



Université de la Manouba École Nationale des Sciences de l'Informatique

Rapport du Projet de Conception et de Développement

Sujet : Notation automatique d'un examen écrit en
langage naturel

Encadrante :
Pr. Chiraz Ben Othmane Zribi

Réalisé par :
Ben Kraiem Mohammed Amine
Hammami Mohammed Soulaimen
Montassar Maha

Année Universitaire : 2020 /2021

Appréciations et signature de l'encadrant

Remerciements

Tout d'abord, Nous remercions Dieu pour ses faveurs de nous avoir donner la patience et le courage pour réaliser ce travail. Avant de présenter notre travail, nous exprimions notre profonde gratitude à toute personne qui a contribué au bon déroulement de projet et en particulier Madame Chiraz Ben Othmane Zribi et Madame Anja Habacha pour leur encadrement, leur précieuses directions qui nous ont permis de mener à bien le projet. Nous tenons à exprimer notre respect aux membres du jury. Enfin, un grand merci aux nos familles pour l'encourage et le support qui nous ont supporté .

Table des matières

ABSTRACT	1
Remerciements	2
Liste des sigles et acronymes	8
Abstract	9
Introduction	10
1 Étude préalable	11
1.1 Introduction	11
1.2 Définitions et normes	11
1.2.1 Syntaxe	11
1.2.2 Sémantique	12
1.2.3 Orthographe	12
1.2.4 Ponctuation	12
1.3 Etude de l'existant	13
1.4 Critique de l'existant et proposition d'une solution	15
1.5 Choix du processus de développement du projet	16
2 Analyse et spécification des besoins	17
2.1 Introduction	17
2.2 Identification des acteurs	17
2.3 Analyse des besoins	18
2.3.1 Besoins fonctionnels	18
2.3.2 Besoins non fonctionnels	18
2.4 Spécification des besoins	19
2.4.1 Diagrammes de cas d'utilisation	19

2.4.2	Description de quelques scénarios	22
2.5	Conclusion	25
3	Conception Globale	26
3.1	Introduction	26
3.2	Architecture globale de l'application	26
3.3	Pré-traitement des données	26
3.3.1	Tokenisation	27
3.3.2	Élimination des mots vides	28
3.3.3	Lemmatisation	28
3.4	Orthographe	29
3.5	Analyse Syntaxique	29
3.5.1	CFG	30
3.5.2	Exemple d'arbre syntaxique	30
3.6	Analyse Sémantique	31
3.7	Ponctuation	32
3.7.1	TF-IDF	32
3.7.2	Le classifieur SVM	33
3.8	Conclusion	35
4	Conception Détailée	36
4.1	Introduction	36
4.2	Conception architecturale	36
4.2.1	Architecture logique	36
4.2.2	Architecture physique	38
4.3	Conception détaillée	39
4.3.1	Diagramme de classes	39
4.3.2	Diagramme d'activités	39
4.3.3	Diagramme de séquence objets	41
4.4	Conclusion	43
5	Réalisation	44
5.1	Introduction	44
5.2	Environnement de travail	44
5.2.1	Environnement matériel	44
5.2.2	Environnement logiciel	45
5.3	Aperçu sur le travail réalisé	47
5.3.1	Interface d'accueil :	47

5.3.2	Les interfaces du professeur :	47
5.3.3	Les interfaces de candidat	49
5.3.4	Les interfaces de l'administrateur	51
Conclusion et perspectives		54
Netographie		56
[francais]babel		

Table des figures

1.1	Page d'accueil de E-rater	13
1.2	Page de contact	14
1.3	Page d'accueil de IntelliMetric	15
16		
2.1	Diagramme de cas d'utilisation du système global	19
2.2	Diagramme de cas d'utilisation de l'administrateur	20
2.3	Diagramme de cas d'utilisation de professeur	21
2.4	Diagramme de cas d'utilisation du candidat	22
2.5	Diagramme de séquence système pour le scénario S'authentifier	23
2.6	Diagramme de séquence système pour le scénario Ajouter un essai	24
3.1	Exemple de Sentence Tokenization	27
3.2	Exemple de Word Tokenization	27
3.3	Exemple d'élimination des Mots Vides	28
3.4	Exemple de Lemmatisation	29
3.5	Arbre syntaxique	31
3.6	La valeur de TF-IDF sur le données d'entraînement	33
34		
3.8	Importance du stop words dans les différents types de phrase	34
3.9	Influence de pos-tag dans les différents types de phrase	35
37		
4.2	Diagramme de paquets	38
4.3	Diagramme de déploiement	39
4.4	Diagramme de classes	40
4.5	Diagramme d'activités pour le cas d'utilisation "Ajouter un essai"	41

4.6	Diagramme de séquence objet pour le cas d'utilisation "Ajouter un sujet"	42
5.1	Interface d'accueil	47
5.2	Interface d'ajout d'un sujet	48
5.3	Interface de feedback de professeur	49
5.4	Interface de liste des sujets	50
5.5	Interface d'ajout d'un essai	50
5.6	Interface de feedback de candidat	51
5.7	Interface de l'administrateur	52
5.8	Interface d'ajout d'un utilisateur	52
5.9	Interface avant suppression d'un utilisateur	53
5.10	Interface après suppression d'un utilisateur	53

Liste des sigles et acronymes

IA Intelligence Artificielle
ML Machine Leraning
NLP Natural Language Processing
UML Unified Modeling Language
MVC Model Vue Controler
CFG Context Free Grammar
SVM Support Vector Machine
TF-IDF Term Frequency-Inverse Document Frequency
NLTK Natural Language ToolKit
re Regular Expressions

Résumé - Ce rapport fait apparaître la conception et le développement d'une application web qui note automatiquement un essai écrit en anglais. Ce projet est basé sur une méthode d'apprentissage automatique et sur les techniques de traitement de langage naturel. Le candidat entre son essai puis le système retourne un feedback montrant les notes attribuées aux différents parties de correction.

Mots clés : IA, Machine Learning, traitement du langage naturel, langue anglaise.

Abstract - This report shows the design and development of a web application that automatically scores a written essay in English. This project is based on a machine learning method and natural language processing techniques. The candidate enters his test then the system returns a feedback showing the marks awarded to the different parts of correction.

Key words : IA, Machine Learning, Natural Language Processing, English language.

Introduction

Dépuis 1966, la notion d'automated essay scoring devient une activité réalisable par la machine. Plusieurs facteurs ont soulevé le besoin d'un système informatique capable de noter automatiquement un essai tel que l'objectivité d'évaluation, la rapidité de scoring, et ces facteurs aident les enseignants d'horizons divers à planifier leurs évaluations de telle sorte que les divergences s'atténuent. De plus, l'un des objectifs de EAS est aider les professeurs de langues à varier leur approche de l'enseignement et à réduire leur charge de travail sur des nombreux travaux académiques et administratifs.

dans ce cadre se met notre projet, son but est de créer un logiciel qui corrige des essais écrits en anglais en utilisant les techniques de traitement automatique du langage naturel et les algorithmes de machine learning et d'attribuer une note selon les normes proposées par le professeur et un feedback sur chaque partie de correction.

Le rapport est composé de cinq sections principales. Dans un premier temps, nous présentons une "Étude préalable", nous présentons quelques notions fondamentaux ainsi qu'une étude de l'existant. Deuxiement, nous présentons le chapitre "Analyse et spécification des besoins" qui décrit les besoins fonctionnels et non fonctionnels du système ainsi que les diagrammes UML. Dans le troisième chapitre intitulé "Conception Globale" nous présentons le cadre de notre projet ainsi que les techniques que nous avons manipuler pour réaliser le projet. Dans le chapitre suivant intitulé "Conception Détailée" nous présentons la conception architecturale et la conception détaillée en utilisant des diagrammes UML bien détaillés. Dans le dernier chapitre intitulé "Réalisation" nous présentons l'environnement matériel, l'environnement logiciel et quelques interfaces graphiques de notre application. Enfin, nous clôturons notre rapport par une conclusion générale en présentant le bilan de projet et des éventuelles perspectives.

Chapitre 1

Étude préalable

1.1 Introduction

L'étude préalable est une étape importante pour bien comprendre les exigences du marché ainsi que la détermination des objectifs de la système. Tout d'abord, Nous allons définir quelques mots clés. Ensuite, nous allons faire une étude de l'existant en discutant les avantages et les inconvénients de quelques systèmes qui se trouvent sur le marché. Enfin, nous proposons des solutions aux problèmes et nous fixons le modèle à suivre.

1.2 Définitions et normes

Automated Essay Scoring est défini comme la technologie informatique qui évolue et attribue une note à un essai écrit, il est utilisé pour surmonter les problèmes de temps, de coût et de fiabilité. L'auto-scoring est basé sur quatre principaux axes : syntaxe, sémantique, orthographe et ponctuation.

1.2.1 Syntaxe

la syntaxe est l'étude de la grammaire d'une phrase, elle traite les règles de construction d'une phrase. Selon la syntaxe nous pouvons classer les phrases en quatre catégories :

- Phrase Déclarative : elle communique une information, elle peut être simple ou complexe et se termine par un point.

- Exclamative : elle exprime une émotion et se termine par un point d'exclamation.
- Interrogative : elle exprime une interrogation ou une question et se termine par un point d'interrogation.
- Impérative : elle exprime un ordre ou un conseil et se termine par un point.

1.2.2 Sémantique

La sémantique est une branche de la linguistique qui représente la deuxième étape de l'analyse d'une phrase, elle étudie le sens d'une phrase. La sémantique représente ce qu'on veut transmettre, selon le contexte un mot peut prendre plusieurs significations.

1.2.3 Orthographe

C'est la première étape de l'analyse du texte, il représente la manière d'écriture d'un mot d'une langue et il sert à déterminer si le mot écrit est correct ou non. Nous distinguons deux catégories d'orthographe, la première est l'orthographe lexicale, elle définit la manière d'écriture d'un mot. La deuxième est l'orthographe grammaticale qui représente les règles d'accord en genre et en nombre des mots selon leurs rôles dans la phrase.

1.2.4 Ponctuation

la ponctuation est indispensable pour lire un texte et le comprendre, elle indique comment séparer les phrases d'un texte et comment lire une phrase pour comprendre son sens.

Les règles de ponctuation sont les suivantes :

- Le point (.) : marque la fin d'une phrase déclarative ou une phrase impérative.
- Le point d'interrogation (?) : marque la fin d'une phrase interrogative.
- Le point d'exclamation (!) : marque la fin d'une phrase exclamative.
- la virgule (,) : sépare deux groupes de mots.

1.3 Etude de l'existant

Dans cette partie de chapitre, nous étudions quelques applications existantes afin de dégager leurs limites et mieux comprendre les objectifs de notre application.

E-rater®¹ :

E-rater est un moteur de notation automatique développé par Educational Testing Service ETS² en 1999, il est adapté par The Criterion® Online Writing Evaluation Service³. Ce moteur fournit un score basé seulement sur le traitement du langage naturel avec des commentaires sur la grammaire, le style et l'organisation et le développement.

La Figure 1.1 représente la page d'accueil de E-rater.

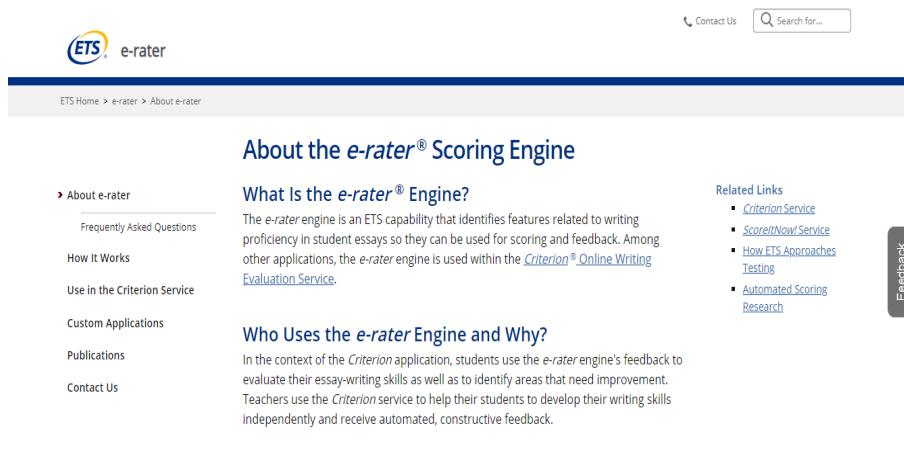


FIGURE 1.1 – Page d'accueil de E-rater

La figure 1.2 représente la page de contact.

-
1. E-rater® : <https://www.ets.org/erater/about>
 2. ETS : <https://www.ets.org/>
 3. The Criterion® : <https://criterion.ets.org/Criterion>

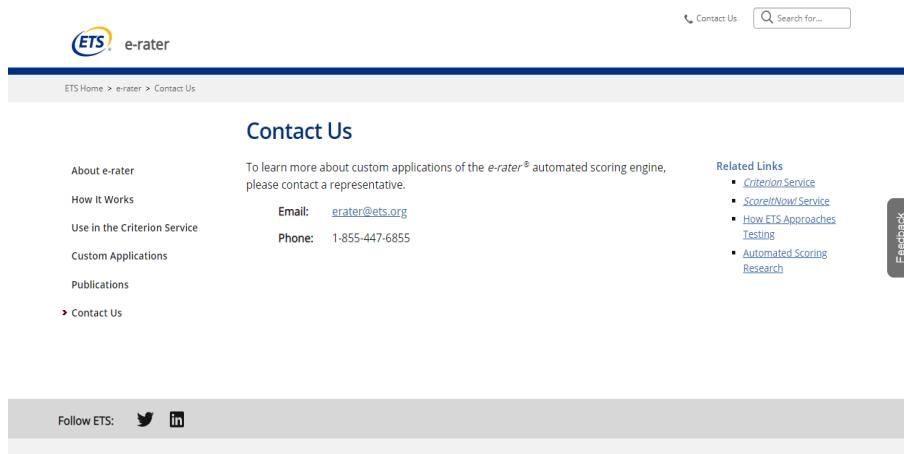


FIGURE 1.2 – Page de contact

IntelliMetric[®]⁴ :

C'est le premier système de notation automatique des essais basé sur l'intelligence artificielle IA et le machine learning ML, il est développé en 1997 avec un taux d'exactitude au dessus de 90 pourcent et il est capable de noter des essais dans plus de 20 langues différentes. Cet outil évalue plus de 300 fonctionnalités liées à la syntaxe, la sémantique et au discours.

La Figure 1.3 montre le site officiel de Intellimetric.

4. IntelliMetric[®] : <http://www.intellimetric.com/direct/>

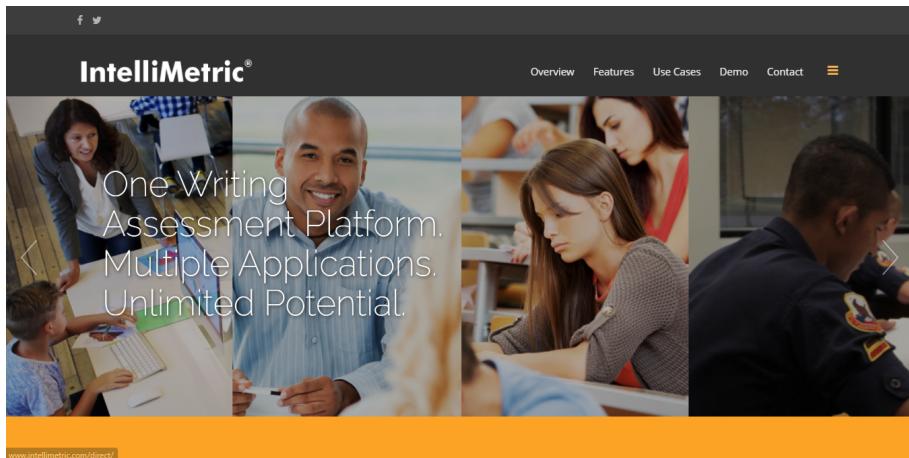


FIGURE 1.3 – Page d'accueil de IntelliMetric

1.4 Critique de l'existant et proposition d'une solution

Après une étude sur les moteurs de notation automatique existants nous avons remarqué que ces outils possèdent des lacunes. Notre étude a mené aux résultats suivants :

- Il n'y a pas un moteur tunisien ou arabe qui fait l'auto-scoring.
- Les deux moteurs sont privés.
- E-rater est basé sur les techniques de NLP non sur les algorithmes de ML.
- E-rater ne fait pas la notation de la sémantique.
- Les deux moteurs ne peuvent pas noter la ponctuation.

Tenant compte de ces lacunes, nous réalisons un système qui fait la notation d'un essai automatiquement, ce système est dédié aux universités tunisiennes et le monde arabe d'une manière générale. Ce plate-forme est basé sur l'apprentissage automatique et les techniques de NLP.

L'avantage de ce système c'est l'amélioration de la performance à l'aide d'un processeur d'apprentissage incrémental.

1.5 Choix du processus de développement du projet

Pour réaliser ce projet, nous avons utilisé le modèle agile Scrum qui s'adapte aux changements et améliore la productivité de l'équipe. Le cycle de vie de Scrum est découpé en des boîtes de temps appelées sprints qui représentent des périodes pendant lesquels un ensemble des tâches bien déterminées doivent être établis. À chaque fin de sprint, une réunion se fait entre les membres de l'équipe pour discuter de ce qui a été achevé. Pour garantir la bonne déroulement du processus deux acteurs principaux doivent être présents : le "Product owner" qui représente le client et le "Scrum master" qui gère l'équipe technique.

La Figure 1.4 présente les étapes du Scrum ainsi que ses acteurs.



FIGURE 1.4 – Modèle de développement SCRUM⁵

5. SCRUM : <https://teamhood.com/agile/scrum-of-scrums-for-scaled-agile/>

Chapitre 2

Analyse et spécification des besoins

2.1 Introduction

Dans ce chapitre, grâce aux diagrammes de séquence, nous décrivons comment les éléments du système interagissent entre eux et avec les acteurs. Ce chapitre sert à comprendre le contexte du système. Nous allons commencer par la phase d'analyse en définissant les acteurs et en précisant les rôles et les fonctionnalités de chaque acteur. Nous finissons par la spécification des besoins fonctionnels et non fonctionnels illustrés par les diagrammes UML cas utilisation et séquence système.

2.2 Identification des acteurs

Un acteur est une entité externe qui interagit avec le système, il peut être soit une personne, soit un périphérique ou un autre système qui joue un rôle avec le système .

Les acteurs de notre système sont : l'Administrateur, le Professeur et le Candidat.

2.3 Analyse des besoins

Dans cette phase de chapitre, nous présentons les besoins fonctionnels et non fonctionnels du système.

2.3.1 Besoins fonctionnels

La spécification des besoins fonctionnels nous permet d'avoir une meilleure approche des utilisateurs, elle sert aussi à mettre en évidence les fonctionnalités du système. Pour cela nous allons identifier les acteurs du système ainsi que les cas utilisations par chaque acteur

- Professeur :
 - S'inscrire au système en remplissant les champs adresse e-mail et mot de passe
 - Ajouter un sujet en spécifiant le barème, le thème et la description
 - Consulter les feedbacks des essais
- Candidat :
 - S'inscrire au système en remplissant les champs adresse e-mail et mot de passe
 - Déposer son essai
 - Consulter le feedback de son essai
- Administrateur :
 - S'inscrire au système en remplissant les champs adresse e-mail et mot de passe
 - Gérer les utilisateurs par ajout, suppression ou modification

2.3.2 Besoins non fonctionnels

Les besoins non fonctionnels sont des exigences qui n'influencent pas sur le fonctionnement du système, au contraire ils garantissent la satisfaction de client en fournissant une bonne qualité de services.

- Utilisabilité : l'interface graphique du système doit être facile à manipuler.
- Fiabilité : l'autocorrection doit être précise et objective.
- Rapidité : la réponse du système doit être rapide et ne dépasse pas un délai raisonnable. En effet, le passage d'une interface à une autre ne doit pas dépasser 3 secondes et le traitement du sujet ne doit pas dépasser 5 minutes.

2.4 Spécification des besoins

À ce niveau du chapitre, nous spécifions, en utilisant le langage de modélisation UML, les besoins fonctionnels du système par des diagrammes cas d'utilisations de tous les acteurs. De plus nous décrivons quelques scénarios par des diagrammes de séquence.

2.4.1 Diagrammes de cas d'utilisation

Le diagramme cas d'utilisation fait partie des diagrammes de comportement du langage UML, il permet de représenter les fonctions du système de point de vue de l'acteur. Nous allons commencer par le diagramme cas d'utilisation général présenté par la Figure 2.1, tous les utilisateurs doivent s'authentifier afin d'utiliser l'application. Pour des raisons de simplifications, nous ne présenterons pas l'identification dans les diagrammes de cas d'utilisation détaillés.

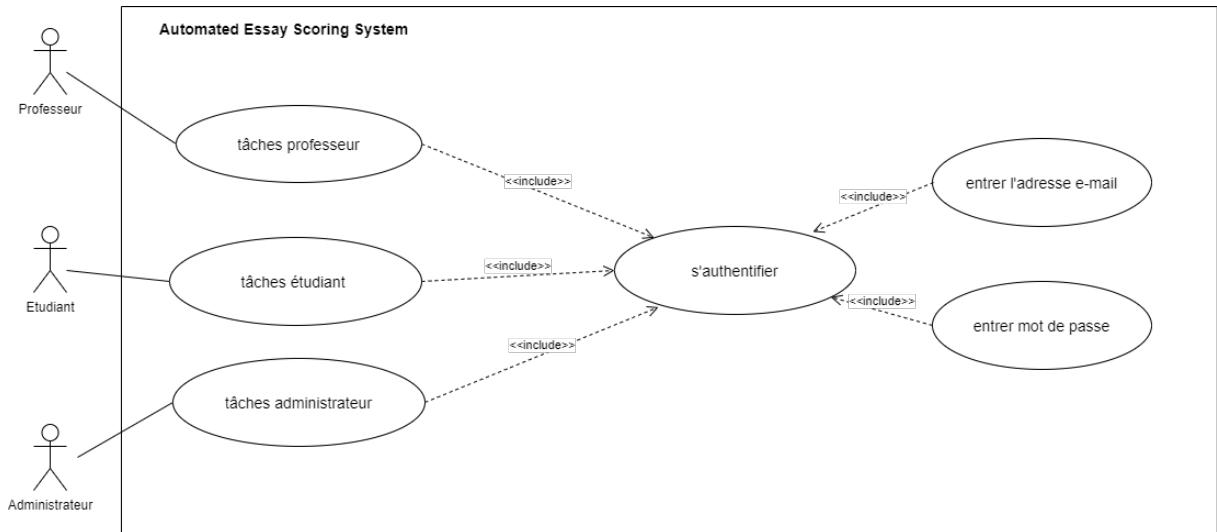


FIGURE 2.1 – Diagramme de cas d'utilisation du système global

La Figure 2.2 montre les fonctionnalités offertes par l'application à l'administrateur.

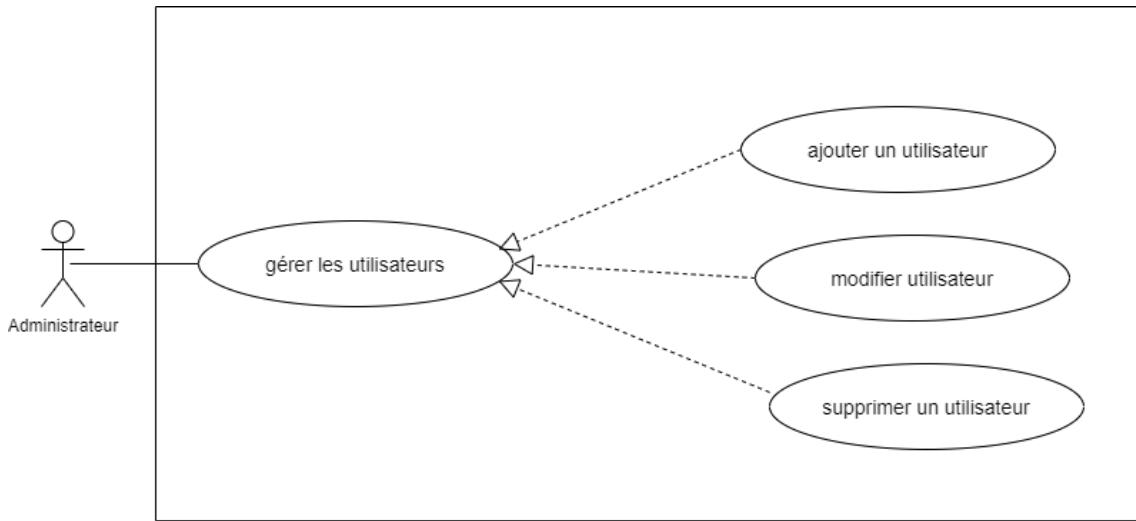


FIGURE 2.2 – Diagramme de cas d'utilisation de l'administrateur

La Figure 2.3 représente les fonctionnalités de professeur à savoir

l'ajout d'un sujet et la consultation des notes des candidats.

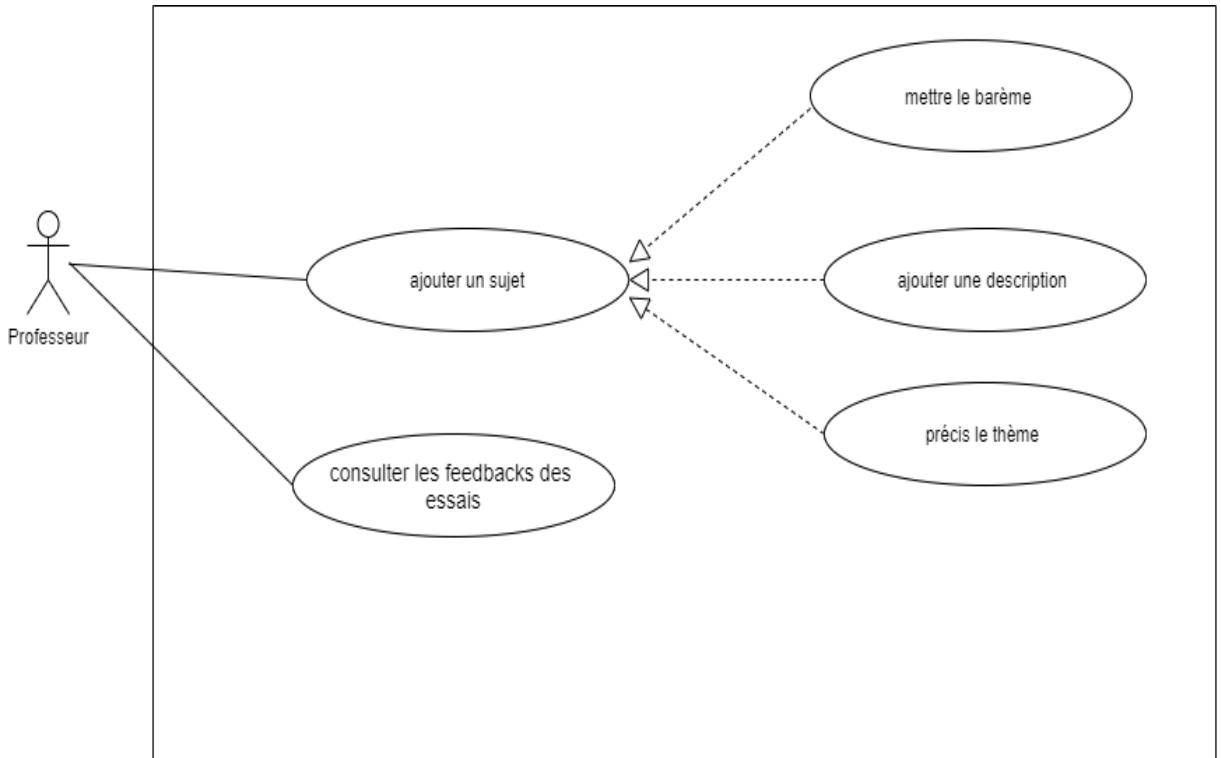


FIGURE 2.3 – Diagramme de cas d'utilisation de professeur

La Figure 2.4 représente les fonctionnalités offertes par le système au candidat. Le candidat peut déposer son essai et consulter son note.

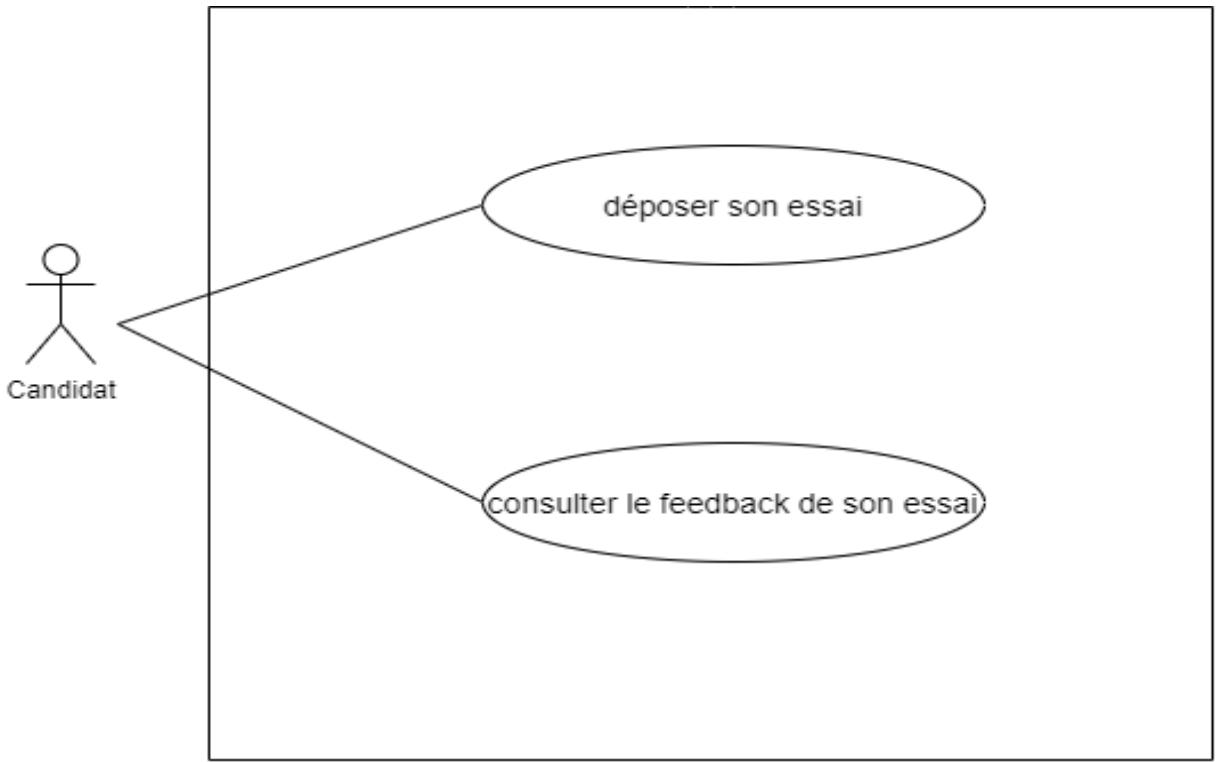


FIGURE 2.4 – Diagramme de cas d'utilisation du candidat

2.4.2 Description de quelques scénarios

Dans cette section, grâce aux diagrammes de séquence, nous décrivons comment les éléments du système interagissent entre eux et avec les acteurs suivant un ordre chronologique.

— Diagramme de Séquence système pour le scénario "s'authentifier"
 La Figure 2.5 montre les étapes à suivre pour s'authentifier au système. En effet, lorsque l'utilisateur demande l'inscription il aura accès au formulaire d'authentification, il doit remplir les champs e-mail et mot de passe. Enfin, l'utilisateur tape sur le bouton "Sign in" et accède à la interface convenable.

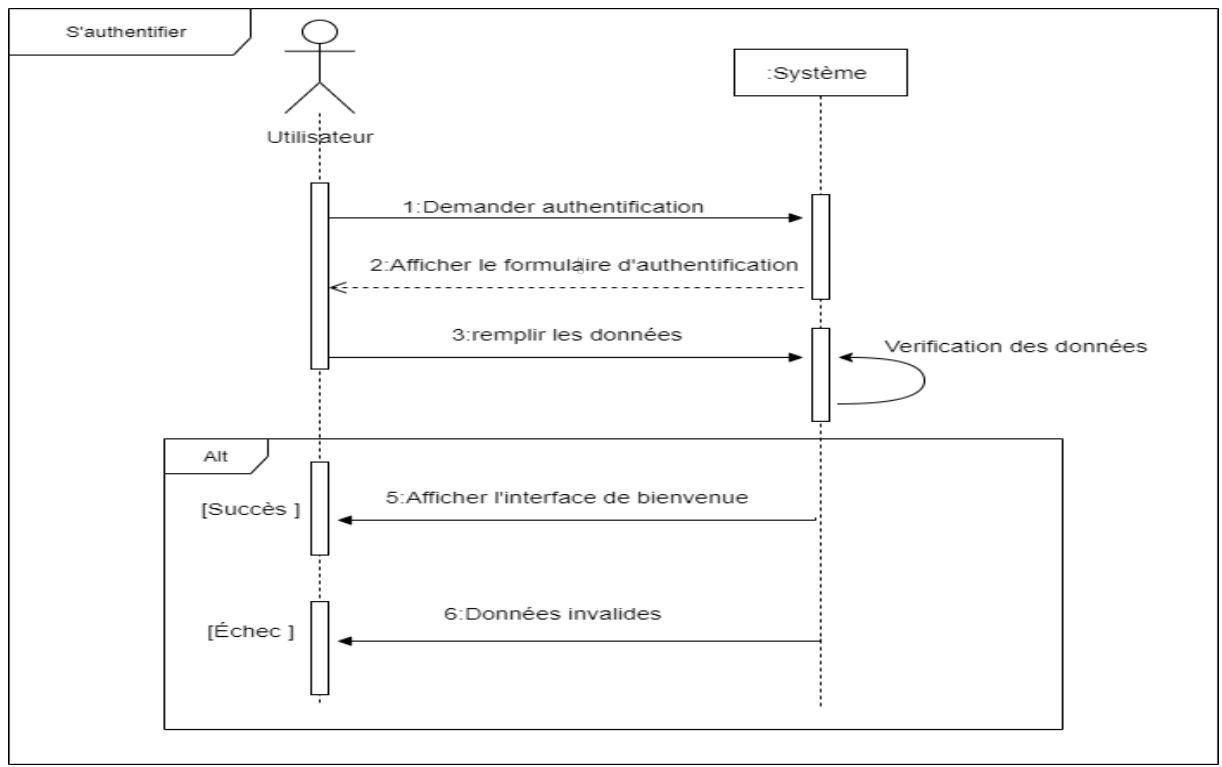


FIGURE 2.5 – Diagramme de séquence système pour le scénario S'authentifier

— Diagramme de Séquence système pour le scénario "Ajouter essai"
La Figure 2.6 représente les étapes qui doit suivre le candidat pour ajouter son essai. En effet, le candidat doit s'authentifier d'abord pour accéder à son propre interface puis il peut remplir le formulaire d'ajout.

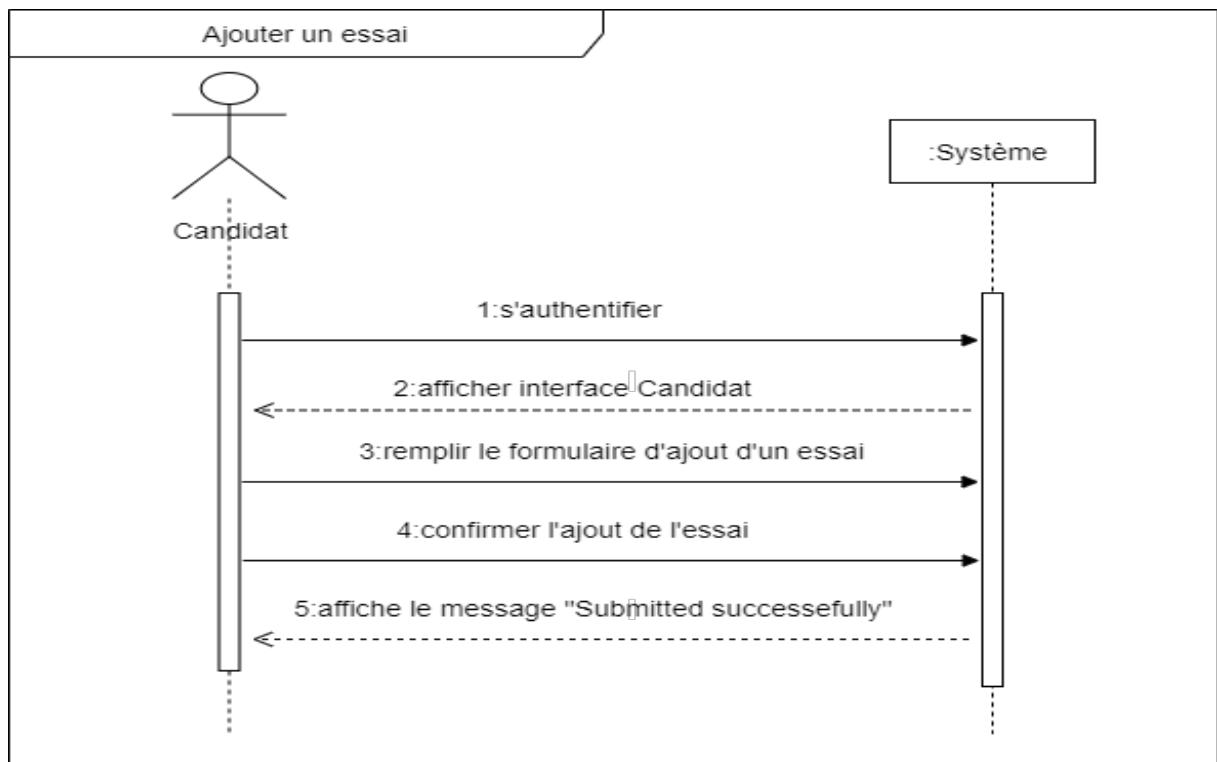


FIGURE 2.6 – Diagramme de séquence système pour le scénario Ajouter un essai

2.5 Conclusion

Au niveau de ce chapitre nous avons présenté au premier lieu les besoins du notre système puis en deuxième lieu nous avons modélisé les diagrammes UML qui décrivent les fonctionnalités du système à travers quatre diagrammes CU et deux diagrammes de séquence.

L'objectif du chapitre suivant est de présenter le "back-end" qui est la phase d'apprentissage automatique de notre application.

Chapitre 3

Conception Globale

3.1 Introduction

Dans ce chapitre, nous expliquons comment la phase d'apprentissage automatique de notre système a été élaboré. Tout d'abord nous présentons l'architecture globale du système puis les techniques que nous avons manipuler pour le pré-traitement des données.

3.2 Architecture globale de l'application

Notre projet est un outil d'apprentissage automatique qui est capable de noter un essai et d'améliorer son performance au fur et à mesure de son utilisation.

Pour ce faire, nous avons suivi les étapes suivantes :

- Pré-traitement des données.
- Orthographe.
- Analyse syntaxique.
- Analyse sémantique.
- Ponctuation.

3.3 Pré-traitement des données

L'entrée de notre système est une donnée textuelle écrite en anglais. Pour l'analyser, cette donnée passe par les techniques de pré-traitement sui-

vantes :

3.3.1 Tokenisation

La tokenization est un outil important pour le pré-traitement des données , elle consiste à découper un texte soit en des phrases ou des mots. Les Figures 3.1 et 3.2 montrent les deux types de tokenisation que nous avons utilisé.

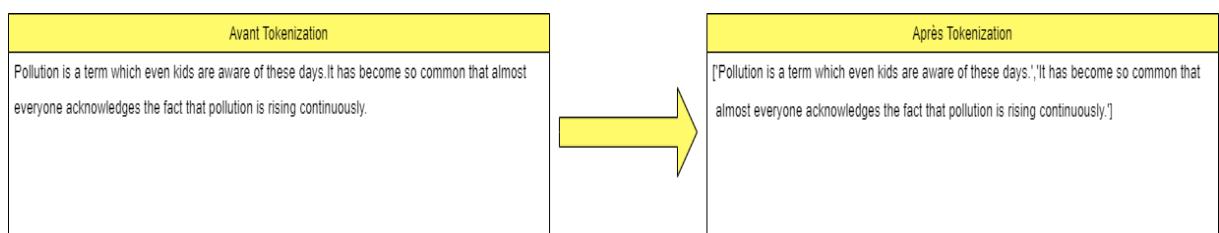


FIGURE 3.1 – Exemple de Sentence Tokenization

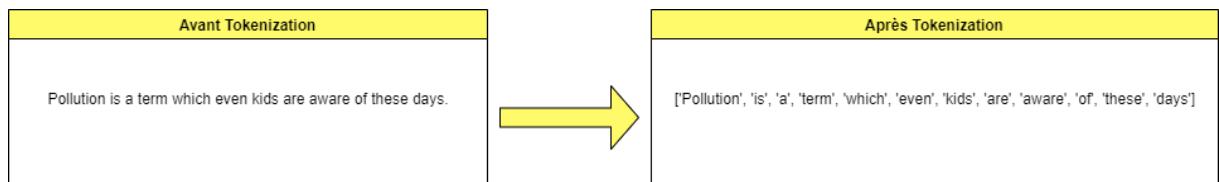


FIGURE 3.2 – Exemple de Word Tokenization

3.3.2 Élimination des mots vides

L'élimination des mots vides (Stop Words) est une étape importante qui consiste à filtrer le texte de tous les mots qui sont inutiles pour le traitement et qui n'apportent pas de sens.

La Figure 3.3 montre un exemple d'élimination de Stop Words.

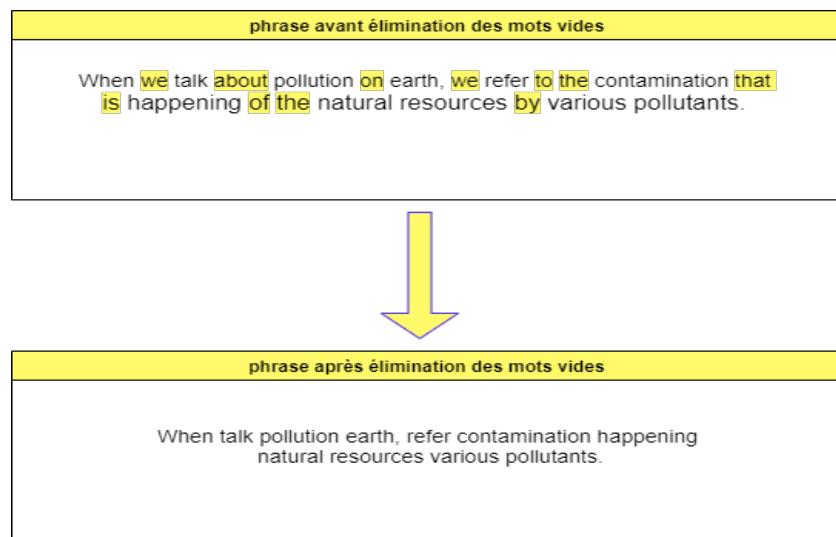


FIGURE 3.3 – Exemple d'élimination des Mots Vides

3.3.3 Lemmatisation

La lemmatisation est une technique de normalisation de texte, elle sert à supprimer les fins flexionnelles d'un mot pour obtenir sa forme de base. La Figure 3.4 illustre un exemple de lemmatisation.

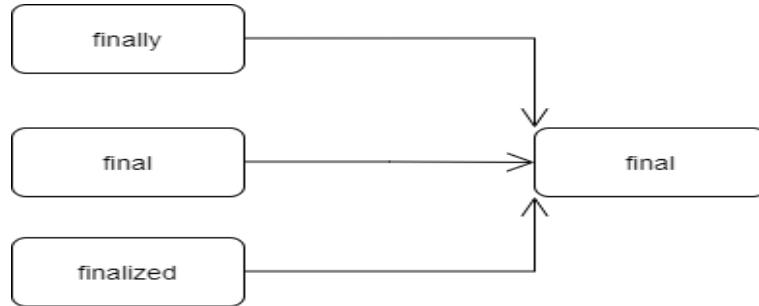


FIGURE 3.4 – Exemple de Lemmatisation

3.4 Orthographe

L'orthographe représente la manière d'écriture d'un mot, il détermine si un mot est écrit correctement ou non. Pour réaliser la correction orthographique nous avons créé une fonction "correctText" qui prend en paramètre le texte du candidat.

Le fonctionnement de "correctText" se fait sur trois étapes :

La première étape consiste à découper le texte en des mots en utilisant la fonction split de python.

La deuxième étape consiste à insérer les mots retournés par la fonction split dans une liste.

Dans la troisième étape nous parcourons la liste pour déterminer si le mot est correct ou non, si le pos-tag du mot est un nom propre nous l'ajoutons à un fichier appelé textCorrected contenant tous les mots après la correction de l'orthographe et si le pos-tag du mot n'est pas un nom propre nous appliquons la méthode textblob qui corrige le mot et l'ajoute à textCorrected.

L'attribution de note se fait par une fonction qui prend en paramètre le text de candidat et le text corrigé et retourne le pourcentage des mots incorrects.

3.5 Analyse Syntaxique

L'analyse syntaxique est la détermination de la structure d'une phrase, pour le faire nous avons utilisé le CFG.[\[URL1\]](#)

3.5.1 CFG

CFG est une technique de NLP qui détermine la grammaire d'une phrase. Le Context Free Grammar G est défini par $G = (V, T, S, P)$ avec :

- **V** : C'est l'ensemble des symboles non terminaux.
- **T** : C'est l'ensemble des symboles terminaux.
- **S** : C'est l'axiome de départ.
- **P** : C'est l'ensemble des règles de production, chaque règle s'écrit sous la forme $A \rightarrow s$ avec A est un non terminal et s est une séquence des terminaux et non terminaux.

Pour chaque type de phrase nous avons identifié sa grammaire.

3.5.2 Exemple d'arbre syntaxique

Nous définissons la grammaire d'une phrase simple :

Règles syntaxiques :

R1 : $S \rightarrow NP VP$

R2 : $VP \rightarrow Verb NP$

R3 : $NP \rightarrow Det Noun \mid NP PP$

R4 : $PP \rightarrow Pre NP$

Règles lexicales :

L1 : $Det \rightarrow 'the' \mid 'a'$

L2 : $Pre \rightarrow 'to'$

L3 : $Noun \rightarrow 'child' \mid 'school'$

L4 : $Verb \rightarrow 'went'$

La Figure 3.5 montre l'arbre syntaxique de la phrase : the child went to school.

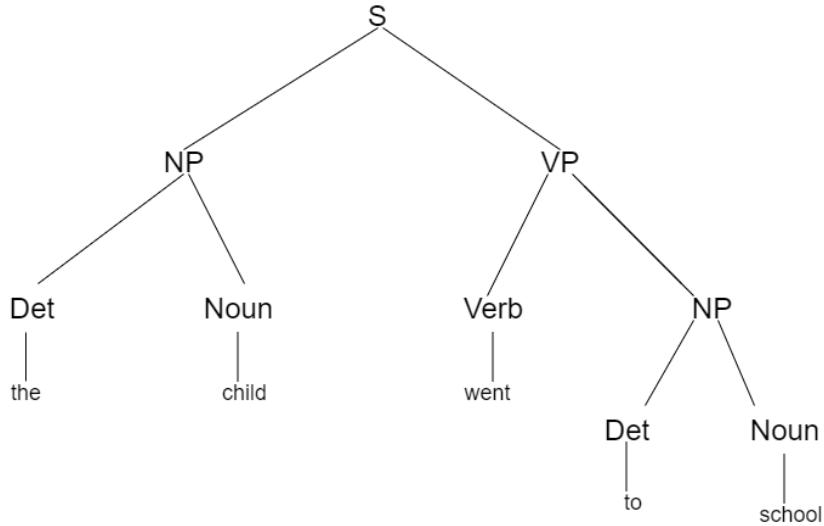


FIGURE 3.5 – Arbre syntaxique

La notation de la grammaire se fait à travers une fonction qui prend en paramètres le nombre total des phrases de l'essai et le nombre des phrases incorrectes et retourne le pourcentage des phrases correctes.

3.6 Analyse Sémantique

La sémantique est l'étude de sens de la phrase. Pour analyser la sémantique d'un essai nous avons utilisé les trois fonctions suivantes :

- preprocess-data : son objectif est de faire le pré-traitement de l'essai en employant la tokanization, élimination des stops words et le stemming.
- prepare-corpus : elle prend en argument le résultat de la fonction preprocess-data, son objectif est de créer un dictionnaire contenant les termes de l'argument et le convertir en une matrice. Cette fonction retourne un dictionnaire.
- create-gensim-lsa-model : cette fonction prend en argument un texte et le nombre des sujets et elle retourne deux listes. La pre-

mière contient les sujets les plus fréquents dans l'essai et la deuxième contient les valeurs des fréquences de chaque sujet.
La note attribuée à la sémantique est le pourcentage d'appartenance de sujet de professeur par rapport à l'essai de candidat.

3.7 Ponctuation

La ponction détermine comment les phrases sont séparées. Pour étudier la ponction et avoir une bonne prédition, nous avons utilisé les algorithmes de Machine Learning sur les différents types de phrase.

3.7.1 TF-IDF

C'est une méthode d'extraction des vecteurs caractéristiques et une méthode d'analyse statique qui détermine la pertinence d'un mot dans un texte en attribuant un poids important aux mots rares.

La valeur de TF-IDF est calculée comme suit :

$$\text{TF} = \frac{(\text{Nombre d'apparition d'un mot})}{(\text{Nombre des mots})}$$

$$\text{IDF} = \log \frac{(\text{Nombre des essai})}{(\text{Nombre des essai dans lesquels le mot est apparu})}$$

$$\text{TF-IDF} = \text{TF} * \text{IDF}$$

La Figure 3.6 montre un résultat de la valeur de TF-IDF sur les données d'entraînement .

	0	1	2	3	4	5	6	7	8	9	...	341	342	343	344	345	346	347	348	349	punctuation_rate
0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10.00
1	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.20
2	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.90
3	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.86
4	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10.53
...
85	0.0	0.0	0.0	0.0	0.0	0.330584	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.04
86	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.26
87	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8.00
88	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.14
89	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.85

FIGURE 3.6 – La valeur de TF-IDF sur le données d’entraînement

3.7.2 Le classifieur SVM

Support Vector Machine (SVM) est un algorithme d’apprentissage automatique supervisé destiné à résoudre les problèmes de régression. c’est un classifieur linéaire qui sépare la base de données en deux groupes pour faire la prédiction. Cette technique d’apprentissage transforme les données pour trouver un écart maximal entre les deux groupes.

La Figure 3.7 présente deux types des données séparées par un plan.

7. SVM : <https://www.codershood.info/2019/01/10/support-vector-machine-machine-learning-algorithm-with-example-and-code/>

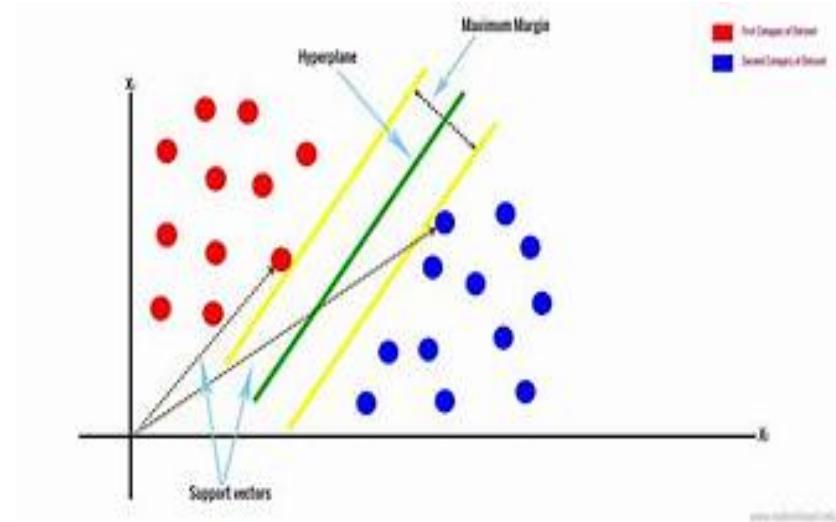


FIGURE 3.7 – SVM avec une limite optimale et une marge maximale⁷

Les Figures 3.8 et 3.9 représentent quelques résultats de l'application de l'algorithem SVM.

La Figure 3.8 montre l'importance des stop words dans la ponctuation.

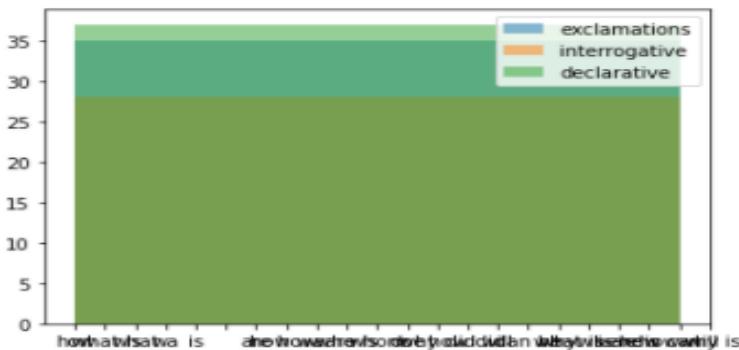


FIGURE 3.8 – Importance du stop words dans les différents types de phrase

La Figure 3.9 représente comment le pos-tag d'un mot influence sur le type de la phrase.

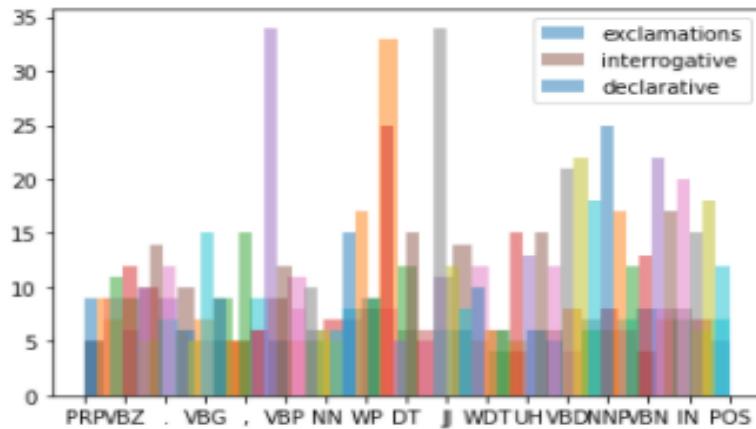


FIGURE 3.9 – Influence de pos-tag dans les différents types de phrase

La notation de la ponctuation se fait par la détection de type de phrase en comptant le nombre des phrases dont la ponctuation est incorrecte puis par une fonction qui prend en paramètres le nombre total des phrases et le nombre des phrases incorrectes et retourne le pourcentage des phrases correctes.

3.8 Conclusion

Dans ce chapitre, nous avons présenté l'architecture du système, nous avons concentré sur le pré-traitement de données et sur les quatre parties de la notation qui sont : l'analyse syntaxique, l'analyse sémantique, l'orthographe et la ponctuation.

Dans le chapitre suivant nous toucherons la conception.

Chapitre 4

Conception Détaillée

4.1 Introduction

Dans ce chapitre, nous détaillons la conception et le développement de notre projet. La conception est une tâche nécessaire pour développer une meilleure solution qui satisfait les besoins fonctionnels du système. La conception architecturale se fait en deux étapes, d'abord nous présentons la conception architecturale puis nous présentons la conception détaillée.

4.2 Conception architecturale

L'objectif de la conception architecturale est l'organisation et la structuration de système d'une manière globale. Elle se fait en deux étapes : l'architecture logique et l'architecture physique.

4.2.1 Architecture logique

L'architecture logique représente la vue logique du système, elle permet d'identifier les composants logiciels du système et indique comment ces composants interagissent entre eux.

L'architecture que nous avons opté est : Modèle/Vue/Contrôleur (MVC). C'est un pattern très puissants qui donne une bonne organisation du code source. Le principe du pattern MVC est la séparation de logique de code en seulement trois parties en définissant les interactions entre eux.

- Modèle : Cette partie sert à récupérer les informations dans la base des données, les organiser et les rassembler pour qu'elles puissent être utilisées par le contrôleur.
- Vue : C'est la seule partie qui sera visible à l'œil nu par l'utilisateur, son rôle est de récupérer les données qui lui sont envoyées par le modèle.
- Contrôleur : C'est l'intermédiaire entre le modèle et la vue, son rôle est de gérer la partie logique du système. Il demande au contrôleur les données à analyser puis il prend les décisions nécessaires à l'affichage et l'envoi à la vue.

La Figure 4.1 montre ce modèle.

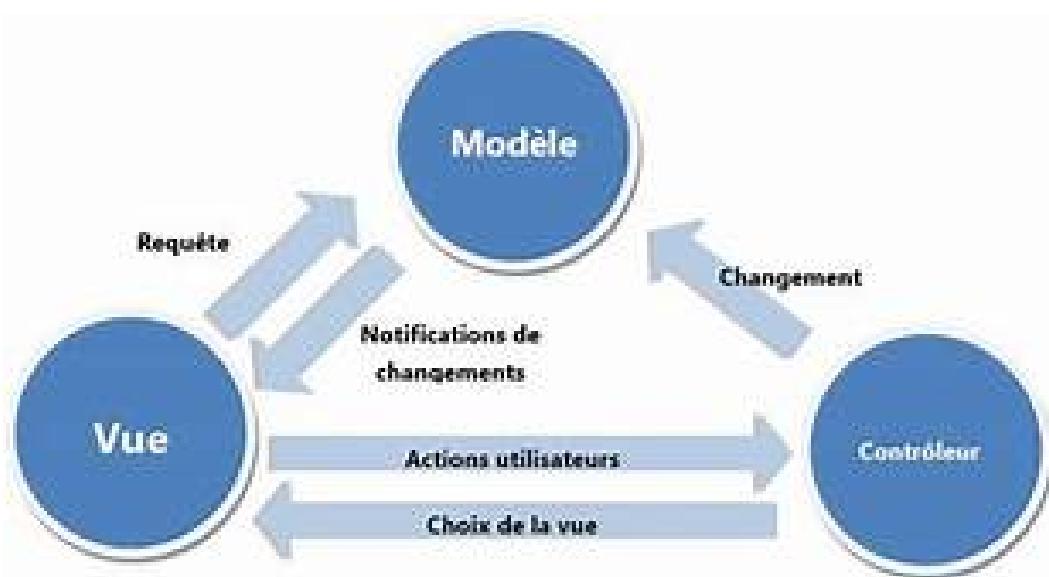


FIGURE 4.1 – Architecture MVC⁶

4.2.1.1 Diagramme de paquets

Le diagramme de package est l'un des diagrammes UML les plus connus, il représente l'organisation des éléments sous forme de paquetage en regroupant les classes qui communiquent entre elles dans un seul paquet et en

6. MVC : <https://c-maneu.developpez.com/tutorial/web/php/symfony/intro/>

identifiant les liens de dépendance entre ces paquets.

- Le paquet Modèle : regroupe les classes représentant les tables de la base de données.
- Le paquet Vue : regroupe les classes représentant les interfaces de l'utilisateur.
- Le paquet Contrôleur : regroupe les classes responsables de l'implémentation de l'algorithme de classification et de gestion des données et des utilisateurs.

La Figure 4.2 présente l'architecture logique de système.

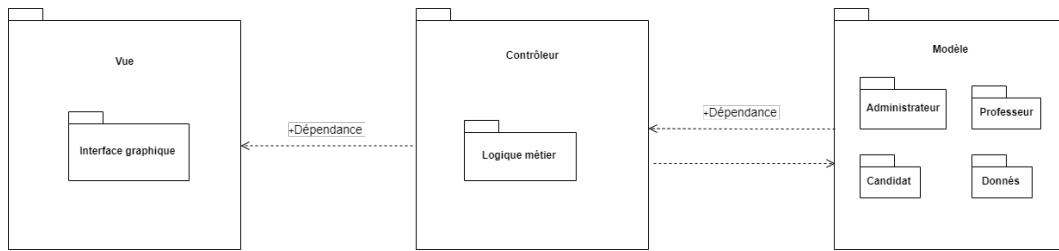


FIGURE 4.2 – Diagramme de paquets

4.2.2 Architecture physique

L'architecture physique est la modélisation de déploiement physique. Elle représente le déploiement des informations sur les différents composants matériels. L'architecture appropriée pour notre application est l'architecture à 3 couches.

- Le tier Client : C'est le demandeur de ressources.
- Le tier Applicatif : C'est le serveur web sur lequel est déployé l'application.
- Le tier Données : C'est le serveur sur lequel est déployé le système de gestion de base des données.

La Figure 4.3 représente l'architecture physique utilisée.

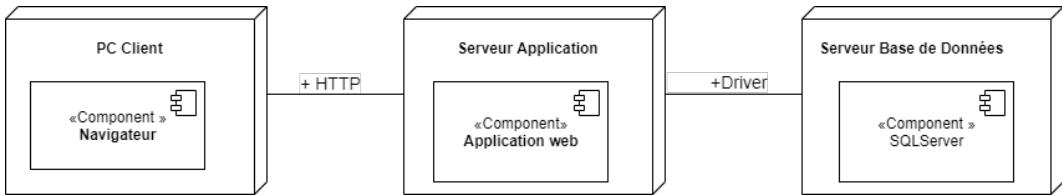


FIGURE 4.3 – Diagramme de déploiement

4.3 Conception détaillée

Nous présentons maintenant la conception détaillée définie d'exprimer comment les tâches sont organisées et réparties entre les différents modules qui la constituent.

4.3.1 Diagramme de classes

Le diagramme de classe est un diagramme objet qui décrit la structure statique d'un modèle. Il décrit les classes, les opérations, les attributs et les relations entre les objets.

Les classes de notre application sont :

- La classe Utilisateur : C'est un internaute qui accède au site web en entrant ses coordonnées : Adresse e-mail et mot de passe.
- La classe Professeur : Elle hérite de la classe Personne et caractérise l'acteur professeur qui dépose le thème de sujet ainsi que le barème.
- La classe Candidat : Elle hérite de la classe Personne et caractérise l'acteur candidat qui dépose son essai écrit en anglais.
- La classe Administrateur : Elle hérite de la classe Personne et caractérise l'acteur administrateur qui gère les comptes des utilisateurs en ajoutant ou supprimant ou modifiant un compte.

La Figure 4.4 représente le diagramme des classes.

4.3.2 Diagramme d'activités

Le diagramme d'activité est un diagramme de comportement du système qui décrit l'enchaînement des activités d'un cas d'utilisation. La Figure

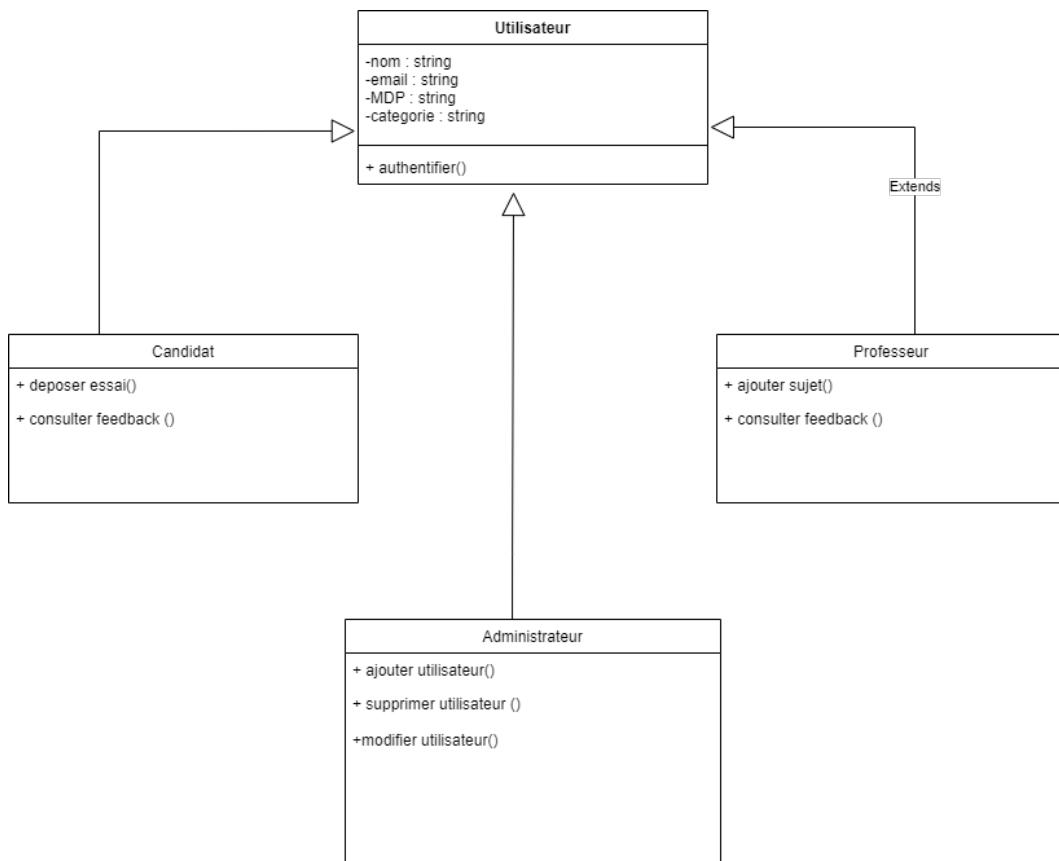


FIGURE 4.4 – Diagramme de classes

4.5 illustre le diagramme d'activité pour le cas d'utilisation "Ajouter un essai" montrant les différentes étapes qui suit le candidat pour ajouter son essai.

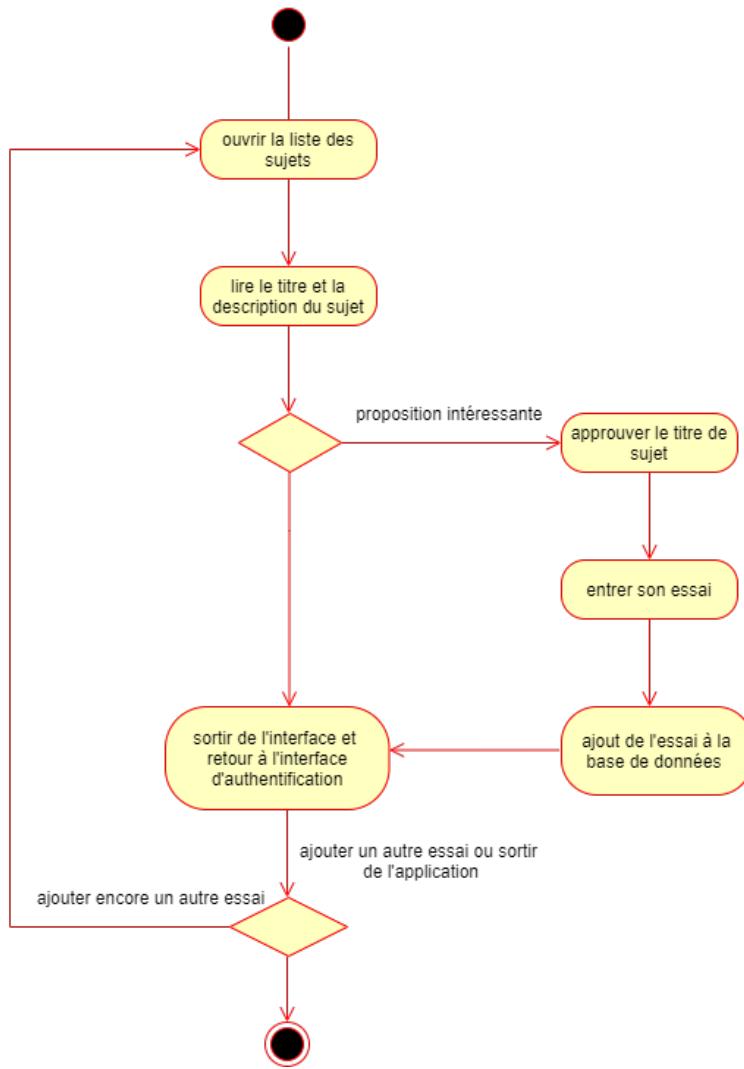


FIGURE 4.5 – Diagramme d'activités pour le cas d'utilisation "Ajouter un essai"

4.3.3 Diagramme de séquence objets

Le diagremme de séquence objets fait partie des diagrammes de comportement, il reprsente la chronologie des échanges des messages entre les

acteurs et les séquences objets.

La Figure 4.6 illustre le diagramme de séquence objet pour le cas d'utilisation "Ajouter un sujet". Après authentification, le professeur accède à son propre interface, il doit cliquer sur le bouton "Add subject" pour accéder à l'interface d'ajout des sujets, puis il ajoute son sujet ainsi qu'une description en cliquant sur le bouton "Add Subject". Enfin le professeur est notifié que son sujet est ajouté par un message affiché par l'interface .

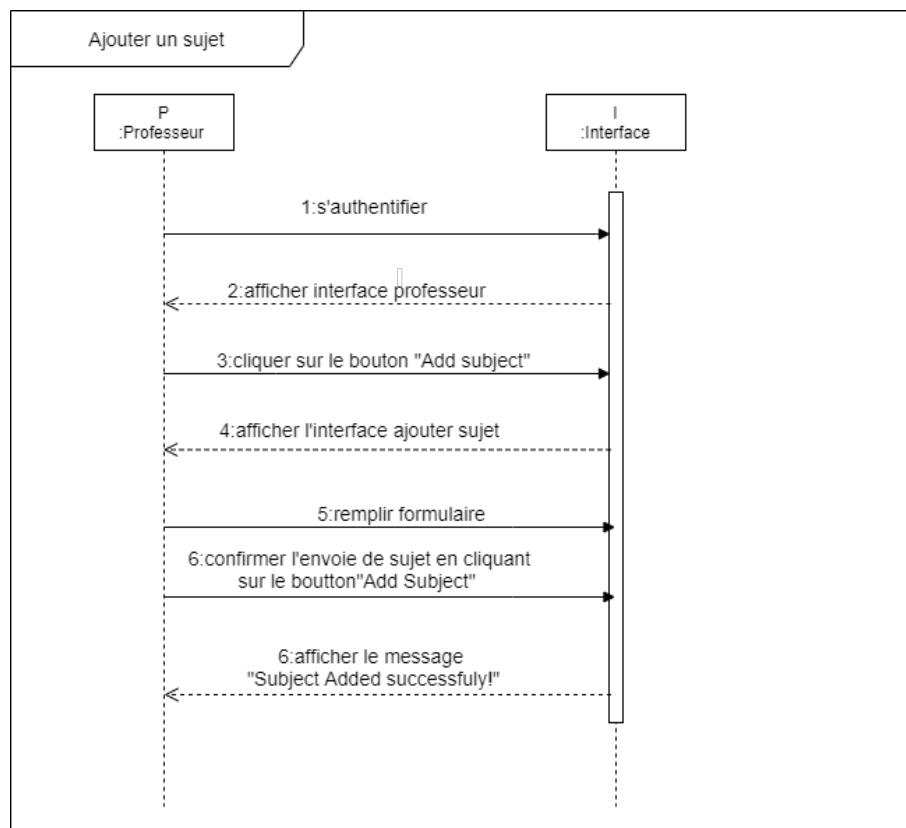


FIGURE 4.6 – Diagramme de séquence objet pour le cas d'utilisation "Ajouter un sujet"

4.4 Conclusion

Dans ce chapitre nous nous sommes focalisés sur la conception architecturale et la conception détaillée de notre système en précisant le diagramme des classes, le diagramme de séquences objet et le diagramme d'activités. Dans le prochain chapitre, nous présentons les étapes de l'implémentation et de réalisation de l'application.

Chapitre 5

Réalisation

5.1 Introduction

Dans ce chapitre nous transformons le modèle conceptuel établi précédemment en des composants logiciels. tout d'abord, nous parlons de l'environnement du travail ainsi que les différentes étapes que nous avons suivis pour construire le système. Enfin, nous clôturons ce rapport par la présentation des quelques interfaces homme/système.

5.2 Environnement de travail

Pour réaliser ce projet nous avons utiliser les outils et les technologies cités ci dessous.

5.2.1 Environnement matériel

Pour le développement de système nous avons utiliser les PCs dont les configurations sont les suivantes :

Marque	Lenovo	ASUS	ASUS
Processeur	Intel(R)Core(TM) i5-8250U	Intel(R)Core(TM) i5-8250U	Intel(R)Core(TM) i5-8250U
Mémoire RAM installée	8,00G	8,00G	8,00G
Disque dur	1To	1To	1To

5.2.2 Environnement logiciel

Pour réaliser ce projet nous avons eu recourt à plusieurs logiciels et bibliothèques que nous les présentons dans ce qui suit :

5.2.2.1 Draw.io

Les diagrammes UML des chapitres Analyse et Conception sont élaborés par la plate-forme en ligne Draw.io. C'est un plate-forme qui offre une interface conviviale et facile à utiliser, de plus il est totalement gratuit et permet d'intégrer ces diagrammes dans les documents et les pdf grâce à d'exportation des différents formats tel que jpg png et gif.[URL2]

5.2.2.2 Visual Studio Code

Visual Studio Code est un éditeur de code gratuit prédefini et optimisé pour la création des applications avec JavaScript ASP.NET et Node.js sur toutes les plates-formes. Il dispose d'un riche écosystème d'extensions pour d'autres langages tel que Python, Java, C++. [URL3]

5.2.2.4 Visual Studio 2019

Visual Studio est une suite d'outils permettant de développer des applications web.[URL4]

5.2.2.5 Angular 11.2.10

Angular est l'un des frameworks TypeScript open-source les plus populaires qui permet de créer des applications Web dynamiques. Grâce à cet outil nous avons développé l'interface de l'application.[URL5]

5.2.2.6 SQL Server

Pour manipuler la base de données de notre application, nous avons utiliser le système relationnelle SQL Server. Ce système est développé par microsoft, il est facile à utiliser et supporte bien la restauration et la récupération de données.[URL6]

5.2.2.7 Python

Python est un langage de programmation interprété, orienté objet et simple à manipuler. De plus il offre des outils de haut niveau et supporte l'usage des modules et des packages afin d'optimiser la productivité des programmeurs. Dans notre projet nous avons utiliser les bibliothèques suivantes :

1. La bibliothèque NLTK

NLTK est l'une des bibliothèques open-source les plus populaires et les plus puissants, dédiée au traitement du langage naturel. C'est une bibliothèque logicielle qui contient les algorithmes du traitement de texte pour la tokenisation, l'analyse sémantique, l'analyse syntaxique, le part-of-speech tagging, le stemming... Grâce à cette bibliothèque nous avons pu faire le prétraitement des nos données.

2. La bibliothèque Gensim

GENSIM est l'une des bibliothèques open-source de Python les plus utilisables. Elle est populaire pour la modélisation de sujets non supervisés et l'extraction automatique de la sémantique d'un sujet.

3. La bibliothèque Tensorflow

C'est une bibliothèque python open-source très populaire qui rend la construction de modèles de ML plus facile.

4. La bibliothèque re

re est l'une des bibliothèques standards de python, c'est un outil puissant qui fait la correspondance des modèles de texte.

5. La bibliothèque TextBlob

C'est une bibliothèque python open-source très puissante pour le traitement des données textuelles telles que l'extraction des phrases, l'analyse des sentiments, etc.

6. La bibliothèque pandas

C'est une librairie python open-source qui effectue une analyse de données pratique, réelle et rapide.

7. La bibliothèque Sklearn

C'est une librairie python open-source dédié pour résoudre les problèmes de machine learning et data science, elle contient des fonctions qui estiment la régression logistique et les algorithmes de classification.

8. La bibliothèque wordcloud

C'est une librairie python open-source qui manipule les textes, elle permet de donner la fréquence de l'importance d'un mot.

9. La bibliothèque String

String est un module populaire de python qui facilite le traitement des chaînes de caractères par des fonctions pré-définies.

5.3 Aperçu sur le travail réalisé

Dans cette partie de chapitre nous exposons les interfaces d'interaction homme-machine de notre application.

5.3.1 Interface d'accueil :

La Figure 5.1 montre l'interface d'accueil de tous les utilisateurs.

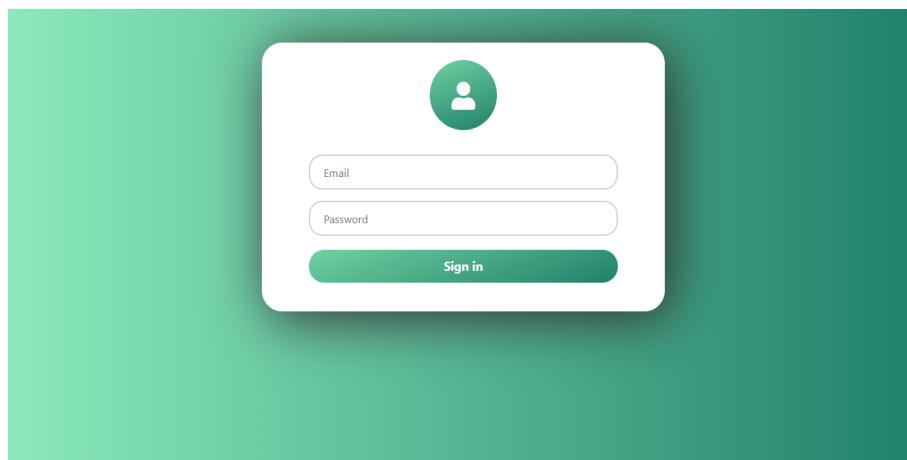


FIGURE 5.1 – Interface d'accueil

5.3.2 Les interfaces du professeur :

Les Figures 5.2 et 5.3 représentent les interfaces graphiques du professeur.

Interface d'ajout d'un sujet :

La Figure 5.2 illustre l'interface d'ajout d'un sujet, elle permet au professeur d'ajouter un sujet accompagné par une description ainsi que le barème.

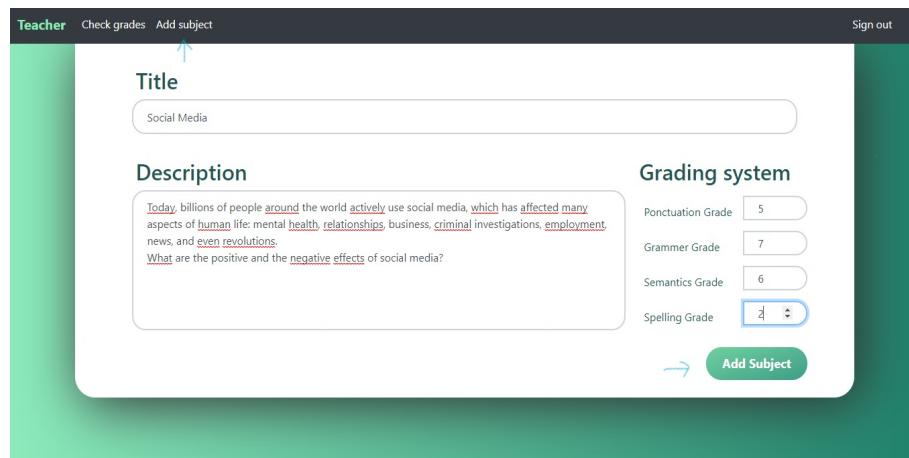


FIGURE 5.2 – Interface d'ajout d'un sujet

Interface de feedback de professeur :

La Figure 5.3 illustre l'interface qui permet à un professeur de consulter les notes des candidats.

The screenshot shows a web-based application interface for teachers. At the top, there are navigation links: 'Teacher', 'Check grades', 'Add subject', and 'Sign out'. Below this, a green sidebar on the left contains a large upward arrow icon. The main content area has a header 'Pollution (surviving in polluted climate)' with a dropdown arrow. Under this, there are three tabs: 'Pollution (problems and solutions)', 'Pollution (surviving in polluted climate)' (which is selected), and 'Social Media'. The selected tab displays a student's text submission:

```

Pollution (surviving in polluted climate)
our mother.

For our health and development, the Earth provides us with so many natural resources. Nevertheless, we become more egoistic and tend to pollute our world over time. If our environment becomes more polluted, we do not know that ultimately too, it will affect our health and the future. We will not be able to survive comfortably on Earth. How can we survive if the climate becomes polluted?

```

To the right of the text, there is a 'Grading system' section with four input fields:

Punctuation Grade	6
Grammer Grade	5
Semantics Grade	6
Spelling Grade	3

Below the grading system, there is a section titled 'Submitted answers' with a single entry:

student1 2021-05-26

With the following details:

- Punctuation Grade :4.84
- Semantics Grade :5
- Spelling Grade :2.97
- Grammer Grade :4.5
- Total Grade :17.310000000000002

FIGURE 5.3 – Interface de feedback de professeur

5.3.3 Les interfaces de candidat

Les Figures 5.4 et 5.5 illustrent les interfaces du candidat.

Interface de liste des sujets

La Figure 5.4 représente comment un candidat peut sélectionner un sujet à partir de la liste des sujets déposés par les professeurs.

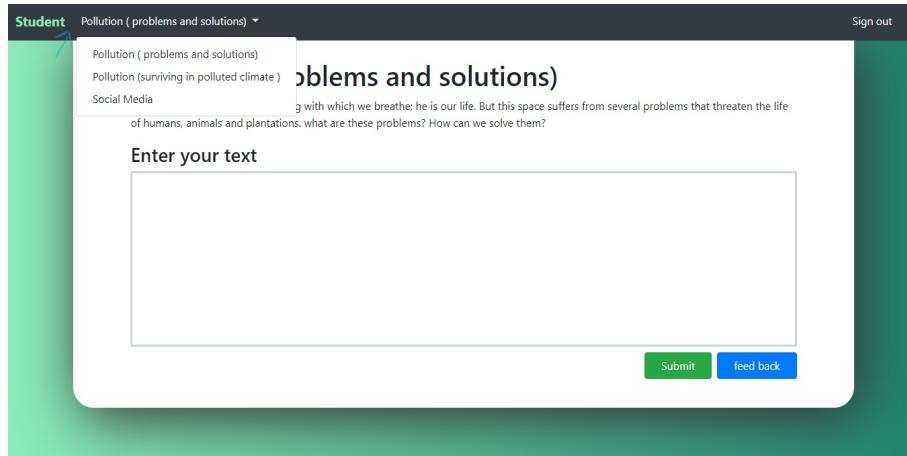


FIGURE 5.4 – Interface de liste des sujets

Interface d'ajout d'un essai

La Figure 5.5 représente l'interface d'ajout d'un essai, le candidat entre son essai puis il clique sur le bouton "Submit".

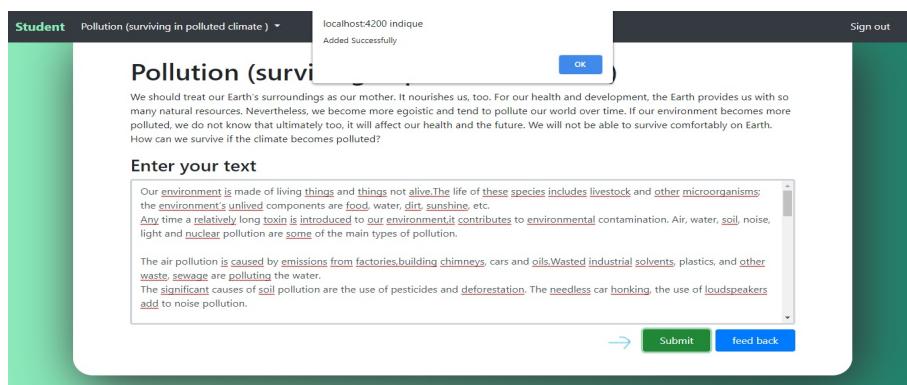


FIGURE 5.5 – Interface d'ajout d'un essai

Interface de feedback de candidat

La Figure 5.6 illustre le feedback sur l'essai en montrant la note de chaque partie et la note totale.

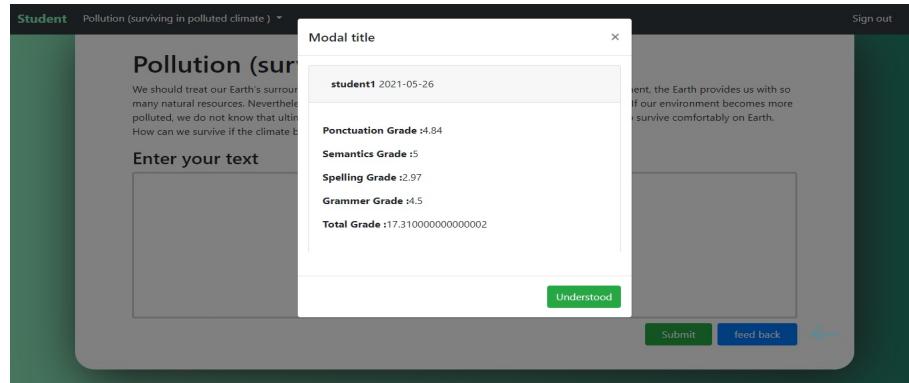


FIGURE 5.6 – Interface de feedback de candidat

5.3.4 Les interfaces de l'administrateur

Les interfaces suivantes montrent les cas utilisations de l'administrateur.

Interface de l'administrateur

La Figure 5.7 montre l'interface de l'administrateur après l'authentification.

The screenshot shows a web-based administrator interface. At the top, there is a dark header bar with the word "Administrator" in white, followed by "View users" and "Add user". On the far right of the header is a "Sign out" link. Below the header is a large green sidebar on the left and a dark sidebar on the right. The main content area contains a table with the following data:

name	email	password	category	actions
admin	admin@gmail.com	admin	Administrator	<button>delete</button> <button>edit</button>
teacher1	teacher1@gmail.com	teacher1	Teacher	<button>delete</button> <button>edit</button>
teacher2	teacher2@gmail.com	teacher2	Teacher	<button>delete</button> <button>edit</button>
student1	student1@gmail.com	student1	Student	<button>delete</button> <button>edit</button>
student2	student2@gmail.com	student2	Student	<button>delete</button> <button>edit</button>
student3	student3@gmail.com	student3	Student	<button>delete</button> <button>edit</button>
student4	student4@gmail.com	student4	Student	<button>delete</button> <button>edit</button>

FIGURE 5.7 – Interface de l'administrateur

Interface d'ajout d'un utilisateur :

La Figure 5.8 montre comment l'administrateur ajoute un utilisateur en cliquant sur "Add user".

The screenshot shows the "Add user" form within the administrator interface. The form is contained within a light gray rounded rectangle. It includes the following fields:

- A text input field containing "student4".
- An email input field containing "student4@gmail.com".
- A dropdown menu set to "Student".
- A password input field showing "*****".
- A large green "Create account" button at the bottom.

FIGURE 5.8 – Interface d'ajout d'un utilisateur

Interface de suppression :

Les Figures 5.9 et 5.10 illustrent comment un administrateur supprime un utilisateur en cliquant sur le bouton "delete".



name	email	password	category	actions
admin	admin@gmail.com	admin	Administrator	<button>delete</button> <button>edit</button>
teacher1	teacher1@gmail.com	teacher1	Teacher	<button>delete</button> <button>edit</button>
teacher	teacher@gmail.com	teacher	Teacher	<button>delete</button> <button>edit</button>
student1	student1@gmail.com	student1	Student	<button>delete</button> <button>edit</button>
student2	student2@gmail.com	student2	Student	<button>delete</button> <button>edit</button>
student3	student3@gmail.com	student3	Student	<button>delete</button> <button>edit</button>
student4	student4@gmail.com	student4	Student	<button>delete</button> <button>edit</button>

FIGURE 5.9 – Interface avant suppression d'un utilisateur



name	email	password	category	actions
admin	admin@gmail.com	admin	Administrator	<button>delete</button> <button>edit</button>
teacher1	teacher1@gmail.com	teacher1	Teacher	<button>delete</button> <button>edit</button>
teacher	teacher@gmail.com	teacher	Teacher	<button>delete</button> <button>edit</button>
student1	student1@gmail.com	student1	Student	<button>delete</button> <button>edit</button>
student2	student2@gmail.com	student2	Student	<button>delete</button> <button>edit</button>
student3	student3@gmail.com	student3	Student	<button>delete</button> <button>edit</button>

FIGURE 5.10 – Interface après suppression d'un utilisateur

Conclusion et perspectives

Ce projet qui a duré presque trois mois, a été réalisé pendant la deuxième semestre dans le cadre des Projets de Développement et de Conception pour les étudiants de la deuxième année de l'ENSI. Pendant cette période nous avons atteint nos objectifs qui manifestent dans l'amélioration de nos compétences techniques et théoriques. Tout au long de cette période, nous avons eu l'occasion de concevoir et de développer une application web publique qui note automatiquement un essai écrit en anglais en utilisant les techniques de NLP et les algorithmes de ML.

Ce rapport est composé de cinq chapitres présentant les étapes que nous avons suivis pour réaliser le projet. Dans le premier chapitre, nous avons défini les normes de notre application. De plus, nous avons présenté quelques systèmes qui se trouvent sur le marché puis nous avons dégagé des critiques sur les moteurs existants. Dans le chapitre suivant nous avons défini les acteurs de notre application, les besoins fonctionnels par acteur et les besoins non fonctionnels de système. De plus nous avons modélisé quelques diagrammes pour présenter les besoins. Au niveau de troisième chapitre, nous avons montré comment s'inscrit notre application dans le domaine de Machine Learning et l'intelligence artificielle en présentant les concepts nécessaires à la solution de problème. Le quatrième chapitre intitulé "Conception Détailée" dévoile l'approche conceptuel de notre application à travers les diagrammes UML de la conception architecturale et la conception détaillée. Nous clôturons ce rapport par le cinquième chapitre qui sert à présenter les environnements de travail et les interfaces graphiques qui montrent le fonctionnement de l'application.

Pendant cette expérience formatrice et instructive nous avons appris plusieurs découvertes. En premier lieu, nous avons appris comment travailler en équipe et gérer le temps. En deuxième lieu, nous avons bien maîtrisé le langage de programmation Python et ses librairies relatives au traitement

automatique du langage naturel, de plus nous avons découvert comment manipuler les bases des données et développer des applications web grâce aux outils suivants : SQL Server, Angular, Visual Studio Code et Visual Studio. Enfin, nous pouvons dire que nous avons atteint notre objectif principal. Cependant, notre projet peut évidemment s'améliorer en ajoutant autres langues que la langue anglaise.

Netographie

[URL1] :<https://www.nltk.org/book/ch08.html>

[URL2] : <https://www.draw.io/index.html>

[URL3] : <https://code.visualstudio.com/>

[URL4] : <https://visualstudio.microsoft.com/fr/downloads/>

[URL5] : <https://angular.io/>

[URL6] : <https://www.microsoft.com/fr-fr/sql-server/sql-server-downloads>