

---

## Résumé

Le présent rapport a été rédigé dans le cadre d'un projet de fin d'études en vue de l'obtention du Diplôme Nationale de Licence en Informatique de l'Institut Supérieur d'Informatique et de Mathématiques de Monastir. Ce projet est effectué au sein du startup Imotion+ dont l'objectif est de concevoir et développer une plateforme d'organisation de processus d'encadrement novatrice pour les encadrants, les étudiants et le gestionnaire de stage. Cette plateforme proposera une gamme d'outils avancés visant à faciliter la gestion de projet, le partage de fichiers, la validation des projets et le suivi régulier de l'avancement. L'objectif principal est d'assurer un suivi efficace et régulier des étudiants, favorisant ainsi leur progression.

**Mots clés : Processus encadrement, Gestion projet, React.js, Express.js, Node.js, MongoDB.**

## Abstract

This report has been written as part of a final year project to obtain the National Bachelor's Degree in Computer Science from the Higher Institute of Computer Science and Mathematics of Monastir. This project is being carried out within the startup Imotion+, which aims to design and develop an innovative supervision process organization platform for supervisors, students, and internship managers. This platform will offer a range of advanced tools to facilitate project management, file sharing, project validation, and regular progress tracking. The main objective is to ensure effective and regular monitoring of students, thereby promoting their progress.

**Keywords : Mentoring process, Project management, React.js, Express.js, Node.js, MongoDB**

# Dédicaces

*À mes très chers Parents **Zemzem Taher** et **Belkhiria Alia**, aucun hommage ne pourrait être à la hauteur de vos sacrifices, de l'amour et de l'affection dont vous n'avez jamais cessé de m'entourer toutes au long de ces années d'études. J'espère que vous trouvez dans ce travail un vrai témoignage de mon profond amour et éternelle reconnaissance.*

*À toute ma famille, mes frères et mes sœurs, **Mohamed**, **Moez**, **Sameh**, **Malak**, **Manel** et ma chère tante **Amel**, vous êtes mes compagnons de vie et mes sources d'inspiration. Votre encouragement constant et votre affection ont contribué à ma réussite. Je vous suis infiniment reconnaissante pour votre présence dans ma vie.*

*À tous mes chères amies **Amira**, **Nour**, **Fadoua** et mes amis palestiniens je vous remercie du fond du cœur pour votre amitié inestimable.*

*À tous ceux qui me sont chers, à vous tous, **Merci**.*

Jihen ZEMZEM

# Remerciements

Au terme de mon travail de fin d'étude, je tiens à adresser mes vifs remerciements et mes sincères gratitude à toutes les personnes qui ont contribué de près ou de loin à sa réussite.

Je tiens d'abord à remercier très chaleureusement **M. BEN SALEM Malek**, qui m'a permis de bénéficier de son encadrement. Les conseils qu'il m'a prodigués, sa patience et la confiance qu'il m'a témoignée ont été déterminants dans la réalisation de mon travail.

Je tiens aussi à adresser mes plus sincères remerciements à **M. KHATIB Yassin**, fondateur du startup Imotion+ pour m'offrir l'opportunité d'intégrer son équipe et pour son soutien.

Un très grand remerciement et une très grande reconnaissance sont destinés à tous mes enseignants pour la qualité de l'enseignement qui m'a été dispensé, ainsi qu'à mes camarades de promotion pour l'ambiance et la convivialité dans laquelle j'ai étudié durant ces trois années.

Enfin, je tiens à remercier également les membres du jury, qui me font un grand honneur en acceptant d'évaluer ce modeste travail. Leur expertise et leur évaluation jouent un rôle important dans ma progression académique.

# Table des matières

<b>Introduction générale</b>	<b>1</b>
<b>1 Présentation générale du projet</b>	<b>2</b>
1.1 Introduction . . . . .	3
1.2 Présentation de l'organisme d'accueil . . . . .	3
1.3 Contexte et problématique du projet . . . . .	3
1.3.1 Cadre du projet . . . . .	3
1.3.2 Problématique du projet . . . . .	3
1.4 Étude préalable . . . . .	4
1.4.1 Etude des solutions existantes . . . . .	4
1.4.2 Critique des solutions existantes . . . . .	7
1.5 Solution proposée . . . . .	8
1.6 Méthode de gestion du projet . . . . .	9
1.6.1 Choix de la méthode utilisée . . . . .	9
1.6.2 Gestion de projet avec la méthode 2TUP . . . . .	9
1.6.3 Planification du projet . . . . .	11
1.7 Conclusion . . . . .	11
<b>2 Présentation de la branche fonctionnelle et technique</b>	<b>12</b>
2.1 Introduction . . . . .	13
2.2 Spécification des besoins fonctionnels . . . . .	13
2.2.1 Identification des acteurs . . . . .	13
2.2.2 Identification des besoins fonctionnels par acteur . . . . .	13
2.2.2.1 Besoins fonctionnels pour « l'étudiant » . . . . .	13
2.2.2.2 Besoins fonctionnels pour « l'encadrant » . . . . .	14
2.2.2.3 Besoins fonctionnels pour « le gestionnaire de stage » . . . . .	15
2.2.2.4 Besoins fonctionnels pour « l'administrateur» . . . . .	15
2.2.3 Diagramme général des cas d'utilisation . . . . .	16
2.2.4 Diagramme de cas d'utilisation détaillé . . . . .	17
2.2.4.1 Diagramme de cas d'utilisation de l'acteur « Utilisateur» . . . . .	17

---

2.2.4.2	Diagramme de cas d'utilisation de l'acteur «Etudiant» . . . . .	17
2.2.4.3	Diagramme de cas d'utilisation de l'acteur «Encadrant» . . . . .	18
2.2.4.4	Diagramme de cas d'utilisation de l'acteur «Gestionnaire de stage»	18
2.2.4.5	Diagramme de cas d'utilisation de l'acteur «Administrateur» . .	19
2.3	Spécification des besoins non fonctionnels . . . . .	19
2.4	Spécification des besoins techniques . . . . .	20
2.4.1	Environnement de développement . . . . .	20
2.4.2	Environnement logiciel . . . . .	20
2.4.3	Architecture de l'application . . . . .	23
2.4.3.1	Architecture logique . . . . .	23
2.4.3.2	Architecture physique . . . . .	25
2.5	Conclusion . . . . .	26
<b>3</b>	<b>Conception</b>	<b>27</b>
3.1	Introduction . . . . .	28
3.2	Langage de modélisation utilisée . . . . .	28
3.3	Conception de la base de données . . . . .	28
3.3.1	Modèle conceptuel des données . . . . .	29
3.3.2	Modèle orientée document . . . . .	31
3.4	Conception détaillée . . . . .	32
3.4.1	Diagramme de classe . . . . .	32
3.4.2	Diagrammes de séquence objet . . . . .	33
3.4.2.1	Diagramme de séquence objet «S'inscrire» . . . . .	33
3.4.2.2	Diagramme de séquence objet «Ajouter une tâche » . . . . .	34
3.4.2.3	Diagramme de séquence objet «Ajouter un commentaire» . . . .	34
3.4.2.4	Diagramme de séquence objet «Valider un projet» . . . . .	35
3.4.3	Diagrammes d'activités . . . . .	36
3.4.3.1	Diagramme d'activité «Activer un utilisateur» . . . . .	36
3.4.3.2	Diagramme d'activité «Ajouter un commentaire» . . . . .	37
3.4.4	Diagramme de déploiement . . . . .	38
3.5	Conclusion . . . . .	39

<b>4 Réalisation</b>	<b>40</b>
4.1 Introduction . . . . .	41
4.2 Implémentation technique de la solution . . . . .	41
4.2.1 Espace utilisateur . . . . .	41
4.2.2 Espace administrateur . . . . .	42
4.2.3 Espace gestionnaire . . . . .	43
4.2.4 Espace Etudiant . . . . .	45
4.2.5 Espace Encadrant . . . . .	47
4.3 Conclusion . . . . .	49
<b>Conclusion générale</b>	<b>50</b>
<b>Bibliographie</b>	<b>51</b>

# Table des figures

1.1	Logo Imotion+ . . . . .	3
1.2	Interface d'accueil de moodle [1] . . . . .	4
1.3	Interface d'accueil de thinkific [2] . . . . .	5
1.4	Interface d'accueil de mentornity [3] . . . . .	6
1.5	Interface d'accueil de wrike [4] . . . . .	7
1.6	Cycle de vie de 2TUP [6] . . . . .	9
1.7	Diagramme de Gantt . . . . .	11
2.1	Diagramme général de cas d'utilisation . . . . .	16
2.2	Diagramme de cas d'utilisation associé à l'Utilisateur . . . . .	17
2.3	Diagramme de cas d'utilisation associé à l'Etudiant . . . . .	17
2.4	Diagramme de cas d'utilisation associé à l'Encadrant . . . . .	18
2.5	Diagramme de cas d'utilisation associé au gestionnaire de stage . . . . .	18
2.6	Diagramme de cas d'utilisation associé à l'administrateur . . . . .	19
2.7	Logo MongoDB [8] . . . . .	21
2.8	Logo Express.js [9] . . . . .	21
2.9	Logo React Js [10] . . . . .	21
2.10	Logo Node Js [11] . . . . .	22
2.11	Logo Vscode [12] . . . . .	22
2.12	Logo Draw.io [13] . . . . .	22
2.13	Logo Github [14] . . . . .	23
2.14	Logo Canva [15] . . . . .	23
2.15	Architecture MVC [16] . . . . .	24
2.16	Architecture 3 tiers avec MERN [17] . . . . .	25
3.1	Modèle conceptuel des données . . . . .	29
3.2	Modèle orienté document . . . . .	31
3.3	Diagramme de classe général . . . . .	32
3.4	Diagramme de séquence objet «S'inscrire» . . . . .	33
3.5	Diagramme de séquence objet «Ajouter une tâche» . . . . .	34

3.6	Diagramme de séquence objet «Ajouter un commentaire» . . . . .	35
3.7	Diagramme de séquence objet «Valider un projet » . . . . .	36
3.8	Diagramme d'activité «Activer un utilisateur» . . . . .	37
3.9	Diagramme d'activité «Ajouter un commentaire» . . . . .	37
3.10	Diagramme de déploiement . . . . .	38
4.1	Interface d'accueil . . . . .	41
4.2	Interface d'inscription . . . . .	42
4.3	Interface «Gérer les utilisateurs» . . . . .	43
4.4	Interface «Gérer les projets» . . . . .	43
4.5	Message «Confirmer validation» . . . . .	44
4.6	Message «Succès de validation» . . . . .	44
4.7	Interface «Suivre progrès» pour le «Gestionnaire» . . . . .	45
4.8	Interface «Gérer les tâches» . . . . .	46
4.9	Interface «Ajouter une tâche» . . . . .	46
4.10	Interface «Details d'une tâche» . . . . .	47
4.11	Interface «Consulter projets» pour l'«encadrant» . . . . .	47
4.12	Interface «Suivre progrès » pour l'«encadrant» . . . . .	48
4.13	Interface «Ajouter un commentaire» . . . . .	48

# Liste des tableaux

1.1 Comparaison entre les solutions existantes . . . . .	8
3.1 Les entités du modèle conceptuel de données . . . . .	30

# Liste des abréviations

- **2TUP** = **2** Track Unified Process
- **API** = Application Programming Interface
- **BPMN** = Business Process **Model** and Notation
- **DTO** = Data Transfer Object
- **HTML** = HyperText Markup Language
- **JSON** = JavaScript Object Notation
- **JWT** = JSON Web Tokens
- **MCD** = Modèle Conceptuel de Données
- **MERN** = MongoDB, Express, React, Node
- **MVC** = Model-View-Controller
- **REST** = Representational State Transfer
- **SPA** = Single Page Applications
- **SysML** = Systems Modeling Language
- **UML** = Unified Modeling Language

# Introduction générale

Le processus d'encadrement des projets de fin d'études représente souvent un défi pour les encadrants, les étudiants et le gestionnaire de stage, pouvant causer de la souffrance et de la frustration pour les trois parties impliquées. Les étudiants peuvent se sentir perdus, stressés et sous pression s'ils ne bénéficient pas d'un encadrement clair et bien structuré. Les encadrants, quant à eux, doivent jongler entre les demandes des étudiants, la garantie de la qualité des travaux et le respect des délais, ce qui peut être une tâche ardue. De même, le gestionnaire de stage peut souffrir de la non-récupération des fichiers qu'il envoie par mail tels que les lettres d'appui ou les fiches d'information, ce qui peut entraîner des retards ou des problèmes dans le processus de validation du projet ou du stage.

C'est pour remédier à ces problèmes que nous avons créée une plateforme d'organisation de processus d'encadrement en ligne qui vise à simplifier et à fluidifier la supervision des travaux de fin d'études. Les étudiants peuvent utiliser cette plateforme pour gérer les tâches de leur projet, tandis que les encadrants peuvent suivre l'avancement des travaux et évaluer les résultats tout à partir d'une seule plateforme. De plus, le gestionnaire de stage peut déposer les fichiers nécessaires pour la confirmation de projet. Ainsi, il joue un rôle crucial en validant les stages, assurant leur conformité aux exigences établies, et en déposant les dates des soutenances pour permettre aux étudiants de présenter leurs travaux de fin d'études devant un jury. Cette approche peut aider à soulager la pression ressentie par les stagiaires en leur fournissant un encadrement clair et structuré, tout en aidant les superviseurs à répondre aux besoins individuels de chaque étudiant de manière plus efficace et en permettant au gestionnaire de fournir les documents nécessaires pour la confirmation de projet et de stage.

Le rapport de stage suit une structure organisée pour présenter le travail de manière claire et précise. Le premier chapitre donne une présentation générale de projet, tandis que le deuxième chapitre se concentre sur l'étude des besoins fonctionnels, non fonctionnels et techniques. Le troisième chapitre aborde la conception de notre plateforme, tandis que le quatrième chapitre présente les interfaces de la solution développée. La conclusion récapitule les principales réalisations du projet. Cette structure bien organisée facilite la présentation du travail effectué durant la période de stage.

# PRÉSENTATION GÉNÉRALE DU PROJET

## Plan

1	Introduction	3
2	Présentation de l'organisme d'accueil	3
3	Contexte et problématique du projet	3
4	Étude préalable	4
5	Solution proposée	8
6	Méthode de gestion du projet	9
7	Conclusion	11

## 1.1 Introduction

Ce chapitre fournit un aperçu du contexte de notre projet. Nous présentons d'abord l'organisme d'accueil, puis nous abordons la problématique et l'étude de l'existant. Ensuite, nous présentons la solution proposée. Enfin, nous décrivons la méthodologie de travail adoptée et la planification du projet.

## 1.2 Présentation de l'organisme d'accueil

Imotion+ est une startup créée en 2019 et née de la fusion de deux métiers, conseil en marketing, communication, développement et production audiovisuelle, elle propose d'associer ses compétences pour répondre mieux aux différentes demandes de ses clients, pouvant ainsi offrir une double expertise dans les domaines de développement et de la production graphique et audiovisuelle. Elle accompagne les entreprises, les associations, les fondations, les entreprises sociales, les ONG internationales et les initiateurs de bonnes causes.

La figure 1.1 présente le logo de l'organisme d'accueil Imotion+.



FIGURE 1.1 : Logo Imotion+

## 1.3 Contexte et problématique du projet

### 1.3.1 Cadre du projet

Le présent travail s'inscrit dans le cadre du projet de fin d'études en vue de l'obtention du Diplôme National de Licence en Informatique de l'Institut Supérieur d'Informatique et de Mathématiques de Monastir (ISIMM).

### 1.3.2 Problématique du projet

La gestion des processus d'encadrement peut être une tâche complexe pour les étudiants, les encadrants et le gestionnaire de stage avec des défis tels que la gestion du projet de fin d'étude, le suivi des progrès des stagiaires, la récupération des documents nécessaires à la validation du

projet de fin d'étude et la planification des soutenances. Les solutions existantes, offrent un certain accompagnement, mais peuvent ne pas répondre aux besoins spécifiques des utilisateurs cibles ou garantir la sécurité et la confidentialité des données. Par conséquent, la problématique de ce projet de fin d'études est de trouver comment concevoir une plateforme d'organisation de processus d'encadrement qui réponde aux besoins spécifiques des superviseurs, des stagiaires et gestionnaires, tout en garantissant la sécurité et la confidentialité des données, pour permettre une gestion optimale des processus d'encadrement.

## 1.4 Étude préalable

Avant de proposer une solution, il est important d'examiner les applications existantes pour identifier les limitations préoccupantes et identifier les vulnérabilités de ses solutions.

### 1.4.1 Etude des solutions existantes

**Moodle** : est une plateforme d'apprentissage open source. Elle permet aux éducateurs de tout type de créer un espace privé en ligne, rempli d'outils permettant de créer facilement des cours et des activités, tous optimisés pour l'apprentissage collaboratif.

La figure 1.2 présente l'interface d'accueil de l'outil moodle.



**FIGURE 1.2 :** Interface d'accueil de moodle [1]

### Les points forts

- Open source et donc gratuit

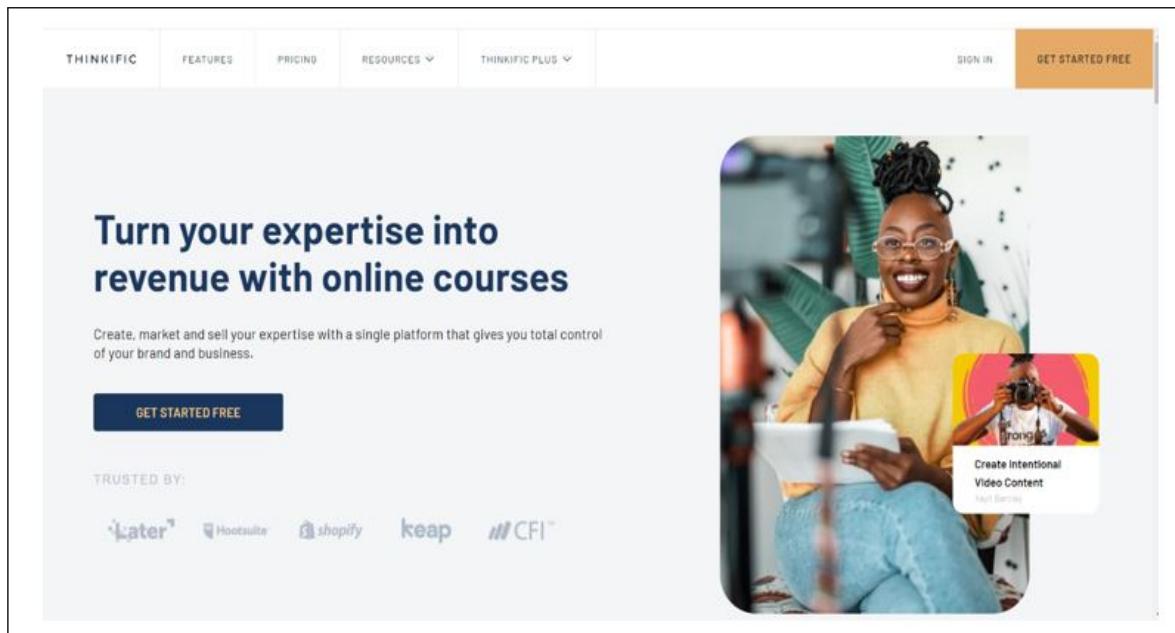
- Fonctionnalités avancées
- Gestion de projets avec un tableau Kanban

### Les points faibles

- Interface utilisateur semble complexe pour les débutants
- Configuration de la plate-forme peut être difficile

**Thinkific** : est une plateforme de formation en ligne complète permettant à tous les utilisateurs de la plateforme, qu'ils soient enseignants, formateurs ou experts dans leur domaine de créer et dispenser rapidement des formations de qualité supérieure aux étudiants, avec des outils de suivi et de supervision intégrés.

La figure 1.3 présente l'interface d'accueil de l'outil Thinkific.



**FIGURE 1.3 :** Interface d'accueil de thinkific [2]

### Les points forts

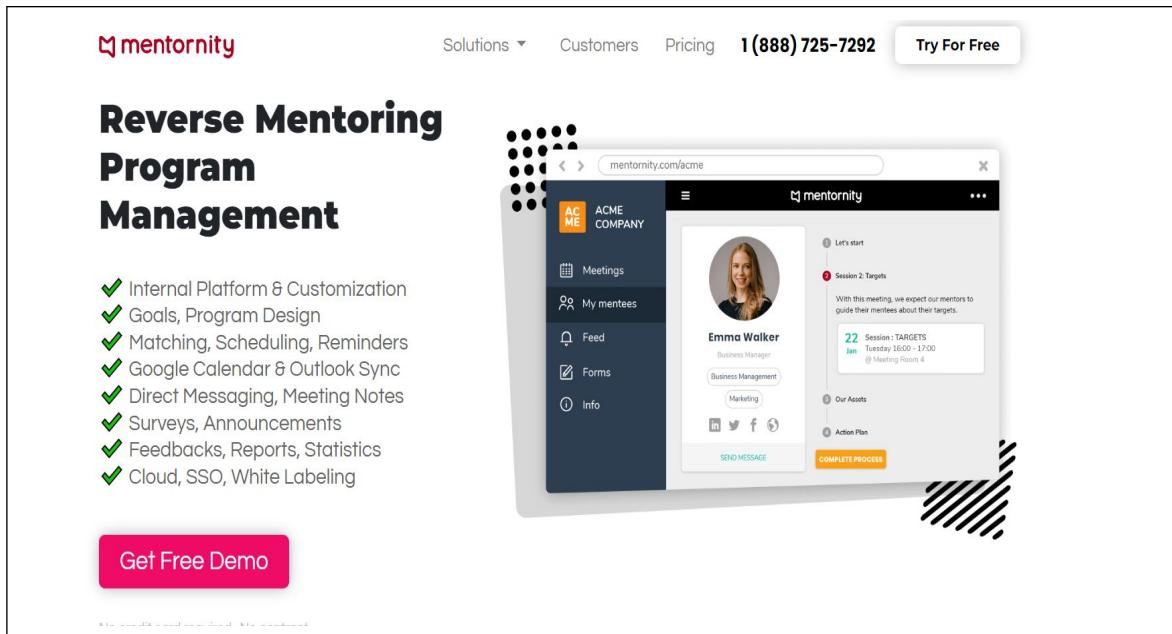
- Interface utilisateur intuitive et facile à utiliser
- Fonctionnalités de mentorat et de coaching
- Suivi des apprenants et de commentaires

### Les points faibles

- La version gratuite est très limitée en termes de fonctionnalités
- Pas de fonctionnalité de suivi de projet
- Pas de fonctionnalité de chat en direct avec les étudiants

**Mentornity** : est une plateforme interne pour les programmes de mentorat, de coaching et de développement organisationnel qui permet aux enseignants et étudiants de se connecter pour des sessions de mentorat en ligne.

La figure 1.4 présente l'interface d'accueil de l'outil Mentornity.



**FIGURE 1.4 :** Interface d'accueil de mentornity [3]

### Les points forts

- Interface utilisateur intuitive et facile à utiliser
- Possibilité de suivre l'avancement des apprenants et de leur fournir des commentaires
- Fonctionnalité de messagerie intégrée

### Les points faibles

- Moins de fonctionnalités avancées
- Coût élevé par rapport à certaines autres solutions

**Wrike** : Wrike est une plateforme de gestion du travail et de collaboration en ligne. Elle permet aux équipes de planifier, suivre et gérer leurs projets et leurs tâches de manière efficace. L'objectif principal de Wrike est d'améliorer la productivité de l'équipe en centralisant les informations, en facilitant la collaboration et en fournissant une visibilité complète sur les projets en cours.

La figure 1.5 présente l'interface d'accueil de l'outil Wrike.

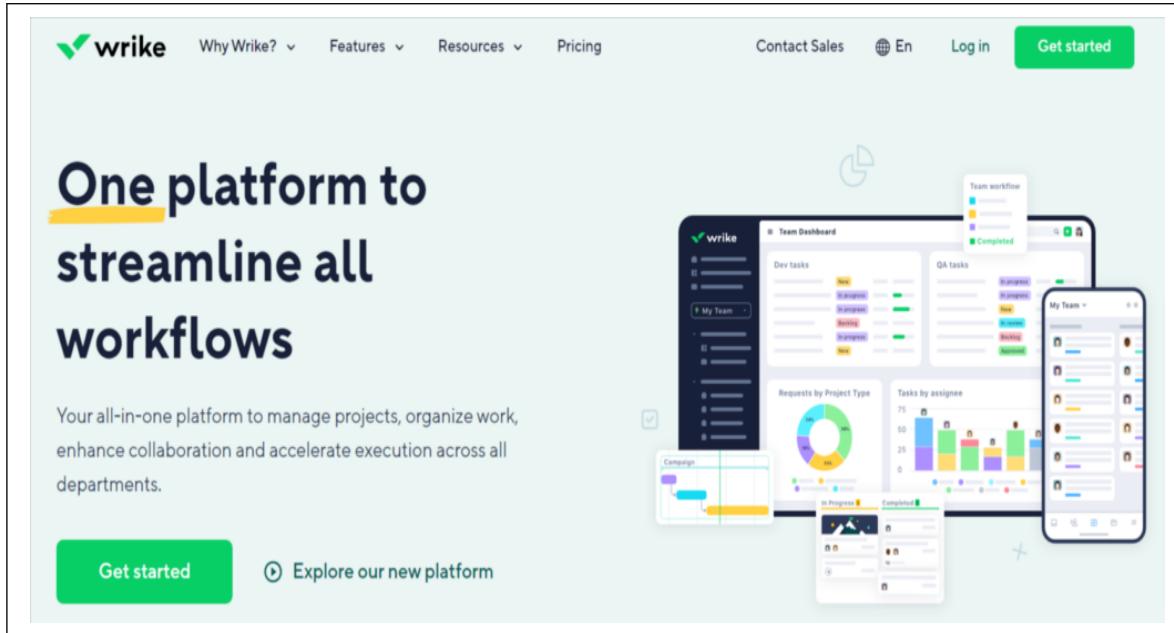


FIGURE 1.5 : Interface d'accueil de wrike [4]

### Les points forts

- Gestion complète du cycle de vie des projets
- Flexibilité dans l'organisation
- Collaboration en temps réel

### Les points faibles

- Courbe d'apprentissage initiale pour les utilisateurs. En raison de la richesse des fonctionnalités offertes par Wrike, il peut être nécessaire de consacrer du temps et des efforts pour apprendre à utiliser pleinement l'outil.
- Coût élevé par rapport à certaines autres solutions

#### 1.4.2 Critique des solutions existantes

Nous avons identifié plusieurs critères essentiels à prendre en compte lors de la comparaison et de l'évaluation des applications, comme indiqué dans le tableau comparatif 1.1.

- **Facilité d'utilisation :** Ce critère évalue la convivialité de l'interface utilisateur et la facilité de navigation pour les utilisateurs.
- **Attractivité :** Ce critère évalue l'aspect visuel et esthétique de chaque plateforme.
- **Performance :** Ce critère mesure la vitesse, la stabilité et la réactivité de chaque plateforme.
- **Fonctionnalités :** Ce critère évalue la présence des fonctionnalités essentielles pour le processus d'encadrement, telles que la validation et la gestion des projets, la gestion des

tâches, le suivi et l'évaluation d'avancement, etc.

- **Sécurité** : Ce critère évalue le niveau de protection des données et des informations des utilisateurs.
- **Coût** : Ce critère évalue les coûts associés à l'utilisation de l'application.

**TABLEAU 1.1** : Comparaison entre les solutions existantes

Critères	Moodle	Thinkific	Mentornity	Wrike
<b>Facilité d'utilisation</b>	Complexe	Simple	Simple	Complexe
<b>Attractivité</b>	Moyenne	Élevée	Élevée	Élevée
<b>Performance</b>	Moyenne	Moyenne	Moyenne	Moyenne
<b>Fonctionnalités</b>	Complète	Limitée	Limitée	Complète
<b>Sécurité</b>	Moyenne	Moyenne	Moyenne	Moyenne
<b>Coût</b>	Faible	Moyen à Élevé	Moyen à Élevé	Élevée

## 1.5 Solution proposée

Après une analyse approfondie des plateformes existantes, nous avons constaté que la plupart d'entre elles étaient initialement développées pour l'apprentissage et la formation en général. Cependant, nous proposons de concevoir et de mettre en œuvre une plateforme appelée "ENCADRA" conçue pour offrir un encadrement optimal des étudiants tout au long de leur stage de fin d'études.

Notre solution doit être en mesure de satisfaire les exigences suivantes :

- **Une interface facile à utiliser et conviviale** pour permettre aux utilisateurs de naviguer facilement dans l'application et d'accomplir leurs tâches de manière efficace.
- **Une gestion de projets simplifiée** facilitant leur validation, la gestion des tâches, le dépôt des fichiers nécessaires et la planification des soutenances.
- **Un suivi et une surveillance avancée des étudiants** permettant de contrôler la progression des étudiants, d'évaluer leurs avancements, d'ajouter des commentaires et de visualiser clairement leur progression à l'aide de graphiques et d'un tableau de bord.
- **Une Importance primordiale accordée à la sécurité des données** afin de garantir la confidentialité et l'intégrité des informations partagées sur la plateforme.

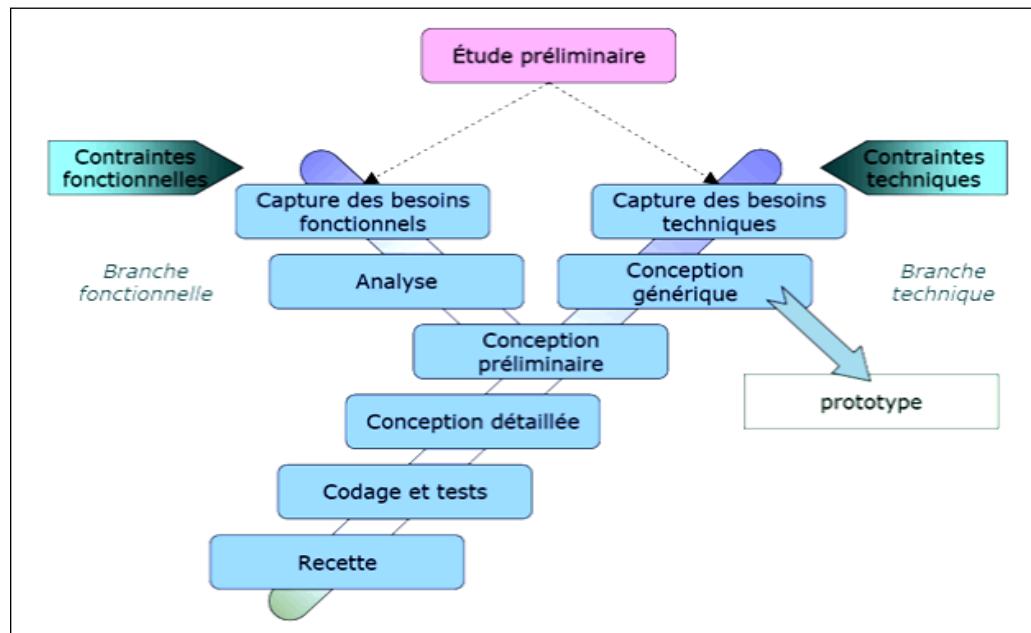
## 1.6 Méthode de gestion du projet

### 1.6.1 Choix de la méthode utilisée

Pour la création de tout logiciel, l'utilisation d'une méthode de travail est essentielle. Il existe diverses méthodes de développement ainsi que plusieurs types de cycles de développement qui jouent un rôle important dans la conception d'un logiciel. Ces cycles prennent en compte toutes les étapes nécessaires à la création du logiciel.

En se référant sur des mémoires de projet de fin d'étude [5] déjà réalisé au sein de l'institut et selon nos besoins, notre choix est orienté vers le processus unifié en particulier la méthode 2TUP, du fait de son approche nouvelle et originale et essentiellement de la séparation entre les besoins techniques et fonctionnels. Notre projet est basé sur un processus de développement bien défini qui va de la détermination des besoins fonctionnels attendus du système jusqu'à la conception et le codage final. Cette méthode ne se base aucunement sur un processus linéaire mais bien, sur un développement itératif et incrémental.

La figure 1.6 illustre les différentes phases de développement.



**FIGURE 1.6 :** Cycle de vie de 2TUP [6]

### 1.6.2 Gestion de projet avec la méthode 2TUP

Le 2TUP est un processus de développement logiciel qui se base sur le Processus Unifié. Il met en place un cycle de développement en Y, qui sépare les aspects techniques des aspects fonctionnels. Le processus de 2TUP démarre par une étude préliminaire dont l'objectif principal

est d'identifier les parties prenantes impliquées dans le système à construire, les échanges de messages entre elles et le système, la création du cahier des charges, ainsi que la modélisation du contexte [7]. Le processus se divise ensuite en trois phases essentielles :

La branche de gauche (branche fonctionnelle) contient deux phases :

- **Capture des besoins fonctionnels** : Capitalise la connaissance du métier de l'entreprise. Cette branche capture des besoins fonctionnels, ce qui produit un modèle des besoins focalisé sur le métier des utilisateurs. Elle qualifie, au plus tôt le risque de produire un système inadapté aux utilisateurs.
- **L'analyse** : Consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en termes de métier.

La branche de droite (branche technique) intègre deux phases :

- **Capture des besoins techniques** : Recense toutes les contraintes sur les choix de dimensionnant et la conception du système. Il s'agit de la spécification des outils, de la structure des matériels à exploiter ainsi que la prise en compte des contraintes d'intégration avec l'existant.
- **La conception générique** : Définit ensuite les composants nécessaires à la construction de l'architecture technique. Cette conception est complètement indépendante des aspects fonctionnels spécifiés dans la branche gauche. Elle a pour objectif de d'uniformiser et de réutiliser les mêmes mécanismes pour tout un système. L'architecture technique construit le squelette du système, son importance est telle qu'il est conseillé de réaliser un prototype de manière à valider les principes par le codage et les tests.

La branche du milieu se compose de quatre phases :

- **La conception préliminaire** : Consiste à appliquer les concepts liés aux fonctionnalités du système et à intégrer les composants techniques au système. Il s'agit d'intégrer les fonctions métiers et applicatives dans l'architecture technique définie dans la phase de conception générique.
- **La conception détaillée** : Dans cette étape, nous examinons de près la réalisation de chaque composant spécifique du système. Nous fournissons une description détaillée des spécifications techniques de chaque élément, en expliquant précisément comment il sera implémenté, son fonctionnement et ses interactions avec les autres composants. Nous utilisons des éléments tels que le schéma de base de données, les diagrammes de classes et les diagrammes de séquence pour visualiser ces informations.

- **L'étape de codage** : Cette étape consiste à traduire les spécifications techniques et les conceptions réalisées précédemment en un code informatique fonctionnel.
- **L'étape de recette** : Elle comprend la définition d'une stratégie de validation et le test de l'application dans un environnement d'exécution approprié.

### 1.6.3 Planification du projet

La planification est un élément essentiel de la gestion de projet, car elle permet de définir les travaux à réaliser, d'établir des objectifs, de coordonner les actions, de gérer les ressources, de réduire les risques, de suivre les progrès en cours et de rendre compte de l'état d'avancement du projet. En somme, la planification est un outil indispensable pour le management efficace d'un projet.

La Figure 1.7 présente le diagramme de GANTT de notre projet.

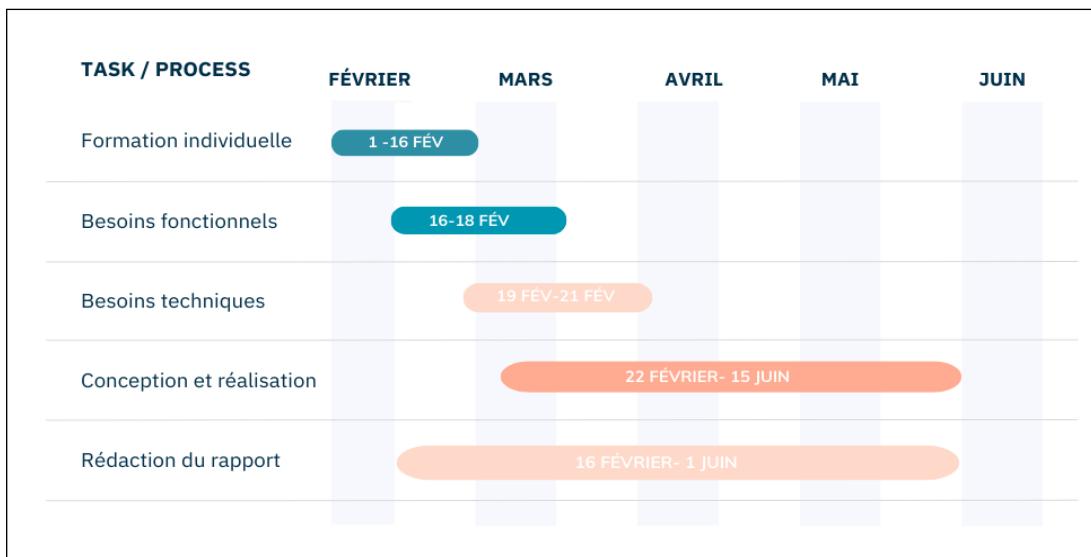


FIGURE 1.7 : Diagramme de Gantt

## 1.7 Conclusion

Dans ce chapitre, nous avons abordé le contexte global de notre projet et défini son objectif. Nous avons également effectué une analyse approfondie de solutions existantes et choisi la méthodologie de gestion de projet à adopter. Le suivant chapitre se concentre sur la présentation de la partie fonctionnelle et technique du projet.

# PRÉSENTATION DE LA BRANCHE FONCTIONNELLE ET TECHNIQUE

---

## Plan

1	Introduction . . . . .	13
2	Spécification des besoins fonctionnels . . . . .	13
3	Spécification des besoins non fonctionnels . . . . .	19
4	Spécification des besoins techniques . . . . .	20
5	Conclusion . . . . .	26

## 2.1 Introduction

Le présent chapitre se concentre sur l'analyse de la branche gauche et la branche droite à l'aide de la méthodologie 2TUP. Cette approche nous permet d'identifier les acteurs impliqués, recueillir les besoins fonctionnels, et comprendre les exigences spécifiées par les besoins non fonctionnels. L'objectif est de déterminer toutes les fonctionnalités nécessaires pour le système à venir, ainsi que d'expliquer les besoins techniques en détaillant l'architecture logique et physique utilisée.

## 2.2 Spécification des besoins fonctionnels

### 2.2.1 Identification des acteurs

Tout système interactif doit permettre de simplifier l'interaction entre les utilisateurs qui agissent en tant qu'acteurs et le système lui-même. Les acteurs sont des entités externes qui utilisent le système via ses différentes interfaces. L'application regroupe quatre principaux acteurs avec des fonctionnalités pour chacun d'entre eux :

- **Étudiant** : Cette personne est le stagiaire effectuant un stage dans le cadre de son projet de fin d'étude et bénéficiaire de l'encadrement dispensé par l'enseignant.
- **Encadrant** : C'est l'enseignant qui est chargée de la supervision et de l'encadrement des étudiants tout au long de la période de stage.
- **Gestionnaire de stage** : Cette personne est responsable de dépôt des fichiers nécessaires pour les étudiants et la validation de leurs projets.
- **Administrateur de la plateforme** : Cette personne est responsable de la gestion des utilisateurs de la plateforme et de la garantie de son bon fonctionnement.

### 2.2.2 Identification des besoins fonctionnels par acteur

Dans cette partie, nous allons exposer les besoins fonctionnels de notre application en étroite relation avec les acteurs précédemment mentionnés.

#### 2.2.2.1 Besoins fonctionnels pour « l'étudiant »

- **S'inscrire à la plateforme** : L'étudiant peut s'inscrire en fournissant ses informations personnelles telles que son nom, son adresse e-mail et son mot de passe.
- **S'authentifier pour accéder à son espace** : Après l'inscription et l'autorisation de

l'admin, l'étudiant peut se connecter à la plateforme en utilisant ses identifiants (adresse e-mail et mot de passe) pour accéder à son espace personnel.

- **Gérer son profil :** L'étudiant peut modifier et mettre à jour son profil, y compris des informations personnelles telles que son nom, ses coordonnées, sa photo, etc.
- **Gérer ses tâches :** L'étudiant peut consulter et gérer ses tâches assignées dans un tableau de bord kanban pour une meilleure visibilité. Cela peut inclure la création de nouvelles tâches, la modification et la suppression des tâches existantes.
- **Consulter les fichiers diffusés :** L'étudiant peut accéder aux fichiers diffusés par le gestionnaire de stage. Ces fichiers peuvent contenir des instructions relatives au stage, des documents administratifs, et d'autres informations pertinentes liées au stage de l'étudiant.
- **Envoyer des fichiers :** L'étudiant peut envoyer des fichiers au gestionnaire de stage.
- **Consulter les dates de dépôt des rapports :** L'étudiant peut consulter les dates de dépôt des rapports. Cela lui permet de connaître les délais pour soumettre ses travaux.
- **Consulter les dates des soutenances :** L'étudiant peut accéder aux dates des soutenances de mémoires. Cela lui permet de prévoir et de se préparer en conséquence pour sa présentation.

#### 2.2.2.2 Besoins fonctionnels pour « l'encadrant »

- **S'inscrire à la plateforme :** L'encadrant peut s'inscrire en fournissant ses informations personnelles telles que son nom, son adresse e-mail et son mot de passe.
- **S'authentifier pour accéder à son espace :** Après l'inscription et l'autorisation de l'admin, l'encadrant peut se connecter à la plateforme en utilisant ses identifiants (adresse e-mail et mot de passe) pour accéder à son espace personnel.
- **Gérer son profil :** L'encadrant peut modifier et mettre à jour son profil, y compris des informations telles que son nom, ses coordonnées, sa photo, etc.
- **Consulter ses projets :** L'encadrant consulte la liste des projets qui les encadrent. Cette liste lui permet d'accéder au tableau de bord kanban des tâches relatives à chaque projet, il peut suivre leur progression avec la possibilité d'ajouter un commentaire.
- **Consulter les dates de dépôt des rapports :** L'encadrant peut consulter les dates de dépôt des rapports.
- **Consulter les dates des soutenances :** L'encadrant peut accéder aux dates des soutenances de mémoires.

### 2.2.2.3 Besoins fonctionnels pour « le gestionnaire de stage »

- **S'inscrire à la plateforme :** Le gestionnaire peut s'inscrire en fournissant ses informations personnelles telles que son nom, son adresse e-mail et son mot de passe.
- **S'authentifier pour accéder à son espace :** Après l'inscription et l'autorisation de l'admin, le gestionnaire peut se connecter à la plateforme en utilisant ses identifiants (adresse e-mail et mot de passe) pour accéder à son espace personnel.
- **Gérer son profil :** Le gestionnaire peut modifier et mettre à jour son profil, y compris des informations telles que son nom, ses coordonnées, sa photo, etc.
- **Gérer les projets :** Le gestionnaire est responsable pour l'ajout et la validation des projets . Il peut également suivre l'avancement de chaque projet.
- **Envoyer des fichiers :** Le gestionnaire est responsable pour la diffusion des fichiers qui peuvent contenir des instructions relatives au stage, des documents administratifs, et d'autres informations pertinentes liées au stage de l'étudiant.
- **Publier les dates de dépôt des rapports :** Le gestionnaire de stage est chargé de publier les dates limites de dépôt des rapports de stage pour les étudiants.
- **Publier les dates des soutenances :** De même, le gestionnaire de stage est responsable de la publication des dates des soutenances de stage. Ces dates indiquent les plages horaires auxquelles les étudiants devront présenter et défendre leurs travaux devant un jury.

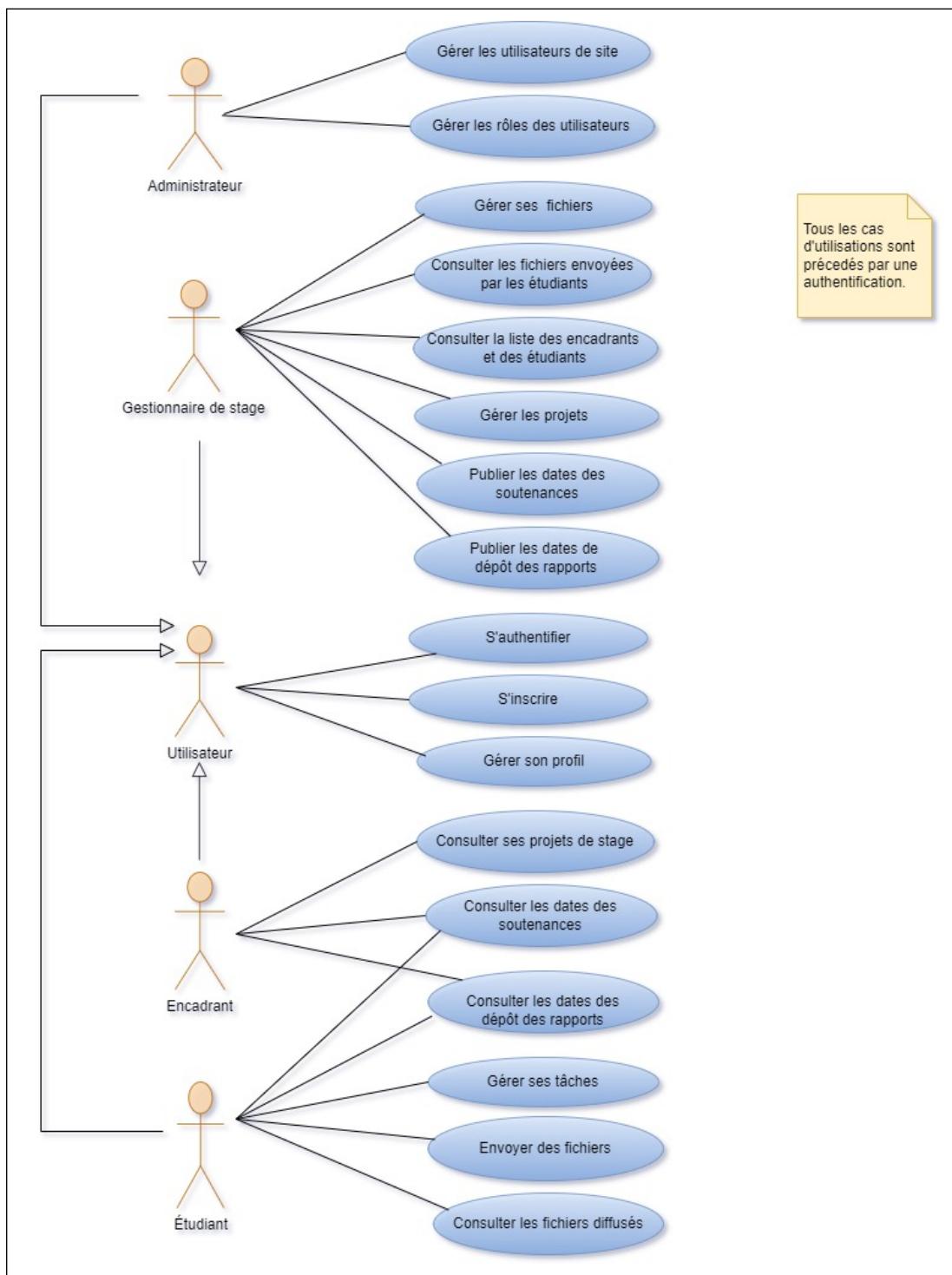
### 2.2.2.4 Besoins fonctionnels pour « l'administrateur»

- **S'authentifier pour accéder à son espace :** l'admin peut se connecter à la plateforme en utilisant ses identifiants pour accéder à son espace personnel.
- **Gérer son profil :** L'admin peut modifier et mettre à jour son profil, y compris des informations telles que son nom, ses coordonnées, sa photo, etc.
- **Gérer les utilisateurs :** l'administrateur dispose de la capacité d'activer et de désactiver les comptes des utilisateurs sur la plateforme.
- **Gérer les rôles :** L'administrateur peut gérer les rôles des utilisateurs sur la plateforme. Cela inclut la possibilité d'attribuer des rôles spécifiques à chaque utilisateur en fonction de leurs responsabilités et de leurs droits d'accès.

### 2.2.3 Diagramme général des cas d'utilisation

Le diagramme de cas d'utilisation permet de visualiser les fonctionnalités offertes par le système du point de vue des utilisateurs, en identifiant les différentes actions qu'ils peuvent effectuer et les résultats obtenus.

La Figure 2.1 illustre le diagramme global de notre projet.



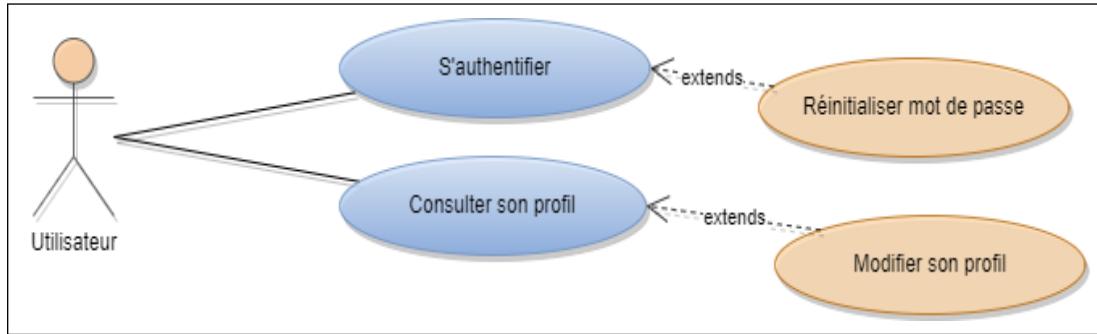
**FIGURE 2.1 :** Diagramme général de cas d'utilisation

#### 2.2.4 Diagramme de cas d'utilisation détaillé

Pour simplifier le diagramme de cas d'utilisation, nous avons procédé à sa décomposition en cinq diagrammes distincts.

##### 2.2.4.1 Diagramme de cas d'utilisation de l'acteur «Utilisateur»

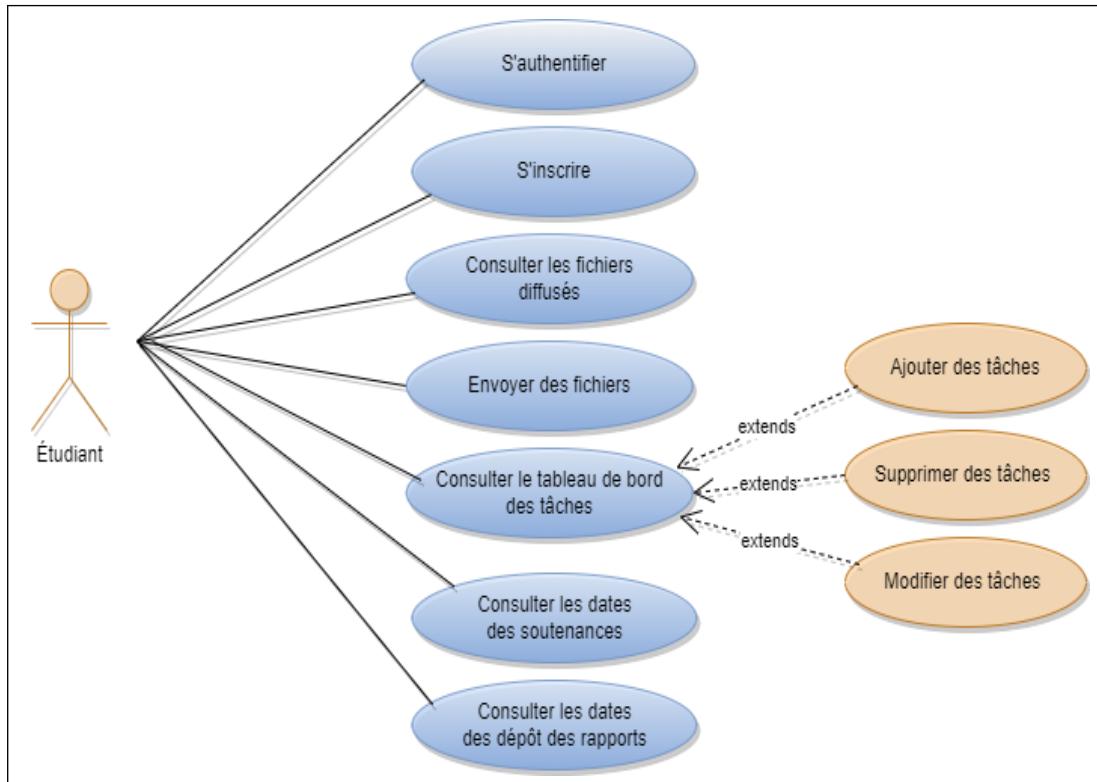
La Figure 2.2 présente le diagramme des cas d'utilisation détaillé pour l'utilisateur.



**FIGURE 2.2 :** Diagramme de cas d'utilisation associé à l'Utilisateur

##### 2.2.4.2 Diagramme de cas d'utilisation de l'acteur «Etudiant»

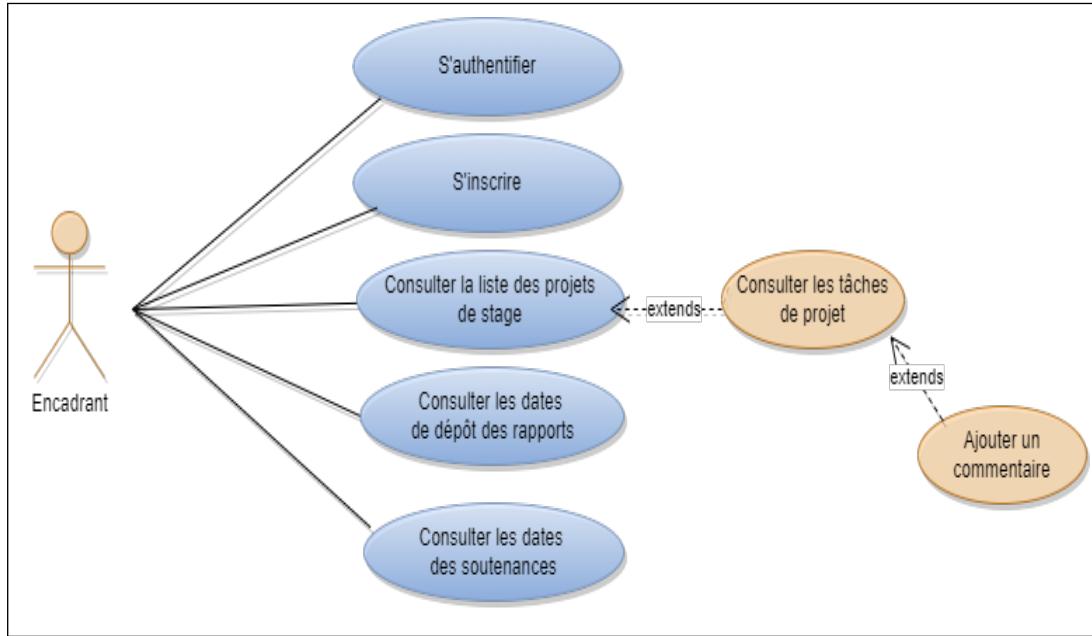
La Figure 2.3 présente le diagramme des cas d'utilisation détaillé pour l'étudiant.



**FIGURE 2.3 :** Diagramme de cas d'utilisation associé à l'Etudiant

#### 2.2.4.3 Diagramme de cas d'utilisation de l'acteur «Encadrant»

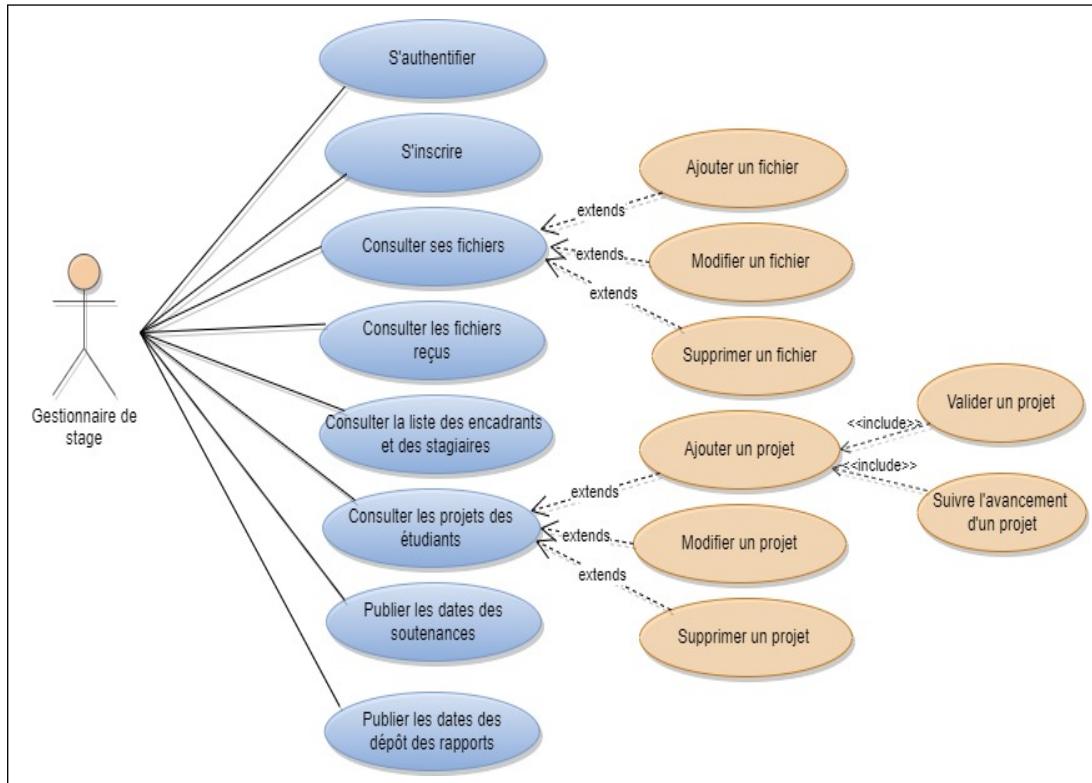
La Figure 2.4 présente le diagramme des cas d'utilisation détaillé pour l'encadrant.



**FIGURE 2.4 :** Diagramme de cas d'utilisation associé à l'Encadrant

#### 2.2.4.4 Diagramme de cas d'utilisation de l'acteur «Gestionnaire de stage»

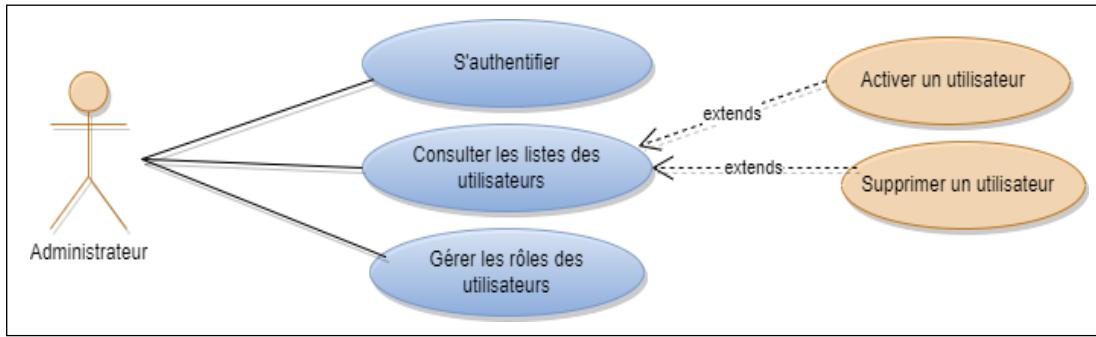
La Figure 2.5 présente le diagramme des cas d'utilisation détaillé pour le gestionnaire.



**FIGURE 2.5 :** Diagramme de cas d'utilisation associé au gestionnaire de stage

#### 2.2.4.5 Diagramme de cas d'utilisation de l'acteur «Administrateur»

La Figure 2.6 présente le diagramme des cas d'utilisation détaillé pour l'administrateur.



**FIGURE 2.6 :** Diagramme de cas d'utilisation associé à l'administrateur

### 2.3 Spécification des besoins non fonctionnels

Un besoin non fonctionnel correspond à une exigence qui décrit une caractéristique désirée du système, et notre solution satisfait les critères suivants :

- Simplicité** : La solution est facile à comprendre et à utiliser pour les utilisateurs finaux. Les fonctionnalités sont claires et intuitives pour minimiser la confusion et les erreurs potentielles.
- Sécurité** : Notre application utilise un système d'authentification robuste renforcé par l'utilisation de JSON Web Tokens (JWT) pour empêcher les intrusions non autorisées. Lors de l'inscription, les utilisateurs reçoivent un e-mail de confirmation pour vérifier la validité de leur adresse e-mail. De plus, L'administrateur doit approuver la demande d'authentification de chaque utilisateur pour lui permettre l'accès au système. Cette approche renforce la sécurité en empêchant l'utilisation de fausses adresses e-mail et en garantissant que seuls les utilisateurs authentiques peuvent accéder à notre application.
- Ergonomie** : Nous avons amélioré la convivialité de l'interface utilisateur pour rendre la navigation dans le système plus facile. Pour ce faire, nous avons incorporé des icônes et des images représentatives afin de faciliter l'identification des actions et des fonctionnalités essentielles. De plus, l'utilisation de couleurs appropriées et d'une structure de navigation cohérente contribue à offrir une expérience utilisateur optimale en favorisant une utilisation intuitive de l'application.
- Performance** : Nous avons réalisé une application web mono page (en anglais Single Page Application ou SPA) qui optimise le temps de chargement de l'application et évite

le chargement de page à chaque action demandée par l'utilisateur, il en résulte une expérience utilisateur fluide en ce qui concerne la navigation dans le site.

## 2.4 Spécification des besoins techniques

### 2.4.1 Environnement de développement

La création d'une application web implique l'utilisation d'une combinaison de technologies comme les frameworks, les serveurs, le langage, etc, qui peuvent fonctionner ensemble afin de garantir le bon fonctionnement de l'application. Pour notre plateforme, nous avons choisi d'utiliser la pile technologique MERN stack gratuite et open source et qui offre une approche de développement web efficace et moderne. Le terme MERN signifie MongoDB, Express, React, Node, d'après les quatre technologies clés qui composent la pile.

Nous présentons quelques avantages de l'utilisation de MERN Stack pour le développement web :

- + Développement plus rapide grâce à l'utilisation d'un même langage (Javascript) sur le front-end et le back-end.
- + Flexibilité et évolutivité avec une grande variété de technologies compatibles.
- + Hautement extensible et maintenable grâce à la modularité et à la lisibilité du code.
- + Open-source et rentable grâce à son téléchargement et utilisation gratuits et sa haute évolutivité.
- + Collaboration facile pour permettre à plusieurs développeurs de travailler sur le même projet.

### 2.4.2 Environnement logiciel

Cette partie présente les logiciels et les outils spécifiques que nous avons utilisés dans notre projet. Ces outils sont soigneusement sélectionnés pour répondre aux besoins de conception, de développement et de gestion de projet. Nous avons pris en compte les meilleures pratiques de l'industrie et les technologies les plus récentes pour garantir une efficacité et une qualité optimales dans notre travail.

**MongoDB** : Une base de données open source de type NoSQL qui stocke les données dans des documents JSON flexibles. Elle offre de bonnes performances et une évolutivité facile pour les applications web.



FIGURE 2.7 : Logo MongoDB [8]

**ExpressJS** : Un framework léger qui fournit une interface facile à utiliser pour connecter NodeJS à MongoDB. Il permet l'utilisation de routes RESTful et la connexion directe de pages HTML ou de modèles aux données de la base de données.



FIGURE 2.8 : Logo Express.js [9]

**ReactJS** : Une technologie front-end basée sur JavaScript qui utilise la programmation déclarative pour développer des applications monopages (SPA). ReactJS est un outil puissant qui permet d'étendre le langage HTML pour l'application.

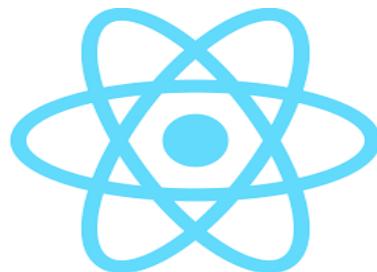


FIGURE 2.9 : Logo React Js [10]

**NodeJS** : Une plateforme qui utilise un modèle d'E/S non bloquant et piloté par les événements, ce qui la rend légère et efficace. Elle est capable de gérer des requêtes simultanées avec des débits élevés. NodeJS est livré avec une riche bibliothèque de divers modules JavaScript et permet l'utilisation de CoffeeScript.

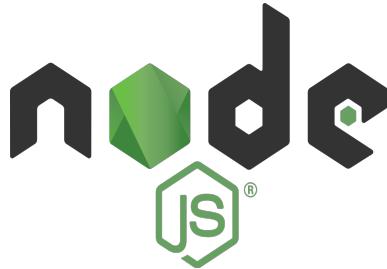


FIGURE 2.10 : Logo Node Js [11]

**Visual Studio Code** : est un éditeur de code open-source développée par Microsoft capable de supporter un très grand nombre de langages grâce à des extensions. Il supporte l'auto-completion, la coloration syntaxique, le débogage, et les commandes git.

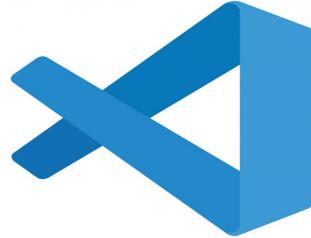


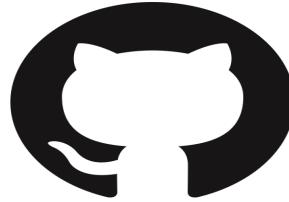
FIGURE 2.11 : Logo Vscode [12]

**Diagrams.net** est un logiciel de dessin graphique multiplateforme gratuit et open source développé en HTML5 et JavaScript. Son interface peut être utilisée pour créer des diagrammes tels que des organigrammes, des structures filaires, des diagrammes UML, des organigrammes et des diagrammes de réseau.



FIGURE 2.12 : Logo Draw.io [13]

**GitHub** : Lancé en 2008, GitHub est un site Web conçu pour fédérer et partager le code source d'un projet de développement d'application, mis à œuvre par plusieurs programmeurs. Il est basé sur l'outil open source de gestion de versions décentralisées Git.



**FIGURE 2.13 :** Logo Github [14]

**Canva** : est un outil graphique qui permet aux utilisateurs de réaliser des logos, des invitations, des publications pour vos réseaux sociaux, des CV, des flyers et beaucoup d'autres choses.



**FIGURE 2.14 :** Logo Canva [15]

### 2.4.3 Architecture de l'application

La conception architecturale vise à comprendre comment un système logiciel devrait être organisé et à concevoir sa structure globale. Lors de la conception d'un système logiciel, il est important de choisir des modèles architecturaux logiques et physiques qui s'y adaptent de manière appropriée afin d'assurer un bon fonctionnement entre les différentes couches. Cette approche garantit un fonctionnement optimal du système et favorise la coordination entre ses différentes composantes.

#### 2.4.3.1 Architecture logique

En termes d'architecture logique, la stack MERN que nous avons utilisé pour réaliser notre projet suit généralement une architecture en couches ou architecture MVC Modèle, Vue, Contrôleur pour organiser et structurer le code de l'application.

Cette architecture permet de séparer efficacement la logique de présentation, et elle est parfaitement adaptée à la nature et à la taille de notre projet. Grâce à cette séparation, il est facile d'ajouter ou de modifier du code dans une section sans affecter le reste. Le pattern MVC est particulièrement adapté au développement d'applications de petite et moyenne envergure. Le

but de MVC est justement de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts.

- **Modèle (Model)** : Cette partie gère ce qu'on appelle la logique métier de site. Elle comprend notamment la gestion des données qui sont stockées, mais aussi tout le code qui prend des décisions autour de ces données.
- **Vue (View)** : Cette partie se concentre sur l'affichage. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML.
- **Contrôleur (Controller)** : Cette partie gère les échanges avec l'utilisateur. C'est en quelque sorte l'intermédiaire entre l'utilisateur, le modèle et la vue. Le contrôleur demande au modèle les données, les analyser, prendre des décisions et finalement les déléguer à la vue.

La Figure 2.15 schématise les étapes de l'architecture MVC.

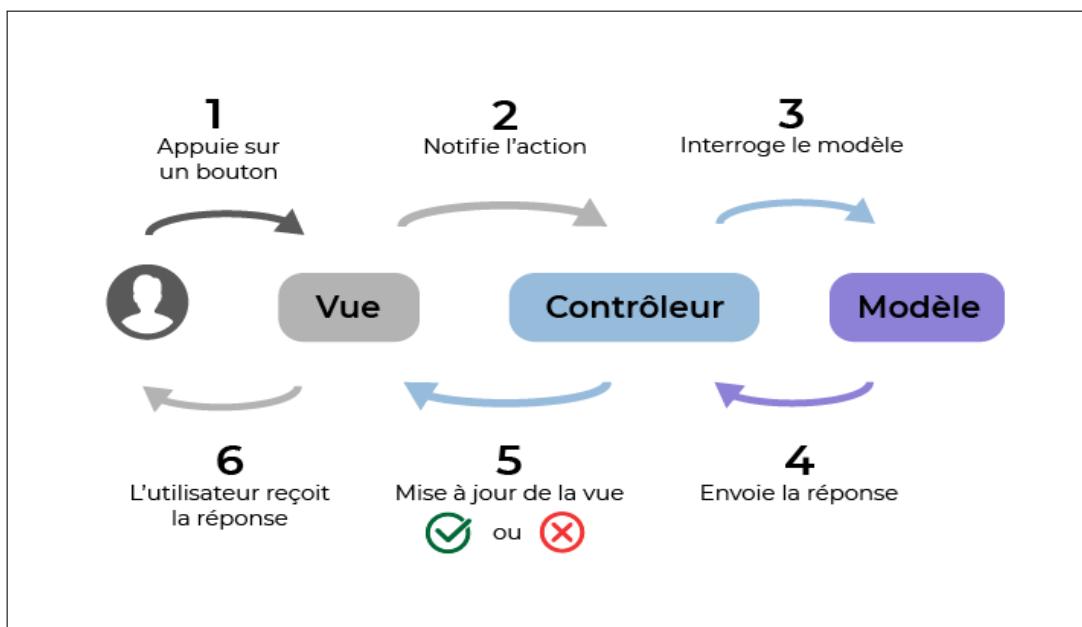


FIGURE 2.15 : Architecture MVC [16]

Dans le contexte de l'architecture MVC, le processus fonctionne de la manière suivante : lorsque le visiteur demande une page, cette requête est dirigée vers le contrôleur. Le contrôleur interagit ensuite avec le modèle pour obtenir les données nécessaires. Une fois que le contrôleur a récupéré les données, il construit la vue appropriée en utilisant ces données. Enfin, la vue est renvoyée au visiteur, qui la voit s'afficher dans son navigateur. Pour le visiteur, ce processus est transparent et il n'est pas conscient de tout ce qui se passe en coulisses sur le serveur.

L'architecture MVC peut sembler plus complexe que les approches plus traditionnelles, mais c'est précisément ce type d'architecture qui sous-tend de nombreux sites web professionnels.

#### 2.4.3.2 Architecture physique

L'architecture physique de l'application MVC peut être mise en œuvre selon différentes approches. L'une de ces approches est **l'architecture physique en trois tiers**, qui répartit les composants de l'application sur **trois couches distinctes** :

- **Couche de présentation** : est responsable de l'interaction avec les utilisateurs. Elle comprend généralement les interfaces utilisateur, les pages web, les formulaires, les vues ou les composants graphiques.
- **Couche logique** : est responsable du traitement des données et de la logique métier de l'application. Elle contient généralement les classes ou les composants qui manipulent les données, effectuent des calculs, exécutent des règles métier et interagissent avec d'autres composants de l'application.
- **Couche de données** : est responsable de la persistance des données de l'application. Elle peut inclure des bases de données, des fichiers, des services web ou d'autres sources de données.

La figure 2.16 présente l'architecture 3-tiers avec les technologies MERN.

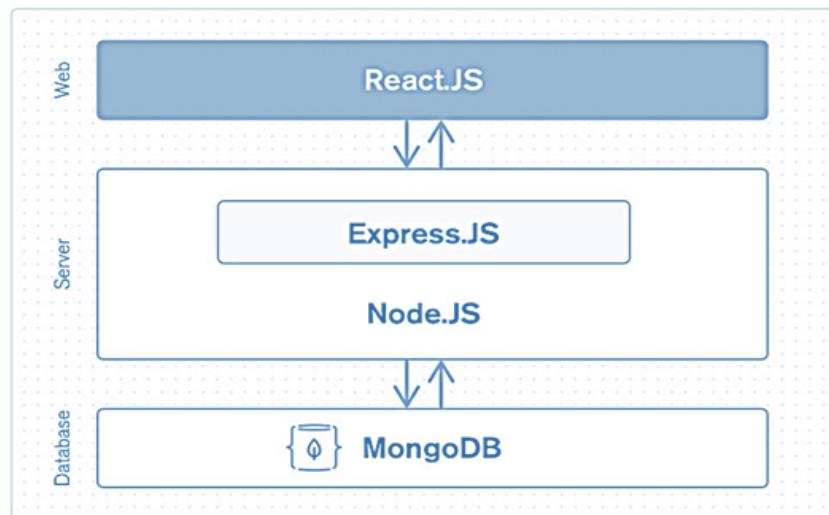


FIGURE 2.16 : Architecture 3 tiers avec MERN [17]

L'architecture physique en trois tiers permet une meilleure séparation des responsabilités et une évolutivité de l'application. Chaque couche peut être développée, testée et déployée

indépendamment, ce qui facilite la maintenance et les modifications futures. De plus, cette approche facilite également la réutilisation des composants et favorise une meilleure gestion des erreurs et de la sécurité.

## 2.5 Conclusion

Ce chapitre a abordé les besoins fonctionnels, non fonctionnels et techniques, ainsi que l'architecture logique et physique du système. Le chapitre suivant se concentre sur la conception détaillée de l'application.

# CONCEPTION

---

## Plan

1	Introduction	28
2	Langage de modélisation utilisée	28
3	Conception de la base de données	28
4	Conception détaillée	32
5	Conclusion	39

### 3.1 Introduction

Dans ce chapitre, nous commençons par une brève présentation de langage de modélisation utilisée. Ensuite, nous abordons la conception de la base de données, en décrivant tout d'abord les concepts généraux. Ensuite, nous nous concentrerons sur la conception détaillée en utilisant différents types de diagrammes et en montrant comment ils interagissent les uns avec les autres.

### 3.2 Langage de modélisation utilisée

Notre choix s'est porté sur l'utilisation de langage de modélisation UML (Unified Modeling Language), qui est un langage graphique utilisé pour représenter visuellement les différents aspects d'un système logiciel. UML permet de capturer les besoins, les fonctionnalités, les processus métier, l'architecture, la conception et l'implémentation d'un système de manière claire et précise [18]. La dernière version de l'UML est la version 2.5.1, publiée en 2017. Cette version a apporté plusieurs améliorations et clarifications par rapport aux versions précédentes.

Les principales améliorations de l'UML 2.5.1 incluent :

- Une meilleure prise en charge des architectures orientées service (SOA) et des modèles de données.
- De nouvelles notations pour les diagrammes de classes et de packages, qui les rendent plus faciles à lire et à comprendre.
- Des améliorations pour les diagrammes d'activités, qui permettent une représentation plus détaillée des processus métier.
- Une meilleure intégration avec d'autres langages de modélisation, comme SysML (Systems Modeling Language) et BPMN (Business Process Model and Notation).

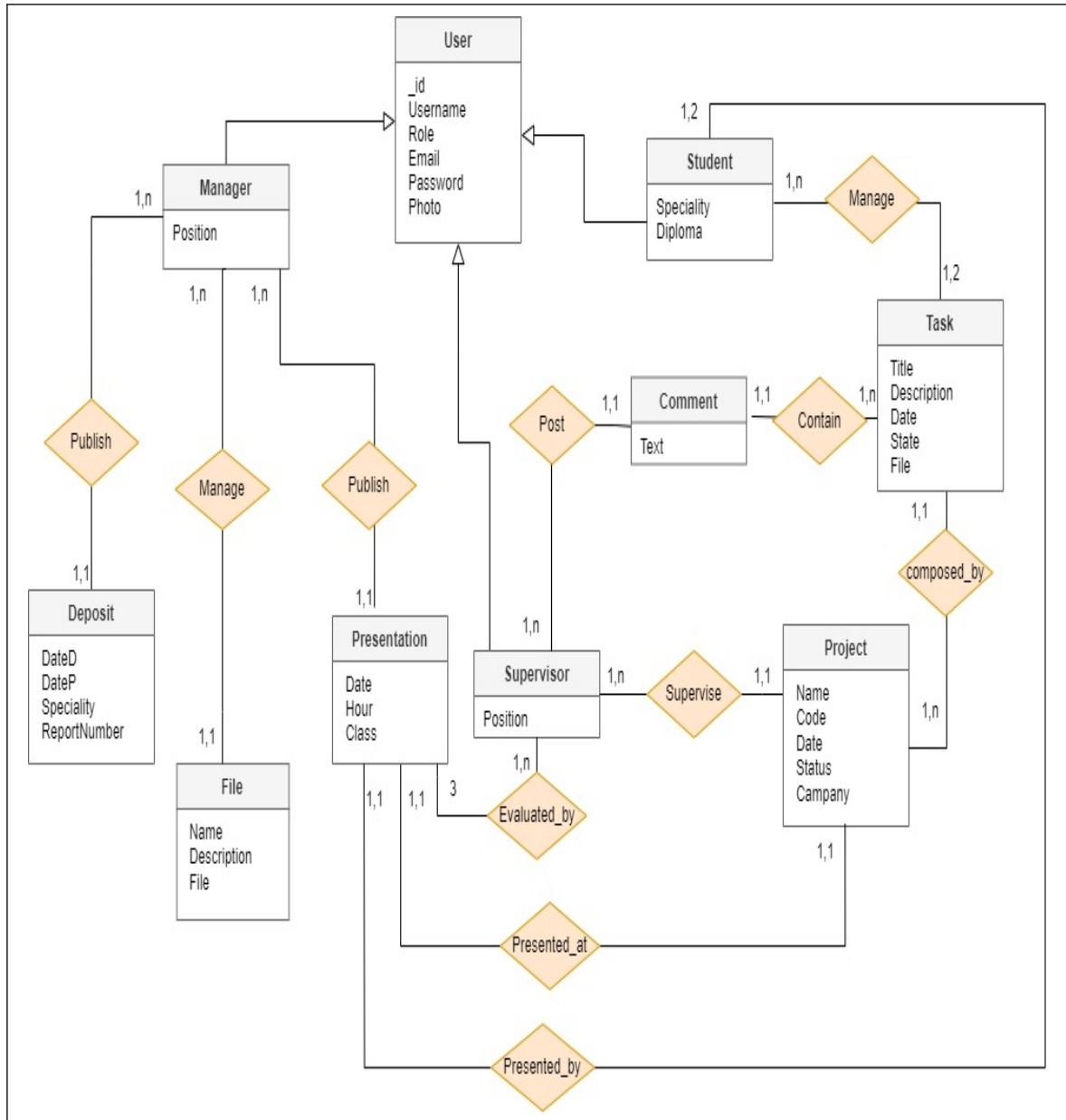
### 3.3 Conception de la base de données

La création d'une base de données implique d'organiser les données selon un modèle spécifique, ce qui permet de déterminer comment elles doivent être stockées et comment elles sont interconnectées. Pour ce faire, nous commençons par élaborer un modèle conceptuel de données (MCD) qui sera ensuite suivi par la création d'un modèle orienté document.

### 3.3.1 Modèle conceptuel des données

Un modèle conceptuel de données est une représentation utilisée pour décrire les entités distinctes d'un système et les relations qui existent entre elles. Il permet de visualiser et de structurer les données de manière conceptuelle, en se concentrant sur les entités et leurs associations. Le MCD offre une vue globale et abstraite des données, facilitant ainsi la compréhension des relations entre les différentes entités d'un système [19].

La figure 3.1 représente le modèle conceptuel de notre base de données.



**FIGURE 3.1 :** Modèle conceptuel des données

Le tableau 3.1 donne les détails de description de chaque entité.

**TABLEAU 3.1 :** Les entités du modèle conceptuel de données

Entités	Description	Attributs et type
<b>User</b>	Tout utilisateur de la plateforme.	-id : ObjectId -Username : String -Role : String -Email : String -Password : String -Photo : String
<b>Manager</b>	Elle hérite la classe «User» et contient un attribut supplémentaire.	-Position : String
<b>Supervisor</b>	Elle hérite «User» et contient un autre attribut.	-Position : String
<b>Student</b>	Elle hérite la classe « User » et contient les attributs supplémentaires relatives à l'étudiant.	-Speciality : Enum -Diploma : Enum
<b>Project</b>	Elle contient les détails du projet effectué par un ou deux étudiants.	-Name : String -Code : String -Date : Date -Status : Enum -Company : String
<b>Task</b>	Elle contient les details des tâches associées à chaque projet.	-Title : String -Description : String -Date : Date -State : Enum -File : String
<b>Comment</b>	Elle le texte de commentaire ajouté par l'encadrant à une tâche.	-Text : String
<b>Presentation</b>	Elle contient les détails spécifiques de la soutenance.	-Date : Date -Hour : String -Class : String
<b>Fiche</b>	Elle contient les informations des fichiers envoyées par le gestionnaire ou l'étudiant.	-Name : String -Description : String -File : String

<b>Deposit</b>	Elle contient les dates du dépôt des rapports et du présentation .	-DateD : Date -DateP : Date -Speciality : Enum -ReportNumber : String
----------------	--	--

### 3.3.2 Modèle orientée document

Un modèle orienté document est une approche de modélisation de données utilisée dans les bases de données NoSQL, telles que MongoDB. Dans ce modèle, les données sont stockées sous forme de documents, qui sont des structures flexibles et auto-descriptives représentées dans un format tel que JSON (JavaScript Object Notation) ou BSON (Binary JSON) [20].

La Figure 3.2 illustre le modèle orienté document de notre projet.

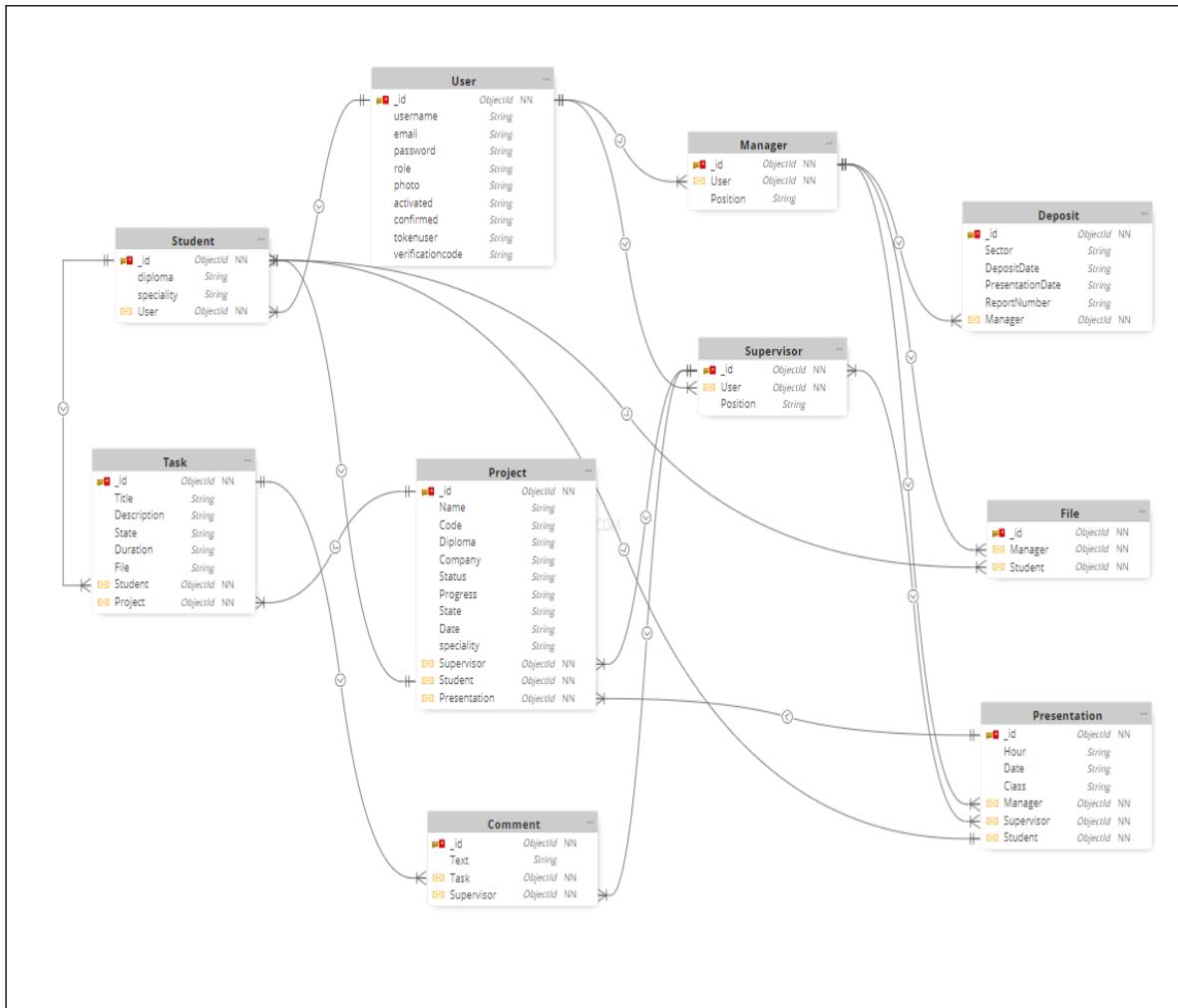


FIGURE 3.2 : Modèle orienté document

## 3.4 Conception détaillée

La conception détaillée fait référence à l'étape de conception approfondie d'un système, où les aspects spécifiques et les détails techniques sont pris en compte pour élaborer une solution concrète. Cela implique de transformer les concepts généraux et abstraits en spécifications précises et détaillées.

### 3.4.1 Diagramme de classe

Un diagramme de classe est un type de diagramme UML utilisé pour représenter la structure statique d'un système logiciel. Il décrit les classes du système, leurs attributs, leurs méthodes, ainsi que les relations et les dépendances entre elles.

Le diagramme de classe de notre projet est présenté dans la Figure 3.3.

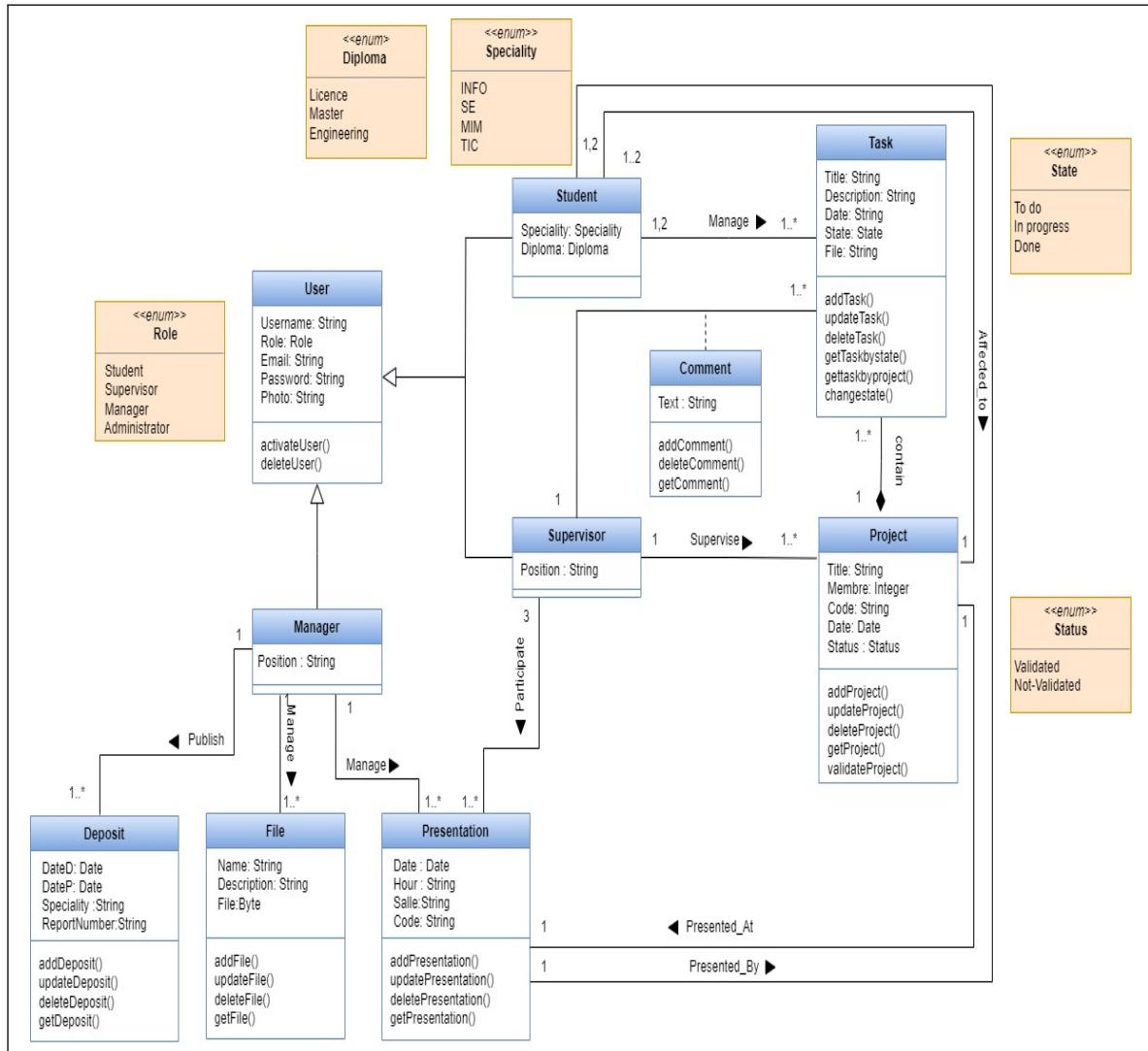


FIGURE 3.3 : Diagramme de classe général

### 3.4.2 Diagrammes de séquence objet

Un diagramme de séquence objet est un type de diagramme UML qui permet de représenter la séquence des interactions entre les objets d'un système logiciel. Il met en évidence les messages échangés entre les objets pendant l'exécution d'un scénario ou d'une fonctionnalité spécifique.

#### 3.4.2.1 Diagramme de séquence objet «S'inscrire»

Le diagramme présenté dans la figure 3.4 est spécifique au cas d'utilisation "s'inscrire" et est réalisé par l'utilisateur présente de manière séquentielle le scénario d'inscription d'un utilisateur, en mettant en évidence les objets impliqués dans le processus et leurs interactions conformément au modèle MVC et en prenant en compte que le mécanisme d'inscription est basé sur l'envoi d'un e-mail. Cela permet d'assurer la sécurité et l'identification précise des utilisateurs.

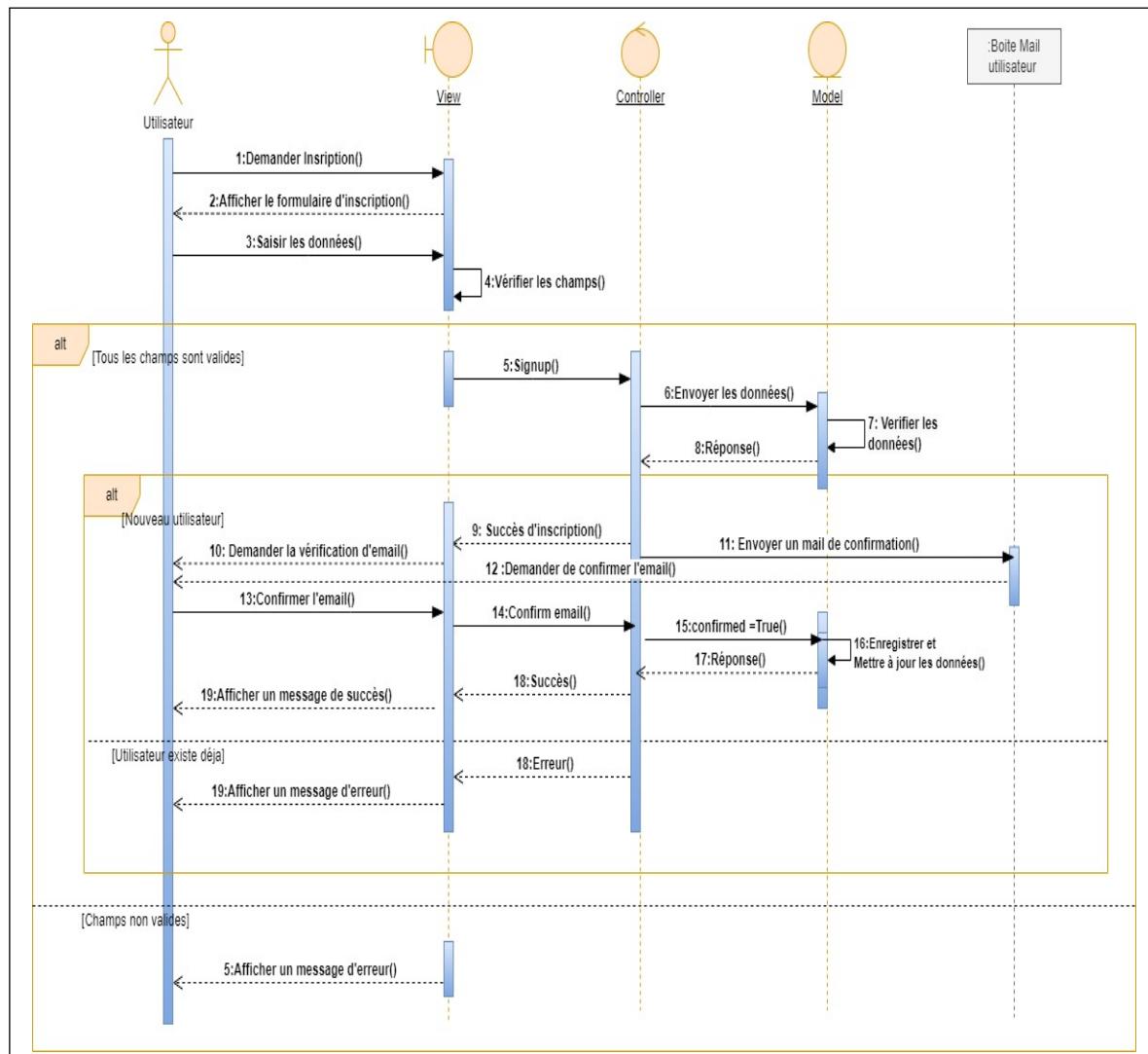


FIGURE 3.4 : Diagramme de séquence objet «S'inscrire»

### 3.4.2.2 Diagramme de séquence objet «Ajouter une tâche»

Le diagramme de séquence présenté dans la figure 3.5 illustre les échanges entre l'étudiant, la vue, le contrôleur et le modèle lors de l'exécution de l'opération "Ajouter une tâche" au projet effectué par un ou deux étudiants.

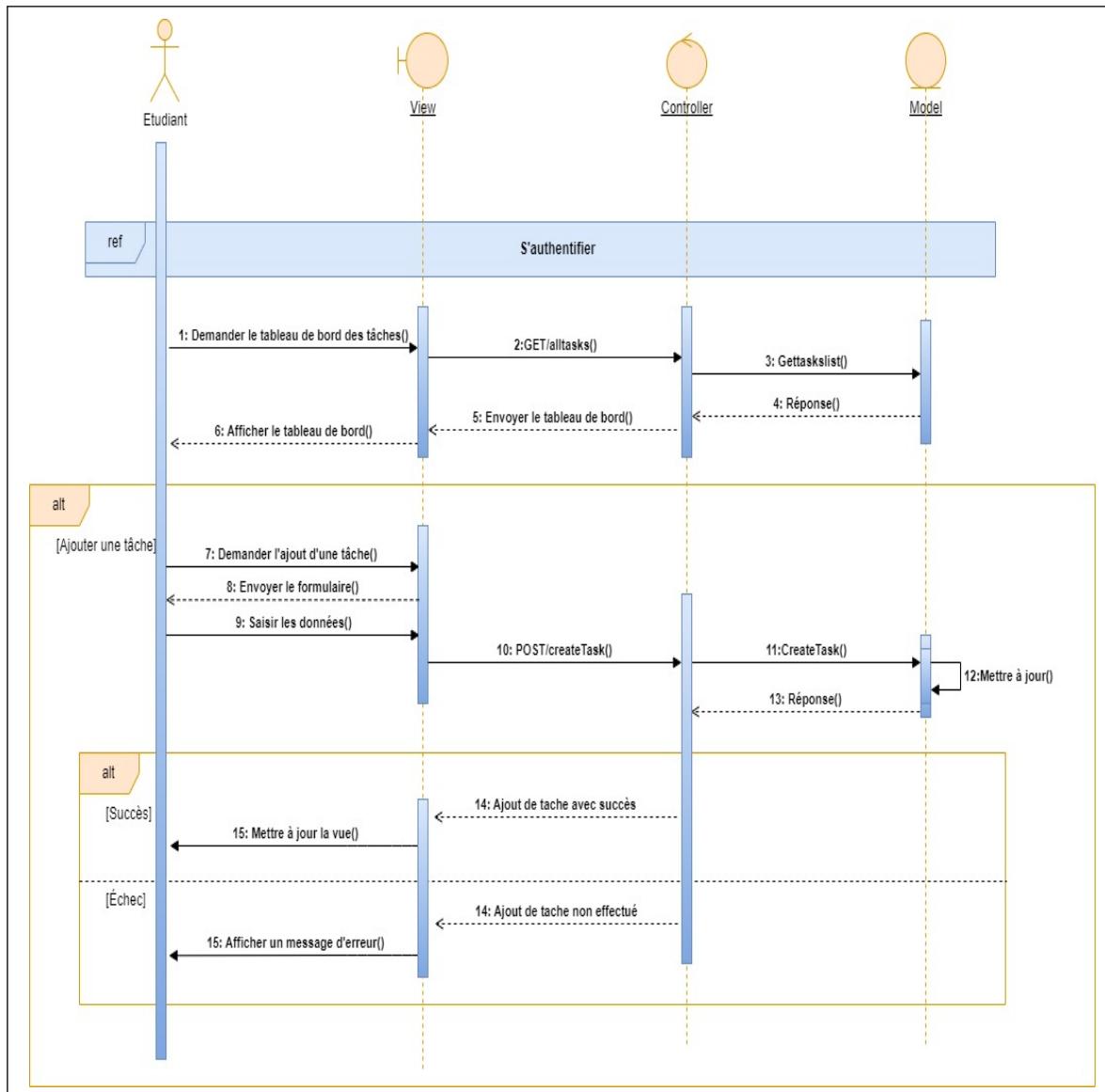
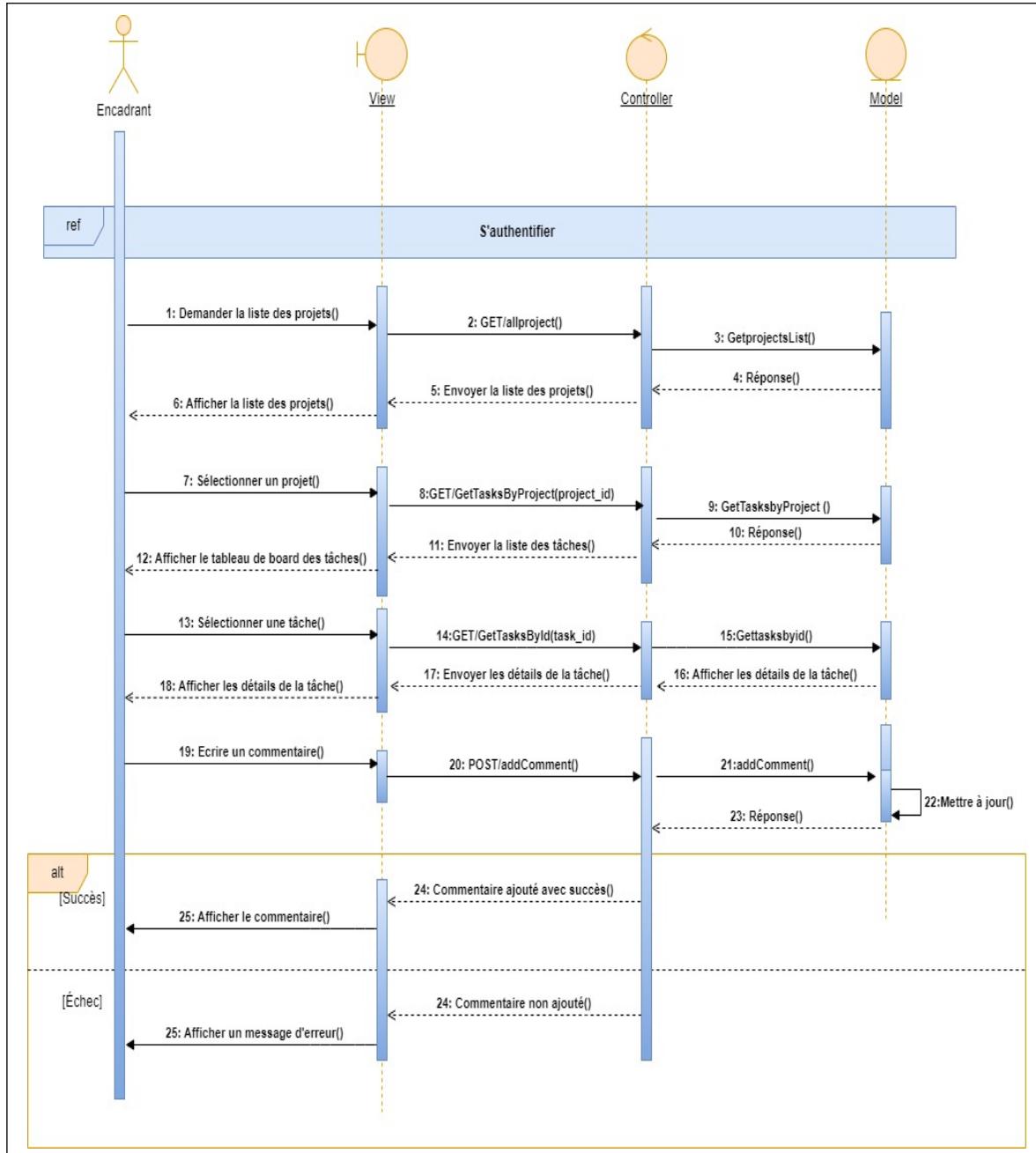


FIGURE 3.5 : Diagramme de séquence objet «Ajouter une tâche»

### 3.4.2.3 Diagramme de séquence objet «Ajouter un commentaire»

La figure 3.6 présente le diagramme de séquence objet intitulé "Ajouter un commentaire". Ce diagramme illustre la manière dont un encadrant ajoute un commentaire à une tâche spécifique d'un projet parmi ceux qu'il supervise.



**FIGURE 3.6 : Diagramme de séquence objet «Ajouter un commentaire»**

#### 3.4.2.4 Diagramme de séquence objet «Valider un projet»

La figure 3.7 présente le diagramme de séquence objet intitulé "Valider un projet". Ce diagramme vise à illustrer de manière détaillée les étapes que suit un gestionnaire pour valider un projet donné. Il met en évidence les interactions entre les différents objets et acteurs impliqués dans le processus de validation.

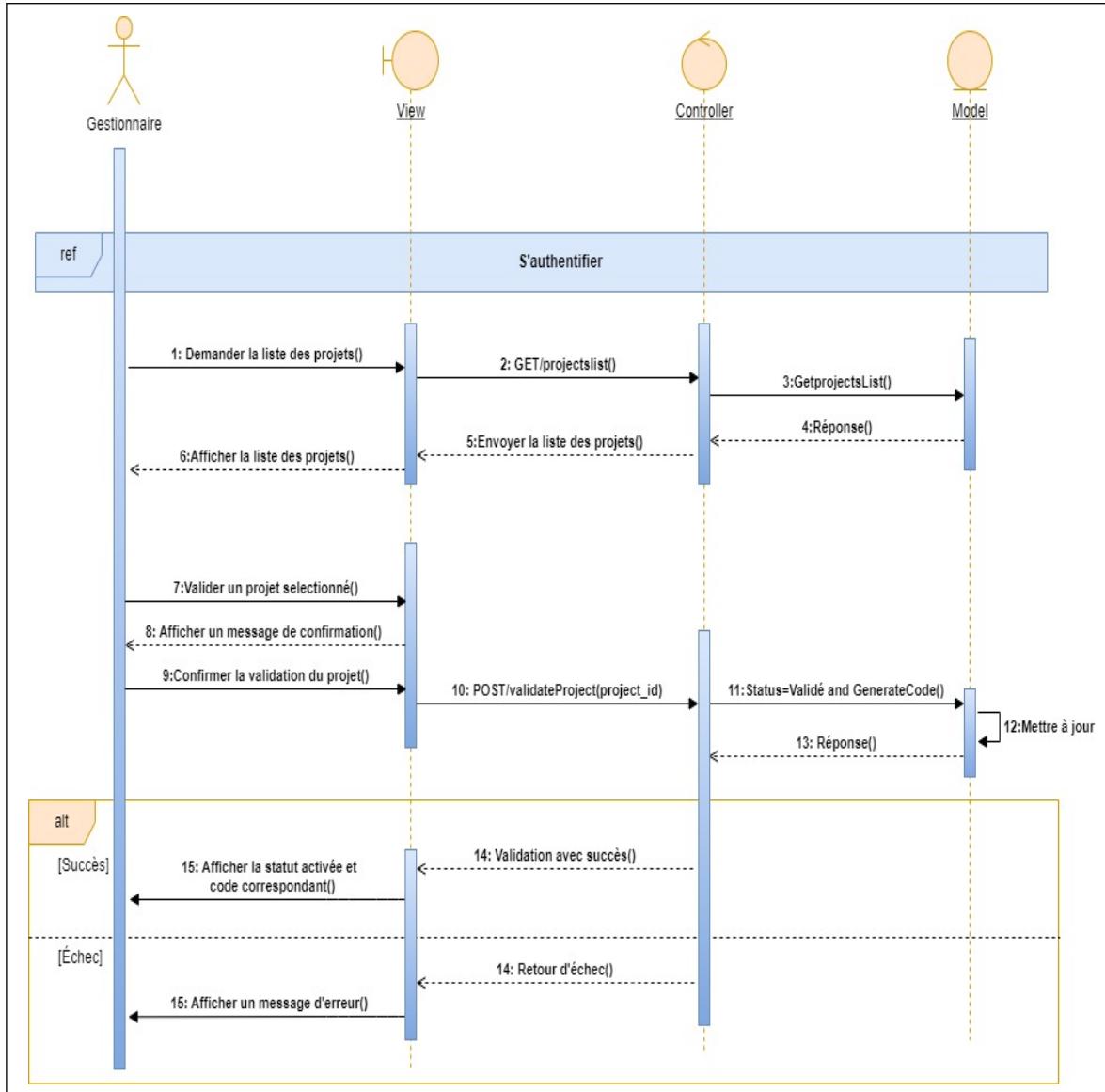


FIGURE 3.7 : Diagramme de séquence objet «Valider un projet »

### 3.4.3 Diagrammes d'activités

Le diagramme d'activités est spécialement conçu pour mettre en évidence les processus et activités. Il est particulièrement approprié pour modéliser les chemins de contrôle et de données, ce qui permet de représenter visuellement le comportement d'une méthode ou le déroulement d'un cas d'utilisation. En d'autres termes, il permet de décrire graphiquement les flux de données et de contrôle à travers les différentes étapes d'un processus ou d'une méthode.

#### 3.4.3.1 Diagramme d'activité «Activer un utilisateur»

Le diagramme d'activité d'activation d'un utilisateur, réalisé par l'administrateur, est présenté dans la Figure 3.8. Il détaille les différentes étapes pour activer un utilisateur dans le système.

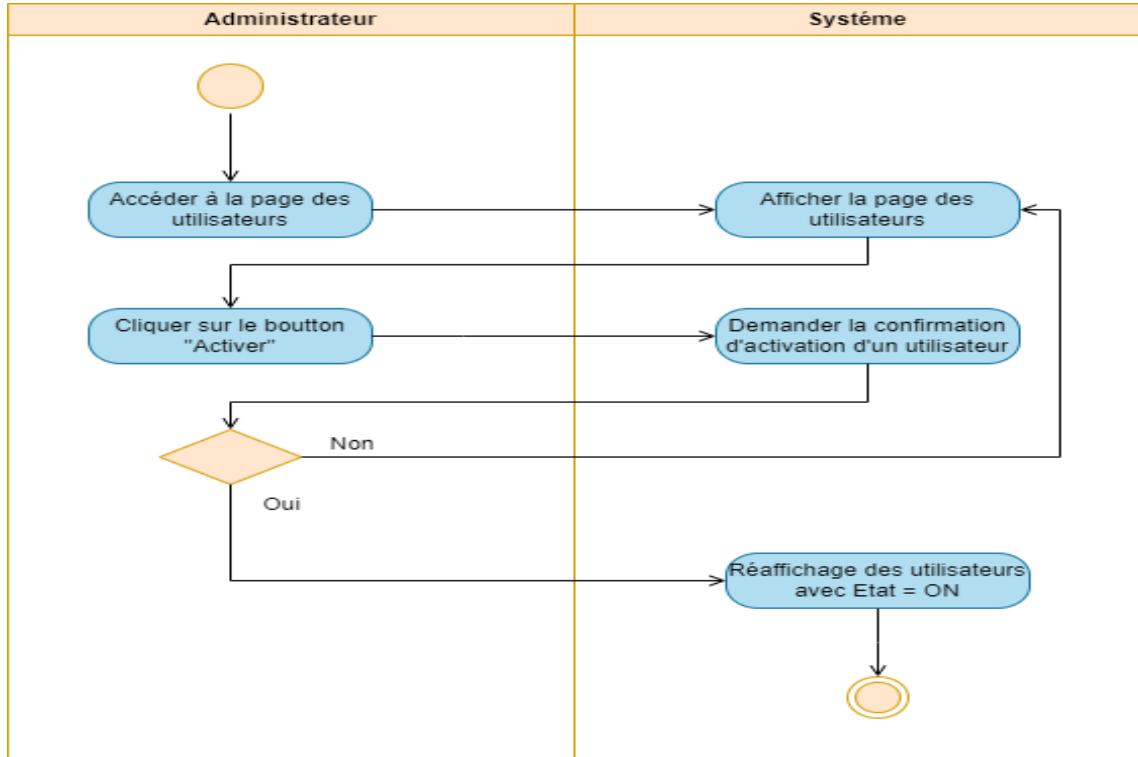


FIGURE 3.8 : Diagramme d'activité «Activer un utilisateur»

#### 3.4.3.2 Diagramme d'activité «Ajouter un commentaire»

La Figure 3.9 illustre de manière détaillée les étapes lorsqu'un encadrant ajoute un commentaire.

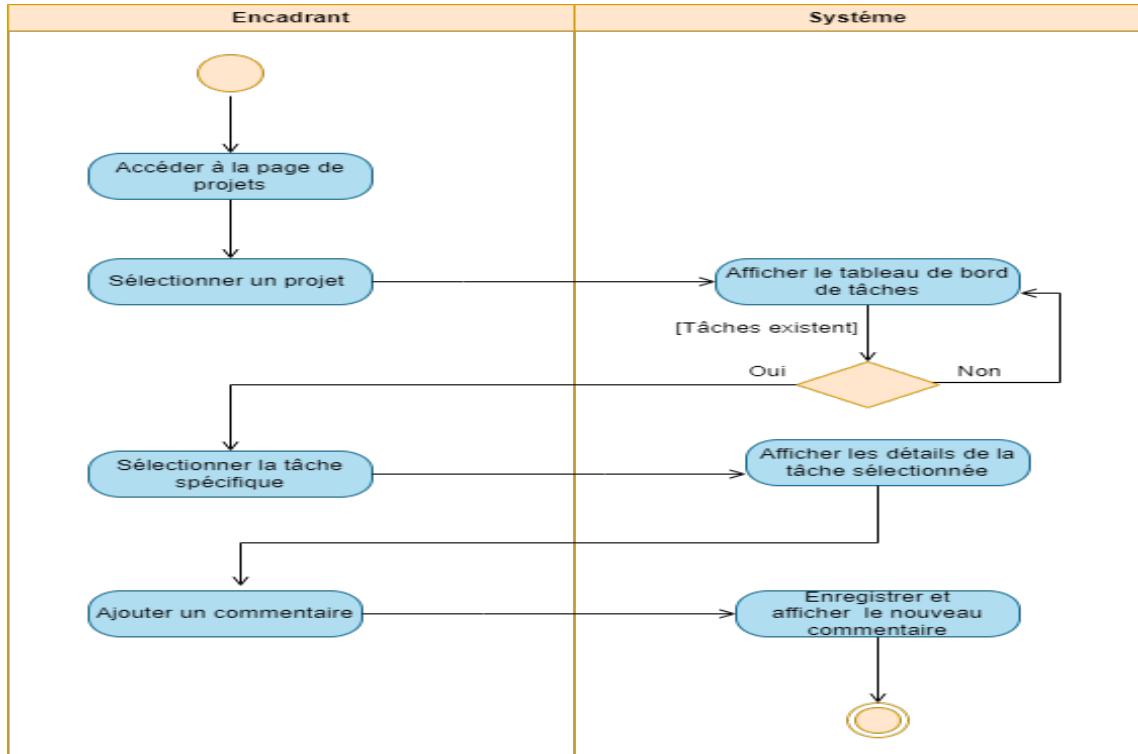
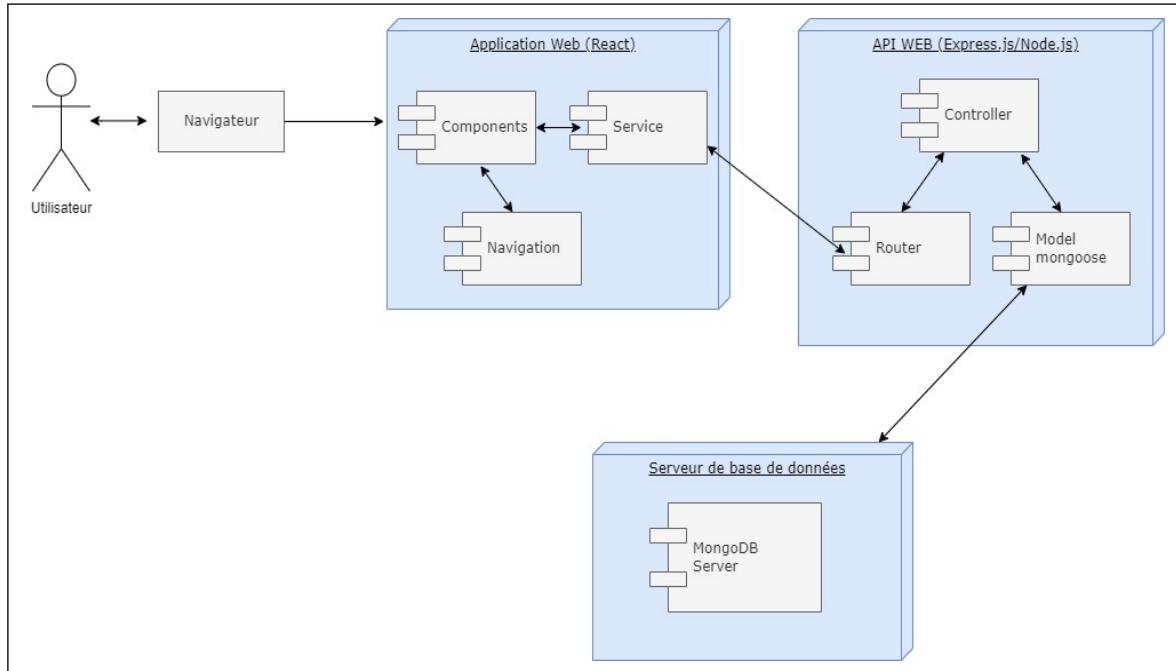


FIGURE 3.9 : Diagramme d'activité «Ajouter un commentaire»

### 3.4.4 Diagramme de déploiement

Un diagramme de déploiement est utilisé pour représenter l'architecture physique d'un système logiciel, en montrant comment les différents composants matériels et logiciels sont déployés et interconnectés. Il décrit les relations entre les nœuds matériels, tels que les serveurs, les ordinateurs, les périphériques réseau, ainsi que les logiciels qui s'exécutent sur ces nœuds.

La Figure 3.10 présente le diagramme de déploiement de notre application



**FIGURE 3.10 :** Diagramme de déploiement

Dans notre modèle, nous avons trois noeuds principaux :

- **Nœud Application Web** : Ce nœud exécute les composants de l'application. Il comprend les composants qui gèrent l'affichage et l'interaction avec l'utilisateur, ainsi que les services qui fournissent des fonctionnalités spécifiques. La navigation est également prise en compte pour gérer les transitions entre les différentes pages de l'application.
- **Nœud API** : Ce nœud exécute les éléments liés à l'API de l'application. Il comprend le modèle, qui définit la structure des données et gère l'accès à la base de données. Le router gère les points d'accès de l'API et dirige les requêtes vers les contrôleurs appropriés. Les contrôleurs traitent les requêtes en interagissant avec le modèle pour récupérer, mettre à jour ou supprimer des données.
- **Nœud Base de données** : Ce nœud exécute MongoDB, qui est utilisé comme système de gestion de base de données pour stocker les données de l'application.

### 3.5 Conclusion

Au cours de ce chapitre, nous avons initialement choisi le langage de modélisation à utiliser, puis nous avons procédé à la conception de la base de données. Ensuite, nous avons défini la vue statique de notre plateforme en créant le diagramme de classe, et la vue dynamique en utilisant les diagrammes de séquence et d'activité. Maintenant, nous sommes prêts à passer à l'étape suivante, qui consiste à réaliser concrètement notre plateforme.

---

# RÉALISATION

---

## Plan

1	Introduction	41
2	Implémentation technique de la solution	41
3	Conclusion	49

## 4.1 Introduction

Dans le dernier chapitre de ce rapport, nous nous focalisons sur la présentation de notre solution, une fois que sa conception a été abordée dans le chapitre précédent. L'objectif principal consiste à fournir une description détaillée des diverses fonctionnalités offertes par notre solution.

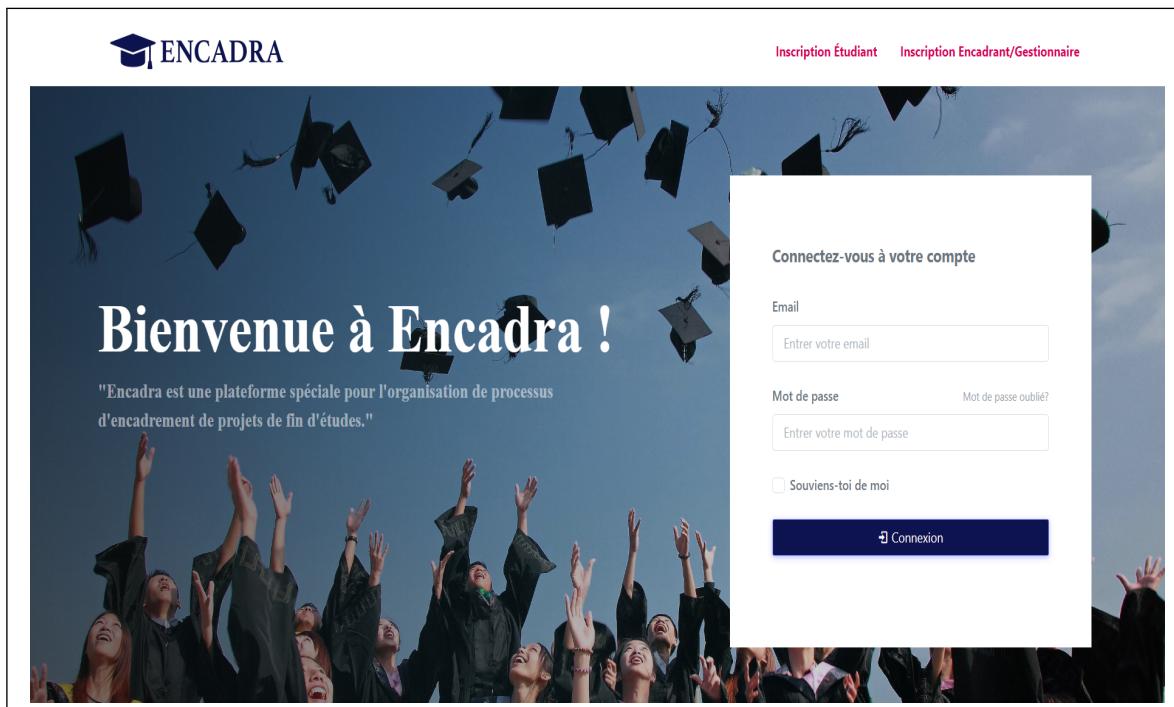
## 4.2 Implémentation technique de la solution

Dans cette section, nous présentons les différents espaces disponibles sur notre plateforme, à savoir l'espace utilisateur, l'espace administrateur, l'espace encadrant, l'espace étudiant et l'espace gestionnaire. Pour chaque espace, nous allons mettre en évidence deux ou trois interfaces spécifiques afin de fournir un aperçu complet des fonctionnalités offertes.

### 4.2.1 Espace utilisateur

- **Interface d'accueil**

La figure 4.1 met en évidence la page d'accueil captivante de notre plateforme. Cette page joue un rôle essentiel en fournissant une première impression positive aux utilisateurs et en les engageant dès le départ.



**FIGURE 4.1 :** Interface d'accueil

- **Interface d'inscription**

Notre plateforme dispose de deux formulaires distincts : l'un destiné aux étudiants, et l'autre réservé aux encadrants et gestionnaires de stage.

La figure 4.2 présente le deux formulaires d'inscription.

The image displays two side-by-side registration forms for the ENCADRA platform. Both forms are titled "Créez votre compte" and feature a logo of a graduation cap and the word "ENCADRA".

**Left Form (Student Registration):**

- Nom et prénom:** Entrer votre nom complet
- Rôle:** Selectionner votre rôle
- Adresse Email:** Entrer votre email
- Diplôme:** Selectionner votre diplôme
- Filière:** Selectionner votre filière
- Mot de passe:** Entrer votre mot de passe
- Photo:** Choose file (No file chosen)

**Right Form (Mentor/Gatekeeper Registration):**

- Nom et prénom:** Entrer votre nom complet
- Rôle:** Selectionner un rôle
- Adresse Email:** Enter your email
- Mot de passe:** Entrer votre mot de passe
- Photo:** Choose file (No file chosen)

Both forms include a "S'inscrire" button at the bottom. A note at the bottom right of the right form says "Vous avez déjà un compte ? Se connecter".

**FIGURE 4.2 :** Interface d'inscription

#### 4.2.2 Espace administrateur

##### • Interface «Gérer les utilisateurs»

L'administrateur a la possibilité de gérer les utilisateurs, comme présenté dans le tableau 4.3. Les utilisateurs sont répertoriés dans un tableau, et l'administrateur peut activer ou supprimer un utilisateur. Lorsque le statut est activé (ON), l'utilisateur est autorisé à utiliser l'application. Cela offre à l'administrateur un contrôle précis sur l'accès des utilisateurs à la plateforme, garantissant ainsi une gestion flexible et sécurisée des comptes.

## Chapitre 4. Réalisation

The screenshot shows the 'Utilisateurs' (Users) management interface. On the left is a sidebar with a dark blue background and white text, titled 'ENCADRA'. It has a 'PAGES' section with links: Utilisateurs, Encadrants, Etudiants, Projets, Fichiers, Dépôt, and Soutenances. The main area is titled 'Utilisateurs' and shows a table of users:

<input type="checkbox"/>	Nom	Rôle	Email	Etat	Actions
<input type="checkbox"/>	Nour saidane	Etudiant	noursaidane@gmail.com	ON	<button>Activer</button> <button>Supprimer</button>
<input type="checkbox"/>	Ben saleh Mohamed	Encadrant	bensalahm@gmail.com	ON	<button>Activer</button> <button>Supprimer</button>
<input type="checkbox"/>	Nasri Tarek	Encadrant	tareknasri@gmail.com	ON	<button>Activer</button> <button>Supprimer</button>
<input type="checkbox"/>	Fawzi slim	Gestionnaire	fawzislim@gmail.com	ON	<button>Activer</button> <button>Supprimer</button>
<input type="checkbox"/>	Ben abdelkader Maher	Encadrant	maherbad@gmail.com	OFF	<button>Activer</button> <button>Supprimer</button>
<input type="checkbox"/>	Ben slimen wael	Encadrant	waelbnslimen@gmail.com	OFF	<button>Activer</button> <button>Supprimer</button>

FIGURE 4.3 : Interface «Gérer les utilisateurs»

### 4.2.3 Espace gestionnaire

- Interface «Gérer les projets»

Le gestionnaire de stage est chargé de gérer les projets. Après leur ajout dans le tableau, il peut sélectionner un projet et le valide, ce qui change son statut en "validé". À ce moment-là, un code sera automatiquement généré et assigné à ce projet comme illustré dans la figure 4.4.

The screenshot shows the 'Projets' (Projects) management interface. On the left is a sidebar with a dark blue background and white text, titled 'ENCADRA'. It has a 'PAGES' section with links: Encadrants, Etudiants, Projets, Fichiers, Dépôt, and Soutenances. The main area is titled 'Projets' and shows a table of projects:

<input type="checkbox"/>	Code	Nom du projet	Etudiant 1	Etudiant 2	Filière	Diplôme	Établissement	Encadrant	Statut	Action
<input type="checkbox"/>	L1INFO	Application web e-commerce	Amen allah adem	Sandid Aziz	INFO	Licence	Clever	Taieb Mohsen	Validé	
<input type="checkbox"/>	L2INFO	Plateforme de gestion de facturation	Mohamed Amira		INFO	Licence	Info go	Toumia Walid	Validé	
<input type="checkbox"/>	L3INFO	Plateforme de gestion de processus d'encadrement	Zemzem Jihen		INFO	Licence	Imotion+	Chiboub Ibrahim	Validé	
<input type="checkbox"/>		Plateforme d'organisation de tournois de jeux vidéos	Nour saidane	Romthane Eslem	INFO	Licence	Mobelite	Chiboub Ibrahim	Non validé	
<input type="checkbox"/>		Application web e-commerce	Amen allah adem		INFO	Licence	Clever	Chiboub Ibrahim	Non validé	

FIGURE 4.4 : Interface «Gérer les projets»

## Chapitre 4. Réalisation

Une fois que le gestionnaire a soumis une demande de validation pour un projet, un message de confirmation s'affiche, tel qu'illustré dans la figure 4.5.

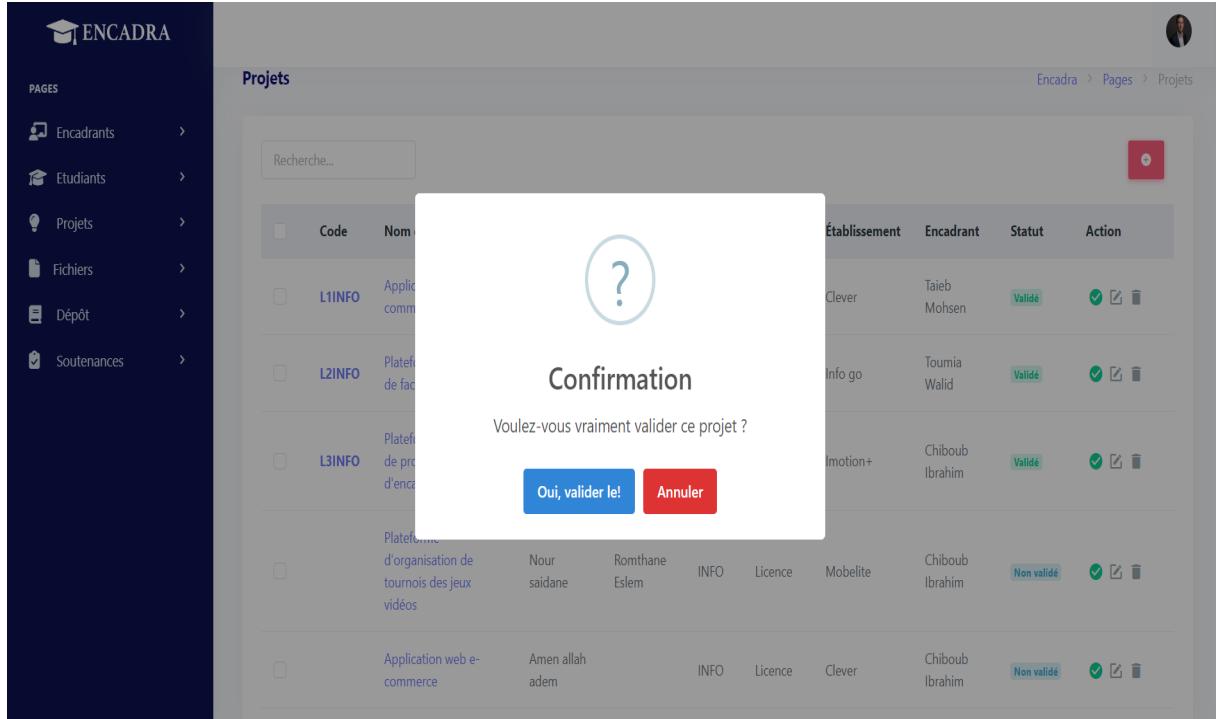


FIGURE 4.5 : Message «Confirmer validation»

Le message affiché après une validation réussie est présenté dans la figure 4.6.

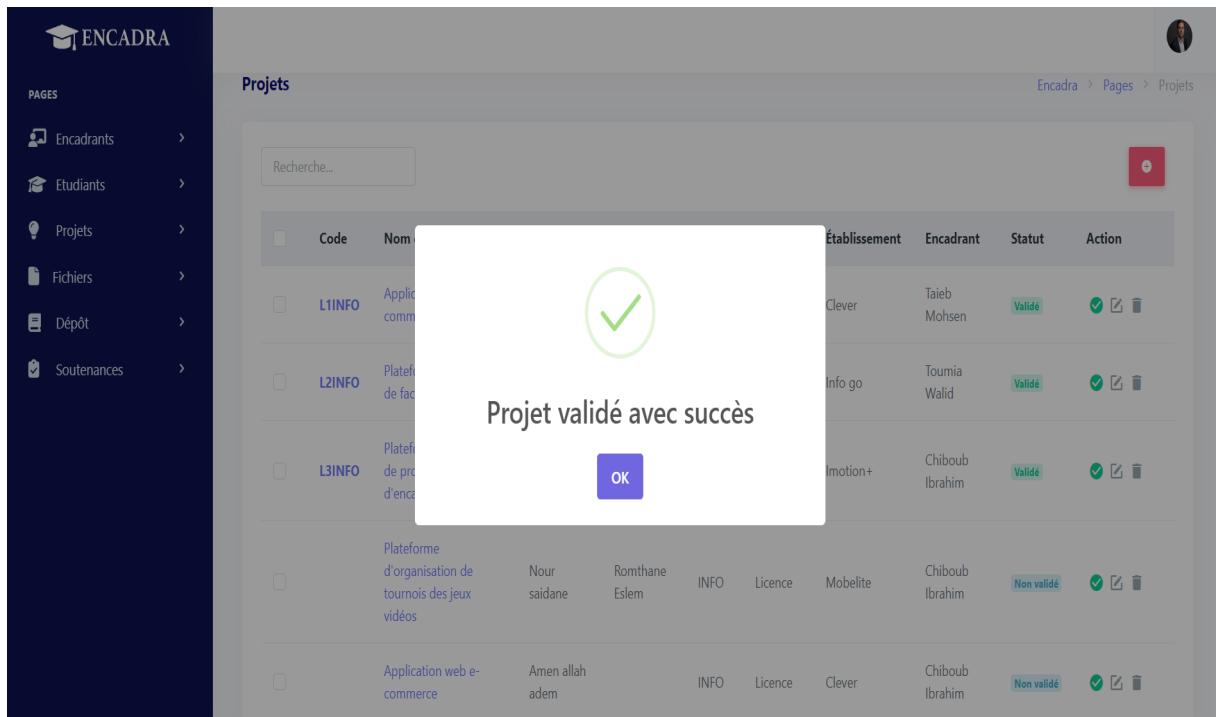


FIGURE 4.6 : Message «Succès de validation»

- Interface «Suivre progrès» pour le «Gestionnaire»

En plus des fonctionnalités précédemment mentionnées, le gestionnaire de stage dispose d'un tableau de bord puissant, illustré par la figure 4.7. Ce tableau de bord lui permet de visualiser les données et les indicateurs importants liés aux projets en cours. Les graphiques offrent une représentation visuelle claire de l'avancement, des performances et de l'état global des projets.

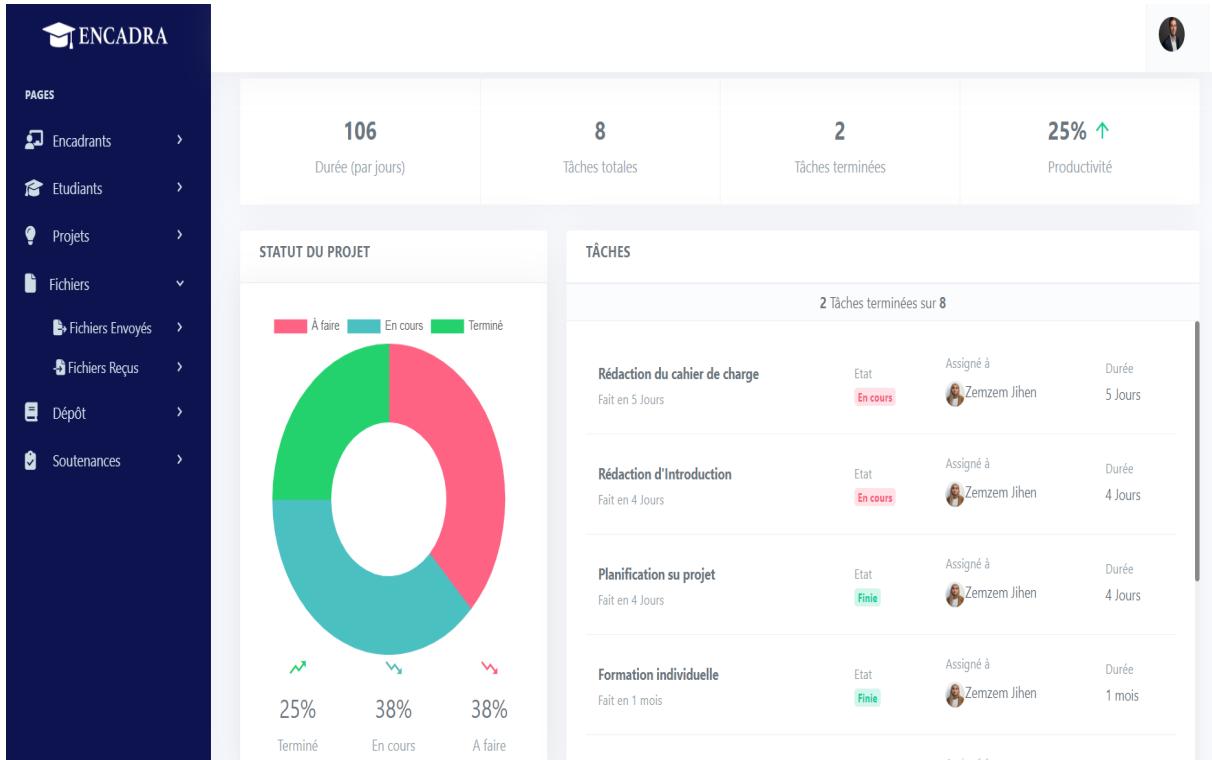


FIGURE 4.7 : Interface «Suivre progrès» pour le «Gestionnaire»

#### 4.2.4 Espace Etudiant

- Interface «Gérer les tâches»

Pour faciliter la gestion de leur propre projet, l'étudiant bénéficie d'un tableau de bord dédié offrant une visibilité et une compréhension accrues. Ils peuvent ajouter des tâches et les fichiers correspondants. Les tâches sont initialement classées dans la section "À Faire" par défaut, mais l'étudiant a la liberté de les déplacer vers la section "En cours" une fois qu'elles sont en cours d'exécution, puis vers la section "Terminé" une fois qu'elles sont achevées. Cela permet à l'étudiant de suivre facilement l'avancement de son projet et de garder une trace claire de ses tâches à effectuer.

La figure 4.8 présente le tableau de bord spécifique à l'étudiant.

## Chapitre 4. Réalisation

**FIGURE 4.8 :** Interface «Gérer les tâches»

- **Interface «Ajouter une tâche »**

La figure 4.9 met en évidence le formulaire d'ajout d'une tâche, qui offre une interface conviviale et intuitive pour les utilisateurs. Ce formulaire permet à l'utilisateur de spécifier les détails de la tâche, tels que le titre, la description, la durée et autres informations pertinentes.

**FIGURE 4.9 :** Interface «Ajouter une tâche»

- Interface «Details d'une tâche»

La figure 4.10 met en évidence les détails d'une tâche, fournissant une vue complète et détaillée de toutes les informations associées à cette tâche spécifique.

**Details du tâche**

**Rédaction du cahier de charge**

**Description**  
Mentionner le contexte ,la problématique , les objectifs , les fonctionnalités , les technologies et la durée

**Membres**  
Zemzem Jihen

**DURÉE**  
5 Jours

**Fichiers**  
Plan.pdf

**Commentaires**  
Chiboub Ibrahim  
Ajoutez le cadre du projet ainsi que la présentation de l'organisme d'accueil !

FIGURE 4.10 : Interface «Details d'une tâche»

#### 4.2.5 Espace Encadrant

- Interface «Consulter projets»

La figure 4.11 met en évidence la liste des projets associés à l'encadrant .

<input type="checkbox"/>	Code	Nom	Etudiant 1	Etudiant 2	Diplôme	Filière
<input type="checkbox"/>	L3INFO	Plateforme de gestion de processus d'encadrement	Zemzem Jihen		Licence	INFO
<input type="checkbox"/>	L4INFO	Plateforme d'organisation de tournois des jeux vidéos	Nour saidane	Romthane Eslem	Licence	INFO
<input type="checkbox"/>	L6INFO	Application web e-commerce	Amen allah adem		Licence	INFO
<input type="checkbox"/>	LSINFO	Application Hotel wallet	Nabi wiem	Touka Ons	Licence	INFO

FIGURE 4.11 : Interface «Consulter projets» pour l'«encadrant»

- Interface «Suivre progrès » pour l'«encadrant»

Lorsque l'encadrant sélectionne un projet, le tableau de bord des tâches associé s'affiche, lui permettant de suivre les progrès de l'étudiant comme illustré dans la figure 4.12.

Tableau de bord kanban		
À FAIRE	EN COURS	TERMINÉ
<b>A faire</b> <b>Réalisation du chapitre 1</b> Réécrire le cahier de charge de manière plus organisée avec des photos et ajouter le diagramme de Gantt .	<b>En cours</b> <b>Rédaction du cahier de charge</b> Mentionner le contexte ,la problématique , les objectifs , les fonctionnalités , les technologies et la durée	<b>Finie</b> <b>Planification su projet</b> Réaliser le diagramme de Gantt pour la planification du projet
<b>A faire</b> <b>Réalisation du chapitre 2</b> Analyse et spécification des besoins	<b>En cours</b> <b>Rédaction d'introduction</b> Parler de l'idée général , cadre du projet et du contenu des chapitres de rapport	<b>Finie</b> <b>Formation individuelle</b> Apprendre les bases des technologies à utiliser
<b>A faire</b> <b>Rédaction de chapitre 4</b>	<b>En cours</b> <b></b> <b></b>	<b></b> <b></b> <b></b>

FIGURE 4.12 : Interface «Suivre progrès » pour l'«encadrant»

- Interface «Ajouter un commentaire»

L'encadrant peut également ajouter des commentaires et des retours d'information aux différentes tâches comme présenté dans la figure 4.13.

ASSIGNÉ AU
DURÉE

Zemzem Jihen
1 mois

**Description**  
Réécrire le cahier de charge de manière plus organisée avec des photos et ajouter le diagramme de Gantt .

**Contenu**  
[Plan.pdf](#)

**Commentaires**

Chiboub Ibrahim  
Bon travail Jihen !

6 Jun, 10:22

N'oubliez pas d'ajouter un tableau comparatif des méthodologies !

**Envoyer**

FIGURE 4.13 : Interface «Ajouter un commentaire»

### 4.3 Conclusion

Dans ce chapitre, nous avons présenté et exploré en détail les différentes interfaces de notre solution web. Chaque interface a été soigneusement conçue pour répondre à des besoins spécifiques et offrir une expérience utilisateur optimale.

# Conclusion générale

Ce projet de fin d'études a été consacré à la conception et au développement d'une plateforme d'organisation de processus d'encadrement destinée aux étudiants, aux encadrants et aux gestionnaires de stage. Notre objectif principal était de fournir une solution efficace et centralisée pour faciliter la gestion de projet de fin d'étude.

Pour mener à bien notre projet, nous avons détaillé les différentes étapes d'analyse, de conception et de réalisation de cette application. Dans le premier chapitre, nous avons abordé le cadre général du projet. Nous avons débuté par une étude approfondie de l'existant, identifiant les lacunes des solutions existantes et mettant en évidence l'adoption de la méthodologie de travail 2TUP pour la création de notre application web. Le deuxième chapitre s'est concentré sur la spécification des besoins fonctionnels, non fonctionnels et techniques. La conception de base de données et la conception détaillée ont été abordés dans le troisième chapitre. Une fois la phase de conception terminée, nous avons entamé la phase de réalisation, qui a été traitée dans le quatrième chapitre.

Au terme de ce projet, nous sommes fiers d'avoir créé une plateforme qui répond aux besoins des utilisateurs en matière d'organisation et de suivi des processus d'encadrement. Cette plateforme offre une interface intuitive, des fonctionnalités puissantes et une expérience utilisateur optimale. Cependant, il convient de souligner que ce projet n'est pas une solution définitive. Il existe toujours des opportunités d'amélioration et d'expansion de la plateforme. Par exemple, nous pourrions envisager d'intégrer des fonctionnalités supplémentaires telles que la gestion des calendriers, la collaboration en temps réel via un système de messagerie et l'intelligence artificielle pour l'analyse des données.

En conclusion, ce projet a été une expérience enrichissante qui nous a permis d'appliquer nos connaissances théoriques à un cas concret, de développer nos compétences techniques et de mieux comprendre les enjeux de la réalisation d'un projet concret. Nous sommes fiers du résultat obtenu et de l'impact potentiel de cette plateforme sur la résolution des problèmes de nos utilisateurs.

# Bibliographie

- [1] « Moodle. » Consulté le 5-Mars-2023. (), adresse : <https://moodle.com/fr/faq/question-moodle>.
- [2] « Thinkific. » Consulté le 5-Mars-2023. (), adresse : <https://www.thinkific.com/>.
- [3] « Mentornity. » Consulté le 5-Mars-2023. (), adresse : <https://www.softwareadvice.fr/software/88738/mentornity>.
- [4] « Wrike. » Consulté le 5-Mars-2023. (), adresse : <https://www.websiterating.com/fr/productivity/best-trello-alternatives>.
- [5] « Comparaison de méthodologie de travail. » Réalisé par Sarra Ben Haddada et Skander Bouraoui en 2020/2021. (), adresse : [Rapport\\_Skander\\_Sarra.pdf](#).
- [6] « Cycle de vie 2TUP. » Consulté le 10-Mars-2023. (), adresse : <https://www.memoireonline.com/>.
- [7] « 2TUP. » Mme : Maissa Hamouda, Matière : Entreprise System Architecture, Année : L3 Info. (), adresse : [Architecture-des-systemes-dentreprise-ESA-Tirage-part-1](#).
- [8] « MongoDB. » Consulté le 20-Mai-2023. (), adresse : <https://www.alamyimages.fr/mongodb-logo-fond-blanc-image>.
- [9] « ExpressJS. » Consulté le 20-Mai-2023. (), adresse : <https://www.sohamkamani.com/nodejs/expressjs-architecture/>.
- [10] « ReactJS. » Consulté le 20-Mai-2023. (), adresse : <https://ubidreams.fr/expertises/development/react-js>.
- [11] « Nodejs. » Consulté le 20-Mai-2023. (), adresse : <https://litslink.com/blog/what-you-should-know-about-the-future-of-nodejs>.
- [12] « Vscode. » Consulté le 20-Mai-2023. (), adresse : <https://www.pngwing.com/en/search?q=react>.
- [13] « Draw.io. » Consulté le 20-Mai-2023. (), adresse : [https://commons.wikimedia.org/wiki/File:Diagrams.net\\_Logo.svg](https://commons.wikimedia.org/wiki/File:Diagrams.net_Logo.svg).
- [14] « Github. » Consulté le 20-Mai-2023. (), adresse : <https://github.com/logos>.
- [15] « Canva. » Consulté le 20-Mai-2023. (), adresse : <https://digitiz.fr/blog/canva-pro/>.

## Bibliographie

---

- [16] « MVC. » Consulté le 25-Mai-2023. (), adresse : <https://openclassrooms.com/fr/courses/4670706-adoptez-une-architecture-mvc-en-php/>.
- [17] « Architecture 3 tiers. » Consulté le 25-Mai-2023. (), adresse : <https://www.javatpoint.com/mern-stack>.
- [18] « UML. » Mme :Asma kerkenni, Matière : Conception, Année : L2 INFO. (), adresse : [Cours/1\\_ch1\\_intro\\_GL\\_4](#).
- [19] « MCD. » Réalisé par Mejri Siwar et Chennaoui Anis en 2020/2021. (), adresse : [LF12.pdf](#).
- [20] « Model orienté document. » Mme : Asma kerkenni, Matière : Big data, Année : L3 INFO. (), adresse : [CoursmongoDBpart1](#).

