

# Mémoire de Projet de Fin d'Année

---

## Mini Système de gestion et suivie des stagiaires

---



4D Logiciels Maroc

**Mme. Amal Talha**

**Encadré par :**

M. Ayoub Metwalli  
M. Abdeljebar Manssour

Encadrante société  
Encadrant ENSATé

# REMERCIEMENTS

---

Tout d'abord, je tiens à exprimer ma gratitude envers **Allah** le Tout-Puissant pour m'avoir donné le courage et la patience nécessaires pour mener ce travail à terme.

Je tiens à remercier sincèrement mon encadrant pédagogique **Mr Abdeljebar Mansour** et mon encadrant professionnel **Mr Ayoub Metwalli** pour leur disponibilité tout au long de la durée de stage.

Je tiens également à adresser mes vifs remerciements à l'ensemble du corps professoral de l'École Nationale des Sciences Appliquées de Tétouan pour la qualité de l'enseignement dispensé durant les années de formation, et pour leur engagement dans l'accompagnement des étudiants vers l'excellence.

Je tiens également à remercier toute l'équipe de 4D Logiciels, pour leur assistance et l'ambiance conviviale qu'ils ont su instaurer

Enfin, je souhaite exprimer ma reconnaissance envers toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce travail. Je vous remercie du fond du cœur pour votre soutien et votre contribution à la réussite de ce projet.

# RÉSUMÉ

---

Dans un contexte académique et professionnel où le suivi et la gestion des stagiaires représentent un enjeu crucial pour les entreprises et les établissements de formation, la supervision efficace des missions, des projets et des échanges entre stagiaires et encadrants demeure un défi majeur. La dispersion des informations, la multiplicité des supports et l'absence d'un outil centralisé rendent le suivi des activités fastidieux et peu fiable, impactant la productivité et la qualité de l'encadrement.

Pour répondre à ces besoins, ce rapport présente la conception et le développement d'une application web dédiée à la gestion et au suivi des stagiaires. Cette solution permet :

- La centralisation des informations personnelles et académiques des stagiaires.
- La gestion des stages et le suivi des missions attribuées.
- La création et le suivi des tickets de support, incluant l'échange de commentaires entre stagiaires et encadrants.
- Une interface intuitive et interactive, favorisant l'expérience utilisateur et l'efficacité opérationnelle.

Les résultats obtenus démontrent que l'application facilite la communication, améliore la traçabilité des activités et optimise le suivi des stagiaires, contribuant ainsi à un encadrement plus structuré et efficace.

---

# TABLE DES MATIÈRES

Remerciements.....	2
Résumé.....	3
Table des Abréviations.....	5
Introduction générale.....	6
Chapitre 1: Contexte générale.....	7
Chapitre 2: Analyse et Conception.....	18
Chapitre 3: Réalisation et mise en œuvre.....	41

# LIST DES ACRONYMES

Abréviation	Signification
API	Application Programming Interface
CRUD	Create, Read, Update, Delete
DB	Database
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
GUI	Grafical User Interface
ORDA	Oriented Relationnal Data Access

# INTRODUCTION GÉNÉRALE

Dans un contexte marqué par la transformation digitale et la nécessité d'optimiser les processus organisationnels, la gestion des ressources humaines et plus particulièrement le suivi des stagiaires constitue un enjeu majeur pour les entreprises. En effet, les stages jouent un rôle essentiel dans le développement des compétences des étudiants et leur intégration progressive dans le monde professionnel. Cependant, le suivi manuel des stagiaires, à travers des documents papier ou des fichiers dispersés, engendre souvent des difficultés telles que la perte d'informations, le manque de visibilité, et une coordination limitée entre les différentes parties prenantes.

Face à ces contraintes, la mise en place d'un système web de gestion et de suivi des stagiaires s'avère indispensable. Une telle solution permet non seulement d'automatiser les tâches administratives liées aux stages, mais également d'assurer une meilleure organisation et un suivi en temps réel des parcours des stagiaires au sein de l'entreprise.

C'est dans ce cadre que s'inscrit le présent Projet de Fin d'Année (PFA), qui a pour objectif de concevoir et développer une plateforme web permettant la gestion centralisée des informations relatives aux stagiaires, depuis leur inscription jusqu'au suivi de leurs missions, en passant par la gestion des encadrants, des rapports et des évaluations. Ce projet vise ainsi à fournir un outil performant, fiable et ergonomique, répondant aux besoins réels des entreprises et contribuant à l'amélioration de l'expérience stagiaire.



---

# CONTEXT GÉNÉRALE

Ce chapitre a pour objectif de présenter l'organisme d'accueil, en spécifiant ses offres de service, de présenter les termes associés au projet et de présenter une planification à suivre durant la conception du projet afin de réaliser le travail souhaité.

## **I.1 Organisme d'accueil**

Dans cette section nous présentons l'entreprise d'accueil, son effectif et les secteurs d'activités

### **I.1.1 Présentation de 4D Logiciels Maroc.**



FIGURE I.1: Logo de 4D Logiciels

4D Logiciels Maroc est la filiale marocaine du groupe international 4D, un éditeur de logiciels de renom qui se spécialise dans la conception de plateformes de développement d'applications métiers. L'entreprise est un acteur central du paysage technologique marocain, servant de centre de services et de développement pour le groupe à l'échelle mondiale. Son siège social est situé au 46, Avenue Al Melia, à Hay Ryad, Rabat, et elle est enregistrée en tant que Société à Responsabilité Limitée à Associé Unique avec un capital social de 10 000 DHS. L'activité principale de 4D Logiciels Maroc est le conseil et le développement de logiciels dans le domaine des systèmes d'information, de l'organisation et de l'informatique, ce qui en fait un partenaire technologique stratégique pour de nombreuses entreprises.



### **I.1.1.1 Secteurs d'activités de 4D:**

de plusieurs marchés technologiques en pleine expansion. L'entreprise opère d'abord dans le domaine de l'édition de logiciels, en développant et en commercialisant une plateforme de développement unique. Cette plateforme est conçue pour permettre à d'autres entreprises et développeurs de créer leurs propres solutions logicielles métiers, en mettant l'accent sur la rapidité et l'efficacité. Le produit Qodly, en particulier, cible spécifiquement le marché du low-code et du SaaS (Software as a Service), qui connaît une croissance exponentielle à l'échelle mondiale et, de plus en plus, au Maroc.

Ce secteur d'activité est caractérisé par une demande croissante pour la transformation numérique des entreprises, qu'il s'agisse de start-ups cherchant à prototyper rapidement leurs idées ou de grandes entreprises souhaitant optimiser leurs processus internes. La philosophie "low-code" répond directement à cette demande en réduisant le temps de mise sur le marché des applications tout en limitant les coûts de développement. 4D Logiciels Maroc joue un rôle clé dans cet écosystème en fournissant non seulement la technologie de base (Qodly) mais aussi une expertise en services informatiques. Son équipe de consultants et de développeurs intervient directement auprès des clients pour les accompagner dans la conception, le développement, la modernisation et la maintenance de leurs applications, agissant ainsi comme un partenaire technologique et un centre de compétences de premier plan pour le groupe 4D.

4D opère dans le domaine des technologies de l'information et de la communication (TIC), avec une spécialisation dans :

- L'édition logicielle : 4D développe et commercialise sa propre plateforme de développement, reconnue pour sa capacité à créer des applications performantes et multiplateformes (macOS, Windows, mobile, web).
- Les services informatiques : L'entreprise fournit des prestations de conseil, de développement et de maintenance pour ses clients, constituant ainsi un véritable partenaire technique.

4D Logiciels Maroc, via son équipe 4D Professional Service basée à Rabat, offre un éventail de services pour accompagner les clients dans leurs projets de développement :

- Développement d'applications métiers : Création de solutions logicielles sur mesure pour répondre aux besoins spécifiques des entreprises dans divers secteurs, tels que la santé, l'industrie ou l'administration. Les services incluent l'analyse, la conception, le développement, et le déploiement.
- Modernisation et optimisation : Assistance pour la migration d'applications existantes vers les dernières versions de la plateforme 4D, l'audit de code, et l'optimisation des performances des systèmes.
- Renforcement d'équipes techniques : L'équipe de 4D Maroc peut s'intégrer de manière ponctuelle ou durable aux équipes de développement des clients pour apporter une expertise technique pointue et aider à surmonter les défis complexes.
- Formation : L'entreprise propose des formations spécialisées pour les développeurs souhaitant se perfectionner sur la plateforme 4D, notamment sur des composants spécifiques comme 4D Write Pro ou 4D View Pro.
- Expertise technologique : Les développeurs de 4D Maroc possèdent des compétences au-delà de la plateforme 4D, incluant le développement web, mobile, l'infrastructure Cloud et l'analyse de données, ce qui leur permet de relever une grande variété de défis techniques.



FIGURE I.2: schéma d'accompagnement

### **I.1.2 Qodly:**

Qodly est une plateforme de développement d'applications hybride en mode SaaS (Software as a Service) développée par la société 4D. Contrairement aux plateformes "no-code" qui ne permettent pas d'ajouter de code, ou aux outils traditionnels qui sont lents, Qodly se positionne dans le segment du "low-code". Cette approche permet aux développeurs de combiner des outils de conception visuelle rapides et intuitifs avec la liberté et la puissance de la programmation traditionnelle. L'objectif est de leur permettre de créer des applications métier web et des solutions SaaS robustes et performantes, sans les contraintes de temps et de ressources liées à la programmation manuelle de chaque élément.

La plateforme se compose principalement de deux parties distinctes qui fonctionnent en parfaite synergie. D'une part, on trouve le Qodly Studio, un environnement de développement web visuel et interactif. Il permet aux développeurs de concevoir l'interface utilisateur des applications grâce à une interface de glisser-déposer (Drag and Drop). Ils peuvent y créer des formulaires, organiser les éléments et lier les composants à des sources de données en quelques clics, tout en ayant un aperçu instantané du résultat. Cette partie de l'outil vise à automatiser les tâches de conception les plus fastidieuses. D'autre part, Qodly Cloud Console sert de tableau de bord central pour la gestion des applications. Elle facilite le déploiement en un clic dans des environnements de test ou de production, ainsi que la surveillance et la mise à jour des applications en cours de fonctionnement.

Le modèle hybride de Qodly constitue son principal avantage concurrentiel. Il offre aux développeurs la rapidité du "low-code" pour les tâches courantes et répétitives, et la flexibilité du code pour les fonctionnalités complexes et personnalisées. Les développeurs ne sont jamais bloqués par les limitations de la plateforme et peuvent toujours ajouter leur propre code pour créer des logiques métier uniques.

## **I.3 Présentation générale du projet**

### **I.3.1 Problématique**

Dans de nombreuses entreprises, la gestion et le suivi des stagiaires se fait encore de manière traditionnelle, à travers des fichiers Excel, des documents papier ou des échanges informels.

Cette approche présente plusieurs limites :

- Perte et dispersion de l'information : les données concernant les stagiaires, leurs encadrants, leurs missions et leurs rapports ne sont pas centralisées, ce qui complique la consultation et le suivi.
- Manque de visibilité et de traçabilité : les responsables rencontrent des difficultés à assurer un suivi efficace du parcours des stagiaires et à évaluer l'avancement de leurs travaux.
- Charge administrative élevée : la saisie et la gestion manuelles des informations entraînent une perte de temps et un risque d'erreurs.
- Communication limitée : l'absence d'une plateforme commune rend la collaboration entre stagiaires, encadrants et responsables peu fluide.

Ces constats mettent en évidence le besoin d'un outil informatique moderne, fiable et centralisé, permettant d'optimiser la gestion et le suivi des stagiaires au sein de l'entreprise.

### **I.3.2. Solution**

Pour répondre aux limites identifiées, nous proposons la mise en place d'un système web de gestion et de suivi des stagiaires au sein de l'entreprise. Ce système vise à centraliser toutes les informations relatives aux stagiaires, à automatiser certaines tâches administratives et à offrir une interface simple et ergonomique aux différents acteurs (administrateur, encadrants et stagiaires).

La solution repose sur les axes suivants :

1. Centralisation des données : toutes les informations concernant les stagiaires (coordonnées, parcours académique, missions, rapports, évaluations) seront regroupées dans une base de données unique et sécurisée.
2. Gestion des utilisateurs et des rôles : le système distingue plusieurs profils (administrateur, encadrant, stagiaire) avec des droits d'accès adaptés.
  - Administrateur : gestion des comptes, des stages, des encadrants et des stagiaires.
  - Encadrant : suivi des stagiaires, validation des missions et des rapports.
  - Stagiaire : consultation des informations personnelles, dépôt des documents et suivi des missions de l'encadrant.
3. Suivi en temps réel : le système permet de visualiser l'état d'avancement des stages, les rapports déposés, les missions des encadrants et les tickets des stagiaires.
4. Automatisation : réduction de la charge administrative par la génération automatique des attestations et la traçabilité des interactions.
5. Accessibilité et ergonomie : grâce à une interface web responsive, la plateforme sera accessible à tout moment et depuis n'importe quel support.

Ainsi, le système proposé offre une solution moderne, fiable et performante, permettant à l'entreprise de gagner en efficacité, de faciliter la communication entre les différents acteurs et d'assurer un meilleur suivi des stagiaires.

### **I.3.3. Objectif du Projet**

L'objectif principal du système web de gestion et suivi des stagiaires est de fournir une plateforme centralisée, fiable et accessible, permettant d'optimiser la gestion des stages et de faciliter la communication entre les différents acteurs.

De manière plus spécifique, le système vise à :

1. Automatiser la gestion des stagiaires : réduire la charge administrative liée à la saisie et au traitement manuel des informations.
2. Améliorer le suivi : permettre aux encadrants et responsables d'assurer un suivi en temps réel de l'avancement des stagiaires.
3. Faciliter la communication : offrir un espace d'échange structuré entre stagiaires et encadrants (remarques, validations, rapports).
4. Assurer la centralisation et la traçabilité des données : garantir que toutes les informations (stages, missions, tickets, évaluations, documents) soient regroupées dans une base unique et consultables à tout moment.
5. Renforcer l'efficacité et la transparence : améliorer la visibilité sur l'ensemble des stages en cours, leur état d'avancement, et les performances des stagiaires.
6. Garantir l'accessibilité : proposer une interface web ergonomique et responsive, utilisable aussi bien sur ordinateur que sur mobile.

En atteignant ces objectifs, ce système pourra moderniser le processus de gestion des stages, tout en apportant un gain de temps, une meilleure organisation et une meilleure expérience aussi bien pour l'entreprise qui souhaitera accueillir les stagiaires après la durée du stage, que pour les stagiaires.

## I.4 Conduite du projet

### I.4.1 planification du projet

La planification du projet représente une étape essentielle dans la phase de préparation. Elle permet d'organiser et de structurer les différentes tâches à réaliser tout au long du cycle de développement.

Dans un premier temps, une planification prévisionnelle a été élaborée en début de projet afin d'organiser les grandes étapes de développement selon une logique progressive.

La figure suivante illustre la planification initiale :



FIGURE I.3: Diagramme de Gantt prévisionnel

Ainsi, une planification réelle a été reconstituée en fin de parcours, pour refléter l'évolution effective du projet et les ajustements opérés au fil du temps.

La figure suivante présente cette planification réelle :



FIGURE I.4: Diagramme de Gantt réel

Cette double visualisation permet de comparer l'organisation prévue avec le déroulement effectif, et met en évidence la capacité d'adaptation nécessaire pour mener à bien le projet dans un environnement professionnel dynamique.



## **I.5 Conclusion**

Ce chapitre met le point sur le contexte général du projet, dans la mesure où il décrit l'organisme d'accueil, le projet et ses objectifs ainsi que les étapes de mise en oeuvre. Le chapitre suivant concerne l'analyse et conception du projet.



---

# ANALYSE ET CONCEPTION

Ce chapitre est dédié à l'analyse et à la conception du projet. Il débute par le cadrage du besoin, à travers l'état de l'art et la spécification des exigences. Il présente ensuite les différents acteurs impliqués dans le système ainsi que leurs rôles. Enfin, les principales composantes de la solution seront modélisées à l'aide des diagrammes UML les plus pertinents.

## II.1 Phase d'analyse

### II.1.1 Etat de l'art

Le suivi des stagiaires dans les entreprises peut être assuré de différentes manières.

Traditionnellement, il se fait à travers des fichiers Excel, des rapports papiers ou encore des réunions ponctuelles, mais ces méthodes restent limitées et peu adaptées à une gestion moderne et centralisée.

Avec l'évolution des pratiques de gestion de projet et l'adoption des méthodologies agiles telles que Scrum ou Kanban, plusieurs entreprises se sont tournées vers des outils collaboratifs pour organiser et suivre le travail des stagiaires. Ces outils, bien qu'initialement conçus pour la gestion de projets informatiques, sont également utilisés pour accompagner les stagiaires dans leurs missions. Parmi les plus connus, on peut citer :

- Jira : outil largement utilisé en entreprise pour la gestion de projets agiles (Scrum, Kanban). Les encadrants l'emploient pour assigner des tâches aux stagiaires, suivre leur avancement et gérer les sprints.
- Trello : une solution simple basée sur des tableaux Kanban, permettant d'organiser les missions des stagiaires sous forme de cartes et de listes.
- Microsoft Teams : intégrés dans l'écosystème Microsoft, ces outils permettent de gérer les tâches, partager des documents et organiser des réunions de suivi.
- GitLab / GitHub Projects : pour les stagiaires en informatique, ces plateformes offrent un suivi intégré du code source et des tâches.

#### Limites des outils existants

Bien que ces outils soient efficaces, ils ne sont pas spécifiquement conçus pour la gestion globale des stagiaires dans une entreprise. Ils nécessitent souvent une configuration complexe, ne centralisent pas les informations administratives (coordonnées, attestations, rapports de stage), et ne permettent pas un suivi adapté aux besoins RH.

C'est dans ce contexte que s'inscrit le présent projet, qui propose un système web dédié à la gestion et au suivi des stagiaires, combinant la facilité d'utilisation d'une application interne avec les fonctionnalités de suivi adaptées aux besoins réels d'une entreprise.

### II.1.2 Identification des Acteurs

Un acteur représente un rôle ou une entité qui interagit directement avec le système modélisé. Cet acteur peut être un utilisateur humain, un service externe, une machine ou une organisation. L'identification précise des acteurs permet de mieux comprendre les interactions possibles avec le système et de définir les cas d'utilisation associés.

*1-Les stagiaires*: Ce sont les étudiants ou les diplômés accueillis par l'entreprise dans le cadre de leur formation académique ou professionnel. Ils utilisent le système pour se connecter à leur comptes existants, accéder à leurs missions, déposer des tickets de support et recevoir les missions de leurs encadrants. Pour eux, la plateforme constitue un outil simple et accessible qui facilite leur intégration et leur suivi tout au long du stage.

*2-Les encadrants*: Ce sont généralement des ingénieurs, des responsables techniques ou des chefs d'équipe désignés pour accompagner les stagiaires. Leur rôle est de guider les stagiaires dans leurs missions, de valider leur travail et de leur fournir des évaluations. Grâce au système, ils disposent d'un espace dédié pour suivre l'avancement des stagiaires, consulter leurs rapports et communiquer efficacement avec eux.

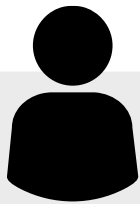
*3-L'administrateur ou Responsables RH*: C'est la personne en charge de la gestion globale de la plateforme. Il peut s'agir d'un membre du service des ressources humaines ou d'un responsable de formation. L'administrateur gère les comptes utilisateurs, attribue les stagiaires aux encadrants et supervise le bon déroulement du processus. Son rôle est essentiel pour assurer l'organisation et la cohérence de l'ensemble du système.

Chaque acteur joue un rôle clé dans le bon fonctionnement du système, depuis sa mise en place jusqu'à son utilisation quotidienne.

### II.1.3 Spécification des besoins

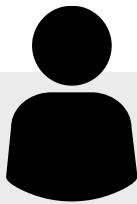
Il est important de souligner que certains modules peuvent être partagés entre différents types d'utilisateurs, mais leurs exigences spécifiques peuvent varier en fonction du rôle ou du profil de l'acteur concerné.

#### STAGIAIRE



- **Authentification** : se connecter via email/mot de passe (JWT dans cookie).
- **Consultation des informations personnelles** : voir ses affectations, son encadrant, ses missions, ses dates de stage.
- **Dépôt de documents** : soumettre son rapport final.
- **Suivi des missions d'un stage** : consulter l'état d'avancement de ses missions.
- **Réception des évaluations** : visualiser les commentaires et notes de son encadrant.
- **Dépôt et suivi des tickets de support** : mettre à jour certaines informations personnelles si autorisé.

#### ADMINISTRATEUR



- **Authentification** : se connecter via email/mot de passe.
- **Gestion des utilisateurs** : créer, modifier et supprimer les comptes stagiaires et encadrants.
- **Affectation des stagiaires** : assigner les stagiaires aux encadrants selon les besoins de l'entreprise.
- **Gestion des stages** : suivre l'état global des stages, document d'évaluation et des rapports.
- **Supervision et statistiques** : Consulter les statistiques sur les stages et les utilisateurs.

#### ENCADRANT



- **Authentification** : se connecter via email/mot de passe.
- **Gestion des stages** : consulter la liste des stages (affectations) qui lui sont assignés.
- **Suivi des missions** : suivre l'avancement des missions des stagiaires.
- **commentaires** : Consultation des tickets et les commenter, Evaluation du stagiaire et génération de document d'évaluation.
- **Génération de rapports** : éventuellement télécharger des rapports d'avancement des stagiaires.

FIGURE II.1: Les modules de l'application par acteur

### II.1.4 Spécification des besoins non-fonctionnelles

Les besoins non fonctionnels désignent les exigences qui ne portent pas directement sur les services rendus par le système, mais sur ses propriétés globales et les contraintes qu'il doit respecter. Ils sont essentiels pour garantir la qualité, la robustesse et l'acceptabilité de la solution au sein d'un environnement professionnel. Dans le cadre de notre projet, plusieurs critères non fonctionnels doivent être pris en compte afin d'assurer une intégration fluide et efficace dans des architectures dynamiques.

- **Performance:** Le système doit être rapide et réactif. Les utilisateurs ne doivent pas attendre longtemps pour accéder à leurs informations, consulter un rapport ou déposer un document. Même si plusieurs utilisateurs travaillent en même temps, le système doit rester fluide.
- **Sécurité:** Les données des stagiaires et des encadrants sont sensibles. Le système doit donc protéger ces informations contre tout accès non autorisé. Chaque utilisateur ne peut accéder qu'aux informations qui le concernent, grâce à une connexion sécurisée (login/mot de passe, JWT dans cookie) et middleware de contrôle d'accès aux ressources.
- **Ergonomie et convivialité:** L'interface doit être simple et intuitive, pour que chacun puisse accomplir ses tâches facilement, sans avoir besoin de formation particulière. L'objectif est de rendre l'expérience utilisateur agréable et naturelle.
- **Évolutivité et maintenabilité:** Le système doit pouvoir évoluer facilement. Par exemple, il doit être possible d'ajouter de nouvelles fonctionnalités ou de modifier certaines parties sans devoir tout reconstruire. Le code et la base de données doivent être clairs pour faciliter les interventions futures.

Pour assurer une adoption efficace de la solution et offrir une expérience utilisateur optimale, il est indispensable de concevoir une application qui respecte ces exigences non fonctionnelles.

### II.1.4.2 Diagramme de Cas d'Utilisation

Le diagramme de cas d'utilisation en UML permet de représenter les interactions entre les acteurs et le système, en mettant en évidence les fonctionnalités principales du point de vue utilisateur. Il décrit ce que fait le système sans détailler comment il le fait techniquement.

Dans le cadre de notre application, ce diagramme permet d'illustrer les différents cas d'utilisation associés aux acteurs identifiés, et de clarifier la portée fonctionnelle du système.

Les diagramme ci-dessous présente les principales interactions avec la solution d'intégration Utilisateurs – système :

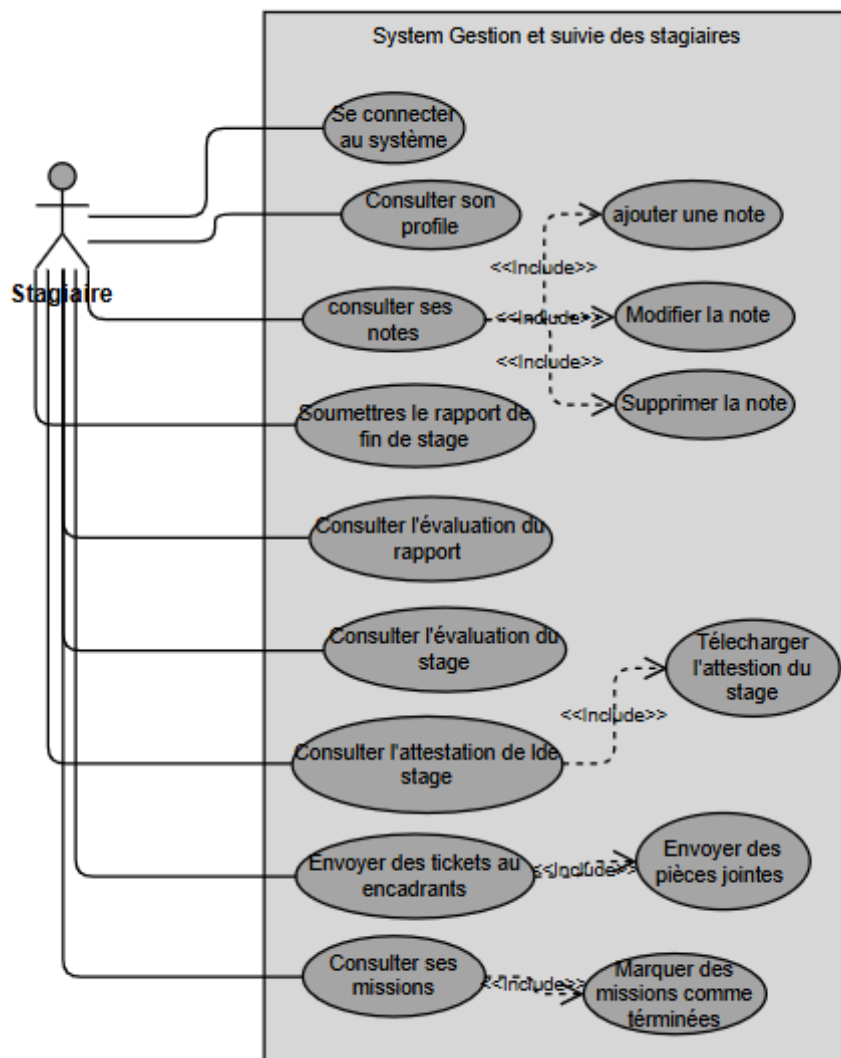


FIGURE II.2: diagramme de cas d'utilisations du stagiaire

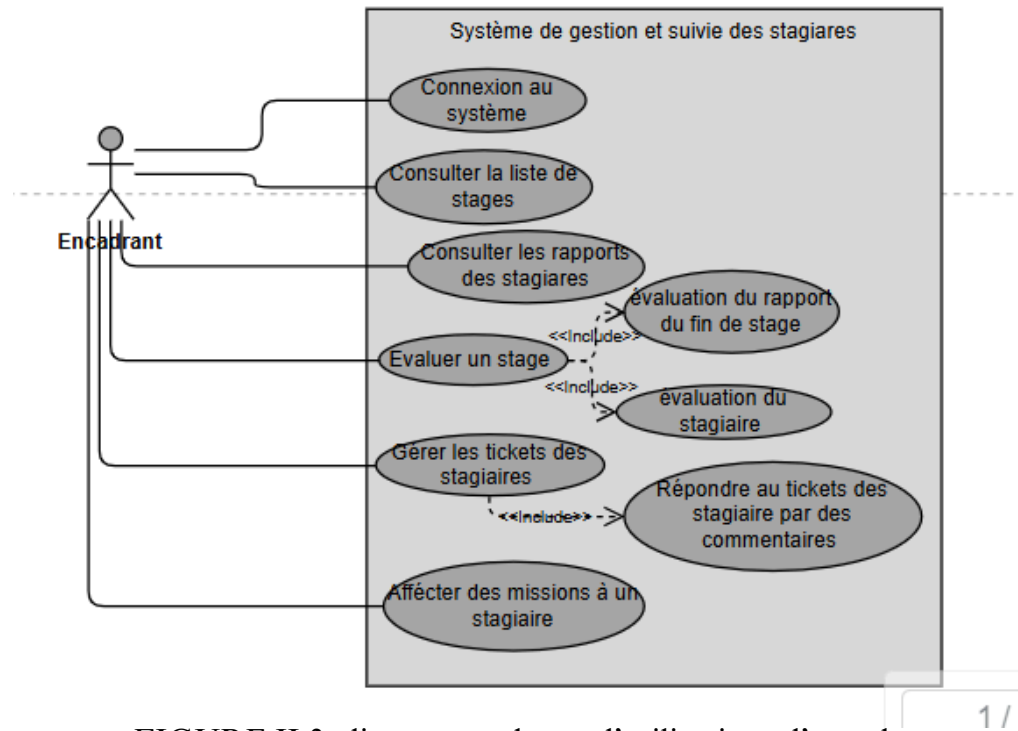


FIGURE II.3: diagramme de cas d'utilisations d'encadrant

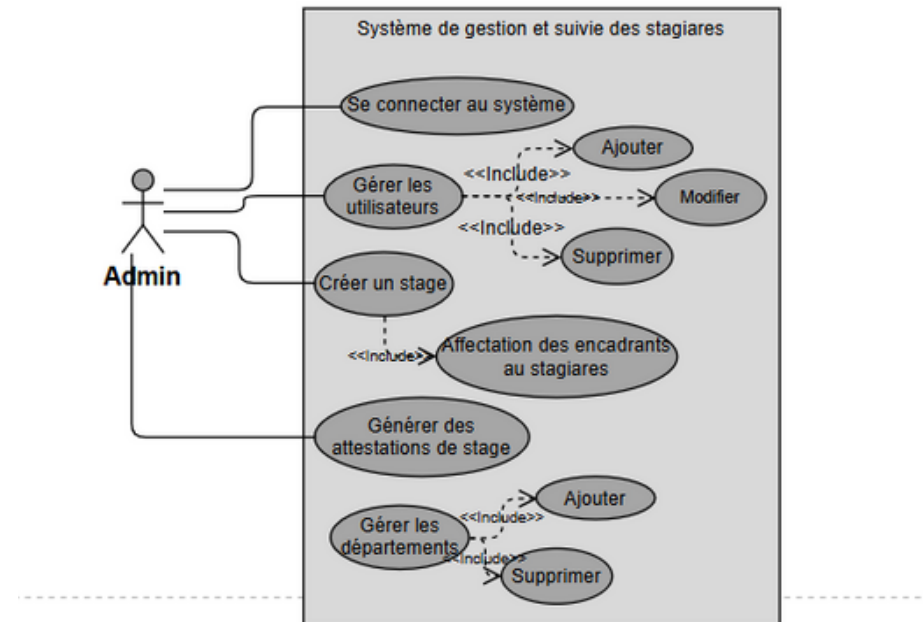


FIGURE II.4: diagramme de cas d'utilisations d'Admin



Ces cas d'utilisation couvrent l'ensemble des interactions clés entre les différents acteurs et le système de gestion et de suivie des stagiaires. Ils permettent de structurer les fonctionnalités principales de l'application en tenant compte des responsabilités spécifiques de chaque profil.

## II.2 Phase de conception

### II.2.1 L'architecture du projet

L'objectif de cette phase est de représenter de manière claire et structurée l'architecture fonctionnelle de la solution. Elle met en lumière les composants essentiels du système, leurs relations, ainsi que les interactions avec les utilisateurs et le système.

Elle inclut également une vue globale du workflow d'interaction, illustrant comment les utilisateurs final accèdent aux données de façon sécurisée et fluide.

Le schéma suivant illustre l'architecture globale de la solution d'intégration.

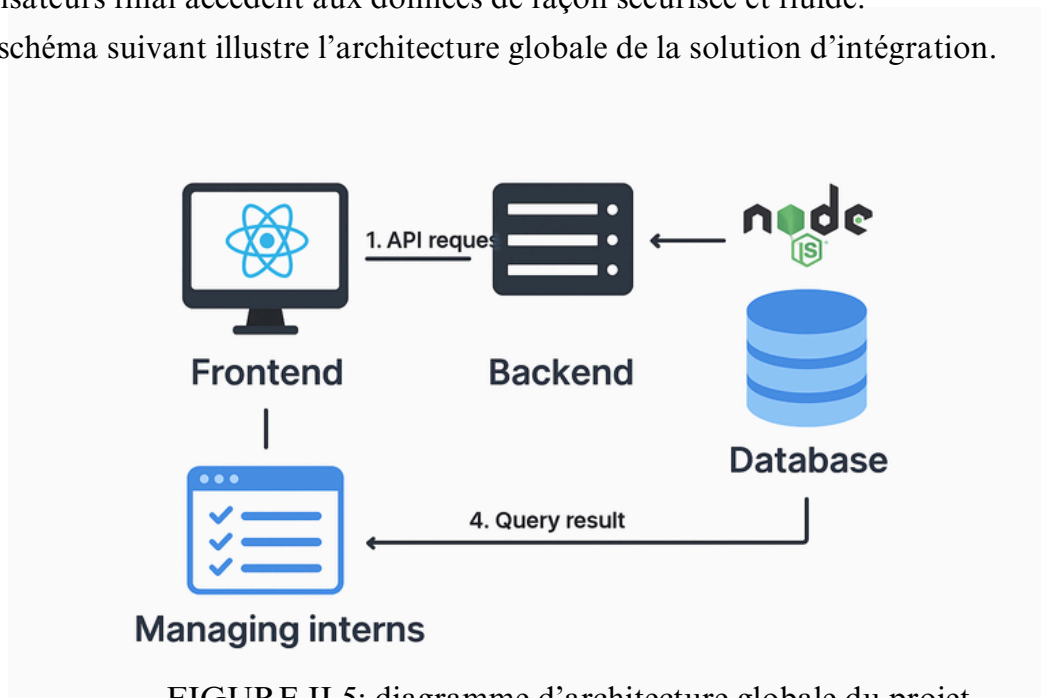


FIGURE II.5: diagramme d'architecture globale du projet

## II.2.1 Diagramme de séquence

Dans ce qui suit, nous présentons les diagrammes de séquence correspondant aux scénarios les plus complexes de l'application.

Ces diagrammes jouent un rôle de feuille de route, en décrivant de manière précise l'enchaînement des interactions entre les différents composants du système, et serviront de référence lors du développement des processus métiers.

### II.2.1.1 Le parcours des Requête API REST.

Le diagramme de séquence ci-dessous décrit l'interaction entre l'administrateur, l'interface, les APIs REST et la base de données, lors de la création d'une connexion sécurisée.

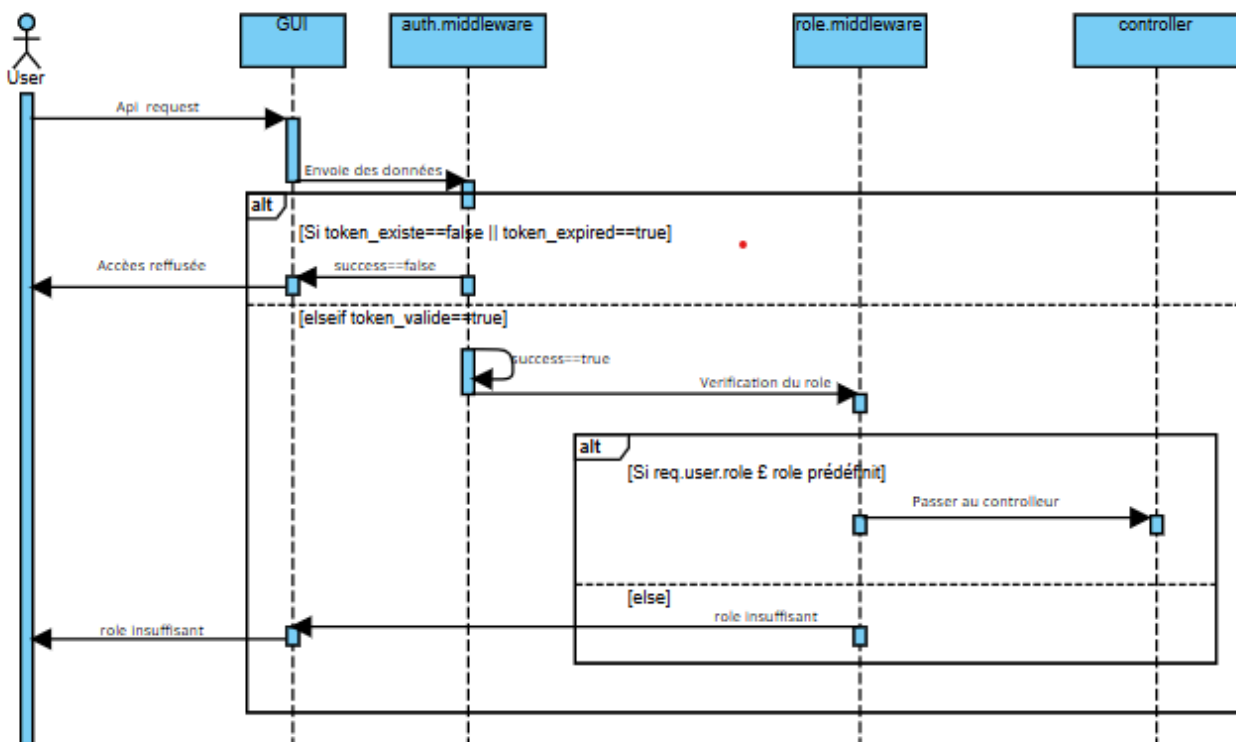


FIGURE II.6: Diagramme de séquence d'authentification et d'autorisation middlewares

Ce diagramme de séquence illustre le processus initial d'une API REST envoyé par un admin, avant d'entamer l'exécution de la logique décrit par les contrôleurs. En effet, pour exécuter une actions l'utilisateur doit d'abord être déjà authentifié, chose qui sera vérifié par le auth.middleware, puis le role.middleware vérifie s'il s'agit bien d'un JWT qui correspondant à un administrateur par la vérification du rôle de l'utilisateur connecté. En fin, si les étapes précédentes sont réalisées avec succès on aura l'autorisation pour exécuter le contrôleur.

### II.2.1.2 le login:

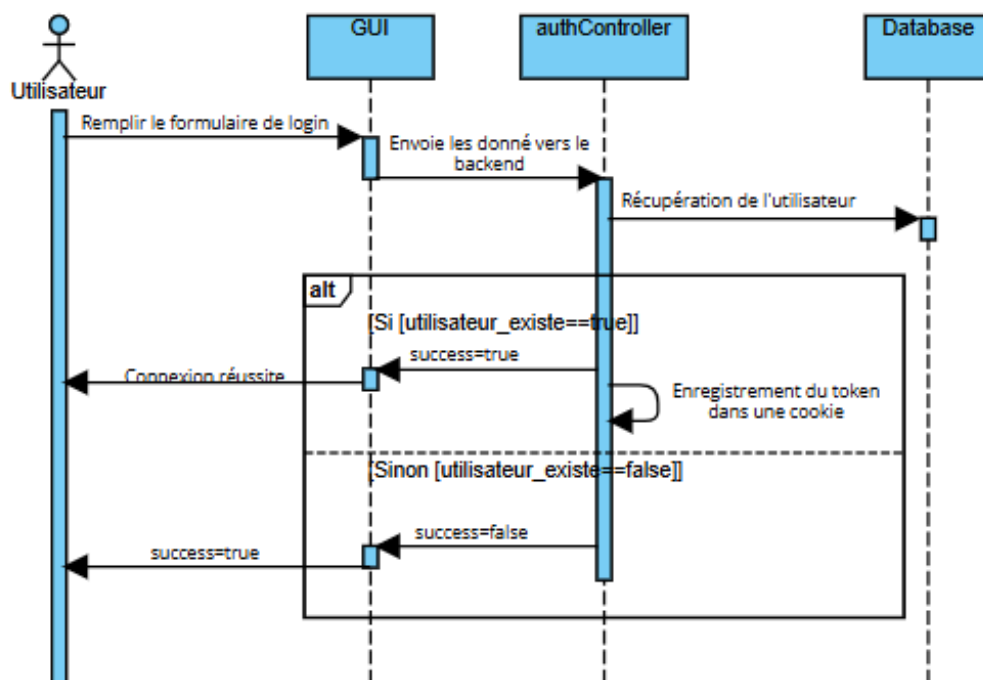


FIGURE II.7: Diagramme de séquence de login

Ce diagramme de séquence illustre le processus de login, où on doit noter que cet API n'arrive au contrôleur que s'il a bien passé par les middleware noté dans le diagramme de la FIGURE II.4.

### II.2.1.3 Enregistrer un utilisateur (Stagiaire ou encadrant) dans le système:

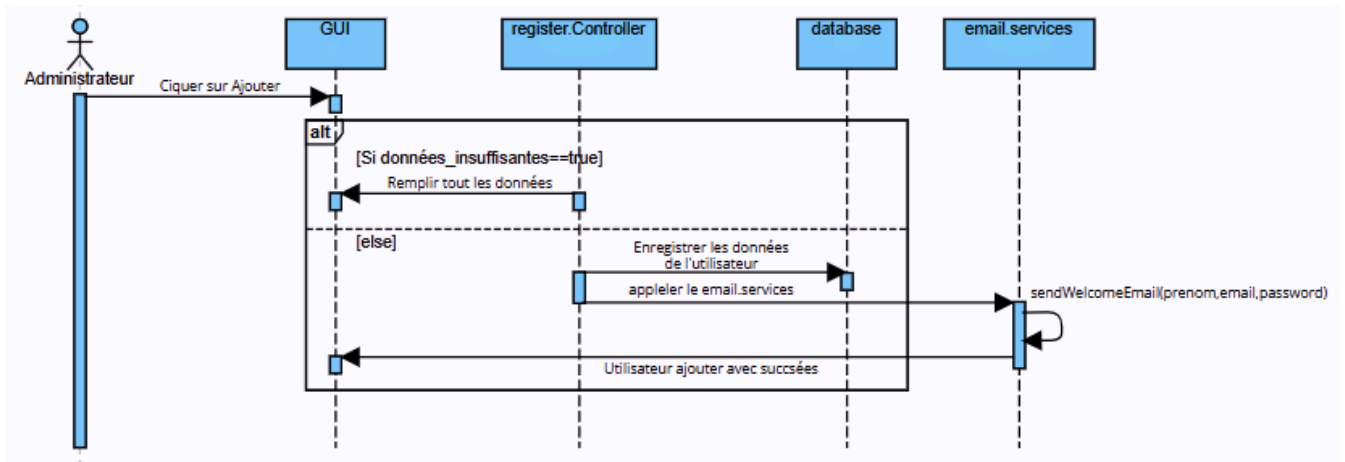


FIGURE II.8: Enregistrer un utilisateur (Stagiaire ou Encadrant) dans le système

Le diagramme suivant illustre le processus d'enregistrement d'un utilisateur (Stagiaire ou Encadrant) dans le système. Après avoir enregistré les données de l'utilisateur dans la base de données, un service sera appelé pour envoyer à ce dernier un email contenant son mot de passe afin qu'il pourra se connecter au système.

### II.2.1.4 Création d'un département:

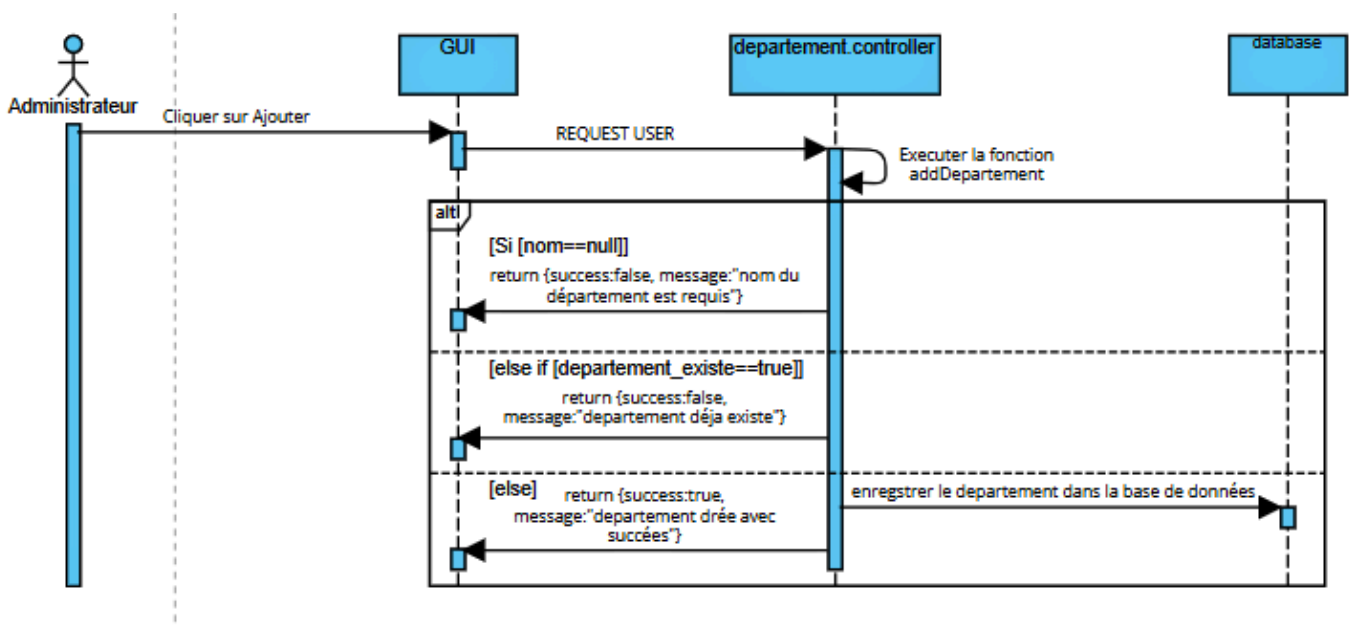


FIGURE II.9: Diagramme de séquence de la création d'un département

### II.2.1.4 Diagramme de séquence se la suppression du département:

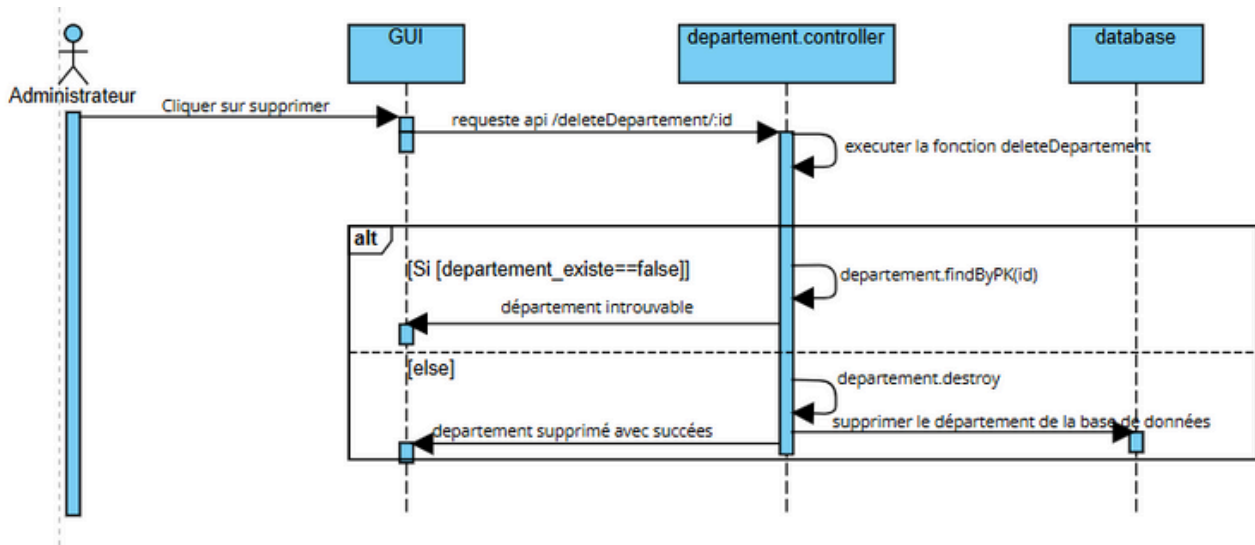


FIGURE II.10: Diagramme de séquence de la suppression d'un departement

### II.2.1.4 Diagramme de séquence de la suppression d'un utilisateur:

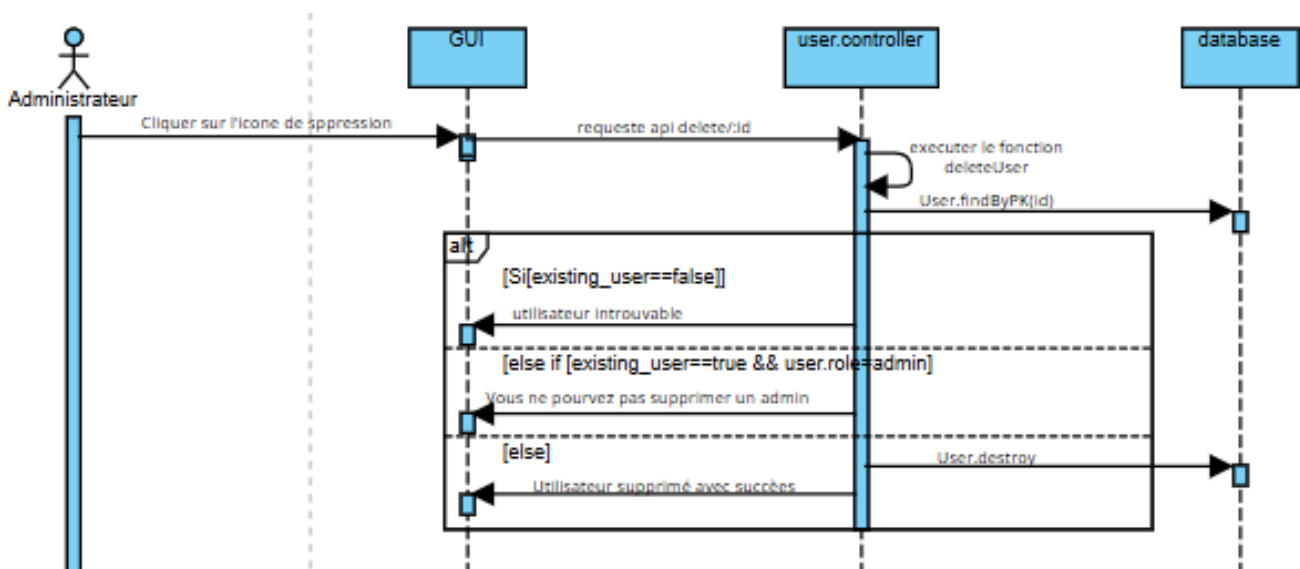


FIGURE II.11: Diagramme de séquence de la suppression d'un utilisateur

L'administrateur peut supprimer un utilisateur, en effet ceci aboutie à la suppression de toute affectation lié à ce derniers.

#### II.2.1.4 Diagramme de séquence d'ajout d'une affectation.

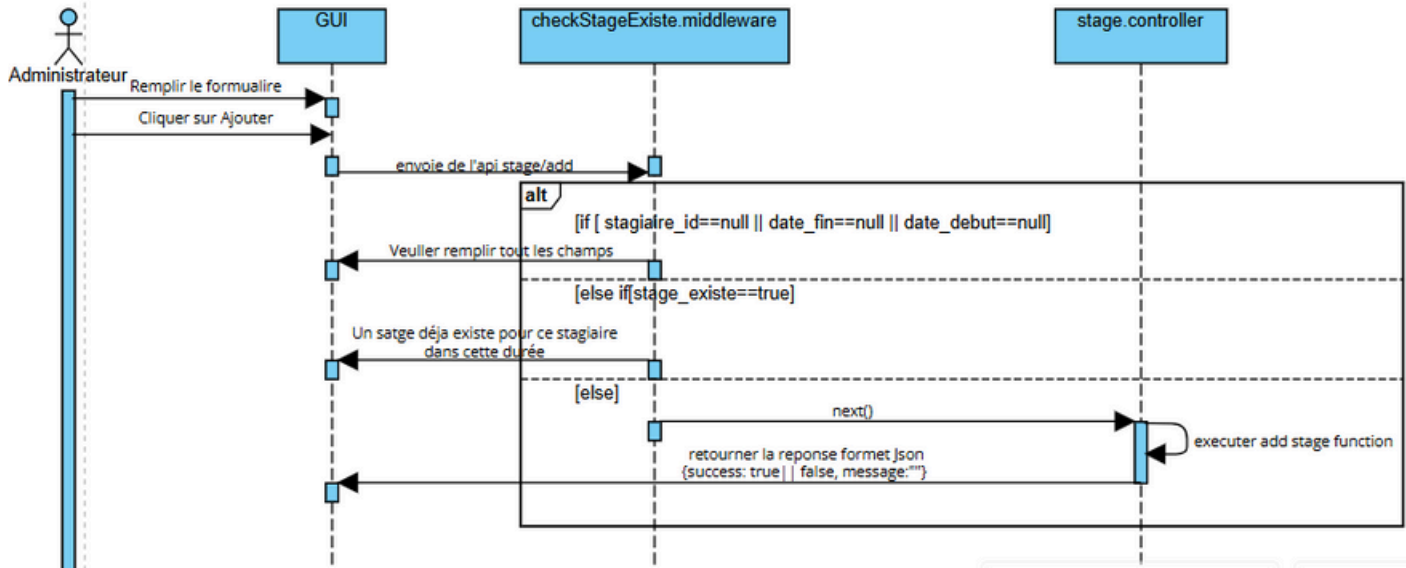


FIGURE II.12: Diagramme d'ajout d'une affectation (Stage) dans le système

Ce diagramme illustre le processus de l'ajout d'une nouvelle affectation, alors le middleware `checkStageExiste.middleware` Vérifie s'il existe un stage ayant le meme stagiaire dans le JSON entrés par l'administrateur, et ayant une date de fin supérieur à la date de début entré, ou une date de début inférieur à la date de fin entrée. Si c'est trouvé on notifie l'utilisateur par le meassage "Vous nous pouvez pas créer deux stage dans la meme durée", Sinon on execute la logique du controlleur.

### II.2.1.4 Diagramme de séquence de génération de l'attestation d'un stage:

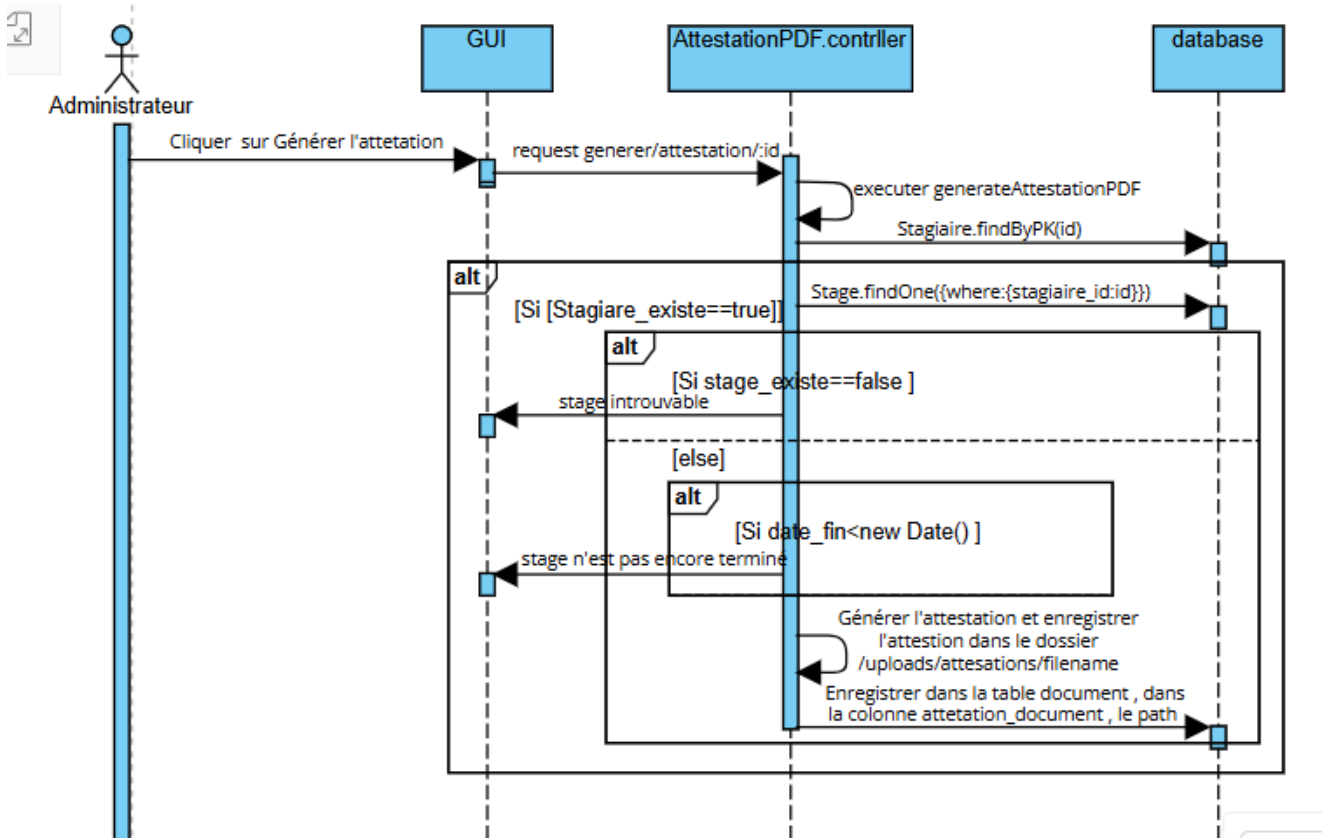


FIGURE II.13: Diagramme de séquences d'ajout d'un Stage

### II.2.1.4 Gestion des droits d'accès aux données.

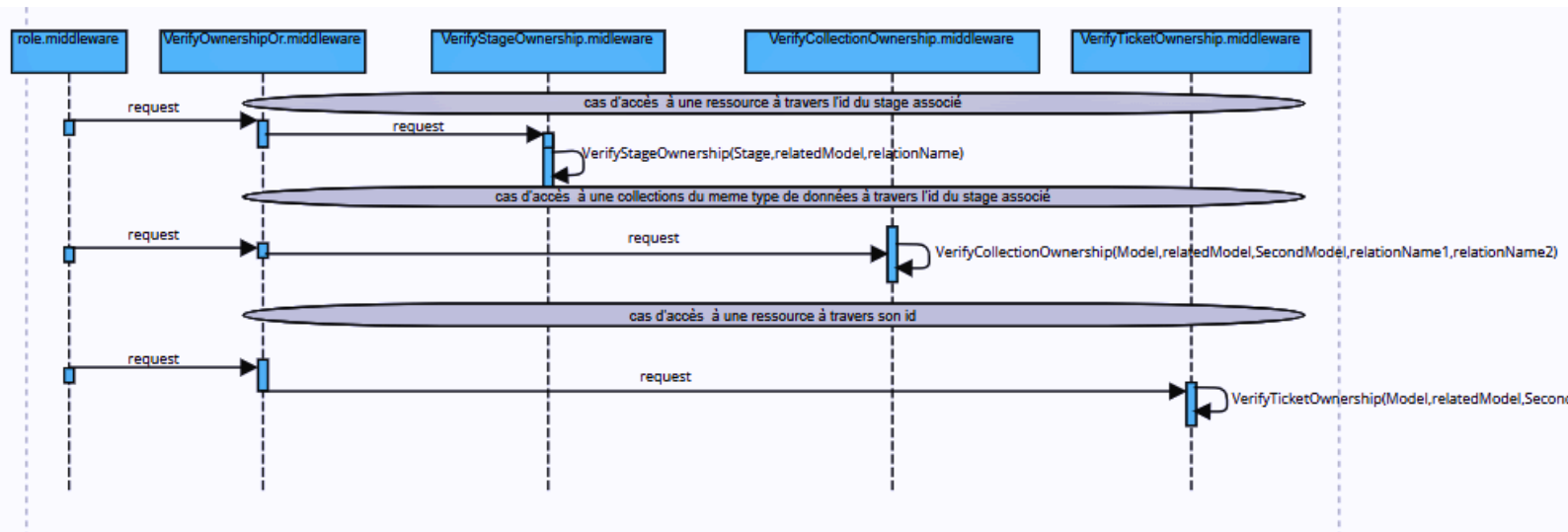


FIGURE II.14: Diagramme de séquences de processus de verification du droit d'accès aux données

Le processus commence par un premier filtre, le `Role.middleware`, qui s'assure que l'utilisateur a le bon rôle avant de continuer. Une fois ce premier contrôle passé, la requête est transmise à un "gestionnaire" central appelé `VerifyOwnershipOr.middleware`. C'est ce middleware qui est responsable de déléguer la vérification de la propriété à d'autres modules spécialisés, en fonction du type de ressource demandée. Le diagramme met en évidence trois scénarios distincts de vérification de propriété :

Le premier scénario concerne l'accès à une ressource via l'ID d'un "stage" associé, un cas d'usage courant pour des données liées à une étape ou un événement. Le `VerifyOwnershipOr.middleware` transmet alors la requête au `VerifyStageOwnership.middleware` qui se charge de vérifier la propriété. Le deuxième scénario, un peu plus complexe, s'applique lorsqu'on accède à une collection de données via le même ID de "stage". Dans ce cas, la vérification est confiée au `VerifyCollectionOwnership.middleware`, qui doit s'assurer de la propriété sur un ensemble de données. Enfin, le troisième scénario représente un accès direct à une ressource spécifique, comme un ticket, via son propre ID. La vérification est alors déléguée à un module dédié, le `VerifyTicketOwnership.middleware`.

Ce diagramme met en lumière une conception modulaire et sécurisée, où la logique de vérification de propriété est clairement séparée et organisée de manière hiérarchique. Cela simplifie la maintenance du code, en isolant les responsabilités et en rendant la sécurité plus facile à auditer.



### II.2.1.5 le Processus de CRUD d'une mission (coté Encadrant).

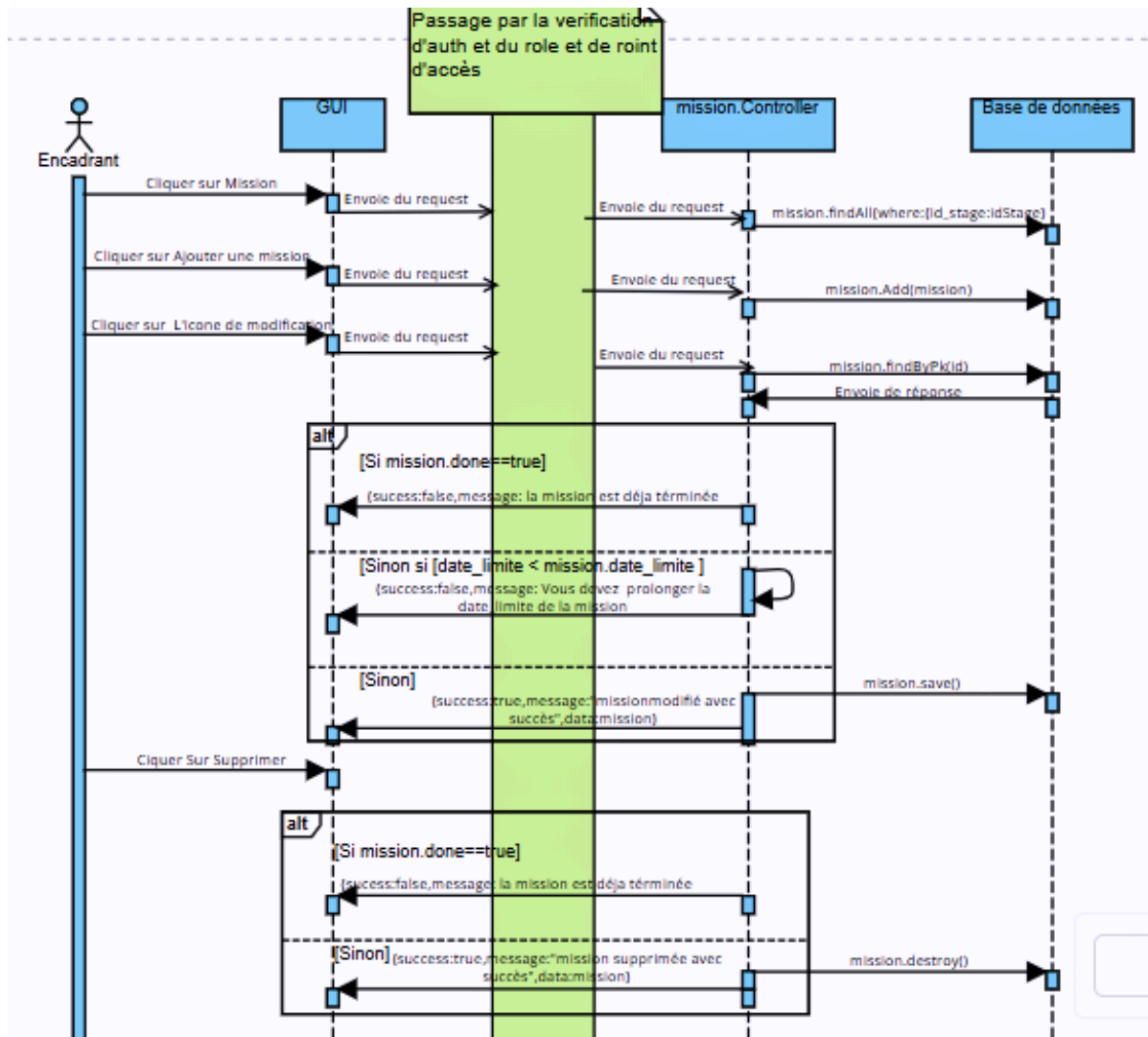


FIGURE II.14: Diagramme de séquence de processus de gestion de smission (coté Encadrant)

Le diagramme ci dessous illustre les séquences pour la gestion d'une missions dans un stage par l'Encadrant. diagramme de séquence décrit le flux d'interactions entre un encadrant et une application pour gérer les missions. Tout d'abord, on remarque que chaque action de l'encadrant, qu'il s'agisse d'afficher, d'ajouter, de modifier ou de supprimer une mission, passe par une étape initiale de vérification de l'authentification et du rôle, garantissant ainsi que seules les personnes autorisées peuvent effectuer ces opérations.

### II.2.1.5 le Processus de CRUD d'une mission (coté Stagiaire).

Le diagramme ci dessous illustre les séquences pour la gestion d'une missions dans un stage par l'Encadrant.

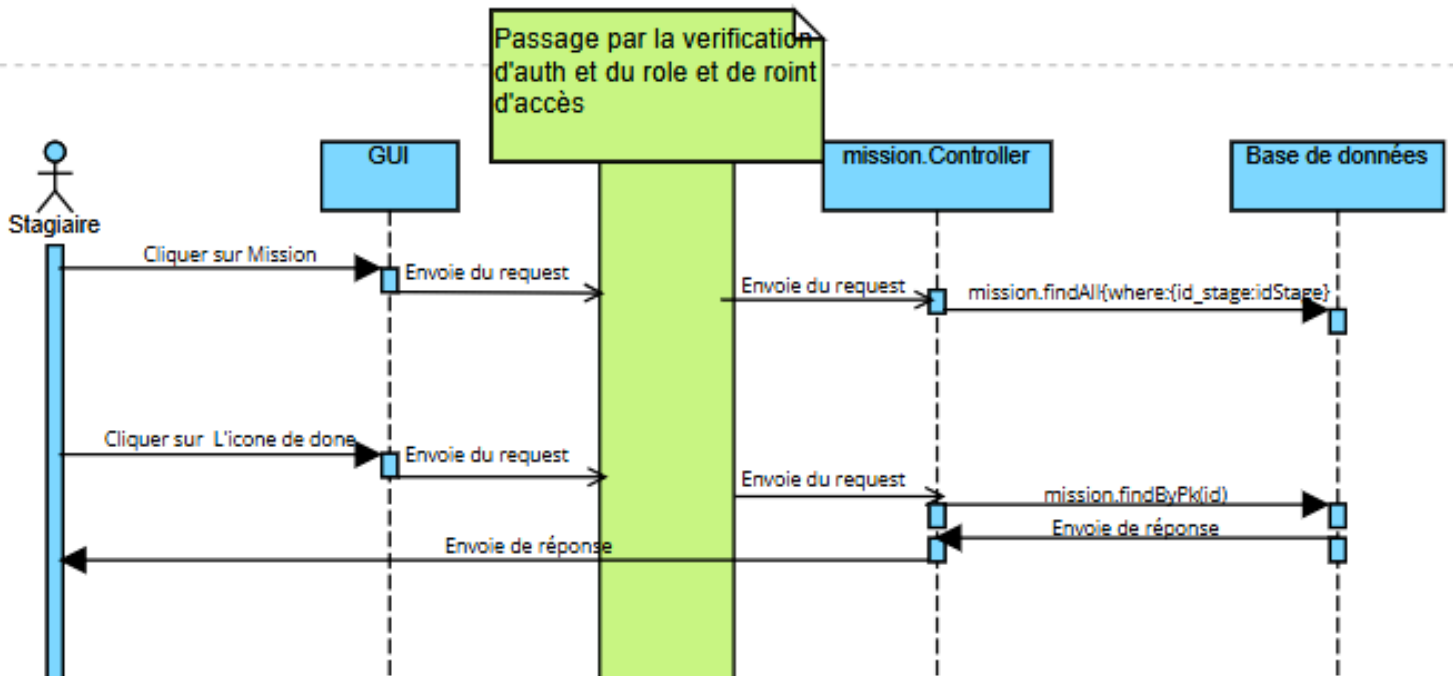


FIGURE II.15: Diagramme de séquence de processus de gestion des mission (coté Stagiaire)

Ce diagramme de séquence illustre le parcours d'un stagiaire dans une application de gestion de missions. Comme pour l'encadrant, chaque requête débute par une vérification de l'authentification et du rôle, ce qui assure que l'accès est sécurisé et que le stagiaire ne peut interagir qu'avec les fonctionnalités qui lui sont destinées. Le diagramme présente deux interactions principales : l'affichage des missions et la validation de leur statut.

### II.2.1.5 le Processus de CRUD d'une ticket (coté Stagiaire).

Le diagramme ci dessous illustre les séquences pour la gestion d'une Ticket dans un stage par le stagiaire.

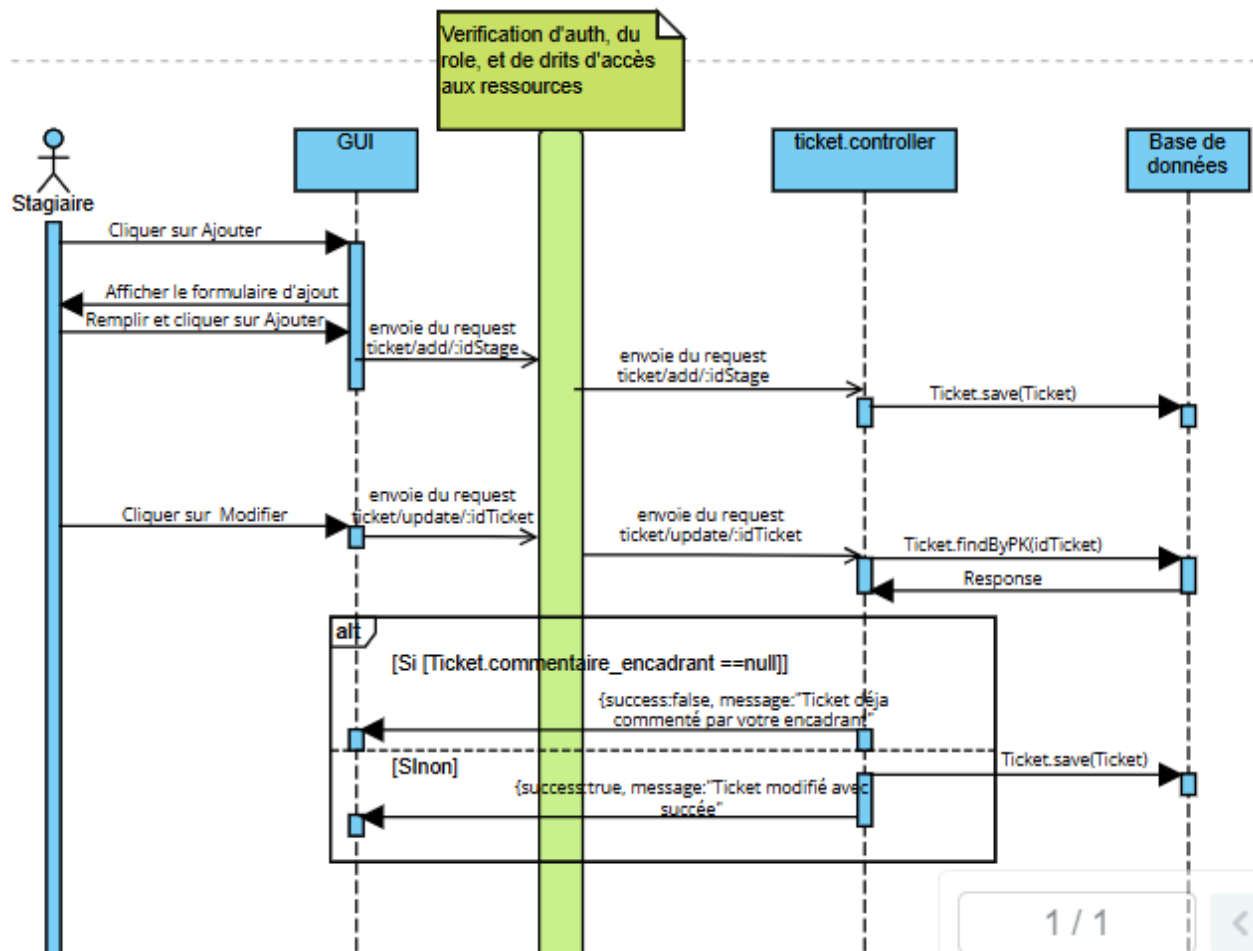


FIGURE II.16: Diagramme de séquence de processus de gestion des tickets (coté Stagiaire)

Ce diagramme de séquence décrit le flux d'interactions d'un stagiaire pour la gestion de ses "tickets", qui peuvent représenter des tâches ou des requêtes. Comme pour les missions, toute action débute par une vérification de l'authentification, du rôle de l'utilisateur et de possession des ressources, ce qui garantit un accès sécurisé.

### II.2.1.5 le Processus de CRUD d'une ticket (coté Encadrant).

Le diagramme ci dessous illustre les séquences pour la gestion d'une Ticket dans un stage par l'Encadrant.

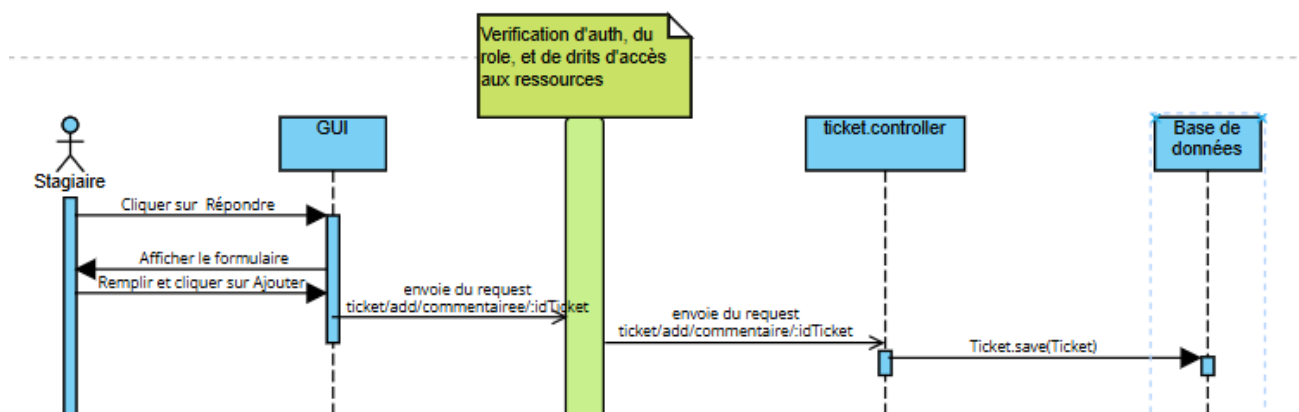


FIGURE II.17: Diagramme de séquence de processus de gestion des tickets (coté Encadrant)

Ce diagramme de séquence illustre le processus par lequel un encadrant répond à un ticket, en détaillant le flux de travail depuis l'interaction avec l'interface utilisateur jusqu'à l'enregistrement dans la base de données. Le processus est conçu pour être à la fois simple et sécurisé.

## II.2.2 Modèle de base de données.

### II.2.2.1 Structure de données.

La structure de données représente l'organisation logique et physique des informations manipulées par le système. Dans notre projet, nous avons opté pour une base de données relationnelle MySQL, afin de garantir la cohérence des données, la facilité de manipulation via SQL et la compatibilité avec l'ORM Sequelize côté backend Node.js.

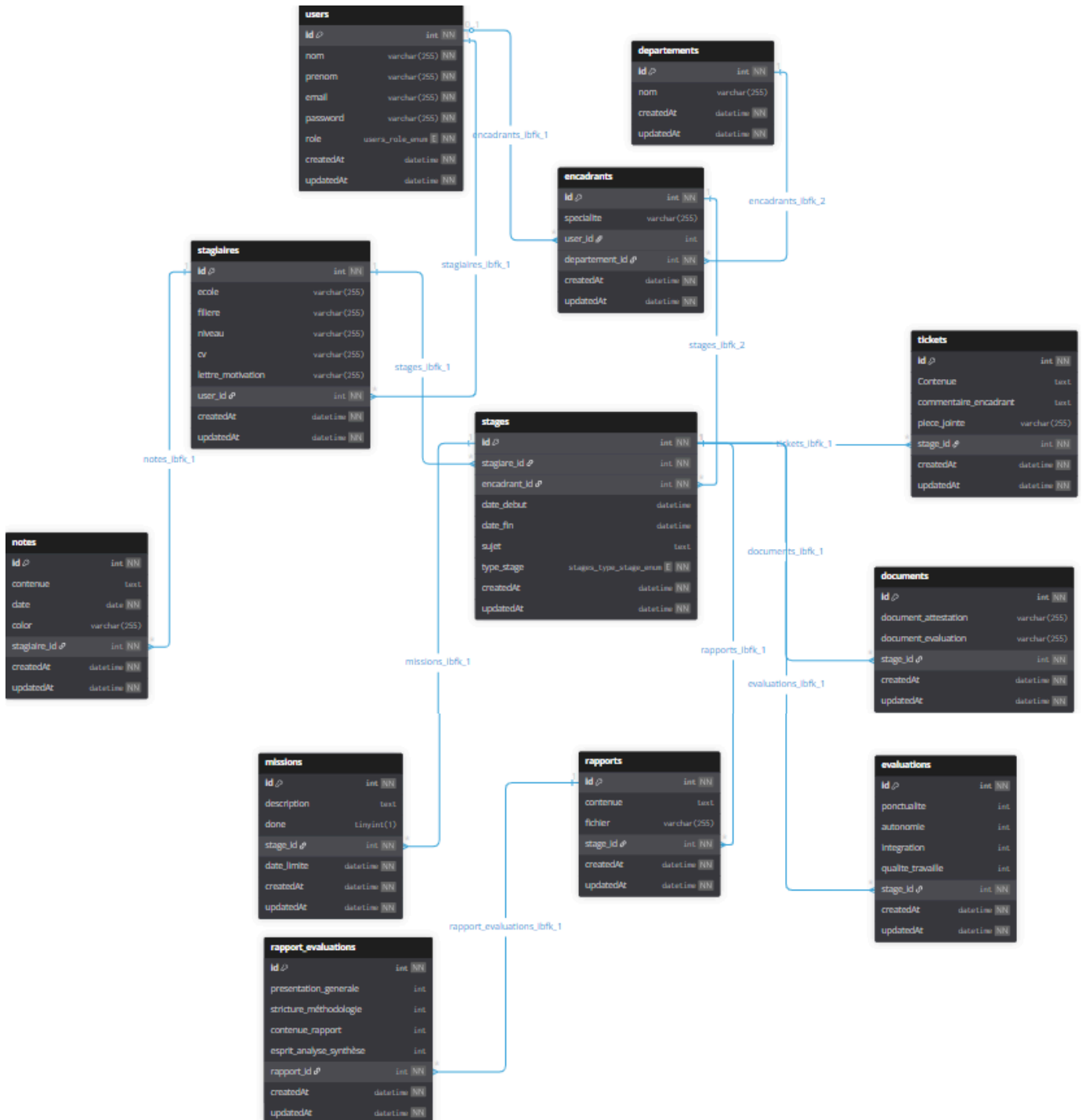


TABLE II.18: Modèle de base de données.

### Détail des Principales Entités et Leurs Relations :

#### 1. Gestion des Utilisateurs (users, stagiaires, encadrants, departements)

- **users** : Table centrale qui stocke les informations de base et les identifiants de connexion de tous les acteurs du système (admin, encadrant, stagiaire). Le champ **role** définit les permissions.
- **stagiaires** et **encadrants** : Ces tables étendent la table **users** via une clé étrangère (**user\_id**), stockant les informations spécifiques à chaque profil (école, filière, CV pour le stagiaire ; spécialité pour l'encadrant).
- **departements** : Table de référence qui structure l'organisation. Un encadrant appartient à un département (**departement\_id**).

#### 2. Gestion du Stage (stages)

- C'est la table pivot du système. Elle fait le lien entre un **stagiaire** (**stagiaire\_id**) et son **encadrant** (**encadrant\_id**).
- Elle contient les informations fondamentales du stage : dates de début et de fin, sujet et type (initiation, PFA, PFE).

#### 3. Suivi et Évaluation (evaluations, rapports, rapport\_evaluations, notes)

- **evaluations** : Permet à l'encadrant d'évaluer le stagiaire sur des critères précis (ponctualité, autonomie, etc.) pour un stage donné.
- **rapports** : Stocke le rapport de stage (texte et fichier) lié à un stage.
- **rapport\_evaluations** : Évalue spécifiquement la qualité du rapport sur des critères académiques (méthodologie, analyse, etc.). Elle est liée de manière unique à un rapport.
- **notes** : Fonctionnalité de prise de notes pour le stagiaire.

#### 4. Fonctionnalités Opérationnelles (missions, tickets)

- **missions** : Permet de définir des tâches ou missions pour le stagiaire avec une description, une date limite et un statut (**done**).
- **tickets** : Système de communication où le stagiaire peut poser une question (**Contenue**) et l'encadrant y répondre (**commentaire\_encadrant**). Une pièce jointe peut être ajoutée.

## 5. Gestion des Documents (documents)

- Centralise le stockage des chemins vers les documents générés pour un stage, comme l'attestation de stage et le document d'évaluation.

### Intégrité et Contraintes :

Le schéma utilise des contraintes d'intégrité référentielle (**FOREIGN KEY**) pour garantir la cohérence des données. Des relations **one-to-one** (ex: **stage** -> **documents**) sont assurées via des clés étrangères uniques (**UNIQUE**), tandis que des relations **one-to-many** (ex: **encadrant** -> **stages**) sont implémentées avec une simple clé étrangère. Les timestamps **createdAt** et **updatedAt** sur chaque table permettent un audit des données.

### **II.3 Conclusion**

Cette phase d'analyse et de conception a permis d'établir une base solide pour la mise en œuvre de la solution. Elle a débuté par une identification claire des besoins fonctionnels et non fonctionnels, ainsi qu'une modélisation précise des acteurs, des cas d'utilisation et des flux d'interaction entre les différents composants du système.

Les différents diagrammes UML produits, qu'il s'agisse des diagrammes de cas d'utilisation ou de séquence, ont permis de visualiser de manière structurée l'ensemble des processus métiers et des interactions techniques. Par ailleurs, le modèle de données a été soigneusement défini afin d'assurer la cohérence, la traçabilité et la flexibilité nécessaires à l'évolution future de la solution.





---

# RÉALISATION ET MISE EN OEUVRE

Dans cette partie, nous abordons le plan technique de mise en œuvre du projet. Nous commencerons par présenter les contraintes techniques associées aux outils et technologies utilisés durant la réalisation. Ensuite, l'architecture physique et applicative de la solution sera détaillée afin d'offrir une vision claire du processus d'implémentation.

### III.1 Introduction

La phase de réalisation constitue le passage de la conception à l'action. Elle englobe l'ensemble des développements techniques nécessaires pour transformer les spécifications fonctionnelles en une solution concrète. Dans ce chapitre, nous mettons en lumière les outils, les configurations et les développements personnalisés qui ont permis la mise en œuvre de l'application, tout en respectant les exigences métier définies dans les étapes précédentes.

### III.2. Outils utilisés

Frontend : React + Vite + TypeScript

- React : une bibliothèque JavaScript permettant de développer des interfaces utilisateur dynamiques et réactives. Elle favorise la réutilisation des composants et facilite la gestion de l'état de l'application.
- Vite : un outil de build moderne et rapide qui offre un temps de démarrage très court et un hot reload performant, idéal pour le développement en React.
- TypeScript : un sur-ensemble typé de JavaScript qui apporte la sécurité des types et une meilleure maintenabilité du code.

Backend : Node.js + Express.js

- Node.js : permet d'exécuter du JavaScript côté serveur. Elle est adaptée aux applications temps réel et aux API REST.
- Express.js : un framework minimaliste pour Node.js, qui facilite la création et la gestion des routes, des middlewares et des API REST sécurisées.

Base de données : MySQL

- MySQL : un système de gestion de base de données relationnel (SGBDR), qui permet de stocker et gérer les données des stagiaires, encadrants et administrateurs de manière structurée et efficace.

Sécurité et Authentification : JWT

- JWT (JSON Web Token) : utilisé pour gérer l'authentification des utilisateurs dans le système. Les tokens permettent de sécuriser les API et d'assurer que seuls les utilisateurs autorisés peuvent accéder aux fonctionnalités protégées.

#### Gestion de version et intégration continue : GitHub + GitHub Actions

- GitHub : héberge le code source et permet une gestion collaborative grâce au contrôle de version (Git).
- GitHub Actions : utilisé pour l'intégration continue et le déploiement automatique (CI/CD). Chaque push sur le dépôt déclenche la construction et la publication des images Docker.

#### Conteneurisation et orchestration : Docker + Docker Compose + Makefile

- Docker : permet d'exécuter le backend, le frontend dans des conteneurs isolés et portables.
- Docker Compose : simplifie la configuration et le lancement de plusieurs services en définissant l'architecture applicative dans un fichier docker-compose.yml.
- Makefile : facilite l'automatisation des commandes fréquentes (pull, build, run, des images Docker) via des raccourcis simples.

#### Autres outils:

- phpMyAdmin : pour la gestion graphique de la base MySQL.
- Postman : pour tester les API REST développées.
- Visual Studio Code : comme IDE principal de développement.
- VisualParadigme : pour la création des diagrammes UML.

### III.3 Architecture applicative

L'architecture applicative décrit la manière dont les modules logiciels et couches logicielles interagissent. Le projet suit une architecture en couches :

#### 1. Couche présentation (Frontend)

- Développée en React + TypeScript.
- Responsable de l'affichage de l'interface utilisateur (dashboard, formulaires de gestion des stagiaires, encadrants, etc.).
- Communique avec le backend via des appels API REST.

#### 2. Couche logique métier (Backend)

- Implémentée avec Node.js + Express.js.
- Contient les règles de gestion (enregistrement des stagiaires, gestion des encadrants, suivi des stages, etc.).
- Assure la sécurité via JWT pour l'authentification et les rôles (administrateur, stagiaire, encadrant).

#### 3. Couche données (Base de données)

- Gérée par MySQL.
- Stocke les informations relatives aux utilisateurs, stages, encadrants, documents, etc.
- L'accès aux données se fait via un ORM Sequelize (si tu l'utilises) ou via des requêtes SQL directes.

### **II.3 Conclusion**

La phase de réalisation et de mise en œuvre a permis de concrétiser l'ensemble des spécifications fonctionnelles et techniques définies lors des étapes préparatoires. À travers l'utilisation des technologies choisies, on a pu développer une application web complète et opérationnelle, intégrant la gestion des stagiaires, le suivi des assignments ainsi que le module de tickets de support.

---

## **les références bibliographiques**

Documentation:

- <https://developer.4d.com/docs/>
- <https://fr.4d.com/4d-professional-services>
- <https://docs.docker.com/compose/>