



Master 2 MBDS de l'Université de Nice Sophia Antipolis (UNS)  
« Stage en entreprise »

# **OUTIL D'AIDE A LA CORRECTION DES MEMOIRES DES ETUDIANTS DE L'IT-UNIVERSITY**

par  
RATEFINJANAHARY Jean Michaël

Jury :

Mme. Baovola Rajaonarison, examinateur

M. Rojo Rabenanahary, encadreur professionnel

## Résumé

Le thème du travail effectué durant ce stage au sein de l'ITU-Labs s'intitule « Outil d'aide à la correction des mémoires des étudiants de l'IT-University ». L'objectif principal de ce projet est d'implémenter un système de correction et de validation des mémoires sur la plateforme ITUStage. Ce nouveau module aidera les encadreurs dans la correction et permettra aux responsables de l'IT-University d'avoir un aperçu sur les thèmes les plus fréquents afin d'améliorer la qualité de l'enseignement, celui-ci aidera également les étudiants à éviter le plagiat d'anciens mémoires. C'est dans ce système que seront centralisés les mémoires des étudiants ainsi que les retours et les éventuelles corrections et remarques des encadreurs. Chaque mise à jour effectuée est historisée. Ce projet facilitera ainsi la relation et l'interaction entre l'encadreur et l'étudiant pendant la rédaction de leur mémoire.

Ce dernier a été modélisé selon le processus unifié en utilisant « UML » et développé sous PHP, React et NodeJs avec les bases de données Elasticsearch et MySQL et l'utilisation du « Natural Language Processing » ou NLP lors de l'extraction des données d'un mémoire.

**Mots clés :** Mémoire, Elasticsearch, NLP, Full-text search

## Abstract

The theme of the work done during this internship at ITU-Labs is entitled « Tool to assist in the correction of IT-University students' theses ». The main objective of this project is to implement a thesis correction and validation system on the ITUStage platform. This new module will help the supervisors in the correction and will allow the IT-University managers to have an overview on the most frequent topics in order to improve the quality of teaching, it will also help the students to avoid plagiarism of old theses. This system will centralize the students theses as well as the feedback and possible corrections and comments from the supervisors. Each update made is historized. This project will thus facilitate the relationship and interaction between the supervisor and the student during the writing of their thesis.

The latter was modeled according to the unified process using « UML » and developed under PHP, React and NodeJs with the Elasticsearch and MySQL databases and the use of « Natural Language Processing » or NLP when extracting data from a dissertation.

**Keywords :** Thesis, Elasticsearch, NLP, Full-text search

## Liste des figures

Figure 1 : Description externe du système logiciel existant.....	16
Figure 2 : Architecture de la solution envisagée .....	17
Figure 3 : Planification du projet (Diagramme de GANTT) .....	27
Figure 4 : Cas d'utilisation - Upload mémoire.....	29
Figure 5 : Cas d'utilisation - Vérification des copier-coller .....	30
Figure 6 : Cas d'utilisation - Envoi d'un mémoire à l'encadreur.....	31
Figure 7 : Cas d'utilisation - Correction d'un mémoire.....	32
Figure 8 : Page d'upload de mémoire .....	34
Figure 9 : Page d'upload mémoire après sélection d'un fichier .....	34
Figure 10 : Page de lecture .....	35
Figure 11 : Affichage des phrases copiées .....	35
Figure 12 : Envoi d'un mémoire pas encore corrigé.....	36
Figure 13 : Envoi d'un mémoire du même type déjà corrigé précédemment .....	36
Figure 14 : Correction d'un mémoire – Surlignements .....	37
Figure 15 : Envoi des corrections .....	37
Figure 16 : Description des API .....	38
Figure 17 : Architecture technique .....	39
Figure 18 : Diagramme de structuration interne des packages (Node.js).....	40
Figure 19 : MPD de la base de données MySql .....	41
Figure 20 : Cas d'utilisation - Vérification des copier-coller (Vue interne/développeur) ...	42
Figure 21 : Package models.....	45
Figure 22 : Package controllers Node.js .....	45
Figure 23 : Package controllers PHP.....	46
Figure 24 : Package repositories.....	46
Figure 25 : Package services .....	46

## Liste des tableaux

Tableau 1 : Tableau comparatif des solutions au vu des critères .....	15
Tableau 2 : Outils utilisés .....	22
Tableau 3 : Contraintes et risques sur le projet .....	24
Tableau 4 : Démarche du projet .....	26
Tableau 5 : Budget du projet .....	28

## Liste des acronymes

API : Application Programming Interface

CC : Copier-coller

HTTPS : HyperText Transfer Protocol Secure

IHM : Interface Homme Machine

JWT : Json Web Token

MPD : Modele Physique de Données

MVC : Model View Controller

NLP : Natural Language Processing

PDF : Portable Document Format

PHP : PHP Hypertext Preprocessor

SDK : Software Development Kit

SQL : Structured Query Language

## Table des matières

Liste des figures.....	2
Liste des tableaux .....	3
Liste des acronymes.....	4
1. Introduction générale.....	7
2. Présentation du stage .....	8
2.1 Présentation de l'entreprise .....	8
2.2 Présentation du sujet et objectifs du projet.....	8
2.2.1 Sujet et contexte générale .....	8
2.2.2 Objectifs du projet .....	9
2.2.3 Innovation apportée .....	9
2.2.4 Principaux risques et enjeux .....	9
3. État de l'art sur le sujet traité.....	10
3.1 Critères de comparaison .....	10
3.2 Étude de chaque solution au vu des critères .....	11
3.3 Tableau comparatif des solutions au vu des critères .....	15
4. Étude de l'existant et solution envisagée.....	16
4.1 Étude de l'existant .....	16
4.1.1 Description externe du système logiciel existant (vision utilisateur).....	16
4.1.2 Description interne du système logiciel existant (vision développeur/conception) .....	17
4.2 Critiques de l'existant.....	17
4.3 Solutions envisagées.....	17
4.4 Objectifs principaux et livrables.....	18
4.4.1 Serveur d'application du projet .....	18
4.4.2 Back-Office du projet .....	18
4.4.3 Application web pour les étudiants .....	19
5. Démarche projet .....	20
5.1 Principes de la démarche projet.....	20
5.1.1 Activités d'ingénierie logicielle.....	20
5.1.2 Méthode de gestion de projet utilisé.....	20
5.1.3 Rôles et responsabilités .....	21

5.1.4	Outils utilisés .....	21
5.1.5	Gestion de la configuration.....	23
5.2	Contraintes et risques sur le projet .....	24
5.3	Démarche du projet : mise en œuvre .....	24
5.4	Planification.....	27
5.5	Budget du projet .....	28
6.	Exigences réalisées dans le projet (vision externe / utilisateur) .....	29
6.1	Exigences fonctionnelles – Cas d’utilisation.....	29
6.1.1	Cas d’utilisation – Upload de mémoire .....	29
6.1.2	Cas d’utilisation – Vérification des copier-coller.....	30
6.1.3	Cas d’utilisation – Envoi d’un mémoire à l’encadreur.....	31
6.1.4	Cas d’utilisation – Correction d’un mémoire .....	32
6.2	Exigences non fonctionnelles transverses .....	33
6.3	Interfaces détaillées .....	34
6.3.1	IHM .....	34
6.3.2	Interfaces avec d’autres systèmes.....	38
7.	Architecture(s) système .....	39
7.1	Architecture logicielle .....	39
7.2	Architecture technique.....	39
8.	Conception du système logiciel réalisée dans le projet (vision interne/développeur). 40	
8.1	Conception du logiciel développé .....	40
8.1.1	Conception du code source.....	40
8.1.2	Le code source – vue statique.....	40
8.1.3	Modélisation de données .....	41
8.1.4	Réalisation des cas d’utilisation .....	42
8.1.5	Les composants et leur déploiement.....	42
9.	Tests du système logiciel.....	42
10.	Conclusion générale .....	43
11.	Références et Bibliographie.....	44
12.	Annexes .....	45

# 1. Introduction générale

Avec presque 300 nouveaux étudiants en 2019, les professeurs de l'IT-University auront à encadrer potentiellement près de 300 personnes en 2022 contre environ une centaine avant la construction de l'amphithéâtre.

La rédaction d'un mémoire est le résultat des efforts de l'étudiant d'un côté mais également de l'encadrement fourni par l'encadreur d'autre part. Un bon encadrement nécessite ainsi une bonne communication et une bonne organisation entre les deux parties.

Avec ce nombre croissant d'étudiants, l'encadrement peut devenir une tâche difficile pour l'encadreur notamment lors de la correction des mémoires.

En effet, pendant la période d'avant soutenance, lors de laquelle les étudiants finalisent leurs mémoires, les encadreurs sont inondés d'une vague de mails d'étudiants et d'une multitude de livres à lire, à corriger. Ils doivent vérifier et s'assurer manuellement que chaque étudiant ait pris en compte ses commentaires.

Quels moyens peuvent être envisagés afin de permettre un encadrement efficace et agile sans en compromettre sa qualité ?

C'est dans ce contexte que s'intègre ce projet, qui a pour objectif de mettre en place un module de correction de mémoire sur la plateforme ITUStage, un outil de validation de thème de stage.

En appliquant toutes les connaissances acquises tout au long des cinq années d'études au sein de l'IT-University et des expériences professionnelles, ce projet peut améliorer la qualité et l'organisation interne de l'IT-University.

Le présent mémoire aborde deux parties :

## Partie 1 : GENERALITE SUR LE MODULE DE CORRECTION DE MÉMOIRE.

- Présentation du stage.

- État de l'art.

- Études de l'existant et solution envisagée.

## Partie 2 : APPROCHE DU PROJET.

- Démarche du projet.

- Exigences du projet.

- Architecture et conception du système logiciel.

- Tests du système logiciel.



## **2. Présentation du stage**

### **2.1 Présentation de l'entreprise**

Intégré au sein de l'ITUniversity depuis 2014, ITU-Labs est un laboratoire d'idées qui va permettre aux étudiants de contribuer aux développements de Madagascar.

De nombreuses accessoires sont disponibles pour les étudiants:

- Caméra
- Divers capteurs
- Prise intelligente
- Ecran tactile
- Kit IOT, Internet of things
- Smartphone
- Raspberry Pi

Autour de ces nouveaux matériels à bas coût et miniaturisé, de projet social (MAS-LOW), de projet innovant (SMART FRIDGE, SMART CAR) pourront être appliquer directement.

Avec ses partenaires issues du secteur privée et du secteur public, ITULABS souhaite réduire le temps entre le prototypage et la mise sur le marché des produits issus de la recherche.

Une collaboration plus étroite sera mise en place avec les entreprises. Ainsi, elles pourront mettre en œuvre avec les étudiants des idées issus de leur connaissance des besoins du marché et de la clientèle.

### **2.2 Présentation du sujet et objectifs du projet**

#### **2.2.1 Sujet et contexte générale**

La rédaction de mémoire est une étape importante dans la vie estudiantine car elle conditionne la qualité de sa soutenance et ainsi l'obtention du diplôme. Dans le but de minimiser les risques d'erreur de la part de l'étudiant lors de sa rédaction, un encadreur lui est affecté. Ce dernier a pour rôle de le guider et l'accompagner. Toute une organisation

doit ainsi être mise en place tout en sachant qu'un encadreur peut accompagner plusieurs étudiants, lui rendant la tâche difficile. D'où le titre du projet :

« Outil d'aide à la correction des mémoires des étudiants de l'IT-University. »

### **2.2.2 Objectifs du projet**

Étant un outil d'aide à la correction de mémoire, le premier objectif de ce projet est de faciliter le travail des encadreurs et de leur permettre un gain de temps lors de la correction des mémoires. Il servira également d'outil de communication entre étudiants et encadreurs pendant la correction.

Le second objectif de ce projet est d'aider les étudiants dans la rédaction de leur mémoire grâce aux fonctionnalités de l'outil comme la vérification des plagats sur d'autres mémoires.

Le volume de données obtenu, en raison, d'une part de l'augmentation du nombre d'étudiants et d'autre part de l'évolution des thèmes des étudiants permettra de faire une analyse et d'en dégager les informations servant à l'amélioration de la qualité de l'enseignement.

### **2.2.3 Innovation apportée**

Actuellement, l'IT-University ne dispose pas encore d'outil pour gérer efficacement la correction des mémoires. Ce projet permettra donc la centralisation des mémoires des étudiants qui facilitera l'organisation entre l'encadreur et ses étudiants. Il permettra de laisser de côté la méthode de travail par e-mail qui prend du temps et est difficile à suivre.

### **2.2.4 Principaux risques et enjeux**

Le principal risque et enjeu de ce projet est l'intégration de celui-ci dans la plateforme ITUStage qui est déjà fonctionnel. Il faut bien prendre en compte les données déjà existantes et bien assurer les migrations de celles-ci pour qu'il n'y ait pas de régression.

### **3. État de l'art sur le sujet traité**

#### **3.1 Critères de comparaison**

Les critères suivants sont étudiés afin d'établir une comparaison du projet avec les logiciels comparables :

##### **Fonctionnalités :**

La première fonctionnalité recherchée est la capacité à s'intégrer dans le système existant, c'est-à-dire en ayant la possibilité d'interagir avec les données fournies par ce dernier.

Un autre critère clé est la possibilité aux étudiants de communiquer avec leurs encadreurs et d'envoyer leurs mémoires et la possibilité aux encadreurs de notifier les étudiants d'une éventuelle correction ou la validation de leurs mémoires.

Certaines fonctionnalités liées à l'usage même de l'outil sont extrêmement utiles :

- La lecture des mémoires sans utiliser des logiciels tiers.
- La possibilité pour les encadreurs de surligner et commenter des textes ou un bloc de texte dans les mémoires envoyés par les étudiants.
- La possibilité pour les étudiants de vérifier des phrases qui existent déjà dans les autres mémoires des étudiants.
- Le téléchargement du mémoire.

Enfin, des fonctionnalités d'analyse de texte sont également importantes : l'extraction et le traitement pour réaliser des statistiques.

##### **Sécurité des données :**

La sécurité des données est un enjeu majeur et doit être l'un des critères de comparaison les plus importants. En effet, ce critère est nécessaire puisqu'il faut gérer de grandes quantités de données. Plusieurs fonctionnalités sont nécessaires à la bonne utilisation de l'outil d'aide à la correction des mémoires : le contrôle des différentes actions sur les documents ainsi que la traçabilité des modifications effectuées.

### **Usage :**

Comme tout type de logiciel, il est nécessaire d'avoir une interface simple, intuitive et ergonomique. Il faut que la prise en main se fasse le plus naturellement possible.

### **Prix :**

Le prix est un critère accessoire mais qui reste important dans une comparaison. Cependant, il ne doit pas être le critère principal. De plus, il faut bien se rendre compte que le prix ne représente pas que l'investissement d'achat, mais aussi la maintenance, la formation, et les différentes mises à jour.

## **3.2 Étude de chaque solution au vu des critères**

A ce jour, aucun outil mis en vente ou téléchargeable gratuitement ne répond simultanément à ces critères. Toutefois, il existe des solutions qui répondent à un ou quelques critères.

Les solutions suivantes ont été identifiées :

### **PDFTRON SDK<sup>1</sup>:**

#### **- Fonctionnalités :**

Ce SDK fournit une possibilité d'intégration dans tous les domaines : web, Android, iOS, Windows, linux et MacOS.

Cet outil comporte les fonctionnalités à l'usage suivantes :

- La lecture de document PDF.
- La possibilité de mettre des surlignements, des commentaires sur un texte ou un bloc de textes ou une image.
- Le téléchargement du document en cours de lecture.
- La possibilité de modifier le contenu d'un document.
- La possibilité de comparer visuellement deux documents sans déchiffrer les informations.

D'autres fonctionnalités sont comprises dans ce SDK mais ne sont pas utiles dans le contexte du projet.

---

<sup>1</sup> Software Development Kit ou Kit de développement

En termes d'analyse, ce SDK se limite à l'extraction de textes dans un document PDF sans permettre des traitements supplémentaires.

- **Sécurité des données :**

Ce SDK permet la sécurité d'un document PDF contre les accès non autorisés. Les documents PDF peuvent être cryptés à l'aide de différents systèmes de cryptage. Les gestionnaires de sécurité de l'outil effectuent l'autorisation de l'utilisateur et définissent diverses autorisations sur les documents PDF.

En matière de traçabilité, ce SDK ne permet pas de gérer l'historique des modifications effectuées sur un document.

- **Usage :**

Ce SDK fournit une interface facile à prendre en main.

- **Prix :**

PDFTRON SDK est offert gratuitement pour 30 jours d'essai et est disponible sur abonnement mensuel à partir de \$350 USD.

**PSPDFKIT:**

- **Fonctionnalités :**

Cet outil a été construite sur une pile technologique moderne et peut s'intégrer sur différentes plateformes : web, Android, iOS, Windows

Cet outil comporte les fonctionnalités à l'usage suivantes :

- La lecture de document PDF.
- La possibilité de mettre des surlignements, des commentaires sur un texte ou un bloc de textes ou une image.
- Le téléchargement du document en cours de lecture.

Tout comme PDFTRON SDK, cet outil comporte d'autres fonctionnalités mais ne sont pas utiles dans le contexte du projet.

En termes d'analyse, il se limite également à l'extraction de textes.

- **Sécurité des données :**

PSPDFKIT permet de mettre en place une authentification utilisant JWT<sup>2</sup> qui permet de contrôler l'accès à un document PDF.

En matière de traçabilité, ce SDK ne permet pas également de gérer l'historique des documents.

- **Usage :**

Ce SDK fournit une interface facile à prendre en main.

- **Prix :**

PSPDFKIT offre un essai pour 60 jours et est disponible sur abonnement mensuel à partir de \$400 USD.

### **FoxitPDF SDK :**

- **Fonctionnalités :**

Ce SDK fournit une possibilité d'intégration dans les domaines : web, Android, iOS, Windows, mac, linux

Cet outil comporte les fonctionnalités à l'usage suivantes :

- La lecture de document PDF.
- La possibilité de mettre des surlignements, des commentaires sur un texte ou un bloc de textes ou une image.
- Le téléchargement du document en cours de lecture.

En termes d'analyses, il se limite également à l'extraction de textes.

- **Sécurité des données :**

Ce SDK permet également la sécurité d'un document PDF contre les accès non autorisés. Il fournit différents types de cryptages.

Il ne permet pas la gestion des historiques des documents.

- **Usage :**

Son interface est simple, intuitive, ergonomique et facile à prendre en main.

---

<sup>2</sup> JSON Web Token

- **Prix :**

En raison de la crise sanitaire du Covid-19, la période d'essai de FoxitPDF SDK a été prolongée de 10 à 60 jours. La licence de ce SDK est de \$3000 USD par plateforme, par an.

### 3.3 Tableau comparatif des solutions au vu des critères

	Le projet	PDFTRON SDK	PSPDFKIT	FoxitPDF SDK
<b>Fonctionnalités :</b> <ul style="list-style-type: none"> <li>- Intégration dans le système existant (web)</li> <li>- Lecture</li> <li>- Surlignement, commentaires</li> <li>- Téléchargement</li> <li>- Extraction de textes</li> <li>- Traitement de textes</li> </ul>	<ul style="list-style-type: none"> <li>•</li> <li>•</li> <li>•</li> <li>•</li> <li>•</li> <li>•</li> </ul>	<ul style="list-style-type: none"> <li>•</li> <li>•</li> <li>•</li> <li>•</li> <li>•</li> <li>•</li> </ul>	<ul style="list-style-type: none"> <li>•</li> <li>•</li> <li>•</li> <li>•</li> <li>•</li> <li>•</li> </ul>	<ul style="list-style-type: none"> <li>•</li> <li>•</li> <li>•</li> <li>•</li> <li>•</li> <li>•</li> </ul>
<b>Sécurité :</b> <ul style="list-style-type: none"> <li>- Authentification</li> <li>- Gestion historique</li> </ul>	<ul style="list-style-type: none"> <li>•</li> <li>•</li> </ul>	<ul style="list-style-type: none"> <li>•</li> </ul>	<ul style="list-style-type: none"> <li>•</li> </ul>	<ul style="list-style-type: none"> <li>•</li> </ul>
Usage	Facile à prendre en main	Facile à prendre en main	Facile à prendre en main	Facile à prendre en main
Prix	Hébergement elasticsearch à partir de \$16 USD par mois	Essai 30 jours et abonnement mensuel à partir de \$350 USD	Essai 60 jours et abonnement mensuel à partir de \$400 USD	Essai 60 jours et \$3000 USD par plateforme, par an

Tableau 1 : Tableau comparatif des solutions au vu des critères



## 4. Étude de l'existant et solution envisagée

### 4.1 Étude de l'existant

#### 4.1.1 Description externe du système logiciel existant (vision utilisateur)

Le système existant à l'IT-University couvre uniquement le traitement des fiches de stages. La correction des mémoires des étudiants suit le processus suivant :

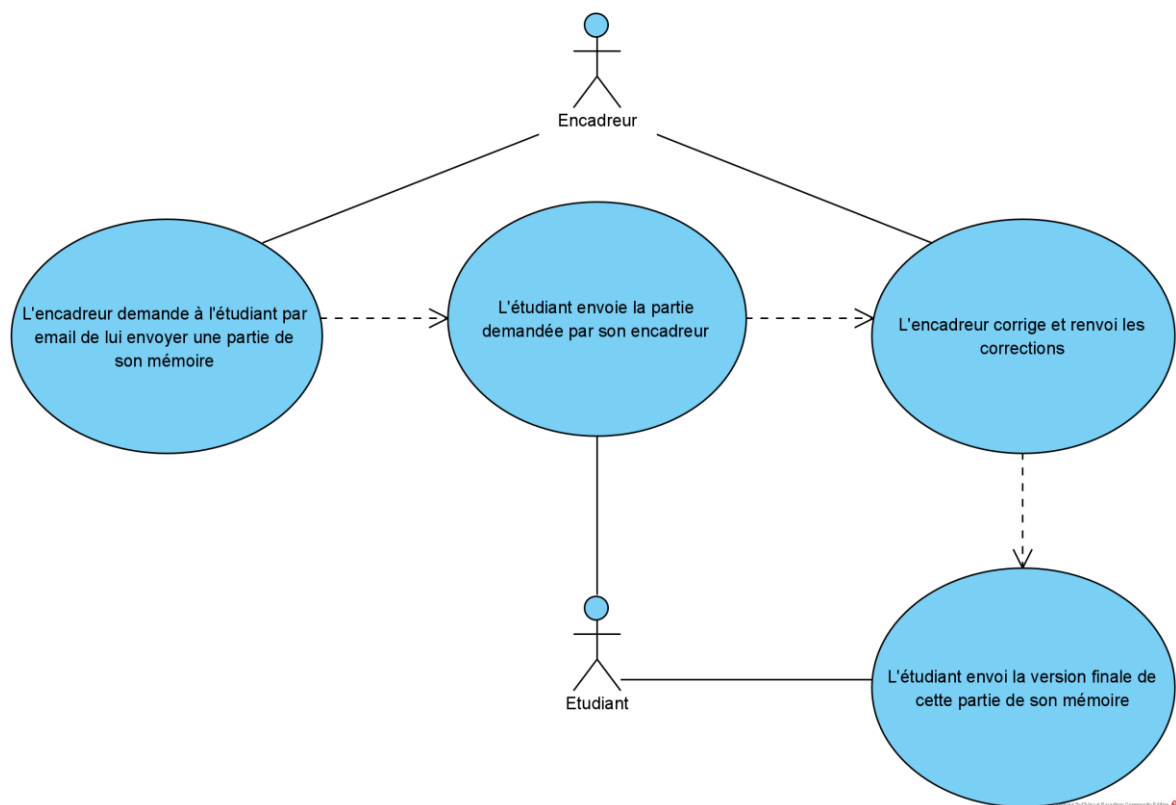


Figure 1 : Description externe du système logiciel existant.

Ce processus se divise en 4 étapes :

- Contact de l'étudiant par l'encadreur pour qu'il lui envoie une partie de son mémoire
- Envoi de la partie demandée par l'encadreur
- Correction de la partie reçue et envoi des retours à l'étudiant dans un fichier à part
- Envoi de la version finale de cette partie à l'encadreur

Ce processus est cyclique jusqu'à l'envoi du mémoire complet.

### 4.1.2 Description interne du système logiciel existant (vision développeur/conception)

L'IT-University utilise actuellement une application web qui gère les fiches de stages de chaque étudiant. Celle-ci a été développée avec PHP utilisant le framework Codeigniter, et une base de données MySQL.

Quant à la correction des mémoires, l'IT-University les traite par email.

## 4.2 Critiques de l'existant

Le système actuel permet à l'encadreur d'anticiper le contenu et l'étendu du mémoire d'un étudiant grâce à la fiche de stage qui lui renseignant le thème du projet et les tâches à réaliser durant son travail.

Néanmoins, le système existant se limite au remplissage des fiches de stages et à la validation de celles-ci. La correction, de ce fait, est lente parce que les mémoires des étudiants ne sont pas centralisés. En effet, les envois se font par email, et s'entremêlent. Ce système rend difficile la traçabilité des mémoires : à quelle fiche de stage correspond tel mémoire ?

## 4.3 Solutions envisagées

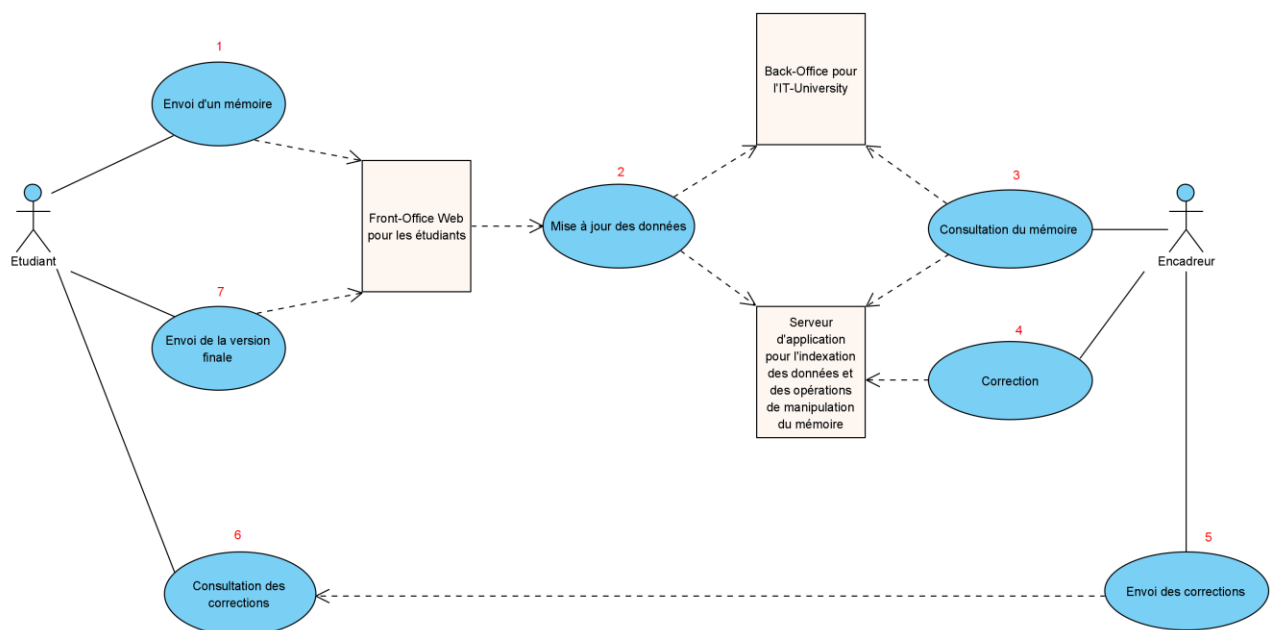


Figure 2 : Architecture de la solution envisagée

Face aux lacunes de l'existant, le présent projet permet d'établir un système pour optimiser la correction d'un mémoire.

En effet, les mémoires des étudiants sont actuellement en relation avec leurs fiches de stages respectifs, ce qui permet la centralisation de ceux-ci dans un même système. Chaque encadreur n'aura plus à se soucier de trouver les emails des étudiants contenant leurs mémoires.

De plus, chaque mémoire envoyé dans le système est indexé, permettant une grande flexibilité dans la manipulation des textes.

Les encadreurs ont aussi la possibilité de corriger les mémoires directement sur la plateforme.

## **4.4 Objectifs principaux et livrables**

### **4.4.1 Serveur d'application du projet**

C'est le serveur qui fournit tous les web services pour le traitement des mémoires exigés par le back-office et le front-office. Cette partie du projet met en place l'indexation des mémoires envoyés par les étudiants dans la base de données Elasticsearch. En effet, toute manipulation sur les mémoires se fait sur ce serveur.

Cette application propose :

- L'upload et l'indexation des mémoires
- Toutes les étapes de la correction des mémoires
- La vérification des copier-coller dans un mémoire
- Toutes autres opérations sur les mémoires comme le téléchargement

### **4.4.2 Back-Office du projet**

Le back-office du système logiciel existant consiste en l'administration des fiches de stages et la gestion des données nécessaires à celle-ci. Au cours du présent projet, ce système a été repris et d'autres fonctionnalités ont été intégrées :

- La gestion de profil utilisateurs : ajout du profil encadreur.
- L'assignation des encadreurs aux fiches de stages des étudiants.

- La notion de session sur les fiches de stages.
- La correction de mémoires.
- Notifications.

#### **4.4.3 Application web pour les étudiants**

La même application web de soumission de fiche stage a été reprise en ajoutant à celle-ci la possibilité d'envoyer les mémoires pour être corrigés et de consulter les corrections faites par l'encadreur.

Cette application permet également aux étudiants de vérifier les plagiats sur leurs mémoires avant de les soumettre aux encadreurs.

## **5. Démarche projet**

### **5.1 Principes de la démarche projet**

#### **5.1.1 Activités d'ingénierie logicielle**

Afin d'obtenir un système simple, performant et facilement maintenable, les démarches suivantes ont été appliquées :

- La spécification des exigences
- Le maquettage du système logiciel
- La conception
- Le codage et les tests
- Le déploiement

#### **5.1.2 Méthode de gestion de projet utilisé**

La méthode itérative a été appliquée tout au long du développement. Le projet est divisé en « sprints » qui correspondent chacun à un objectif. Chaque « sprint » dure en moyenne 2 semaines.

Durant chaque « sprint », des déploiements vers un serveur de test ont été réalisées afin d'avoir des retours rapides et faire les ajustements nécessaires.

D'une manière générale, la gestion du projet se fait tout en considérant les principaux objectifs, les risques et les contraintes qui y sont liés.

Une communication s'effectue de manière régulière avec le directeur de stage pour voir l'avancement du projet.

### 5.1.3 Rôles et responsabilités

Les rôles et responsabilités sont divisés en deux catégories : les parties prenantes et l'équipe réalisant le système.

- Les parties prenantes :

Les parties prenantes du projet sont le corps enseignant de l'IT-University et les étudiants.

- L'équipe réalisant le système :

Les rôles et responsabilités de l'équipe de réalisation du projet se résument comme suit :

- L'encadreur professionnel : validation et retours sur les fonctionnalités développées, et recommandations de nouvelles fonctionnalités à intégrer.
- Moi : joue le rôle de concepteur et développeur.

### 5.1.4 Outils utilisés

Pour la mise en œuvre du projet, voici les divers outils et langages utilisés :

Partie du projet	Outils utilisés	Langages et technologies utilisés
Maquettage du projet	Pencil	
Conception du projet	Visual paradigm	
Serveur d'application pour l'indexation des données et manipulation des mémoires	Visual Studio Code	Javascript, Node.js, Passenger Library
Back-Office du projet	Visual Studio Code	Php, Javascript, ReactJS, CodeIgniter, Babel, Webpack
Application web pour les étudiants	Visual Studio Code	Php, Javascript, ReactJS, CodeIgniter, Babel, Webpack

Base de données	PhpMyAdmin, Kibana	Elasticsearch, MySQL
Gestionnaire de version de code source	Gitlab	Git

Tableau 2 : Outils utilisés

#### 5.1.4.1 Node.js

Node.js est une plate-forme de développement open source pour l'exécution de code JavaScript côté serveur.

#### 5.1.4.2 Passenger Library

Passenger Library est une ressource en ligne complète sur le déploiement, l'administration, la mise à l'échelle, la haute disponibilité et plus encore de Ruby, Python et Node.js grâce à l'utilisation du serveur d'application Phusion Passenger.

#### 5.1.4.3 CodeIgniter

CodeIgniter est un framework open source écrit en PHP qui suit le motif de conception en MVC<sup>3</sup>.

#### 5.1.4.4 ReactJS

ReactJS est une bibliothèque Javascript open source pour créer des interfaces utilisateurs.

#### 5.1.4.5 Babel

Babel est un transcompilateur JavaScript gratuit et open-source qui est principalement utilisé pour convertir le code ECMAScript 2015+ en une version rétrocompatible de JavaScript.

---

<sup>3</sup> Modèle-vue-contrôleur

#### **5.1.4.6 Webpack**

Webpack est un module de regroupement qui a pour principal objectif de regrouper des fichiers JavaScript pour les utiliser dans un navigateur, mais il est également capable de transformer, de regrouper ou d'empaqueter à peu près n'importe quelle ressource.

#### **5.1.4.7 Elasticsearch**

Elasticsearch est une base de données NoSQL dont la particularité est de pouvoir indexer des documents fortement orientés textes.

#### **5.1.4.8 MySql**

La première raison du choix de MySql est sa facilité de déploiement. Bien que d'autre système de gestion de base de données relationnelles soit plus performant, comme Oracle et PostgreSQL, le moteur de stockage InnoDB qu'utilise MySql est plutôt préférable dû à sa performance dans les requêtes courtes et le grand nombre de mises à jour rapides.

### **5.1.5 Gestion de la configuration**

Pour l'organisation des livrables, ils se situent sur un même serveur linux :

- Le back office et front office : étant donné que le framework utilisé suit le patron MVC, les fichiers sont déjà bien répartis. Les « Controller » sont préfixés d'un « C\_ » et les « Models » ont le même nom que les tables.
- Le serveur d'application avec Node.js : le patron MVC a été utilisé, et les « Controller » ont suivi la règle de nommage suivant : xxxxx.controller.js, les « Models » ont également le même nom que les tables et utilisant la notation « Camel case ».

Les codes sources ont été placés sur Gitlab pour l'organisation des versions.

Une procédure de sauvegarde de données (backup SQL et backup des fichiers) du système existant sera mise en place ainsi qu'une procédure de déploiement sur l'environnement de production.



## 5.2 Contraintes et risques sur le projet

N°	Libelle du risque	Facteur contribuant	Solution proposée	STATUT (en cours, réalisé, solution choisie)
1	Risque de régression sur le système logiciel existant.	Modification des codes existants.	Mise en place des tests unitaires sur le projet existant.	En cours.
2	Pertes des données sur le système logiciel existant.	Modification de la structure de la base de données.	Mise en place de la gestion de migration.	Solution proposée en cours.

Tableau 3 : Contraintes et risques sur le projet

## 5.3 Démarche du projet : mise en œuvre

Chaque sprint dure deux semaines sauf pour le sprint 0 qui dure une semaine.

Sprints	Objectifs	Taches
Sprint 0	Prise en main sur Elasticsearch.	
Sprint 1	Upload d'un mémoire et possibilité de vérifier les copier-coller.	<p>En tant qu'étudiant, je peux :</p> <ul style="list-style-type: none"> <li>- uploader mon mémoire et vérifier les plagats avant de l'envoyer à mon encadreur.</li> <li>- également supprimer ce mémoire s'il n'est</li> </ul>

		pas encore envoyé.
Sprint 2	Gestion de profil et mise en place de la correction.	<p>En tant qu'administrateur, je peux :</p> <ul style="list-style-type: none"> <li>- attribuer le statut d'encadreur à un utilisateur.</li> <li>- assigner des fiches stages à des encadreurs.</li> </ul> <p>En tant qu'encadreur :</p> <ul style="list-style-type: none"> <li>- Je peux voir les mémoires envoyés par les étudiants qui me sont assignés.</li> </ul> <p>En tant qu'étudiant :</p> <ul style="list-style-type: none"> <li>- Je peux envoyer mes mémoires à mon encadreur.</li> </ul>
Sprint 3	Finalisation du module de correction, ajout de notion de session sur les stages et export Excel des fiches de stages.	<p>En tant qu'encadreur ou administrateur, j'ai la possibilité de :</p> <ul style="list-style-type: none"> <li>- surligner les textes et images sur les mémoires.</li> <li>- envoyer mes corrections aux étudiants.</li> <li>- exporter les fiches de stages sous</li> </ul>

		<p>format Excel.</p> <p>En tant qu'étudiant, je peux :</p> <ul style="list-style-type: none"> <li>- voir les corrections apportées par mon encadreur.</li> <li>- choisir la session de soutenance lors de la soumission de la fiche de stage.</li> </ul>
Sprint 4	Téléchargement des mémoires, notification et mise à jour automatique lors des corrections.	<p>En tant qu'utilisateur :</p> <ul style="list-style-type: none"> <li>- Je peux télécharger les mémoires et lire hors connexion les surlignements sur les textes et images.</li> <li>- J'aimerais voir des notifications s'il y a des modifications faites par les étudiants ou encadreurs.</li> </ul> <p>En tant qu'encadreur :</p> <ul style="list-style-type: none"> <li>- Je peux reprendre mes corrections et revoir mes surlignements à l'état où je les ai laissés.</li> </ul>

Tableau 4 : Démarche du projet

## 5.4 Planification

Le planning du présent projet se résume comme suit :

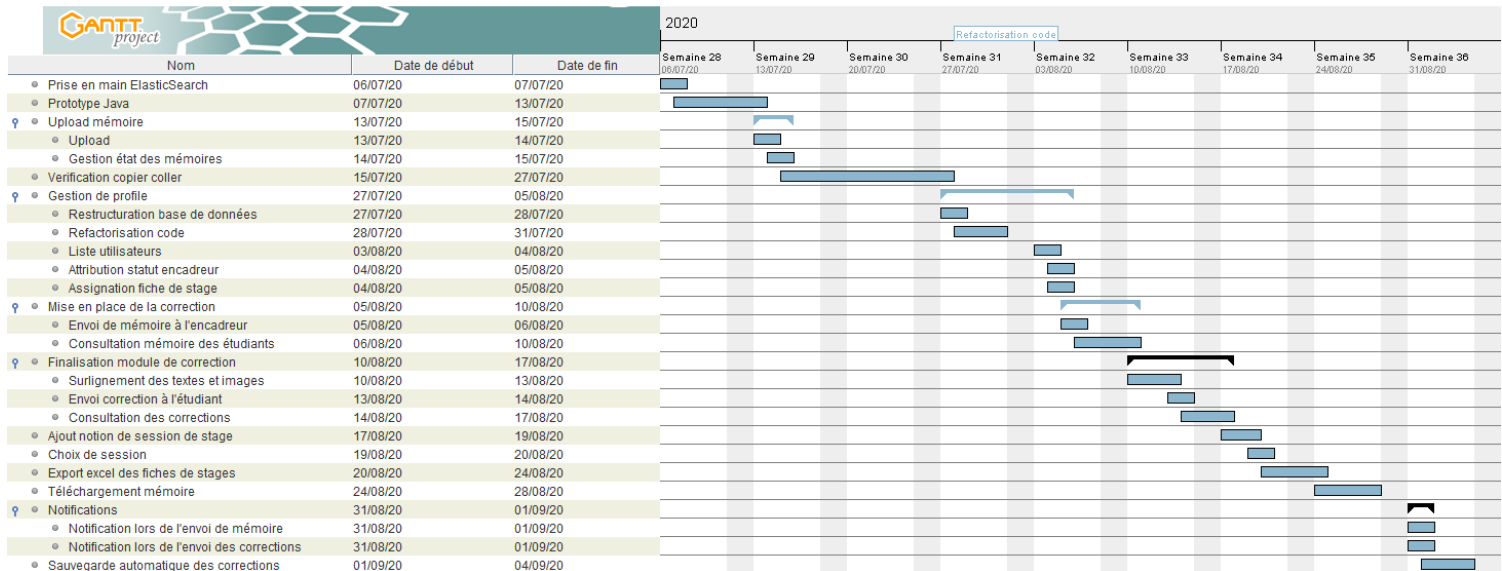


Figure 3 : Planification du projet (Diagramme de GANTT)

Le projet s'est étalé sur 65 jours qui s'est réparti sur cinq sprints.

Le début a été marqué par une prise en main sur ElasticSearch sur Java, qui par la suite a été remplacé par Nodejs. Sur chaque sprint, une analyse et conception a été nécessaire afin d'intégrer les nouvelles fonctionnalités sans impacter le fonctionnement de l'existant. Des déploiements sur le serveur de test ont été également réalisés pour que l'encadreur puisse suivre l'avancement du projet et donner ses retours, qui ont été ensuite pris en compte dans le même sprint.

## 5.5 Budget du projet

Le tableau ci-dessous montre toutes les charges de départ occasionnées par le présent projet :

Désignation	Unité	Prix unitaire (MGA)	Total (MGA)	Total (EUR <sup>4</sup> )
Ordinateur	1 unité	3 500 000	3 500 000	777,03
Connexion internet	3 mois	257 000	771 000	171,20
		<b>Total</b>	<b>4 271 000</b>	<b>948,23</b>

Tableau 5 : Budget du projet

---

<sup>4</sup> 1 EUR = 4 504 MGA

## 6. Exigences réalisées dans le projet (vision externe / utilisateur)

### 6.1 Exigences fonctionnelles – Cas d'utilisation

#### 6.1.1 Cas d'utilisation – Upload de mémoire

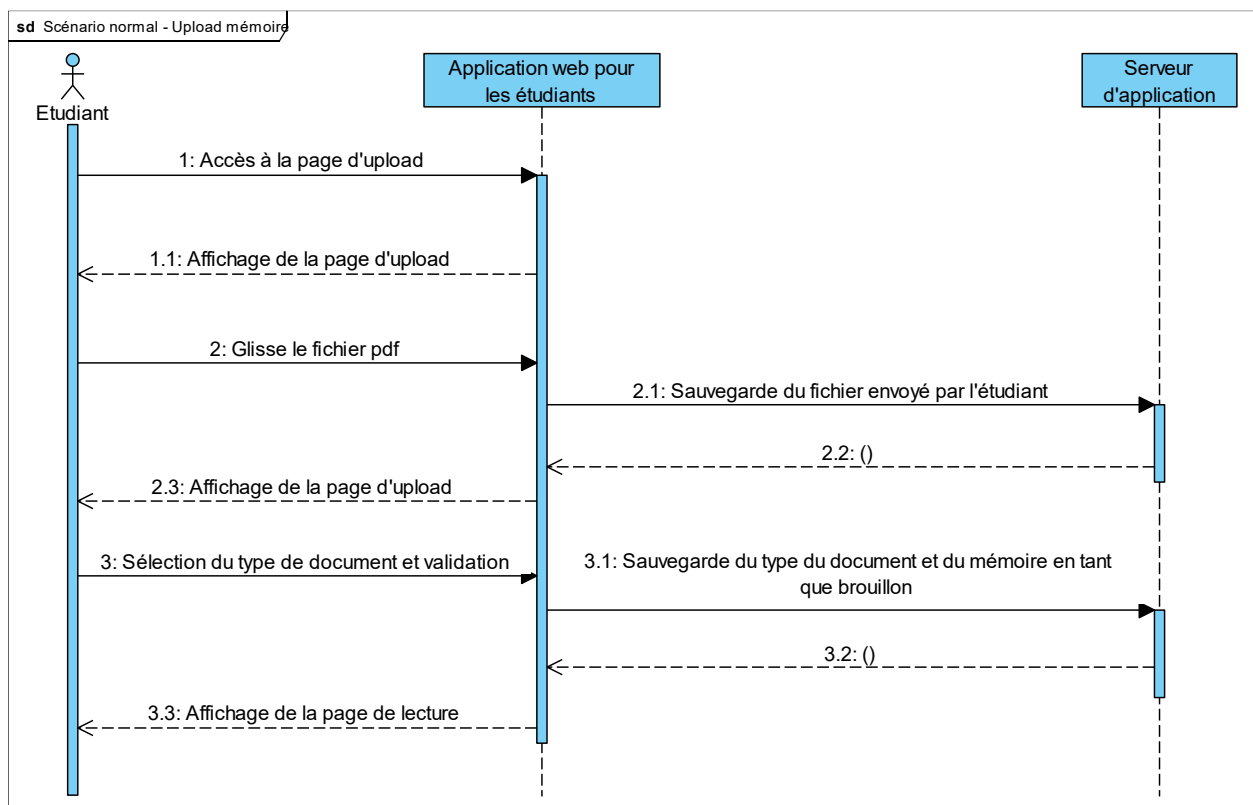


Figure 4 : Cas d'utilisation - Upload mémoire

Dans ce cas d'utilisation, l'étudiant est l'acteur. Après s'être connecté sur l'application web, l'étudiant se rend sur la page d'upload. Il va ensuite glisser son fichier pdf sur la zone d'upload et celui-ci va être directement sauvegardé sur le serveur. L'étudiant choisit ensuite le type du document qu'il vient d'uploader, par exemple, c'est un « Mémoire complet » ou simplement le « Plan » et en validant, son mémoire sera enregistré en tant que brouillon et il sera redirigé vers la page de lecture.

## 6.1.2 Cas d'utilisation – Vérification des copier-coller

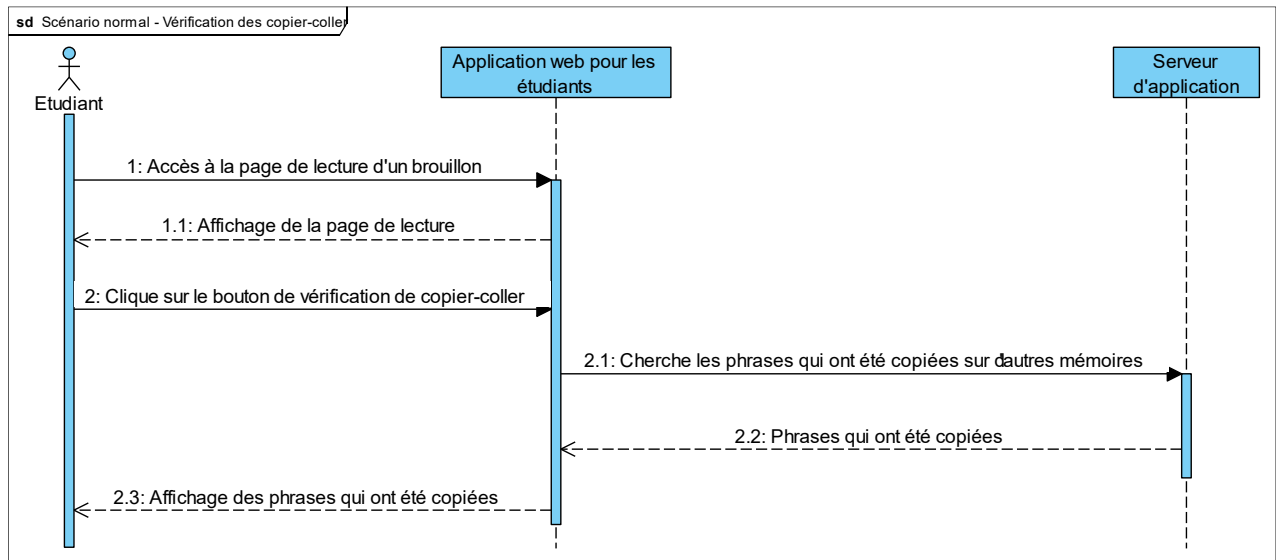


Figure 5 : Cas d'utilisation - Vérification des copier-coller

Dans ce cas d'utilisation, l'étudiant est l'acteur. Après s'être connecté sur l'application web, l'étudiant se rend sur la page de lecture d'un de ses brouillons. Il peut cliquer sur le bouton qui permet de détecter si son mémoire contient des phrases plagiées. En cliquant sur ce bouton, le serveur va vérifier l'existence de telles phrases dans la bibliothèque de mémoire de l'IT-University. S'il détecte de tels cas, ces phrases seront affichées sur la page de lecture.

### 6.1.3 Cas d'utilisation – Envoi d'un mémoire à l'encadreur

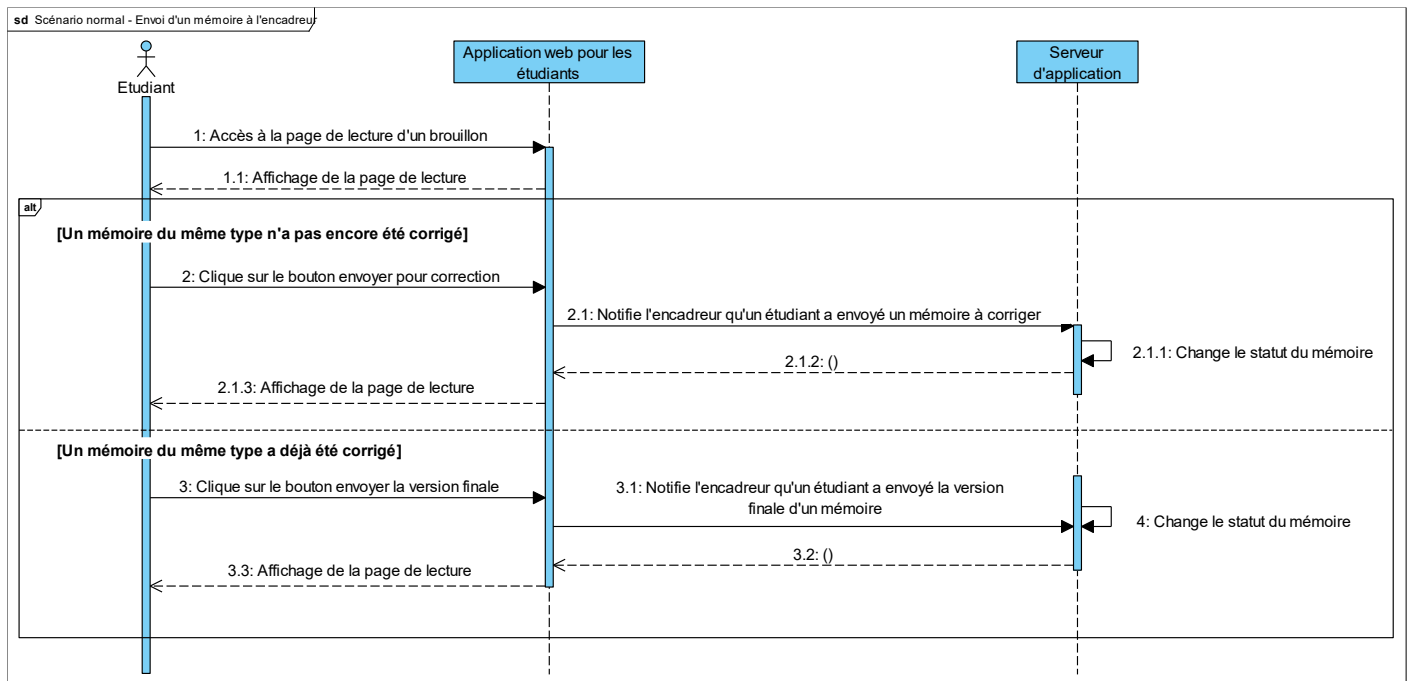


Figure 6 : Cas d'utilisation - Envoi d'un mémoire à l'encadreur

Dans ce cas d'utilisation, l'étudiant est l'acteur. Après s'être connecté sur l'application web, l'étudiant se rend sur la page de lecture d'un de ses brouillons. Deux cas possibles se présentent :

- Un mémoire du même type n'a pas encore été corrigé : en cliquant sur le bouton « Envoyer pour correction », le serveur notifie l'encadreur par email qu'un étudiant a envoyé son mémoire pour être corrigé et précise le type du document. Ensuite, le serveur change le statut du mémoire.
- Un mémoire du même type a déjà été corrigé : même scénario que le précédent cas mais le bouton est remplacé par « Envoyer la version finale » et la notification que reçoit l'encadreur change également dans le même sens.



## 6.1.4 Cas d'utilisation – Correction d'un mémoire

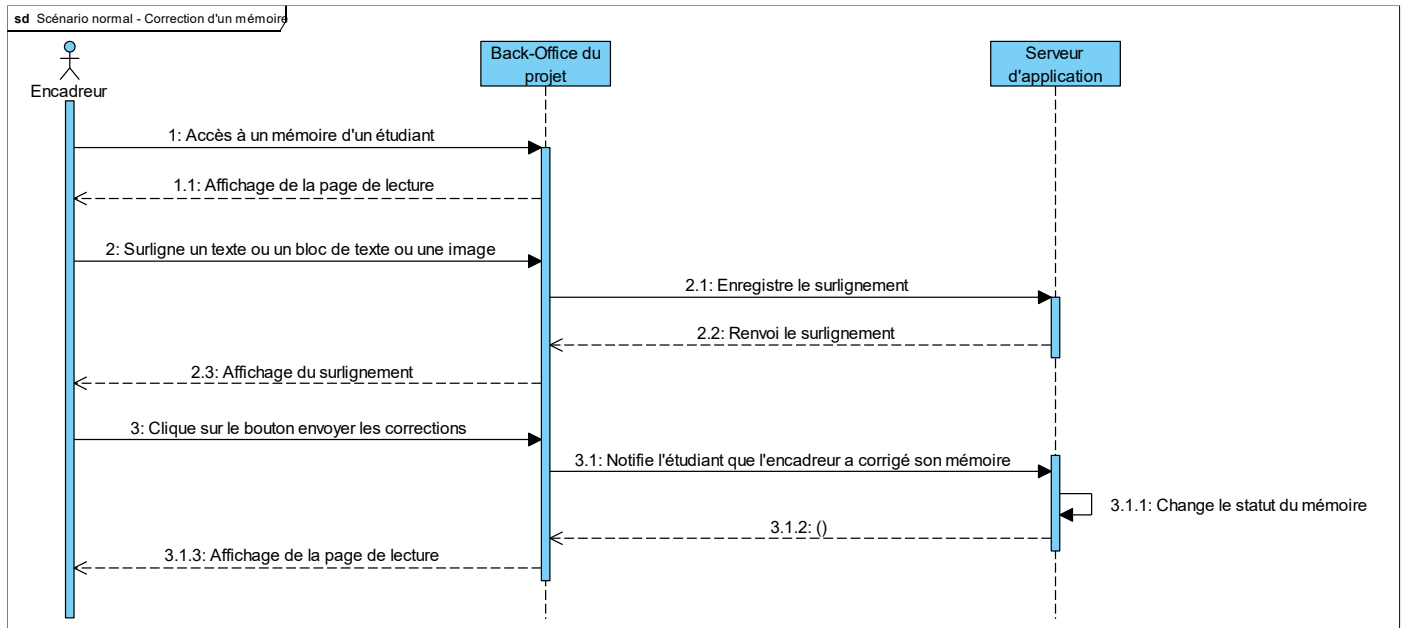


Figure 7 : Cas d'utilisation - Correction d'un mémoire

Dans ce cas d'utilisation, l'encadreur est l'acteur. Après s'être connecté sur le back-office du projet, l'encadreur se rend sur la page de lecture d'un mémoire d'un étudiant. Il a la possibilité de surligner un texte, un bloc de texte ou une image, qui est directement enregistré par le serveur.

En cliquant sur le bouton « Envoyer les corrections », le serveur notifie l'étudiant par email que son encadreur a corrigé son mémoire et statut du mémoire change également.

## **6.2 Exigences non fonctionnelles transverses**

### **Utilisabilité**

- L'interface graphique est simple à utiliser.
- Les utilisateurs peuvent effectuer ses actions rapidement et de manière intuitive.

### **Performance**

- L'utilisation de la base de données « Elasticsearch » procure une meilleure performance sur les traitements des mémoires, puisque la particularité de celle-ci est de pouvoir indexer des documents fortement orientés textes.

### **Robustesse et sécurité**

- Une authentification est requise afin d'accéder au site web.
- Les requêtes des applications web vers le serveur sont chiffrés grâce au protocole HTTPS. En plus, lors de l'authentification de l'utilisateur, le serveur génère un « Json Web Token » qui sera ensuite utilisé à chaque requête vers le serveur.

### **Supportabilité**

- L'utilisation de l'architecture MVC permet de mieux séparer les différentes couches de l'application, engendrant une facilité de maintenance.
- Les codes sources sont versionnés.

## 6.3 Interfaces détaillées

### 6.3.1 IHM

Cette partie abordera quelques affichages illustrant les cas d'utilisations du projet.

#### 6.3.1.1 Cas d'utilisation – Upload de mémoire

La page d'upload apparaît comme ci-après :

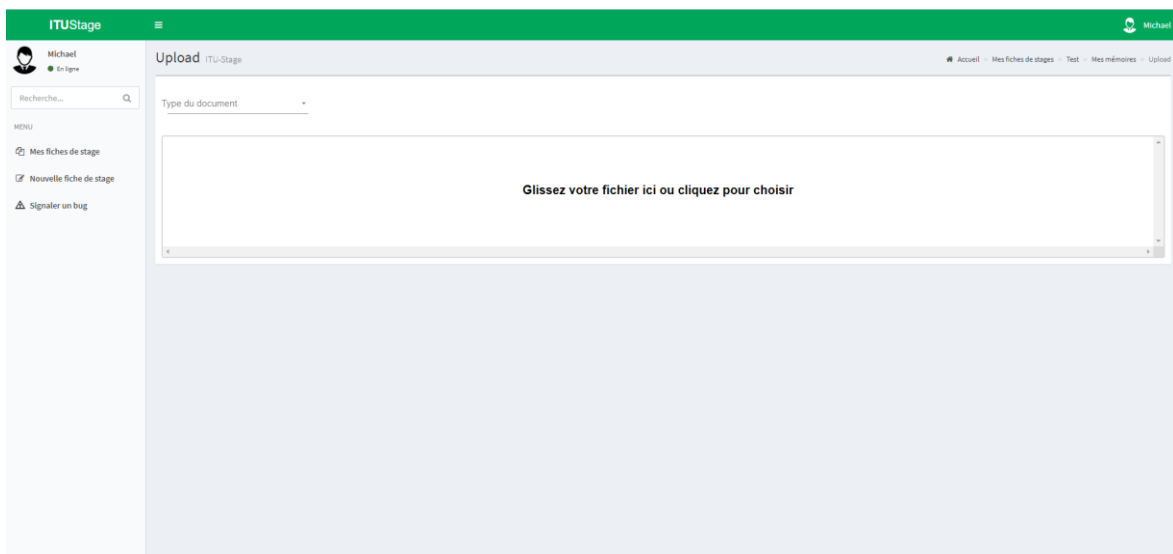


Figure 8 : Page d'upload de mémoire

En glissant le fichier pdf sur la zone d'upload, nous obtenons l'affichage suivante :

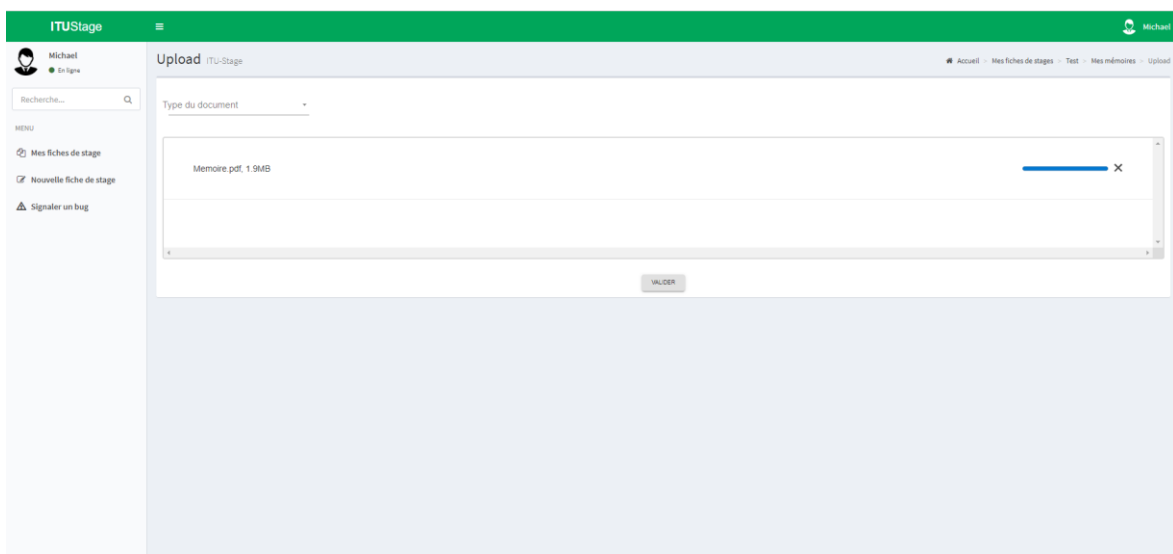


Figure 9 : Page d'upload mémoire après sélection d'un fichier

En sélectionnant un type de document et en cliquant sur le bouton valider, l'étudiant sera redirigé vers la page de lecture.

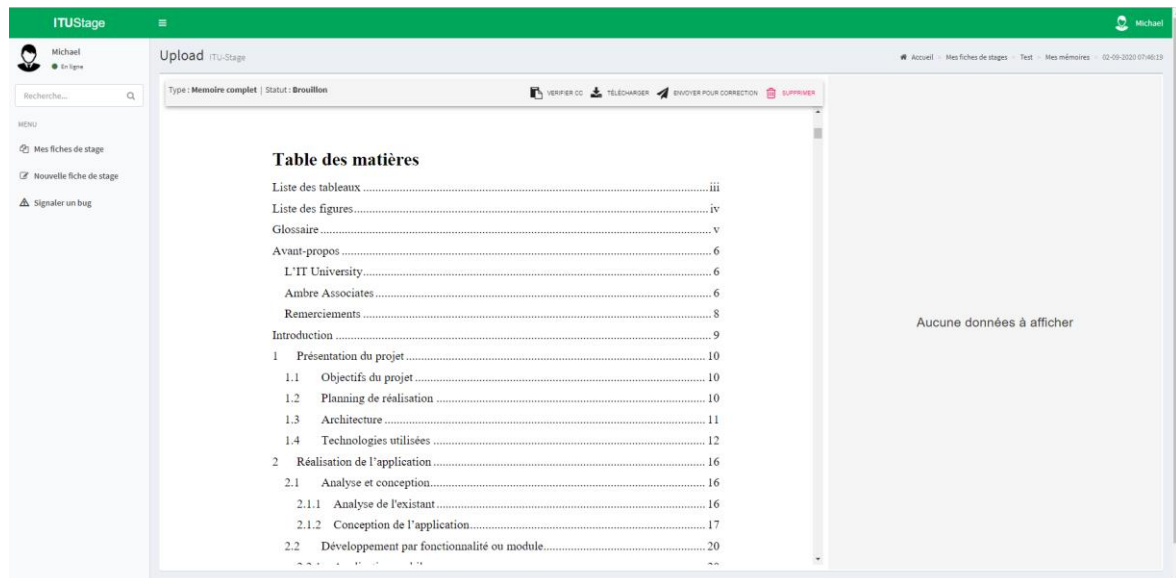


Figure 10 : Page de lecture

### 6.3.1.2 Cas d'utilisation – Vérification des copier-coller

En cliquant sur le bouton « Vérifier CC », les phrases qui ont été copiées seront affichées comme suit :

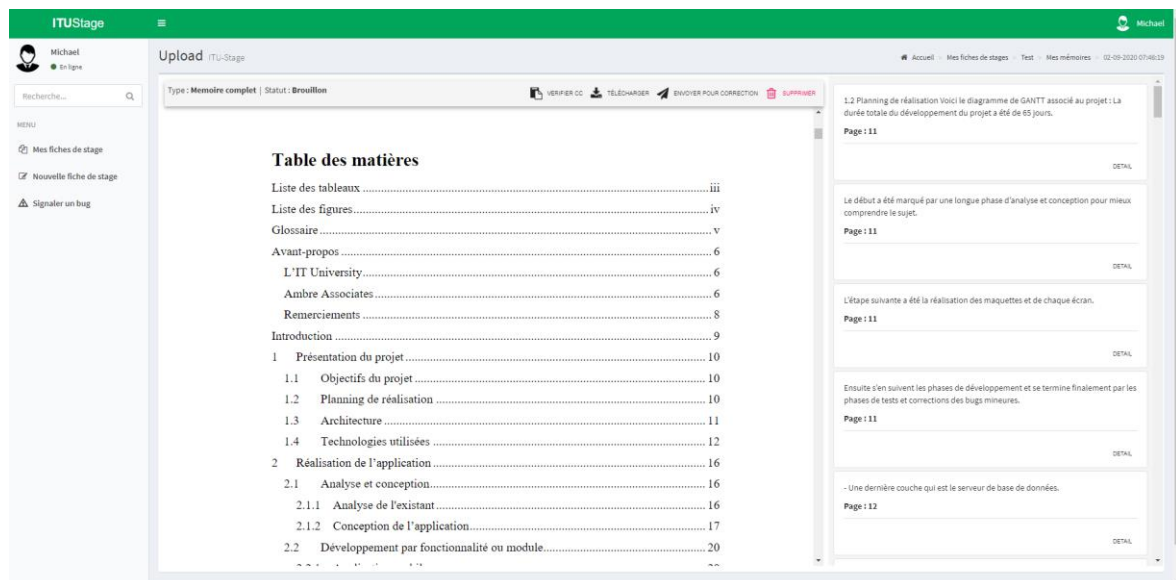


Figure 11 : Affichage des phrases copiées

### 6.3.1.3 Cas d'utilisation – Envoi d'un mémoire à l'encadreur

Deux possibilités se présentent lors de l'envoi d'un mémoire à l'encadreur :

- Un mémoire du même type n'a pas encore été corrigé

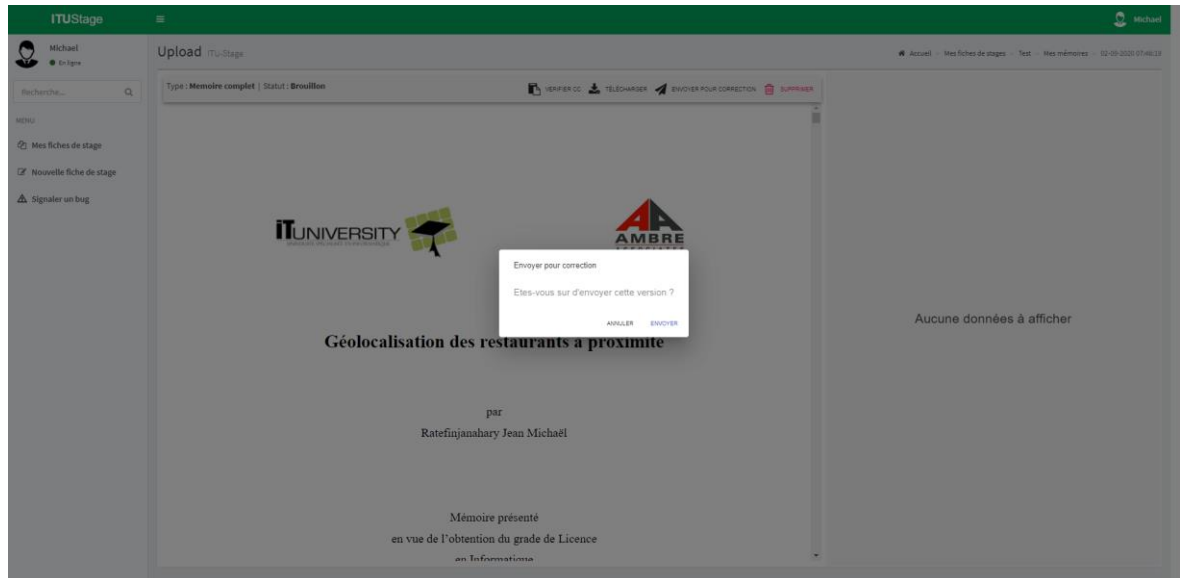


Figure 12 : Envoi d'un mémoire pas encore corrigé

- Un mémoire du même type a déjà été corrigé

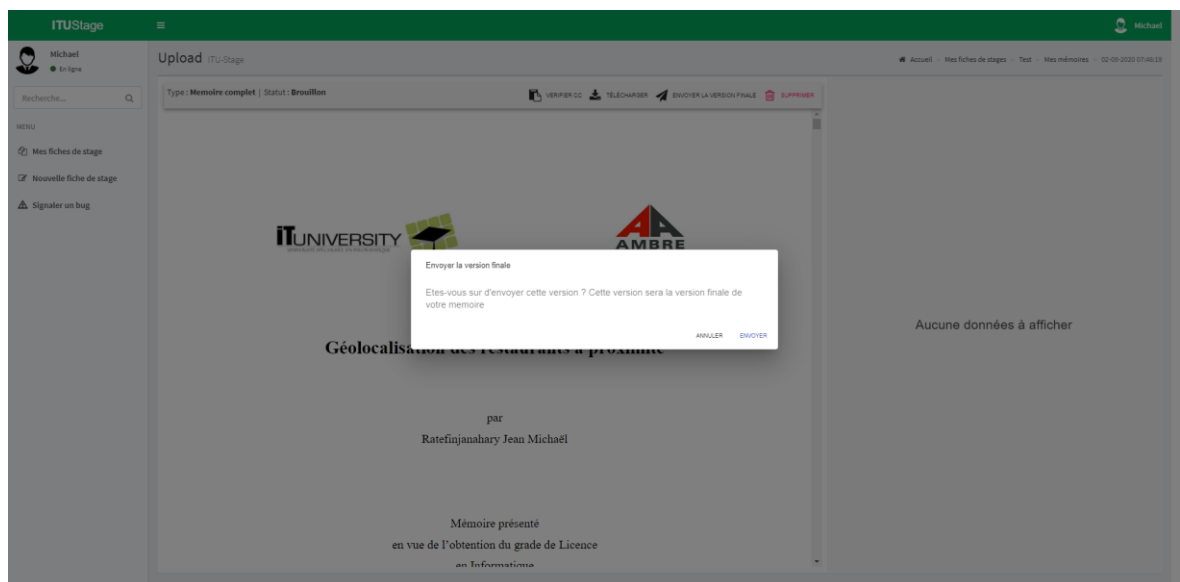


Figure 13 : Envoi d'un mémoire du même type déjà corrigé précédemment

### 6.3.1.4 Cas d'utilisation – Correction d'un mémoire

L'encadreur a la possibilité de surligner des textes, des images comme le montre la figure suivante :

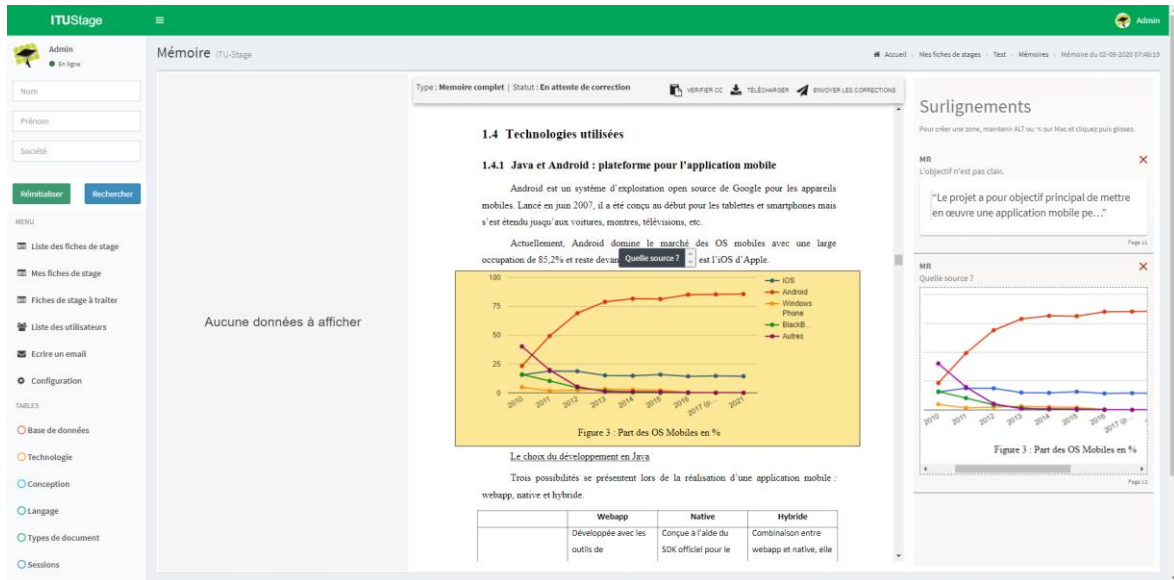


Figure 14 : Correction d'un mémoire – Surlignements

Les surlignements faites par l'encadreur sont visibles sur une liste à droite.

En cliquant sur le bouton « Envoyer pour correction », l'encadreur est invité à confirmer sa décision :

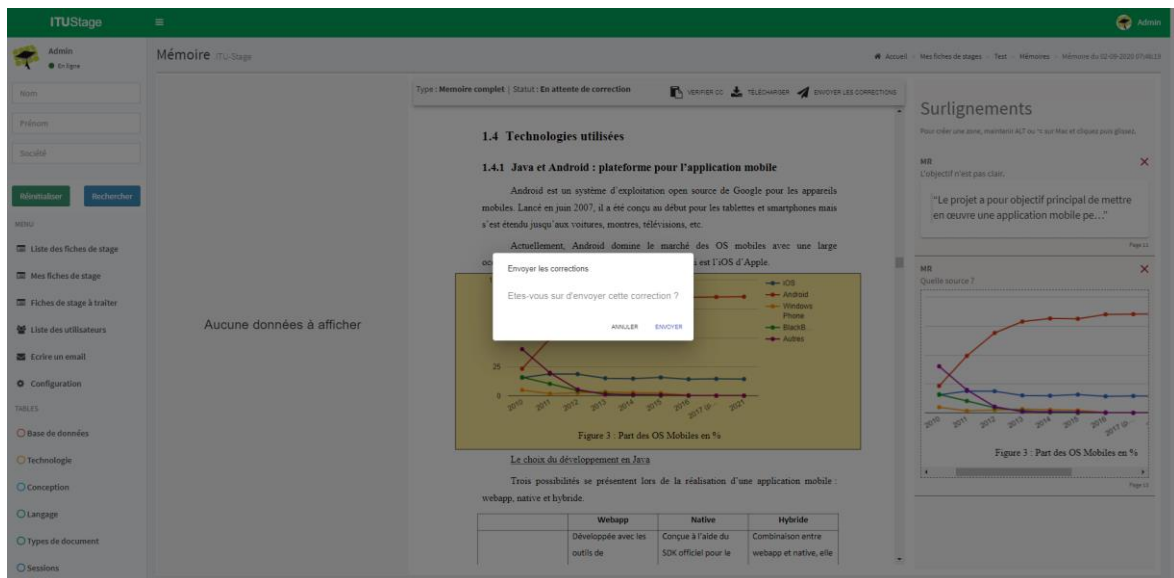


Figure 15 : Envoi des corrections

### 6.3.2 Interfaces avec d'autres systèmes

Schemes	HTTPS	Authorize
memoire		
POST	/memoire	Ajout d'un nouveau mémoire
POST	/memoire/upload	Upload le fichier pdf
GET	/memoire/{memoireId}	Cherche un memoire par son id
DELETE	/memoire/{memoireId}	Supprime un memoire à l'état brouillon
GET	/memoire/{memoireId}/pdf	Cherche le fichier pdf d'un memoire par son id
GET	/memoire/{memoireId}/copypaste	Vérifie les copier-coller dans un memoire
POST	/memoire/{memoireId}/publish	Envoi le memoire à l'encadreur
POST	/memoire/{memoireId}/correction	Envoi les corrections à l'étudiant
POST	/memoire/{memoireId}/highlights	Ajout de surlignement sur un memoire
POST	/memoire/{memoireId}/highlights/areaupdate	Modifie un bloc de surlignement
POST	/memoire/{memoireId}/highlights/remove	Supprime un surlignement
GET	/memoire/{memoireId}/download	Télécharge un memoire

Figure 16 : Description des API

Cette figure illustre les différentes apis sur le traitement des mémoires.

## 7. Architecture(s) système

### 7.1 Architecture logicielle

Le pattern MVC est utilisé sur les différents livrables du projet. En effet, ce type de structuration permet aisément à n'importe qui de reprendre en main le projet mais cela suit également les bonnes pratiques car on isole les éléments selon leur nature.

### 7.2 Architecture technique

Une architecture n-tiers a été adoptée :

- Un serveur de base de données MySQL qui se trouve sur le serveur.
- Une base de données Elasticsearch qui se trouve sur le cloud.
- Un serveur d'application web Phusion Passenger qui permet de lancer Node.js qui se situe également sur le serveur. Node.js communique avec la base de données MySQL et Elasticsearch.
- Un serveur d'application web qui permet de déployer les deux applications web : le back-office et le front-office, ces applications imbriquent avec eux la version compilée de ReactJS. Elles se communiquent directement avec la base de données MySQL.

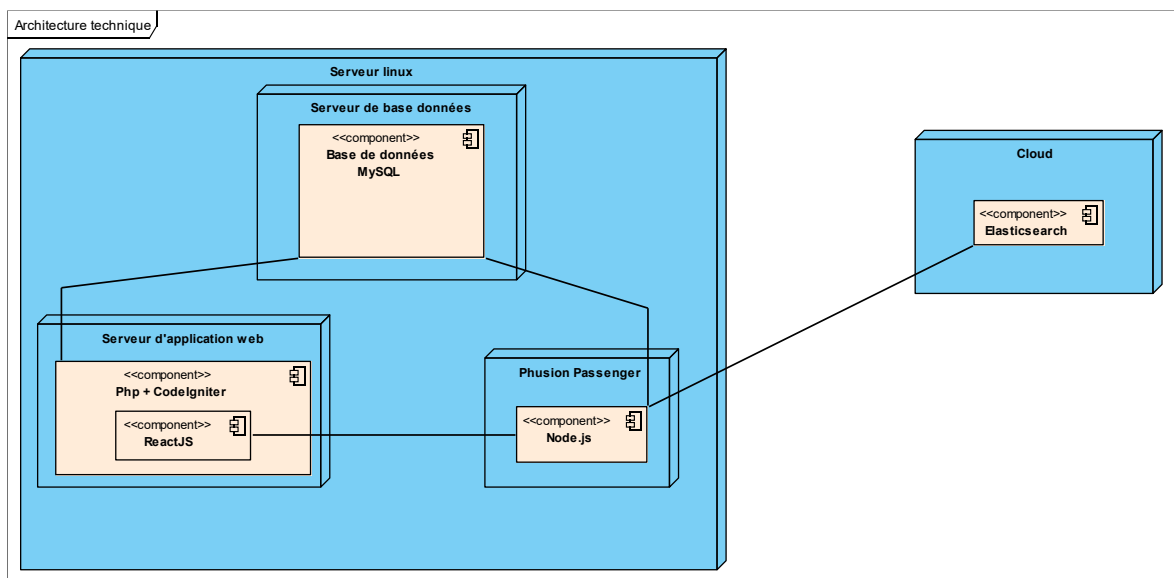


Figure 17 : Architecture technique



## 8. Conception du système logiciel réalisée dans le projet (vision interne/développeur)

### 8.1 Conception du logiciel développé

#### 8.1.1 Conception du code source

Les fichiers sont organisés suivant le pattern MVC. Tous les modèles se trouvent dans le package « models »; les contrôleurs se situent dans le package « controller » et les vues dans le package « views ».

Les noms des classes du package « models » sont équivalents à ceux des tables afin de faciliter la généralisation et la gestion du code.

A part les modèles, les noms des autres sont préfixés par l'initial du package, par exemple, pour un contrôleur, la classe se nomme de manière C\_XXXXX du côté PHP et du côté Node.js les fichiers se nomment comme suit : xxxxx.controller.js.

#### 8.1.2 Le code source – vue statique

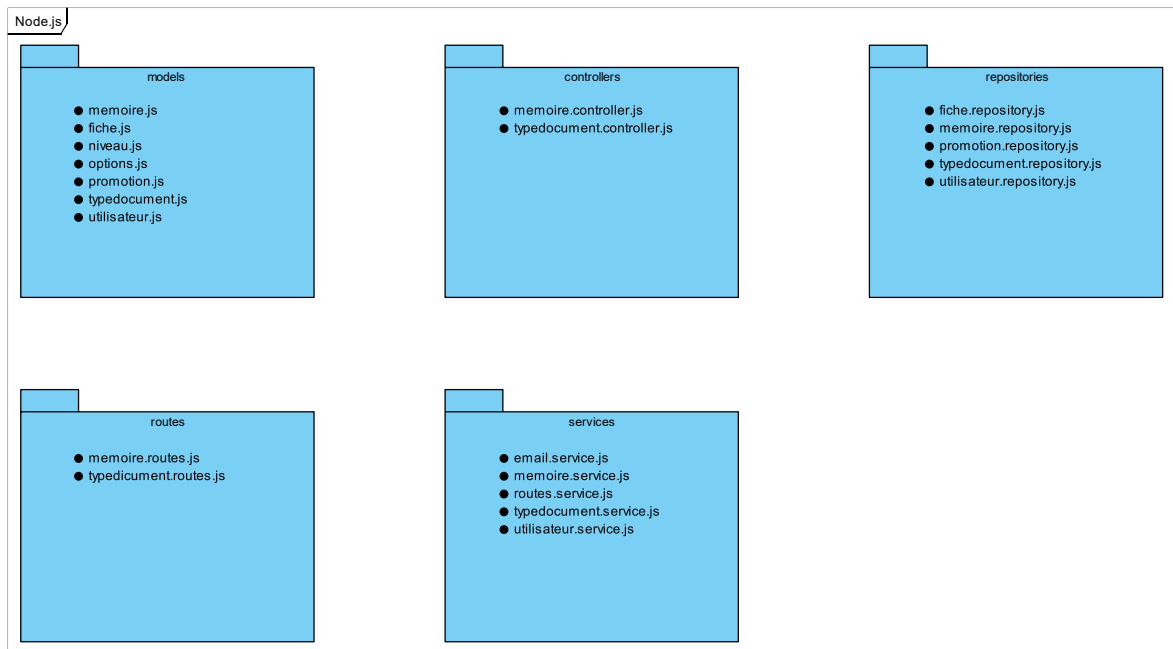


Figure 18 : Diagramme de structuration interne des packages (Node.js)

Pour chaque « package », la structuration en classes/fichiers sont en « **Annexe 1 : Structuration en classes/fichiers** »

### 8.1.3 Modélisation de données

Le système logiciel existant déjà, uniquement les modifications apportées seront illustrées dans la figure de la structure MPD MySQL ci-après :

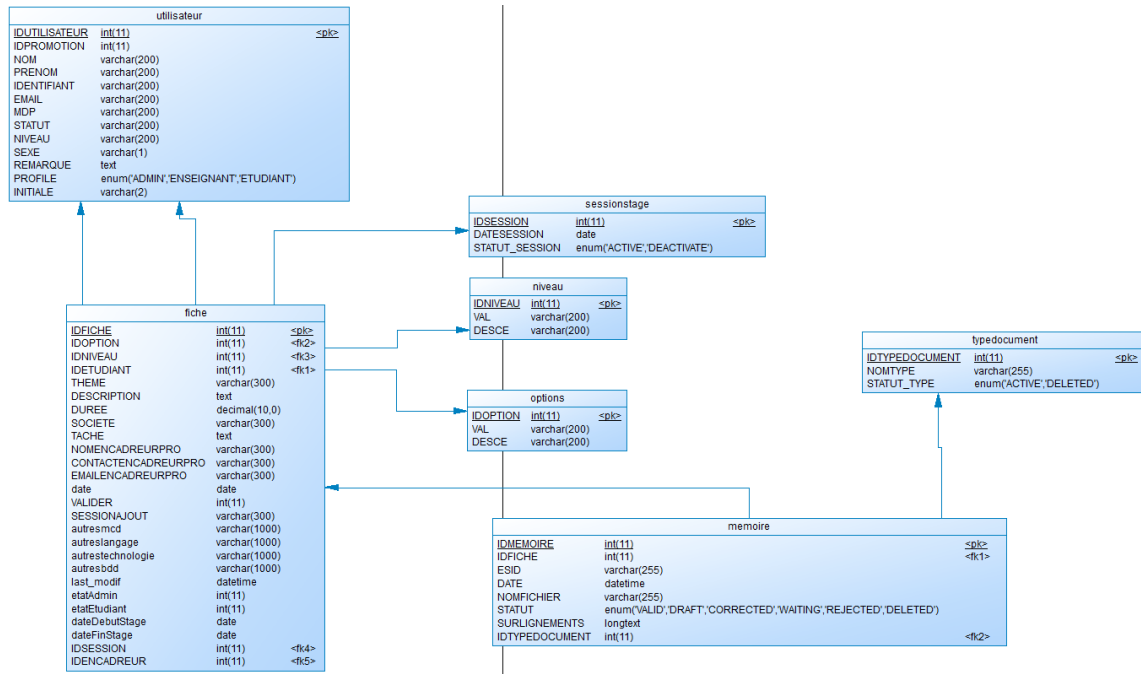


Figure 19 : MPD de la base de données MySQL

Pour la partie Elasticsearch, il ne comporte qu'une seule index « mémoire » qui regroupe les textes dans le pdf du mémoire.

## 8.1.4 Réalisation des cas d'utilisation

Cette partie montre le cas d'utilisation « Vérification des copier-coller » :

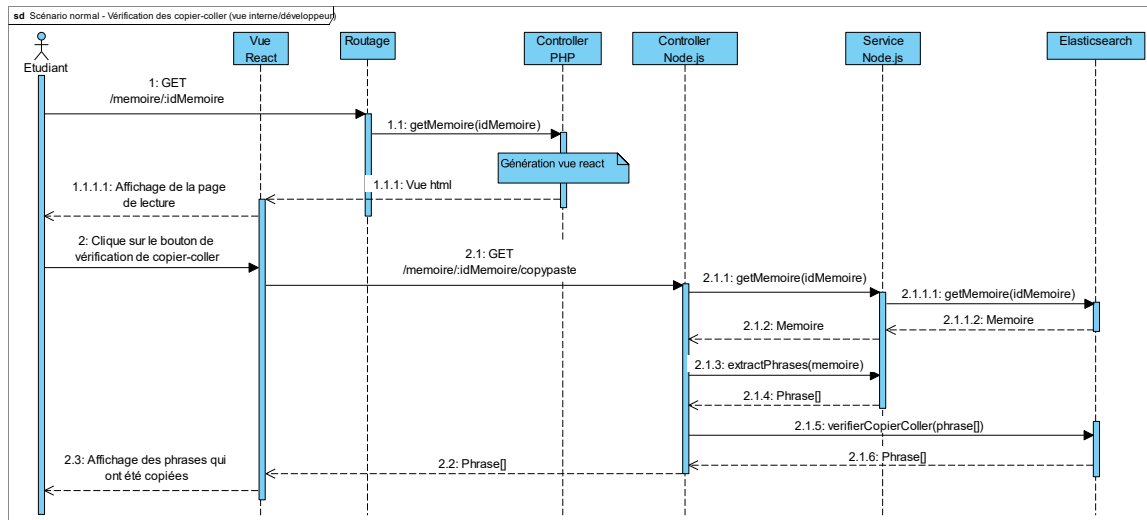


Figure 20 : Cas d'utilisation - Vérification des copier-coller (Vue interne/développeur)

## 8.1.5 Les composants et leur déploiement

Les site web BackOffice et FrontOffice se trouvent dans un même conteneur web Apache et la partie Node.js dans le serveur d'application web Phusion Passenger. La base données Elasticsearch est déployée sur le cloud.

## 9. Tests du système logiciel

### Test fonctionnels :

Un système d'automatisation des tests fonctionnels n'a pas encore été mise en place. L'encadreur professionnel s'est chargé de tester l'application à chaque déploiement.

### Tests unitaires :

Des outils comme « Postman » ont été utilisés pour tester chaque différent cas sur les web services fournis et « Kibana » sur l'intégrité des données dans la base de données ElasticSearch.

La mise en place des tests unitaires est encore en cours. Ces tests seront intégrés de sorte qu'après chaque push vers git, les tests unitaires sont lancés.

## 10. Conclusion générale

Le projet avait pour objectif d'optimiser le système de gestion et correction des mémoires des étudiants de l'IT-University. Si un nouveau système était le but au début du projet, des propositions ont finalement été travaillées pour améliorer le système actuel.

Au titre du projet, 2 nouveaux programmes et modifications sur un programme existant ont été réalisés pour un total de 71 classes et approximativement 10000 lignes de codes. Les objectifs principaux ont été réalisés et le projet peut être mis en service. Néanmoins, les livrables sont déployés dans un serveur de test et sont achevés à 90%.

Le projet ne s'est pas déroulé sans problèmes. En effet, des difficultés ont été rencontrées au début car le projet a été pensé et commencé à être développé sous Java et React. Or, au cours d'un test de déploiement du premier sprint, il a été constaté qu'il était préférable de lier le projet et le système logiciel existant de l'IT-University. Le projet s'est ainsi heurté à une nouvelle problématique : comment lier les deux projets ? La solution a été d'utiliser la base de données du système existant. Mais une application en Java ne pouvait pas être déployé sur le serveur. Il a été décidé de garder React en utilisant Webpack pour le compiler en un seul fichier Javascript car il fallait le lier avec les vues dans PHP CodeIgniter. Enfin, il a été décidé de remplacer Java en Node.js, qui était disponible sur le serveur.

Les perspectives d'amélioration du projet seraient de le terminer à 100%. La dernière phase consiste à implémenter des fonctions permettant de réaliser des statistiques sur les mémoires des étudiants, en profitant des fonctionnalités offertes par Elasticsearch.

Le projet a été l'occasion pour moi de découvrir de nouvelles technologies et de renforcer mes compétences. Il m'a également permis de contribuer au développement de mon université qui profitera aux enseignants et aux étudiants.

## 11. Références et Bibliographie

- [1] Paul Nelson, « Natural Language Processing (NLP) Techniques for Extracting Information », <https://www.searchtechnologies.com/blog/natural-language-processing-techniques>
- [2] Ekaterina Novoseltseva, « Elasticsearch : Advantages, Case Studies, And Stats », 26 Mai 2017, <https://dzone.com/articles/elastic-search-advantages-case-studies-amp-books>
- [3] H.P. Edmundson, « New Methods in Automatic Extracting », 1969
- [4] Jeffrey C. Reynar, « Topic Segmentation : Algorithms and Applications », 1998
- [5] O'Neil, John, « Doing Things with Words, Part Two : Sentence Boundary Detection », 03 Janvier 2009
- [6] « Elasticsearch Introduction Node.js », <https://www.elastic.co/guide/en/elasticsearch/client/javascript-api/current/introduction.html>
- [7] « Quickstart : Node.js + Phusion Passenger », <https://www.phusionpassenger.com/library/walkthroughs/start/nodejs.html>

## 12. Annexes

### Annexe 1 : Structuration en classes/fichiers

- Le package « models »

Le package « models » est présent sur la partie Node.js et PHP. La figure ci-après illustre les classes qui ont été modifiées/ajoutées sur le projet.

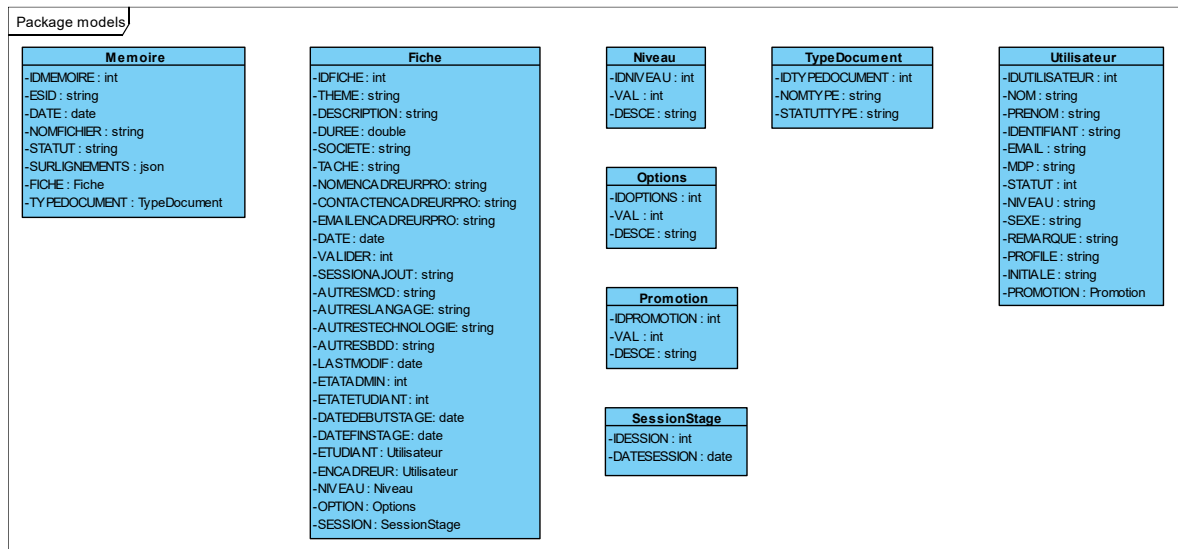


Figure 21 : Package models

- Le package « controllers »

- o Partie Node.js

Le package « controllers » sur la partie Node.js contient 2 fichiers comme le montre la figure suivante :

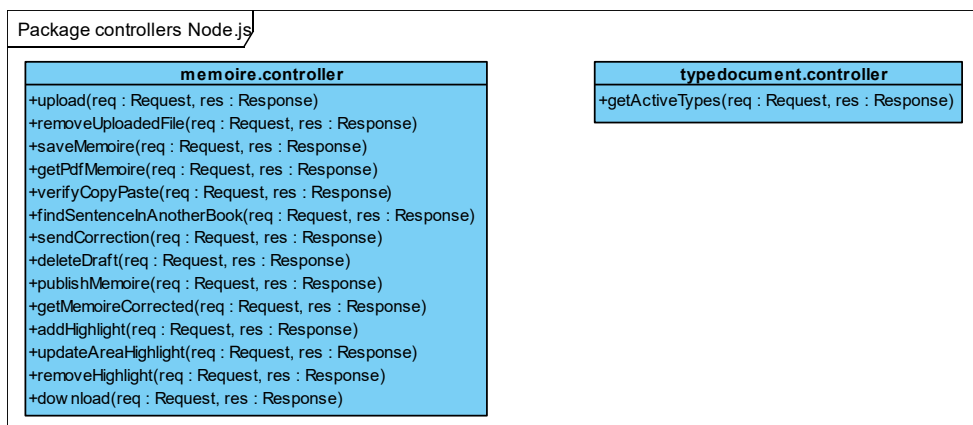


Figure 22 : Package controllers Node.js

- Partie PHP

Seuls les fichiers qui ont été modifiés et les fonctions qui ont été ajoutées et modifiées seront illustrées sur la figure suivante :

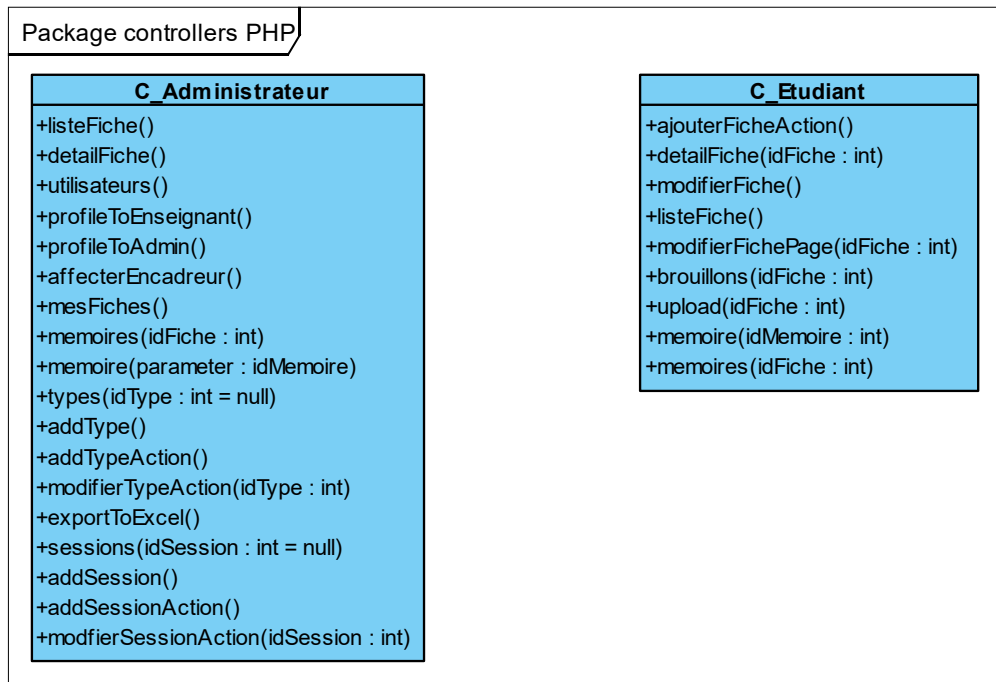


Figure 23 : Package controllers PHP

- Le package « repositories »

Le package « repositories » de la partie Node.js contient 5 fichiers.

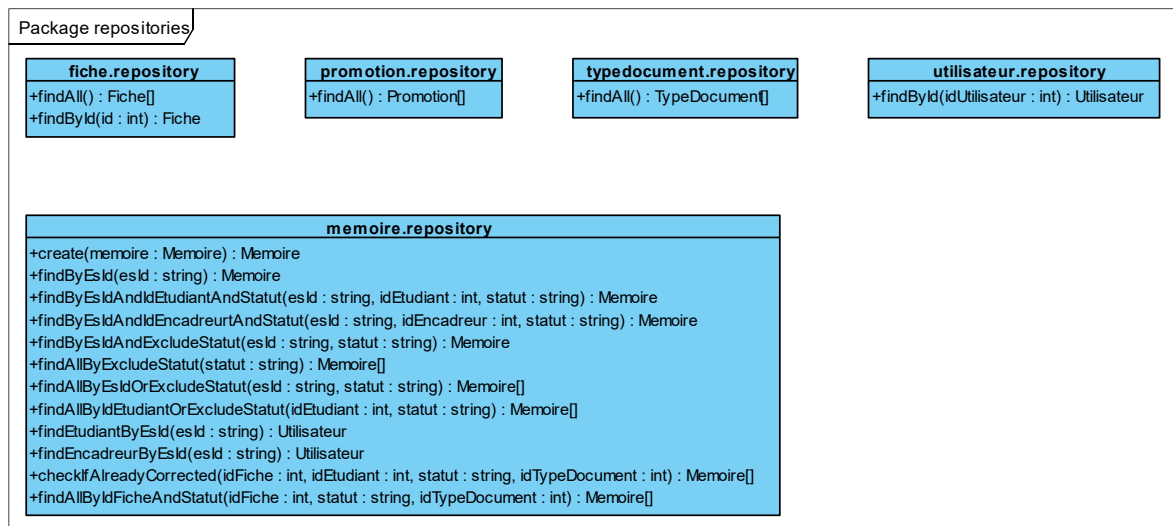


Figure 24 : Package repositories

- Le package « services »
- Le package « services » de la partie Node.js contient 5 fichiers.

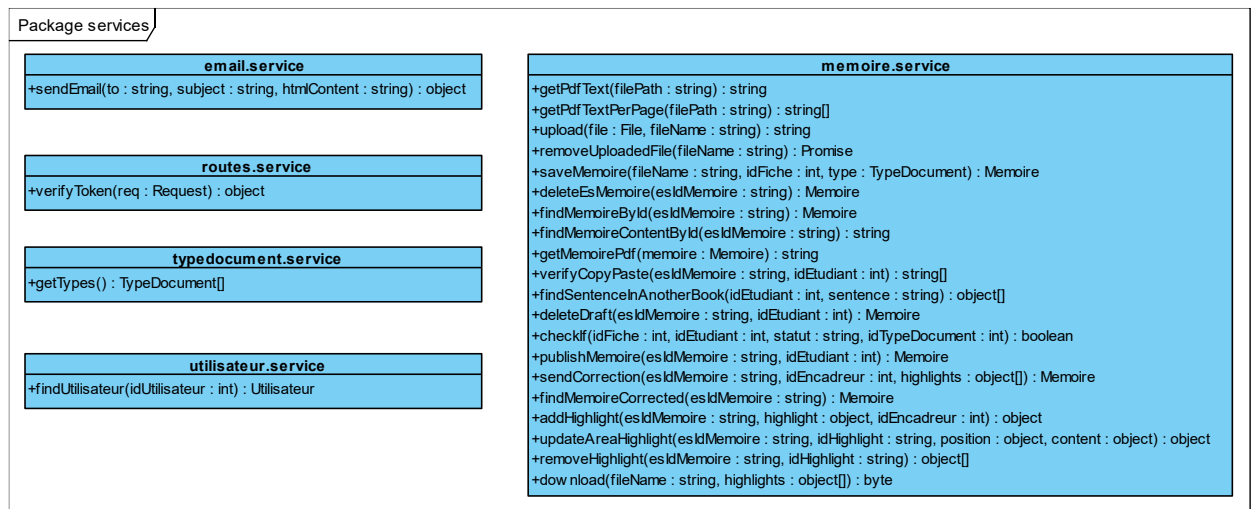


Figure 25 : Package services