

Bio-informatics and Data Sciences

Mémoire de Master

STAGE DE FIN D'ÉTUDES

Création d'une Plateforme de Suivi des Stages

Intégration d'un Bot pour l'Assistance et le
Suivi Automatisé des Stagiaires

Fourniture d'Offres de Stages, d'Informations Institutionnelles et
de Suivi des stagiaires

DEFENDED ON: SEPTEMBER 11, 2024

Supervised By **Pr. Badir Hassan**

Realised By:
ELHAJJAJI IMAD

Jury Members:

Pr. Ahmed DRISSI

ENSA of Tangier, Morocco

President

Pr. Houda BEL MOKADEM,

Labtic ,Ensa Tangier, Morocco

Rapporteur

Pr. Hassan Badir

IDS Research Team. ENSA of Tangier

Encadrant

Titre: Développement d'une Application de Suivi des Stages avec Intégration d'un Chatbot pour la réponse automatique.

Mots clés: Stages, Stagiaires, Encadrant, Chatbot, , Gestion des stages, Application de suivi, Communication, Avancement des stages

Résumé: Ce projet vise à concevoir et à déployer une application avancée pour le suivi en temps réel des stages effectués par les étudiants de l'École Nationale des Sciences Appliquées de Tanger (ENSA Tanger). L'objectif principal est de créer une plateforme numérique qui facilite la gestion, la supervision et l'évaluation des stages de manière systématique et détaillée. Cette application permettra non seulement de suivre l'évolution des stages mais aussi d'évaluer leur réussite en temps réel, grâce à une série de fonctionnalités innovantes.

L'application comprendra des tableaux de bord interactifs pour les différents utilisateurs, incluant les étudiants, les encadrants académiques et les tuteurs d'entreprise. Les étudiants pourront mettre à jour leurs rapports de stage, suivre leurs tâches et échéances, et visualiser leur progression à travers des graphiques et des statistiques. Les encadrants auront accès à des outils pour suivre la performance des étudiants, fournir des commentaires et générer des rapports de suivi détaillés. Les tuteurs d'entreprise pourront aussi suivre les progrès des stagiaires, échanger avec eux via la plateforme et fournir un retour d'information directement accessible aux étudiants et à l'administration.

Un aspect clé de cette application est l'intégration d'un chatbot intelligent. Ce chatbot, alimenté par une technologie de traitement du langage naturel avancée, sera capable de répondre instantanément aux questions des utilisateurs. En fonction du rôle de l'utilisateur (étudiant,

encadrant, ou tuteur), le chatbot fournira des réponses personnalisées concernant les aspects variés du suivi des stages, tels que les procédures administratives, les exigences de rapport, et les conseils pour améliorer la performance en stage. Par exemple, un étudiant pourrait demander des conseils sur la rédaction de son rapport de stage, tandis qu'un encadrant pourrait obtenir des informations sur les critères d'évaluation des stages.

En plus de ces fonctionnalités de base, l'application intégrera également des mécanismes d'alerte et de notification pour informer les utilisateurs des mises à jour importantes, des échéances imminentes et des actions requises. Des outils d'analyse et de génération de rapports permettront également d'extraire des données sur les tendances et les performances globales des stages, fournissant ainsi des insights précieux pour améliorer les processus de gestion des stages.

Ce projet ne se contente pas de moderniser le suivi des stages ; il aspire à transformer la manière dont les informations sont collectées, traitées et utilisées pour enrichir l'expérience de stage des étudiants, améliorer la communication entre toutes les parties impliquées et optimiser la gestion des stages au sein de l'ENSA Tanger. En offrant une solution numérique intégrée et intelligente, ce projet contribuera à une gestion plus efficace et plus réactive des stages, tout en favorisant une meilleure coordination et un meilleur soutien aux stagiaires tout au long de leur expérience professionnelle.

Remerciements



Nous tenons à exprimer notre profonde gratitude à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce projet de recherche.

Nous souhaitons tout particulièrement remercier notre encadrant, **Mr Hassan Badir**, Professeur Associé au Département de Systèmes d'Information et de Communication de l'École Nationale des Sciences Appliquées de Tanger, pour son soutien indéfectible, ses précieux conseils et son encadrement attentif tout au long de ce projet. Ses orientations ont été déterminantes pour la réussite de cette recherche.

Nos remerciements s'adressent également à **Mme Rachida Fissoune**, pour son accompagnement constant et ses conseils éclairés, qui ont été d'une aide inestimable tout au long de ce parcours académique.

Nous exprimons aussi notre gratitude aux membres du jury, **Mr Hassan Drissi** et **Mme Houda Bel Moukadem**, pour leur disponibilité, leur rigueur et leurs remarques constructives, qui ont enrichi ce travail et permis d'en améliorer la qualité.

À tous, merci pour votre précieuse contribution.

Dédicace



Ces lettres je les cède à ma famille. Ils m'ont appris à ne jamais perdre confiance en soi. Ils m'ont convaincu que malgré tous les obstacles on peut toujours avancer. Tout ce qu'on doit avoir c'est une inépuisable volonté.

Je me suis toujours dit que je ne peux rien leur offrir. Cependant, ils m'ont fait sentir que juste le fait de vouloir apprendre et avancer dans sa vie. Juste le fait de vouloir un jour être là pour eux, même si j'y n'arrive pas. C'est déjà la paix pour eux, car ils savent que j'y arriverai. Ils me font comprendre que j'arriverai un jour à être mieux, et pas pour eux mais pour moi même.

Ils sont la raison que j'apprend, j'écris, je rêve, je pense, j'espère et je désire. Merci père, mère, petite soeur...

Contents

Acknowledgements	2
Dédicace	4
List of Figures	7
1 spécification et analyse des besoins	10
1.1 Etude de l'existant	11
1.1.1 Description de l'existant	11
1.1.2 Critique de l'existant	11
1.1.3 Solution proposée	11
1.2 Besoins Fonctionnels et Non Fonctionnels	11
1.2.1 Besoins Fonctionnels	12
1.2.2 Besoins Non Fonctionnels	12
1.3 Cadre de Projet	12
1.3.1 Présentation de l'application SStage	13
2 Conception	15
2.1 Conception Générale	16
2.2 Conception détaillée	18
2.3 Maquette	25
2.4 Conclusion	27
3 Réalisation	28
3.1 Introduction aux Technologies Utilisées	30
3.2 Page d'Accueil	32
3.3 Section Admin	33
3.4 section de Stagiaire	38
3.5 section Encadrant	41
3.6 Conclusion	44
4 Intégration de bot pour la réponse automatique des stages	45
4.1 Collecte des Données	46

4.2	Création du Fichier data.py	47
4.3	Importation des Bibliothèques	47
4.4	Initialisation du Lemmatiser	48
4.5	Définition de la Fonction de Prétraitement des Données	48
4.6	Tokenisation et Construction des Listes	49
4.7	Lemmatisation et Nettoyage des Mots	49
4.8	Création des Vecteurs de Caractéristiques (Bag of Words)	50
4.9	Application de la Fonction et Création des Données d'Entraînement	51
4.10	Entraînement de Modèle	51
4.11	Prédiction des Tags	53
4.12	Génération des Réponses	54
4.13	Résumé des Performances	55
4.14	Conclusion sur l'intégration du bot dans la plateforme de stage	57
	Bibliography	58

List of Figures

1.1	Schéma du cadre de projet SStage	13
2.1	Schéma du modèle de cycle de vie	16
2.2	Architecture 3-tiers de l'AGSS	17
2.3	Diagramme de cas d'utilisation pour l'étudiant	19
2.4	Description du cas d'utilisation « Ajouter un document »	19
2.5	Description du cas d'utilisation « Ajouter un document »	20
2.6	Diagramme de cas d'utilisation pour l'encadrant	20
2.7	Diagramme cas utilisation affecter une tache	21
2.8	affecter une tache dans une avancement	21
2.9	Diagramme cas utilisation D'administrateur	22
2.10	Diagramme d'activité : Affectation des tâches hebdomadaire	23
2.11	Diagramme de séquence : Affectation des tâches hebdomadaire	24
2.12	Diagramme de séquence : Validation d'un stage	24
2.13	Maquette : Structure de l'application	26
2.14	Carte Mentale : Exemple de cohérence visuelle	26
3.1	l'application Frontend	31
3.2	l'application Backend	31
3.3	Page d'Accueil de l'application	32
3.4	Modal de Connexion pour accéder à l'espace privé	33
3.5	Statistiques de l'application et progression des données	34
3.6	Gestion des utilisateurs dans l'application	34
3.7	l'état des utilisateur en fil de temps	34
3.8	Gestion des stages selon la filière et le niveau	35
3.9	Gestion des stages : validés selon la filière et le niveau	35
3.10	Étudiants avec encadrant	36
3.11	Étudiants sans encadrant	36
3.12	Étudiants sans stage	36
3.13	Étudiants sans rapport	37
3.14	Notes des stages et évaluations	37
3.15	Gestion des documents des utilisateurs	37

3.16	Page d'accueil du stagiaire avec vue d'ensemble des stages et notifications . . .	38
3.17	Modal d'ajout ou de mise à jour des informations professionnelles du stagiaire .	39
3.18	Modal d'ajout d'un nouveau stage pour le stagiaire	39
3.19	gestion des documents : ajout des fichiers principaux	40
3.20	gestion des documents : ajout des fichiers d'avancement	40
3.21	Illustration de la communication entre stagiaires et professeurs	40
3.22	Page d'accueil de l'encadrant avec vue d'ensemble des stagiaires et notifications	41
3.23	Page des listes des stages pour les encadrants, avec détails des stages et options de gestion	42
3.24	Interface de messagerie pour les encadrants, permettant des discussions en temps réel avec les étudiants	43
4.1	Scraping des données depuis LinkedIn avec Selenium	46
4.2	Structure du fichier data.py illustrant les intents, les tags, et les réponses. . . .	47
4.3	Bibliothèques utilisées pour le traitement et l'entraînement des données	47
4.4	Initialisation du lemmatiseur avec WordNetLemmatizer	48
4.5	Définition de la fonction de prétraitement des données	48
4.6	Tokenisation et construction des listes	49
4.7	Lemmatisation et nettoyage des mots	49
4.8	Création des vecteurs de caractéristiques (Bag of Words)	50
4.9	prétraitement et création des données d'entraînement	51
4.10	le processus d'entraînement de model	51
4.11	la fonction pour prédiction de tag	53
4.12	précision de entrainement	56
4.13	précision de entrainement	56

Chapter 1 spécification et analyse des besoins

Chapter Contents

1.1	Etude de l'existant	11
1.1.1	Description de l'existant	11
1.1.2	Critique de l'existant	11
1.1.3	Solution proposée	11
1.2	Besoins Fonctionnels et Non Fonctionnels	11
1.2.1	Besoins Fonctionnels	12
1.2.2	Besoins Non Fonctionnels	12
1.3	Cadre de Projet	12
1.3.1	Présentation de l'application SStage	13

Abstract

Dans ce chapitre, nous mettons le sujet dans son cadre général. Par la suite, nous abordons l'étude de l'existant du projet, suivie d'une critique pour dégager les contraintes à respecter pendant la réalisation de notre projet.

Ainsi, ce chapitre présente l'ensemble des besoins, qu'ils soient fonctionnels ou non fonctionnels, nécessaires à la réalisation du projet.

Dans ce chapitre, nous nous concentrons sur la réalisation d'une plateforme de suivi des stages, en tenant compte des besoins spécifiques et des critiques pour son développement. Cette plateforme a pour objectif de permettre un suivi efficace des étudiants en stage, ainsi que de les aider à trouver des opportunités de stage. Voici les principaux besoins et critiques pour la réalisation de cette plateforme

1.1 Etude de l'existant

1.1.1 Description de l'existant

Actuellement, les plateformes de suivi de stage offrent les fonctionnalités suivantes :

- **Gestion des Informations de Stage** : Détails sur les stages.
- **Attribution d'un Encadrant** : Supervision par un encadrant dédié.
- **Gestion des Documents** : Rapport descriptif, rapport final, présentation PowerPoint.
- **Absence de Communication Intégrée** : Pas de contact direct entre stagiaire, encadrant, et entreprise.

1.1.2 Critique de l'existant

La gestion actuelle des stages présente plusieurs limitations :

- **Communication non optimale** : Absence de communication efficace entre stagiaires, encadrants, et entreprises.
- **Gestion des Stages** : Suivi limité aux documents, sans interaction continue ni suivi régulier.

1.1.3 Solution proposée

Pour améliorer la gestion des stages, les solutions suivantes sont proposées :

- **Gestion hebdomadaire** : Suivi régulier avec e-mails.
- **IA intégrée** : Aide à la recherche de stages et gestion intelligente du temps.
- **Amélioration de la communication** : Outils pour faciliter les échanges.

1.2 Besoins Fonctionnels et Non Fonctionnels

Dans cette section du chapitre, nous nous intéressons aux besoins des utilisateurs à travers les spécifications fonctionnelles et non fonctionnelles afin de développer une application de qualité conforme aux attentes du client.

1.2.1 Besoins Fonctionnels

Les besoins fonctionnels se divisent en plusieurs catégories importantes :

a) **L'échange de l'information et de la documentation**

- **Gestion de la documentation** : Permet aux utilisateurs d'ajouter, modifier, consulter des documents et de gérer les commentaires associés.
- **Affectation des tâches** : Chaque encadrant peut assigner des tâches aux étudiants avec des dates limites pour la remise du travail.

b) **La gestion du cahier du stage**

- **Suivi des tâches** : Ajout, modification et consultation des tâches réalisées par les étudiants chaque jour.

c) **Le Forum**

- **Discussion entre utilisateurs** : Permet la création de sujets et l'ajout de réponses pour favoriser les échanges.

d) **La gestion du BackOffice**

- **Gestion des stages** : Ajout, modification et suppression des stages, gestion des types de stages, et affectation des membres de jury.
- **Gestion des utilisateurs** : Inclut la gestion des encadrants, des étudiants, des membres de jury, et des profils utilisateurs.

1.2.2 Besoins Non Fonctionnels

Les besoins non fonctionnels incluent les aspects suivants :

- **Performance** : Assurer que l'application reste rapide et réactive.
- **Sécurité** : Protéger les données des utilisateurs contre les accès non autorisés.
- **Accessibilité** : Rendre l'application utilisable par tous, y compris les personnes handicapées.
- **Scalabilité** : Permettre à l'application de gérer une augmentation du nombre d'utilisateurs et de données.
- **Ergonomie et bon IHM** : L'application doit être facile à utiliser avec peu d'effort de la part de l'utilisateur.
- **Efficacité** : Permettre l'accomplissement des tâches avec un minimum de manipulations.

1.3 Cadre de Projet

Dans cette section, nous décrivons le cadre du projet de développement d'une application web nommée ****SStage****. Cette application a été conçue pour améliorer le suivi des stages des étudiants, en facilitant la gestion, la communication, et la documentation associée aux stages.

1.3.1 Présentation de l'application SStage

L'application web **SStage** a été développée pour répondre aux besoins suivants :

- **Suivi des Stages :**
 - Permet aux utilisateurs de suivre les progrès des stagiaires.
 - Gère les documents liés aux stages.
 - Assure une communication fluide entre les parties impliquées.
- **Gestion de la Documentation :**
 - Offre des fonctionnalités pour ajouter, modifier, et consulter les documents de stage.
- **Communication :**
 - Facilite les échanges entre les stagiaires, les encadrants, et les entreprises.
 - Intègre des forums et des messageries pour la communication.

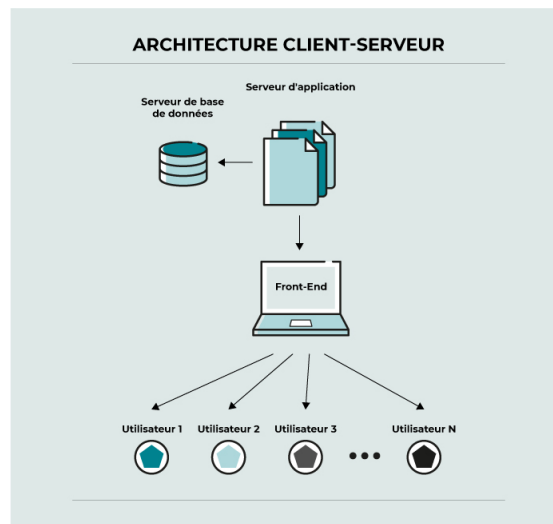


Figure 1.1: Schéma du cadre de projet SStage

1.3.2 Conclusion

Ce chapitre se concentre sur la réalisation de l'application "Sstage", en détaillant les technologies utilisées et les différentes sections de l'application. La partie administrative permet de gérer les utilisateurs et de suivre l'évolution des stages au fil du temps. Les sections dédiées aux encadrants facilitent le suivi et l'encadrement des stages sous leur responsabilité. Enfin, les pages destinées aux stagiaires offrent la possibilité d'ajouter et de gérer leurs stages, ainsi que de soumettre les documents requis, comme les rapports et autres fichiers relatifs au stage.

Chapter 2 Conception

Chapter Contents

2.1	Conception Générale	16
2.1.1	Le Modèle de Cycle de Vie	16
2.1.2	Méthodologie adoptée	16
2.1.3	Concept et architecture de l'AGSS (Application de Gestion et Suivi des Stages)	17
2.2	Conception détaillée	18
2.2.1	Les diagrammes de cas d'utilisation	18
2.2.2	Diagramme d'activité	22
2.2.3	Diagramme de séquence	23
2.2.4	Diagramme de classe	25
2.3	Maquette	25
2.4	Conclusion	27

Abstract

La phase de conception constitue un élément crucial dans le cycle de vie de développement d'une application. Elle a pour but de transformer le modèle conceptuel du système, qui a été élaboré lors de l'étape précédente d'analyse des besoins, en modèles détaillés de l'architecture du système. Cette étape est essentielle pour clarifier et organiser les spécifications fonctionnelles et non fonctionnelles en une structure cohérente qui guide le développement ultérieur.

En somme, la phase de conception vise à élaborer une feuille de route précise pour le développement, en s'assurant que toutes les exigences sont couvertes et que l'architecture du système est optimisée pour répondre aux besoins identifiés lors de l'analyse.

2.1 Conception Générale

2.1.1 Le Modèle de Cycle de Vie

Le modèle de cycle de vie d'une application est un cadre structuré qui décrit les étapes et les processus nécessaires pour développer une application depuis sa conception jusqu'à sa mise en œuvre et son maintien. Ce modèle assure que chaque phase du développement est systématiquement abordée et que les exigences du projet sont satisfaites de manière efficace et cohérente.

a. Présentation du Modèle

La présentation du modèle de cycle de vie décrit comment chaque phase du développement est abordée, en commençant par la conception initiale jusqu'à la mise en œuvre et le support continu. Ce modèle offre une vue d'ensemble des processus impliqués et assure une approche structurée pour le développement d'applications.

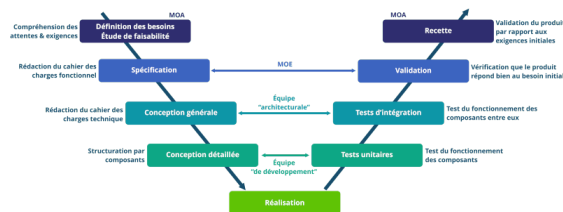


Figure 2.1: Schéma du modèle de cycle de vie

a. Description du modèle

La présentation du modèle de cycle de vie décrit comment chaque phase du développement est abordée, en commençant par la conception initiale jusqu'à la mise en œuvre et le support continu. Ce modèle offre une vue d'ensemble des processus impliqués et assure une approche structurée pour le développement d'applications.

2.1.2 Méthodologie adoptée

Pour maximiser la compréhension et assurer la bonne exécution du projet, nous avons opté pour la **méthodologie** de développement **RUP** (*Rational Unified Process*). Ce choix repose sur le fait que cette approche est un standard reconnu, offrant une **conception** détaillée qui prend en compte chaque **aspect** de la conception, y compris les systèmes existants. Le **RUP** se distingue

par sa capacité à offrir une vue d'ensemble sur le **système d'information** dans son ensemble, ce qui le rend particulièrement adapté aux projets initiés à partir de zéro.

Cette méthodologie repose sur le langage de modélisation **UML** (*Unified Modeling Language*), que nous avons choisi pour guider tout notre travail. En réalité, **UML** n'est pas une méthode ou un processus, mais bien un **langage** de modélisation. **UML** se caractérise par une **notation** très précise et des règles grammaticales spécifiques qui permettent de créer des modèles de **logiciels**.

L'**UML** propose un ensemble varié d'éléments de **notation graphique**, permettant de modéliser divers aspects tels que les **classes**, les **composants**, les **nœuds**, les **activités**, le **workflow**, les **cas d'utilisation**, les **objets**, et les **états**, ainsi que les relations entre ces éléments. De plus, **UML** permet de personnaliser ces modèles à travers des **extensions personnelles** sous forme d'éléments stéréotypés.

2.1.3 Concept et architecture de l'AGSS (Application de Gestion et Suivi des Stages)

Le projet consiste à concevoir une application web avec une architecture 3-tiers. Cette architecture est conçue pour séparer les responsabilités et faciliter la gestion des différentes couches de l'application. L'architecture 3-tiers est une approche bien établie pour le développement d'applications web, assurant la modularité, la scalabilité et la sécurité du système.

- **Client** : Il s'agit de l'interface utilisateur où les utilisateurs interagissent avec l'application via un navigateur web. Cette couche est responsable de l'affichage des données et de la capture des interactions de l'utilisateur.
- **Middleware** (Serveur d'application) : Cette couche sert d'intermédiaire entre le client et le serveur de données. Elle gère la logique métier de l'application, traite les requêtes des utilisateurs, et communique avec la base de données pour récupérer ou sauvegarder des informations.
- **Serveur de données** : Il s'agit de la base de données où toutes les informations nécessaires à l'application sont stockées. Cette couche est responsable de la gestion des données, y compris leur stockage, récupération, et manipulation.

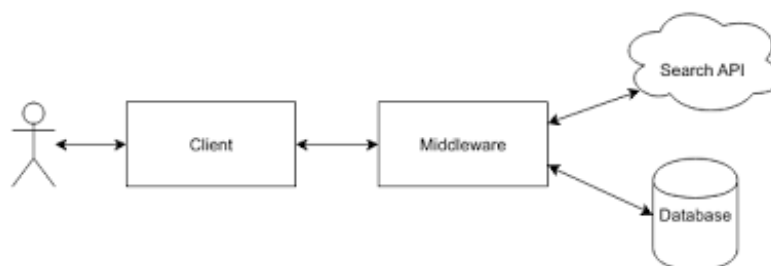


Figure 2.2: Architecture 3-tiers de l'AGSS

2.2 Conception détaillée

2.2.1 Les diagrammes de cas d'utilisation

L'une des étapes cruciales dans la conception d'un système est la définition des cas d'utilisation fondamentaux. L'objectif principal de cette étape est d'identifier les principaux cas d'utilisation. Dans cette partie, nous nous concentrons sur la réalisation des diagrammes de cas d'utilisation. Ces diagrammes décrivent de manière précise les besoins du client final et spécifient le comportement attendu du système à développer. En général, un diagramme de cas d'utilisation modélise un service rendu par le système.

a. Présentation des acteurs

Dans cette section, nous présentons les principaux acteurs impliqués dans le système et leurs rôles respectifs. Trois acteurs principaux sont identifiés : l'étudiant, l'encadrant et l'administrateur.

→ **Étudiant en Master :**

- Peut ajouter un stage
- Ajouter des documents relatifs au stage
- Participer au forum
- Chercher dans la base de données des stages

→ **Étudiant en Ingénieur :**

- Peut ajouter un stage
- Ajouter des documents
- Participer au forum
- Chercher dans la base de données des stages

→ **Encadrant :**

- Ajouter des états d'avancement
- Participer au forum
- Chercher dans la base de données
- Attribuer la date de soutenance
- Noter les stages encadrés

→ **Administrateur :**

- Paramétrer l'application
- Gérer les utilisateurs
- Gérer les stages
- Gérer les paramètres globaux du système

a. Description de le cas d'utilisation

Dans cette section, nous présentons les principaux acteurs impliqués dans le système et leurs rôles respectifs. Trois acteurs principaux sont identifiés : l'étudiant, l'encadrant et

l'administrateur.

b. Diagramme de cas d'utilisation

Le diagramme ci-dessous illustre les cas d'utilisation spécifiques à l'étudiant. Ce diagramme fournit une vue d'ensemble des interactions entre l'étudiant et le système, mettant en évidence les fonctionnalités accessibles à cet acteur.

1. Diagramme cas d'utilisation étudiant

- Permet à l'utilisateur d'ajouter des documents relatifs à son stage.
- Inclut le téléchargement et l'association des fichiers avec le stage en cours.

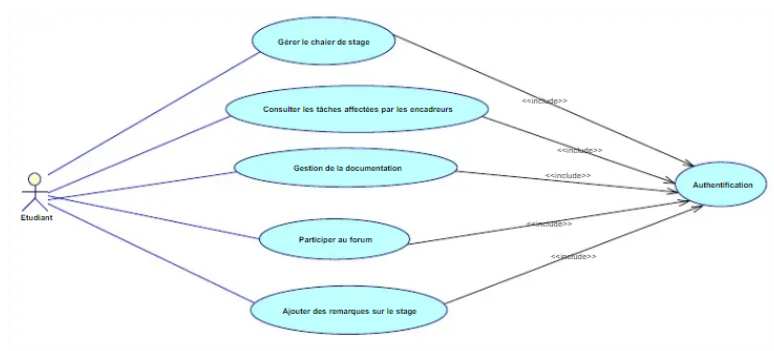


Figure 2.3: Diagramme de cas d'utilisation pour l'étudiant

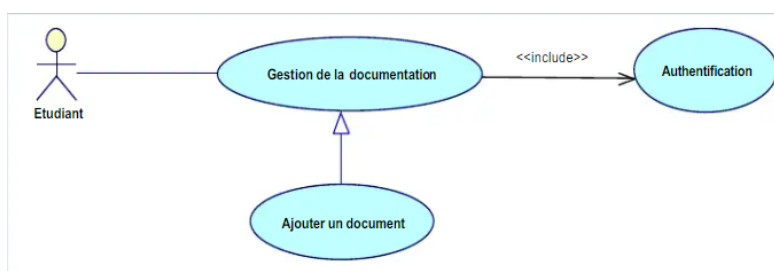


Figure 2.4: Description du cas d'utilisation « Ajouter un document »

Pour ajouter un document, l'étudiant doit d'abord s'authentifier sur la plateforme. Une fois connecté, il peut accéder à la page dédiée aux documents. Sur cette page, l'étudiant a la possibilité de télécharger et d'insérer des documents en fonction de leur type (par exemple, rapports de stage, fiches de suivi, etc.). Il sélectionne le fichier à ajouter, choisit le type approprié, et valide l'action. Le système procède alors au téléchargement du document et l'associe au stage en cours, le rendant disponible pour une consultation ultérieure.

SOMMAIRE d'authentification	
Titre :	Ajouter un document
But :	L'échange de documentation entre l'étudiant et les autres acteurs
Résumé :	L'étudiant sélectionne le document et confirme l'action. Le système télécharge le fichier sur le serveur et l'enregistre dans la base de données
Acteur :	étudiant
Description Des Enchainement	
Pré condition	Post condition
-l'étudiant doit être authentifié	-le document est déposé
Enchainement nominal	
-l'étudiant parcourt le document -l'étudiant valide l'action -le système vérifie le type de document -le système télécharge le document dans le serveur -le système mise à jour la base de données	

Figure 2.5: Description du cas d'utilisation « Ajouter un document »

1.1. Diagramme de cas d'utilisation pour l'encadrant



Figure 2.6: Diagramme de cas d'utilisation pour l'encadrant

L'encadrant joue un rôle clé dans le suivi et l'évaluation du stage de l'étudiant. Il a la possibilité d'ajouter des remarques sur les travaux effectués, de gérer la documentation associée au stage, et de consulter les états d'avancement hebdomadaires. L'encadrant a également accès aux fichiers principaux du stage tels que le rapport descriptif, le rapport final, et les présentations (PPT). De plus, il participe activement au forum de communication, facilitant les échanges entre les stagiaires et contribuant à leur

encadrement global.

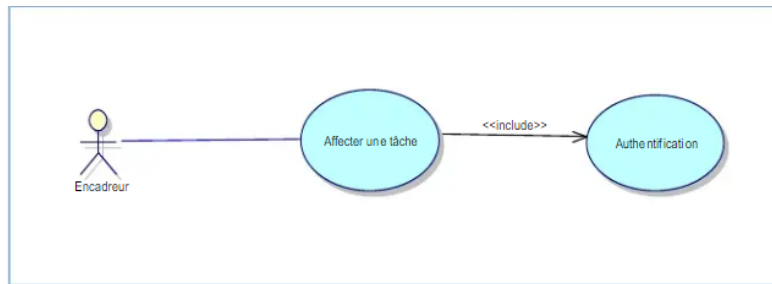


Figure 2.7: Diagramme cas utilisation affecter une tâche

SOMMAIRE d'authentification	
Titre :	Affecter une tâche
But :	Assistant les étudiants pour bien passé leur projet de fin étude
Résumé :	L'encadreur remplir le formulaire d'affectation d'une tâche et valide l'action. Le système enregistre la tâche dans la base de données
Acteur :	Encadreur
Description Des Enchainement	
Pré condition	Post condition
L'encadreur doit être authentifié	-la tâche est affectée
Enchainement nominal	
-l'encadrant remplit le formulaire. -l'encadrant valide l'affectation. -le système enregistre la tâche dans la base de donnée et envoie un email vers les stagiaires dans chaque stage pour les informer d'effectuer les tâches dans un délai précis	

Figure 2.8: affecter une tâche dans une avancement

En plus de ces tâches, l'encadrant est également responsable de la validation des stages après avoir consulté l'ensemble des documents soumis par les stagiaires. Il a la possibilité de valider ou de rejeter le stage en fonction de la qualité des travaux présentés. Par ailleurs, l'encadrant peut consulter les tâches qui ont été assignées aux stagiaires pour suivre leur progression et s'assurer qu'ils respectent les échéances prévues.

1.1.1. Diagramme de cas d'utilisation pour l'Administrateur



Figure 2.9: Diagramme cas utilisation D'administrateur

L'administrateur a la responsabilité de gérer l'ensemble des stages au sein de l'organisation. Il peut consulter et modifier tous les détails relatifs à chaque stage, y compris les informations sur les stagiaires, les encadrants, et les documents associés. De plus, l'administrateur a accès à des outils statistiques qui lui permettent de suivre et d'analyser les performances et les progrès de chaque encadrant et stagiaire. Cela inclut la possibilité de générer des rapports sur les stages, d'évaluer les taux de réussite, et d'obtenir des informations détaillées sur l'état d'avancement des projets. Grâce à cette vue d'ensemble, l'administrateur peut prendre des décisions informées et assurer une gestion efficace des stages.

2.2.2 Diagramme d'activité

Le diagramme d'activité est un diagramme comportemental d'UML, permettant de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements parallélisables (Multi-threads ou Multi-processus). Il permet de modéliser un processus interactif, global ou partiel pour un système donné (logiciel, système d'information). Il est recommandé pour exprimer une dimension temporelle sur une partie du modèle.

1. Diagramme d'activité : Affectation des tâches hebdomadaire par l'encadrant

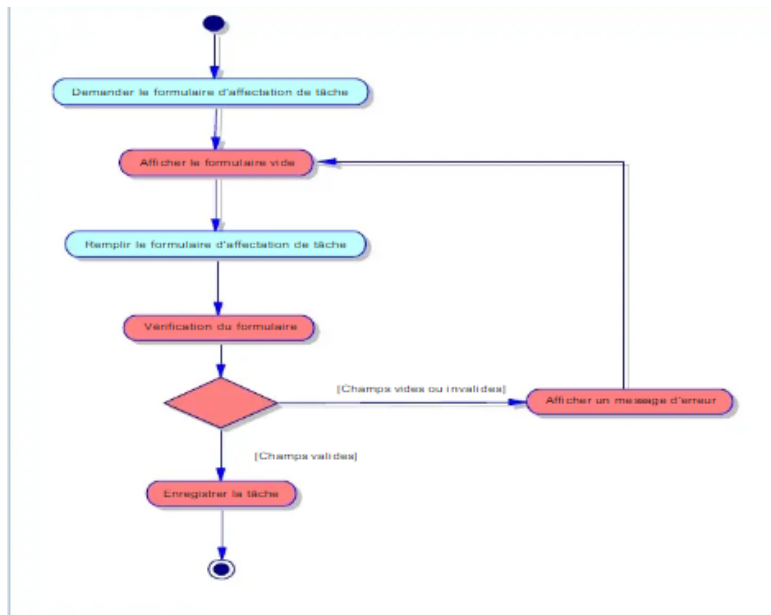


Figure 2.10: Diagramme d'activité : Affectation des tâches hebdomadaire

2.2.3 Diagramme de séquence

Le diagramme de séquence est un diagramme comportemental d'UML utilisé pour représenter les interactions entre objets ou composants d'un système au fil du temps. Il montre comment les messages sont échangés entre les objets et la séquence dans laquelle ces messages sont envoyés. Ce diagramme est utile pour illustrer le déroulement d'un processus ou d'une opération spécifique, mettant en évidence l'ordre et la durée des interactions.

a. Diagramme de séquence : Affectation des tâches hebdomadaire par l'encadrant

Dans ce diagramme, l'encadrant, qui joue un rôle crucial dans la supervision et le suivi des stagiaires, commence par remplir un formulaire spécifique pour attribuer une tâche hebdomadaire. Ce formulaire contient des informations clés telles que la nature de la tâche, les objectifs à atteindre, les critères d'évaluation, ainsi que la date de début et de fin de la mission. L'encadrant peut également inclure des instructions détaillées, des ressources nécessaires (documents, outils, contacts) et des notes supplémentaires pour clarifier les attentes.

Une fois le formulaire complété, l'encadrant procède à sa validation interne pour s'assurer que toutes les informations requises sont correctement renseignées. Il vérifie que les tâches attribuées sont adaptées au niveau de compétence de chaque stagiaire et alignées avec les objectifs du stage. Cette étape de validation est cruciale pour éviter toute confusion ou malentendu ultérieur et pour garantir que les stagiaires ont toutes les informations nécessaires pour accomplir leurs tâches efficacement.

Ensuite, le formulaire rempli est envoyé électroniquement aux stagiaires concernés. Les

stagiaires reçoivent une notification les informant de la nouvelle tâche à réaliser. Ce processus de notification peut se faire via un e-mail, un système de gestion de tâches ou une plateforme de collaboration en ligne. Les stagiaires accèdent alors au formulaire, prennent connaissance des instructions et des attentes de l'encadrant, et commencent à travailler sur la tâche assignée.

Par la suite, les stagiaires effectuent la tâche en respectant les directives fournies et en suivant le calendrier établi. Ils peuvent, si nécessaire, poser des questions ou demander des clarifications à l'encadrant via des outils de communication tels que des messageries internes ou des réunions en ligne. Cette interaction continue permet de s'assurer que les stagiaires restent sur la bonne voie et comprennent pleinement leurs responsabilités.

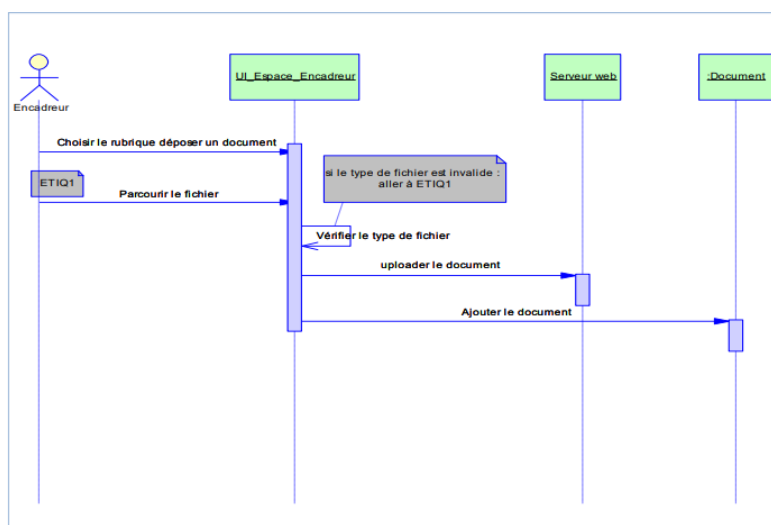


Figure 2.11: Diagramme de séquence : Affectation des tâches hebdomadaire

b. Diagramme de séquence : Validation d'un stage

Ce diagramme montre le processus de validation d'un stage par l'encadrant. Après que le stagiaire ait terminé ses tâches assignées et soumis un rapport final.

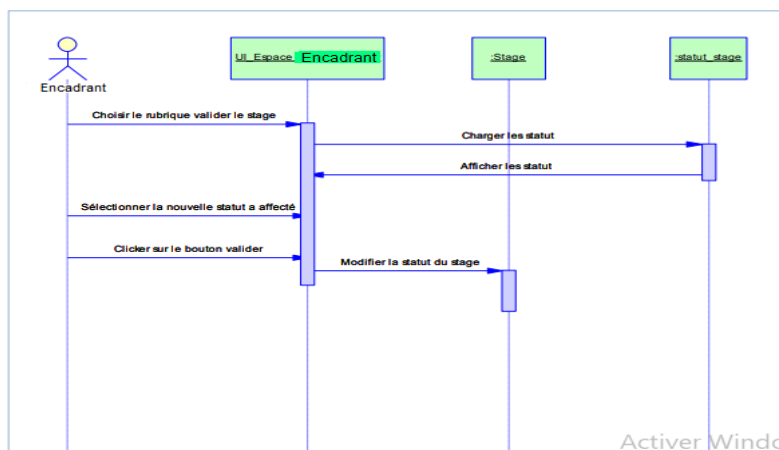


Figure 2.12: Diagramme de séquence : Validation d'un stage

2.2.4 Diagramme de classe

Le diagramme de classe est un diagramme structurel d'UML qui illustre la structure d'un système en montrant ses différentes classes, leurs attributs, leurs méthodes, ainsi que les relations entre les objets. Il permet de modéliser les aspects statiques du système et de visualiser les principales entités et leurs interactions. Les diagrammes de classe sont essentiels pour comprendre la conception d'un système orienté objet, car ils mettent en lumière la manière dont les différentes classes sont liées et interagissent entre elles.

- **Classe ProfileUser** : Cette classe représente le profil d'un utilisateur avec des informations telles que le rôle de l'utilisateur et l'image de profil. Elle contient un attribut pour le rôle de l'utilisateur et un autre pour l'image de profil, qui est optionnelle.
- **Classe Stagiaire** : Modélise un stagiaire participant à un stage. Cette classe comprend des attributs tels que le type de formation, le département, le niveau, et la filière. Elle est liée à un utilisateur spécifique et permet d'associer des informations académiques et de stage à chaque stagiaire.
- **Classe Stage** : Décrit un stage avec des attributs tels que l'entreprise d'accueil, le sujet du stage, les dates de début et de fin, ainsi qu'une description. Elle permet de gérer les stages, y compris les fichiers associés comme les rapports descriptifs et finaux, les présentations, et les informations sur les avancements et la note du stage. Elle est également liée aux utilisateurs et aux avancements de stage.
- **Classe Avancement** : Représente l'avancement d'un stage avec des attributs tels que le fichier d'avancement, la date, la semaine, et le statut. Cette classe permet de suivre le progrès du stagiaire et de gérer les documents liés aux avancements.
- **Classe AvancementStage** : Modélise les détails des avancements d'un stage, incluant une description, la date, la semaine, et un statut de manquant. Elle permet de documenter les différents aspects des avancements et de vérifier leur conformité avec les attentes du stage.

2.3 Maquette

a. Structure de l'application

Dans notre application, nous avons choisi d'adopter une structure évolutive, conçue pour répondre efficacement aux besoins variés et en constante évolution des utilisateurs. Ce type de structure permet de maintenir une hiérarchisation claire et cohérente des contenus, garantissant ainsi un accès rapide et direct à l'information recherchée par les utilisateurs. Grâce à cette organisation bien pensée, l'application permet une navigation fluide et intuitive, réduisant le temps nécessaire pour trouver des informations importantes tout en améliorant l'expérience utilisateur globale.

La structure évolutive a été privilégiée car elle offre une grande flexibilité, ce qui est essentiel dans un environnement numérique dynamique où les attentes et les exigences des utilisateurs peuvent changer rapidement. Elle permet de moduler et de réorganiser les éléments de contenu en fonction des besoins actuels, sans compromettre la clarté ou la simplicité d'utilisation. Cette adaptabilité est d'autant plus importante lorsqu'il s'agit de mettre à jour ou d'ajouter de nouvelles fonctionnalités, car elle minimise les interruptions pour l'utilisateur final et facilite une intégration fluide des modifications.

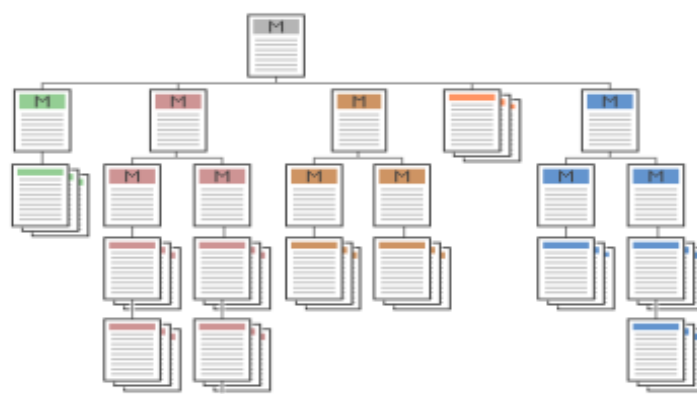


Figure 2.13: Maquette : Structure de l'application

b. Carte Graphique

La charte graphique vise à maintenir une uniformité visuelle à travers toutes les réalisations graphiques d'un projet. Pour garantir une telle cohérence, nous avons adopté une structure homogène pour toutes les interfaces de notre application. Cette approche assure que les éléments graphiques sont alignés avec l'identité visuelle du projet et offrent une expérience utilisateur harmonieuse.

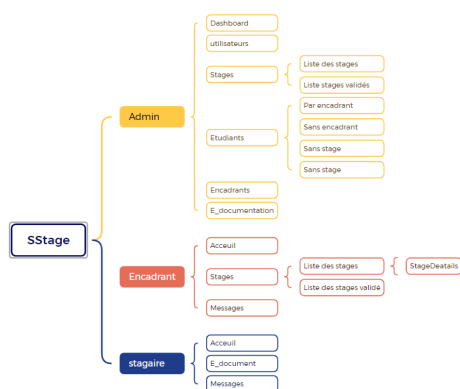


Figure 2.14: Carte Mentale : Exemple de cohérence visuelle

2.4 Conclusion

Ce chapitre se concentre principalement sur l'étude de la conception de l'architecture du système, qui est une étape essentielle pour garantir le bon déroulement de la réalisation du projet. La conception de l'architecture joue un rôle crucial en définissant la structure globale du système, en identifiant les principaux composants et en établissant les relations entre ces différents éléments. En offrant une vue d'ensemble claire et détaillée de la manière dont les différentes parties du système interagissent, cette phase permet de minimiser les ambiguïtés et les incohérences qui pourraient surgir lors du développement.

Dans le cadre de notre projet, l'étude de l'architecture du système couvre plusieurs aspects clés. Tout d'abord, elle s'attache à définir les composants fondamentaux du système, tels que les modules de gestion des utilisateurs, la gestion des stages, et le suivi des tâches hebdomadaires. Ensuite, elle détaille les interactions entre ces différents modules, afin de s'assurer que chaque composant fonctionne de manière harmonieuse et efficace au sein de l'ensemble.

L'approche architecturale choisie facilite également l'identification et la priorisation des tâches principales, comme la gestion des profils des utilisateurs, l'attribution et le suivi des tâches aux stagiaires, la gestion des rapports de stage et la validation des stages. Ces tâches constituent le cœur fonctionnel du système, et leur conception bien étudiée permet d'assurer une base solide pour le développement et l'intégration ultérieure des fonctionnalités.

En mettant l'accent sur la conception de l'architecture, ce chapitre illustre comment une planification soignée et une analyse approfondie de la structure du système peuvent simplifier la phase de réalisation. Une architecture bien conçue facilite non seulement le développement de chaque composant de manière modulaire, mais elle permet également de s'adapter aux évolutions futures et aux besoins émergents, assurant ainsi la pérennité et la flexibilité du système.

Chapter 3 Réalisation

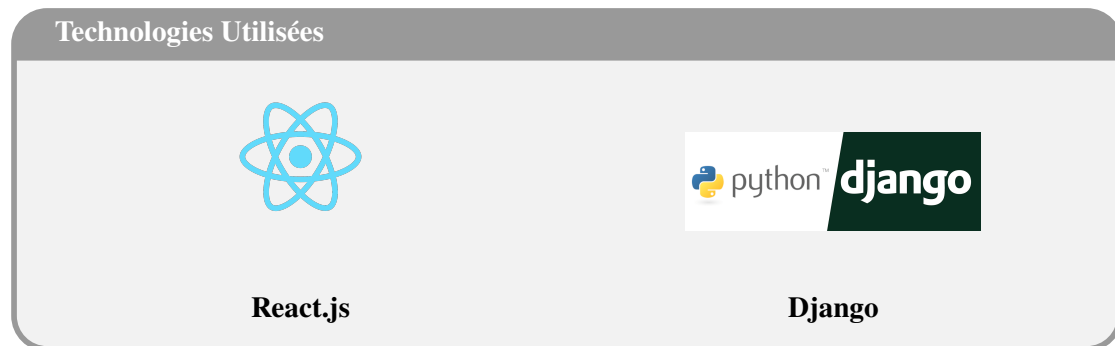
Chapter Contents

3.1	Introduction aux Technologies Utilisées	30
3.1.1	React.js	30
3.1.2	Django	30
3.1.3	Illustration de la Structure Technologique	31
3.2	Page d'Accueil	32
3.2.1	Page d'Accueil	32
3.2.2	Modal de Connexion	32
3.3	Section Admin	33
3.3.1	Accueil	33
3.3.2	Utilisateurs	34
3.3.3	Stages	35
3.3.4	Étudiants	35
3.3.5	Notes des Stages	37
3.3.6	Documents des Utilisateurs	37
3.4	section de Stagiaire	38
3.4.1	Accueil de Stagiaire	38
3.4.2	Modal d'Ajout des Informations Professionnelles	38
3.4.3	Modal d'Ajout d'un Stage	39
3.4.4	Page de Documents	39
3.4.5	Page de messagerie	40
3.5	section Encadrant	41
3.5.1	Accueil de l'Encadrant	41
3.5.2	Page des Listes des Stages pour les encadrants	42
3.5.3	Page de Messagerie	42
3.6	Conclusion	44

Abstract

La partie de réalisation est une étape cruciale réservée au développement concret du système étudié. Elle constitue le cœur du projet, où les concepts théoriques et les spécifications fonctionnelles sont transformés en une application opérationnelle. Cette phase implique l'utilisation de technologies adaptées et la mise en œuvre de fonctionnalités clés pour répondre aux besoins définis, garantissant ainsi la réussite et la robustesse de l'ensemble du système.

3.1 Introduction aux Technologies Utilisées



3.1.1 React.js

React.js est une bibliothèque JavaScript dédiée à la création d'interfaces utilisateur. Elle est particulièrement adaptée pour le développement de composants front-end modernes et dynamiques. Voici quelques-uns de ses avantages clés :

- **Gestion efficace de l'état** : React utilise un système de gestion d'état (comme `useState` et `useReducer`) permettant de manipuler l'état de l'application de manière performante et organisée.
- **Composants réutilisables** : La logique de développement orientée composants permet de créer des éléments modulaires et réutilisables à travers l'application, facilitant ainsi la maintenance et l'évolution du code.
- **Interfaces utilisateur interactives et réactives** : Grâce à son approche basée sur le DOM virtuel, React assure des mises à jour rapides et optimisées de l'interface utilisateur, offrant une expérience fluide à l'utilisateur.
- **Écosystème riche** : React dispose d'une vaste communauté et d'un écosystème de bibliothèques et d'outils (comme React Router, Redux) qui simplifient la gestion du routage, de l'état global, et d'autres fonctionnalités avancées.

3.1.2 Django

Django est un framework Python pour le développement web côté serveur. Il est conçu pour permettre la création rapide d'applications robustes et sécurisées. Ses principaux avantages incluent :

- **Cadre de travail rapide** : Django suit le principe "Don't Repeat Yourself" (DRY), permettant une production de code plus rapide et moins redondante.
- **Sécurité intégrée** : Ce framework inclut par défaut de nombreuses fonctionnalités de sécurité (comme la prévention des injections SQL et des attaques CSRF), ce qui facilite la construction d'applications sécurisées.

- **Gestion simplifiée des bases de données :** Django inclut un ORM (Object-Relational Mapping) qui permet de manipuler des bases de données de manière simple et intuitive sans écrire de SQL brut.
- **Facilité d'intégration avec d'autres services :** Django peut facilement s'intégrer avec des services externes et APIs, facilitant ainsi la communication avec des bases de données tierces, des systèmes d'authentification, et d'autres services.

3.1.3 Illustration de la Structure Technologique

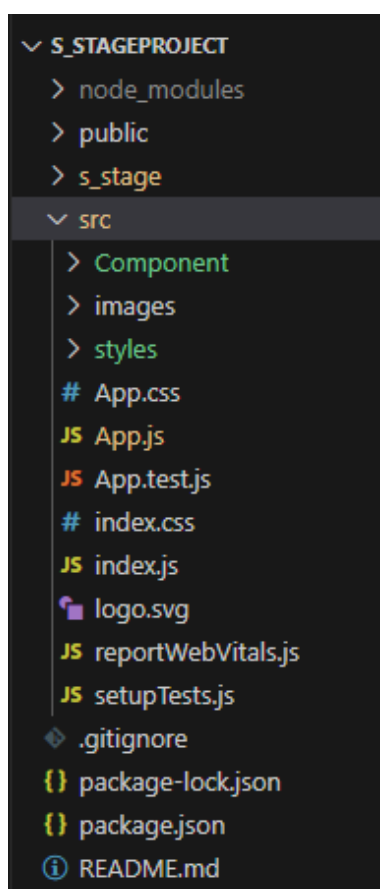


Figure 3.1: L'application Frontend

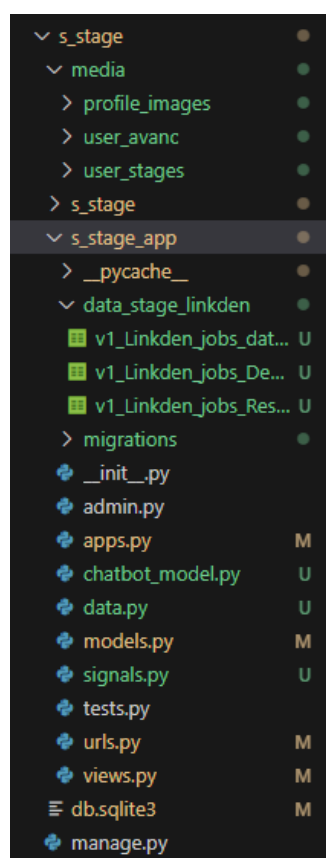


Figure 3.2: L'application Backend

Les deux applications composant notre système sont soigneusement structurées pour répondre à des besoins distincts mais complémentaires. La première application, dédiée à l'interface utilisateur, est construite avec React.js et se concentre sur l'expérience utilisateur. Elle est conçue pour offrir une interface intuitive et réactive, facilitant la navigation et l'interaction des utilisateurs avec le système. Cette application gère l'affichage des données, la soumission des formulaires et l'interaction directe avec les utilisateurs finaux.

La seconde application, développée en Django, représente la partie administration et back-end du système. Elle s'occupe de la gestion des données, de l'authentification des utilisateurs

et de l'exécution des opérations métier. Ce backend robuste est responsable de la validation des données, de la gestion des utilisateurs et des tâches, ainsi que de la communication avec la base de données.

Pour assurer une communication fluide entre ces deux applications, nous utilisons des API bien définies. Ces API permettent d'envoyer et de recevoir des données entre le frontend et le backend, assurant ainsi une synchronisation efficace et une intégrité des données à travers le système. Le frontend envoie des requêtes HTTP au backend pour récupérer ou soumettre des informations, tandis que le backend traite ces requêtes, met à jour la base de données et renvoie les réponses appropriées.

Cette architecture en deux parties garantit non seulement une séparation claire des responsabilités mais aussi une flexibilité et une évolutivité dans le développement et la maintenance du système.

3.2 Page d'Accueil

3.2.1 Page d'Accueil

La page d'accueil de notre application est conçue pour offrir une vue d'ensemble intuitive et engageante aux utilisateurs. Elle sert de point d'entrée principal vers les différentes fonctionnalités et sections de l'application. Voici une représentation visuelle de la page d'accueil :

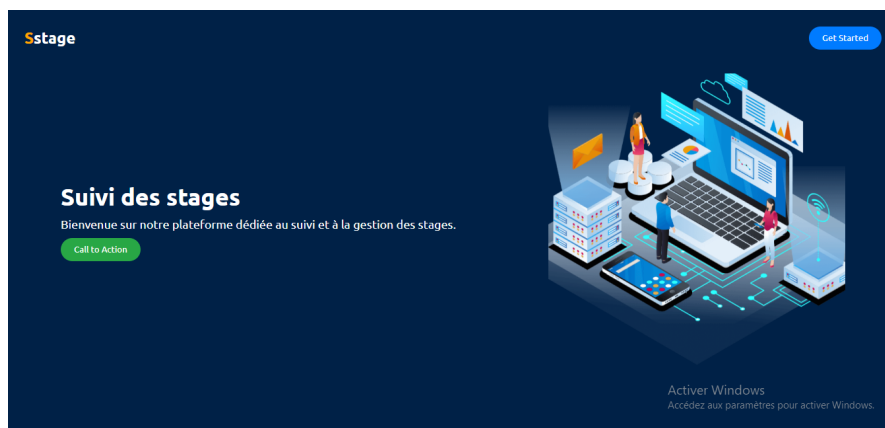


Figure 3.3: Page d'Accueil de l'application

3.2.2 Modal de Connexion

Pour accéder à leur espace privé, les utilisateurs doivent se connecter via un modal dédié à cet effet. Ce modal permet aux utilisateurs de saisir leurs informations d'identification et d'accéder ainsi à leurs fonctionnalités et données personnelles. Voici à quoi ressemble le modal de connexion :

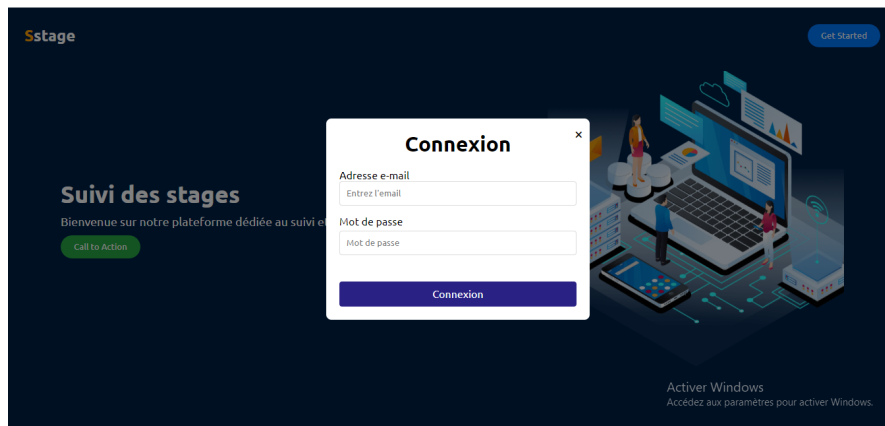


Figure 3.4: Modal de Connexion pour accéder à l'espace privé

La page d'accueil et le modal de connexion sont des éléments essentiels pour garantir une expérience utilisateur fluide et sécurisée. La page d'accueil offre une introduction aux principales fonctionnalités de l'application, tandis que le modal de connexion assure que seuls les utilisateurs authentifiés peuvent accéder à leur espace privé, renforçant ainsi la sécurité et la personnalisation de l'expérience utilisateur.

3.3 Section Admin

La section Admin est essentielle pour la gestion et la supervision de l'application. Elle permet aux administrateurs de suivre les statistiques, gérer les utilisateurs, et superviser les stages et les étudiants. Voici un aperçu des différentes sous-sections disponibles dans l'interface d'administration.

3.3.1 Accueil

Cette sous-section présente une vue d'ensemble des statistiques de l'application et de la progression des données au fil du temps. Elle permet aux administrateurs de surveiller l'activité globale de l'application et d'évaluer les tendances.

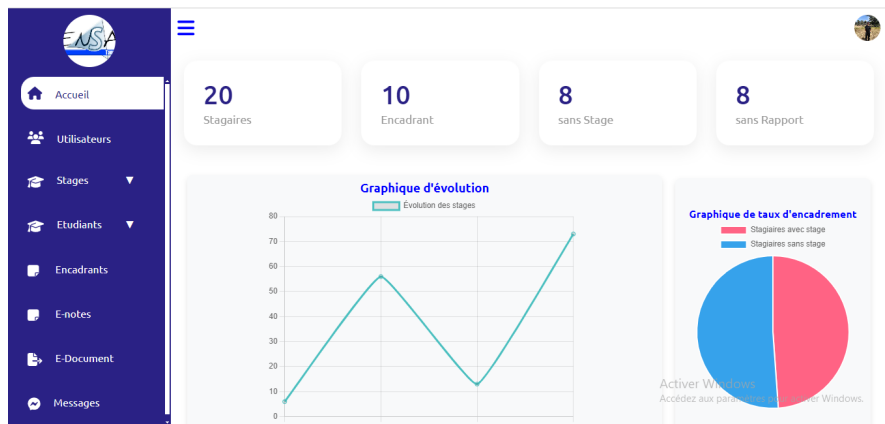


Figure 3.5: Statistiques de l'application et progression des données

3.3.2 Utilisateurs

La gestion des utilisateurs est cruciale pour le bon fonctionnement de l'application. Cette sous-section permet aux administrateurs de gérer les comptes des utilisateurs, d'attribuer des rôles, et de superviser leurs activités.

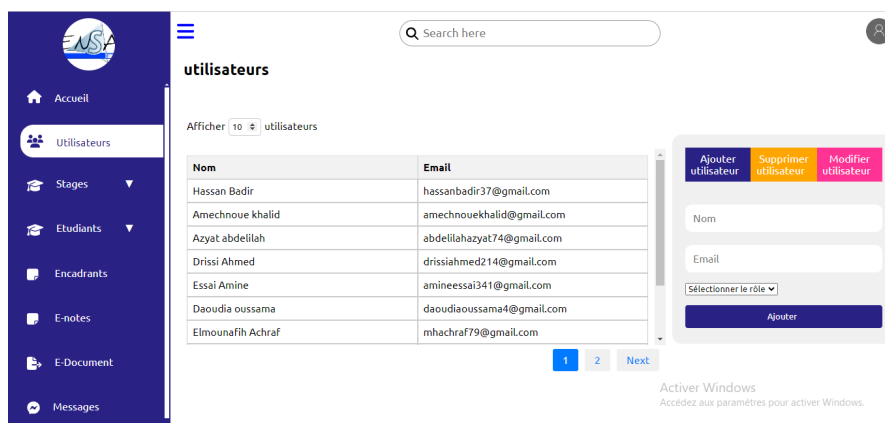


Figure 3.6: Gestion des utilisateurs dans l'application

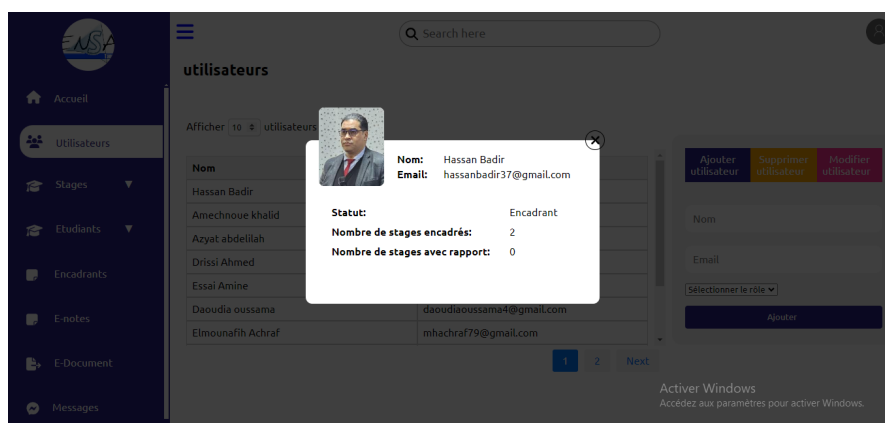


Figure 3.7: l'état des utilisateur en fil de temps

3.3.3 Stages

Cette sous-section permet aux administrateurs de visualiser et de gérer les stages en cours. Elle affiche les stages validés et non validés pour chaque filière et niveau choisis, offrant ainsi un aperçu détaillé des différentes phases des stages.



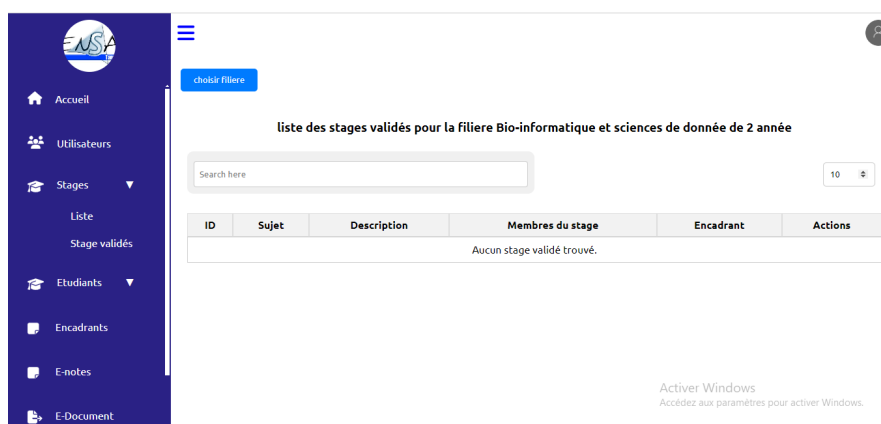
liste des stages pour la filière Bio-informatique et sciences de donnée de 2 année

Rechercher...

ID	Sujet	Description	Groupe de stage	Encadrant	Action
20	plateforme suivi...	suivi des...	Essai Amine	Azyat abdelilah	⊗
21	Traitement intelligent...	Le traitement...	Daoudia oussama	Hassan Badir	⊗

1

Figure 3.8: Gestion des stages selon la filière et le niveau



liste des stages validés pour la filière Bio-informatique et sciences de donnée de 2 année

Search here

ID	Sujet	Description	Membres du stage	Encadrant	Actions
Aucun stage validé trouvé.					

Figure 3.9: Gestion des stages : validés selon la filière et le niveau

3.3.4 Étudiants

Cette sous-section permet de visualiser les informations concernant les étudiants sous différentes catégories : par encadrant, sans encadrant, sans stage, et sans rapport. Chaque étudiant peut également télécharger des fichiers PDF ou CSV pour compléter les informations nécessaires.

choix Filtrer

Liste des étudiants par Encadrant pour la Bio-informatique et sciences de donnée de 2 année

Excel PDF Column Visibility Rechercher...

ID	membres	Sujet	Encadrant
20	Essai Amine	plateforme suivi...	Azyat abdellah
21	Daoudia oussama	Traitement intelligent...	Hassan Badir

Activer Windows
Accédez aux paramètres pour activer Windows.

Figure 3.10: Étudiants avec encadrant

choix Filtrer

Liste des étudiants sans Encadrant pour la filiere Bio-Informatique et sciences de donnée de 2 année

Excel PDF Column Visibility Rechercher...

ID	membres	Sujet
Aucun élément à afficher		

Activer Windows
Accédez aux paramètres pour activer Windows.

Figure 3.11: Étudiants sans encadrant

choix Filtrer

Liste des étudiants Sans stage pour la filiere GINF de 2 année

Excel PDF Column Visibility Rechercher...

Id	Nom complet
42	outaleb nouridine

Activer Windows
Accédez aux paramètres pour activer Windows.

Figure 3.12: Étudiants sans stage

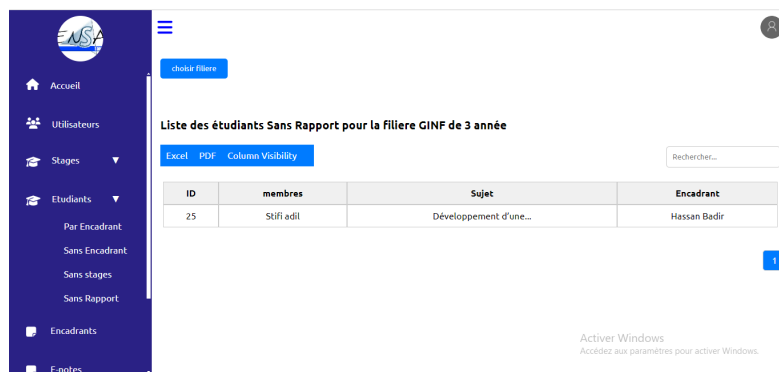


Figure 3.13: Étudiants sans rapport

3.3.5 Notes des Stages

Cette sous-section présente les notes attribuées aux stages, avec des informations détaillées sur les évaluations et les performances des étudiants. Elle aide les administrateurs à suivre les résultats académiques et à assurer le suivi des progrès des stagiaires.

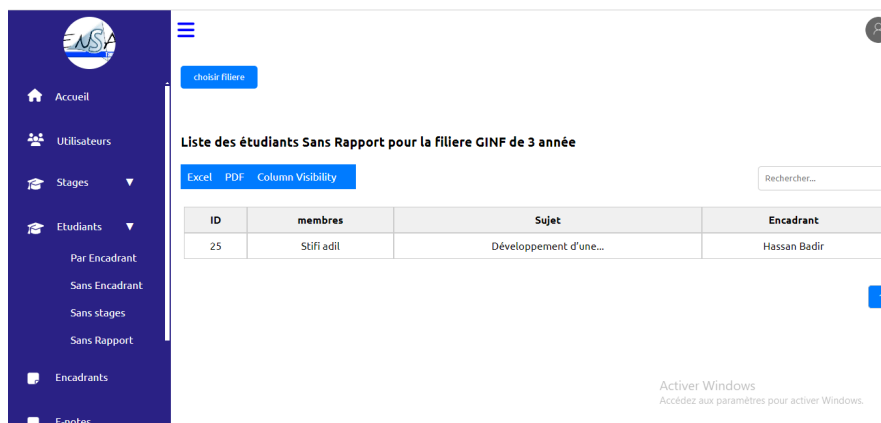


Figure 3.14: Notes des stages et évaluations

3.3.6 Documents des Utilisateurs

Cette sous-section permet aux administrateurs de gérer les documents soumis par les utilisateurs. Elle inclut la possibilité de consulter, organiser, et vérifier les documents associés aux comptes des utilisateurs.

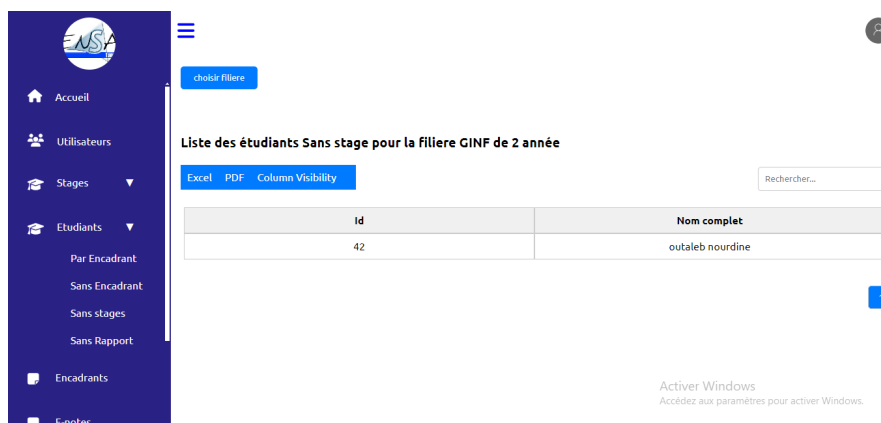


Figure 3.15: Gestion des documents des utilisateurs

3.4 section de Stagiaire

La section "Page de Stagiaire" est dédiée aux stagiaires de l'application, leur permettant de gérer leurs informations personnelles, de suivre l'état de leurs stages, et d'organiser les documents nécessaires à leur progression. Voici les principales pages et fonctionnalités disponibles pour les stagiaires.

3.4.1 Accueil de Stagiaire

Cette page d'accueil fournit aux stagiaires une vue d'ensemble de leur profil, de leurs stages en cours, et des tâches à accomplir. Elle permet également de visualiser les notifications importantes et les nouvelles mises à jour liées à leurs stages.

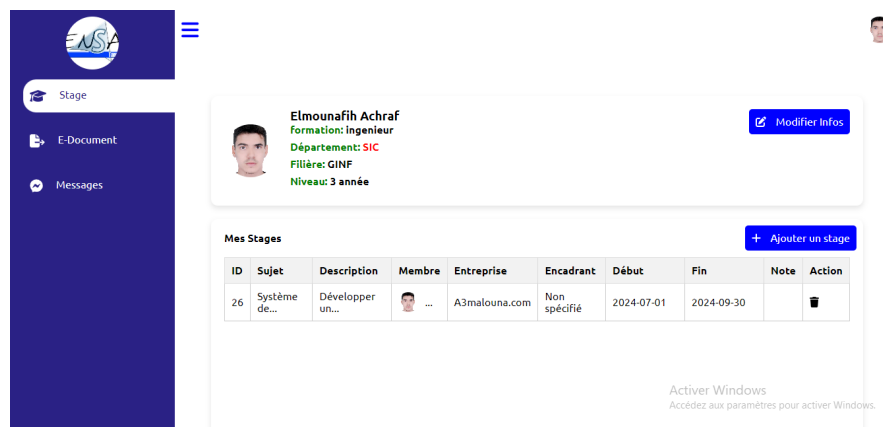


Figure 3.16: Page d'accueil du stagiaire avec vue d'ensemble des stages et notifications

3.4.2 Modal d'Ajout des Informations Professionnelles

Ce modal permet aux stagiaires de compléter ou mettre à jour leurs informations, y compris les détails sur leur formation, filière, département, niveau, ainsi que leurs compétences et expériences professionnelles. Le formulaire est conçu pour être intuitif, facilitant l'entrée rapide et précise de toutes ces données essentielles.

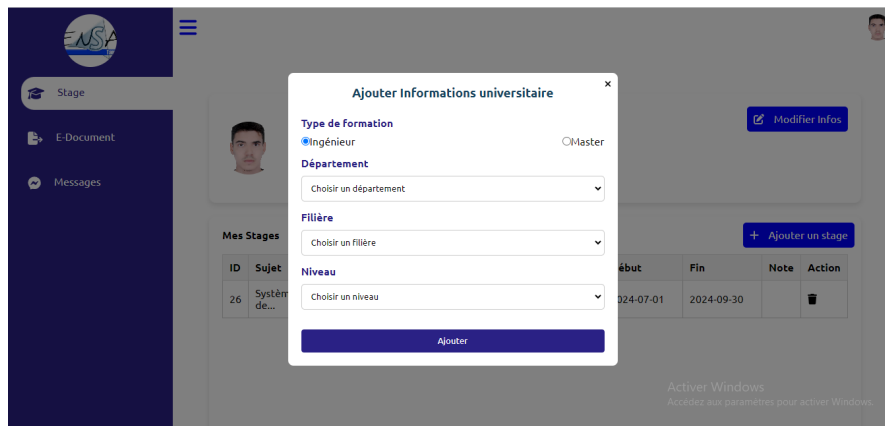


Figure 3.17: Modal d'ajout ou de mise à jour des informations professionnelles du stagiaire

3.4.3 Modal d'Ajout d'un Stage

Ce modal est conçu pour permettre aux stagiaires de soumettre des demandes de stage. Ils peuvent remplir les détails pertinents, tels que le titre du stage, la description, l'entreprise d'accueil, et la période de stage. Cette fonctionnalité simplifie le processus de candidature et assure que toutes les informations nécessaires sont recueillies de manière structurée.

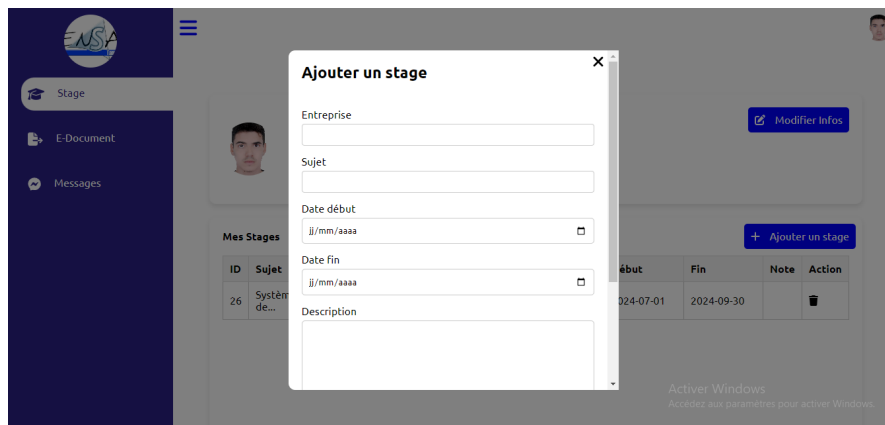


Figure 3.18: Modal d'ajout d'un nouveau stage pour le stagiaire

3.4.4 Page de Documents

Cette page permet aux stagiaires d'ajouter et de gérer leurs fichiers en deux catégories distinctes : la première catégorie concerne les fichiers principaux, tels que le rapport descriptif, le rapport final, et la présentation PPT. La seconde catégorie regroupe les fichiers d'avancement, documentant les différentes étapes du parcours de stage au cours de leur période.

- **Ajout des fichiers principaux** : Permet aux stagiaires de télécharger des documents essentiels, notamment un rapport initial en début de stage pour offrir à l'encadrant une vision globale du sujet de stage, ainsi qu'un rapport final et une présentation PPT à la fin du stage.

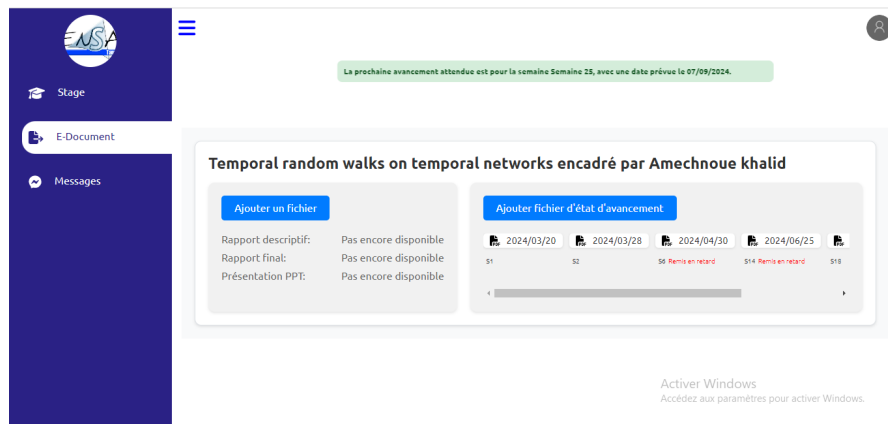


Figure 3.19: gestion des documents : ajout des fichiers principaux

- **Ajout des états d'avancement** : Permet aux stagiaires de suivre et de soumettre les rapports d'étape réguliers sur l'avancement de leur stage.

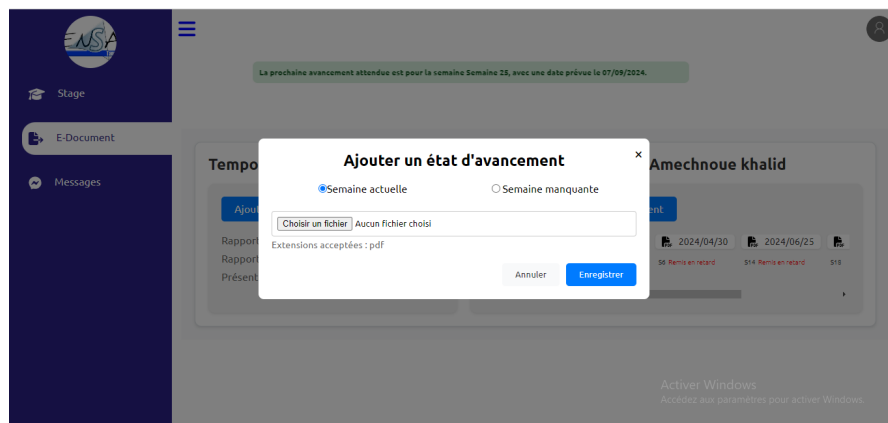


Figure 3.20: gestion des documents : ajout des fichiers d'avancement

3.4.5 Page de messagerie

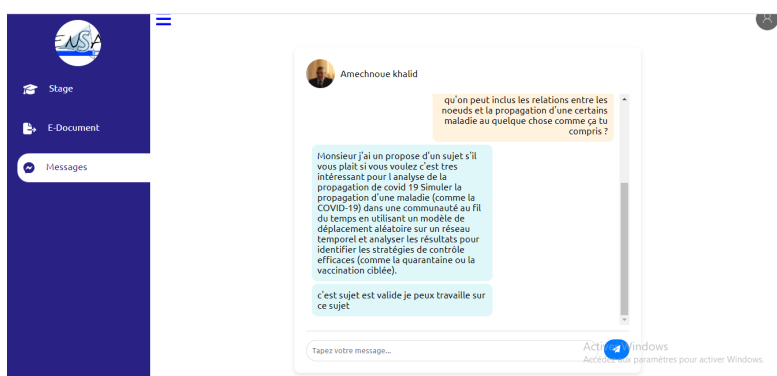


Figure 3.21: Illustration de la communication entre stagiaires et professeurs

Notre plateforme facilite la communication entre les stagiaires et leurs encadrants grâce à des outils dédiés. Les stagiaires peuvent poser des questions, partager des mises à jour sur leurs projets, et recevoir des retours instantanés de leurs professeurs. Les encadrants peuvent suivre les progrès des stagiaires, fournir des conseils personnalisés et gérer les sessions de feedback de manière centralisée. Avec notre système de messagerie intégré et des notifications en temps réel, la collaboration est fluide et productive, assurant ainsi une meilleure expérience de stage pour tous.

3.5 section Encadrant

La section "Page des Encadrants" est dédiée aux encadrants de l'application, leur offrant les outils nécessaires pour superviser et gérer les stagiaires sous leur responsabilité. Voici les principales pages et fonctionnalités disponibles pour les encadrants.

3.5.1 Accueil de l'Encadrant

La page d'accueil des encadrants est conçue pour offrir une vue d'ensemble complète et pratique des informations professionnelles pertinentes. En tant que coordonnateur des filières d'ingénieur et des programmes de master, vous aurez accès à des informations détaillées sur les départements et les filières sous votre responsabilité. Cette page vous permet de visualiser et de gérer les informations relatives aux différents départements académiques, les spécialités des filières d'ingénieur, ainsi que les programmes de master proposés. Vous pourrez suivre les évolutions des différents parcours, consulter les demandes de stage et de projet en lien avec chaque filière, et assurer une coordination efficace entre les stagiaires et leurs encadrants. Grâce à cette interface centralisée, la gestion des informations professionnelles et des activités académiques devient plus fluide et organisée.

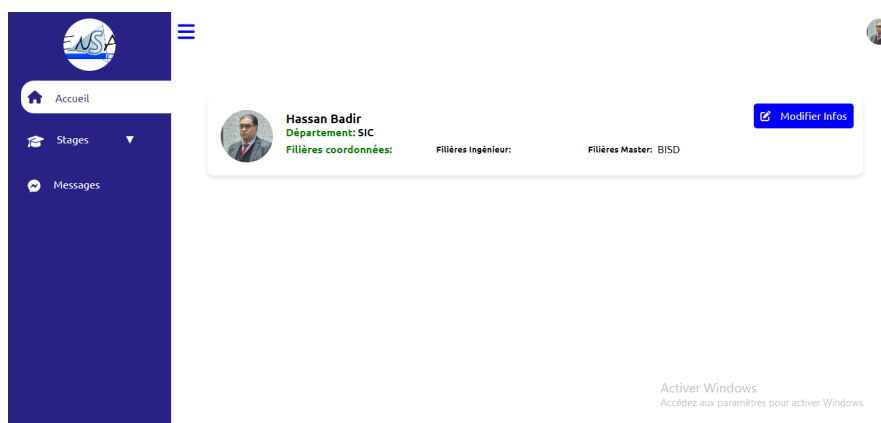


Figure 3.22: Page d'accueil de l'encadrant avec vue d'ensemble des stagiaires et notifications

3.5.2 Page des Listes des Stages pour les encadrants

La page des listes des stages permet aux encadrants de visualiser et de gérer tous les stages attribués sous leur supervision. Cette interface centralisée offre une vue complète des stages en cours, incluant des informations détaillées sur chaque stage, telles que le titre, la description, l'entreprise d'accueil, la période de stage, et le statut d'avancement. Les encadrants peuvent également accéder aux documents soumis par les stagiaires, suivre les progrès et fournir des retours ou des recommandations. Cette fonctionnalité facilite la gestion efficace des stages, en permettant aux encadrants de suivre et d'évaluer les performances des stagiaires, tout en assurant une communication fluide et continue avec eux.

ID	Sujet	Description	Groupe de stage	Entreprise	Début	Fin	Action
26	Système de...	Développer un...	...	A3malouna.com	2024-07-01	2024-09-30	Encadrer
27	SYSTÈME DE...	Développer une...	...	Astek Morocco	2024-07-08	2024-09-30	Encadrer

ID	Sujet	Description	Groupe de stage	Entreprise	Début	Fin	Action
25	Développement d'une...	création d'une...	...	Expleo Group	2024-07-15	2024-11-04	Voir Valider

Figure 3.23: Page des listes des stages pour les encadrants, avec détails des stages et options de gestion

3.5.3 Page de Messagerie

La page de messagerie permet aux encadrants de communiquer efficacement avec les étudiants qu'ils supervisent pour l'année en cours. Grâce à cette fonctionnalité, les encadrants peuvent initier des discussions en temps réel, poser des questions, fournir des retours, et offrir des conseils personnalisés directement aux stagiaires. Cette interface de messagerie est conçue pour faciliter la communication fluide entre les encadrants et les stagiaires, avec des options pour envoyer des messages, recevoir des notifications instantanées, et consulter l'historique des conversations. En utilisant ce système intégré, les encadrants peuvent suivre les échanges, résoudre rapidement les problèmes, et garantir que les stagiaires reçoivent le soutien nécessaire tout au long de leur stage.

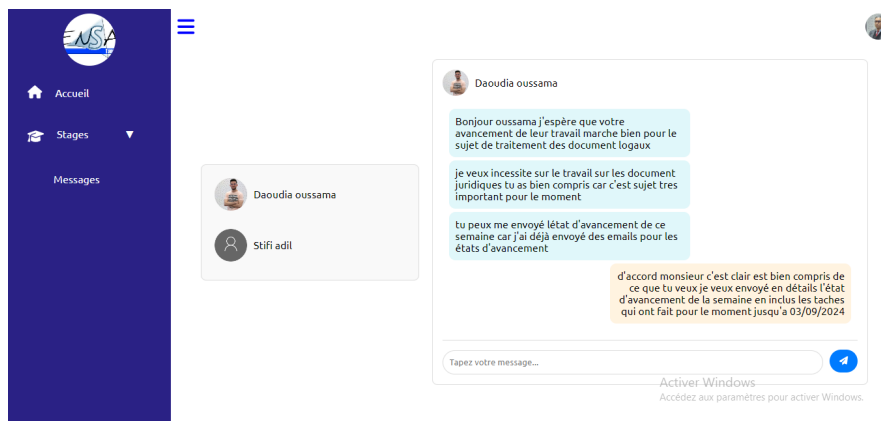


Figure 3.24: Interface de messagerie pour les encadrants, permettant des discussions en temps réel avec les étudiants

3.6 Conclusion

Ce chapitre se concentre sur la réalisation de l'application "Sstage", en détaillant les technologies utilisées et les différentes sections de l'application. La partie administrative permet de gérer les utilisateurs et de suivre l'évolution des stages au fil du temps. Les sections dédiées aux encadrants facilitent le suivi et l'encadrement des stages sous leur responsabilité. Enfin, les pages destinées aux stagiaires offrent la possibilité d'ajouter et de gérer leurs stages, ainsi que de soumettre les documents requis, comme les rapports et autres fichiers relatifs au stage.

Chapter 4 Intégration de bot pour la réponse automatique des stages

Chapter Contents

4.1	Collecte des Données	46
4.1.1	Collecte des Données à partir d'une Base de Données	46
4.1.2	Collecte des Données par Scraping avec Selenium	46
4.2	Création du Fichier data.py	47
4.3	Importation des Bibliothèques	47
4.4	Initialisation du Lemmatiser	48
4.5	Définition de la Fonction de Prétraitement des Données	48
4.6	Tokenisation et Construction des Listes	49
4.7	Lemmatisation et Nettoyage des Mots	49
4.8	Création des Vecteurs de Caractéristiques (Bag of Words)	50
4.9	Application de la Fonction et Création des Données d'Entraînement	51
4.10	Entraînement de Modèle	51
4.11	Prédiction des Tags	53
4.12	Génération des Réponses	54
4.13	Résumé des Performances	55
4.14	Conclusion sur l'intégration du bot dans la plateforme de stage	57

Abstract

L'intégration d'un chatbot intelligent dans une application permet d'améliorer l'interaction utilisateur en offrant des réponses automatisées et pertinentes aux questions posées. Ce chapitre explore en détail le processus de développement d'un chatbot utilisant le traitement du langage naturel (NLP) et le machine learning pour prédire des intentions basées sur les entrées des utilisateurs. Le chatbot est conçu pour comprendre et répondre à des questions spécifiques concernant des stages, des offres d'emploi et des informations institutionnelles.

4.1 Collecte des Données

La collecte des données est une étape cruciale dans le développement d'un chatbot. Voici deux méthodes couramment utilisées pour recueillir des données :

4.1.1 Collecte des Données à partir d'une Base de Données

Pour collecter des données à partir d'une base de données, vous pouvez utiliser des requêtes SQL pour extraire des informations pertinentes. Cela peut inclure des dialogues historiques, des questions fréquemment posées, ou d'autres interactions pertinentes. Voici une procédure générale :

- **Connexion à la Base de Données** : Établissez une connexion avec la base de données de django.
- **Exécution des Requêtes** : Utilisez des requêtes SQL pour extraire les données nécessaires.
- **Prétraitement des Données** : Nettoyez et transformez les données extraites pour les rendre adaptées à l'entraînement du modèle.

4.1.2 Collecte des Données par Scraping avec Selenium

Le scraping est une méthode pour extraire des données depuis des sites web. Pour scraper des données de LinkedIn ou d'autres sites, vous pouvez utiliser la bibliothèque Selenium. Voici un exemple de procédure pour le scraping :

- **Configuration de Selenium** : Installez Selenium et un navigateur web comme ChromeDriver.
- **Développement du Script de Scraping** : Écrivez un script pour naviguer sur le site, localiser les éléments à extraire (par exemple, profils LinkedIn), et recueillir les données.
- **Nettoyage des Données** : Traitez et nettoyez les données extraites pour les rendre utilisables pour l'entraînement du modèle.

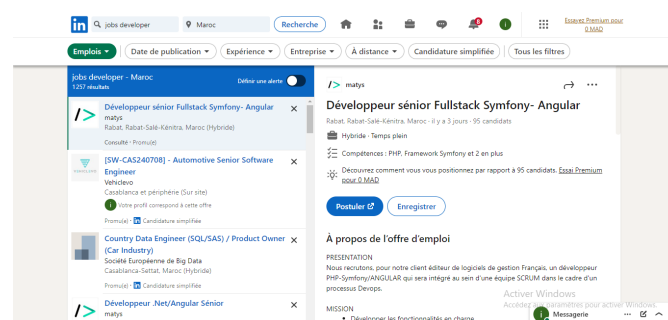


Figure 4.1: Scraping des données depuis LinkedIn avec Selenium

4.2 Création du Fichier data.py

Le fichier data.py contiendra les intents sous forme de liste de dictionnaires. Chaque dictionnaire correspond à une intention spécifique du chatbot, avec les attributs suivants :

- tag : Une étiquette qui représente l'intention ou la catégorie de la phrase (par exemple, "greeting" ou "goodbye").
- patterns : Une liste de phrases ou d'expressions que les utilisateurs pourraient utiliser pour exprimer l'intention associée au tag (par exemple, "Hello", "Salut").
- responses : Une liste de réponses que le chatbot donnera en fonction de l'intention détectée (par exemple, "Salut à toi!", "Comment vas-tu?").

```
data = {
    "intents": [
        {
            "tag": "greeting",
            "patterns": ["Hello", "Salut", "Bonjour"],
            "responses": ["Salut à toi!", "Hello", "Comment vas-tu?", "Salutations!", "Enchanté"]
        },
        {
            "tag": "goodbye",
            "patterns": ["bye", "Salut", "see ya", "adios", "cya"],
            "responses": ["C'était sympa de te parler", "à plus tard", "On se reparle très vite!"]
        },
        {
            "tag": "stagiaire_info",
            "patterns": [
                "Est-ce que {nom} a un stage?",
                "Quelle est l'adresse email de {nom}?",
                "Donne-moi des infos sur {nom}?",
                "Donne-moi des détails sur {nom}?",
                "le prenom de {nom} est {prenom}",
                "l'adresse email de {nom} est {email}"
            ],
            "responses": []
        }
    ]
}
```

Figure 4.2: Structure du fichier data.py illustrant les intents, les tags, et les réponses.

4.3 Importation des Bibliothèques

Pour préparer les données et entraîner le modèle de chatbot, nous utilisons plusieurs bibliothèques en Python. Voici une vue d'ensemble des bibliothèques utilisées :

```
import nltk
from nltk.stem import WordNetLemmatizer
import numpy as np
```

Figure 4.3: Bibliothèques utilisées pour le traitement et l'entraînement des données

- nltk (Natural Language Toolkit) : utilisé pour le traitement du langage naturel.
- WordNetLemmatizer : un outil de lemmatisation qui réduit les mots à leur forme de base.
- numpy : une bibliothèque pour les opérations numériques en Python.

4.4 Initialisation du Lemmatiser

```
# Initialisation de lemmatizer pour obtenir la racine des mots
lemmatizer = WordNetLemmatizer()
```

Figure 4.4: Initialisation du lemmatiseur avec WordNetLemmatizer

On crée une instance de `WordNetLemmatizer`, qui sera utilisée pour réduire les mots à leur forme canonique. La lemmatisation est un processus qui transforme les mots en leurs formes de base ou racine. Par exemple, le mot "running" sera réduit à "run" grâce à cette instance. Cela permet au modèle de traiter les différentes variations d'un mot comme équivalentes, ce qui est essentiel pour le traitement du langage naturel.

4.5 Définition de la Fonction de Prétraitement des Données

```
def preprocess_data(patterns, tags):
    words = []
    classes = []
    doc_X = []
    doc_y = []
```

Figure 4.5: Définition de la fonction de prétraitement des données

La fonction `preprocess_data` prend deux paramètres : `patterns` (les phrases d'entrée) et `tags` (les étiquettes correspondant à chaque phrase). Cette fonction est responsable de la préparation des données pour l'entraînement du modèle.

On initialise les listes `words`, `classes`, `doc_X`, et `doc_y` pour stocker les données traitées. `words` et `classes` sont utilisés pour conserver les mots uniques et les étiquettes uniques respectivement. `doc_X` et `doc_y` sont utilisés pour stocker les phrases d'entrée et les étiquettes associées à ces phrases.

Cette étape de prétraitement est cruciale pour transformer les données brutes en un format que le modèle peut utiliser pour l'entraînement.

4.6 Tokenisation et Construction des Listes

```
for intent in data["intents"]:
    for pattern in intent["patterns"]:
        tokens = nltk.word_tokenize(pattern)
        words.extend(tokens)
        doc_X.append(pattern)
        doc_y.append(intent["tag"])

# Ajouter le tag aux classes s'il n'est pas déjà là
if intent["tag"] not in classes:
    classes.append(intent["tag"])
```

Figure 4.6: Tokenisation et construction des listes

Pour chaque **pattern** (phrase), on effectue la tokenisation de la phrase en mots (tokens). La tokenisation est le processus de découpage d'une phrase en unités de sens plus petites (mots) que le modèle peut traiter.

Les étapes de la tokenisation et de la construction des listes sont les suivantes :

- On **tokenise** chaque phrase en mots en utilisant la fonction `nltk.word_tokenize`.
- On **ajoute** ces mots à la liste `words`, qui contient tous les mots uniques trouvés dans les phrases d'entrée.
- On **ajoute** la phrase originale à la liste `doc_X`, qui contient toutes les phrases d'entrée.
- On **ajoute** le tag associé à chaque phrase à la liste `doc_y`, qui contient les étiquettes correspondantes aux phrases.

Cette étape prépare les données pour la transformation en vecteurs de caractéristiques et permet au modèle de comprendre les relations entre les phrases et leurs tags associés.

4.7 Lemmatisation et Nettoyage des Mots

```
words = [lemmatizer.lemmatize(word.lower()) for word in words if word not in string.punctuation]

# Trier le vocabulaire et les classes par ordre alphabétique et prendre le
# set pour s'assurer qu'il n'y a pas de doublons
words = sorted(set(words))
classes = sorted(set(classes))
```

Figure 4.7: Lemmatisation et nettoyage des mots

La lemmatisation et le nettoyage des mots sont des étapes cruciales pour préparer les données textuelles à l'entraînement du modèle. Voici les principales opérations effectuées :

- **Lemmatisation** : Chaque mot est réduit à sa forme canonique à l'aide de `WordNetLemmatizer`. Par exemple, le mot "running" est transformé en "run". Cette étape aide à réduire la variabilité des mots en les ramenant à une forme de base.
- **Conversion en minuscule** : Tous les mots sont convertis en minuscules pour assurer que les variantes majuscules et minuscules du même mot soient traitées comme identiques.
- **Filtrage des signes de ponctuation** : Les signes de ponctuation sont supprimés des listes de mots afin d'éviter qu'ils n'affectent le modèle.
- **Suppression des doublons** : Les doublons sont éliminés en convertissant la liste des mots en un ensemble (set), ce qui permet de ne conserver que des mots uniques.
- **Tri des listes** : Les listes `words` et `classes` sont triées pour faciliter le traitement ultérieur. La liste `words` contient tous les mots uniques, tandis que la liste `classes` contient toutes les étiquettes uniques.

Ces étapes assurent que les données sont normalisées et prêtes pour la création des vecteurs de caractéristiques utilisés dans l'entraînement du modèle.

4.8 Création des Vecteurs de Caractéristiques (Bag of Words)

```
# Création du modèle d'ensemble de mots
for idx, doc in enumerate(doc_X):
    bow = []
    text = lemmatizer.lemmatize(doc.lower())
    for word in words:
        bow.append(1) if word in text else bow.append(0)

    # Marque l'index de la classe à laquelle le pattern actuel est associé à
    output_row = list(out_empty)
    output_row[classes.index(doc_y[idx])] = 1
```

Figure 4.8: Création des vecteurs de caractéristiques (Bag of Words)

La création des vecteurs de caractéristiques, souvent appelée *Bag of Words* (BoW), est une méthode essentielle pour transformer les phrases textuelles en vecteurs numériques que le modèle peut comprendre. Voici les principales étapes :

- **Initialisation de training** : On crée une liste `training` pour stocker les vecteurs de caractéristiques pour chaque document (phrase).
- **Création de `out_empty`** : On initialise `out_empty`, une liste de zéros dont la longueur est égale au nombre total de classes (tags). Cette liste est utilisée pour créer les vecteurs de sortie.
- **Création du vecteur `bow`** : Pour chaque document (`doc`), on crée un vecteur `bow` où chaque position indique la présence ou l'absence d'un mot spécifique du vocabulaire. Si le mot est présent dans le document, la position correspondante est mise à 1, sinon elle reste à 0.
- **Création de `output_row`** : On crée `output_row` pour représenter la sortie attendue avec un 1 à l'index correspondant au tag (étiquette) du document et des zéros ailleurs. Cela indique la

catégorie à laquelle appartient le document.

- **Ajout des vecteurs à training** : On ajoute les vecteurs bow et output_row à la liste training.

Ces vecteurs de caractéristiques sont ensuite utilisés pour entraîner le modèle de classification, permettant au chatbot de prédire les tags appropriés en fonction des phrases d'entrée.

4.9 Application de la Fonction et Création des Données d'Entraînement

```
# Séparer les features et les labels target
train_X = np.array(list(training[:, 0]))
train_y = np.array(list(training[:, 1]))
```

Figure 4.9: prétraitement et création des données d'entraînement

Pour créer les données d'entraînement pour le modèle de chatbot, nous appliquons la fonction de prétraitement aux phrases d'entrée et aux étiquettes :

- On appelle `preprocess_data` avec les patterns (phrases d'entrée) et les tags (étiquettes) pour obtenir les données d'entraînement, qui incluent les vecteurs de caractéristiques et les étiquettes associées.
- On sépare les vecteurs de caractéristiques (`train_X`) et les étiquettes de sortie (`train_y`) pour les utiliser lors de l'entraînement du modèle.

Cette approche prépare les données pour entraîner un modèle de chatbot en transformant des phrases d'entrée en vecteurs numériques que le modèle peut comprendre.

4.10 Entraînement de Modèle

```
model = Sequential()
model.add(Dense(128, input_shape=input_shape, activation="relu"))
model.add(Dropout(0.5))
model.add(Dense(64, activation="relu"))
model.add(Dropout(0.3))
model.add(Dense(output_shape, activation="softmax"))

adam = tf.keras.optimizers.Adam(learning_rate=0.01)

model.compile(loss='categorical_crossentropy', optimizer=adam, metrics=["accuracy"])

model.fit(x=train_X, y=train_y, epochs=200, verbose=1)
```

Figure 4.10: le processus d'entraînement de model

Configuration du Modèle :

- **Construction du Modèle :**

Le modèle est construit en utilisant l'API `Sequential` de Keras, qui permet d'empiler les couches de manière séquentielle.

- **Première Couche :**

La première couche dense (`Dense(128)`) contient 128 neurones et utilise la fonction d'activation `relu`. Elle prend en entrée le vecteur de caractéristiques (features) dont la forme est définie par `input_shape = (len(train_X[0]),)`, où `train_X` est la matrice d'entraînement contenant les vecteurs de caractéristiques.

- **Dropout :**

`Dropout(0.5)` est appliqué après cette couche pour désactiver aléatoirement 50% des neurones pendant l'entraînement, ce qui aide à réduire le surapprentissage (overfitting).

- **Deuxième Couche Dense :**

Une deuxième couche dense (`Dense(64)`) avec 64 neurones et également une fonction d'activation `relu` est ajoutée, suivie d'un autre `Dropout` de 30%.

- **Dernière Couche :**

La dernière couche dense (`Dense(output_shape)`) utilise une fonction d'activation `softmax`, ce qui est approprié pour la classification multicatégorie. `output_shape` correspond au nombre de classes uniques (tags).

Compilation du Modèle :

- **Fonction de Perte :**

Le modèle est compilé avec la fonction de perte `categorical_crossentropy`, qui est adaptée pour les problèmes de classification multi-classes où les étiquettes sont one-hot encodées.

- **Optimiseur :**

L'optimiseur Adam est utilisé avec un taux d'apprentissage de 0,01. Adam est un optimiseur stochastique qui combine les avantages de deux autres optimisateurs populaires, AdaGrad et RMSProp, et est souvent utilisé pour entraîner des modèles de deep learning.

- **Métrique d'Évaluation :**

`metrics=["accuracy"]` spécifie que l'exactitude de la prédiction est mesurée pendant l'entraînement.

Entraînement :

- **Exécution de l'Entraînement :**

Le modèle est entraîné avec `model.fit()` en utilisant les données d'entraînement `train_X`

(vecteurs de caractéristiques) et `train_y` (étiquettes de sortie).

- **Nombre d'Époques :**

`epochs=200` indique que le modèle passera 200 fois sur l'ensemble des données d'entraînement.

- **Affichage des Détails :**

`verbose=1` indique que les détails de chaque époque seront affichés pendant l'entraînement.

4.11 Prédiction des Tags

```
def pred_class(text, vocab, labels):
    bow = bag_of_words(text, vocab)
    result = model.predict(np.array([bow]))[0]
    thresh = 0.2
    y_pred = [[idx, res] for idx, res in enumerate(result) if res > thresh]

    y_pred.sort(key=lambda x: x[1], reverse=True)
    return_list = []
    for r in y_pred:
        return_list.append(labels[r[0]])
    return return_list
```

Figure 4.11: la fonction pour prédiction de tag

Explication de la Fonction de Prédiction:

La fonction `pred_class` est utilisée pour prédire le tag ou les tags les plus probables pour un texte d'entrée donné. Voici une explication détaillée du fonctionnement de cette fonction:

- **Bag of Words (BoW):** La fonction commence par transformer le texte d'entrée en un vecteur numérique appelé "Bag of Words" (BoW). Ce vecteur représente la présence ou l'absence de mots spécifiques dans le texte d'entrée.
- **Prédiction du Modèle:** Le vecteur BoW est ensuite passé au modèle d'apprentissage automatique entraîné, qui renvoie une liste de probabilités pour chaque tag possible. Chaque valeur représente la probabilité que le texte d'entrée corresponde à un tag spécifique.
- **Filtrage par Seuil:** La fonction applique un seuil (`thresh = 0.2`) pour filtrer les prédictions qui ont une probabilité inférieure à ce seuil. Seules les prédictions avec une probabilité plus élevée que 0.2 sont retenues.
- **Tri des Prédictions:** Les prédictions sont triées par ordre décroissant de probabilité. Cela permet de prioriser les tags avec les probabilités les plus élevées.
- **Création de la Liste des Résultats:** La fonction crée une liste de tags qui sont les plus probables pour le texte d'entrée. Cette liste est renvoyée comme résultat final.

Dans ce scénario applicatif, nous allons utiliser le modèle pour traiter les demandes de stages. Imaginons qu'un étudiant pose la question suivante : "Quels sont les stages disponibles en informatique ?".

Voici comment le modèle utilise la fonction de prédiction pour identifier les tags les plus pertinents :

- **Entrée de l'Utilisateur:** L'utilisateur saisit la question "Quels sont les stages disponibles en React js à Rabat ?".
- **Tokenisation et BoW:** La phrase est convertie en vecteur numérique (BoW) représentant la présence de mots-clés comme "stages", "informatique", etc.
- **Prédiction du Modèle:** Le modèle utilise ce vecteur pour prédire les tags correspondants, par exemple, *stages*, *informatique*, *offres*.
- **Filtrage et Tri:** Seuls les tags avec une probabilité supérieure à 0.2 sont retenus et triés par probabilité décroissante.
- **Résultat Final:** Le modèle retourne les tags les plus pertinents comme *stages* et *informatique*, ce qui permet au chatbot de fournir une réponse appropriée, comme une liste de stages disponibles.

Ce processus permet d'améliorer l'efficacité du chatbot dans la gestion des requêtes relatives aux stages, en garantissant que les réponses soient ciblées et pertinentes.

4.12 Génération des Réponses

```
def get_response(intents_list, intents_json, message):
    tag, msg_lower = intents_list[0], message.lower()
    df = df_analyste if extract_domaine(message) in ["data-analyste", "scientifique-de-données", "data", "bi", "power"]
    for intent in intents_json["intents"]:
        if intent["tag"] == tag:
            if tag in ["stagiaire_info", "stagiaire_partenaires_info", "encadrant_info_stagiaire", "stage_info"]:
                return test_tags(tag, msg_lower, intents_json)
            if tag == "offres_stages":
                return rechercher_profil_offres_stages(df, extract_ville(message), extract_titre_job(message), extract_random.choice(intent["responses"]) if intent["responses"] else "Réponse par défaut.")
    return "Intention non trouvée."
```

La fonction `get_response` est responsable de la génération des réponses en fonction de l'intention détectée dans le texte d'entrée. Voici les étapes principales du fonctionnement de cette fonction :

- **Identification de l'intention :** La fonction commence par identifier l'étiquette (tag) de l'intention en utilisant la première intention de la liste des intentions prédites.
- **Préparation des données :** Le message est transformé en minuscule pour normaliser les comparaisons, et les informations spécifiques telles que la ville, le domaine, le titre du poste et la compagnie sont extraites du message.
- **Sélection du DataFrame :** En fonction du domaine extrait, la fonction choisit le DataFrame approprié (`df_analyste` ou `df_jobs`) pour effectuer la recherche des offres de stage.
- **Recherche de l'intention correspondante :** La fonction parcourt les intentions définies dans

les données JSON pour trouver celle qui correspond à l'étiquette identifiée. Elle applique ensuite une logique spécifique selon l'intention détectée.

- **Génération de la réponse :** En fonction de l'intention, la fonction génère la réponse appropriée. Si l'intention est liée aux informations sur les stages, elle appelle des sous-fonctions pour obtenir des détails spécifiques. Si aucune intention correspondante n'est trouvée, une réponse par défaut est retournée.

Cette approche permet de répondre de manière dynamique aux requêtes des utilisateurs, en extrayant les informations nécessaires du message et en appliquant une logique métier spécifique à chaque intention.

4.13 Résumé des Performances

Voici un résumé de certaines époques notables :

Époque	Perte (Loss)	Précision (Accuracy)
Époque 1	2.1697	0.1296
Époque 5	1.0112	0.7037
Époque 10	0.4002	0.8889
Époque 15	0.1990	0.9630
Époque 20	0.1214	0.9444
Époque 30	0.0489	1.0000
Époque 40	0.0378	0.9815
Époque 50	0.0438	1.0000
Époque 60	0.0461	0.9815
Époque 70	0.0350	(non affiché)

Table 4.1: Résumé des performances du modèle à différentes époques.

Le tableau ci-dessus présente les performances du modèle à différentes époques d'entraînement. On observe une diminution significative de la perte (loss) au fil des époques, indiquant que le modèle améliore progressivement sa capacité à minimiser l'erreur sur les données d'entraînement. Dès l'Époque 1, où la perte est élevée à 2.1697 et la précision est faible à 0.1296, on note une amélioration constante avec une perte réduite à 0.1214 et une précision élevée à 0.9444 à l'Époque 20. La précision atteint 1.0000 à l'Époque 30, ce qui suggère que le modèle est bien adapté aux données d'entraînement à ce stade. Cependant, à partir de l'Époque 40, bien que la perte continue de diminuer, la précision fluctue légèrement, atteignant une valeur de 0.9815 à l'Époque 60. À l'Époque 70, les résultats de précision sont partiellement affichés, mais la tendance générale montre que le modèle continue à s'améliorer avec le temps. Cette tendance positive est un bon signe de la capacité du modèle à apprendre et à généraliser, bien qu'une surveillance continue soit nécessaire pour éviter le surapprentissage.

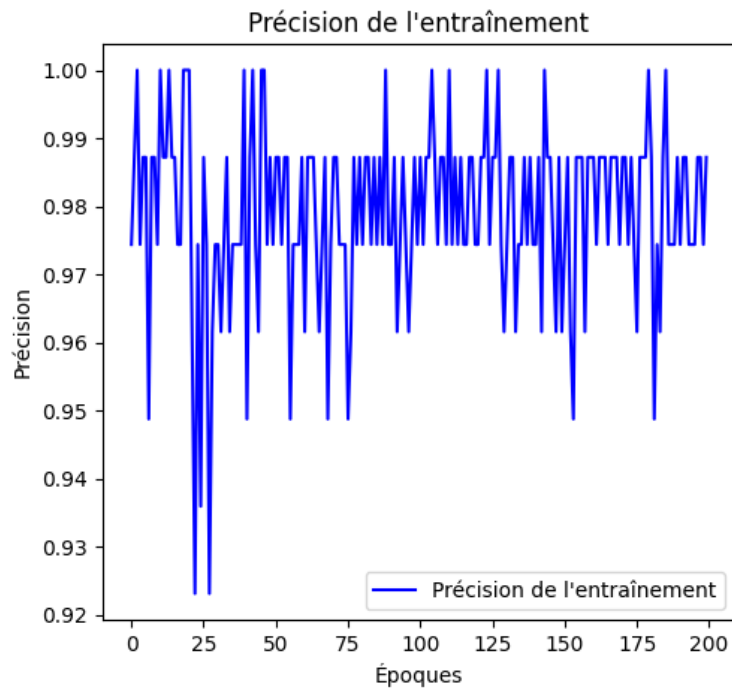


Figure 4.12: précision de entrainement

The screenshot shows the ENSA Tanger website interface. The main content area displays a list of users under the heading 'utilisateurs'. The sidebar on the left contains navigation links: Accueil, Utilisateurs, Stages, Etudiants, Encadrants, E-notes, E-Document, and Messages. A search bar is located at the top right.

Nom	Email
Hassan Badir	hassanbadir37@gmail.com
Amechnoue khalid	amechnoue.khalid@gmail.com
Azyat abdelilah	abdelilahazyat74@gmail.com
Drissi Ahmed	drissiahmed214@gmail.com
Essai Amine	amineessai341@gmail.com
Daoudia oussama	daoudiaoussama4@gmail.com
Elmounafih Achraf	mhachraf79@gmail.com

Les départements de l'ENSA Tanger sont :

- SIC : Système d'Information et Communication
- MI : Mathématiques et Informatique
- GEI : Génie Électrique et Industriel
- LCM : Langues et Communication

Figure 4.13: précision de entrainement

4.14 Conclusion sur l'intégration du bot dans la plateforme de stage

L'intégration du bot dans la plateforme de stage, enrichie par des outils de machine learning, représente une avancée significative pour améliorer l'interaction et la réponse aux utilisateurs. En appliquant des techniques d'apprentissage automatique, le modèle de bot peut être formé pour comprendre et répondre efficacement à une variété de questions et de demandes dans plusieurs domaines, tels que les informations sur les stages, les partenaires, et les départements académiques. Grâce à l'apprentissage continu et à l'optimisation des performances du modèle, le bot pourra offrir des réponses précises et pertinentes, enrichissant ainsi l'expérience utilisateur. Il est crucial de continuer à affiner le modèle en fonction des nouvelles données et des retours d'expérience pour garantir que le bot reste efficace et pertinent dans un environnement en constante évolution.

Bibliography

- [1] **Abadi, Martín, Paul Barham, Jianmin Chen et al.**, “TensorFlow: A System for Large-Scale Machine Learning,” *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283. Utilisé comme backend de Keras, populaire dans les modèles de Deep Learning.
- [2] **Chollet, François**, “Deep Learning with Python,” *Manning Publications*, 2017. Référence sur Keras et le Deep Learning en Python.
- [3] **Django Software Foundation**, “Django: The Web framework for perfectionists with deadlines,” *Django Official Documentation*, 2023. Framework web back-end utilisé pour le développement de la plateforme de suivi de stages.
- [4] **Goodfellow, Ian, Yoshua Bengio, and Aaron Courville**, “Deep Learning,” *MIT Press*, 2016. Référence complète pour les réseaux de neurones et leur utilisation dans les chatbots.
- [5] **Karpathy, Andrej**, “A Recipe for Training Neural Networks,” *Medium Blog*, 2019. Guide sur l’entraînement des réseaux de neurones, popularisé dans la communauté deep learning.
- [6] **Kingma, Diederik P and Jimmy Ba**, “Adam: A Method for Stochastic Optimization,” *arXiv preprint arXiv:1412.6980*, 2014. Présentation de l’algorithme Adam, largement utilisé pour l’optimisation en deep learning.
- [7] **McKinney, Wes**, “Data Analysis in Python with Pandas,” *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*, 2012. Référence pour la bibliothèque Pandas en manipulation de données.
- [8] **Murphy, Kevin P**, “Machine Learning: A Probabilistic Perspective,” *MIT Press*, 2012. Introduction complète aux méthodes d’apprentissage automatique, incluant le NLP et les réseaux de neurones.
- [9] **Ouyang, Long, Jeffrey Wu et al.**, “Training Language Models to Follow Instructions with Human Feedback,” *NeurIPS*, 2022. Exploration des réseaux de neurones pour les chatbots intelligents.
- [10] **React Community**, “React - A JavaScript library for building user interfaces,” *React Official Documentation*, 2023. Documentation officielle pour le développement front-end avec React.js.
- [11] **Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin**, “Attention Is All You Need,” *Advances in Neural Information Processing Systems*, 2017, 30. Article fondateur sur les Transformers, modèle NLP.