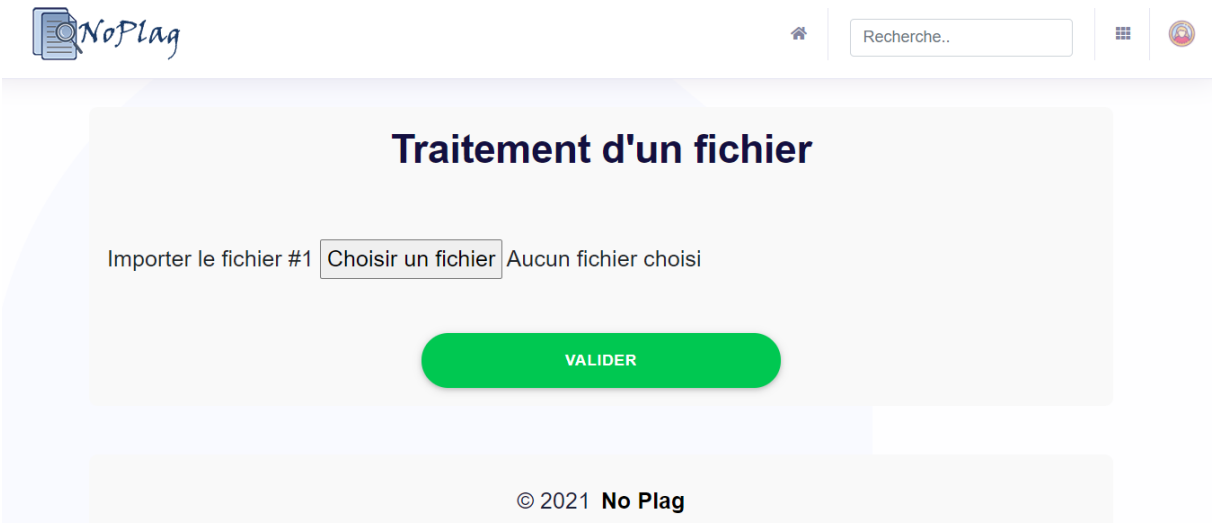

FIGURE 4.12 – L'interface "Traitement d'un fichier"

10.4.2 Traitement de deux fichiers

L'utilisateur peut obtenir le taux de similarité entre deux fichiers, ou bien choisir d'effectuer une comparaison plus détaillée.

FIGURE 4.13 – L'interface "Comparer deux fichiers"

10.4.3 Comparaison rapide de deux fichiers

L'utilisateur doit importer deux fichiers afin d'effectuer une comparaison rapide entre eux.

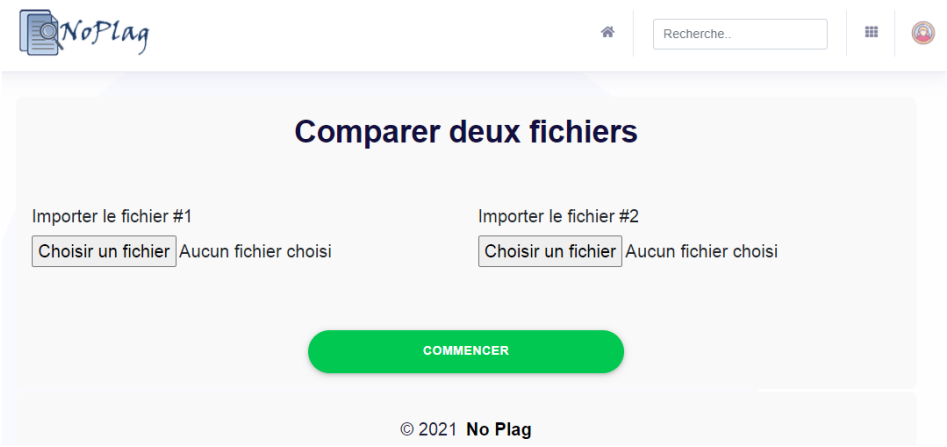

FIGURE 4.14 – L'interface "Comparaison rapide de deux fichiers"

10.4.4 Comparaison détaillée de deux fichiers

L'utilisateur choisit les critères de comparaison entre deux fichiers après les avoir importer. le système parmet la comparaison : ligne **par ligne**, phrase **par phrase** et **par sous-chaînes**, en choisissant le nombre de caractères à prendre en considération durant la comparaison.

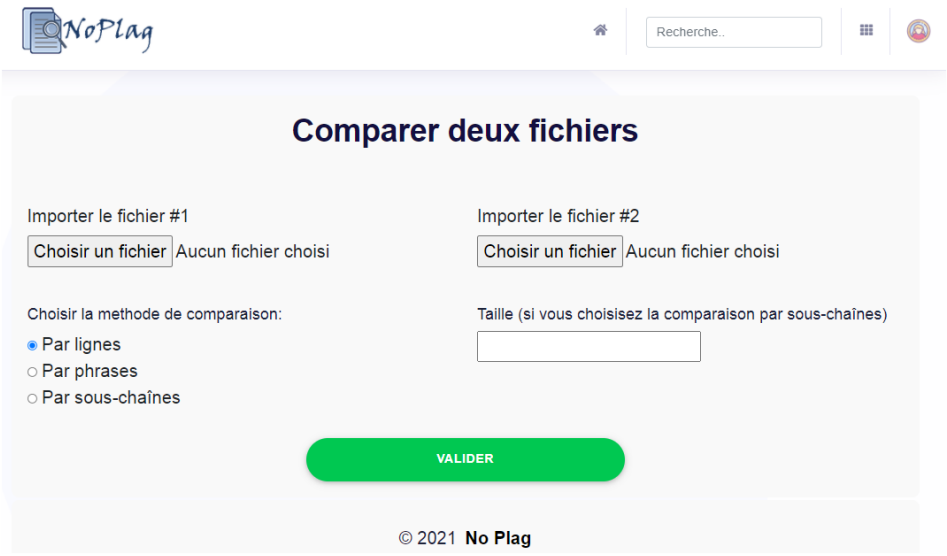

FIGURE 4.15 – L'interface "Comparaison détaillée de deux fichiers"

10.5 Admin

L'administrateur saisit son email et mot de passe afin de s'authentifier.

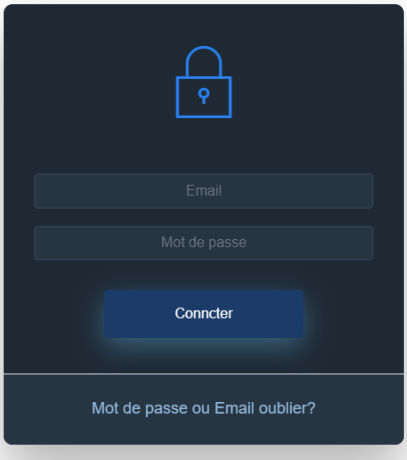

FIGURE 4.16 – L'interface "Connexion à l'espace Admin"

10.6 Dashboard admin

Après avoir effectué l'authentification, l'administrateur accède à sa page d'accueil où il pourra effectuer ses fonctionnalités : l'ajout d'un utilisateur, l'ajout de documents, la consultation des listes des utilisateurs ainsi que des fichiers.

FIGURE 4.17 – L'interface "Dashboard Admin"

10.6.1 Ajouter des utilisateurs

L'administrateur remplit un formulaire contenant les informations relatives à un utilisateur pour ajouter celui-ci à la liste des utilisateurs.

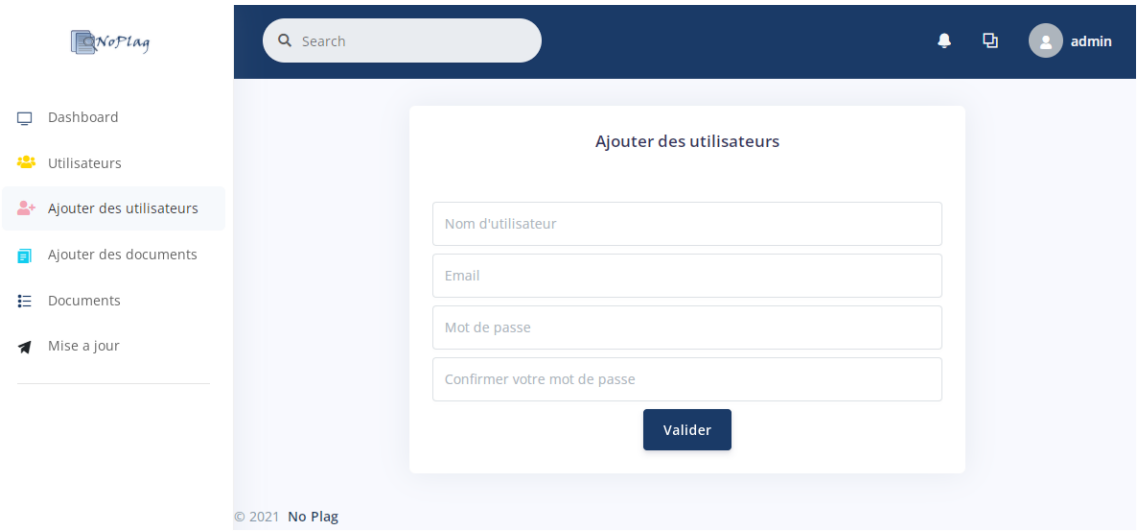

FIGURE 4.18 – L'interface "Ajouter des utilisateurs"

10.6.2 Ajouter des documents

L'administrateur remplit un formulaire comprenant les informations relatives à un document afin d'ajouter ce dernier à la liste des documents.

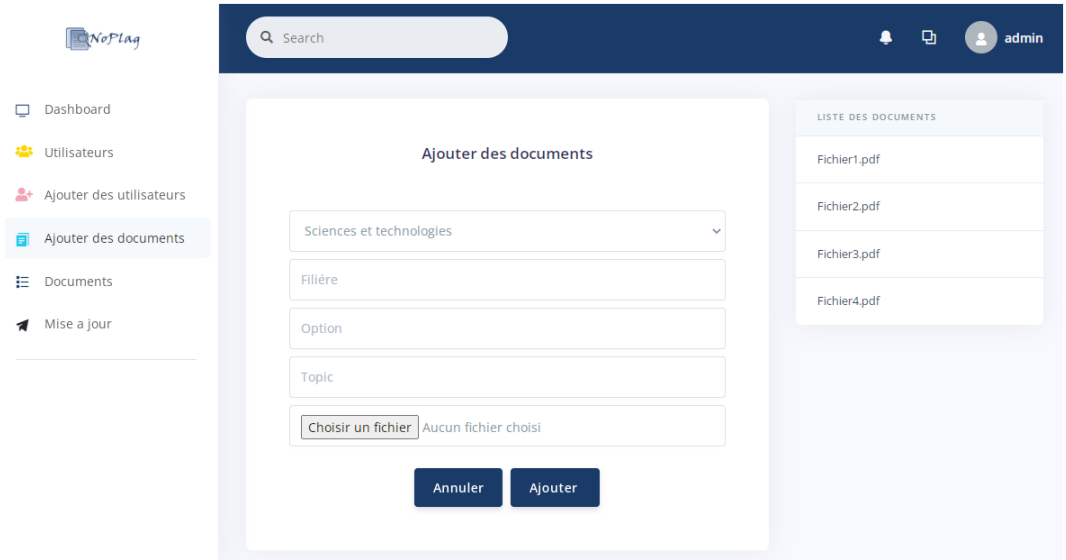

FIGURE 4.19 – L'interface "Ajouter des documents"

10.6.3 Consulter la liste des utilisateurs

L'administrateur peut ainsi avoir accès à la liste de tous les utilisateurs inscrits/ajoutés sur le site.

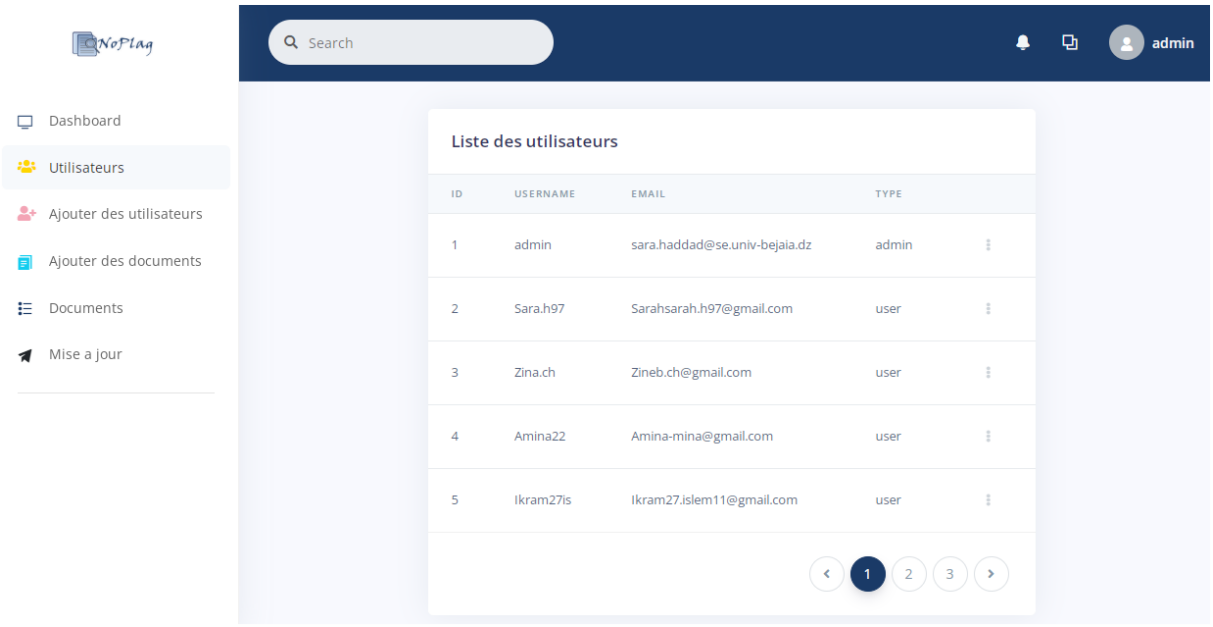

FIGURE 4.20 – L'interface "Liste des utilisateurs"

10.6.4 Consulter la liste des fichiers

L'administrateur peut accéder à la liste détaillée des documents contenus dans le corpus et.

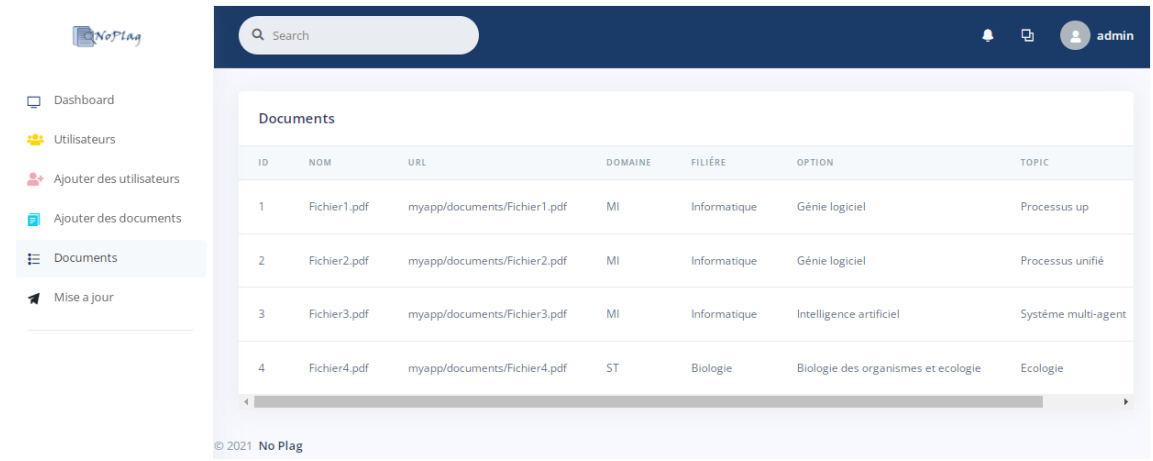

FIGURE 4.21 – L'interface "Liste des documents"

11 Fonctionnement de quelques interfaces

11.1 Analyse d'un fichier

L'utilisateur importe un fichier, et obtient le taux de similarité avec le corpus de documents, après avoir validé.

FIGURE 4.22 – Importer un fichier

La figure 4.23 montre le résultat obtenu.

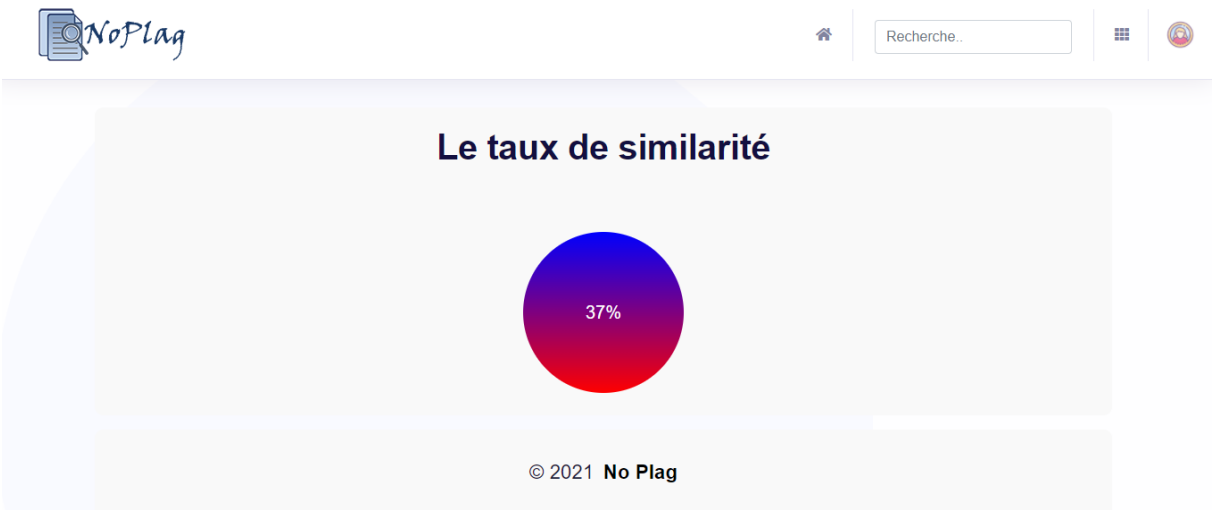

FIGURE 4.23 – Résultat de l'analyse

11.2 Traitement de deux fichier

11.2.1 Traitement simple

L'utilisateur démarre une comparaison après avoir importé deux fichiers, et obtient ensuite le taux de similarité entre ces derniers.

FIGURE 4.24 – Importer deux fichiers

La figure 4.25 montre le résultat obtenu.

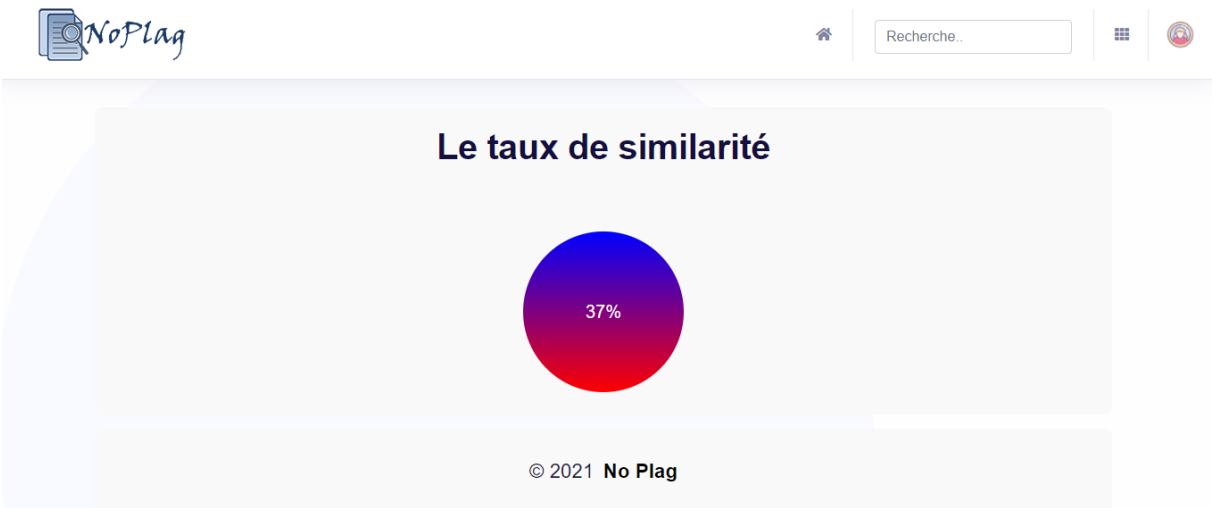

FIGURE 4.25 – Résultat de l'analyse

11.2.2 Comparaison détaillée

Exemple de comparaison détaillée avec la méthode **par phrase** .

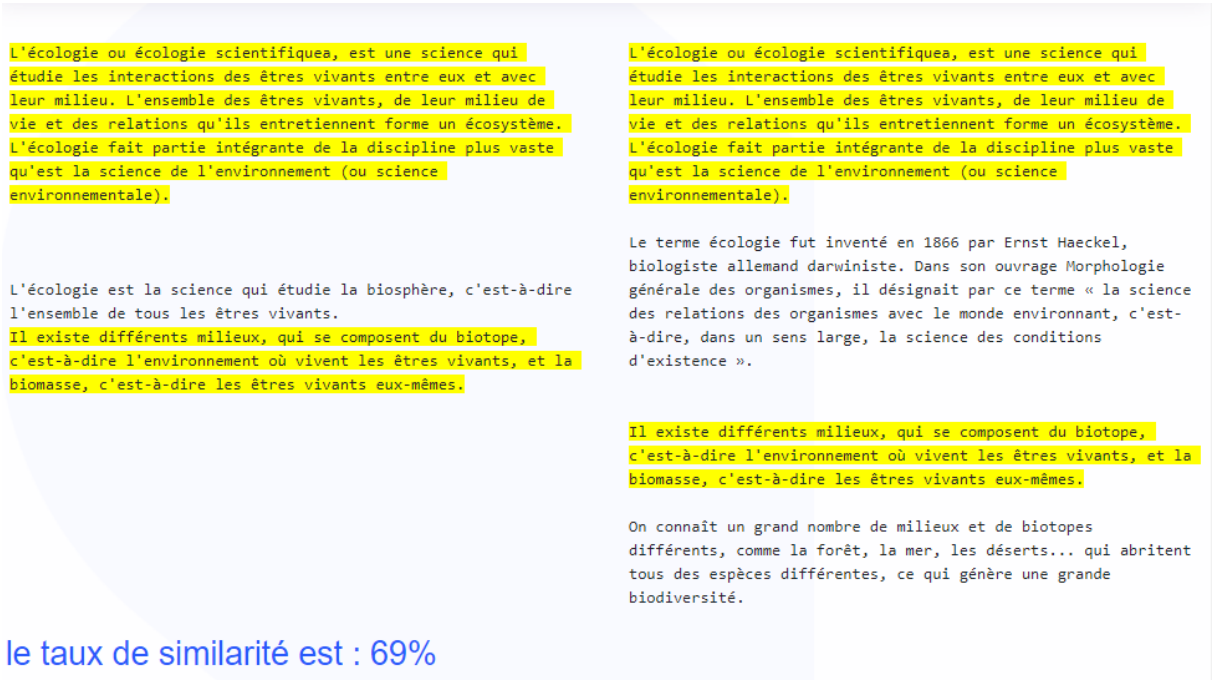

FIGURE 4.26 – Résultat de la comparaison détaillée

12 Conclusion

Dans ce dernier chapitre, nous avons mis en avant les langages, frameworks, outils et logiciels qui ont contribué à l'élaboration de notre application web ainsi que quelques interfaces de l'application afin d'avoir une vue générale de cette dernière.

Conclusion générale

Ce travail a été réalisé dans le cadre de notre projet de fin de cycle master en génie logiciel. Au cours de ce mémoire, nous avons présenté les différentes étapes de la conception et réalisation de notre application web pour la détection de plagiat, qui n'est qu'une initiation au système que nous souhaitons réaliser. En effet, par faute de temps, la réalisation complète de ce dernier été impossible.

Pour cela, nous avons commencé la conception en utilisant le formalisme UML, et cela, en suivant le cycle de vie du processus de développement UP. La réalisation a été faite avec le langage Python, en utilisant le microframework Flask.

Le présent travail nous a permis de mettre en pratique toutes nos connaissances théoriques acquises durant notre parcours universitaire, mais aussi d'enrichir davantage notre expérience notamment dans le domaine de la science de données.

Comme perspective, nous espérons voir notre application évoluer, et ce, en intégrant l'affichage des sources afn d'aboutir à une application utilisable qui permettra aux utilisateurs d'analyser leurs travaux de façon optimale.

Webographie

[1] *568px-Processus_unifié_-_enchainements_d'activités_au_cours_du_cycle_de_vie.png (568×284)*. URL : https://upload.wikimedia.org/wikipedia/commons/thumb/a/a4/Processus_unifi%C3%A9_-_enchainements_d%27activit%C3%A9s_au_cours_du_cycle_de_vie.png/568px-Processus_unifi%C3%A9_-_enchainements_d%27activit%C3%A9s_au_cours_du_cycle_de_vie.png (visité le 12/06/2021).

[2] AdminLaou. *Tout savoir sur la méthode du web scraping*. fr-FR. Oct. 2020. URL : https://www.laou.fr/conseils-pro/tout-savoir-sur-la-methode-du-web-scraping/ (visité le 21/09/2021).

[3] *Adobe Photoshop - Définition et Explications*. fr-FR. URL : https://www.techno-science.net/glossaire-definition/Adobe-Photoshop.html (visité le 12/09/2021).

[4] *Application Web - Définition et Explications*. fr-FR. URL : https://www.techno-science.net/glossaire-definition/Application-Web.html (visité le 04/09/2021).

[5] *Argon Dashboard - Free Bootstrap 4 HTML5 Admin Template*. en-US. URL : https://themewagon.com/themes/free-bootstrap-4-html5-admin-template-argon-dashboard/ (visité le 20/09/2021).

[6] Shivam Arora. *Similitude de cosinus en Python*. fr. Juill. 2021. URL : https://www.delftstack.com/fr/howto/python/cosine-similarity-between-lists-python/ (visité le 20/09/2021).

[8] *Besoins fonctionnels & Besoins non fonctionnels - Savoir+*. fr-FR. Déc. 2019. URL : https://savoir.plus/besoins-fonctionnels-non-fonctionnels/ (visité le 16/06/2021).

[11] Benoit Cayla. *Du NLP avec Python NLTK*. fr-FR. Août 2020. URL : https://www.datacorner.fr/nltk/ (visité le 10/10/2021).

[12] *Chapitre3_ g\u00e9nie.pdf - Minist\u00e8re de l\u2019Enseignement Sup\u00e9rieur et de la Recherche Scientifique Universit\u00e9 de Mila D\u00e9partement Math Informatique | Course Hero*. URL : https://www.coursehero.com/file/53389601/Chapitre3-g%C3%A9niepdf/?fbclid=IwAR2vQAIBm58ImkUSV0wR6jxuWwa2c4tpKTsawd_7CA_nA3w0Qnpp0n87wdo (visité le 07/09/2021).

[14] *database - Qu'est ce qu'un ORM, comment ça fonctionne, et comment dois-je utiliser ?* fr-FR. Juill. 2019. URL : https://askcodez.com/quest-ce-quun-orm-comment-ca-fonctionne-et-comment-dois-je-utiliser.html (visité le 12/09/2021).

[15] *Definition of PyCharm*. en. URL : https://www.pcmag.com/encyclopedia/term/pycharm (visité le 12/09/2021).

[18] *Framework : définition simple et objectifs du terme*. fr-FR. Section : Développement web. URL : https://www.1min30.com/dictionnaire-du-web/framework (visité le 12/09/2021).

[20] la rédaction de FUTURA. *HTML*. fr. Section : Internet. URL : https://www.futura-sciences.com/tech/definitions/internet-html-480/ (visité le 12/09/2021).

[26] *HTML (HyperText Markup Langage) : définition, traduction*. fr. URL : https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203255-html-hypertext-markup-langage-definition-traduction/ (visité le 12/09/2021).

[27] KAHERECODE. *Démarrer avec Flask - Un micro framework Python*. fr. URL : https://www.kaherecode.com/tutorial/demarrer-avec-flask-un-micro-framework-python (visité le 12/09/2021).

[29] +Bastien L. *Qu'est-ce que SQL ? Tout savoir sur le langage des bases de données*. fr. Section : Analytics. Mai 2021. URL : https://www.lebigdata.fr/sql-tout-savoir-guide (visité le 12/09/2021).

[31] *Les ORM (Object-Relational Mapping) — Documentation Java ORM / Spring*. URL : https://gayerie.dev/epsi-b3-orm/javaee_orm/intro.html (visité le 12/09/2021).

[32] *Logiciel de diagramme de contexte*. fr. URL : https://www.lucidchart.com/pages/fr/exemple/logiciel-diagramme-de-contexte (visité le 15/05/2021).

[33] *Logiciels de data mining et BI : définition, techniques et outils (gratuits, open source et pro)*. fr-FR. URL : https://www.appvizer.fr/magazine/analytique/data-mining/logiciels-bi-data-mining-portee-de-tous (visité le 15/09/2021).

[34] manu rnx manu. *Expression des besoins avec UML*. en. Juin 2015. URL : https://medium.com/@manurnx/les-cas-d-utilisation-64cc69b9a67f (visité le 13/05/2021).

[35] *Memoire Online - conception et réalisation d'une application de gestion d'un centre de kinésie - abdelbasset jarray*. URL : https://www.memoireonline.com/07/08/1363/m_conception-realisation-application-gestion-centre-kinesie9.html (visité le 10/05/2021).

[36] *NTLK : guide de l'outil de Traitement Naturel du Langage en Python*. fr-FR. Avr. 2021. URL : https://datascientest.com/nltk (visité le 21/09/2021).

[37] *Plagiat : qu'est-ce que le plagiat ?* fr-FR. URL : https://www.scribbr.fr/category/le-plagiat/ (visité le 05/05/2021).

[38] *Présentation de Bootstrap - Pierre Giraud*. URL : https://www.pierre-giraud.com/bootstrap-apprendre-cours/introduction/ (visité le 12/09/2021).

[39] *PyCharm : l'IDE Python pour développeurs professionnels par JetBrains*. fr. URL : https://www.jetbrains.com/fr-fr/pycharm/ (visité le 12/09/2021).

[40] *python — Quelle est la différence entre venv, pyvenv, pyenv, virtualenv, virtualenvwrapper, pipenv, etc. ?* URL : https://www.it-swarm-fr.com/fr/python/quelle-est-la-difference-entre-venv-pyvenv-pyenv-virtualenv-virtualenvwrapper-pipenv-etc. (visité le 20/09/2021).

[41] *Python venv - virtual environments in Python using venv*. URL : https://zetcode.com/python/venv/?fbclid=IwAR3Oa1VovrI5pTJHB1Kf0uEJtnFhvSf3leYi6mgpM4T-tNuEg74PREKVgCg (visité le 10/10/2021).

[42] *Qu'est-ce qu'un framework ? - Wild Code School*. fr. URL : https://www.wildcodeschool.com/fr-FR/blog/framework-definition-developpement-web-programmation (visité le 12/09/2021).

[43] *Que signifie HTML5 ? - Definition IT de Whatis.fr*. fr. URL : https://www.lemagit.fr/definition/HTML5 (visité le 12/09/2021).

[44] *Quelles sont les différences entre la version HTML Et HTML5 ?* fr-FR. Avr. 2017. URL : https://www.hostinger.fr/tutoriels/differences-html (visité le 12/09/2021).

[45] *Quels services essentiels le système d'information (SI) apporte-t-il aux organisations ? COURS par internet pendant le coronavirus PREMIERE GESTION.* URL : https://formations.valeurs-et-passion.org/cours-internet-lmn/cours-formation-lycee-premiere-gestion-chapitre8.asp# (visité le 12/05/2021).

[46] *Raisons pouvant mener un étudiant à plagier.* fr. Fév. 2016. URL : https://etudiant.polymtl.ca/plagiat/raisons-pouvant-mener-un-etudiant-plagier (visité le 10/05/2021).

[56] *SQL (Structured Query Language) : définition, traduction et acteurs.* fr. URL : https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203603-sql-structured-query-language-definition-traduction-et-acteurs/ (visité le 12/09/2021).

[57] *SQLAlchemy - The Database Toolkit for Python.* URL : https://www.sqlalchemy.org/ (visité le 12/09/2021).

[58] *Système d'information et organisation.* URL : http://www.jybaudot.fr/SI/orga.html (visité le 12/05/2021).

[59] *UQAM | Infosphère | Citer ses sources.* URL : http://www.infosphere.uqam.ca/rediger-un-travail/citer-ses-sources (visité le 10/05/2021).

[60] *Visual Studio Code.* fr. Mars 2019. URL : https://framalibre.org/content/visual-studio-code (visité le 12/09/2021).

[61] *WampServer.* fr-FR. URL : https://www.wampserver.com/ (visité le 12/09/2021).

Bibliographie

[7] Fankar Armash Aslam et al. “Efficient way of web development using python and flask”. en. In : *International Journal of Advanced Research in Computer Science* 6.2 (2015), p. 54-57. issn : 09765697.

[9] Hadj Ahmed Bouarara et Hanane Bendenia. *Détection des Différents types de Plagiat : Plagiat des idées, plagiat paraphrasé, plagiat avec traduction, plagiat avec synonymie, plagiat des images*. Éditions universitaires européennes, 2019.

[10] Olivier Capuozzo. “Cas d’utilisation, une introduction”. In : *France, Editions CERTA* (2004).

[13] Daniel Charnay et Philippe Chaléat. *HTML et Javascript*. Eyrolles, 1998.

[16] HAMBI El Mostafa. “Contribution dans les méthodes de détection du plagia académique”. In : (2016).

[17] Jérémy Ferrero et Alain Simac-Lejeune. “Méthode alternative à la détection de «copier/coller» : intersection de textes et construction de séquences maximales communes”. In : *15e Conférence Internationale Francophone sur l’Extraction et la Gestion des Connaissances*. 2015.

[19] Patrick FJ Fuchs et Pierre Poulain. “Introduction à la programmation Python pour la biologie”. Thèse de doct. Université de Paris, 2020.

[21] Joseph Gabay et David Gabay. *UML 2 Analyse et conception : Mise en œuvre guidée avec études de cas*. Dunod, 2008.

[22] Pierre Gérard. “Processus de Développement Logiciel”. In : (), p. 27.

[23] Chantal Gribaumont. *Administrez vos bases de données avec MySQL*. OpenClassrooms, 2014.

[24] Salima Hassas. “Unified Modeling Language UML”. In : (2005).

[25] Florent HIVERT. “Introduction à la programmation en langage Python”. fr. In : (fév. 2016), p. 19.

[28] Sebastien KRAMM. “Introduction à la modélisation UML - Module RCPI01”. fr. In : (), p. 10.

[30] Imen LARIBI. “Conception et implémentation d’une application (2-tiers) avec UML et Java Application de gestion de magasin de livres (GML).” Thèse de doct.

[47] Hassen Ben REBAH, Hafedh BOUKTHIR et Antoine CHÉDEBOIS. *Conception et réalisation de sites web avec HTML5 et CSS3*. ISTE Group, 2021.

[48] Pascal ROQUES. *UML 2 : Modéliser une application web*. Editions Eyrolles, 2008.

[49] Pascal ROQUES et Franck VALLÉE. “UML 2 en action”. In : *De l’analyse des besoins ı*** la conception J2EE* (2004), p. 15.

[50] Pascal ROQUES et Franck VALLÉE. *UML 2 en action : de l’analyse des besoins à la conception*. Editions Eyrolles, 2011.

[51] Rima ROUIBIA, Imane BELHADJ, MOHAMED AMINE CHERAGUI et al. “Outil de détection de plagiat dans un document”. Thèse de doct. Université Ahmed Draı¨a-Adrar, 2016.

[52] Boubker SBIHI. “Informatisation d’une médiathèque à travers la norme UML”. In : *Revue EPI : Enseignement Public et Informatique* (2005).

[53] Boubker SBIHI. “Informatisation d’une médiathèque à travers la norme UML”. In : 2005.

[54] Brigitte SIMONNOT. “Le plagiat universitaire, seulement une question d’éthique ?” In : *Questions de communication* 26 (2014), p. 219-233.

[55] Christian SOUTOU et Frédéric BROUARD. *UML 2 pour les bases de données*. Editions Eyrolles, 2012.

Résumé

Ce document a été rédigé en vue de l'obtention du diplôme de master en génie logiciel.

L'Université de Béjaia a besoin de faire analyser ses mémoires, afin de détecter le plagiat, et acquérir ainsi un maximum de crédibilité. Pour atteindre cet objectif, il nous a été proposé de concevoir et d'implémenter une application web assurant la détection de plagiat dans les mémoires de master.

Pour ce faire, nous avons suivi la démarche de développement logiciel UP et avons choisi de modéliser notre système avec le formalisme UML, notre choix s'est porté sur ce dernier par rapport à sa simplicité et sa performance en matière de conception. Pour ce qui est de la phase de réalisation, elle s'est caractérisée par l'utilisation de Flask, qui est un framework Python, ainsi que d'autres librairies pour l'implémentation de l'application web.

Mots-clés : Application web, détection de plagiat, plagiat universitaire, types de plagiat, méthodes de détection de plagiat.

Abstract

This document was written for a Master's degree in Software Engineering.

The University of Béjaia needs to have its dissertations analyzed, in order to detect plagiarism, and thus acquire maximum credibility. To achieve this goal, we were asked to design and implement a web application ensuring plagiarism detection in documents.

To do this, we followed the UP software development process and the UML modeling language. The implementation of our application was carried out under Pycharm, using Python and the Flask framework, as well as other libraries.

Keywords : web application, plagiarism detection, academic plagiarism, types of plagiarism, plagiarism detection methods.
