

Mémoire de Projet de Fin d'Etudes

Présenté en vue de l'obtention du

Diplôme National de Licence en Science
d'Informatique

Author Name

Date

ABSTRACT

A brief summary of the project.

DÉDICATION

To my parents.

TABLE DES MATIÈRES

1	Contexte général et étude préalable	1
1	Contexte général et cadre académique du projet	2
1.1	Cadre académique du projet	2
1.2	Présentation de l'entité d'accueil	2
1.3	La problématique	3
1.4	Idée générale de notre projet	4
1.5	Concepts de base liés à notre projet	4
2	Analyse de l'existant et perspectives critiques	5
2.1	Étude des solutions existantes	5
2.2	Évaluation des solutions étudiées	7
3	Solution proposée	9
4	Méthodologie du projet	10
4.1	Processus Unifié (PU)	10
4.2	Méthode Two Track Unified Process	11
2	Fondements de l'Intelligence Artificielle et de l'Infrastructure Cloud dans notre Plateforme	12
1	L'intelligence artificielle dans notre plateforme	13
1.1	Définitions	13
1.2	Explication des concepts clés	13
1.3	Algorithme d'Apprentissage Adopté	15
2	Le Cloud dans notre plateforme	17
2.1	Présentation du cloud computing	17
2.2	Type de cloud	18

2.3	Avantages de l'adoption du cloud dans notre projet	18
2.4	Service cloud adoptés	19
3	Analyse et spécification des besoins	20
1	Identification des acteurs	21
2	Modèle informationnel de contexte	22
3	Recueil des besoins	22
3.1	Capture des besoins fonctionnels	23
3.2	Capture des besoins non fonctionnels	24
4	Spécification des besoins	25
4.1	Diagramme de cas d'utilisation global	25
4.2	Diagrammes de cas d'utilisation détaillés	25
4.3	Analyse des besoins	28
5	Capture des besoins techniques	38
5.1	Framework frontend	39
5.2	Framework backend	39
5.3	Base de données	40
4	Conception	42
1	Modèle architectural	43
1.1	Architecture physique (Architecture 3-tiers)	43
1.2	Architecture logique (Modèle MVC)	44
1.3	Architecture applicative	45
2	Conception de la base de donnée	47
2.1	Modèle conceptuel de données	47
2.2	Modèle logique de données	48
3	Conception de la vue statique : Diagramme de classes	49
4	Conception de la vue dynamique	51
4.1	Diagramme de séquence de conception	51
4.2	Diagramme d'activité « Modifier profil »	55
4.3	Diagramme d'états-transitions « Archiver classroom »	55
4.4	Diagramme de timing	56
5	Conception graphique : Maquettage	57
5	Réalisation	59
1	Environnement et outils de travail	60
1.1	Environnement matériel	60

TABLE DES MATIÈRES

1.2	Environnement logiciel	60
2	Framework Next.js	62
3	Implementation du Modèle LLM	63
3.1	API GroqCloud	63

TABLE DES FIGURES

1.1	Logo du « Globe Services Informatique (GSI) »	3
1.2	Logo de « Google Classroom »	6
1.3	Logo de « Piazza »	6
1.4	Logo de « ChatGPT »	6
1.5	Logo de « Poe »	7
1.6	Logo du « Undrstnd »	10
1.7	Représentation de la méthodologie 2TUP	11
2.1	Le déroulement et les étapes de Dynamic RAG	15
2.2	Le déroulement de l'algorithme Sparse Mixture of Experts	16
3.1	Diagramme de contexte dynamique de notre plateforme	22
3.2	Diagramme de cas d'utilisation global	25
3.3	Diagramme de cas d'utilisation détaillé de l'acteur « Etudiant »	26
3.4	Diagramme de cas d'utilisation détaillé de l'acteur « Enseignant »	27
3.5	Diagramme de cas d'utilisation détaillé de l'acteur «Modèle MoE»	28
3.6	Diagramme de séquence « S'identifier »	31
3.7	Diagramme de séquence « Créer classroom »	35
3.8	Diagramme de séquence « Démarrer chat »	38
3.9	Les statistiques de la satisfaction des utilisateurs à l'usage des frameworks et bibliothèques dans le domaine du développement web	39
3.10	Les statistiques des options de base de données les plus populaires	41
4.1	L'architecture 3-tiers	44
4.2	L'architecture du modèle MVC	45

4.3	L'architecture applicative	46
4.4	Diagramme de déploiement	47
4.5	Modèle conceptuel de données (MCD)	48
4.6	Modèle conceptuel de données (MCD)	50
4.7	Diagramme de séquence de conception « S'identifier »	52
4.8	Diagramme de séquence de conception « Créer classroom»	53
4.9	Diagramme de séquence de conception « Démarrer chat »	54
4.10	Diagramme d'activité « Modifier Profil »	55
4.11	Diagramme d'activité « Modifier Profil »	56
4.12	Diagramme de timing« Démarrer chat »	57

LISTE DES TABLEAUX

1.1	Tableau comparatif entre les solutions existantes	8
2.1	Descriptions des services de cloud computing	19
3.1	Description textuelle du cas d'utilisation « s'identifier »	28
3.2	Description textuelle du cas d'utilisation « Gérer classroom »	32
3.3	Description textuelle du cas d'utilisation « Démarrer chat »	36
5.1	Caractéristiques de l'environnement matériel	60
5.2	Liste des outils utilisé lors du développement de l'application	61

Introduction

A long introduction.

CHAPITRE 1

CONTEXTE GÉNÉRAL ET ÉTUDE PRÉALABLE

Contents

1	Contexte général et cadre académique du projet	2
1.1	Cadre académique du projet	2
1.2	Présentation de l'entité d'accueil	2
1.3	La problématique	3
1.4	Idée générale de notre projet	4
1.5	Concepts de base liés à notre projet	4
2	Analyse de l'existant et perspectives critiques	5
2.1	Étude des solutions existantes	5
2.2	Évaluation des solutions étudiées	7
3	Solution proposée	9
4	Méthodologie du projet	10
4.1	Processus Unifié (PU)	10
4.2	Méthode Two Track Unified Process	11

Introduction

Ce chapitre vise à exposer l'étude préliminaire de notre projet. Tout d'abord, nous présenterons, dans la première section, le contexte général et l'entité qui nous a accueilli pour faire un stage de fin d'études. De plus, nous aborderons, dans cette section, la problématique et l'idée générale de notre projet. Puis, nous entreprendrons une analyse approfondie de l'existant en mettant en évidence les avantages et les limites des solutions similaires présentées sur le marché afin de nous en inspirer et de retenir une solution plus raffinée. Ensuite, nous détaillerons la solution retenue. Enfin, nous exposerons, dans les deux dernières sections, la méthodologie de développement la plus adaptée à la réalisation de notre projet et le diagramme de Gantt illustrant le planning général de celui-ci. Finalement, nous clôturons ce chapitre par une conclusion.

1 Contexte général et cadre académique du projet

Dans cette partie, nous représenterons le contexte général de notre projet qui inclut le cadre académique, la présentation de l'organisme d'accueil, la problématique et l'idée générale de notre projet.

1.1 Cadre académique du projet

Notre projet, intitulé « *Conception et développement d'une plateforme E-learning intégrant un modèle intelligent d'élucidation des documents* », s'inscrit dans le cadre de la préparation d'un projet de fin d'études en vue de l'obtention du diplôme national de Licence en Sciences d'Informatique à « **l'Institut Supérieur d'Informatique et de Mathématiques de Monastir (ISIMM)** » pour l'année universitaire 2023/2024. Le stage a été effectué au sein de la société **Globe Services Informatique GSI** durant une période de 4 mois.

1.2 Présentation de l'entité d'accueil

L'idée initiale de notre projet est proposée par nous-même et elle est adoptée par **STE Globe Services Informatique (GSI)**, une entreprise établie à Houmet Souk, Monastir, en Tunisie. Fondée le 20 mai 2000 en tant que société à responsabilité limitée (SARL), **GSI** se concentre principalement sur le commerce et la maintenance de matériel informatique, ainsi que sur le développement web.

Les secteurs d'activités de GSI sont :

- La vente et la maintenance de matériel informatique.



FIGURE 1.1 – Logo du « Globe Services Informatique (GSI) »

- Maintenance et réparation de matériel informatique et pièces électroniques.
- Développement de logiciels et création de sites web.
- Assistance aux entreprises et travaux publicitaires assistés par ordinateur.

Identité de l'organisation :

- **Nom** : Globe Services Informatique (GSI).
- **Fondateur** : Adel Sriha.
- **Adresse** : 5 Rue de la République, Houmet Souk, Monastir, Tunisie.
- **Contact** :
 - **Téléphone** : +216 73 447 836.
 - **Fax** : +216 73 468 696.
 -
- **E-mail** : commercial@gsi.com.tn.
- **Site web** : www.gsi.com.tn.

1.3 La problématique

Dans le quotidien étudiantin et dans le déroulement ordinaire d'un cours, où l'enseignant anime la classe et les étudiants s'investissent pleinement dans l'acquisition et la compréhension

des connaissances s'engagent pleinement dans l'assimilation des connaissances, des perturbations diverses sont fréquemment rencontrées. En effet, les étudiants se heurtent souvent à des obstacles dans la compréhension de leurs cours ce qui entrave leur progression académique.

De plus, les supports pédagogiques fournis par les enseignants tels que les cours, exercices, corrections et examens peuvent, parfois, se révéler insuffisants pour une assimilation complète. Par conséquent, de nombreux étudiants sont amenés à chercher d'autres ressources par eux-mêmes. Néanmoins, l'accès à ces ressources présente un autre défi incontournable. Cette situation limite considérablement leur capacité à acquérir efficacement des connaissances.

Exemple : Le coût élevé de l'accès à certains documents et ressources pédagogiques d'intérêt qui constitue un obstacle financier majeur qui prive les étudiants d'outils essentiels fondamentaux à leur épanouissement éducatif.

1.4 Idée générale de notre projet

À l'origine de notre projet, nous avons puisé notre inspiration dans notre expérience de la vie estudiantine. Ainsi, nous cherchons à relever les défis auxquels nos pairs sont confrontés tels que la compréhension de leurs cours, l'accès aux ressources pédagogiques adaptées et le besoin de soutien académique personnalisé.

Notre idée consiste à concevoir et mettre en œuvre une plateforme interactive et intelligente visant à révolutionner l'expérience d'apprentissage des étudiants et des enseignants. Cette plateforme fournira des contenus multimédias et des outils d'intelligence artificielle (IA) pour favoriser un apprentissage personnalisé et collaboratif. En outre, elle sera une solution open-source riche en services pertinents et intelligents. Pour se faire, nous prévoyons d'exploiter les technologies de pointe en matière d'IA et de tirer parti des services de cloud pour garantir une solution robuste.

Il est primordial de souligner que notre projet repose sur une idée préliminaire, mais celle-ci peut être affinée en étudiant les solutions similaires disponibles sur le marché. Avant de plonger dans une analyse et une évaluation approfondies des options existantes, nous allons d'abord expliquer les concepts fondamentaux qui sont pertinents pour notre cadre de travail.

1.5 Concepts de base liés à notre projet

Comme nous avons déjà mentionné, dans la section précédente, notre projet reposera sur deux concepts fondamentaux : le cloud computing et l'IA.

- **Cloud computing** : Le cloud computing révolutionne la manière dont les services informatiques sont fournis et consommés. En exploitant le cloud, notre projet peut tirer

parti d'une infrastructure évolutive et flexible permettant un accès rapide et sécurisé aux ressources informatiques. Cette approche offre également une réduction des coûts opérationnels et une amélioration de l'efficacité grâce à la mise à l'échelle automatique des ressources en fonction des besoins.

- **Intelligence artificielle (IA) :** L'intelligence artificielle constitue le cœur de notre projet en offrant des fonctionnalités avancées telles que l'analyse de documents, la recommandation de contenu personnalisé et l'assistance virtuelle. Grâce à l'IA, notre plateforme peut comprendre, interpréter et répondre aux besoins des utilisateurs de manière intelligente pour offrir une expérience d'apprentissage plus personnalisée et efficace.

En combinant les capacités du cloud computing et de l'IA, notre projet vise à fournir une solution innovante et intelligente pour répondre aux défis de l'éducation contemporaine.

2 Analyse de l'existant et perspectives critiques

L'étude de l'existant est une étape primordiale qui permet de définir les points forts et les points faibles des systèmes similaires actuellement en place. Alors, cette section sera dédiée à faire une étude approfondie et critique des solutions existantes.

2.1 Étude des solutions existantes

Dans le domaine des technologies éducatives, plusieurs plateformes se démarquent par leur contribution à l'apprentissage. Nous étudierons ces solutions pour comprendre leurs forces. Nous identifierons également les opportunités d'amélioration que notre solution pourrait exploiter.

Étant donné qu'il n'existe pas de solutions tunisiennes similaires à notre projet, notre revue se concentre sur le marché international.

Ainsi, nous avons porté notre attention sur les solutions étrangères les plus connues qui s'alignent avec le contexte de notre projet. Notre étude se focalise, particulièrement sur *Google Classroom*, *Poe*, *ChatGPT* et *Piazza*, les quatre solutions les plus populaires et adaptées dans le monde entier.

- **Google Classroom** est une plateforme éducative qui permet aux enseignants de créer des salles de classe virtuelles pour leurs étudiants, partager des documents, des devoirs et communiquer avec les apprenants. Le logo de Google Classroom est illustré à la figure 1.2,

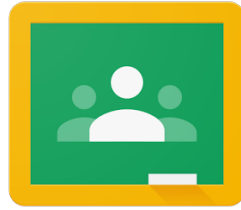


FIGURE 1.2 – Logo de « Google Classroom »

Google classroom permet de :

- Gérer les cours ainsi que les ressources pédagogiques pour les étudiants.
 - Faire des discussions dans les classes.
- **Piazza** est une plateforme de collaboration en ligne conçue pour faciliter la communication entre les étudiants et les professeurs. Le logo de cette plateforme est affiché dans la figure 1.3, permet de :



FIGURE 1.3 – Logo de « Piazza »

- Stocker les documents.
 - Organiser des discussions entre les étudiants et les enseignants.
- **ChatGPT** ou Chat Generative Pretrained Transformer, est un chatbot doté d'intelligence artificielle qui fournit des réponses textuelles instantanées aux questions des utilisateurs. Il se positionne comme un outil polyvalent répondant aux besoins diverse. Le logo de ChatGPT est illustré à la figure 1.4, permet de :



FIGURE 1.4 – Logo de « ChatGPT »

- Répondre aux questions et aux requêtes en se basant sur le contexte de la conversation.
- Fournir des informations sur divers sujets.

- **Poe** est un chatbot qui fonctionne à partir de textes, offrant des interactions conversationnelles pour répondre aux questions des utilisateurs et fournir une assistance. Le logo de Poe est illustré à la figure 1.5, permet de :

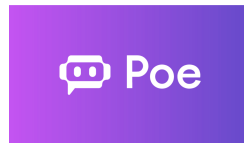


FIGURE 1.5 – Logo de « Poe »

- Résoudre des problèmes en décrivant votre contexte.
- Obtenir des réponses personnalisées à vos besoins.

2.2 Évaluation des solutions étudiées

Après une étude approfondie de l'existant, nous évaluons les plateformes selon les neuf critères (Cx) suivants :

- **Tokens par contexte (C1)** : Ce critère d'évaluation permet d'indiquer le nombre de tokens par contexte, représentant une mesure de la capacité du modèle à comprendre et générer du texte cohérent.
- **Accès gratuit (C2)** : Ce critère d'évaluation permet de vérifier si l'utilisation de la plateforme ne nécessite aucun paiement ou frais d'inscription.
- **Benchmark de la AlpacaEval (C3)** : L'évaluation de la plateforme selon l'outil AlpacaEval, qui est un évaluateur automatique pour les modèles de langage de suivi d'instructions.
- **Gestion de contenu multimédia (C4)** : Ce critère teste la possibilité de télécharger, stocker et accéder à divers formats de contenu tels que des audios, PDFs, vidéos et images.
- **Système de chat interactif avec l'Intelligence Artificiel (C5)** : Ce critère d'évaluation vérifie si une fonctionnalité de messagerie instantanée permettant aux étudiants de poser des questions et de recevoir des réponses basées sur les contenus téléchargés.
- **Création et gestion de salles de classe virtuelles (C6)** : Ce critère d'évaluation examine l'existence d'outils pour créer, gérer et partager des ressources dans des espaces de classe virtuels.
- **Résumés automatiques (C7)** : Ce critère d'évaluation vérifie l'existence des fonctionnalités pour résumer et traduire automatiquement le contenu téléchargé en facilitant la compréhension dans différentes langues.

- **Système de partage et collaboration (C8) :** Ce critère d'évaluation permet de vérifier l'existence d'un système de partage et de collaboration qui permet aux étudiants de partager des informations et de collaborer sur des projets ou des discussions.
- **Temps de génération de la réponse (C9) :** Ce critère d'évaluation évalue le temps estimé et nécessaire pour qu'une application réponde à une requête. Il est spécifiquement conçu pour les solutions axées sur la rapidité et qui utilisent des algorithmes intelligents, en se concentrant sur les interactions entre l'utilisateur et le système pour fournir une réponse efficace.

Le résultat de l'évaluation est affiché dans le tableau 1.1 ci-dessous.

TABLE 1.1 – Tableau comparatif entre les solutions existantes

Critère	Google Classroom	Piazza	ChatGPT	POE (Llama-2-70b)
C1	-	-	6.25%	6.25%
C2	100%	100%	40%	50%
C3	-	-	14.13%	13.87%
C4	80%	70%	-	-
C5	-	-	100%	100%
C6	100%	40%	-	-
C7	-	-	60%	65%
C8	70%	50%	-	-
C9	-	-	12%	14%

D'après notre étude et les résultats d'analyse présentés dans le tableau 1.1, il est évident qu'aucune des solutions étudiées n'a entièrement satisfait à nos critères d'évaluation. Les solutions comme *Google Classroom* et *Piazza* ne présentent pas d'aspect d'intelligence artificielle permettant d'aider les étudiants dans leurs études. En revanche, les solutions dotées d'intelligence artificielle telles que ChatGPT et Poe ne répondent pas aux critères de gestion de salle de classe virtuelle ni de gestion de contenu de différents types de documents. Bien que ces dernières proposent un chat interactif avec une intelligence artificielle, celui-ci se limite au texte et ne permet pas de partager d'autres types de documents tels que des PDF ou des fichiers audio, à moins d'utiliser leur version payante. En outre, le temps de réponse de leur chatbot devient plus long en raison de l'augmentation du nombre d'utilisateurs, ce qui peut affecter la qualité de l'expérience utilisateur.

3 Solution proposée

Après avoir identifié les lacunes des solutions existantes sur le marché, nous envisageons de concevoir et de développer une plateforme plus innovante, gratuite et open-source. Cette plateforme comprendra une gamme de fonctionnalités, notamment :

Fonctionnalités classiques d'apprentissage en ligne : Notre plateforme inclura les fonctionnalités classiques existantes dans les solutions similaires de e-learning :

- **Gestion de contenu multimédia :** Possibilité de télécharger, stocker et accéder à divers formats de contenu tels que des audios, PDFs, vidéos et images.
- **Création et gestion de salles de classe virtuelles :** Outils pour les enseignants pour créer, gérer et partager des ressources dans des espaces de classe virtuels.
- **Système de partage et collaboration :** Permettre aux étudiants de partager des informations et de collaborer sur des projets ou des discussions.

Fonctionnalités du système de chat intelligent : Espace de chat instantané qui se base sur un algorithme intelligent permettant aux apprenants d'interagir avec le système. Les apprenants pourront partager un document de format différent (image, pdf, vidéo, etc.) dans cet espace de chat, bénéficiant ainsi d'une variété de services automatiques avec un temps de réponse très rapide (200 ms vous permettant d'obtenir une réponse de 2 pages), tels que :

- Synthèse.
- Résumé.
- Traduction.
- Explication.
- Réponse aux questions,
- etc.

Nous visons à établir notre plateforme comme une solution de référence, dépassant les applications existantes. Fondée sur des services cloud et des algorithmes intelligents, notre plateforme offrira des temps de réponse extrêmement rapides et une convivialité exceptionnelle. Ainsi, grâce à l'IA, nous pourrions proposer des fonctionnalités avancées telles que la traduction automatique et la recommandation de contenu personnalisé. Le stockage sur le cloud assure une accessibilité optimale aux ressources en garantissant la sécurité des données. Nous avons décidé de baptiser notre plateforme "Undstnd" et de lui d'attribuer le logo présenté dans la figure 1.6.



FIGURE 1.6 – Logo du « Undrstnd »

4 Méthodologie du projet

Afin de réussir un projet, il est important de suivre une méthodologie de développement adaptée pour maîtriser les coûts, structurer et planifier le développement d'applications. Cette méthodologie de travail est utilisée pour structurer, planifier, organiser et contrôler le développement des applications.

Pour notre contexte de travail, nous avons décidé d'adopter la méthode Two Track Unified Process (2TUP) dérivée du Processus Unifié (PU). Dans cette section, nous allons expliquer ce processus ainsi que la méthode que nous avons choisie. Nous expliquerons, également, les raisons du choix de 2TUP.

4.1 Processus Unifié (PU)

Le Processus Unifié est une méthodologie de développement logiciel orientée objet qui intègre toutes les activités de conception et de réalisation dans des cycles de développement comprenant plusieurs phases : Création, élaboration, construction et transition, chacune avec plusieurs itérations. Parmi les avantages du Processus Unifié, on peut citer :

- Un pilotage par les cas d'utilisation.
- Une démarche centrée sur l'architecture.
- Une approche basée sur les modèles, et en particulier les modèles UML.
- Une approche itérative et incrémentale.

4.2 Méthode Two Track Unified Process

2TUP est une méthode de développement logiciel qui utilise le processus unifié pour construire un système. Elle utilise un cycle de développement en Y qui sépare les aspects techniques des aspects fonctionnels. Au début du processus, une étude préliminaire est réalisée pour identifier les acteurs, leurs interactions avec le système et produire un cahier des charges et une modélisation du contexte.

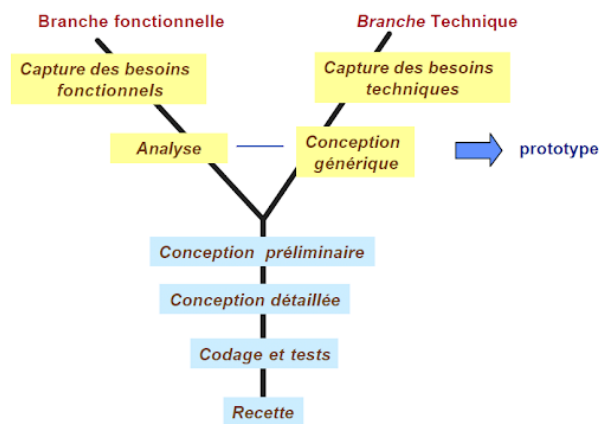


FIGURE 1.7 – Représentation de la méthodologie 2TUP

Le processus s'articule ensuite autour de trois phases essentielles comme l'indique la figure 1.7 :

- **Une branche fonctionnelle** : Elle capitalise la connaissance du métier de l'entreprise. Cette branche capture des besoins fonctionnels, ce qui produit un modèle focalisé sur le métier des utilisateurs finaux.
- **Une branche technique** : Cette phase consiste à concevoir l'architecture du système, à définir les composants et à établir les spécifications techniques.
- **Une phase de réalisation** : Cette phase consiste à développer le système, à tester les composants et à valider le système.

CHAPITRE 2

FONDEMENTS DE L'INTELLIGENCE ARTIFICIELLE ET DE L'INFRASTRUCTURE CLOUD DANS NOTRE PLATEFORME

Contents

1	Identification des acteurs	21
2	Modèle informationnel de contexte	22
3	Recueil des besoins	22
3.1	Capture des besoins fonctionnels	23
3.2	Capture des besoins non fonctionnels	24
4	Spécification des besoins	25
4.1	Diagramme de cas d'utilisation global	25
4.2	Diagrammes de cas d'utilisation détaillés	25
4.3	Analyse des besoins	28
5	Capture des besoins techniques	38
5.1	Framework frontend	39
5.2	Framework backend	39
5.3	Base de données	40

Introduction

Ce chapitre explore les fondements de l'Intelligence Artificielle (IA) et de l'infrastructure cloud dans notre plateforme. Nous aborderons des définitions et explications des concepts clés de l'AI et de l'apprentissage profond, notre choix de l'algorithme MoE et l'intégration du cloud. Cette compréhension nous permettra de saisir les capacités de notre plateforme.

1 L'intelligence artificielle dans notre plateforme

Cette section sera consacrée à la présentation des concepts de base sur l'intelligence artificielle.

1.1 Définitions

Dans cette partie, nous allons définir l'intelligence artificielle et l'apprentissage profond.

Intelligence artificielle

L'intelligence artificielle concerne la création de machines capables de penser et d'agir comme des êtres humains [X]. En d'autres termes, c'est la science qui vise à créer des programmes informatiques et des machines qui peuvent imiter le raisonnement humain, apprendre de l'expérience et accomplir des tâches variées de manière autonome.

Apprentissage profond

L'apprentissage profond, connu sous le nom de deep learning, est une branche de l'IA qui se concentre sur l'entraînement de réseaux de neurones artificiels [x]. Ces réseaux de neurones sont organisés en couches. Chaque couche transforme l'entrée qu'elle reçoit pour produire une sortie. Ces transformations sont ajustées automatiquement pendant l'apprentissage pour améliorer les performances du système.

1.2 Explication des concepts clés

Dans cette section, nous allons clarifier quelques concepts fondamentaux.

Natural Language Model (NLP)

Les modèles de langage naturel (NLP) sont des composants de l'intelligence artificielle conçus pour comprendre et générer un langage humain naturel. Ils sont largement utilisés

dans des applications telles que la traduction automatique, la génération de texte et l'analyse du sentiment. Ces modèles sont entraînés sur de grandes quantités de données textuelles afin d'apprendre les structures linguistiques et de capturer les nuances du langage.

Large Model Language (LLM)

Les modèles de langage de grande taille (LLM) sont des modèles d'intelligence artificielle qui ont été entraînés sur de vastes ensembles de données textuelles pour acquérir une compréhension approfondie du langage naturel. Ces modèles sont capables de générer du texte cohérent et de qualité et sont souvent utilisés pour une variété de tâches en NLP.

Tokenizers

Les tokenizers sont des outils essentiels en traitement automatique du NLP qui découpent le texte en unités plus petites appelées "tokens". Ces tokens peuvent être des mots, des sous-mots ou même des caractères individuels. Ils servent de points de départ pour l'analyse et le traitement du texte.

Retrieval Augmented Generation (RAG)

RAG est un concept utilisé dans le domaine de l'intelligence artificielle et le traitement du langage naturel. Il utilise des techniques de récupération d'informations pour améliorer la génération de texte par des modèles de langage.

En d'autres termes, au lieu de simplement générer du texte en fonction de ce que le modèle a appris pendant son entraînement, un système RAG va d'abord chercher dans une grande base de données de textes pour trouver des informations pertinentes. Il utilisera ensuite ces informations pour générer une réponse plus informée et précise.

Notons que cette base de données de textes peut être vectorielle. C'est-à-dire que les textes sont représentés sous forme de vecteurs dans un espace vectoriel. Ceci permet de les comparer et de les rechercher de manière efficace.

Dynamic Retrieval Augmented Generation (Dynamic RAG)

Dynamic RAG (Retrieval Augmented Generation) est une approche avancée de RAG qui permet d'adapter dynamiquement la récupération d'informations en fonction du contexte de la requête de l'utilisateur.

Dans un système RAG standard, la stratégie de récupération d'informations est généralement fixe et déterminée à l'avance. Par exemple, le système doit toujours chercher dans la même base

de données de textes.

Alors que dans un système Dynamic RAG, le système peut adapter sa stratégie de récupération d'informations en fonction de la tâche à accomplir. Par exemple, pour une tâche de réponse à des questions, le système peut chercher dans une base de données de textes spécifique.

Cette approche permet au système d'être plus flexible et adaptable, ce qui lui permet d'améliorer ses performances sur une variété de tâches.

La Figure 2.1. synthétise les étapes et le déroulement de Dynamic RAG.

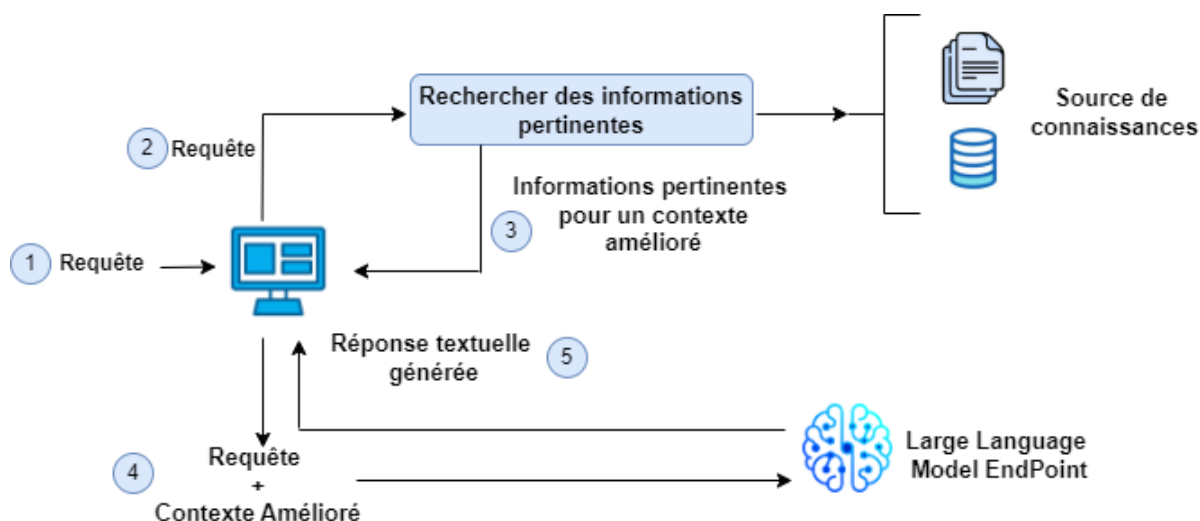


FIGURE 2.1 – Le déroulement et les étapes de Dynamic RAG

1.3 Algorithme d'Apprentissage Adopté

Dans cette section, nous expliquerons le Sparse Mixture of Experts (MoE), l'algorithme d'apprentissage adopté. Nous aborderons l'introduction à cet algorithme, la définition de la couche Sparse MoE et son architecture ainsi que le processus de formation et d'inférence associé.

Notons que les informations de cette section sont issues de la référence suivante.[x]

Couche Sparse MoE et son architecture

La couche Sparse MoE est une composante clé des modèles de transformers remplaçant les couches traditionnelles de réseaux de neurones denses. Chaque couche Sparse MoE est constituée de multiples "experts" qui sont essentiellement des réseaux de neurones. Ces experts sont souvent des réseaux feed-forward (FFN) et peuvent aussi être plus complexes ou former un autre MoE permettant ainsi des MoEs hiérarchiques.

En plus des experts, une couche Sparse MoE inclut un réseau de portes ou "routeur". Ce routeur

détermine comment les tokens sont distribués aux experts. Il est important de noter qu'un même token peut être attribué à plusieurs experts.

Le routeur est constitué de paramètres appris et entraîné avec le reste du réseau.

La Figure 2.2. synthétise l'architecture de l'algorithme Sparse Mixture of Experts.

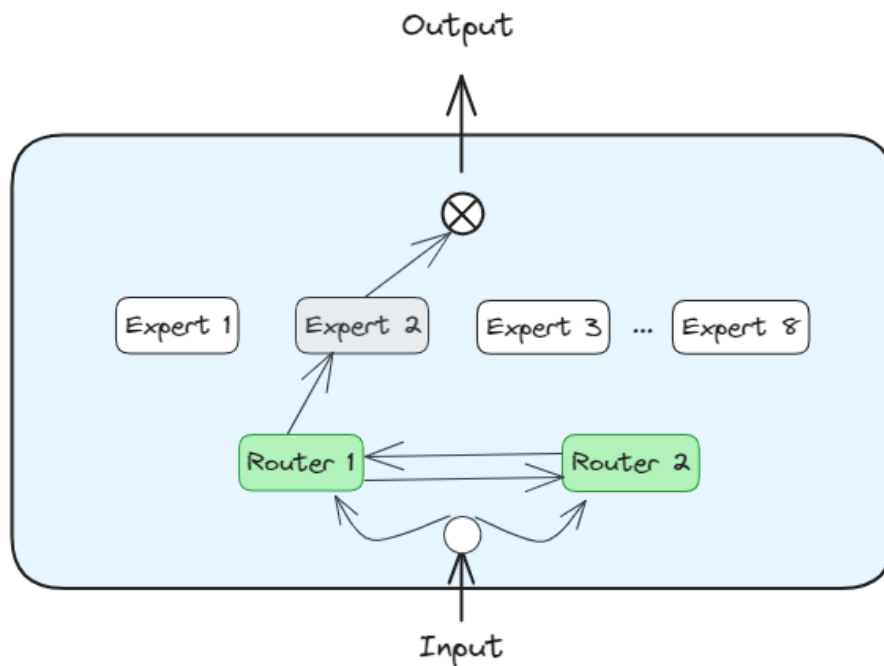


FIGURE 2.2 – Le déroulement de l'algorithme Sparse Mixture of Experts

Fonctionnement de l'algorithme Sparse Mixture of Experts

L'algorithme sparse Mixture of Experts est une méthode avancée qui combine les avantages des modèles experts spécialisés avec la flexibilité des modèles généralistes.

Contrairement aux réseaux neuronaux traditionnels qui utilisent une seule architecture de modèle pour traiter toutes les entrées, le MoE divise le processus d'apprentissage en plusieurs experts spécialisés. Ces experts sont ensuite combinés de manière pondérée pour générer une prédiction finale (voir Figure 2.2). Chaque expert est responsable de traiter une portion spécifique de l'entrée en apportant ainsi sa propre expertise à la résolution du problème. Ces experts peuvent être considérés comme des sous-modèles qui se spécialisent dans différentes parties de l'espace d'entrée.

L'approche sparse du MoE signifie que seuls quelques experts sont activés pour chaque exemple d'entrée. Elle permet une utilisation efficace des ressources computationnelles et une adaptation dynamique aux différentes conditions d'entrée. Cette activation sélective des experts est déter-

minée par des routeurs qui dirigent l'entrée vers les experts les plus pertinents en fonction de son contenu.

Processus de formation et d'inférence

Le processus d'entraînement de Mixture of Experts (MoE) implique simultanément l'apprentissage des paramètres des experts et du routeur.

Pendant la formation, chaque jeton d'entrée est dirigé vers un ou plusieurs experts en fonction des poids appris par le routeur. Les experts traitent ces jetons et retournent leurs sorties respectives. Ces sorties sont combinées pour former la sortie finale du MoE utilisée pour calculer la perte et mettre à jour les paramètres du modèle. En phase d'inférence, seuls les experts les plus pertinents sont sélectionnés selon les poids du routeur en réduisant ainsi la complexité de calcul et améliorant l'efficacité du MoE. Bien que tous les paramètres doivent être chargés en mémoire, seuls ceux nécessaires sont utilisés pendant l'inférence.

Raisons du Choix de l'Algorithme Sparse Mixture of Experts

Le choix d'adopter l'algorithme Sparse Mixture of Experts (MoE) repose sur sa capacité à optimiser l'utilisation des ressources computationnelles en ne faisant fonctionner que les parties pertinentes du modèle. Cela permet d'augmenter l'efficacité de l'apprentissage et réduire la charge de calcul qui est essentiel pour des applications à grande échelle nécessitant des modèles complexes.

En intégrant le Sparse MoE dans notre plateforme, nous sommes en mesure de fournir des réponses plus précises et adaptées aux besoins spécifiques de chaque utilisateur en améliorant ainsi l'expérience globale d'apprentissage.

2 Le Cloud dans notre plateforme

Dans cette partie, nous allons présenter le cloud computing, les types de cloud ainsi que le type de cloud adopté dans notre plateforme.

2.1 Présentation du cloud computing

Le cloud computing est une technologie qui permet d'accéder à des ressources informatiques en utilisant l'internet.

L'infrastructure cloud joue un rôle crucial dans le bon fonctionnement et l'évolutivité de notre

application. Elle nous permet de bénéficier de services performants et sécurisés en optimisant nos coûts d'exploitation.

2.2 Type de cloud

Dans cette section, nous présenterons les types de cloud, suivi par le type adopté.

Types d'infrastructures de cloud computing

Le cloud computing est divisé en trois types principaux : public, privé et hybride.

- **Cloud privé**

Un cloud privé est un environnement de cloud computing dédié à une seule organisation. Dans un cloud privé, toutes les ressources sont isolées et sous le contrôle d'une seule organisation. Ainsi, le cloud privé est également appelé cloud interne ou d'entreprise. [x]

- **Cloud publique**

Un cloud public est une infrastructure informatique dans laquelle un fournisseur de services met des ressources à la disposition du public via internet. Les ressources varient selon le fournisseur mais peuvent inclure des capacités de stockage, des applications ou des machines virtuelles. [x]

- **Cloud hybride**

Un cloud hybride est un environnement informatique mixte dans lequel des applications s'exécutent à l'aide d'une combinaison de ressources de calcul, de stockage et de services dans différents environnements (clouds publics et clouds privés, y compris des centres de données sur site ou en périphérie). [x]

Type de cloud adopté

Notre application repose sur une architecture Cloud hybride qui combine à la fois des services de Cloud public et de Cloud privé. Cette approche nous permet de tirer parti des avantages de chaque modèle en fonction des besoins spécifiques de notre application.

2.3 Avantages de l'adoption du cloud dans notre projet

L'infrastructure Cloud offre de nombreux avantages pour notre projet. Tout d'abord, elle nous permet de bénéficier d'une grande scalabilité en adaptant automatiquement les ressources allouées en fonction des besoins de notre application. De plus, elle garantit une haute disponi-

bilité et une redondance des données grâce à la répartition géographique des serveurs. Enfin, l'infrastructure Cloud nous permet de réduire nos coûts d'exploitation.

2.4 Service cloud adoptés

Parmi les services Cloud intégrés dans notre plateforme, nous retrouvons le Runtime Edge, le Serverless et le CDN (Content Delivery Network).

Les services cloud utilisés dans notre plateforme sont présentés dans le tableau 2.1.

TABLE 2.1 – Descriptions des services de cloud computing

Service	Description
Runtime Edge	Edge Runtime est idéal pour la diffusion de contenu dynamique et personnalisé avec une faible latence en utilisant de petites fonctions simples. Sa rapidité provient de l'utilisation minimale des ressources, mais cela peut être limité dans de nombreux scénarios. Dans notre application, Edge Runtime est utilisé pour optimiser la diffusion de contenu personnalisé pour garantir ainsi une expérience utilisateur rapide et réactive.
Serverless	Les architectures sans serveur, ou Serverless, permettent de déléguer la gestion des serveurs à un fournisseur de services Cloud. Cette approche offre plusieurs avantages tels qu'une réduction des coûts d'exploitation, une scalabilité automatique et une simplification de la maintenance. Dans notre application, le Serverless est utilisé pour gérer les fonctions backend telles que le traitement des données et la gestion des Application Programming Interface (API).
Content Delivery Network	Un Content Delivery Network (CDN) est un réseau de serveurs distribués géographiquement qui permet de distribuer du contenu à grande échelle. L'utilisation d'un CDN permet d'améliorer les performances de notre application en réduisant la latence et en optimisant la bande passante. Dans notre cas, le CDN est utilisé pour distribuer les ressources statiques de notre application telles que les images, les feuilles de style et les scripts.

Conclusion

Notre application exploite des techniques d'apprentissage profond et une infrastructure cloud avancée pour offrir une expérience utilisateur optimale et répondre aux exigences de performance et de fiabilité. Le chapitre suivant approfondira les exigences fonctionnelles et non fonctionnelles pour concevoir notre solution adaptée.

CHAPITRE 3

ANALYSE ET SPÉCIFICATION DES BESOINS

Contents

1	Environnement et outils de travail	60
1.1	Environnement matériel	60
1.2	Environnement logiciel	60
2	Framework Next.js	62
3	Implementation du Modèle LLM	63
3.1	API GroqCloud	63

Introduction

L'étape d'analyse et spécification des besoins est une étape indispensable pour comprendre les fonctionnalités que le système doit fournir. Ce chapitre sera consacré à détailler cette étape : Nous commencerons par l'identification des acteurs. Puis, nous élaborons le diagramme de contexte de notre système. Ensuite, nous allons faire un recueil sur les besoins fonctionnels et non fonctionnels de notre projet. Par la suite, nous allons spécifier et analyser les besoins identifiés en se basant sur le modèle de cas d'utilisation et le diagramme de séquence d'analyse du langage **Unified Modeling Language (UML)**. Enfin, un recueil sur les besoins techniques sera présenté tout en citant les frameworks choisis.

1 Identification des acteurs

Un acteur est une entité externe qui définit le rôle joué par un utilisateur, humain ou non humain, qui interagit avec un système interactif. [x]

Notre système comporte les acteurs suivants :

- **Etudiant** : L'étudiant, en tant qu'utilisateur inscrit, est au cœur de notre système qui peut interagir avec notre chatbot Mixture of Experts et bénéficier de ses services intelligents. Ainsi, il peut accéder aux salons virtuels (Classrooms) et leurs contenus auxquels il est affilié c-à-d les ressources pédagogiques de ses enseignants. Ensuite, cet utilisateur peut partager des supports pédagogiques (de ses enseignants ou provenant de sources externes) dans un espace de chat instantané, sollicitant ainsi d'une gamme de services intelligents tels que l'explication, la traduction, la correction d'exercices, la génération de résumés, etc. Ces fonctionnalités sont conçues pour clarifier ses cours, assurant une assistance personnalisée tout au long de son apprentissage.
- **Enseignant** : L'acteur enseignant est un utilisateur qui admet un compte et joue un rôle actif dans l'expérience d'apprentissage en ligne en facilitant le partage de connaissances et en favorisant un environnement collaboratif au sein de notre plateforme. Ainsi, il bénéficie d'outils conviviaux lui permettant de créer et de gérer des classrooms dédiés à ses cours et y partage ses ressources pédagogiques (cours, fascicule, corrections des fascicules, etc..). Il peut organiser les ressources pédagogiques de manière structurée, favorisant ainsi un accès facile et une interaction fluide avec les étudiants.
- **Modèle MoE (Modèle Mixture of Experts)** : Le modèle Mixture Of Experts est un composant système qui permet de répondre aux questions et requêtes des étudiants en se basant sur les ressources pédagogiques publiées sur la plateforme et ainsi de clarifier

leurs lacunes. En analysant les matériaux pédagogiques, le modèle fournit des explications détaillées et des exemples pertinents pour clarifier les concepts difficiles offrant un soutien individualisé aux étudiants et améliorant leur compréhension du contenu éducatif.

2 Modèle informationnel de contexte

Le modèle informationnel de contexte donne un aperçu global de notre plateforme en identifiant les acteurs et leurs interactions avec le système. À travers le diagramme de contexte, nous illustrons ces échanges pour une meilleure compréhension. Cette représentation simplifiée du fonctionnement de notre application aide à saisir les flux d'informations et le rôle des différents intervenants.

La figure 3.1 montre le diagramme de contexte et illustre bien le modèle informationnel de l'application.

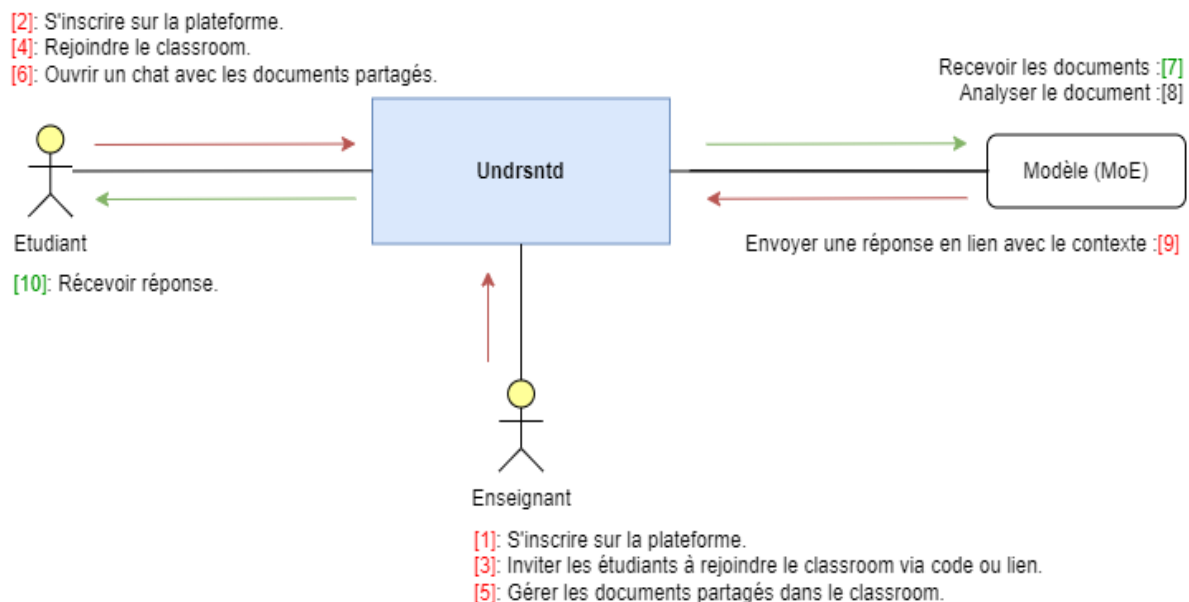


FIGURE 3.1 – Diagramme de contexte dynamique de notre plateforme

3 Recueil des besoins

Dans cette section, nous allons présenter les besoins fonctionnels et non fonctionnels de notre plateforme.

3.1 Capture des besoins fonctionnels

Les besoins fonctionnels définissent les actions que notre système doit accomplir pour répondre aux attentes des utilisateurs. Nous identifions ces besoins en fonction des différents acteurs impliqués. Ce capture nous permet de délimiter le périmètre fonctionnel de notre application et de garantir sa conformité aux exigences des utilisateurs.

- **Besoins fonctionnels de l'étudiant :**

- **S'inscrire :** Permettre à l'étudiant de créer un compte sur la plateforme en fournissant des informations nécessaires.
- **S'identifier :** Permettre à l'étudiant de se connecter à son compte pour accéder aux fonctionnalités de la plateforme.
- **Accéder au classroom :** Permettre à l'étudiant de rejoindre et quitter les classrooms, d'ouvrir ou télécharger les documents partagés, de démarrer un chat avec le modèle dans le contexte des documents partagés et d'interagir avec les publications.
- **Démarrer chat :** Permettre à l'étudiant de créer un espace de communication instantanée avec notre chatbot. Dans cette espace, l'étudiant peut non seulement démarrer une nouvelle conversation mais aussi consulter son historique et poursuivre la communication dans une conversation existante. Cette fonctionnalité lui permet d'envoyer une requête en se référant à un document spécifique ou sans faisant référence. Le premier cas est préconditionné par l'importation d'un document dans la conversation courante.
- **Gérer propre média :** Permettre à l'étudiant de consulter ses documents partagés ainsi que de les supprimer de la base de données.
- **Consulter historique :** Permettre à l'étudiant de consulter l'historique des classrooms, des commentaires ou des postes.
- **Gérer profil :** Permettre à l'étudiant de créer un compte sur la plateforme en fournissant des informations nécessaires.

- **Besoins fonctionnels de l'enseignant :**

- **S'inscrire :** Permettre à l'enseignant de créer un compte sur la plateforme en fournissant des informations nécessaires.
- **S'identifier :** Permettre à l'enseignant de se connecter à son compte pour accéder aux fonctionnalités de la plateforme.
- **Gérer classroom :** Permettre à l'enseignant de créer, modifier, archiver, désarchiver et supprimer ses classrooms.

- **Accéder au classroom :** Permettre à l’enseignant de gérer des publications en créant, modifiant et supprimant des publications, d’ouvrir ou télécharger des documents partagés, d’inviter des étudiants au classroom et d’interagir sur les publications.
 - **Consulter historique :** Permettre à l’enseignant de consulter l’historique des classrooms, des commentaires ou des postes.
 - **Gérer profil :** Permettre à l’enseignant de mettre à jour et de gérer les informations de son profil utilisateur telles que ses informations personnelles et ses préférences ainsi que de supprimer son profil.
- **Besoins fonctionnels du modèle MoE :**
 - **Répondre à une requête :** Permettre au modèle (MoE) de générer des réponses pertinentes en se référant à un document importé par l’étudiant ou à des ressources externes. Le modèle (MoE) est capable de traiter divers types de documents tels que des textes, des articles, des livres et même des codes sources. En fonction de cette analyse approfondie, il peut non seulement répondre aux questions de l’étudiant, mais, aussi, accomplir une multitude de tâches supplémentaires. Ainsi, il peut générer des résumés condensés des documents pour faciliter la compréhension, expliquer des concepts difficiles et donner des exemples concrets, corriger les exercices en fournissant des explications détaillées et bien plus encore. . .

3.2 Capture des besoins non fonctionnels

Les besoins non fonctionnels sont des critères de qualité qui décrivent les attentes non liées, directement, aux comportements fonctionnels.

Les besoins non fonctionnels de notre système sont :

- **La sécurité :** L’application doit garantir à l’utilisateur connecté l’intégrité et la confidentialité de ses données.
- **La convivialité :** La conception de l’interface utilisateur doit être simple et intuitive permettant aux utilisateurs de naviguer facilement dans le système et d’accomplir leurs tâches de manière efficace et agréable.
- **La performance :** Le système doit être en mesure de fournir des performances rapides et efficaces.
- **Fiabilité :** L’application doit fonctionner de façon cohérente pour les utilisateurs.
- **Rapidité :** Le système doit traiter les requêtes en temps réel et fournir des réponses instantanées aux étudiants pour répondre à leurs besoins.

4 Spécification des besoins

Pour une meilleure documentation de notre plateforme, nous allons formaliser, dans cette section, les fonctionnalités offertes par notre application en utilisant le diagramme de cas d'utilisation de l'UML.

4.1 Diagramme de cas d'utilisation global

Le diagramme de cas d'utilisation est le mécanisme le plus populaire pour la spécification des exigences. Il illustre les interactions entre les acteurs et le système. En outre, il donne une vue d'ensemble des fonctionnalités que le système doit offrir pour répondre aux besoins des utilisateurs.

La figure 3.2 illustre le diagramme de cas d'utilisation global de notre application.

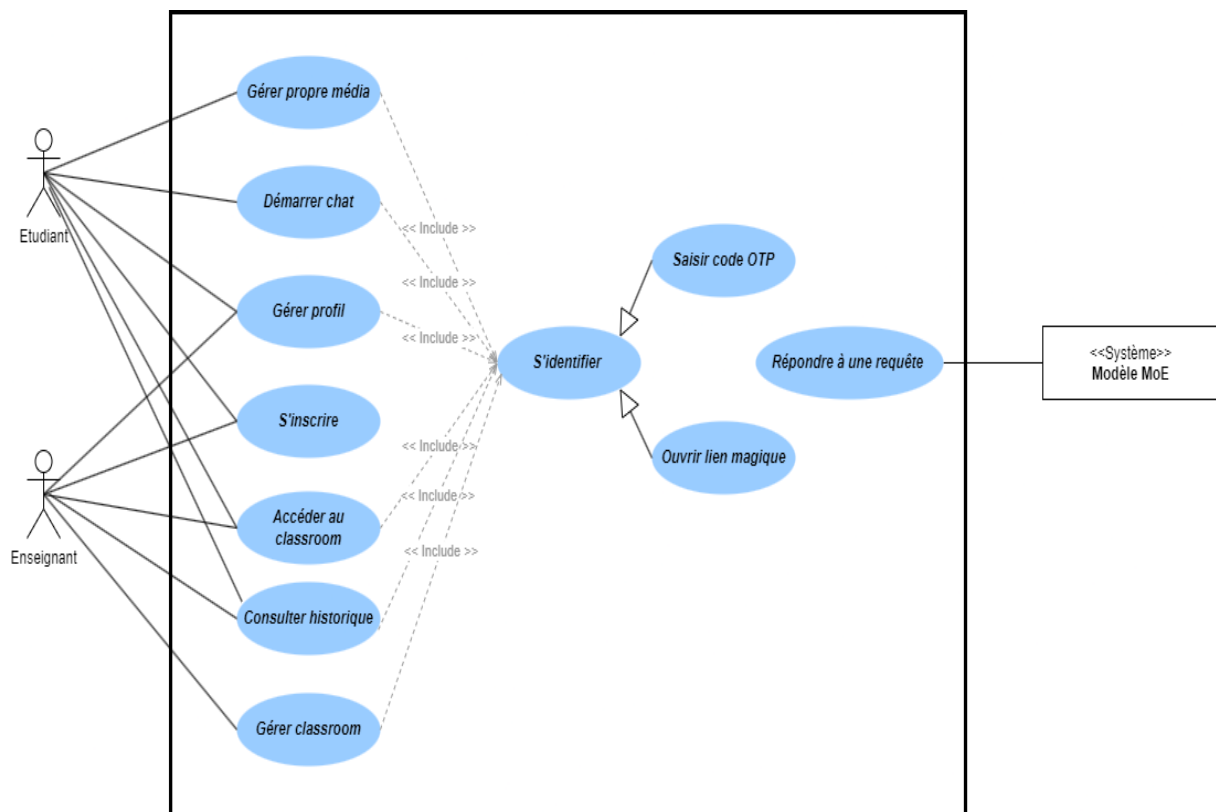


FIGURE 3.2 – Diagramme de cas d'utilisation global

4.2 Diagrammes de cas d'utilisation détaillés

Dans cette partie, nous allons raffiner les cas d'utilisation présentés dans la figure 3.2. Le raffinement sera représenté par un diagramme de cas d'utilisation détaillé de chaque acteur.

Diagramme de cas d'utilisation détaillé de l'acteur « Etudiant »

La figure 3.3 illustre le diagramme de cas d'utilisation détaillé de l'acteur étudiant après avoir s'identifier.



FIGURE 3.3 – Diagramme de cas d'utilisation détaillé de l'acteur « Etudiant »

Diagramme de cas d'utilisation détaillé de l'acteur « Enseignant »

La figure 3.4 illustre le diagramme de cas d'utilisation détaillé de l'acteur enseignant après avoir s'identifier.



FIGURE 3.4 – Diagramme de cas d'utilisation détaillé de l'acteur « Enseignant »

Diagramme de cas d'utilisation détaillé de l'acteur « Modèle MoE »

La figure 3.5 illustre le diagramme de cas d'utilisation détaillé de l'acteur modèle MoE.

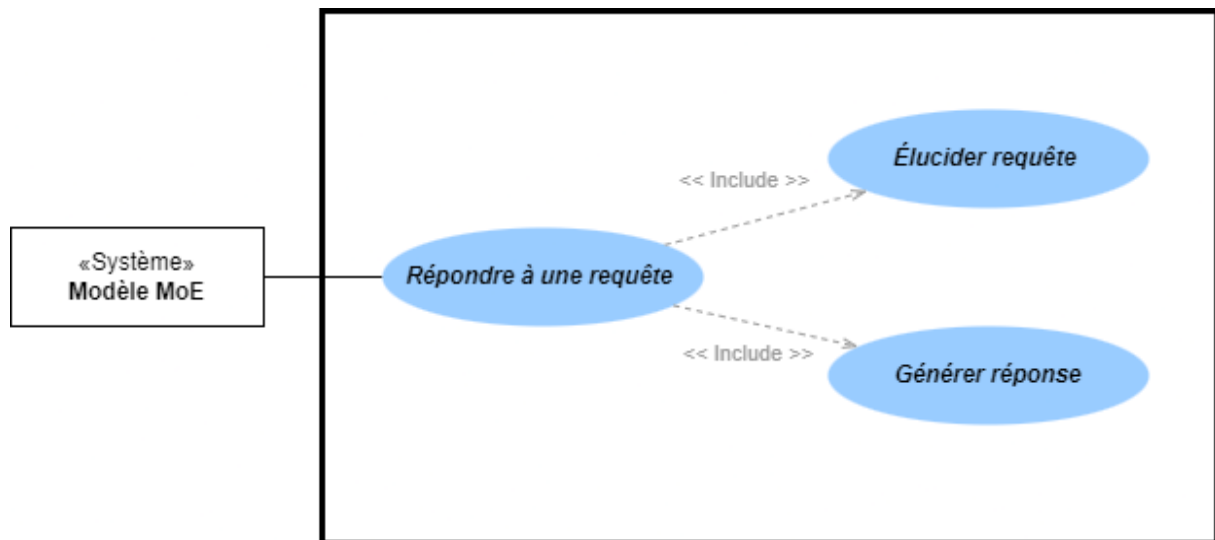


FIGURE 3.5 – Diagramme de cas d'utilisation détaillé de l'acteur «Modèle MoE»

4.3 Analyse des besoins

L'analyse des besoins est une étape importante permettant de représenter le comportement du système au fil du temps. Dans le contexte de la modélisation orienté objet l'analyse des besoins peut se faire par les descriptions textuelles et les diagrammes de séquence d'analyse. Une description textuelle est description des scénarios nominaux, d'extension et d'exception d'un cas d'utilisation. Un diagramme de séquence d'analyse, dite aussi acteur-système, est une représentation graphique utilisée pour montrer l'interaction entre les acteurs et le système où le système est représenté sous d'une boîte noire. Dans cette section, nous présenterons les descriptions textuelles et les diagrammes de séquence d'analyse relatifs aux principaux cas d'utilisation de notre plateforme.

Cas d'utilisation « S'identifier »

- **Description textuelle**

Le tableau 4.3

montre la description textuelle du cas d'utilisation « **S'identifier** ».

TABLE 3.1 – Description textuelle du cas d'utilisation « s'identifier »

Titre de cas d'utilisa- tion	S'identifier
Acteurs principaux	Utilisateur (Enseignant, Étudiant)

Description	Grâce à ce cas d'utilisation, un utilisateur peut s'identifier et accéder à notre plateforme.
Précondition	L'utilisateur doit avoir un email universitaire.
Postcondition	L'utilisateur doit avoir un compte sur la plateforme.
Scénarios nominaux	<p>Scénario nominal 1 : Se connecter</p> <ol style="list-style-type: none"> 1. L'utilisateur accède à la page de connexion. 2. L'utilisateur saisit son email universitaire. 3. L'utilisateur essaie de se connecter en appuyant sur le bouton 'Se connecter'. 4. Si le champ mail est vide, l'Exception 1 se déclenche. 5. Le système envoie un email One Time Password (OTP) à l'email universitaire propre de l'utilisateur avec un code OTP et un lien magique. 6. L'utilisateur choisit une méthode de connexion : <ul style="list-style-type: none"> – Via un lien magique : L'utilisateur clique sur le lien magique dans l'email. – Via la saisie du code OTP : L'utilisateur entre le code OTP dans le champ de la saisie du code OTP. Si le code est erroné, l'Exception 2 se déclenche. 7. Le système affiche la page principale de la plateforme. <p>Scénario nominal 2 : Se déconnecter</p> <ol style="list-style-type: none"> 1. L'utilisateur clique sur son avatar. 2. L'utilisateur se déconnecte en cliquant sur le bouton 'Se déconnecter'. 3. Le système affiche la page de connexion.

Scénarios alternatifs	<ol style="list-style-type: none">1. L'utilisateur accède à la page de connexion.2. L'utilisateur essaie de se connecter en appuyant sur le bouton 'Github'.3. Le système affiche la page principale de la plateforme. Notez bien que l'utilisateur doit utiliser son email universitaire comme email principal et publique sur son compte GitHub.
Exceptions	<ul style="list-style-type: none">– Exception 1 : Si l'utilisateur ne remplit pas le champ email, un message d'erreur « <i>Le champ email est requis</i> » est affiché.– Exception 2 : Si l'utilisateur saisit un code erroné, un message d'erreur « <i>Le code est invalide</i> » est affiché.

- **Diagramme de séquence acteur-système**

Pour ce cas d'utilisation, l'utilisateur peut être soit un étudiant, soit un enseignant lors de l'identification.

La figure 3.6 illustre le diagramme de séquence « **S'identifier** ».

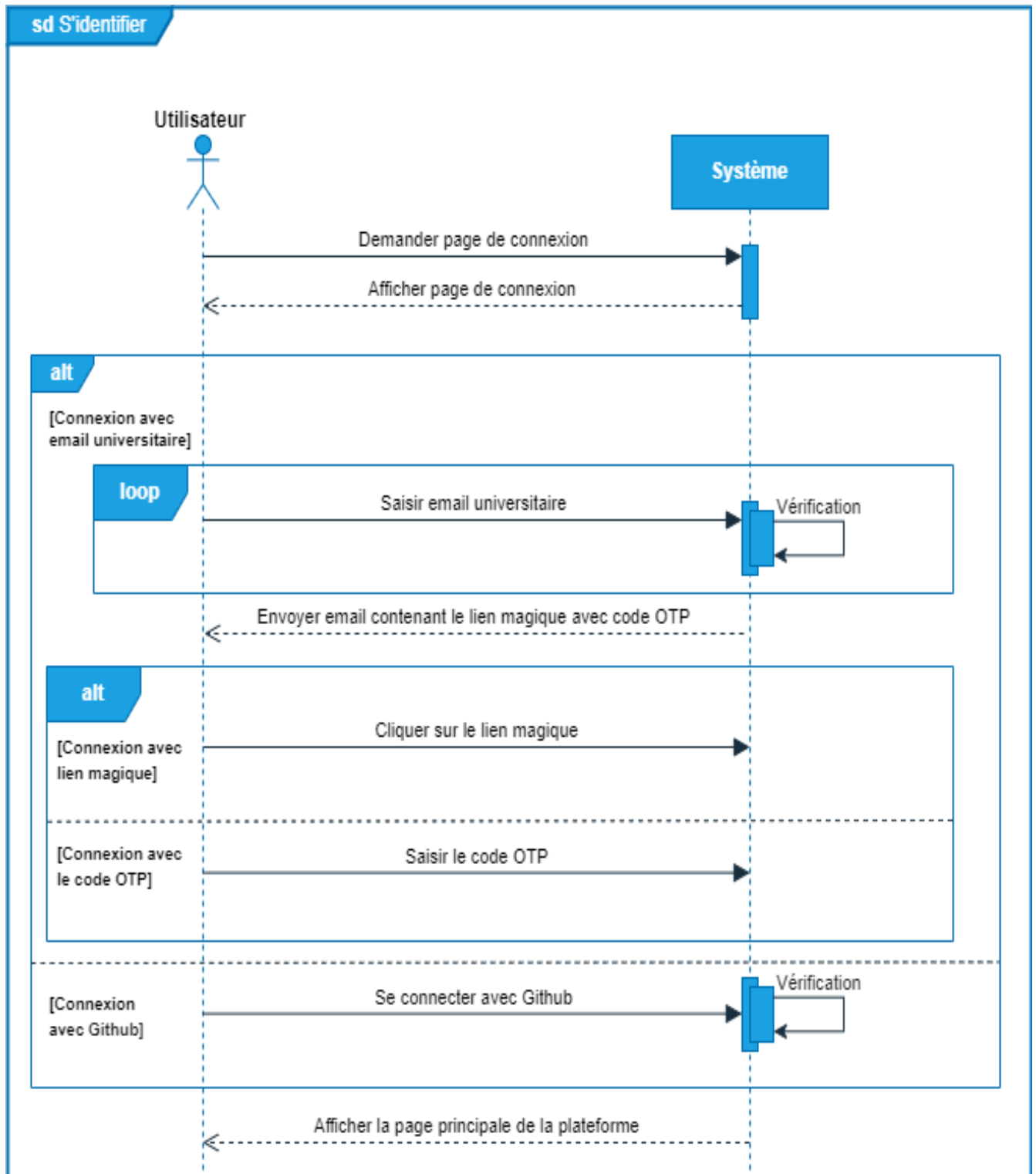


FIGURE 3.6 – Diagramme de séquence « S'identifier »

Cas d'utilisation « Gérer classroom »

- **Description textuelle**

Le tableau 3.2 montre la description textuelle du cas d'utilisation « **Gérer classroom** ».

TABLE 3.2 – Description textuelle du cas d'utilisation « Gérer classroom »

Titre de cas d'utilisation	Gérer classroom
Acteurs principaux	Enseignant
Description	Grâce à ce cas d'utilisation, un enseignant peut créer, modifier, supprimer, archiver et désarchiver un classroom.
Précondition	Enseignant identifié.
Postcondition	La liste des classrooms est mise à jour avec les modifications apportées (création, modification, suppression, archivage du classroom).
Scénarios nominaux	<p>Scénario nominal : Créer classroom</p> <ol style="list-style-type: none"> 1. L'enseignant demande la page de classrooms. 2. Le système affiche la liste des classrooms pour l'enseignant. 3. L'enseignant clique sur le bouton 'Créer classroom'. 4. Le système affiche le formulaire d'ajout d'un classroom. 5. L'enseignant remplit le formulaire de la création du nouveau classroom. 6. (a) S'il existe un champ vide, l'Exception 1 s'affiche. (b) Si un champ ne vérifie pas sa contrainte, l'Exception 2 s'affiche. 7. Le système ajoute le classroom dans la base de données.

Scénarios alternatifs

Scénario alternatif 1 : Modifier classroom

1. L'enseignant demande la page de classrooms.
2. Le système affiche la liste des classrooms pour l'enseignant.
3. L'enseignant clique sur le bouton de la modification du classroom spécifié.
4. Le système affiche le formulaire de modification du classroom.
5. L'enseignant modifie les informations du classroom à modifier.
6. (a) S'il existe un champ vide, l'*Exception 1* s'affiche.
(b) Si un champ ne vérifie pas sa contrainte, l'*Exception 2* s'affiche.
7. Le Système modifie les informations du classroom dans la base de données.

Scénario alternatif 2 : Supprimer classroom

1. L'enseignant demande la page de classrooms.
2. Le système affiche la liste des classrooms pour l'enseignant.
3. L'enseignant clique sur le bouton de suppression de classroom spécifié.
4. Le système affiche une fenêtre superposée pour confirmer la suppression.
5. L'enseignant clique sur le bouton '*Confirmer*'.
6. Le système retire le classroom de la base de données.

Scénario alternatif 3 : Archiver classroom

1. L'enseignant demande la page de classrooms.
2. Le système affiche la liste des classrooms pour l'enseignant.
3. L'enseignant clique sur le bouton d'archivage de classroom spécifié.
4. Le système affiche une fenêtre superposée pour confirmer l'archivage.
5. L'enseignant clique sur le bouton '*Confirmer*'.
6. Le système archive le classroom.

Scénario alternatif 4 : Désarchiver classroom

1. L'enseignant demande la page de classrooms.
2. Le système affiche la liste des classrooms pour l'enseignant.

Exceptions	<ul style="list-style-type: none">– Exception 1 : Si l’enseignant ne remplit pas un ou plusieurs champs du formulaire, un message d’erreur spécifique est affiché pour chaque champ vide.– Exception 2 : Si l’enseignant remplit un ou plusieurs champs du formulaire mais que les contraintes ne sont pas vérifiées, un message d’erreur spécifique est affiché pour chaque champ invalide.
-------------------	---

- **Diagramme de séquence acteur-système**

La figure 3.7 illustre le diagramme de séquence « **Créer classroom** ».

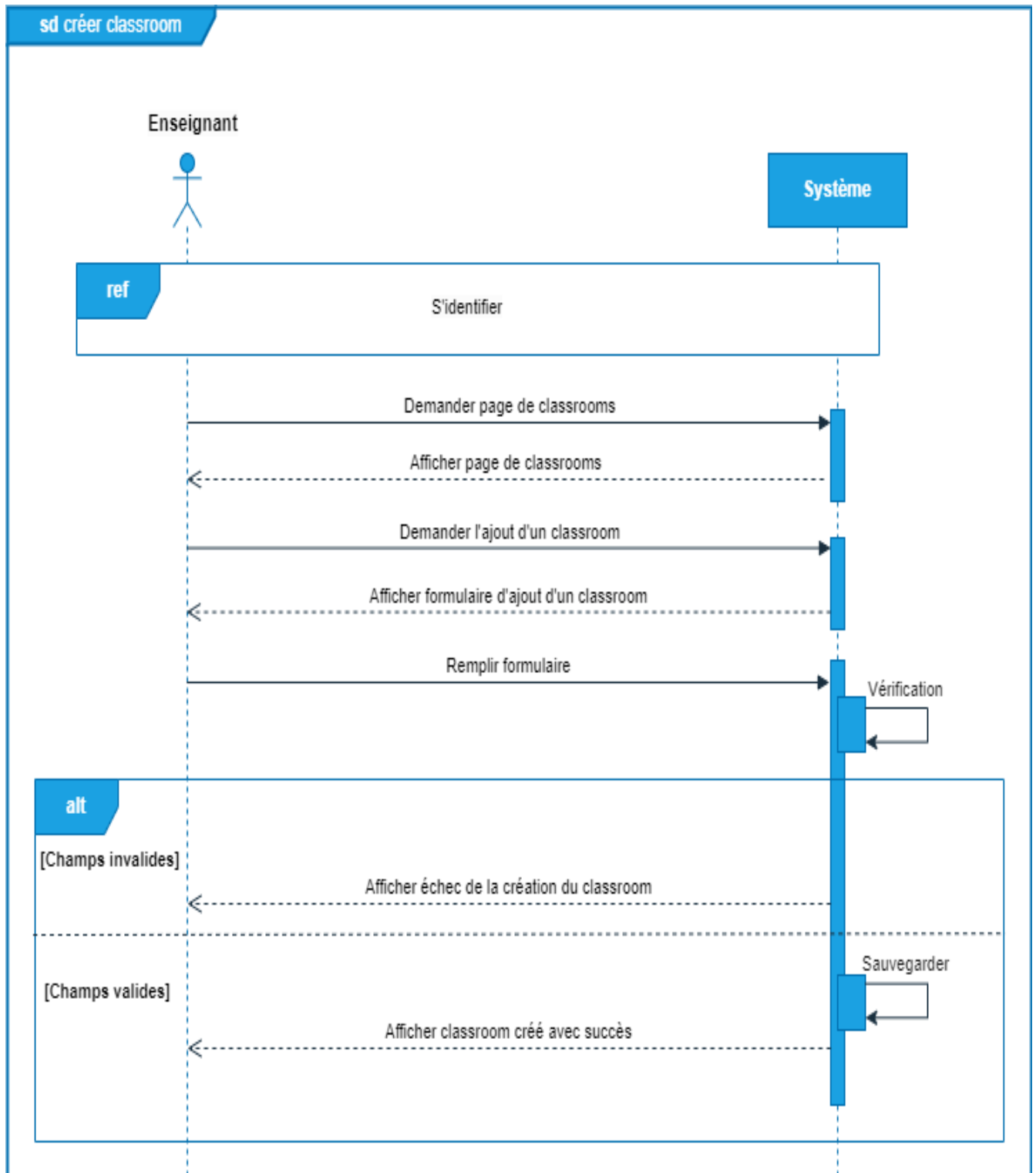


FIGURE 3.7 – Diagramme de séquence « Créer classroom »

Cas d'utilisation « Démarrer chat »

- **Description textuelle**

Le tableau 3.3 montre la description textuelle du cas d'utilisation « **Démarrer chat** ».

TABLE 3.3 – Description textuelle du cas d'utilisation « Démarrer chat »

Titre de cas d'utilisation	Démarrer chat
Acteurs principaux	Etudiant
Description	Grâce à ce cas d'utilisation, un étudiant peut créer une conversation avec le modèle MoE dans notre plateforme. Il peut envoyer des questions, des requêtes (demande d'explication, traduction, synthèse, etc.) sur un document partagé avec le système.
Précondition	Étudiant identifié.
Postcondition	Conversation créée.

<p>Scénarios nominaux</p>	<ol style="list-style-type: none"> 1. L'étudiant demande la page de chat avec le modèle MoE. 2. Le système affiche au étudiant la page de chat. 3. L'étudiant clique sur le bouton '<i>conversations</i>'. 4. Le système affiche liste des conversations précédentes. 5. L'étudiant choisit la manière de communication : <ul style="list-style-type: none"> – Conversation en tenant les conversations précédentes : <ol style="list-style-type: none"> 5.1. L'étudiant sélectionne une conversation (conversation x) – L'étudiant veut démarrer une nouvelle conversation : <ol style="list-style-type: none"> 5.1. L'étudiant remplit le formulaire d'une nouvelle conversation. 6. L'étudiant communique avec le modèle. 7. Le système élucide les demandes de l'étudiant : <ul style="list-style-type: none"> – Réponse en se référant aux ressources externes : Le modèle génère les réponses de l'étudiant en utilisant les informations provenant de ressources externes mentionnées. – Réponse en se référant aux ressources internes : Le modèle génère les réponses de l'étudiant en se basant sur les informations et le contexte mentionnés dans le document partagé, tout en permettant une explication plus approfondie en se référant à des ressources externes. 8. Le système envoie les réponses.
<p>Exceptions</p>	<ul style="list-style-type: none"> – Exception 1 : Si l'étudiant importe un document de format invalide, le système affiche un message d'erreur « <i>Fichier invalide, veuillez vérifier le format de fichier</i> ». – Exception 2 : Si la taille de document dépasse 25 Méga octets, le système affiche un message d'erreur « <i>Veuillez réduire la taille du document pour continuer</i> ».

- **Diagramme de séquence acteur-système**

La figure 3.8 illustre le diagramme de séquence « Démarrer chat ».

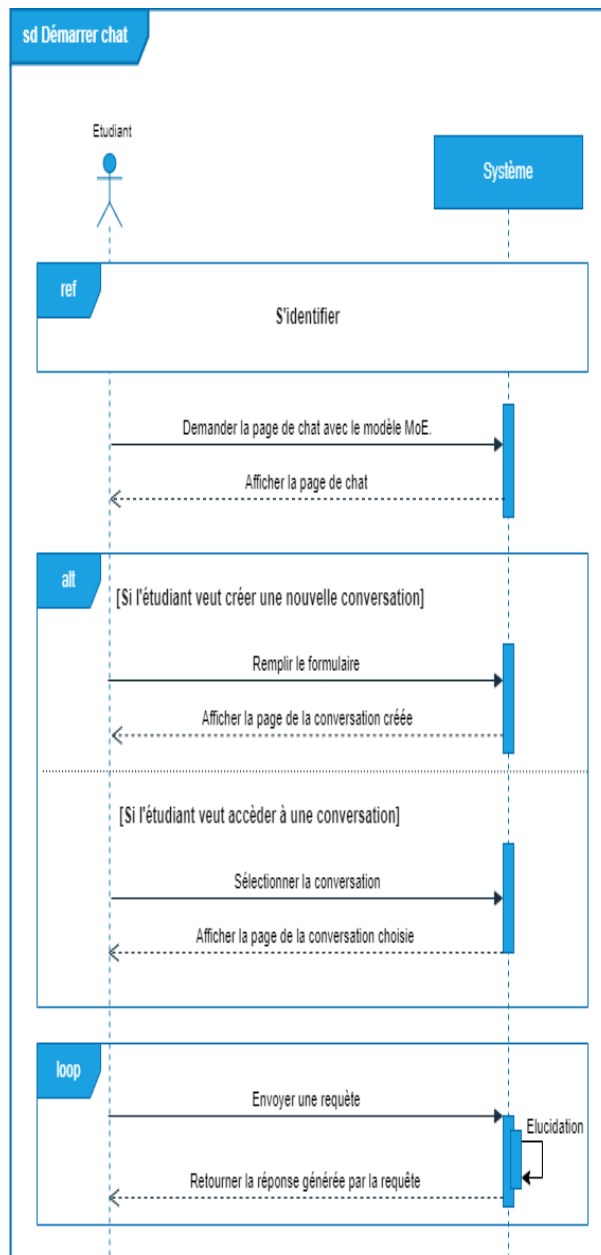


FIGURE 3.8 – Diagramme de séquence « Démarrer chat »

5 Capture des besoins techniques

Dans cette section, nous allons présenter les logiciels et outils utilisés pour développer et mettre en œuvre notre plateforme. Nous allons mettre en valeur le Framework utilisé pour le développement frontend et backend ainsi que la base de donnée choisie pour notre projet.

5.1 Framework frontend

Pour la partie frontend, nous allons utiliser le framework Next.js.

- **Nextjs**

Next.js est un framework gratuit et open source s'appuyant sur la bibliothèque javascript React. Next.js se distingue par sa capacité à offrir une expérience utilisateur performante et réactive. Grâce à sa structure basée sur React, Next.js permet de construire des interfaces utilisateur dynamiques et interactives avec une facilité. Son approche basée sur le rendu côté serveur garantit des performances optimales. Il offre aussi ses fonctionnalités avancées telles que le pré rendu des pages et le routage dynamique facilitant la navigation fluide et la gestion efficace des données.

La figure 3.9 présente les statistiques de la satisfaction des utilisateurs à l'usage des frameworks et bibliothèques dans le domaine du développement web.

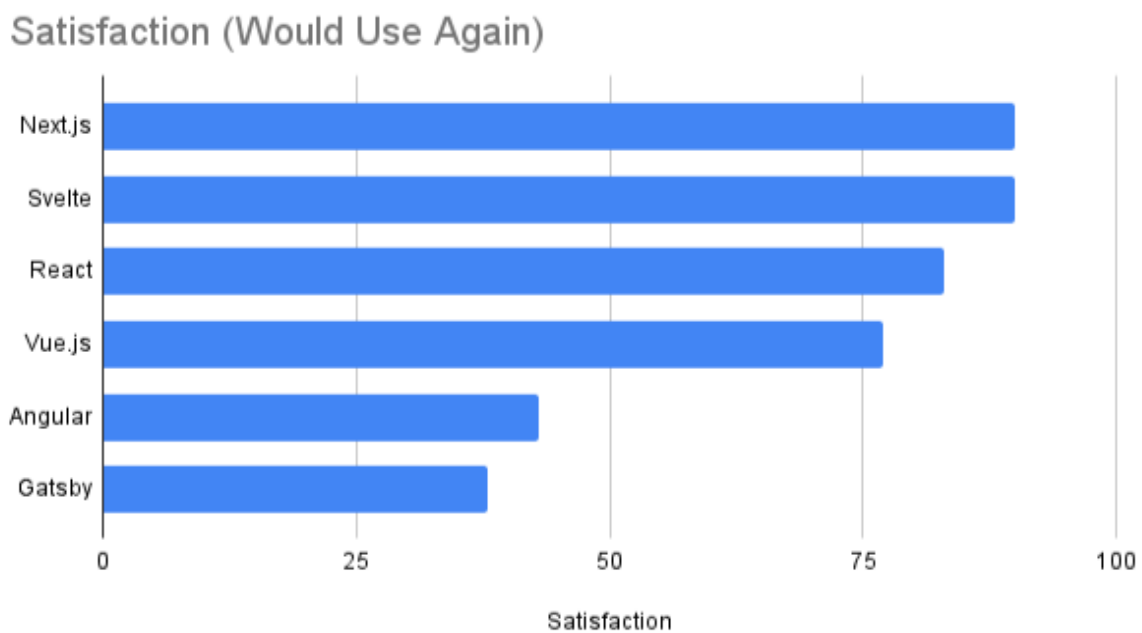


FIGURE 3.9 – Les statistiques de la satisfaction des utilisateurs à l'usage des frameworks et bibliothèques dans le domaine du développement web

[6]

5.2 Framework backend

Next.js est souvent associé au développement frontend en raison de sa capacité à créer des interfaces utilisateur réactives et performantes. Il offre également des fonctionnalités robustes

pour la partie backend. Grâce à sa compatibilité avec Node.js, Next.js permet de développer facilement des applications web full-stack en combinant efficacement le frontend et le backend. L'utilisation de Next.js pour la partie backend offre plusieurs avantages comme la possibilité de créer des API RESTful, la gestion des bases de données et la mise en œuvre de la logique métier côté serveur. Avec son architecture flexible et sa facilité d'utilisation, Next.js permet aux développeurs de créer des applications web complètes et cohérentes en offrant ainsi une expérience utilisateur fluide et satisfaisante pour la partie frontend et aussi la partie backend.

5.3 Base de données

Pour notre système de gestion de base de données, nous avons opté pour une approche SQL en utilisant PostgreSQL. Cette décision découle des nombreux avantages offerts par les bases de données SQL telles que la robustesse des transactions et la gestion efficace des données structurées.

- **PostgreSQL**

PostgreSQL est un système de gestion de base de données relationnelle open source qui offre une performance robuste, une fiabilité élevée et une extensibilité exceptionnelle. Connu pour sa conformité aux normes SQL, PostgreSQL prend en charge une large gamme de fonctionnalités avancées telles que les jointures complexes, les transactions ACID (Atomicité, Cohérence, Isolation, Durabilité) et la réplication asynchrone. Il est préféré pour être un choix pour les développeurs et les entreprises qui cherchent une solution de base de données puissante et fiable pour leurs applications.

La figure 3.10 présente les statistiques des options de base de données les plus populaires.

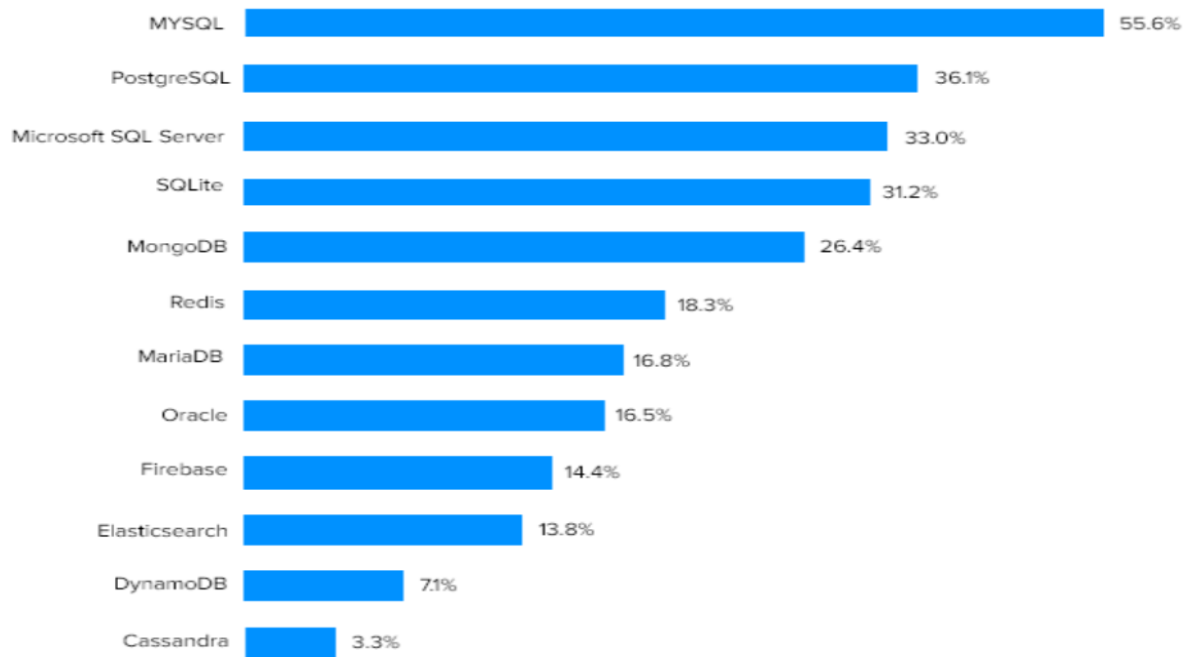


FIGURE 3.10 – Les statistiques des options de base de données les plus populaires [7]

Conclusion

Dans ce chapitre, nous avons mis en œuvre les branches fonctionnelle et technique de la méthode 2TUP se utilisant le modèle de cas d'utilisation (diagramme et description textuelle) et le diagramme de séquence d'analyse. Dans le chapitre suivant, nous présenterons la conception préliminaire et détaillée de notre application en appliquant la branche de la réalisation de la méthode 2TUP.

CHAPITRE 4

CONCEPTION

Introduction

Dans ce chapitre, nous allons nous concentrer sur la conception de notre application en détail. Tout d'abord, nous présenterons l'architecture globale de l'application suivie de la conception de la base de données. Ensuite, nous présenterons la conception détaillée qui comprendra les vues statiques illustrées à l'aide de diagrammes de classes ainsi que les vues dynamiques illustrées à l'aide d'une variété de diagrammes telle que les diagrammes de séquence de conception, diagramme d'activité, diagramme d'états-transitions et diagramme de timing. Enfin, nous ferons quelques maquettes pour la conception graphique.

1 Modèle architectural

Avant de démarrer la conception de notre application, il est essentiel de déterminer le modèle architectural qui répondra le mieux à nos besoins. Dans le cas de notre application, nous avons choisi d'adopter l'architecture 3-tiers ainsi que le modèle de conception **Modèle-Vue-Contrôleur (MVC)**.

1.1 Architecture physique (Architecture 3-tiers)

L'architecture 3-tiers illustrée par la Figure 4.1 est constituée de trois couches distinctes : la couche de présentation, la couche logique de l'application et la couche de stockage des données.

- **La couche de présentation** : Cette couche est chargée d'afficher les données et d'interagir avec l'utilisateur. Elle peut être intégrée à l'aide d'un navigateur web pour une application web.
- **La couche de logique de l'application** : Cette couche contient la logique métier de l'application. Les opérations de l'application sont gérées par cette couche comme la validation des données, la logique d'authentification, la gestion des erreurs et etc.
- **La couche de stockage de données** : Cette couche occupe la position la plus basse. Elle gère les données de l'application. Cette couche peut être constituée d'un ou plusieurs bases de données, de services web ou de toute autre source de données.

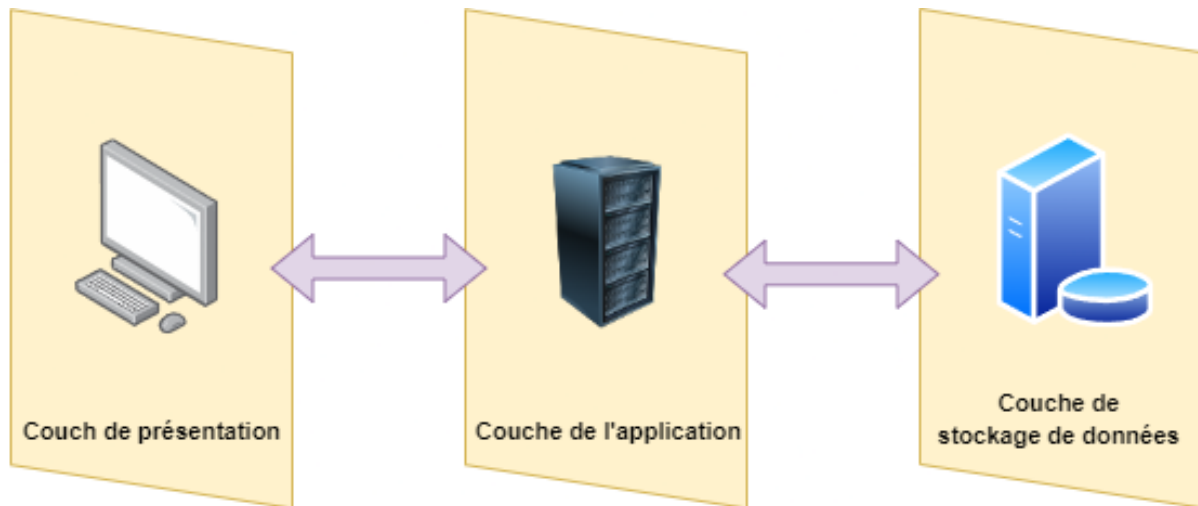


FIGURE 4.1 – L'architecture 3-tiers

L'architecture à 3 couches est préférée pour sa capacité à séparer clairement les différentes fonctionnalités de l'application et sa clarté dans la distinction entre les différentes couches. Cette approche facilite la maintenance, l'évolutivité et la réutilisation du code.

1.2 Architecture logique (Modèle MVC)

Le modèle MVC (Model-View-Controller) est un modèle de conception utilisé pour les applications logicielles qui divise les diverses responsabilités de l'application en trois éléments distincts :

- **Le Modèle (Model) :** Il symbolise le niveau de données de l'application. Il gère la logique métier et les échanges avec la base de données ou les services externes.
- **La Vue (View) :** Elle présente la partie de présentation de l'application. Elle a pour but de présenter visuellement les données et d'interagir avec l'utilisateur.
- **Le Contrôleur (Controller) :** Il présente le niveau de contrôle de l'application. Son rôle consiste à gérer les interactions des utilisateurs et à orchestrer les actions à réaliser. Le contrôleur établit une communication avec le modèle afin de récupérer ou de mettre à jour les données ainsi qu'avec la vue pour visualiser les résultats.

Le modèle MVC offre plusieurs avantages pour la conception et la maintenance d'applications logicielles. Tout d'abord, il permet une séparation claire des préoccupations qui facilite la gestion du code et la réutilisation des composants. En divisant les responsabilités de l'application en trois éléments distincts, chaque élément peut être développé et testé de manière autonome pour simplifier la collaboration et la gestion du code.

La Figure 4.2 illustre le modèle MVC.

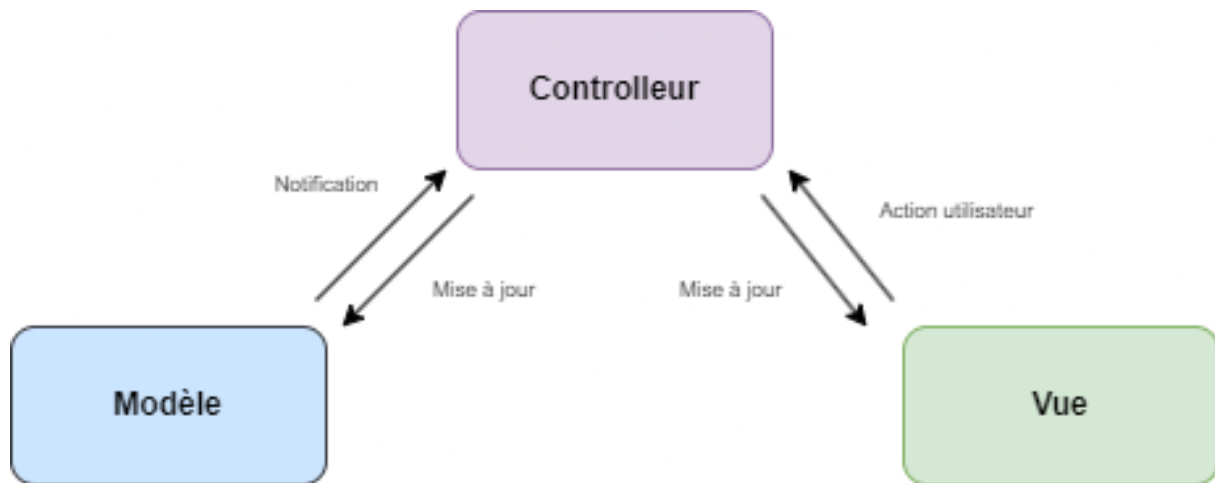


FIGURE 4.2 – L'architecture du modèle MVC

1.3 Architecture applicative

Dans notre architecture, nous avons divisé les différentes parties en fonction des serveurs utilisés. Voici les différentes parties de notre architecture :

- **La couche de présentation (Navigateur) :** Il s'agit de l'endroit où l'application Next.js s'exécute dans le navigateur de l'utilisateur.
- **La couche de logiciel de l'application :** Cette couche a été divisée en 2 parties.
 - **Runtime Node.js :** Il s'agit de l'endroit où le code Next.js côté serveur s'exécute sur un serveur Node.js.
 - **Runtime Edge :** Il s'agit de l'endroit où les fonctions Vercel Edge s'exécutent sur un serveur dédié aux fonctions Edge.
- **La couche de stockage de données :** Cette couche a été divisée en 3 parties.
 - **Postgres :** Il s'agit de l'endroit où la base de données Supabase Postgres réside sur un serveur dédié à la base de données.
 - **Stockage :** Il s'agit de l'endroit où le stockage Supabase réside sur un serveur AWS dédié au stockage.
 - **Base de données vectorielles :** Il s'agit de l'endroit où la base de données vectorielles Pinecone réside sur un serveur dédié à la base de données vectorielles.

En divisant notre structure en fonction des serveurs utilisés, nous avons réussi à séparer les différentes parties de l'application en les divisant en tiers puis en subdivisant chaque niveau en serveurs. De cette façon, chaque serveur peut être administré de manière autonome qui permet aux développeurs de se focaliser sur leur domaine de spécialisation respectif pour simplifier la

maintenance, l'évolution et la réutilisation du code.

On peut représenter cette architecture via la Figure 4.3.

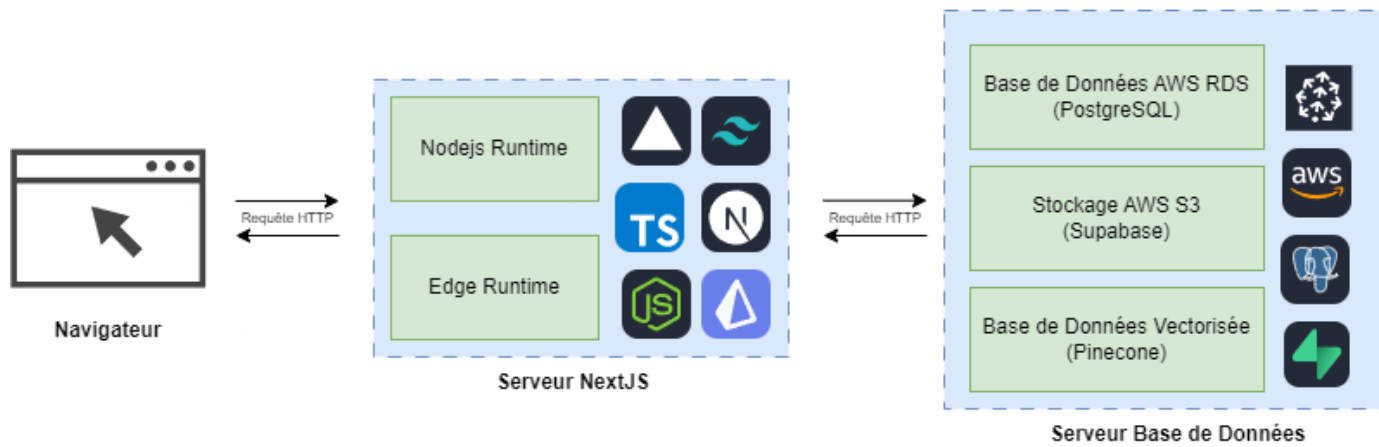


FIGURE 4.3 – L'architecture applicative

Pour représenter le déploiement de notre plateforme, nous avons utilisé le diagramme de déploiement. Ce diagramme est une vue statique qui illustre l'utilisation de l'infrastructure physique par le système ainsi que la répartition des composants du système et les relations entre eux.

La Figure 4.4 illustre le diagramme de déploiement de notre application.

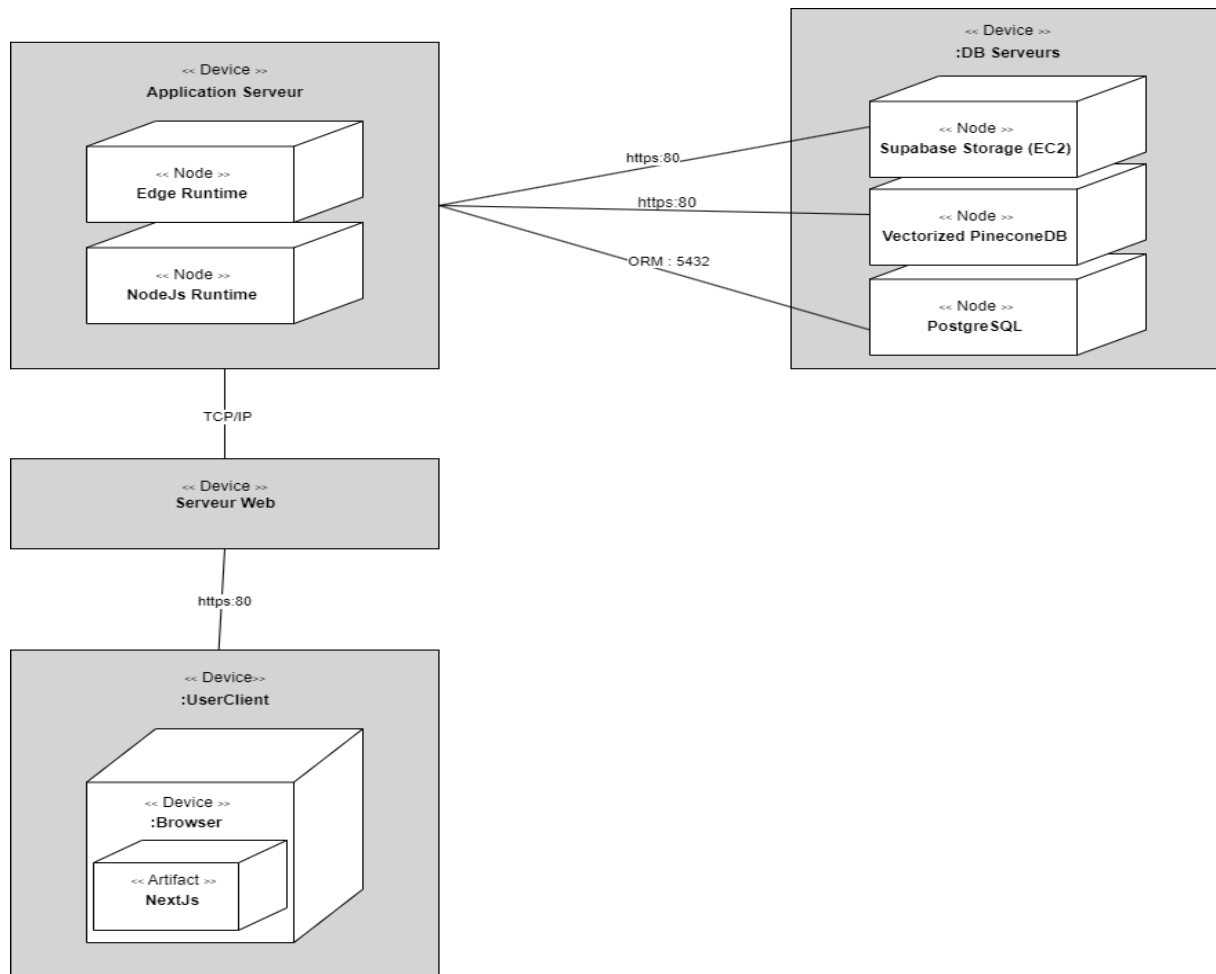


FIGURE 4.4 – Diagramme de déploiement

2 Conception de la base de donnée

La conception d'une base de données implique de structurer les données selon un modèle spécifique qui définit leur organisation, leur stockage et leurs relations. Dans cette optique, nous débuterons par la présentation du modèle conceptuel de données (**MCD**). Ensuite, nous nous tournerons vers le modèle logique de données (**MLD**).

2.1 Modèle conceptuel de données

Un modèle conceptuel de données (MCD) permet de représenter les entités et les relations entre elles dans un système d'information.

La Figure 4.5 illustre le modèle conceptuel de notre application.

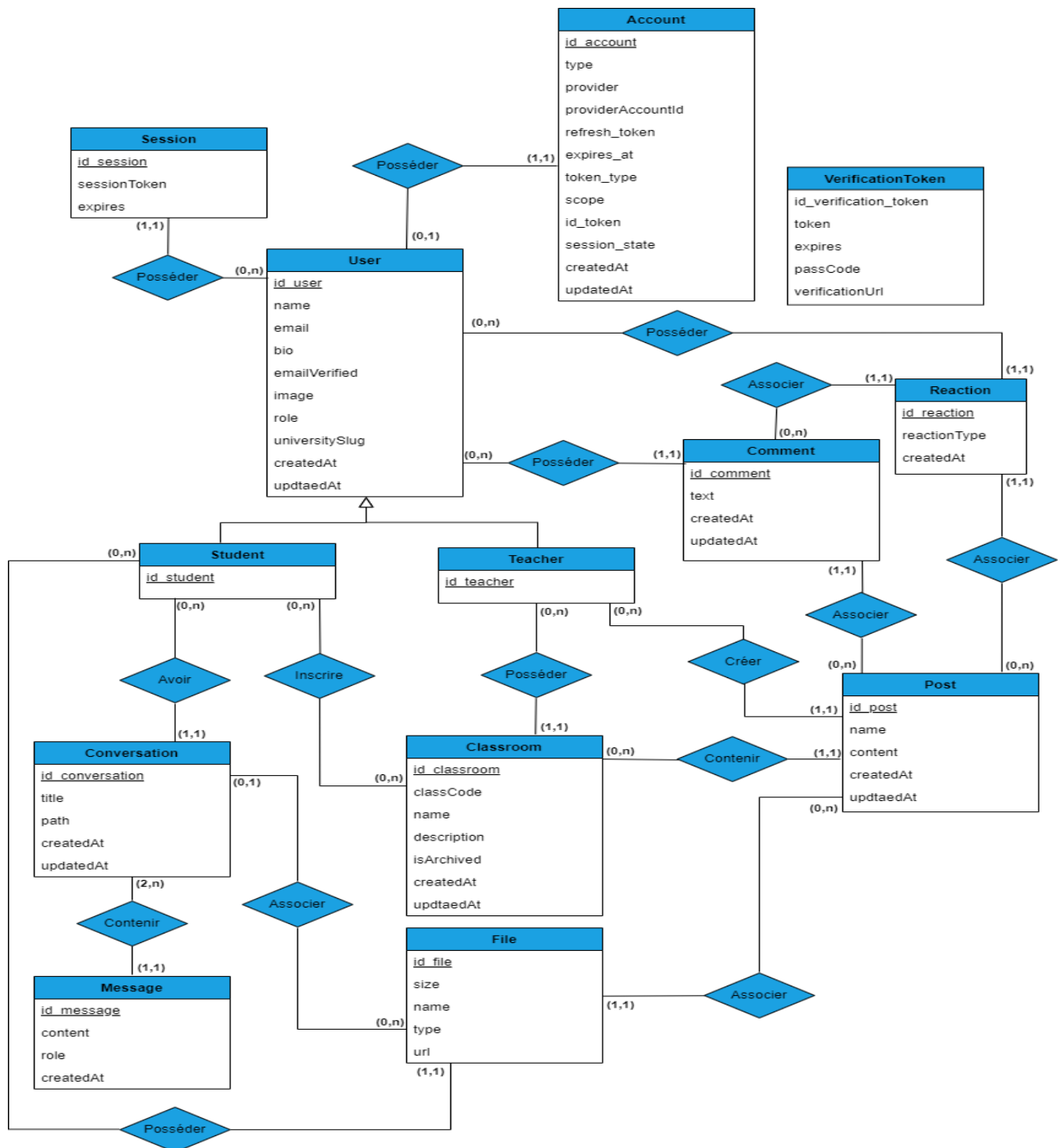


FIGURE 4.5 – Modèle conceptuel de données (MCD)

2.2 Modèle logique de données

Pour organiser un modèle MCD, différents modèles logiques de données peuvent être utilisés. Dans notre projet, nous avons choisi d'utiliser le modèle relationnel car il répond bien à nos

besoins non fonctionnels. En suivant les règles de transformation du modèle entité-association vers un modèle relationnel, nous obtiendrons le schéma relationnel suivant :

Account (id_account , type , provider , providerAccountId , refresh_token , access_token , expires_at , token_type , scope , id-token , session_state , createdAt , updatedAt , #id_user)

Session (id_session , sessionToken , expires , #id_user)

User (id_user , name , email , bio , emailVerified , image , rôle , universitySlug , createdAt , updatedAt)

VerificationToken (id_verification_token , token , expires , passCode , verificationUrl)

Teacher (id_teacher , #id_user)

Student (id_student , #id_user)

Classroom (id_classroom , classCode , name , description , isArchived , createdAt , updatedAt , #id_teacher)

Post (id_post , name , content , createdAt , updatedAt , #id_teacher , #id_classroom)

File (id_file , size , name , type , url , #id_post , #id_student)

Comment (id_comment , text , createdAt , updatedAt , #id_user , #id_post)

Reaction (id_reaction , reactionType , createdAt , #id_user , #id_comment , #id_post)

Conversation (id_conversation , title , path , createdAt , updatedAt , #id_student , #id_file)

Message (id_message , content , role , createdAt , #id_conversation)

Inscrire (#id_student , #id_classroom)

3 Conception de la vue statique : Diagramme de classes

Les diagrammes de classes sont des représentations statiques et structurelles utilisées pour présenter les classes, les interfaces et leurs relations dans un système logiciel. En effet, les diagrammes de classes sont une partie importante de la conception détaillée d'un système logiciel car ils permettent de présenter visuellement la structure du système et les interactions entre ses différents composants.

La Figure 4.6 illustre le diagramme de classe de notre application.

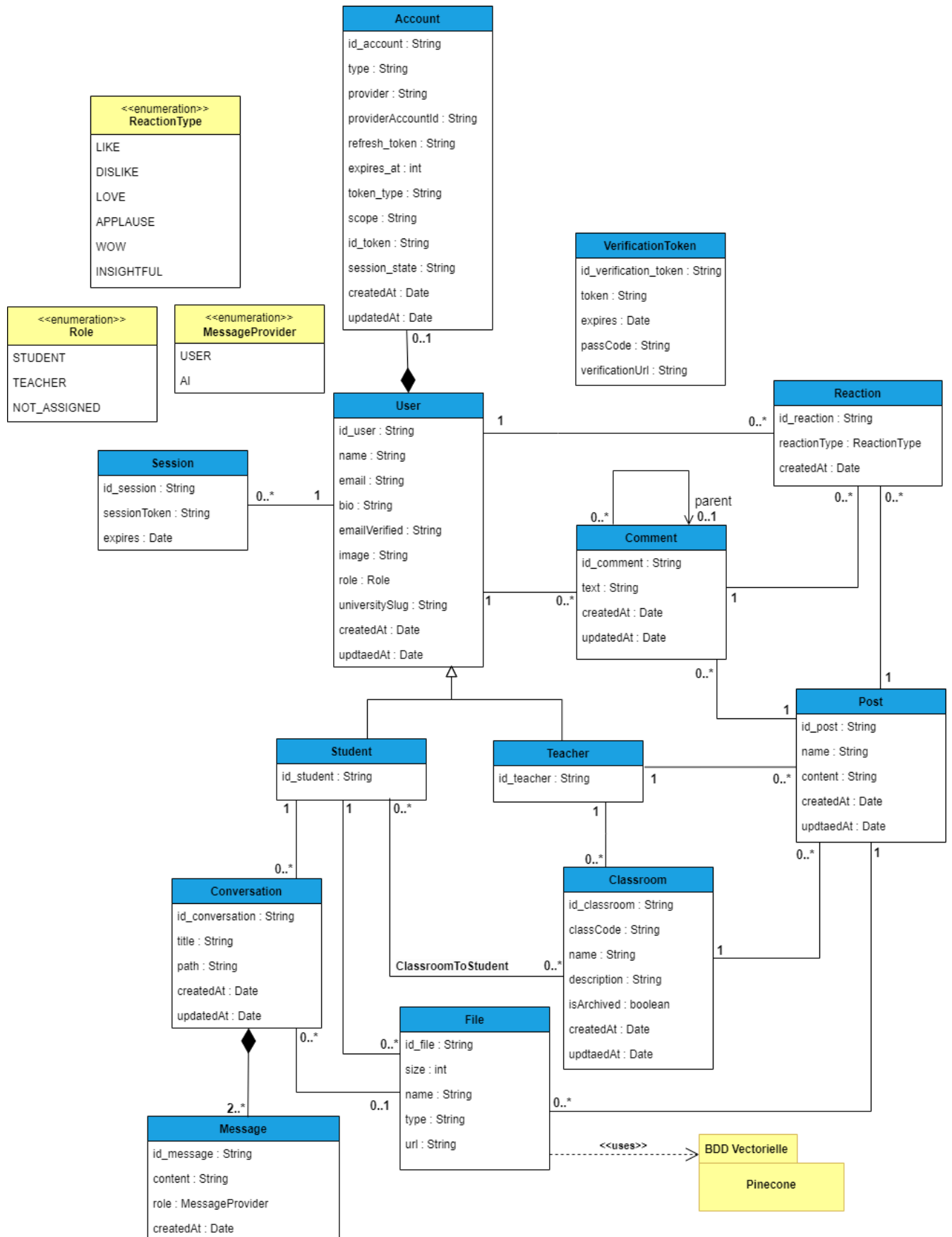


FIGURE 4.6 – Modèle conceptuel de données (MCD)

4 Conception de la vue dynamique

La vue dynamique illustre la manière dont les éléments d'un système logiciel interagissent et se comportent en mettant l'accent sur les séquences d'actions et les échanges de messages entre eux.

4.1 Diagramme de séquence de conception

Dans cette partie, nous mettons en évidence la progression des opérations et des échanges entre les différentes couches de l'application en utilisant des diagrammes de séquence de conception.

- **Diagramme de séquence de conception « S'identifier »**

La Figure 4.7 illustre le diagramme de séquence de conception « **S'identifier** »

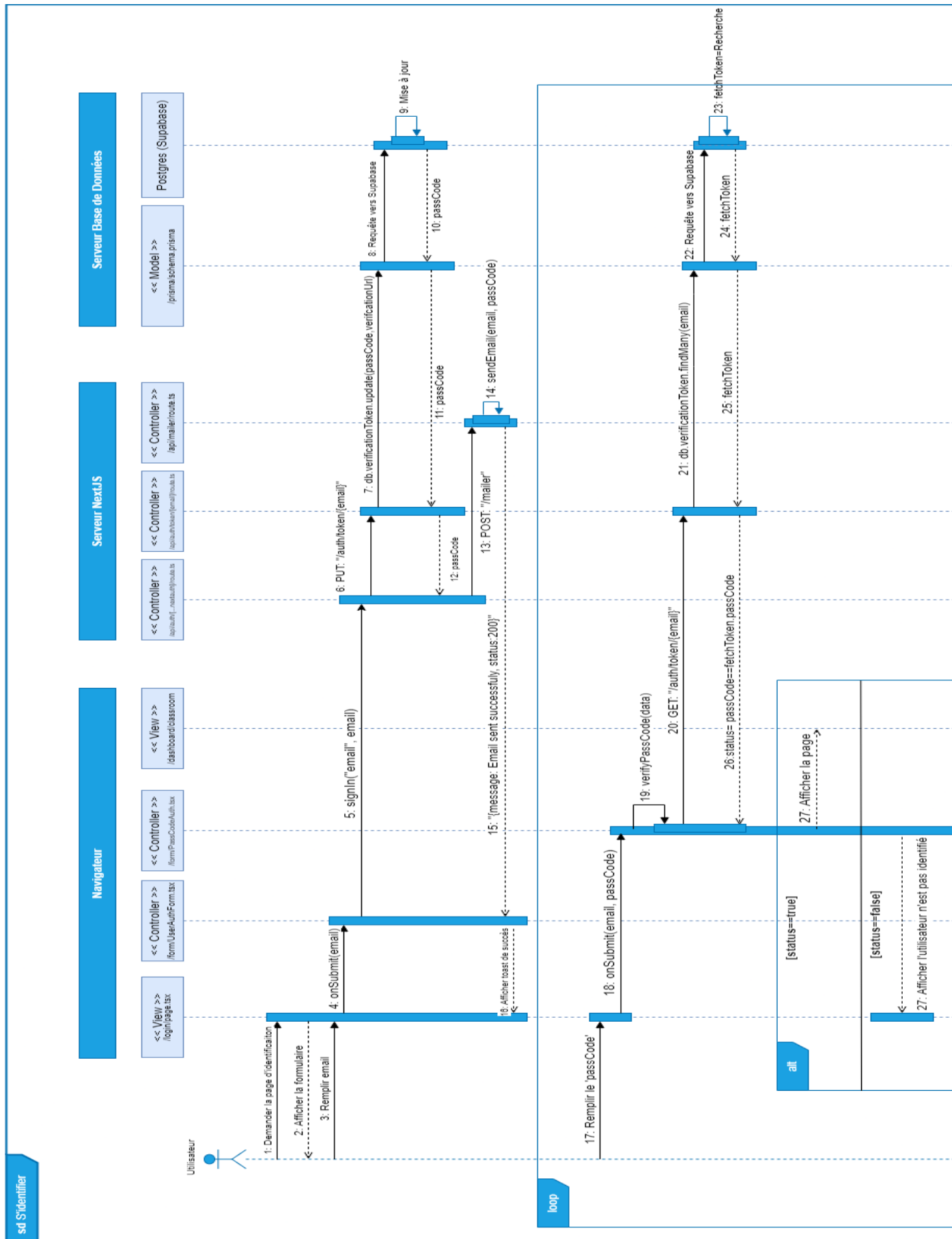


FIGURE 4.7 – Diagramme de séquence de conception « S'identifier » 52

• Diagramme de séquence de conception « Créer classroom »

La Figure 4.8 illustre le diagramme de séquence de conception « Créer classroom »

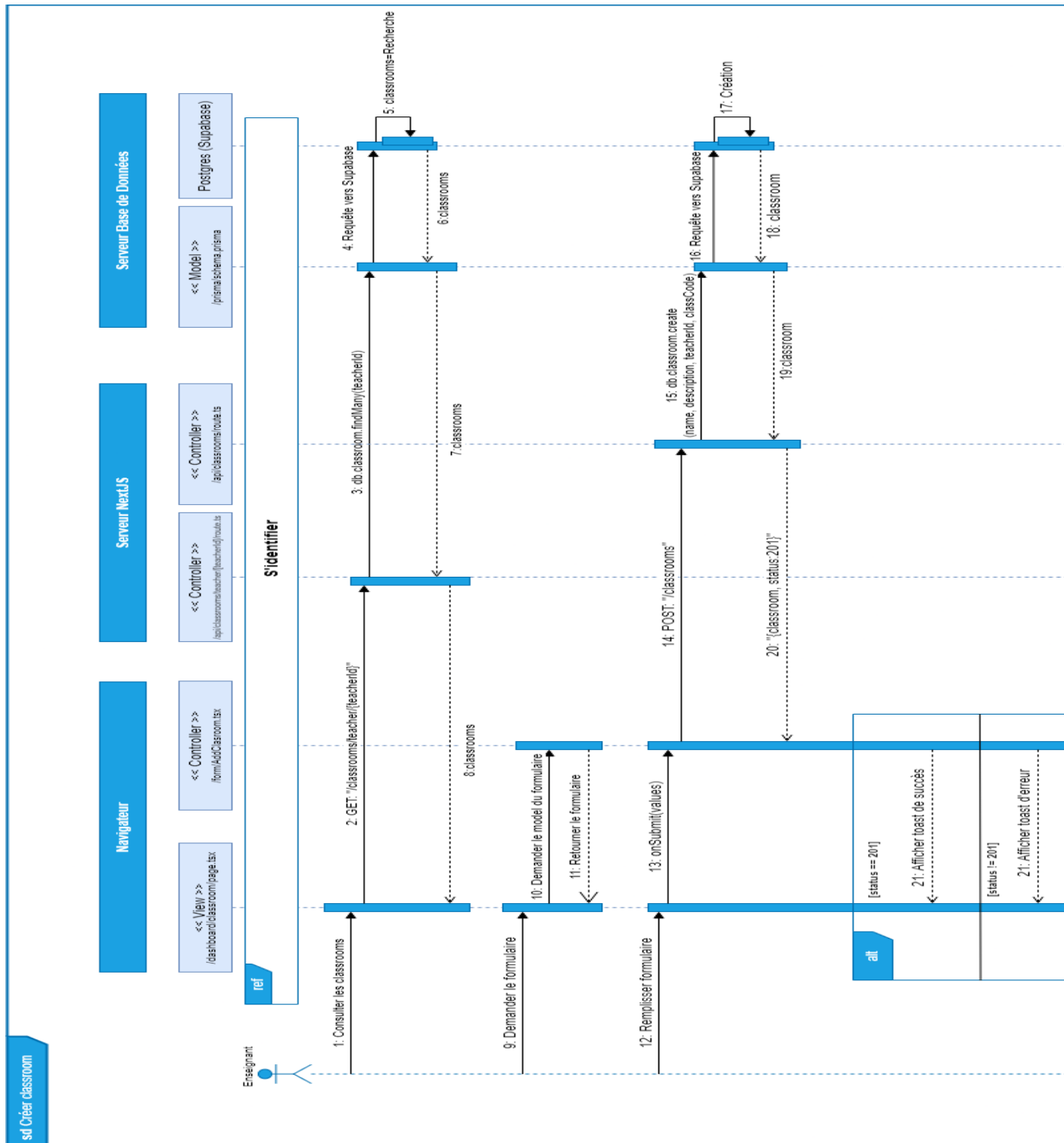


FIGURE 4.8 – Diagramme de séquence de conception « Créer classroom »

• Diagramme de séquence de conception « Démarrer chat »

La Figure 4.9 illustre le diagramme de séquence de conception « Démarrer chat »

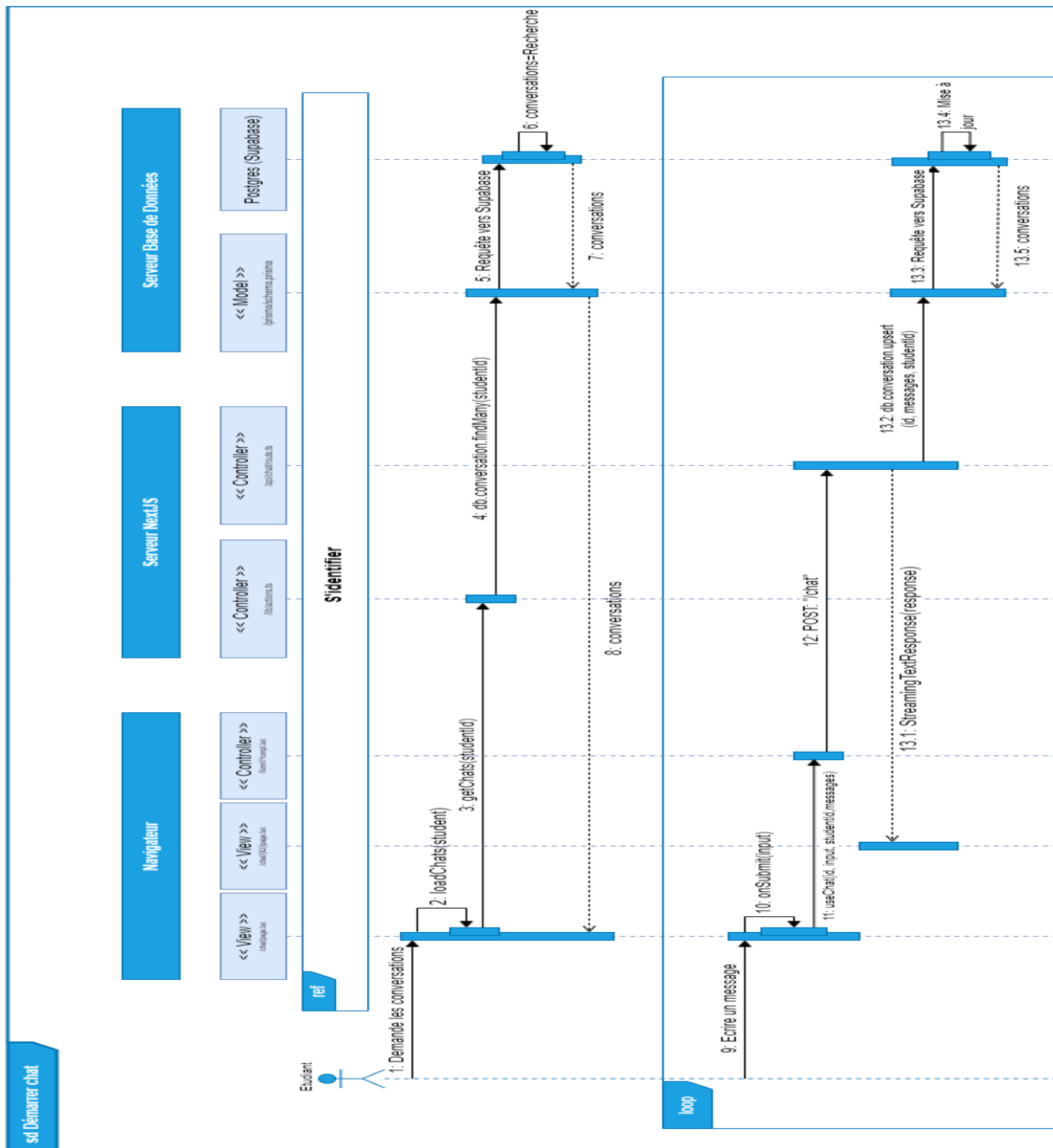


FIGURE 4.9 – Diagramme de séquence de conception « Démarrer chat »

4.2 Diagramme d'activité « Modifier profil »

Le diagramme d'activité illustre le déroulement des opérations et des actions au sein d'un processus. Il offre la possibilité de représenter le processus séquentiel des étapes d'un système ainsi que les choix effectués à chaque étape.

La Figure 4.10 illustre le diagramme d'activité pour le cas d'utilisation «**Modifier profil**» sachant que l'utilisateur peut être soit un étudiant, soit un enseignant.

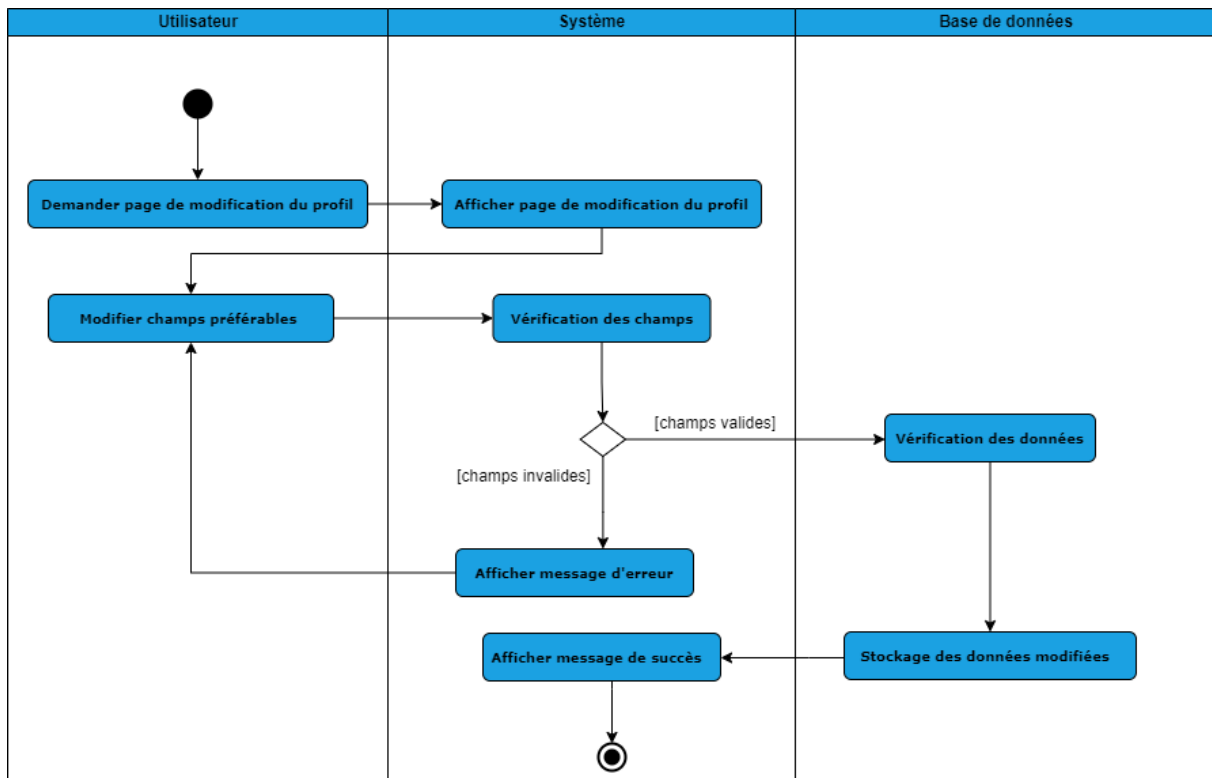


FIGURE 4.10 – Diagramme d'activité « Modifier Profil »

4.3 Diagramme d'états-transitions « Archiver classroom »

Le diagramme d'états-transitions illustre les divers états d'un objet ou d'un système. Il permet de représenter les changements d'état qui peuvent se produire en réponse à des événements spécifiques.

La Figure 4.11 illustre le diagramme d'états-transitions du cas d'utilisation «**Archiver classroom**» pour l'état du classroom qui peut être dans l'état «**désarchivé**» ou l'état «**archivé**».

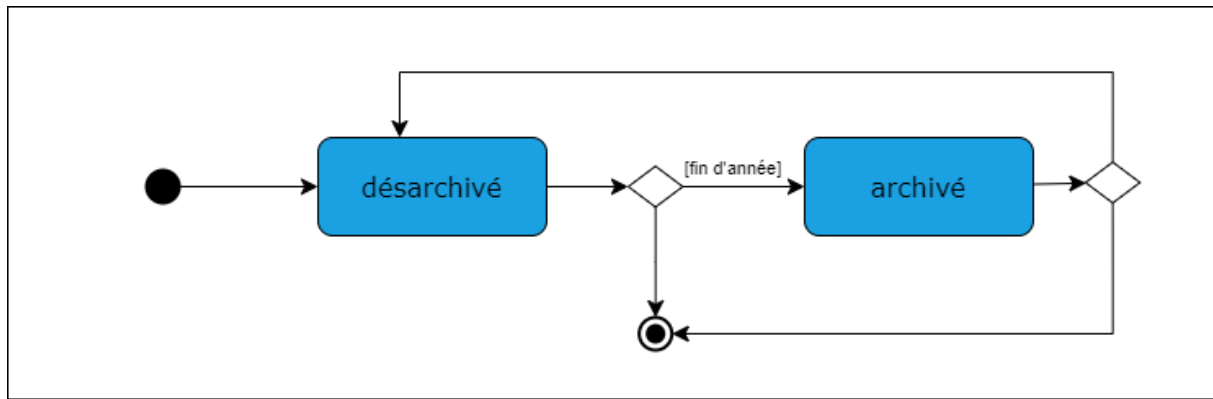


FIGURE 4.11 – Diagramme d'activité « Modifier Profil »

4.4 Diagramme de timing

Un diagramme de timing (temps) est un genre de diagrammes d'interaction qui est spécialement conçu pour prendre en compte les contraintes temporelles d'un logiciel.

Dans ce type de diagramme, les messages représentent les communications ou les interactions entre les différentes entités du système. Ils peuvent être synchrones, asynchrones ou des appels de méthodes. En plus, le diagramme de timing peut inclure aussi des informations sur les états des entités impliquées à des différents moments de l'exécution du système.

La Figure 4.12 illustre le diagramme de timing du cas d'utilisation « **Démarrer chat** » et plus spécifiquement lors de l'envoi d'un message.

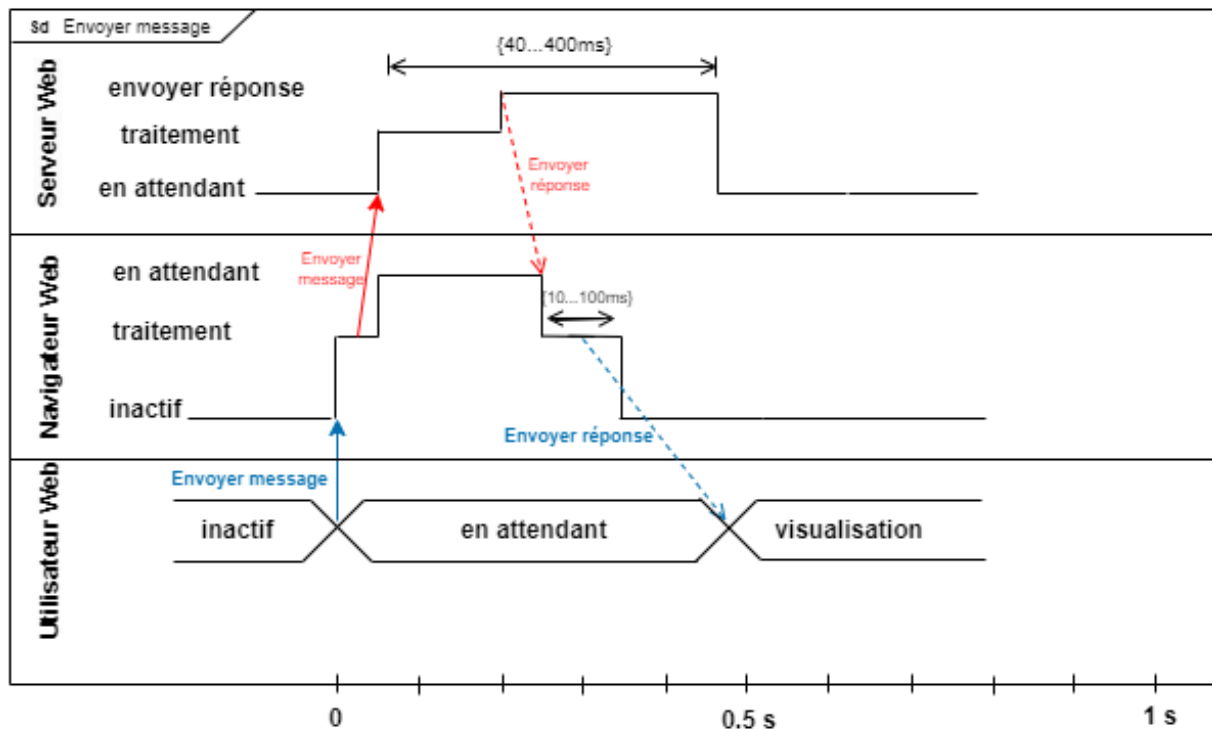


FIGURE 4.12 – Diagramme de timing « Démarrer chat »

5 Conception graphique : Maquettage

La conception graphique, également appelée conception visuelle, joue un rôle essentiel dans la création de produits visuels tels que les sites web, les applications mobiles, les interfaces utilisateur, les affiches, les logos, les brochures (annonces) et d'autres éléments visuels.

Dans ce qui suit, nous allons présenter quelques exemples de maquettes de notre application afin de satisfaire les besoins et les expériences des utilisateurs sur une page spécifique.

Conclusion

Au cours de ce chapitre, nous avons exposé en détail les étapes clés de la conception de notre projet. En effet, nous avons commencé par la présentation de l'architecture de notre application. Puis, nous avons défini le modèle conceptuel et le modèle logique de notre base de données. Ensuite, nous avons exposé la conception logicielle de notre application en spécifiant la vue statique via le diagramme de classes et la vue dynamique via les diagrammes de séquence de conception, un diagramme d'activité, un diagramme d'états-transitions et un diagramme de

timing. Enfin, nous avons présenté quelques maquettes pour la conception graphique.

CHAPITRE 5

RÉALISATION

Introduction

Dans ce chapitre final, nous allons présenter les divers outils et framework employés pour la création de notre application. Aussi, nous présenterons également quelques interfaces de notre application.

1 Environnement et outils de travail

Dans cette section, nous allons présenter les différents outils matériels et logiciels utilisés pour la mise en œuvre de notre application.

1.1 Environnement matériel

Pour mettre en place notre solution, nous avons utilisé deux ordinateurs portables. Le tableau 5.1 illustre leurs caractéristiques.

TABLE 5.1 – Caractéristiques de l’environnement matériel








	Ordinateur 1	Ordinateur 2
Marque	HP	Lenovo
Processeur	Intel i7 10 ^{ème} génération	Intel i5 8 ^{ème} génération
Ram	12 GO	8 GO
Disque Dur	512 GO SSD	512 GO SSD
Système d’exploitation	Windows 10	Kubuntu LTS 22

1.2 Environnement logiciel

Le tableau 5.2 illustre la liste des outils utilisés lors du développement de notre application web.

TABLE 5.2 – Liste des outils utilisé lors du développement de l’application

Outil/Technologie	Description
Visual Studio Code 	Visual Studio Code est un éditeur de code source développé par Microsoft reconnu pour sa légèreté, sa robustesse et ses extensions.
Postman 	Postman est une plateforme de développement API qui permet de créer, tester et déboguer des API de manière efficace.
Git 	Git est un système de contrôle de version distribué et largement utilisé pour suivre les changements dans le code source.
GitHub 	GitHub est une plateforme de développement logiciel basée sur Git qui offre des fonctionnalités de collaboration et de gestion de projets.
Draw.io 	Draw.io est un outil de création de diagrammes en ligne qui permet de créer des diagrammes de manière intuitive et collaborative.
Excalidraw 	Excalidraw est un outil de prototypage de l’interface utilisateur en ligne qui permet de créer des wireframes de manière simple et rapide.
React.js 	React.js est une bibliothèque JavaScript pour la création d’interfaces utilisateur interactives et dynamiques.
Next.js 	Next.js est un framework JavaScript React qui permet de construire des applications web performantes avec une expérience de développement simplifiée.
Tailwind CSS 	Tailwind CSS est une bibliothèque CSS utilitaire qui permet de concevoir rapidement des interfaces utilisateur modernes et personnalisées.

Prisma 	Prisma est un ORM (Object-Relational Mapping) qui facilite l'interaction avec la base de données en utilisant un langage de requête TypeScript sécurisé.
PostgreSQL 	PostgreSQL est un système de gestion de base de données relationnelles robuste et performant.
Supabase 	Supabase est une plateforme de développement intégrée qui offre une base de données PostgreSQL hébergée et d'autres fonctionnalités de backend.
Node.js 	Node.js est un environnement d'exécution JavaScript côté serveur qui permet d'exécuter du code JavaScript en dehors du navigateur.
TypeScript 	TypeScript est un langage basé sur JavaScript développé par Microsoft avec un typage statique optionnel. Il facilite la détection précoce et la correction des erreurs lors du développement.
Vercel 	Vercel est une plateforme de déploiement qui permet de déployer des applications frontend et backend de manière rapide, simple et évolutive.
Langchain 	Langchain est un framework développé dans le but de faciliter la création d'applications en utilisant des modèles de grands modèles de langage (LLM).
Pinecone 	Pinecone est une base de données vectorielle qui offre une infrastructure efficace pour stocker et manipuler des données vectorielles.

2 Framework Next.js

Notre application web est développée en utilisant le framework Next.js qui est construit sur la bibliothèque ReactJS et qui offre des fonctionnalités supplémentaires pour la création d'appli-

cations web modernes. Next.js est un framework full stack qui permet de créer des applications web performantes et optimisées en facilitant la création d'interfaces utilisateur.

Avec Next.js, nous pouvons utiliser le dossier **"app"** pour gérer le routage de notre application. Il permet de créer des routes dynamiques, des groupes de routes et des routes imbriquées en créant simplement des dossiers.

La Figure 5.1 illustre la structure du dossier **"app"**.

En outre, Next.js offre également un dossier spécial appelé **"api"** pour créer des endpoints API pour la partie backend de notre application.

Cela permet de créer facilement des API REST pour notre application.

De plus, il est important de noter que Next.js propose également les **"Server Actions"** qui peuvent également jouer un rôle similaire à celui du dossier **"api"** pour créer des endpoints API.

Les Figures 5.2 et 5.3 illustrent la structure des dossiers **"api"** et **"actions"**.

Next.js simplifie ainsi la création d'applications web full stack en fournissant une solution complète pour la gestion du routage et de la création d'API.

3 Implementation du Modèle LLM

3.1 API GroqCloud

Dans notre plateforme, nous avons mis en œuvre le modèle Mixtral of Experts (MoE) pour répondre à nos besoins en utilisant l'API de Groq, une société américaine spécialisée dans l'intelligence artificielle qui développe un accélérateur d'IA circuit intégré spécifique à l'application sous le nom de **Language Processing Unit (LPU)**. Cette API nous permet d'accéder aux différentes fonctionnalités du modèle MoE afin de créer un chatbot interactif capable de répondre aux questions des étudiants en se basant sur le contexte des documents partagés par eux-mêmes ou par leurs enseignants ainsi que sur des ressources externes. Cette fonctionnalité permettra aux étudiants d'obtenir des réponses rapides et précises à leurs questions et d'améliorer leur expérience utilisateur sur notre plateforme.

La figure 5.4 illustre l'intégration de l'API de Groq dans notre plateforme.