



République Tunisienne

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Monastir

Institut Supérieur d'Informatique et de Mathématiques de Monastir



N° d'ordre : L91INFO

Mémoire de Projet de Fin d'Etudes

Présenté en vue de l'obtention du

Diplôme National de Licence en Sciences Informatique

Spécialité :

Génie Logiciel et Système d'Information

Par :

ZEMZEM Jihen

Mise en place d'une plateforme d'organisation de processus d'encadrement

Soutenu le xxxxxxxx devant le jury composé de :

M./Mme :

M./Mme :

M. : BEN SALEM Malek

M. : KHATIB Yassine

Président

Rapporteur

Encadrant Pédagogique

Encadrant Professionnel

_____ Année Universitaire : 2022 / 2023 _____



DEDICACES

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 chers 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



REMERCIEMENT

Au terme de mon travail de fin d'étude, je tiens à adresser mes vifs remerciements et mes sincères gratitudes 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.

Je tiens à remercier également les membres du jury, qui me font un grand honneur en acceptant d'évaluer ce modeste travail.

Je souhaite adresser des remerciements spéciaux à mes parents, mes frères, sœurs et mes amis pour le soutien qu'ils m'ont témoigné durant toute cette période.

Un spécial merci pour ma chère meilleur amie Amira que j'aime énormément pour son soutien inconditionnel, ses conseils précieux et son amitié sincère.

Enfin, que toute personne ayant contribué de près ou de loin à la réussite de ce travail trouve ici l'expression de ma reconnaissance.

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 (ISIMM). 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 et les étudiants. Cette plateforme offrira des outils avancés pour le partage de documents, la validation des projets, ainsi que le suivi régulier et efficace de l'avancement, afin de garantir que les étudiants bénéficient d'un suivi régulier et efficace facilitant leur progression.

Mots clés : Processus encadrement, Gestion projet, ReactJS, ExpressJS, NodeJs, MongoDB

Abstract

This report has been written as part of a graduation project for the National Bachelor's Degree in Computer Science program at the Higher Institute of Computer Science and Mathematics of Monastir (ISIMM). The goal of this project, carried out within IMOTION+ startup, is to design and develop an innovative platform for organizing the mentoring process for supervisors and interns, with the aim of significantly improving collaboration and work monitoring during the internship. This platform will provide advanced tools for document sharing, project validation, as well as regular and effective progress tracking, ensuring that students benefit from consistent and efficient support to facilitate their progress.

Keywords : Mentoring process, Project management, ReactJS, ExpressJS, NodeJs, MongoDB



TABLE DES MATIÈRES

LISTE DES FIGURES	viii
LISTE DES ABRÉVIATIONS	ix
INTRODUCTION GÉNÉRALE	x
1 Contexte général et étude préalable	2
1.1 INTRODUCTION	3
1.2 Présentation de l'organisme d'accueil	3
1.3 Présentation du projet	4
1.3.1 Cadre du projet	4
1.3.2 Sujet et problématique	4
1.3.3 Objectif du projet	4
1.4 Étude préalable	5
1.4.1 Etude des solutions existantes	5
1.4.2 Critique de l'existant	7
1.5 Solution proposée	8
1.6 Méthode de gestion du projet	8
1.6.1 Choix de la méthode utilisée	8
1.6.2 Gestion de projet avec la méthode 2TUP	9
1.6.3 Planification du projet	11
1.7 CONCLUSION	11
2 Présentation du branche fonctionnelle	12
2.1 INTRODUCTION	13
2.2 Spécification des besoins non fonctionnels	13
2.3 Spécification des besoins fonctionnels	14
2.3.1 Identification des acteurs	14
2.3.2 Identification des besoins fonctionnels par acteur	14

2.3.3	Diagramme général des cas d'utilisation	16
2.3.3.1	Diagramme qui concerne les cas d'utilisation de l'utilisateur .	18
2.3.3.2	Diagramme qui concerne les cas d'utilisation de l'étudiant . .	18
2.3.3.3	Diagramme qui concerne les cas d'utilisation de l'encadrant .	19
2.3.3.4	Diagramme qui concerne les cas d'utilisation de gestionnaire de stage	19
2.3.3.5	Diagramme qui concerne les cas d'utilisation de l'administrateur	20
2.4	CONCLUSION	21
3	CHAPITRE 3 : Présentation du branche technique	22
3.1	INTRODUCTION	23
3.2	Capture des besoins techniques	23
3.2.1	Environnement de développement	23
3.2.2	Environnement logiciel	24
3.3	Architecture de l'application	28
3.3.1	Architecture logique	28
3.3.2	Architecture physique	30
3.4	CONCLUSION	31
4	CHAPITRE 4 : Conception et Réalisation de l'application	33
4.1	INTRODUCTION	34
4.2	Langage de modélisation utilisée	34
4.3	Conception des besoins	35
4.3.1	Diagramme de classe	35
4.3.2	Diagrammes de séquence objet	36
4.3.2.1	Diagramme de séquence objet "s'authentifier"	36
4.3.2.2	Diagramme de séquence détaillé "Ajouter une tâche "	37
4.3.2.3	Diagramme de séquence objet "Ajouter un commentaire" . .	37
4.3.2.4	Diagramme de séquence détaillé "Valider un projet"	38
4.3.3	Diagrammes d'activités	39
4.3.3.1	Diagramme d'activité "Ajouter un commentaire "	40
4.3.4	Diagramme d'état-transition de changer l'état d'une tâche	40
4.4	Conception de la base de données	41
4.4.1	Modèle Logique des données :	42
4.5	Implémentation technique de la solution	42
4.5.1	Page d'accueil	42
4.5.2	Page d'inscription	42

TABLE DES MATIÈRES

4.6	Test et validation de la solution	44
4.7	Conclusion	44
CONCLUSION GÉNÉRALE		45
BIBLIOGRAPHIE		45
ANNEXES		47
A.1	TITRE ANNEXE 1	47
A.2	TITRE ANNEXE 2	47

LISTE DES FIGURES

1.1	Logo IMOTION+	3
1.2	Interface d'accueil de Moodle	5
1.3	Interface d'accueil de Thinkific	6
1.4	Interface d'accueil de Mentornity	6
1.5	Cycle de vie de 2TUP	9
1.6	Diagramme de Gantt	11
2.1	Diagramme général de cas d'utilisation	17
2.2	Diagramme de cas d'utilisation de l'utilisateur	18
2.3	Diagramme de cas d'utilisation de l'étudiant	18
2.4	Diagramme de cas d'utilisation de l'encadrant	19
2.5	Diagramme de cas d'utilisation de gestionnaire de stage	20
2.6	Diagramme de cas d'utilisation de l'administrateur	20
3.1	Les composants de MERN stack	23
3.2	Logo mongoDB	25
3.3	Logo express Js	25
3.4	Logo react Js	26
3.5	Logo node Js	26
3.6	Logo vscode.jpg	26
3.7	Logo draw.io	27
3.8	Logo github	27
3.9	Logo canva	27
3.10	Architecture MVC	29
3.11	Le client et l'architecture MVC	30
3.12	Architecture 3 tiers avec MERN	31
4.1	Diagramme de classe général	35
4.2	Diagramme de séquence objet d'authentification	36
4.3	Diagramme de séquence détaillé d'ajout d'une tâche	37

4.4	Diagramme de séquence détaillé d'ajout d'un commentaire	38
4.5	Diagramme de séquence détaillé de validation de projet	39
4.6	Diagramme d'activité d'ajout d'un commentaire	40
4.7	Diagramme d'état-transition de changer état	41
4.8	Interface d'accueil	42
4.9	Formulaire d'inscription pour l'étudiant	43
4.10	Formulaire d'inscription pour l'encadrant et le gestionnaire	43



LISTE DES ABRÉVIATIONS

API : Application Programming Interface

DTO : Data Transfer Object

HTML : HyperText Markup Language

JSON : JavaScript Object Notation

MERN : MongoDB, Express, React, Node

MVC : Model-View-Controller

ONG : Organisation non gouvernementale

PHP : Hypertext Preprocessor

REST : Representational state transfer

SPA : Single Page Applications

2TUP : 2 Track Unified Process

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 créé 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, évaluer les résultats et planifier les séances d'encadrement, 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 ainsi 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 introduction du contexte général et de l'étude préalable du projet, tandis que le deuxième chapitre se concentre sur l'étude des besoins fonctionnels et non fonctionnels. Le troisième chapitre aborde l'architecture utilisée et les besoins techniques, tandis que le quatrième chapitre présente la conception et 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.

En conclusion, notre plateforme d'organisation de processus d'encadrement est une solution innovante pour faciliter la gestion des travaux de fin d'études. Nous espérons que ce rapport sera utile pour les encadrants et les étudiants qui cherchent à améliorer leur processus d'encadrement et à obtenir des résultats plus efficaces et plus satisfaisants.

Contexte général et étude préalable

Sommaire

1.1	INTRODUCTION	3
1.2	Présentation de l'organisme d'accueil	3
1.3	Présentation du projet	4
1.3.1	Cadre du projet	4
1.3.2	Sujet et problématique	4
1.3.3	Objectif du projet	4
1.4	Étude préalable	5
1.4.1	Etude des solutions existantes	5
1.4.2	Critique de l'existant	7
1.5	Solution proposée	8
1.6	Méthode de gestion du projet	8
1.6.1	Choix de la méthode utilisée	8
1.6.2	Gestion de projet avec la méthode 2TUP	9
1.6.3	Planification du projet	11
1.7	CONCLUSION	11

1.1 INTRODUCTION

Le présent chapitre a pour objectif de contextualiser notre projet. Dans un premier temps, nous présentons l'organisme d'accueil. Ensuite, nous abordons le contexte global du projet en exposant la problématique à laquelle nous faisons face ainsi que les solutions actuellement disponibles sur le marché. Par la suite nous exposons notre solution proposée. Enfin, nous décrivons la méthodologie de travail que nous avons appliquée.

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. IMotion+ dispose d'outils technologiques de développement informatique, de conception et de gestion de communauté qui accélèrent leur impact.

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



FIGURE 1.1 – Logo IMOTION+

1.3 Présentation 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). Le stage a été effectué au siège de « IMOTION+ » pendant deux mois .

1.3.2 Sujet et problématique

La gestion des processus d'encadrement peut être une tâche complexe pour les encadrants, les étudiants et le gestionnaire de stage avec des défis tels que la suggestion de sujets, la coordination des calendriers d'encadrement ,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, telles que les plateformes d'organisation de processus d'encadrement, 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 et des stagiaires, tout en garantissant la sécurité et la confidentialité des données, pour permettre une gestion optimale des processus d'encadrement.

1.3.3 Objectif du projet

Notre projet vise à :

- Mettre en place une plateforme d'organisation de processus d'encadrement efficace qui facilite le suivi régulier des étudiants, permet la réservation des dates d'encadrement, ainsi que le dépôt des documents nécessaires à la validation du stage et la planification des soutenances.

- Élaborer une plateforme conviviale et facile à utiliser pour les encadrants, les étudiants et le gestionnaire de stage, avec une interface utilisateur intuitive et une navigation fluide.
- Garantir la sécurité et la confidentialité des données en concevant une plateforme qui respecte les normes de sécurité et de confidentialité des données.

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 des applications existantes.

1.4.1 Etude des solutions existantes

Moodle :[1] est une plateforme d'apprentissage en ligne libre distribuée sous la Licence publique générale GNU écrite en PHP. Développée à partir de principes pédagogiques, elle permet aux enseignants de créer des cours en ligne, de communiquer avec les étudiants et de suivre les progrès. La figure 1.2 présente l'interface d'accueil de l'outil Moodle.



FIGURE 1.2 – Interface d'accueil de Moodle

Thinkific : [2] est une plateforme de formation en ligne complète permettant à tous 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.

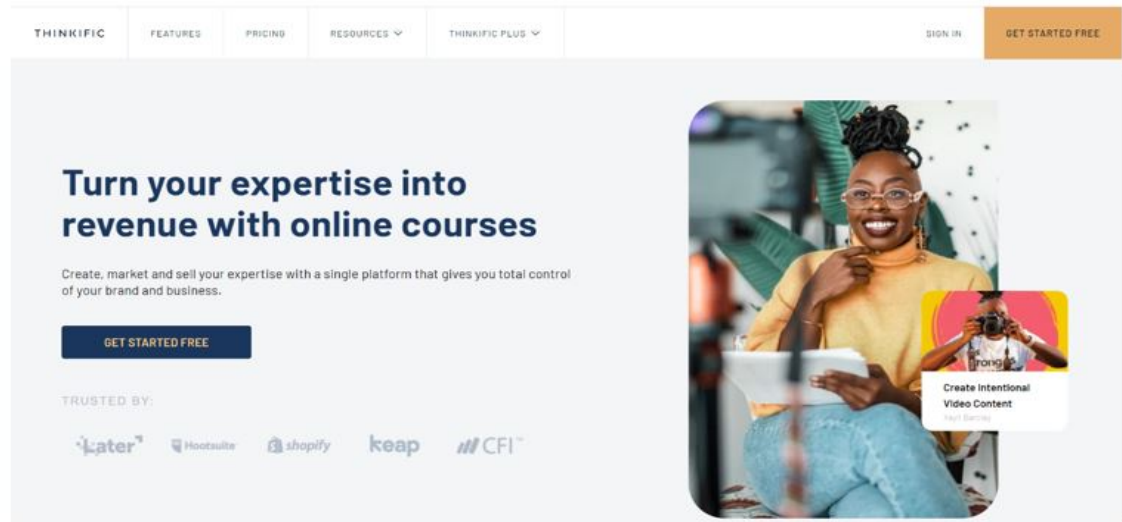


FIGURE 1.3 – Interface d'accueil de Thinkific

Mentornity :[3] 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 ci-dessous représente l'interface d'accueil de l'outil Mentornity.

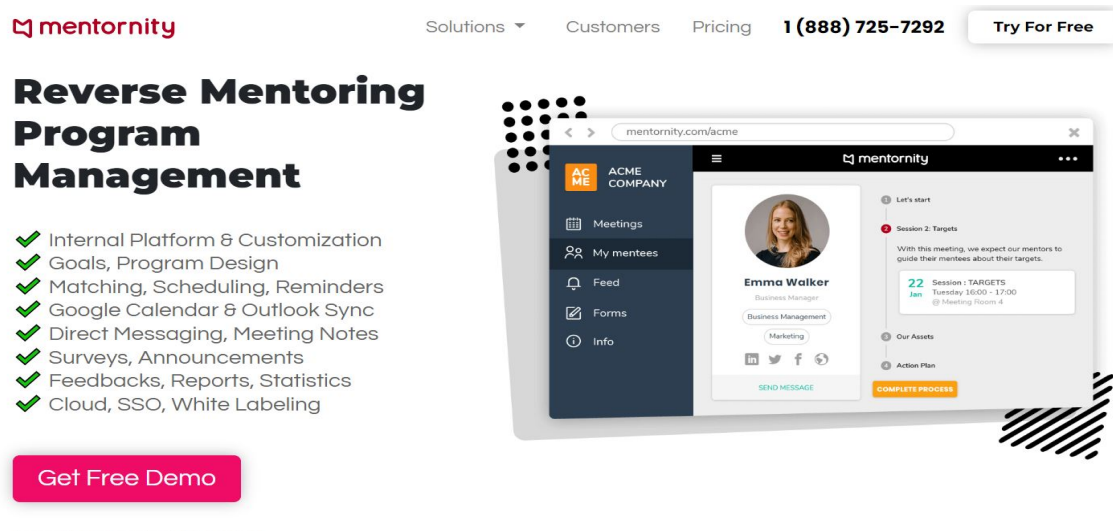


FIGURE 1.4 – Interface d'accueil de Mentornity

1.4.2 Critique de l'existant

D'après les données présentées dans le tableau ci-dessous, les options actuellement disponibles pour les plateformes d'apprentissage et supervision en ligne offrent diverses fonctionnalités fondamentales (telles que la communication audio-vidéo, le chat, le partage de fichiers, le suivi quotidien des étudiants ...). Toutefois, elles comportent des inconvénients tels que la complexité d'utilisation et de configuration et manque de quelques fonctionnalités d'encadrement. Le Tableau 1.1 présente bien une comparaison entre ces trois solutions.

TABLE 1.1 – Comparaison entre les solutions existantes

Outils	Avantages	Inconvénients
Moodle	<ul style="list-style-type: none"> — Open source et donc gratuit — Fonctionnalités avancées de gestion des cours et de suivi des étudiants — Système de messagerie intégré — Gestion de projets avec un tableau Kanban 	<ul style="list-style-type: none"> — Interface utilisateur semble complexe pour les débutants — Configuration de la plate-forme peut être difficile
Thinkific	<ul style="list-style-type: none"> — Interface utilisateur intuitive et facile à utiliser — Fonctionnalités de mentorat et de coaching — Suivi des apprenants et de commentaires 	<ul style="list-style-type: none"> — 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	<ul style="list-style-type: none"> — Interface utilisateur intuitive et facile à utiliser — Possibilité de suivre l'avancement des apprenants et de leur fournir des commentaires — Fonctionnalité de messagerie intégrée 	<ul style="list-style-type: none"> — Moins de fonctionnalités avancées — Coût élevé par rapport à certaines autres solutions

1.5 Solution proposée

Les plateformes actuelles ont été initialement développées pour l'apprentissage et la formation, offrant des services de supervision et d'encadrement. Cependant, notre plateforme se concentrera exclusivement sur l'organisation du processus d'encadrement entre le superviseur et ses stagiaires. Après avoir étudié les plateformes existantes, nous avons identifié plusieurs anomalies que nous avons détaillées dans la section précédente. Pour remédier à ces anomalies, nous proposons la conception et l'implémentation d'une plateforme nommé "ENCADRA" dédiée à l'organisation du processus d'encadrement. Cette solution envisage :

- **Une interface utilisateur facile à utiliser et conviviale** peut contribuer à simplifier la présentation des sujets, optimiser la planification des séances d'encadrement et favoriser le dépôt des documents nécessaires et la planification des soutenances.
- **Surveillance avancée des étudiants** : La récente plateforme pourrait intégrer des caractéristiques évoluées pour contrôler le progrès des apprentis, donner des critiques et évaluer leurs avancements.
- **Facilité d'utilisation et de mise en place** : Cette plateforme doit être facile à utiliser et à mettre en place pour les étudiants, les enseignants, le gestionnaire de stage, le directeur de stage et les administrateurs afin de minimiser les problèmes techniques et les retards.

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 déjà réalisé au sein de l'institut [3] 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. Ce processus se base lui-même sur le Processus Unifié qui est devenu un standard général réunissant les meilleures pratiques de développement. Cette méthode ne se base aucunement sur un processus linéaire mais bien, sur un développement itératif et incrémental.

La figure 1.5 illustre les différentes phases de développement

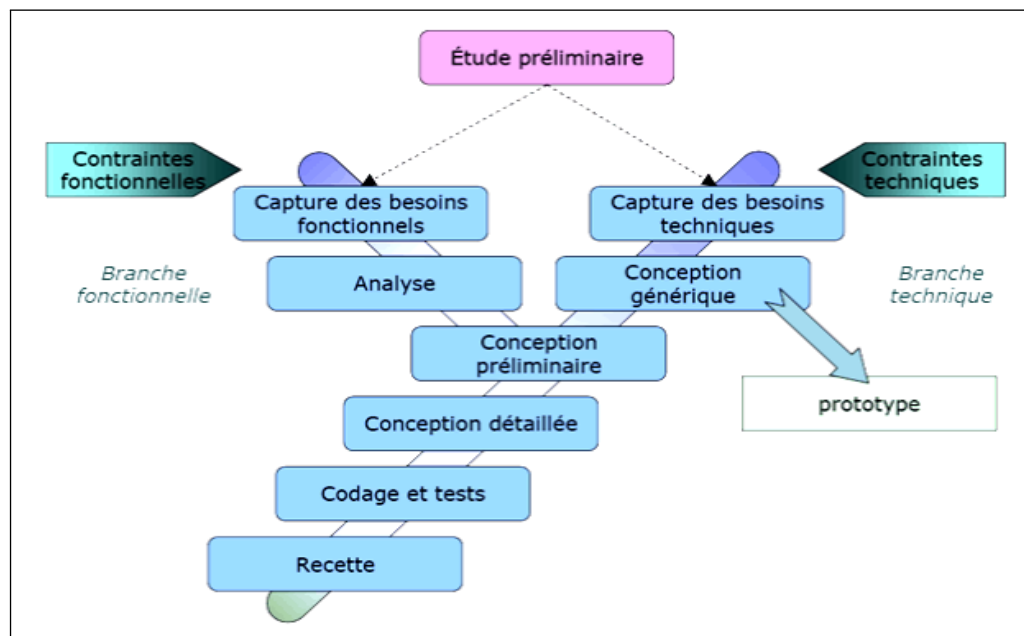


FIGURE 1.5 – Cycle de vie de 2TUP

1.6.2 Gestion de projet avec la méthode 2TUP

Le 2TUP [4] 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. Le processus se divise ensuite en trois phases essentielles :

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

- **Capture des besoins fonctionnels**, qui produit le modèle des besoins focalisés sur le métier des utilisateurs. Elle qualifie, au plus tôt le risque de produire un système inadapté aux utilisateurs.
- **L'analyse**, qui 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**, qui recense toutes les contraintes sur les choix de dimensionnant et la conception du système. Il s'agit de la spécification des outils (logiciels), de la structure des matériels à exploiter ainsi que la prise en compte des contraintes d'intégration avec l'existant (pré requis d'architecture technique).
- **La conception générique**, qui 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**, qui 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**, qui étudie ensuite comment réaliser chaque composant.
- **L'étape de codage**, qui produit ses composants et tests au fur et à mesure les unités de code réalisées.
- **L'étape de recette**, qui consiste enfin à valider les fonctionnalités du système développé

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 1.6 présente le diagramme de GANTT de notre projet.

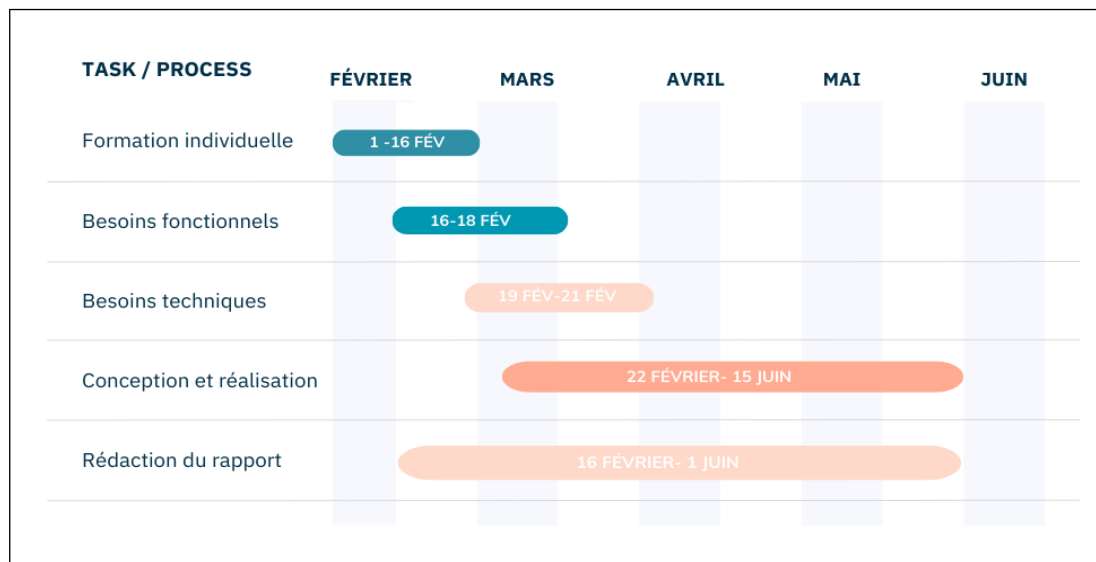


FIGURE 1.6 – Diagramme de Gantt

1.7 CONCLUSION

Dans cette première partie, nous avons introduit l'organisme d'accueil et présenté les objectifs et la problématique du projet à réaliser. Nous avons également examiné les solutions existantes, choisi la méthodologie de gestion de projet à adopter, et élaboré un diagramme de Gantt pour la planification. Le prochain chapitre consiste en la présentation de la branche fonctionnelle du projet.

Présentation du branche fonctionnelle

Sommaire

2.1	INTRODUCTION	13
2.2	Spécification des besoins non fonctionnels	13
2.3	Spécification des besoins fonctionnels	14
2.3.1	Identification des acteurs	14
2.3.2	Identification des besoins fonctionnels par acteur	14
2.3.3	Diagramme général des cas d'utilisation	16
2.4	CONCLUSION	21

2.1 INTRODUCTION

L'utilisation de la méthodologie 2TUP a mené à l'élaboration du présent chapitre, qui se focalisera sur la branche gauche, c'est-à-dire l'identification des acteurs, la collecte des besoins fonctionnels, ainsi que la compréhension des exigences énoncées par les besoins non fonctionnels, dans le but d'identifier toutes les fonctionnalités requises pour le futur système.

2.2 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 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 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.

2.3 Spécification des besoins fonctionnels

2.3.1 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 cinq 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 fiches et documents nécessaires pour les étudiants.
- **Administrateur de la plateforme** : Cette personne est responsable de la gestion des utilisateurs de la plateforme et de la garantie de son bon fonctionnement.

2.3.2 Identification des besoins fonctionnels par acteur

Le tableau 2.1 présente les besoins fonctionnels pour chaque acteur de notre application.

TABLEAU 2.1 – Spécification des besoins fonctionnels

Acteurs	Besoins fonctionnels
Étudiant	<ul style="list-style-type: none"> • S’inscrire à la plateforme • S’authentifier pour accéder à son espace • Gérer son profil • Gérer ses tâches • Consulter les fichiers diffusés • Envoyer des fichiers • Consulter les dates des séances d’encadrement • Consulter les dates de dépôt des rapports • Consulter les dates des soutenances
Encadrant	<ul style="list-style-type: none"> • S’inscrire à la plateforme • S’authentifier pour accéder à son espace • Gérer son profil • Gérer ses projets de stage • Planifier des séances d’encadrement • Consulter les dates des soutenances
Gestionnaire de stage	<ul style="list-style-type: none"> • S’inscrire à la plateforme • S’authentifier pour accéder à son espace • Gérer son profil • Déposer les fichiers de stage • Consulter les fichiers reçus • Valider les projets des étudiants • suivre l’avancement des étudiants • Consulter la liste des encadrants et étudiants • Publier les dates des dépôt des rapports et soutenances

Administrateur	<ul style="list-style-type: none">• S'authentifier pour accéder à son espace• Gérer son profil• Gérer les utilisateurs• Gérer les rôles des utilisateurs
-----------------------	---

2.3.3 Diagramme général des cas d'utilisation

La Figure 2.1 représentée illustre le diagramme général des cas d'utilisation pour donner une vision globale du comportement fonctionnel d'un système logiciel. La dernière version de l'UML est la version 2.5.1, publiée en 2017 a apporté plusieurs améliorations et clarifications par rapport aux versions précédentes, on compte la présentation de l'authentification comme une cas d'utilisation ajoutée seulement à l'utilisateur dans notre cas sans l'inclusion de relations supplémentaires telles que «include»

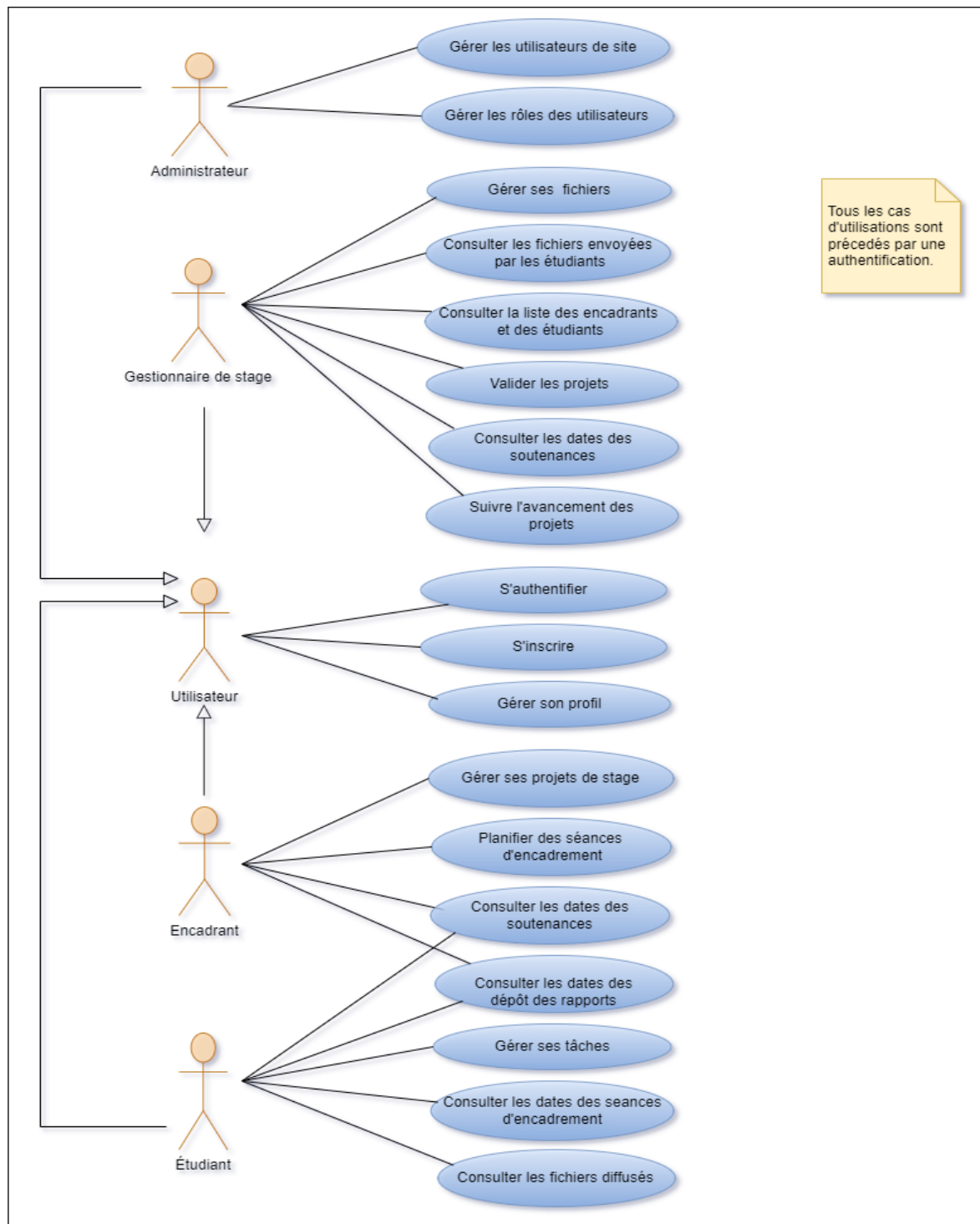


FIGURE 2.1 – Diagramme général de cas d'utilisation

Pour simplifier le diagramme de cas d'utilisation, nous avons procédé à sa décomposition en quatre diagrammes :

2.3.3.1 Diagramme qui concerne les cas d'utilisation de l'utilisateur

La Figure 2.2 présente le diagramme des cas d'utilisation détaillés pour l'utilisateur, décrivant les différentes fonctionnalités accessibles pour lui dans le système.

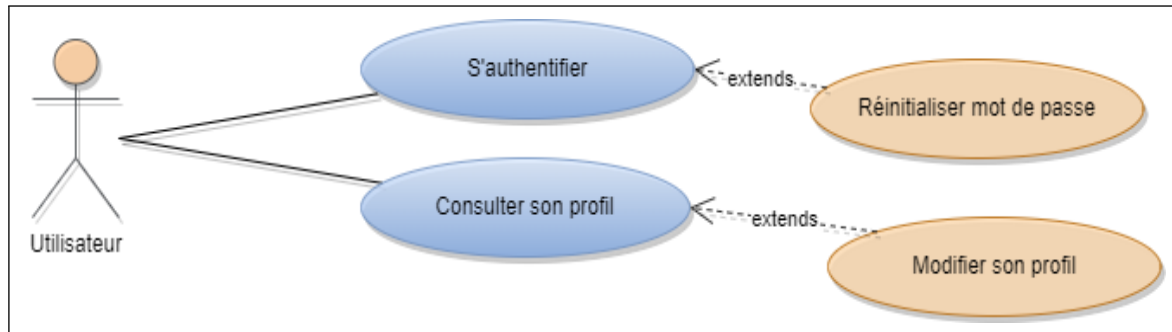


FIGURE 2.2 – Diagramme de cas d'utilisation de l'utilisateur

2.3.3.2 Diagramme qui concerne les cas d'utilisation de l'étudiant

La Figure 2.3 présente le diagramme des cas d'utilisation détaillé pour l'étudiant, décrivant les différentes fonctionnalités accessibles pour lui dans le système.

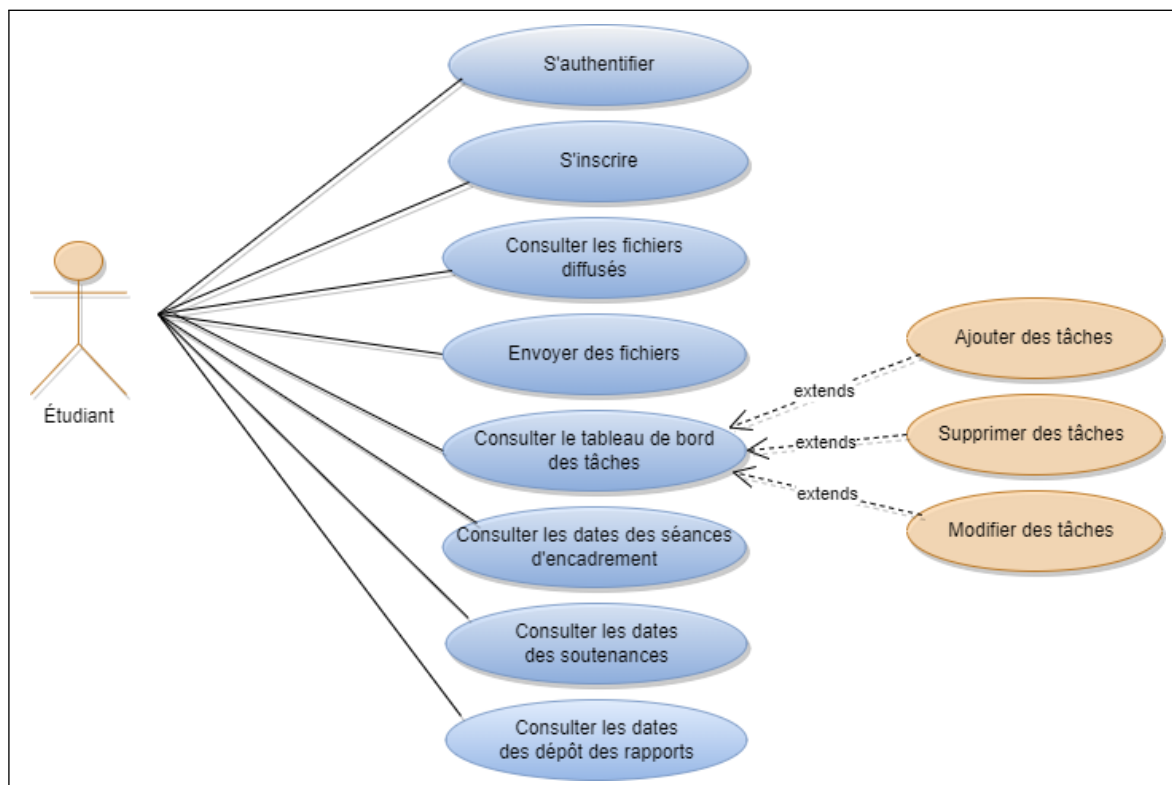


FIGURE 2.3 – Diagramme de cas d'utilisation de l'étudiant

2.3.3.3 Diagramme qui concerne les cas d'utilisation de l'encadrant

La Figure 2.4 présente le diagramme des cas d'utilisation détaillé pour l'encadrant, décrivant les différentes fonctionnalités accessibles pour lui dans le système.

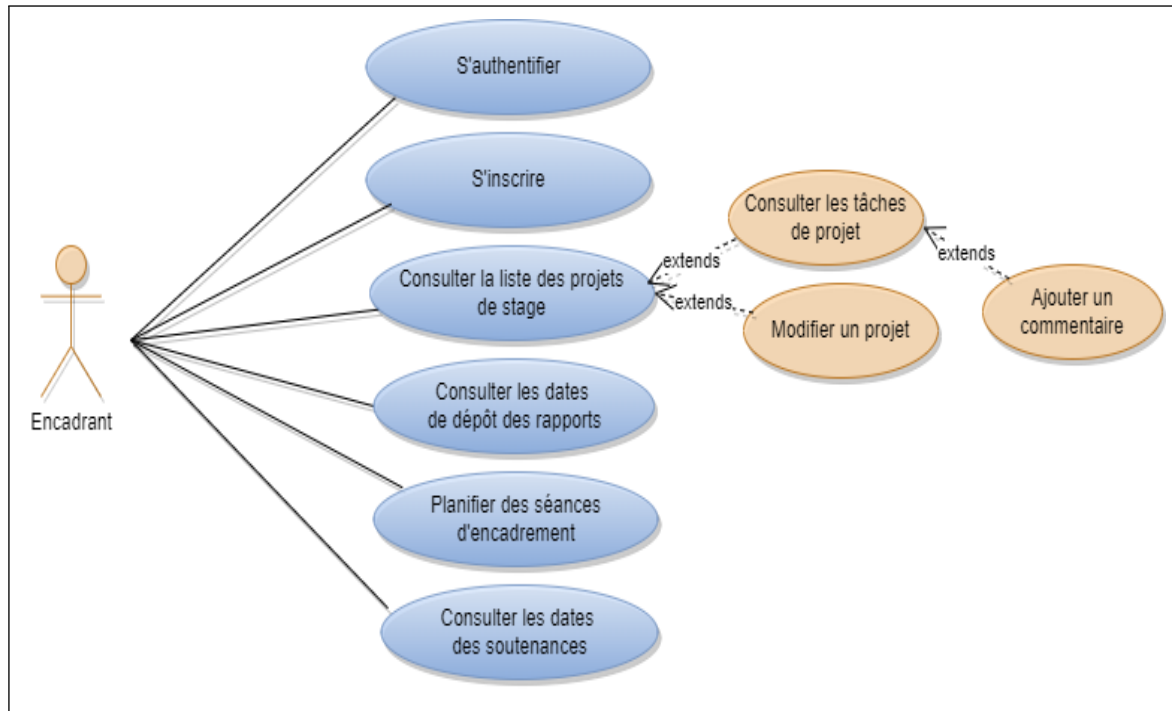


FIGURE 2.4 – Diagramme de cas d'utilisation de l'encadrant

2.3.3.4 Diagramme qui concerne les cas d'utilisation de gestionnaire de stage

La Figure 2.5 présente le diagramme des cas d'utilisation détaillé pour le gestionnaire de stage, décrivant les différentes fonctionnalités accessibles pour lui dans le système.

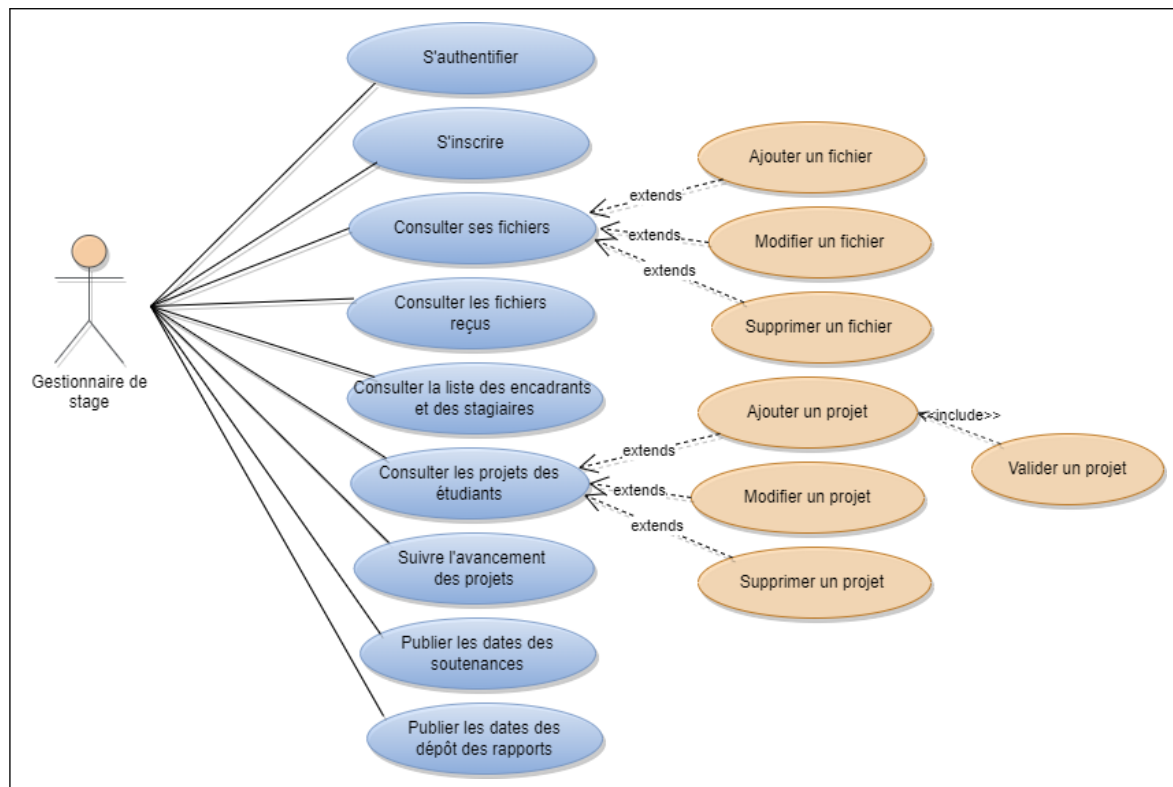


FIGURE 2.5 – Diagramme de cas d'utilisation de gestionnaire de stage

2.3.3.5 Diagramme qui concerne les cas d'utilisation de l'administrateur

La Figure 2.6 présente le diagramme des cas d'utilisation détaillé pour l'administrateur, décrivant les différentes fonctionnalités accessibles pour lui dans le système.

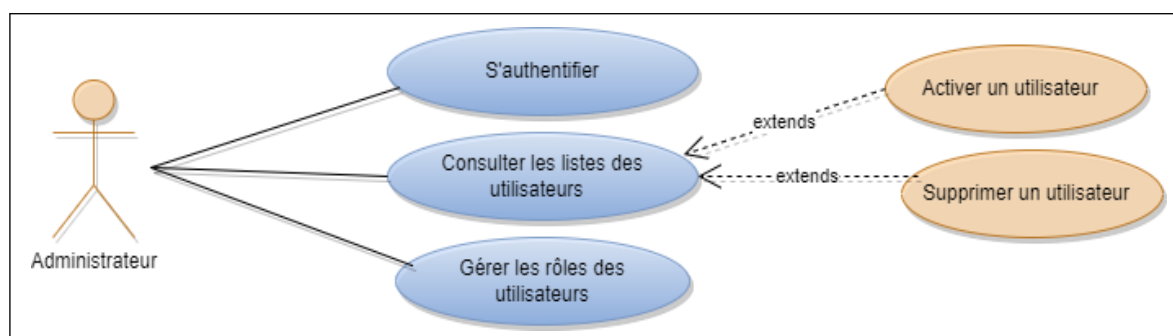


FIGURE 2.6 – Diagramme de cas d'utilisation de l'administrateur

2.4 CONCLUSION

Dans ce chapitre, nous avons défini en détail les besoins fonctionnels et non fonctionnels de notre projet. Ensuite, nous avons effectué une analyse approfondie de ces besoins en utilisant des diagrammes de cas d'utilisation à la fois généraux et détaillés. Dans le prochain chapitre, nous aborderons la partie technique de notre projet, où nous traiterons des aspects liés à la mise en œuvre technique de notre solution.

CHAPITRE 3 : Présentation du branche technique

Sommaire

3.1	INTRODUCTION	23
3.2	Capture des besoins techniques	23
3.2.1	Environnement de développement	23
3.2.2	Environnement logiciel	24
3.3	Architecture de l'application	28
3.3.1	Architecture logique	28
3.3.2	Architecture physique	30
3.4	CONCLUSION	31

3.1 INTRODUCTION

Dans ce chapitre, nous abordons la phase de "Capture des besoins techniques" de la branche droite du cycle en Y. Cette étape vise à capitaliser sur notre savoir-faire technique et à définir l'architecture logique et physique utilisée dans notre projet. Nous allons également aborder les technologies utilisés pour la réalisation de notre solution.

3.2 Capture des besoins techniques

3.2.1 Environnement de développement

La création d'une application internet 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 application, 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.

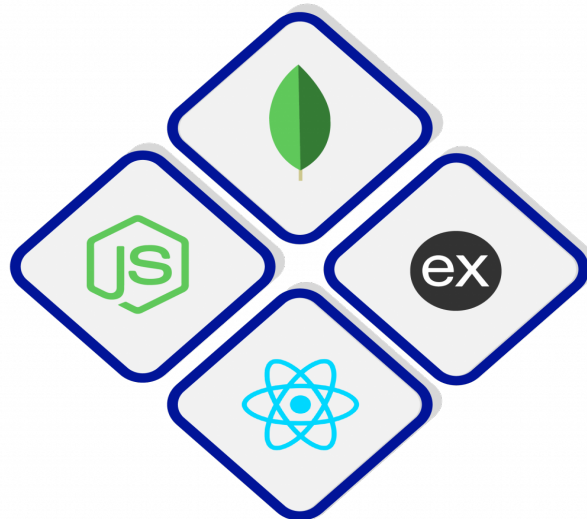


FIGURE 3.1 – Les composants de MERN stack

Le terme MERN signifie MongoDB, Express, React, Node, d'après les quatre technologies clés qui composent la pile.

- **MongoDB** - base de données de documents,
- **Express(.js)** - Framework Web Node.js,
- **Node(.js)** - le premier serveur Web JavaScript,
- **React(.js)** - un framework JavaScript côté client.

Express et Node constituent le niveau intermédiaire (application). Express.js est un framework Web coté serveur et Node.js la plate-forme de serveur JavaScript populaire et puissante. En combinant ces technologies, nous avons pu créer un portail web efficace et moderne.

Voici les multiples 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.
- + Performance et sécurité grâce à l'utilisation de JavaScript pour les deux côtés du développement.
- + Amélioration de la qualité du code grâce à l'utilisation d'API RESTful et d'objets de transfert de données (DTO).
- + Collaboration facile pour permettre à plusieurs développeurs de travailler sur le même projet.
- + Architecture à base de code unique pour simplifier le processus de développement.

3.2.2 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 3.2 – Logo mongoDB

- **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 3.3 – Logo express Js

- **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.

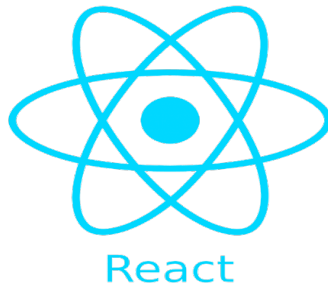


FIGURE 3.4 – Logo react Js

- **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.

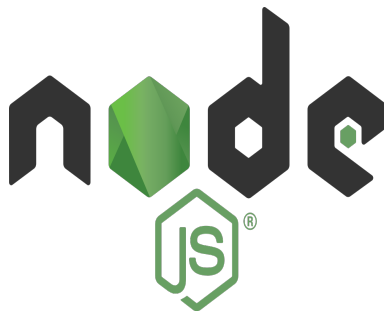


FIGURE 3.5 – Logo node Js

- **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.

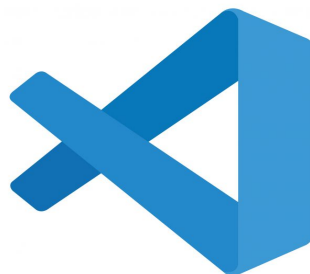


FIGURE 3.6 – Logo vscode.jpg

- **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.

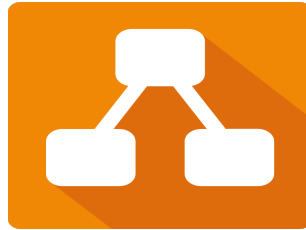


FIGURE 3.7 – Logo draw.io

- **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 3.8 – Logo github

- **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 3.9 – Logo canva

3.3 Architecture de l'application

Lors de la conception d'un système logiciel, il est important de choisir des modèles architecturaux logiques et physiques qui s'y adaptent judicieusement, ce qui assurera un bon fonctionnement entre les différentes couches. Cette démarche garantit un fonctionnement optimal entre les différentes couches du système.

3.3.1 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. Son objectif est de fournir une interface d'action la plus simple possible au contrôleur. On y trouve donc entre autres des algorithmes complexes et des requêtes SQL.
- **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 mais aussi quelques boucles et conditions PHP très simples, pour afficher par exemple une liste de messages.
- **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 va recevoir des requêtes de l'utilisateur. Pour chacune, il va demander au modèle d'effectuer certaines

actions (lire des articles de blog depuis une base de données, supprimer un commentaire) et de lui renvoyer les résultats (la liste des articles, si la suppression est réussie). Puis il va adapter ce résultat et le donner à la vue. Enfin, il va renvoyer la nouvelle page HTML, générée par la vue, à l'utilisateur.

La figure suivante schématise le rôle de chacun de ces éléments.

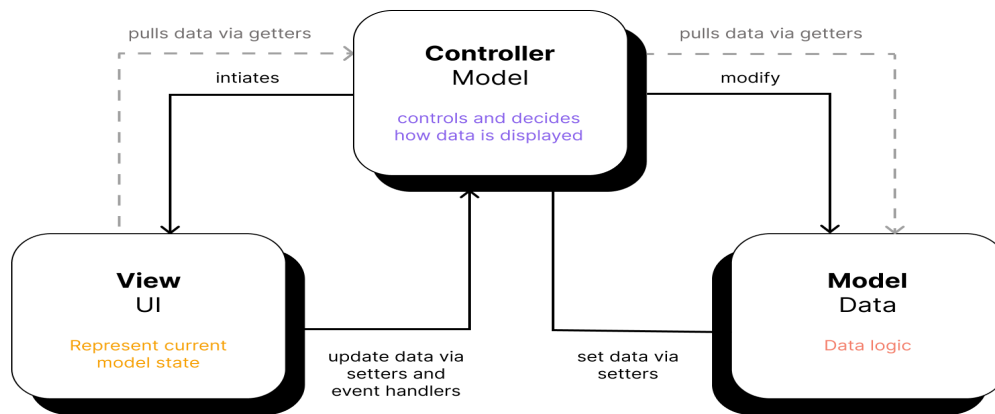


FIGURE 3.10 – Architecture MVC

Concrètement, le visiteur demandera la page au contrôleur et c'est la vue qui lui sera retournée, comme schématisé sur la figure suivante. Bien entendu, tout cela est transparent pour lui, il ne voit pas tout ce qui se passe sur le serveur. C'est un schéma plus complexe que ce à quoi nous avons été habitués, bien évidemment : c'est pourtant sur ce type d'architecture que repose un grand nombre de sites professionnels !

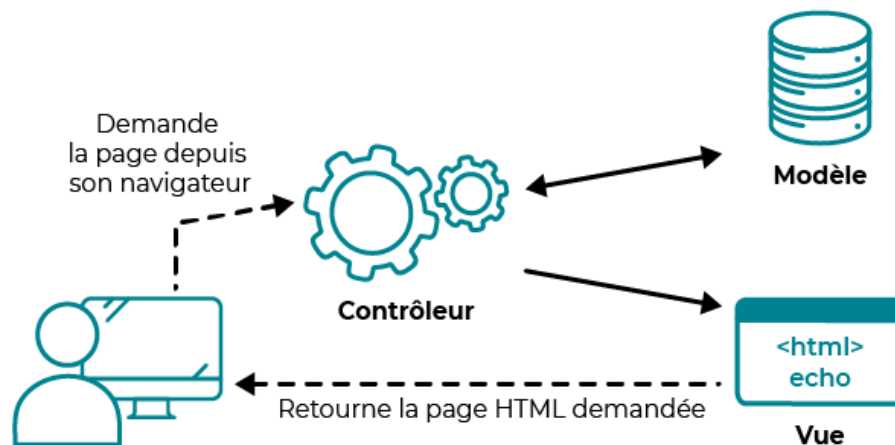


FIGURE 3.11 – Le client et l'architecture MVC

Dans l'architecture MERN, les interactions entre les différents composants sont gérées par des API RESTful (Application Programming Interface) qui permettent aux clients d'envoyer des requêtes au serveur et de recevoir des réponses contenant les données nécessaires.

3.3.2 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** : la couche de **présentation**, la couche **logique** et la couche de **données**.

- **Couche de présentation (interface utilisateur)** : La 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. Dans le contexte MVC, la couche de présentation correspond principalement à la "vue".
- **Couche logique (traitement des données)** : La 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. Dans le contexte MVC, la couche logique correspond principalement au "contrôleur".

- **Couche de données (persistance des données) :** La 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. Dans le contexte MVC, la couche de données correspond principalement au "modèle".

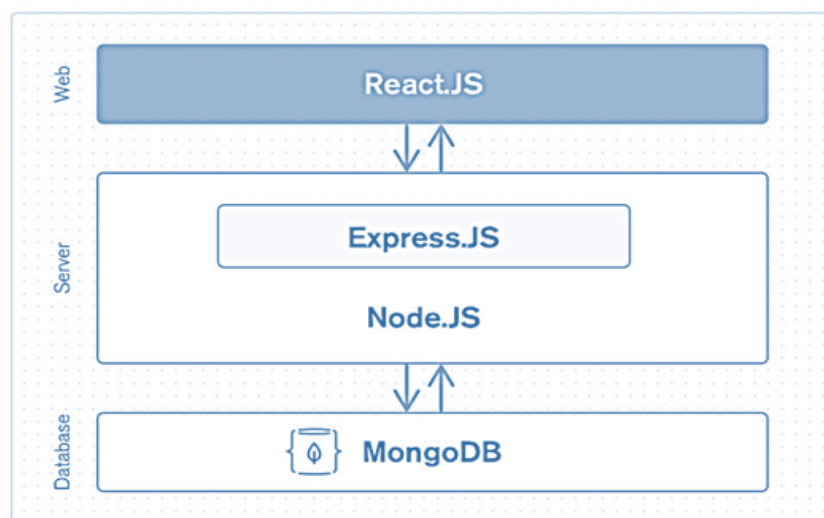


FIGURE 3.12 – Architecture 3 tiers avec MERN

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.

3.4 CONCLUSION

Dans ce chapitre, nous avons abordé les besoins techniques, ainsi que l'architecture logique et physique de notre application. Nous avons identifié et exposé les exigences techniques spécifiques nécessaires à la réalisation de notre projet. De plus, nous avons examiné en détail la structure

globale de notre application, en décrivant comment ses différentes composantes interagissent entre elles. Le prochain chapitre sera consacré pour la conception et la réalisation

CHAPITRE 4 : Conception et Réalisation de l'application

Sommaire

4.1	INTRODUCTION	34
4.2	Langage de modélisation utilisée	34
4.3	Conception des besoins	35
4.3.1	Diagramme de classe	35
4.3.2	Diagrammes de séquence objet	36
4.3.3	Diagrammes d'activités	39
4.3.4	Diagramme d'état-transition de changer l'état d'une tâche . .	40
4.4	Conception de la base de données	41
4.4.1	Modèle Logique des données :	42
4.5	Implémentation technique de la solution	42
4.5.1	Page d'accueil	42
4.5.2	Page d'inscription	42
4.6	Test et validation de la solution	44
4.7	Conclusion	44

4.1 INTRODUCTION

Ce chapitre présente la phase cruciale de conception et réalisation d'un projet. Nous allons détailler la conception logicielle de notre projet à l'aide de diagrammes de classe, de séquence objets et d'état de transition pour représenter les aspects statiques et dynamiques du plateforme. Nous allons concevoir également la base de données pour une gestion efficace des données et créer des diagrammes de navigation détaillés pour la conception graphique. Et finalement nous allons présenter les interfaces de notre solution développée.

4.2 Langage de modélisation utilisée

Notre choix s'est porté sur l'approche orientée objet afin de faciliter l'évolution d'applications complexes et disposer d'une panoplie d'outils et de langages performants pour le développement. Pour ce faire, nous avons opté pour le langage de modélisation UML (Unified Modeling Language).

Le langage de modélisation unifié est un langage de modélisation graphique utilisé pour représenter visuellement différents aspects d'un système logiciel. Il permet de capturer les besoins, les fonctionnalités, les processus métier, l'architecture, la conception et l'implémentation d'un système. 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).

4.3 Conception des besoins

Dans cette section ,nous allons détailler les deux différentes vues de notre projet, comme la vision statique avec le diagramme de classe et dynamique à l'aide de diagrammes de séquence et diagrammes d'activités .

4.3.1 Diagramme de classe

La figure 4.1 représente le diagramme de classes qui contient toutes les informations telles que les classes, les méthodes, les associations et les propriétés.

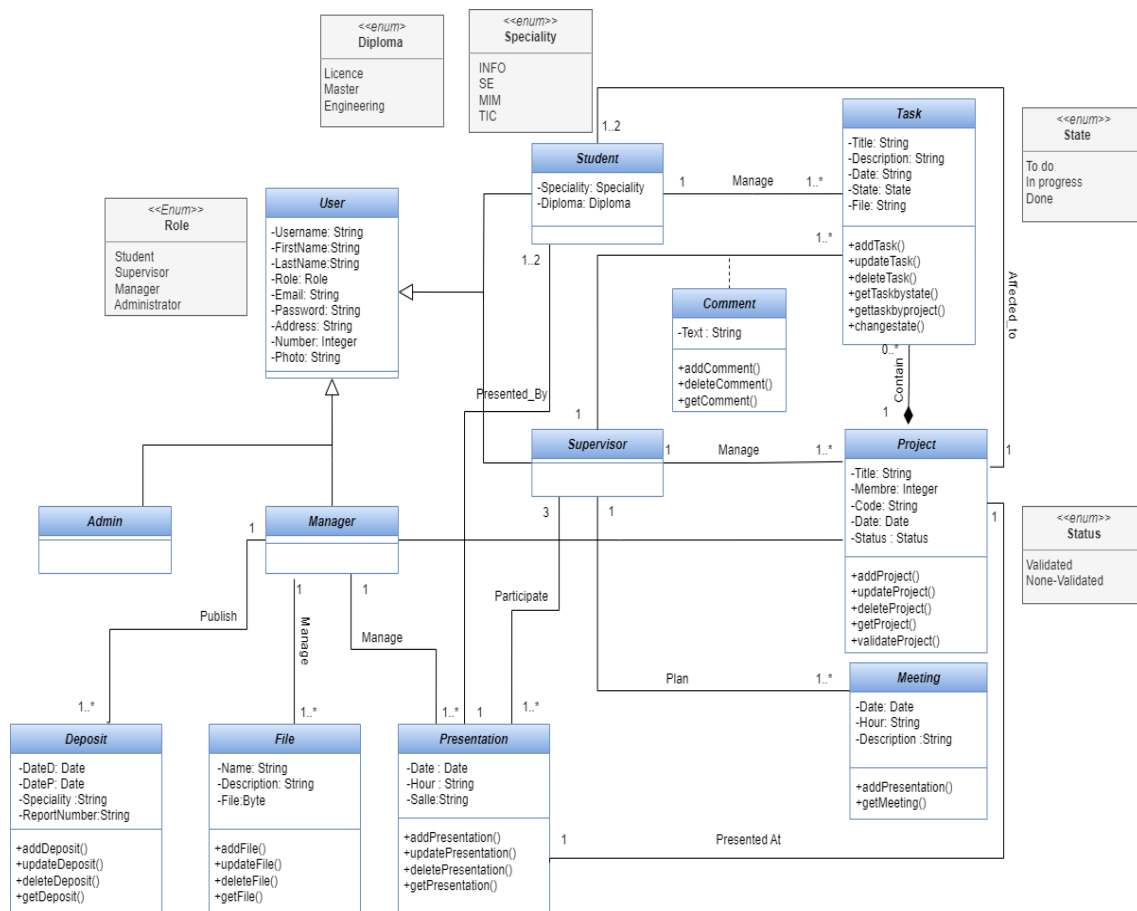


FIGURE 4.1 – Diagramme de classe général

4.3.2 Diagrammes de séquence objet

Dans cette partie nous présentons le déroulement des traitements et des interactions entre les différentes couches via des diagrammes de séquences. Nous présentons un diagramme de séquence objet spécifique pour chaque acteur.

4.3.2.1 Diagramme de séquence objet "s'authentifier"

La figure 4.3 illustre le diagramme de séquence objet d'authentification spécifique à l'encadrant, le gestionnaire et l'étudiant.

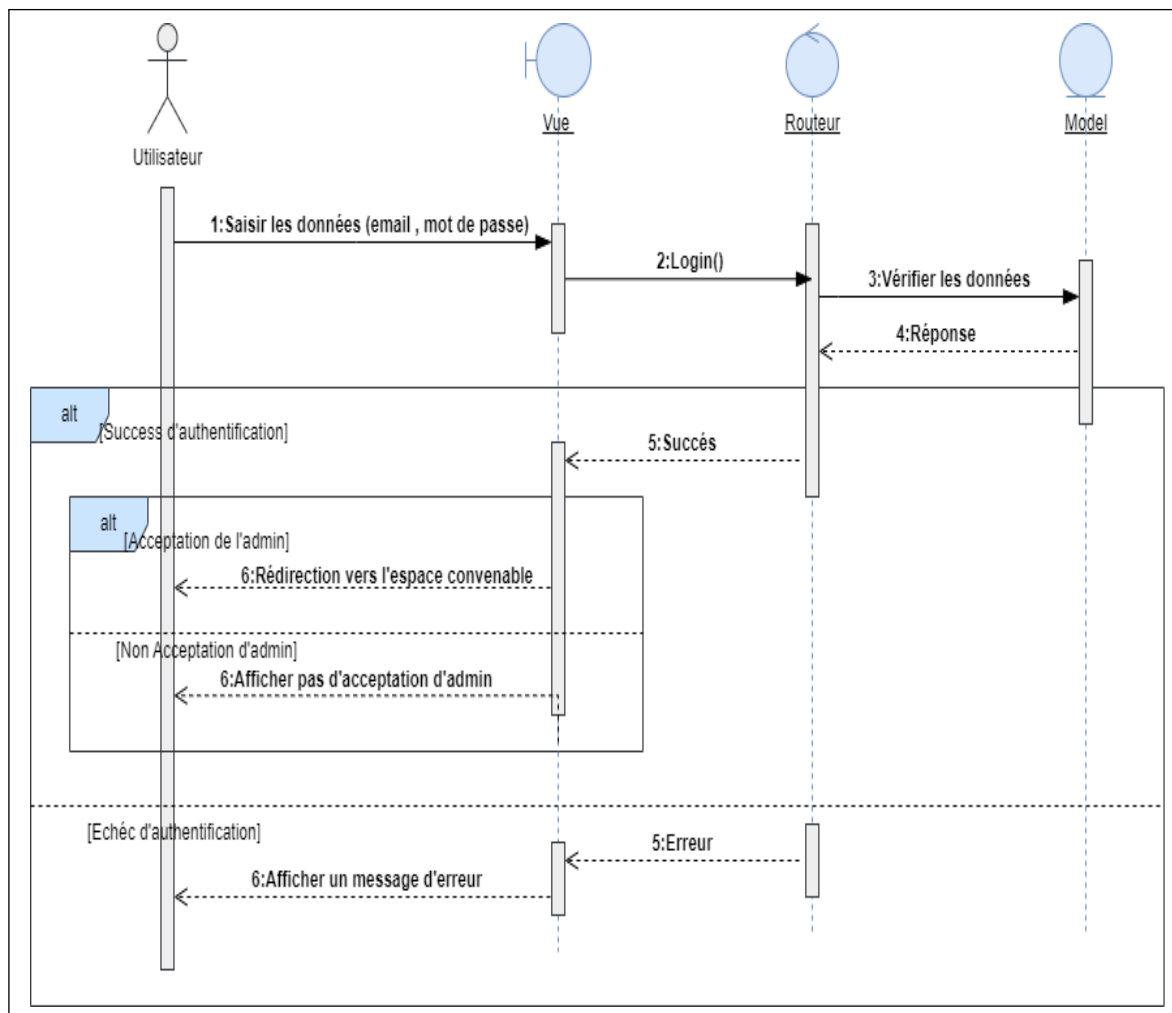


FIGURE 4.2 – Diagramme de séquence objet d'authentification

4.3.2.2 Diagramme de séquence détaillé "Ajouter une tâche "

La figure 4.4 illustre le diagramme de séquence objet "Ajouter une tâche " spécifique à l'étudiant .

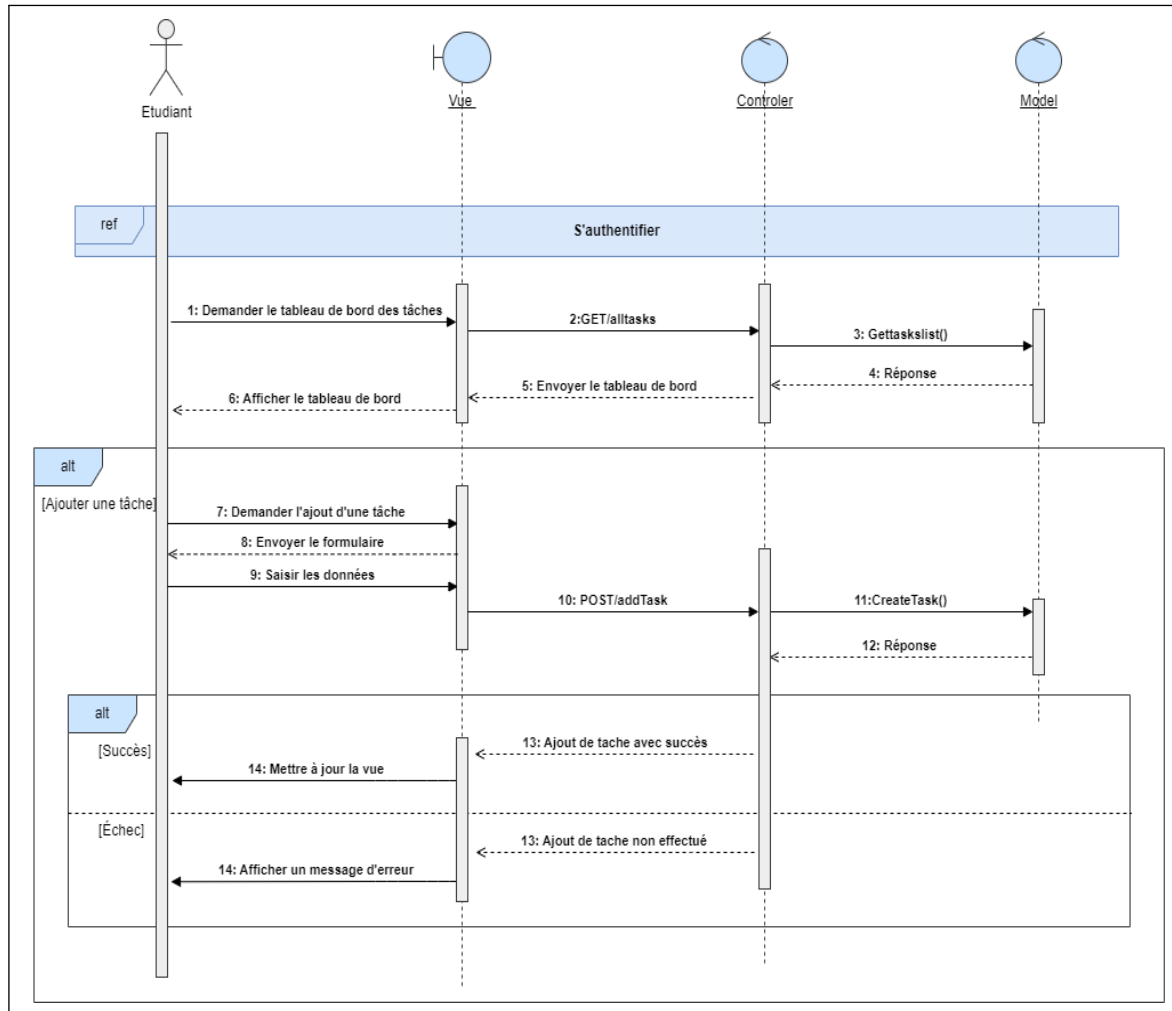


FIGURE 4.3 – Diagramme de séquence détaillé d'ajout d'une tâche

4.3.2.3 Diagramme de séquence objet "Ajouter un commentaire"

La figure 4.5 illustre le diagramme de séquence objet "Ajouter un commentaire" réalisé par l'encadrant .

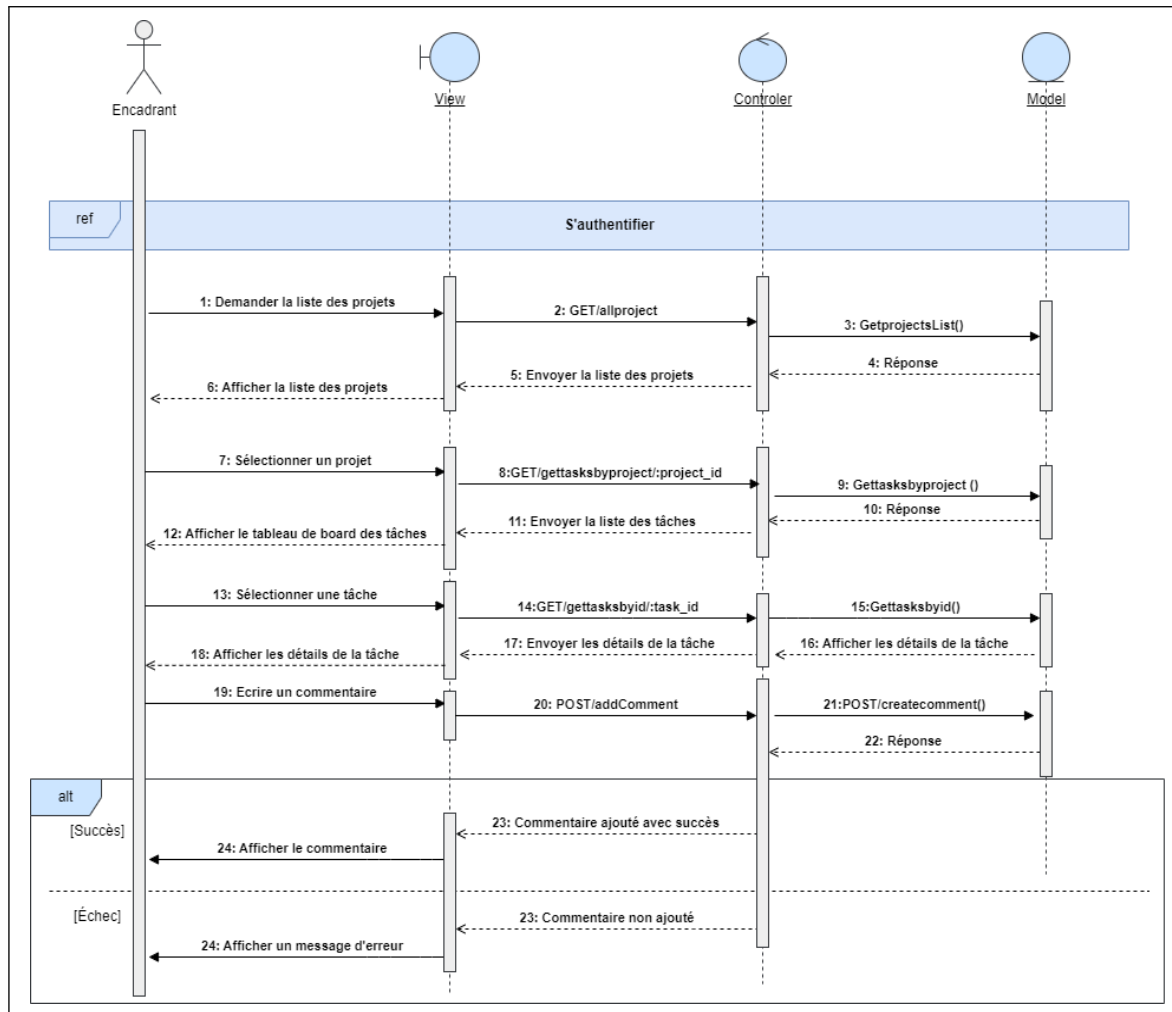


FIGURE 4.4 – Diagramme de séquence détaillé d'ajout d'un commentaire

4.3.2.4 Diagramme de séquence détaillé "Valider un projet"

La figure 4.6 ci-dessous illustre le diagramme de séquence objet " valider un projet " spécifique au gestionnaire .

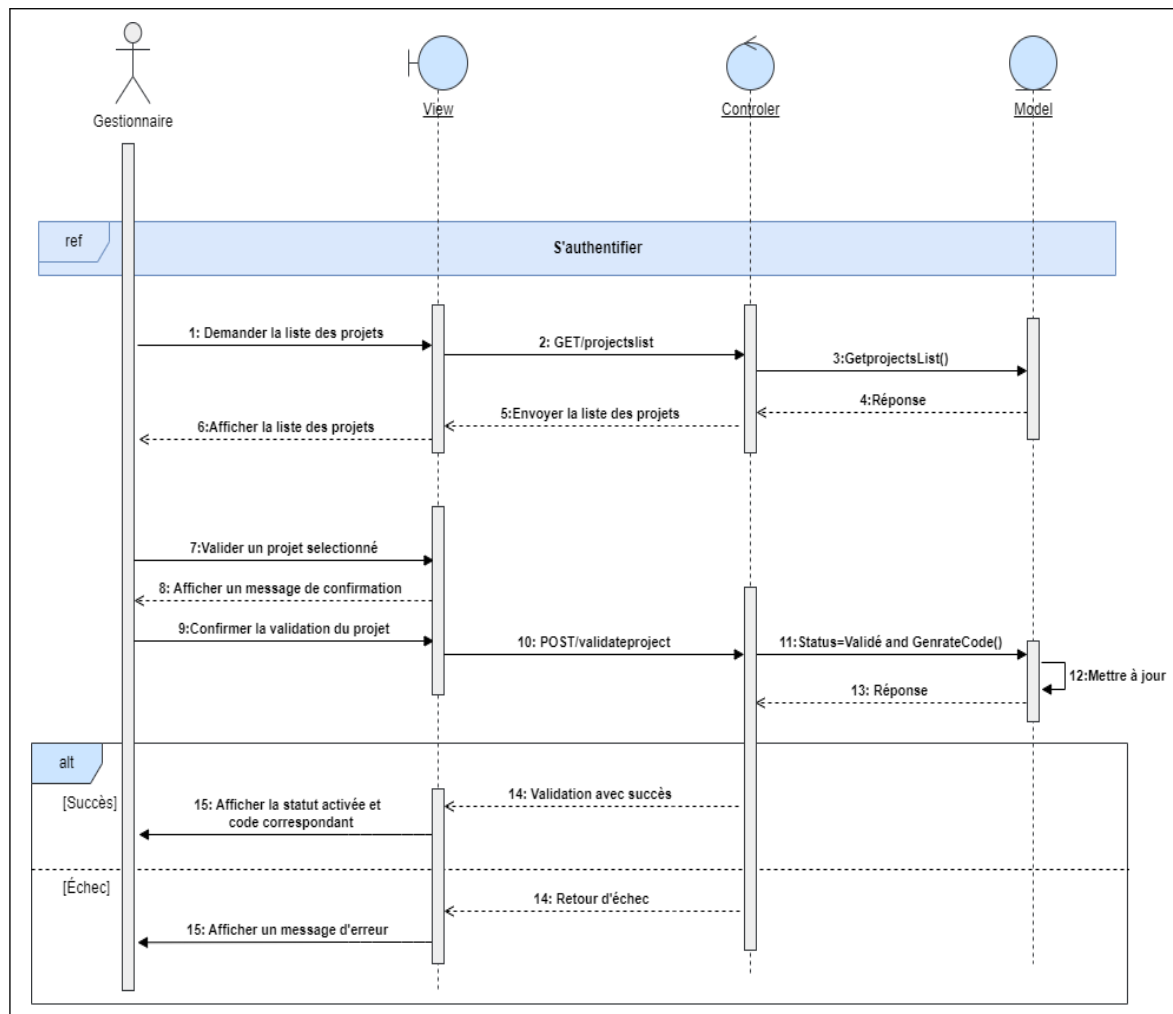


FIGURE 4.5 – Diagramme de séquence détaillé de validation de projet

4.3.3 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.

4.3.3.1 Diagramme d'activité "Ajouter un commentaire "

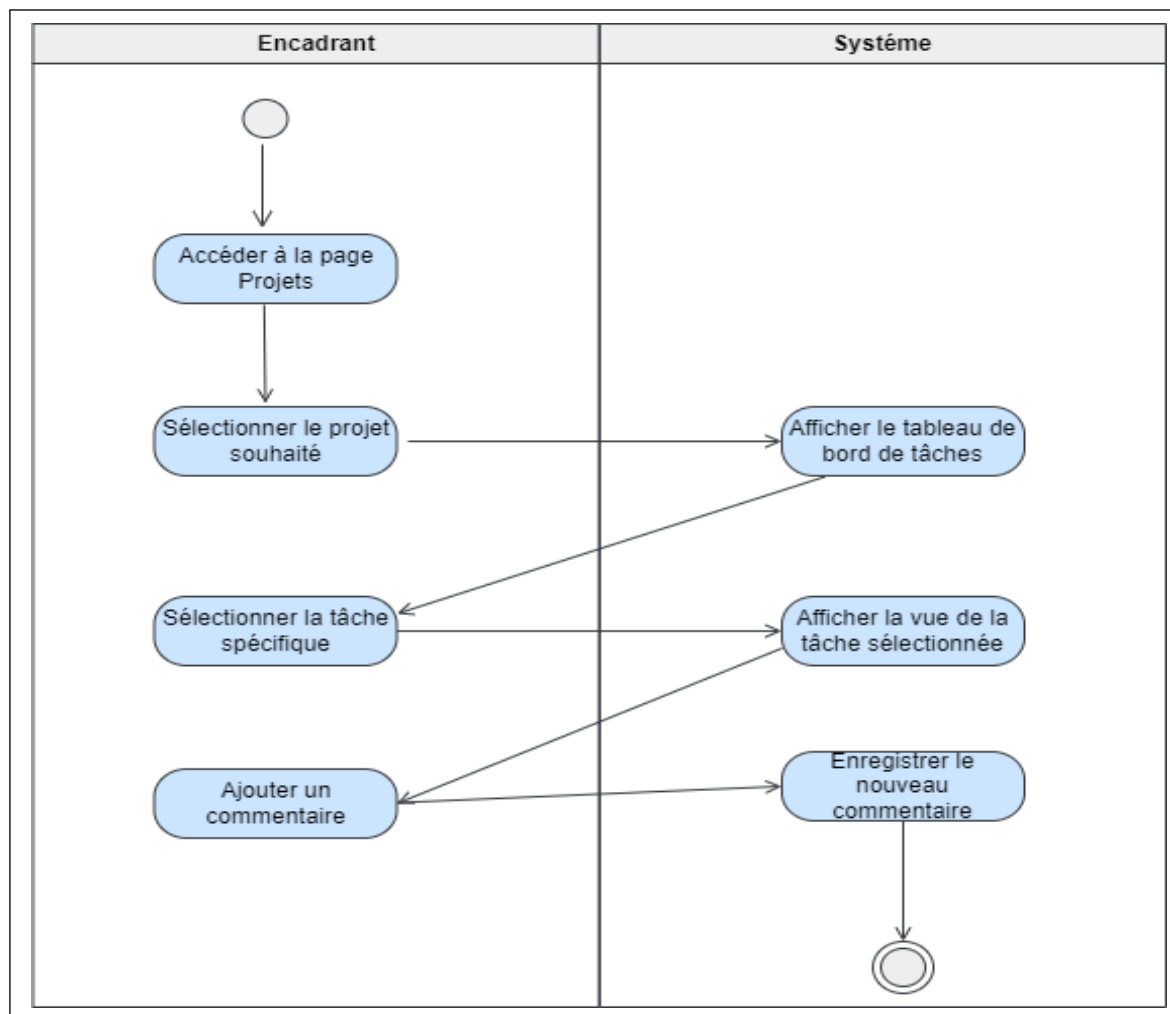


FIGURE 4.6 – Diagramme d'activité d'ajout d'un commentaire

4.3.4 Diagramme d'état-transition de changer l'état d'une tâche

Ce diagramme représente les différents états possibles pour une tâche : "À faire", "En cours" et "Terminé". Les flèches indiquent les transitions entre les états.

Pour passer d'un état à un autre, voici les actions correspondantes :

Si une tâche est dans l'état "À faire" et que vous cliquez sur "Changer état", elle passera automatiquement à l'état "En cours". Si une tâche est dans l'état "En cours" et que vous cliquez sur "Changer état", elle passera automatiquement à l'état "Terminé". Une fois qu'une tâche est dans l'état "Terminé", elle ne peut plus changer d'état.

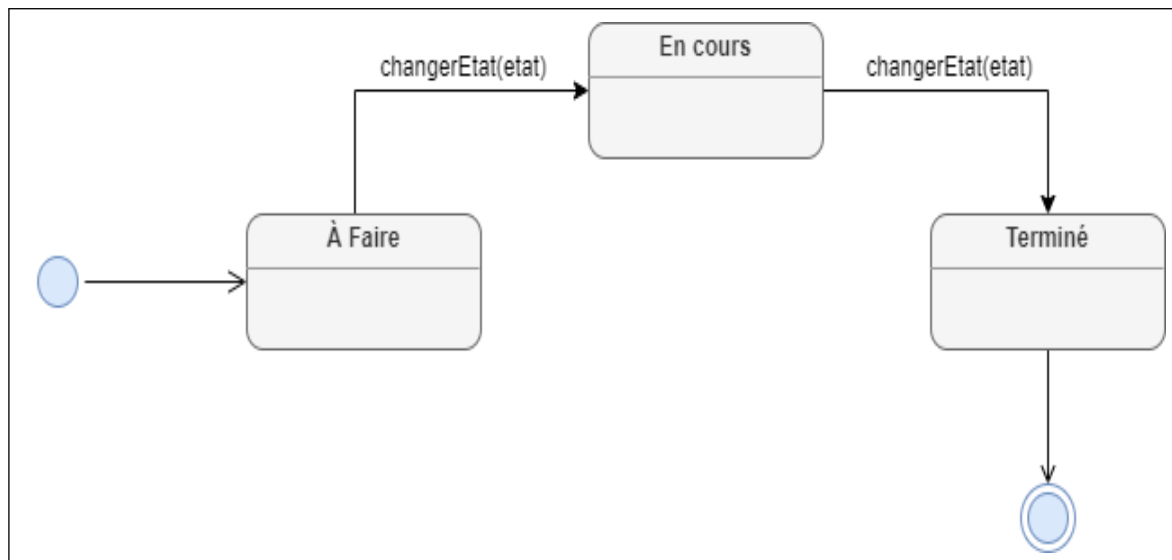


FIGURE 4.7 – Diagramme d'état-transition de changer état

4.4 Conception de la base de données

Pour cette partie, nous allons travailler avec une base de données cloud NoSQL qui est représentée par des collections.

4.4.1 Modèle Logique des données :

4.5 Implémentation technique de la solution

4.5.1 Page d'accueil

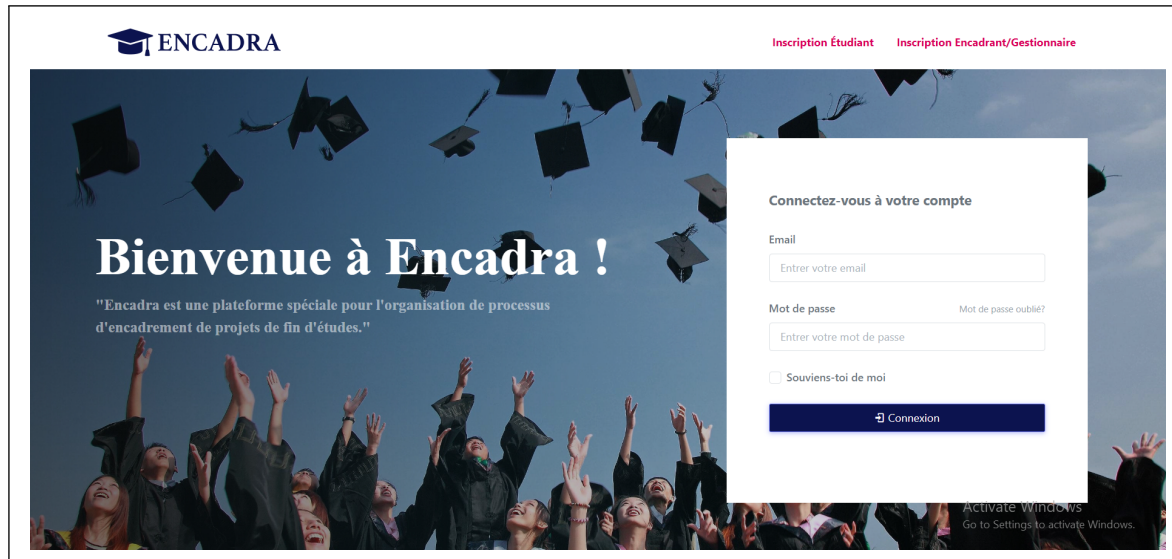
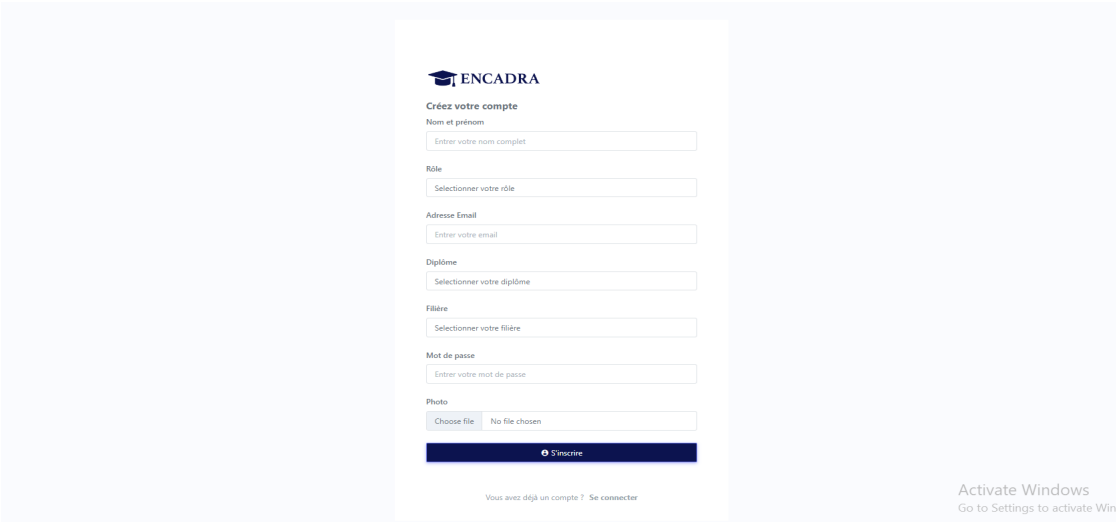


FIGURE 4.8 – Interface d'accueil

4.5.2 Page d'inscription

La figure 4.9 ci-dessous représente l'interface d'inscription pour l'étudiant.



The screenshot shows a web form titled "ENCADRA" with the heading "Créez votre compte". The form contains the following fields and options:

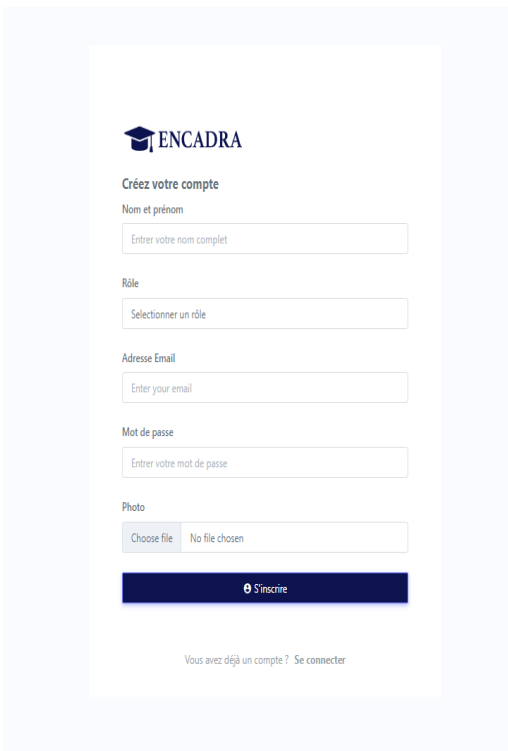
- Nom et prénom:** A text input field with the placeholder "Entrez votre nom complet".
- Rôle:** A dropdown menu with the placeholder "Selectionner votre rôle".
- Adresse Email:** A text input field with the placeholder "Entrez votre email".
- Diplôme:** A dropdown menu with the placeholder "Selectionner votre diplôme".
- Filière:** A dropdown menu with the placeholder "Selectionner votre filière".
- Mot de passe:** A text input field with the placeholder "Entrez votre mot de passe".
- Photo:** A file upload section with a "Choose file" button and the text "No file chosen".

At the bottom of the form is a dark blue button labeled "S'inscrire". Below the button, there is a link: "Vous avez déjà un compte ? Se connecter".

In the bottom right corner of the page, there is a watermark: "Activate Windows Go to Settings to activate Windows".

FIGURE 4.9 – Formulaire d'inscription pour l'étudiant

La figure 4.10 ci-dessous représente l'interface d'inscription pour l'encadrant et le gestionnaire de stage .



The screenshot shows a web form titled "ENCADRA" with the heading "Créez votre compte". The form contains the following fields and options:

- Nom et prénom:** A text input field with the placeholder "Entrez votre nom complet".
- Rôle:** A dropdown menu with the placeholder "Selectionner un rôle".
- Adresse Email:** A text input field with the placeholder "Enter your email".
- Mot de passe:** A text input field with the placeholder "Entrez votre mot de passe".
- Photo:** A file upload section with a "Choose file" button and the text "No file chosen".

At the bottom of the form is a dark blue button labeled "S'inscrire". Below the button, there is a link: "Vous avez déjà un compte ? Se connecter".

FIGURE 4.10 – Formulaire d'inscription pour l'encadrant et le gestionnaire

4.6 Test et validation de la solution

4.7 Conclusion



CONCLUSION GÉNÉRALE



BIBLIOGRAPHIE

[1] **STmicroelectronics**. ST Company Information [**en ligne**]. Mis à jour en février 2019.

Disponible sur :

https://www.st.com/content/st_com/en/about/st_company_information/who-we-are.html

[2] **Florent Lothon**. Introduction aux méthodes agiles [**en ligne**]. Disponible sur :

agiliste.fr/introduction-methodes-agiles

[3] file:///C:/Users/user/Desktop/PFE/old**NUTCACHE**. les methodes agiles [**en ligne**].

Disponible sur :

<https://www.nutcache.com/fr/blog/les-methodes-agiles/>, 2010.

[4] **USB-IF**. USB Power Delivery Specification Revision 2.0, Version 1.3, le 12 January 2017.

Format PDF. Disponible sur :

<https://www.usb.org/document-library/usb-power-delivery-0> **MVC**

<https://openclassrooms.com/fr/courses/4670706-adoptez-une-architecture-mvc-en-php/7847928-decouvr>

architecture MVC : <https://markovate.com/blog/mern-stack-development/>



ANNEXES

A.1 TITRE ANNEXE 1

A.2 TITRE ANNEXE 2