

**UNIVERSITE DE FIANARANTSOA**

**ECOLE NATIONALE D’INFORMATIQUE**

MEMOIRE DE FIN D’ETUDES POUR L’OBTENTION DE DIPLOME DE LICENCE PROFESSIONNELLE

**Mention: Informatique**

**Parcours: Informatique Generale**

Intitule

**Conception et realisation d’une plateforme de communication et d’echange en ligne pour les personnels**

Presente le :

Par **Monsieur RAKOTOVAO Mirindrampitia Diamondraniaina**

**Membres de jury:**

**President:**

**Examinateur:**

**Rapporteurs:**

Annee universitaire: **2023-2024**

**CURRICULUM VITAE**

**RAKOTOVAO Mirindrampitia Diamondraniaina**

Né le 24 Mars 2004 à Fianarantsoa

Adresse Physique : Lot IE 20 Tsianolondroa Fianarantsoa (301)

Adresse Téléphonique : +261 34 95 062 77

Adresse Electronique : mirindrampitirakotovao@gmail.com

**2023 – 2024 :** Ecole Nationale d’Informatique (ENI)

Etudiant en Troisième année de la Formation en Licence Professionnelle.

Parcours : Informatique Generale (IG)

**2022 – 2023 :** Ecole Nationale d’Informatique (ENI)

Etudiant en Deuxième année de la Formation en Licence Professionnelle.

Parcours : Informatique Generale (IG)

**2021 – 2022 :** Ecole Nationale d’Informatique (ENI)

Etudiant en Première année de la Formation en Licence Professionnelle.

Parcours : Informatique Generale (IG)

**2021 – 2022 :** ITTI Madagascar Fianarantsoa

Obtention du Diplôme C2 American English

**2019 – 2020 :** Lycée Prive Liantsoa School Ambatolahikosoa

Obtention du Baccalauréat série D

**2018 – 2019 :** Lycée Prive Liantsoa School Ambatolahikosoa

Obtention du Baccalauréat série A1

**2023 – 2024 :** Stage pratique au sein de l’Universite ACEEM Antananarivo

Thème traité : « Conception et realisation d’une plateforme en ligne pour les personnels », avec les technologies ReactJS et ExpressJS

Durée : 3 mois

Travaux Pratiques au sein de l’Ecole Nationale d’Informatique :

* Réalisation de deux projets avec JAVA
* Réalisation d’un projet avec React Native / Firebase
* Réalisation d’un projet avec NodeJS
* Réalisation de trois projets en Cybersecurite
* Réalisation de deux projets Admin Sys (Linux et Windows)

**2022 – 2023 :** Stage pratique au sein de la Region Haute Matsiatra Fianarantsoa

Thème traité : « Conception et Réalisation d’une plateforme d’echange d’idees pour le developpement de la Region Haute Matsiatra », avec les technologies ReactJS et NodeJS

Durée : 3 mois

Travaux Pratiques au sein de l’Ecole Nationale d’Informatique :

* Réalisation d’un projet avec PHP
* Réalisation d’un projet avec JAVA
* Réalisation d’un projet avec C#

**2021 – 2022 :** Projet pratique au sein de l’Ecole Nationale d’Informatique (ENI)

Thème traité : « Conception et réalisation d’une application pour la gestion de vente de materiels informatiques », sous Qt.

Durée : 1 mois

**COMPETENCES INFORMATIQUES**

* **Systèmes d’exploitation :** Windows (7, 10, 11) - Linux (Ubuntu, Kali, Mint)
* **Langages de programmation :** C++, C#, Java
* **Technologie web :** HTML5, CSS3, JavaScript, NodeJS, PHP
* **Framework :** Laravel, Express, Tailwind CSS,NodeJS
* **Bibliothèques :** ReactJS, VueJS, Typescript
* **Systèmes de gestion de base de données :** MySQL, ORACLE, SQLite, Sequelize, Prisma, Firebase, MongoDB
* **Méthode de conception :** Merise 2
* **Langage de modélisation :** UML
* **Outils de développement :** Visual Studio Code, Eclipse, Code Blocks, WebStorm
* **Outils bureautiques :** Microsoft Word, Excel, Power Point

**COMPETENCES LINGUISTIQUES**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **APTITUDE** | | | |
| **COMPREHENSION** | | **EXPRESSION** | |
| **LANGUES** | **ORALE** | **ECRITE** | **ORALE** | **ECRITE** |
| **FRANÇAIS** | TB | TB | TB | TB |
| **ANGLAIS** | B | B | B | B |
| **ALLEMAND** | AB | AB | AB | AB |

**DIVERS ET ACTIVITES EXTRA-CURRICULAIRES**

* **Sports :** Natation
* **Musique :** Guitares, Piano, Saxophone, Flute

SOMMAIRE

REMERCIEMENTS

Je voudrais remercier en premier lieu, Dieu Tout Puissant de m’avoir donné force et courage durant la formation pratique dans laquelle j’ai pu acquérir des expériences professionnelles.

Je tiens aussi à exprimer mes sincères remerciements à tout ce qui ont contribué à l’élaboration de cette mémoire de fin d’étude et au bon déroulement de mon stage :

* Monsieur HAJALALAINA Aimé Richard, Docteur HDR, Président de l’Université de Fianarantsoa, qui a bien organisé l’année universitaire ;
* Monsieur MAHATODY Thomas, Docteur HDR**,** Directeur de l’Ecole Nationale d’Informatique, qui m’a permis d’accueillir toutes mes connaissances jusqu’à maintenant ;
* Monsieur RASATAMIARINTSOA Nomenjanahary Tefy,Responsable Informatique**,** mon encadreur professionnel, avec sa sympathie, son accueil chaleureux au sein de la société ;
* Monsieur GILANTE Gesazafy , Assistant d’Enseignement Supérieur et de Recherche, Chef de Parcours Informatique Générale au sein de l’ENI;
* President de jury
* Mon examinateur ;
* Mon encadreur pédagogique pour la prélecture, qui m’a beaucoup aidé dans l’accomplissement de tout ce travail et qui m’a suivi jusqu’à la fin du stage ;
* Tout le personnel de l’UACEEM Antananarivo qui m’a accueilli et soutenu avec joie durant les moments de stage passés au sein de la communaute;
* Les membres du corps professoral et administratif de l’Ecole Nationale d’Informatique, pour toutes les connaissances acquises et savoirs, durant la formation théorique ;
* Mes collegues de travail qui ont cree une ambiance merveilleuse dans l’environnement de travail

Mes proches qui m’ont toujours soutenu moralement, ainsi que financièrement durant mon stage jusqu’à l’aboutissement de cette mémoire de fin d’étude.

Un grand merci a tous !

LISTE DES FIGURES

[Figure 1 : Organigramme de l'ENI 17](#_Toc1933396059)

[Figure 2 : Architecture des études correspondant au système LMD 19](#_Toc1303998802)

[Figure 3 : Organigramme de l'UACEEM 27](#_Toc222594490)

[Figure 4 : Chronogramme d'activite 34](#_Toc42346955)

[Figure 5 : Structure en Y du 2TUP 41](#_Toc361701775)

[Figure 6 : Diagramme UML 43](#_Toc1395095010)

[Figure 7 : Diagramme des cas d'utilsiation 56](#_Toc415604326)

[Figure 8 : Diagramme de sequence de systeme de cas d'utilisation "Authentification" 69](#_Toc1102336476)

[Figure 9 : Diagramme de sequence de systeme pour le cas d'utilisation "Gestion de publication" 70](#_Toc1375485971)

[Figure 10 : Diagramme de sequence de systeme pour le cas d’utilisation «Reagir Publication» 71](#_Toc1498470041)

[Figure 11 : Diagramme de sequence systeme pour le cas d'utilisation "Commenter" 72](#_Toc505818375)

[Figure 12 : Diagramme de sequence systeme pour le cas d'utilisation "Depot de cours" 73](#_Toc432814807)

[Figure 13 : Diagramme de sequence systeme pour le cas d'utilisation "Gestion signalement" 74](#_Toc1141745858)

[Figure 14 : Diagramme de sequence systeme pour le cas d'utilisation "Gestion Emploi Du Temps" 75](#_Toc1451378758)

[Figure 15 : Diagramme de sequence systeme pour le cas d'utilisation "Messagerie" 76](#_Toc716332126)

[Figure 16 : Diagramme de sequence systeme pour le cas d'utilisation "Gestion des archives" 77](#_Toc1071065294)

[Figure 17 : Diagramme de sequence systeme pour le cas d'utilisation "Gestion Forum" 78](#_Toc1172437606)

[Figure 18 : Diagramme de sequence systeme pour le cas d'utilisation "Gestion Feedback" 79](#_Toc30656427)

[Figure 19 : Modele de domaine 80](#_Toc2145569400)

[Figure 20 : Architecture su systeme de l'application 82](#_Toc1141239993)

[Figure 21 : Diagramme de sequence de conception pour le cas d'utilisation "Authentification" 83](#_Toc745899511)

[Figure 22 : Diagramme de sequence de conception de cas d'utilisation "Gestion signalement" 84](#_Toc1314060740)

[Figure 23 : Diagramme de sequence de conception de cas d'utilisation "Gestion des cours - professeurs" 84](#_Toc1475021290)

[Figure 24 : Diagramme de sequence de conception du cas d'utilisation "Publier contenu" 85](#_Toc475151486)

[Figure 25 : Diagramme de sequence de conception du cas d'utilisation "Messagerie" 85](#_Toc1844503572)

[Figure 26 : Diagramme de sequence de conception du cas d'utilisation "Gerer emploi du temps" 86](#_Toc1141156685)

[Figure 27 : Diagramme de sequence de conception du cas d'utilisation "Reagir Contenu" 86](#_Toc745747650)

[Figure 28 : Diagramme de classe de conception du cas d'utilisation "Authentification" 88](#_Toc739159695)

[Figure 29 : Diagramme de classe de conception du cas d'utilisation "Gestion de cours" 88](#_Toc1194769723)

[Figure 30 : Diagramme de classe de conception du cas d'utilisation "Gestion emploi du temps" 89](#_Toc1537010693)

[Figure 31 : Diagramme de classe de conception du cas d'utilisation "Gestion de Feedback" 89](#_Toc783570239)

[Figure 32 : Diagramme de classe de conception du cas d'utilisation "Gestion utilisateurs" 90](#_Toc265862740)

[Figure 33 : Diagramme de classe de conception du cas d'utilisation "Messageries" 90](#_Toc688172787)

[Figure 34 : Diagramme de classe globale 91](#_Toc1203556294)

[Figure 35 : Diagramme de paquettage de l'application 91](#_Toc369522549)

[Figure 36 : Diagramme de deploiement de l'application 92](#_Toc1046046067)

[Figure 37 : Logo de Konektea selon le theme de la plateforme 95](#_Toc586163995)

[Figure 38 : Logo UACEEM Manakambahiny et logo CIP 96](#_Toc129442249)

[Figure 39 : Palette de couleur utilise dans l'integration de la plateformes 96](#_Toc481536728)

[Figure 40 : Installation de Figma-Linux 98](#_Toc616725252)

[Figure 41 : Interface d'acceuil Figma-Linux 99](#_Toc430731615)

[Figure 42 : Installation de Webstorm 100](#_Toc381394951)

[Figure 43 : Interface Webstorm 100](#_Toc1184369781)

[Figure 44 : Installation Visual Paradigm 17.2 101](#_Toc916489125)

[Figure 45 : Page d'acceuil Visual Paradigm 102](#_Toc1163609451)

[Figure 46 : Architecture de l'application 103](#_Toc2142757707)

[Figure 47 : Concept de Squelize 104](#_Toc766608509)

[Figure 48 : Message prive entre utilisateurs avec Socket.IO 105](#_Toc1755703889)

[Figure 49 : Creation d'un forum 105](#_Toc557655966)

[Figure 50 : Prototypage & Maquettage de l'application 106](#_Toc2141039836)

[Figure 51 : Page d'accueil KONEKTEA 106](#_Toc1009089175)

LISTE DES TABLEAUX

[Table 1 : Organisation du système de formation pédagogique de l’Ecole 18](#_Toc1135895554)

[Table 2 : Mention et parcours au sein de l’ENI 18](#_Toc1022454933)

[Table 3 : Liste des formations existantes à l’ENI 19](#_Toc912410108)

[Table 4 : Débouchés éventuels des jeunes diplômés 22](#_Toc1621074013)

[Table 5 : Equipe dirigeante de l'UACEEM 29](#_Toc1642517798)

[Table 6 : Moyens humains 32](#_Toc2119233081)

[Table 7 : Moyens materiels 32](#_Toc1368431356)

[Table 8 : Moyens logiciels 33](#_Toc43447316)

[Table 9 : Moyens materiels en possession 37](#_Toc1274102842)

[Table 10 : Moyens logiciels en possession 37](#_Toc1931015183)

[Table 11 : Comparaison des solutiions 39](#_Toc78992749)

[Table 12 : Comparaison entre MERISE et 2TUP 40](#_Toc1840139402)

[Table 13 : Comparaison entre Java et Javascript 44](#_Toc485065920)

[Table 14 : Comparaison entre PHPl et NodeJS 46](#_Toc1160214848)

[Table 15 : Comparaison entre React JS et Angular JS 48](#_Toc43666296)

[Table 16 : Comparaison entre ExpressJS et NestJS 49](#_Toc1246842763)

[Table 17 : Comparaison entre MySQL et MongoDB 50](#_Toc1455419836)

[Table 18 : Comparaison entre Figma et Adobe XD 51](#_Toc1691165563)

[Table 19 : Comparaison entre VS Code et Webstorm 52](#_Toc1465202208)

[Table 20 : Description textuelle de "Se connecter" 57](#_Toc132483804)

[Table 21 : Description textuelle "Déposer Cours" 58](#_Toc1512651740)

[Table 22 : Description textuelle "Gestion utilisateur" 59](#_Toc342994850)

[Table 23 : Description textuelle "Acces/voir KONEKTEA" 60](#_Toc224474741)

[Table 24 : Description textuelle "Gestion de profil" 61](#_Toc94383797)

[Table 25 : Description textuelle "Recevoir Notification" 62](#_Toc2095198647)

[Table 26 : Description textuelle "Envoyer Feedback / Recevoir Feedback" 63](#_Toc942959908)

[Table 27 : Description textuelle "Publier Forum" 65](#_Toc166800645)

[Table 28 : Description textuelle "Gestion de signalement" 66](#_Toc1884451613)

[Table 29 : Tableau de priorisation des cas d'utilsiation 68](#_Toc1961836444)

[Table 30 : Utilisation des couleurs 96](#_Toc1835716460)

LISTE DES ABREVIATIONS

INTRODUCTION GENERALE

Dans le contexte actuel, l'informatique et, plus particulièrement, l'internet jouent un rôle essentiel dans la transformation des secteurs d'activité, facilitant la communication, la gestion de l'information et la collaboration. Les applications web, devenues incontournables dans ce domaine, occupent désormais une place centrale dans le développement technologique.

Pour les étudiants de l’École Nationale d’Informatique de Fianarantsoa, un stage de fin d'année est exigé afin de valider leurs compétences et de les préparer au monde professionnel. Ce projet, en plus de son importance académique pour l'obtention de la Licence, constitue un véritable tremplin pour leur avenir professionnel.

Le projet que je mène porte sur la conception et la mise en œuvre d’une plateforme en ligne appelée **KONEKTEA**. Ce réseau social est spécifiquement destiné aux personnels de l’Université ACEEM, offrant des outils de communication et de collaboration adaptés à leurs besoins. Le projet s'appuie sur une méthodologie bien structurée, intégrant des outils et technologies modernes pour la modélisation, le développement et la gestion des bases de données.

Le développement de **KONEKTEA** s'articule autour de trois grandes phases : la présentation de la solution, l’analyse et la conception du système, et enfin sa mise en œuvre technique.

# PARTIE I : PRESENTATIONS

## CHAPITRE 1: PRESENTATION DE L’ECOLE NATIONALE D’INFORMATIQUE

### Information d’ordre générale

L’Ecole Nationale d’Informatique, en abrégé ENI, est un établissement d’enseignement supérieur rattaché académiquement et administrativement à l’Université de Fianarantsoa. Le siège de l’Ecole se trouve à Tanambao-Antaninarenina à Fianarantsoa. L’adresse pour la prise de contact avec l’Ecole est la suivante : Ecole Nationale d’Informatique (ENI) Tanambao, Fianarantsoa. Le numéro de sa boîte postale est 1487 avec le code postal 301. Téléphone : 034 05 733 36 ou 032 15 204 28. Son adresse électronique est la suivante : **eni@eni.mg**. Il dispose également d'un site web : **www.eni.mg**

### Missions et historiques

L’ENI se positionne sur l’échiquier socio-éducatif malgache comme étant le plus puissant secteur de diffusion et de vulgarisation des connaissances et des technologies informatiques.

Cette Ecole Supérieure peut être considérée aujourd’hui comme la vitrine et la pépinière des élites informaticiennes du pays.

De façon formelle, l’ENI était créée par le décret N° 83- 185 du 24 Mai 1983, comme étant le seul établissement Universitaire Professionnalisé au niveau national, destiné à former des techniciens et des Ingénieurs de haut niveau, aptes à répondre aux besoins et exigences d’Informatisation des entreprises, des sociétés et des organes implantés à Madagascar.

L’ENI a pour conséquent pour mission de former des spécialistes informaticiens compétents et opérationnels de différents niveaux notamment :

* en fournissant à des étudiants des connaissances de base en informatique ;
* en leur transmettant le savoir-faire requis, à travers la professionnalisation des formations dispensées et en essayant une meilleure adéquation des formations par rapport aux besoins évolutifs des sociétés et des entreprises ;
* en initiant les étudiants aux activités de recherche dans les différents domaines des Technologies de l’Information et de la Communication (TIC) ;

La filière de formation d’Analystes Programmeurs a été mise en place à l’Ecole en 1983, et a été gelée par la suite en 1996, tandis que la filière de formation d’ingénieurs a été ouverte à l’Ecole en 1986.

Une formation de troisième cycle a été ouverte à l’Ecole a été ouverte à l’Ecole depuis l’année 2003 – 2004 grâce à la coopération académique et scientifique entre l’Université de Fianarantsoa pour le compte de l’ENI et l’Université Paul Sabatier de Toulouse (UPST).

Cette filière avait pour objectif de former certains étudiants à la recherche dans les différents domaines de l’Informatique, et notamment pour préparer la relève des Enseignants-Chercheurs qui étaient en poste.

Pendant l’année 2007-2008, la formation en vue de l’obtention du diplôme de Licence Professionnelle en Informatique a été mise en place à l’ENI avec les deux parcours de formation :

* Génie Logiciel et base de Données.
* Administration des Système et réseaux.

La mise en place à l’Ecole de ces deux options de formation devait répondre au besoin de basculement vers le système Licence – Master – Doctorat (LMD).

En vue de surmonter les difficultés de limitation de l’effectif des étudiants accueillis à l’Ecole, notamment à cause du manque d’infrastructures, un système de « Formation Hybride » a été mise en place à partir de l’année 2010. Il s’agit en effet d’un système de formation semi présentielle et à distance avec l’utilisation de la visioconférence pour la formation à distance. Le système de formation hybride a été ainsi créé à Fianarantsoa ainsi qu’Université de Toliara. Cette formation est à l’origine du parcours Informatique Générale.

En 2023, une nouvelle mention Intelligence Artificielle (IA) a été ouvert au sein de l’Ecole pour répondre les besoins des entreprises. La formation est destinée aux étudiants titulaires du diplôme de licence (Bac +3) en Mathématiques ou en Statistiques ou en Informatique, etc. La mention IA comporte deux parcours :

* Gouvernance et Ingénierie de Données (GID),
* Objets connectés et Cybersécurité (OCC).

Le principe de l’enseignement pour le parcours GID osffre aux l’étudiants des compétences scientifiques et techniques spécialisées en Science de données. Pour le parcours OCC, les étudiants octroient la double spécialité premièrement en internet des objets et deuxièmement en cybersécurité. La formation de master est axée sur l’ensemble d’applications de l’Intelligence Artificielle.

### Organigramme institutionnel

L’organigramme de l’Ecole est inspiré des dispositions du décret N° 83-185 du 24 Mai 1983. L’ENI est administrée par un Conseil d’Ecole, et dirigée par un directeur nommé par un décret adopté en Conseil des Ministres. Le Collège des enseignants regroupant tous les enseignants-chercheurs permanents de l’Ecole est chargé de résoudre les problèmes liés à l’organisation pédagogique des enseignements. Le Conseil Scientifique propose les orientations pédagogiques et scientifiques de l’établissement, en tenant compte notamment de l’évolution du marché de travail et de l’adéquation des formations dispensées par rapport aux besoins des entreprises.

La figure 1 représente l’organigramme actuel de l’ENI.

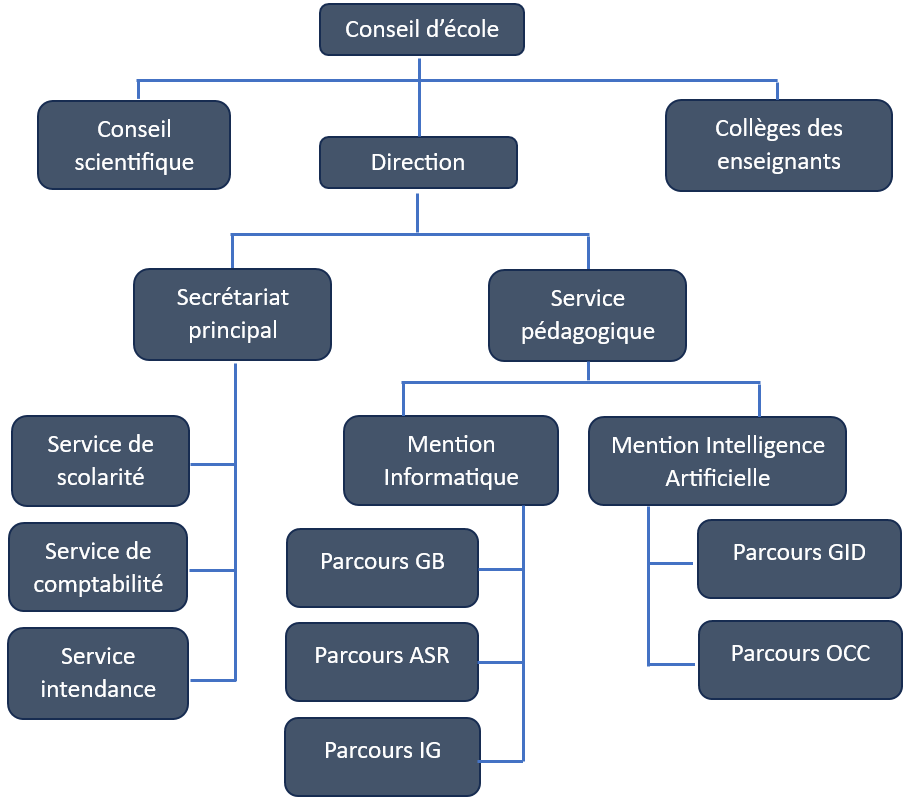


Figure 1: Organigramme de l'ENI

### Domaine de spécialisation

Les activités de formation et de recherche organisées à l’ENI portent sur les domaines suivants :

* Génie logiciel et Base de Données ;
* Administration des Systèmes et Réseaux ;
* Informatique Générale ;
* Modélisation informatique et mathématique des Systèmes complexes ;
* Intelligence artificielle.

Le tableau 1 décrit l’organisation du système de formation pédagogique de l’Ecole.

|  |  |
| --- | --- |
| Formation Théorique | Formation Pratique |
| * Enseignement théorique * Travaux dirigés * Travaux pratiques * Conférences | * Etude de cas * Travaux de réalisation * Projets/ Projets tutorés * Voyages d’Etudes * Stages en entreprise |

Table 1: Organisation du système de formation pédagogique de l’Ecole

### Architecture des formations pédagogiques

Le recrutement des étudiants à l’ENI se fait uniquement par voie de concours d’envergure nationale en première année. Les offres de formation organisées à l’Ecole ont été validées par la Commission Nationale d’Habilitation (CNH). Au sein de l’ENI, il existe deux mentions et cinq parcours. Le tableau 2 récapitule les mentions et les parcours au sein de l’Ecole :

|  |  |
| --- | --- |
| **Mention** | **Parcours** |
| **Informatique** | Génie logiciel et Base de Données (GB) |
| Administration des Systèmes et Réseaux (ASR) |
| Informatique Générale (IG) |
| **Intelligence Artificielle** | Gouvernance et Ingénierie de Données (GID) |
| Objets Connectés et Cyber sécurités (OCC) |

Table 2: Mention et parcours au sein de l’ENI

La figure 2 représente l’architecture des études correspondant au système LMD.

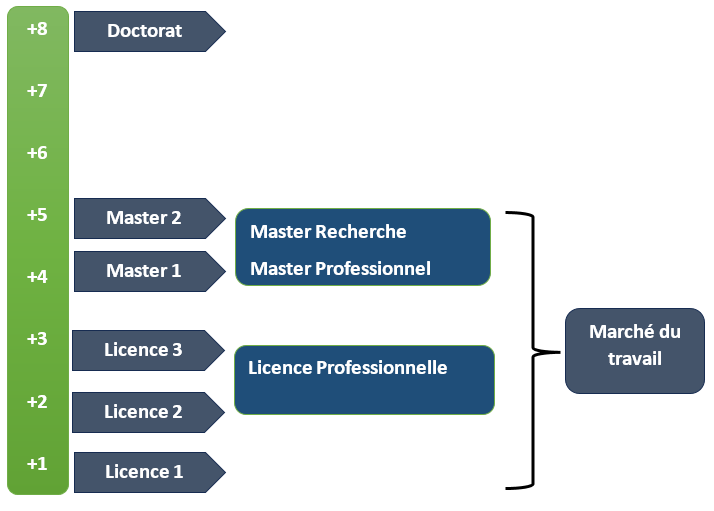


Figure 2: Architecture des études correspondant au système LMD

La licence peut avoir une vocation générale ou professionnelle. Le master peut avoir une vocation professionnelle ou de recherche. L’accès en première année de MASTER se fait automatiquement pour les étudiants de l’Ecole qui ont obtenu le diplôme de Licence Professionnelle. Le tableau 3 illustre la liste des formations existantes à l’ENI.

|  |  |  |
| --- | --- | --- |
|  | FORMATION | |
|  | LICENCE PROFESSIONNELLE | MASTER |
| Condition admission | Par voie de concours | Par voie de concours pour la mention IA |
| Condition d’accès | Bac de série C, D ou Technique | Être titulaire de licence professionnelle |
| Durée de Formation | 3 ans | 2 ans |
| Diplôme délivré | Diplôme de Licence Professionnelle | Diplôme de Master Professionnel  Diplôme de Master Recherche |

Table 3: Liste des formations existantes à l’ENI

Le Master Recherche permet à son titulaire de poursuivre directement des études en doctorat et de s’inscrire directement dans une Ecole Doctorale.

Les étudiants diplômés de l’Ecole sont plutôt bien accueillis dans les instituts universitaires étrangères (Canada, Suisse, France, …)

### Relation de l’ENI avec les organismes externes

Les stages effectués chaque année par les étudiants mettent l’Ecole en rapport permanent avec plus de 400 entreprises et organismes publics, semi-publics et privés, nationaux et internationaux. L’Ecole dispose ainsi d’un réseau d’entreprises, de sociétés et d’organismes publics et privés qui sont des partenaires par l’accueil en stage de ses étudiants, et éventuellement pour le recrutement après l’obtention des diplômes par ces derniers. Les compétences que l’Ecole cherche à développer chez ses étudiants sont l’adaptabilité, le sens de la responsabilité, du travail en équipe, le goût de l’expérimentation et l’innovation.

En effet, la vocation de l’ENI est de former des licenciés et des ingénieurs de niveau MASTER avec des qualités scientifiques, techniques et humaines reconnues, capables d’évoluer professionnellement dans des secteurs d’activité variés intégrant l’informatique. Les stages en milieu professionnel permettent de favoriser une meilleure adéquation entre les formations à l’Ecole et les besoins évolutifs du marché de l’emploi.

Parmi les sociétés, les entreprises et les organismes partenaires de l’Ecole, on peut citer : ACCENTURE Mauritius, AKATA Goavana, Air Madagascar, Ambre Associates, Airtel, Agence Universitaire de la Francophonie (AUF), AXIAN, B2B, Banque Centrale, , BIANCO, BlueLine, CNaPS, Bureau National de Gestion des Risques et des Catastrophes (BNGRC), CEDII-Fianarantsoa, Data Consulting, Central Test, Centre National Antiacridien, CNRE, COLAS, Direction Générale des Douanes, DLC, E-Tech Consulting, , FID, FIHARY Soft, FTM, GNOSYS, GENIUS AT WORK, Hello Tana, IBONIA, INGENOSIA, INSTAT, IOGA, JIRAMA, JOUVE, MADADEV, MAEP, MANAO, MEF, MEN, MESupRES, MFB, , MININTER, Min des Postes/Télécommunications et du Développement Numérique, NEOV MAD, Ny Havana, Madagascar National Parks, OMNITEC, ORANGE, OTME, PRACCESS, QMM Fort-Dauphin, SG Madagasikara SMMC, SMMEC, SNEDADRS Antsirabe, Sénat, Société d’Exploitation du Port de Toamasina (SEPT), SOFTWELL, Strategy Consulting, TELMA, VIVETEC, Société LAZAN’I BETSILEO, WWF, UGD, ARATO, MANAO, MNDPT, NG ACADEMY.NG, Relia …

### Débouchés professionnels et diplômés

Les formations proposées par l’Ecole permettent aux diplômés d’être immédiatement opérationnels sur le marché du travail avec la connaissance d’un métier complet lié à l’informatique aux TIC.

L’Ecole apporte à ses étudiants un savoir-faire et un savoir-être qui les accompagnent tout au long de leur vie professionnelle. Elle a une vocation professionnalisante. Les diplômés en LICENCE et en MASTER issus de l’ENI peuvent faire carrière dans différents secteurs.

L’Ecole bénéficie aujourd’hui de 40 années d’expériences pédagogiques et de reconnaissance auprès des sociétés, des entreprises et des organismes. C’est une Ecole Supérieure de référence en matière informatique.

D’une manière générale, les diplômés de l’ENI n’éprouvent pas de difficultés particulières à être recrutés au terme de leurs études. Cependant, l’ENI recommande à ses diplômés de promouvoir l’entrepreneuriat en TIC et de créer des cybercafés, des SSII ou des bureaux d’études. Le tableau 4 représente les débouchés éventuels des jeunes diplômés.

|  |  |
| --- | --- |
| **LICENCE** | **MASTER** |
| * Analyste - Programmeur * Administrateur de site web/de portail web * Assistant Informatique et internet * Chef de projet web ou multimédia * Développeur Informatique ou multimédia * Intégrateur web ou web designer * Hot liner/Hébergeur Internet * Agent de référencement * Technicien/Supérieur de help desk sur Informatique * Responsable de sécurité web * Administrateur de réseau | * Administrateur de réseau et système * Architecture de système d’information * Développeur d’applications * Ingénieur réseau * Webmaster / Web Designer * Concepteur et réalisateur d’application * Directeur du système d’informations * Chef de projet informatique * Responsable de sécurité informatique * Consultant fonctionnel ou freelance |

Table 4: Débouchés éventuels des jeunes diplômés

### Ressources humaines

Les ressources humaines sont citées ci-dessous selon leurs responsabilités :

* Directeur de l’Ecole : Monsieur MAHATODY Thomas, Docteur HDR
* Responsable de la Mention « Informatique » : Monsieur RABETAFIKA Louis Haja, Maître de Conférences
* Responsable de la Mention « Intelligence Artificielle » : Monsieur DIMBISOA William Germain, Maître de Conférences
* Responsable du Parcours « Génie Logiciel et Base de Données » : Monsieur RALAIVAO Jean Christian, Assistant d’Enseignement Supérieur et de Recherche
* Responsable du Parcours « Administration Systèmes et Réseaux » : Monsieur SIAKA, Assistant d’Enseignement Supérieur et de Recherche
* Responsable du Parcours « Informatique Générale » : Monsieur GILANTE Gesazafy, Assistant d’Enseignement Supérieur et de Recherche
* Responsable du Parcours « Gouvernance et Ingénierie de Données » : Madame RATIANANTITRA Volatiana Marielle, Maître de Conférences
* Responsable du Parcours « Objets Connectés et Cybersécurité » : Monsieur RAZAFIMAHATRATRA Hajarisena, Maître de Conférences

L’ENI compte quinze (15) enseignants permanents dont un (01) Professeur Titulaire, un (01) Professeur, un (01) Docteur HDR, huit (08) Maîtres de Conférences, quatre (04) Assistants d’Enseignement Supérieur et de Recherche, dix (10) enseignants vacataires, quarante un (41) personnel administratif.

## CHAPITRE 2: PRESENTATION DE L’UNIVERSITE ACEEM

### 1.1 Historique

Origine : Ecole Supérieure de Maintenance, de Gestion et de Communication (ESMGC) créée en mars 2008 avec autorisation d’ouverture délivrée par le Ministère de l’Education Nationale par arrêté n°15356-2008/MEN du 21 juillet 2008.

Depuis l’année universitaire 2009-2010, transformation de l’ESMGC en une université dénommée « **Université ACEEM**», un composant (niveau supérieur) du Groupe Educatif ACEEM regroupant trois établissements :

* Faculté de **Droit**, de **Gestion** et de **Communication**
* Ecole Supérieure d’**Informatique** et d’**Electronique**
* Ecole Supérieure des **Sciences de la Santé**

Président Directeur Général du Groupe Educatif ACEEM : Monsieur William RATREMA RAKOTOARINIA.

Recteur de l’Université : Monsieur Benjamin Randrianoelina, Professeur Titulaire

Année universitaire 2012-2013 : ouverture, au niveau de la Mention Gestion, de l’option « Management des Affaires » délivrant le diplôme de *Bachelor of Business Administration* (en jumelage ou Twinning Program avec l’IESEG School de Lille et de Paris).

Année universitaire 2015-2016 : ouverture d’une nouvelle Mention « Economie et Développement ».

Membre du Réseau International Universitaire de Création Numérique (RUN) basé au Canada, qui regroupe plusieurs universités et institutions (canadiennes, françaises, africaines, brésiliennes, chinoises, et de l’Océan Indien).

Leitmotiv de l’Université ACEEM : *Former – Eduquer – Enseigner et Evaluer autrement.*

Valeurs qui guident l’Université dans l’exécution de sa mission : *Dignité – Créativité – Qualité*.

Ethique du Groupe Educatif ACEEM fondée sur le « Hasina maha-olona ».

Disciplines internes consignées dans deux documents :

* Code d’éthique accepté par les enseignants formateurs et adopté depuis octobre 2009.
* Livret de l’étudiant contenant les disciplines et les règlements intérieurs de l’Université ACEEM.

### 1.2 Situation académique

L’UACEEM est intégrée dans le système LMD (Licence - Master - Doctorat) depuis sa création.

1. ***Habilitation par le MESupRES acquise pour les formations suivantes* :**

* Faculté de Droit, de Gestion et de Communication

♦ LICENCE :

* + Arrêté n°1949/2013-MESupRES du 31 janvier 2013 (Mention Droit et Science Politique et Gestion).
  + Arrêté n°9188/2013-MESupRES du 23 avril 2013 (Mention Communication).

♦ MASTER :

* + Arrêté n°21909/2014-MESupRES du 11 juin 2014 (Mention Gestion, option Marketing Management et Commerce International).
  + Arrêté n°33213/2014-MESupRES du 4 novembre 2014 (Mention Gestion, option Finances, Audit et Contrôle de Gestion, et Mention Communication).
  + Arrêté n°37564/2014-MESupRES du 29 décembre 2014 (Mention Droit et Science Politique).
* Ecole Supérieure d’Informatique et d’Electronique

♦ LICENCE :

* + Arrêté n°11118/2012-MESupRES du 5 juin 2012 (Mention Informatique et Electronique).
* Ecole Supérieure des Sciences de la Santé

♦ LICENCE :

* + Arrêté n°31172/2012-MESupRES du 5 décembre 2012 (Mention Sciences de la Santé).
* Faculté des Sciences Economiques et Etudes du Développement

♦ LICENCE :

* Arrêté n°10275/2021-MESupRES du 31.05.21.

1. ***Equivalence des diplômes obtenus auprès du Ministère de la Fonction Publique, du Travail et des Lois Sociales pour les Mentions suivantes :***

* Ecole Supérieure d’Informatique et d’Electronique
  + Arrêté n°13051/2012-CNEAT du 5 juillet 2012.
* Ecole Supérieure des Sciences de la Santé
  + Arrêté n°11994/2013-CNEAT du 29 mai 2013.
* Faculté de Droit, de Gestion et de Communication

♦ LICENCE :

* + Arrêté n°15737/2011-CNEAT du 24 mars 2011 (Mention Gestion).
  + Arrêté n°30244/2011-CNEAT du 17 octobre 2011 (Mention Communication).
  + Arrêté n°13052/2012-CNEAT du 7 juin 2012 (Mention Droit).

♦ MASTER :

* + Arrêté n°30050/2014-CNEAT du 6 octobre 2014 (Mention Gestion, option Marketing Management et Commerce International).
  + Arrêté n°35087/2014-CNEAT du 24 novembre 2014 (Mention Gestion, option Finances Audit et Contrôle de Gestion).
  + Arrêté n°30244/2011-CNEAT du 17 octobre 2011 (Mention Communication).
  + Arrêté n°15286/2016-CNEAT du 22 juillet 2016 (Mention Droit et Science Politique).
* Ecole Supérieure des Relations Internationales et Diplomatie

♦ MASTER

* Arrêté n°31309/2023-MESupRESdu 5 décembre 2023.

L’UACEEM est membre titulaire de l’Agence Universitaire de la Francophonie (AUF). Elle est connue pour la qualité de ses infrastructures sises à Manakambahiny, lesquelles abritent des salles de cours, un centre d’information et de documentation, et quatre salles spécialisées (salle d’informatique, laboratoire de langues, laboratoire informatique/électronique, salle multimédia).

### 1.3 Infrastructures

1. Salles de cours :

* Trois grandes salles pouvant accueillir respectivement 120, 85 et 80 étudiants.
* Seize salles prévues pour 40 étudiants chacune.
* Une salle Informatique.

1. Laboratoires et salle de documentation :

* Un laboratoire de langues.
* Un laboratoire pour Informatique et Electronique.
* Un laboratoire Multimédia.
* Une salle d’Information et de Documentation.

1. Salles de réunions :

* Une salle de réunion pour une vingtaine de personnes (équipée pour visio-conférence).
* Une salle de réunion pouvant accueillir 50 personnes.

1. Autre :

* Un espace pour Loisirs et Sports comprenant :
* Une grande salle multifonctions destinée à abriter les grandes réunions/conférences et les événements festifs (capacité : 150 personnes).
* Une salle de réunion à capacité moyenne (30 personnes).
* Un complexe sportif comportant : un terrain de mini-foot, un terrain de basket-ball, un terrain de volley-ball, un terrain de tennis, une piscine.
* Une salle pour divers jeux : tennis de table, billard, babyfoot
* Un terrain de pétanque.
* Une piscine aux normes.

### 1.4 Organigramme

L'organigramme ci-dessous présente la structure de l’UACEEM, illustrant les rôles clés, les liens hiérarchiques et les interactions entre les différentes équipes. Cet aperçu visuel permet de mieux comprendre les responsabilités et le fonctionnement interne de l’université.

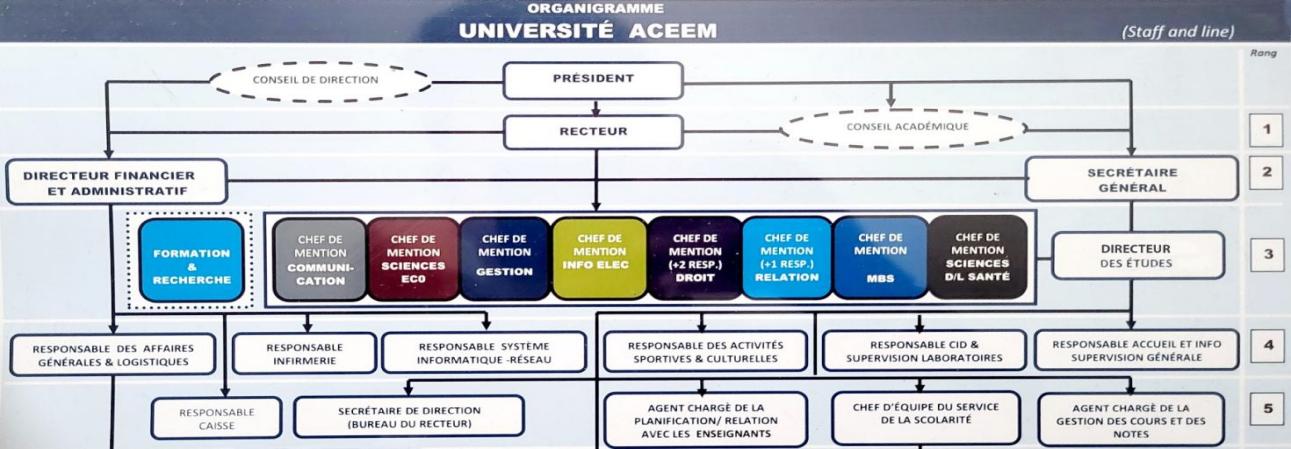


Figure 3: Organigramme de l'UACEEM

### Partenaires

L’UACEEM bénéficie d’un vaste réseau de partenariats avec des entreprises, sociétés et organismes publics et privés. Ils accueillent les étudiants en stage et offrent, dans certains cas, des opportunités de recrutement une fois leur diplôme obtenu.

Ces collaborations jouent un rôle essentiel dans le développement et l'innovation au sein de l'Université ACEEM, en permettant d'atteindre de nouveaux sommets dans la recherche, l'enseignement et la réussite étudiante. C'est grâce à l'engagement et au soutien de ces partenaires que l'université peut offrir des opportunités uniques et faire avancer les projets ambitieux des étudiants.

Ces partenaires partagent la même vision que l’UACEEM et œuvrent chaque jour à son développement :

* TELMA S.A. MADAGASCAR
* SOCIETE GENERALE MADAGASCAR
* Alliance Française d’Antananarivo
* Mercantile Exchange of Madagascar
* Accord d’échange d’étudiants avec IESEG
* BGFIBank
* Ministère de l'Economie et des Finances
* Agence Universitaire de la Francophonie
* MMCI Business School
* FIV.MPA.MA (Fivondronan'ny Mpandraharaha Malagasy)
* S.B.M (State Bank of Mauritius)
* Université d'Antananarivo (Bibliothèque et Archives Universitaires -BAU)
* PÔLE STAGE MADAGASCAR (PSM / SIM) : formation de pré-stage des étudiants et leurs insertions en entreprise
* Tranomboky Iombonana Rajaona Siméon Régis
* PARUL University
* GOANA

### Ressources humaines

L'équipe des ressources humaines est au cœur de l'institut, soutenant le bien-être, la progression professionnelle et la cohésion entre les collaborateurs. Elle s'engage à recruter, encadrer et développer les talents, contribuant ainsi à notre culture d'entreprise et à une performance collective de qualité.

L'équipe dirigeante travaille en étroite collaboration avec les ressources humaines pour définir les orientations stratégiques, inspirer l'innovation et assurer la pérennité de l'institut. Ensemble, elles bâtissent un environnement propice à l'excellence, en alignant les talents individuels avec les objectifs de l'organisation.

L’équipe dirigeante :

|  |  |
| --- | --- |
| NOM ET PRENOMS | Fonction |
| RANDRIANOELINA Benjamin | Recteur |
| RAKOTOARINIA Harivao | Secrétaire Général |
| RAKOTOARINIA Harilala | Directeur Administratif et Financier |
| RAKOTONIAINA Ionintsoa | Directeur des Etudes |
| Les Chefs de Mentions : |  |
| ANDRIANIERENANA Luc | Chef de la Mention Communication |
| RAKOTONDRASOA Seheno | Chef de la Mention Gestion |
| RASTEFANO Elisée | Chef de la Mention Informatique-Electronique |
| ROBIVELO Marie Michel | Chef de la Mention Droit et Science Politique |
| ANDRIANTSOA Jean Rubis | Chef de la Mention Sciences de la Santé |
| RAMANANTSEHENO Domoina | Chef de la Mention Economie  et études du développement |
| RANAIVOSOA Boris | Chef de la Mention Relations Internationales et Diplomatie |
| RAKOTONANAHARY Herizo | Responsable du Développement des Partenariats |

Table 5: Equipe dirigeante de l'UACEEM

* Nombre d’Enseignants : 160
* Effectif du personnel administratif : 36

## CHAPITRE 3. DESCRIPTION DU PROJET

### 3.1 Formulation

Avec l’évolution constante des technologies de l’information, les professionnels de l’UACEEM sont à la recherche de solutions permettant de renforcer leurs interactions, d’optimiser la gestion de leurs informations, et d’encourager la collaboration au sein de leur communauté. La plateforme **KONEKTEA** vise à répondre à ces besoins en fournissant un espace en ligne spécialement conçu pour le personnel(une partie pour les étudiants) de l’UACEEM. Elle se présente comme un réseau social dédié, accessible en tout temps, permettant aux utilisateurs de communiquer, partager des informations( surtout entre les professeurs et etudiants) et interagir de manière dynamique.

Le développement de cette application web répondra à plusieurs besoins, comme la simplification des échanges et la centralisation des ressources documentaires, facilitant ainsi un suivi efficace des activités.

### 3.2 Objectifs et besoins des utilisateurs

Les besoins des utilisateurs représentent les attentes et les demandes spécifiques pour l’usage de la plateforme **KONEKTEA**, tandis que l’objectif final du projet consiste à fournir un espace de communication performant et sécurisé.

1. ****Objectif****

L’objectif de **KONEKTEA** est de créer une plateforme en ligne permettant au personnel de l’UACEEM de rester informé, d’échanger efficacement, et de collaborer facilement autour de projets communs, tout en garantissant la sécurité et l’intégrité des informations partagées. Aussi, la plateforme permet aux professeurs de partager des documents a ses etudiants .

1. ****Besoins des utilisateurs****

La plateforme KONEKTEA doit permettre à chaque utilisateur d’optimiser son expérience tout en assurant la sécurité des données. Il existe trois principaux types d’utilisateurs : les administrateurs, les membres du personnel (simples employes et professeurs), et les etudiants.

Les besoins spécifiques pour ces utilisateurs sont les suivants :

* **S’authentifier et s’inscrire** : Tous les utilisateurs doivent pouvoir s’inscrire, se connecter et gérer leurs informations personnelles.
* **Disposer d’une interface intuitive et ergonomique** : L’interface doit être simple d’utilisation, permettant une navigation fluide et sécurisée.
* **Consulter, réagir et publier du contenu** : Tous les utilisateurs peuvent poster des messages, des fichiers et consulter les publications partagées.

Pour les administrateurs :

* **Gérer les comptes utilisateurs** : Les administrateurs doivent pouvoir ajouter, modifier ou supprimer des comptes et superviser les activités sur la plateforme.
* **Modérer le contenu** : Les administrateurs doivent pouvoir surveiller les publications et s’assurer que le contenu respecte les règles de la communauté.
* **Analyser les interactions** : L’outil de gestion des statistiques permet aux administrateurs de mesurer l’engagement et d’identifier les améliorations nécessaires.
* **Recevoir des feed-back , et des signalements:** Les utilisateurs peuvent envoyer des retours et aussi signaler les contenus inappropriées

Pour les membres du personnel :

* **Communiquer et échanger des documents** : Les membres doivent pouvoir interagir, partager des documents et participer à des discussions
* **Suivre les notifications** : Un système de notifications doit informer les membres des mises à jour et des messages importants.

Pour les étudiants:

* **Recevoir des documents** : Les étudiants reçoivent leurs cours ou autres documents de la part de leurs professeurs
* **Envoyer des feedback** : Ils pourront donner des feedbacks anonymes sur la manière de l’enseignement de leurs professeurs

### 3.3 Moyens nécessaires à la réalisation du projet

##### ****3.3.1 Moyens humains****

Les contributeurs à la réalisation de KONEKTEA sont répertoriés dans le tableau 6.

| **PROFESSION** | **NOMBRE** | **RÔLE DANS LE PROJET** |
| --- | --- | --- |
| Chef de projet | 1 | Responsable de la conception et de la supervision |
| Développeur stagiaire | 1 | Concepteur et Réalisateur de l’application |

Table 6: Moyens humains

##### ****3.3.2 Moyens matériels****

L'ordinateur personnel utilisé pour le développement du projet figure dans le tableau 7.

| **MACHINE** | **RAM** | **PROCESSEUR** | **DISQUE** | **OS** |
| --- | --- | --- | --- | --- |
| Ordinateur portable | 16 Go | Intel(R) Core i5, 8coeurs | 1 To HDD  1To SSD | Garuda Linux 2024 |

Table 7: Moyens materiels

##### ****3.3.3 Moyens logiciels****

Le tableau 8 présente les logiciels utilisés pour le développement de KONEKTEA.

| **DESIGNATION** | **VERSION** | **UTILITÉ DANS LE PROJET** |
| --- | --- | --- |
| Figma |  | Maquette et prototype de l’application |
| Webstorm | 2024.2.2 | Développement de l’application |
| Visual Paradigm | 17.2 | Modélisation |
| WPS Office | 2023 | Rédaction du rapport |
| Navigateur Web |  | Interprétation et tests des codes |

Table 8: Moyens logiciels

### 3.4 Résultats attendus

L'achèvement de la plateforme KONEKTEA permettra de répondre aux divers besoins des utilisateurs mentionnés. Elle doit être fiable, accessible, performante, et capable d’évoluer avec les besoins futurs de la communauté.

Les résultats attendus pour KONEKTEA incluent :

* **Fiabilité et accessibilité** : La plateforme doit être disponible en tout temps et répondre aux besoins des utilisateurs avec fiabilité.
* **Engagement communautaire** : KONEKTEA doit encourager les interactions entre les membres du personnel et favoriser la collaboration.
* **Flexibilité et évolutivité** : La solution doit pouvoir s’adapter aux besoins croissants de l’UACEEM et permettre des mises à jour facilitées.

### 3.5 Chronogramme d’activité

Pour mener a bien ce projet, un planning a été conçu.

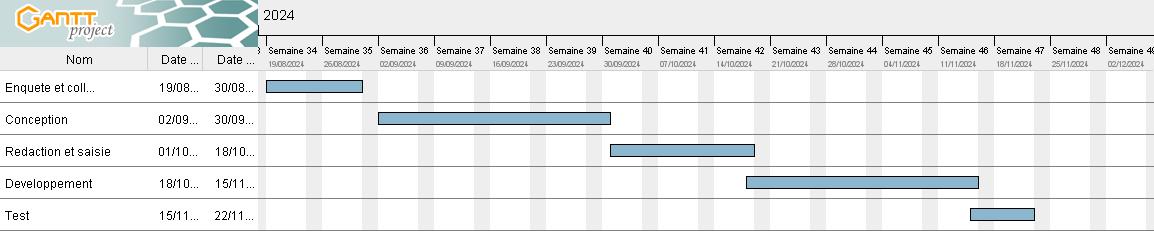


Figure 4: Chronogramme d'activite

# PARTIE II: ANALYSE ET CONCEPTION

## CHAPITRE 4: ANALYSE PREALABLE

C’est une étape importante dans le développement de logiciels ou de systèmes informatiques. Il s’agit de comprendre les besoins des utilisateurs finaux, les exigences fonctionnelles et les contraintes techniques avant de commencer à concevoir et à développer le système.

#### 4.1 Analyse de l’existant

##### 4.1.1 Organisation actuelle

Une analyse neutre et objective de l’actuel service et des applications utilisées par l’UACEEM a été conçue.

En ce moment, le service n’a aucune plateforme ou une sorte d’application pour la digitalisation des activites au sein de université.

##### 4.1.2 Existants materiels et logiciels

**Moyens matériels**

Chez UACEEM, on dispose des desktops et des laptops de configuration variés pour effectuer le travail.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Matériels** | **Processeur** | **OS** | **RAM** | **Disque Dur** |
| **Portable** | Intel core i5 | Garuda Linux | 16Go | 1To SSD - 1To HDD |
| **Serveur** | 16 coeurs | Ubuntu 20.04 | 64Go | 512Go |

Table 9: Moyens materiels en possession

**Moyens logiciels**

|  |  |  |
| --- | --- | --- |
| **DESIGNATION** | **FONCTION** | **VERSION** |
| Figma | Maquette et prototypage | 2024 |
| Webstorm | Editeur de code (IDE) | 2024.2.2 |
| Google Chrome | Navigateur | 119.0.6045 |

Table 10: Moyens logiciels en possession

#### 4.2 Critique de l’existant

##### ****4.2.1 Points faibles****

· **Absence de plateforme numérique:** Le principal point faible est l'absence d'une plateforme centralisée pour la gestion des informations et la communication au sein de l'UACEEM. Cela entraîne une dispersion des données, des difficultés de coordination et une perte de temps.

· **Manque de collaboration:** Les outils de collaboration actuels sont limités et peu efficaces, ce qui entrave le travail en équipe et le partage de connaissances.

· **Accès limité aux ressources:** Les étudiants et le personnel n'ont pas un accès facile et centralisé aux ressources pédagogiques, aux annonces et aux événements.

· **Sécurité des données:** L'absence d'une plateforme sécurisée expose les données sensibles à des risques de perte ou de vol.

· **Difficulté de suivi des activités:** Il est complexe de suivre l'avancement des projets et les performances individuelles en l'absence d'outils de suivi adaptés.

##### ****4.2.2 Points forts****

· **Présence de matériel informatique:** L'UACEEM dispose d'un parc informatique suffisant pour supporter le développement et l'utilisation de la nouvelle plateforme.

· **Compétences techniques:** L'équipe de développement possède les compétences nécessaires pour mener à bien le projet.

· **Besoin exprimé:** Il existe un besoin clairement identifié de la part des utilisateurs d'une plateforme numérique pour améliorer leur quotidien.

#### ****4.3 Conception avant projet****

##### 4.3.1 Proposition des solutions

**Solution 1:** Adoption d'une solution SaaS (Software as a Service)

**Solution 2:** Développement d'une plateforme sur mesure

|  |  |  |
| --- | --- | --- |
| **SOLUTIONS** | **AVANTAGES** | **INCONVENIENTS** |
| **Solution 1** | **Mise en œuvre rapide**, **coût prévisible** et **évolutivité** facilitées.  **Fonctionnalités riches** et régulièrement mises à jour.  **Pas de gestion d'infrastructure** nécessaire. | **Dépendance à un fournisseur tiers**.  **Moins de contrôle** sur les fonctionnalités et les données.  **Coût récurrent** pouvant être élevé à long terme. |
| **Solution 2** | **Personnalisation complète** adaptée aux besoins spécifiques de l'UACEEM. **Intégration** facile avec les systèmes existants.  **Contrôle total** sur les fonctionnalités et l'évolution de la plateforme. | **Coût élevé** de développement et de maintenance.  **Délai de mise en œuvre** plus long.  **Nécessité de ressources techniques internes importantes.** |

Table 11: Comparaison des solutiions

Le choix de développer une plateforme sur mesure pour l'UACEEM a été retenue puisqu’elle correspond aux besoins spécifiques de la communauté et aussi, elle apporte des innovations au sein de la communauté de l’universite.

##### 4.3.2 Methodes de conception et outils utilises

Une méthode de conception est un processus organisé et systématique utilisé pour planifier, développer et réaliser un projet, un produit ou un système en suivant des étapes structurées et des principes spécifiques.

1. Méthode de conception

|  |  |  |
| --- | --- | --- |
| **METHODES** | **AVANTAGES** | **INCOVENIENTS** |
| **MERISE** | **Structure rigoureuse:** Propose une approche structurée et détaillée pour la conception de systèmes d'information, facilitant la compréhension et la maintenance.  **Indépendance des données et des traitements:** Permet une modélisation claire et séparée des aspects fonctionnels et des données. | **Rigide:** Peut être perçue comme trop contraignante pour certains projets, notamment ceux nécessitant une grande flexibilité.  **Moins adaptée aux projets agiles:** La méthode est plus adaptée à des projets de développement traditionnels qu'à des projets agiles où les besoins évoluent rapidement. |
| **2TUP** | **Orientée objet:** Utilise le langage UML pour une modélisation objet, favorisant la réutilisabilité et la maintenabilité du code.  **Itérative et incrémentale:** Permet de développer le système par itérations successives, ce qui facilite la gestion des changements et réduit les risques. | **Maitrise:** Nécessite une bonne maîtrise d'UML, ce qui peut représenter un investissement en temps pour les équipes.  **Complexité:** Peut être perçue comme complexe pour des projets de petite envergure. |

Table 12: Comparaison entre MERISE et 2TUP

Compte tenu des objectifs modulaires et évolutifs de notre projet, nous avons opté pour le processus unifié 2TUP (Two Track Unified Process). Ce choix s'est imposé pour plusieurs raisons :

**Qu'est-ce qu'un processus unifié ?**

Un processus unifié est une méthodologie de développement logiciel qui s'appuie sur les meilleures pratiques du secteur. Il offre un cadre structuré et flexible pour concevoir, développer et déployer des applications. Ce cadre est itératif, ce qui signifie que le développement se fait par cycles successifs, permettant ainsi d'ajuster le projet en fonction des retours et des évolutions.

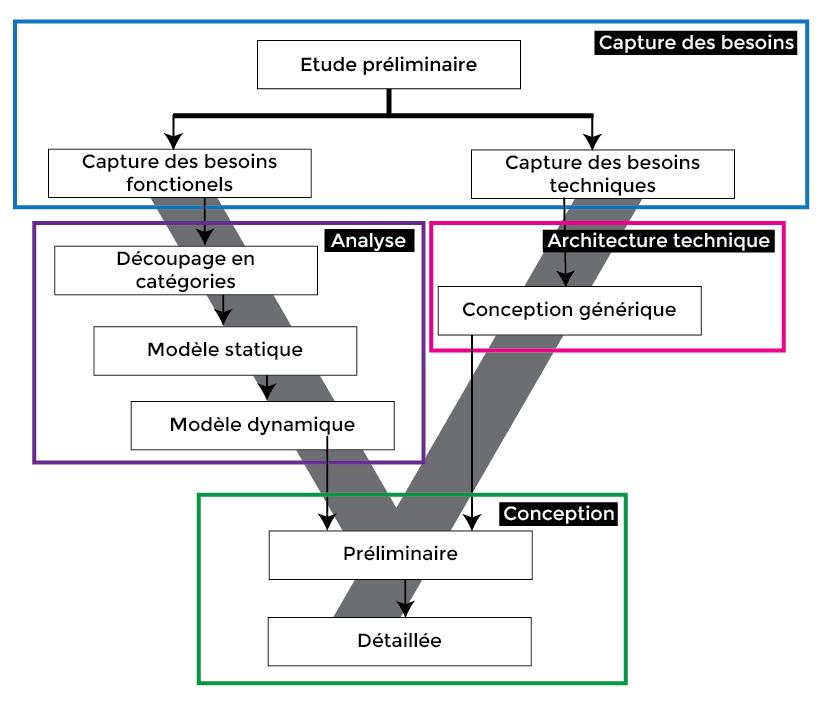
Le 2TUP, une évolution du processus unifié, se distingue par son approche en deux volets : un volet fonctionnel et un volet technique. Cette double perspective permet de mieux gérer les changements inévitables dans les projets informatiques.

Figure 5: Structure en Y du 2TUP

**Les phases du processus 2TUP**

Le processus 2TUP se déroule en quatre phases principales :

* **Inception :** Étude de faisabilité, définition des besoins et des objectifs du projet.
* **Élaboration :** Élaboration de l'architecture technique et développement des fonctionnalités prioritaires.
* **Construction :** Développement complet des fonctionnalités et intégration des différents composants.
* **Transition :** Déploiement de l'application, formation des utilisateurs et support post-lancement.

**Les activités du processus 2TUP**

Le 2TUP s'articule autour de six disciplines clés :

* **Modélisation métier:** Analyse des activités et des processus de l'entreprise.
* **Capture des besoins:** Définition des besoins des utilisateurs et des fonctionnalités à implémenter.
* **Analyse et conception:** Conception de l'architecture du système et des modèles de données.
* **Implémentation:** Développement du code.
* **Test:** Vérification de la qualité et du bon fonctionnement du système.
* **Déploiement:** Mise en production du système.

**Le processus en Y : une spécificité du 2TUP**

Le 2TUP se distingue par son processus en Y, qui illustre la dualité entre les aspects fonctionnels et techniques du développement. Les évolutions du système sont gérées de manière parallèle sur ces deux axes, avant d'être fusionnées en une version finale.

**Le rôle de UML dans le 2TUP**

UML (Unified Modeling Language) est le langage de modélisation utilisé dans le 2TUP. Il permet de représenter de manière graphique les différents aspects du système, facilitant ainsi la communication entre les membres de l'équipe et la compréhension du système.

UML offre une grande variété de diagrammes pour représenter :

* **Les cas d'utilisation:** Les interactions entre les utilisateurs et le système.
* **La structure statique du système:** Les classes, les objets et leurs relations.
* **Le comportement dynamique du système:** Les séquences d'actions, les états, les activités.
* **L'architecture du système:** Les composants, les déploiements, etc.

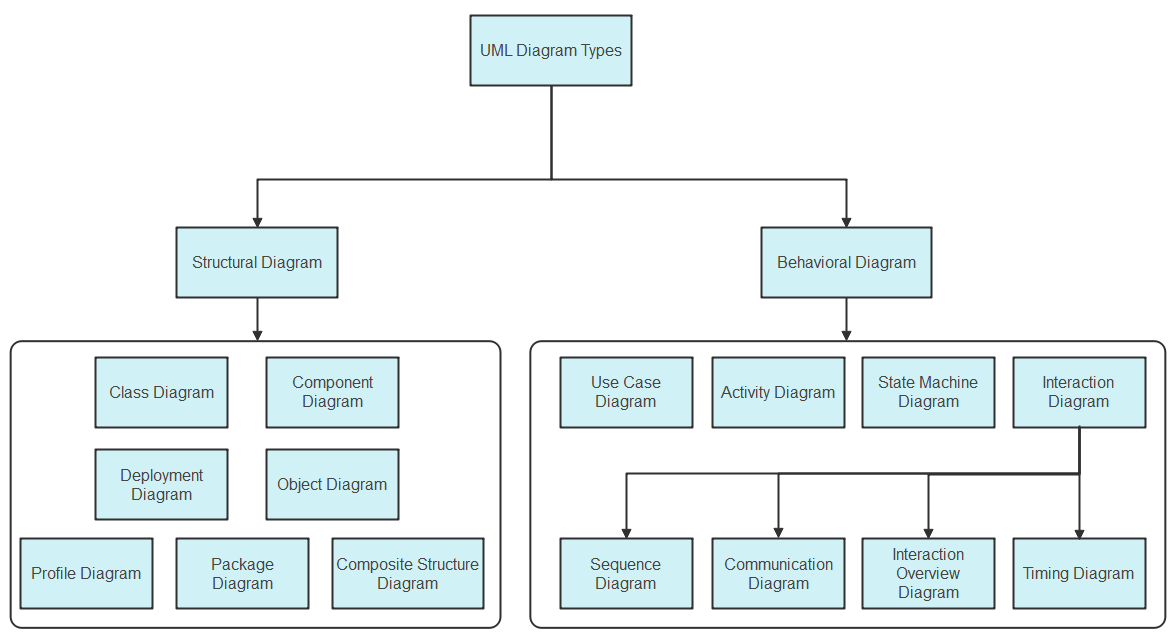


Figure 6: Diagramme UML

**En résumé,** le processus 2TUP, associé à UML, nous offre un cadre méthodologique rigoureux et flexible pour mener à bien notre projet. Cette approche nous permet de garantir la qualité, la maintenabilité et l'évolutivité de notre application.

1. **Langage de programmation, Librairie et Framework**

Langage de progrmmation

* Frontend

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Caractéristique** | **Java** | **JavaScript** |
| **Nature** | | Langage compilé, orienté objet, exécuté dans une machine virtuelle. | Langage interprété, principalement orienté objet, exécuté directement dans le navigateur. |
|  | **Utilisation en frontend** | Souvent utilisé avec des frameworks comme JavaFX ou GWT pour créer des interfaces riches. Nécessite l'installation d'un plugin (Java applet). | Principalement utilisé pour créer des interfaces interactives et dynamiques directement dans le navigateur. |
|  | **Avantages pour les interfaces utilisateur** | Interfaces riches, possibilité d'utiliser des technologies côté client. | Intégration directe avec le DOM, compatibilité multi-navigateurs, large écosystème de frameworks et bibliothèques. |
|  | **Inconvénients pour les interfaces utilisateur** | Complexité de mise en œuvre, problèmes de compatibilité avec les plugins, taille des applications. | Performances potentiellement moins bonnes, complexité pour les grandes applications. |

Table 13: Comparaison entre Java et Javascript

Pourquoi choisir JavaScript pour les interfaces utilisateur ?

JavaScript s'impose comme le langage de prédilection pour le développement d'interfaces utilisateur modernes pour plusieurs raisons :

* **Intégration directe avec le DOM:** JavaScript permet de manipuler directement les éléments HTML d'une page, offrant ainsi un contrôle précis sur l'apparence et le comportement de l'interface.
* **Compatibilité universelle:** Tous les navigateurs modernes supportent nativement JavaScript, garantissant une large portée des applications développées.
* **Écosystème riche:** JavaScript bénéficie d'un écosystème très actif avec de nombreux frameworks (React, Angular, Vue) et bibliothèques (jQuery) qui facilitent le développement d'applications complexes.
* **Interactivité:** JavaScript permet de créer des interfaces dynamiques et réactives, répondant en temps réel aux actions de l'utilisateur.

**En résumé,** si Java offre des possibilités intéressantes pour créer des interfaces riches, JavaScript est devenu le standard de facto pour le développement d'interfaces utilisateur web en raison de sa simplicité d'utilisation, de sa flexibilité et de son large adoption.

* Backend

|  |  |  |
| --- | --- | --- |
| **Caractéristique** | **PHP (Laravel)** | **Node.js** |
| **Langage principal** | PHP | JavaScript |
| **Écosystème** | Composer, Packagist | npm |
| **Framework populaire** | Laravel | Express.js, NestJS, Koa.js |
| **Performance** | Bonne pour la plupart des applications, peut être optimisée | Excellente pour les applications en temps réel et à forte charge |
| **Scalabilité** | Bonne, mais peut nécessiter des optimisations pour de très fortes charges | Très bonne, grâce au modèle événementiel |
| **Communauté** | Très grande et mature | Très grande et dynamique |
| **Courbe d'apprentissage** | Relativement facile pour les développeurs PHP | Peut être plus raide pour les débutants en raison du modèle événementiel |
| **Utilisation typique** | Applications web traditionnelles, CMS, e-commerce | Applications en temps réel, API RESTful, applications monopage (SPA), microservices |
| **Forces** | Structure conventionnelle, écosystème mature, nombreux outils | Performances, flexibilité, fullstack JavaScript |
| **Faiblesses** | Peut être moins performant pour les applications à très forte charge | Peut être plus complexe à mettre en œuvre pour les petits projets |

Table 14: Comparaison entre PHPl et NodeJS

Le choix pour le backend a ete fait et on a choisi le Node JS en raison de ces quelques points:

* **Applications en temps réel:** Node.js est idéal pour les applications nécessitant une interaction en temps réel avec les utilisateurs (chats, jeux, etc.).
* **API RESTful:** Node.js est très performant pour créer des API RESTful.
* **Applications monopage (SPA):** Node.js est souvent utilisé pour créer le backend des applications SPA.
* **Microservices:** Node.js est bien adapté à l'architecture des microservices.

Framework

* **Frontend:**

JavaScript propose des frameworks comme ReactJS, AngularJS, VueJS,.....

|  |  |  |
| --- | --- | --- |
| **Caractéristique** | **React** | **Angular** |
| **Nature** | Bibliothèque JavaScript pour construire des interfaces utilisateur | Framework JavaScript complet pour applications web |
| **Architecture** | Composants, JSX, Virtual DOM | Modèle-Vue-Controller (MVC), TypeScript |
| **Flexibilité** | Très flexible, permet une grande personnalisation | Moins flexible, mais offre une structure bien définie |
| **Courbe d'apprentissage** | Relativement facile | Peut être plus abrupte, surtout pour les débutants |
| **Communauté** | Très grande et active | Grande et active, mais légèrement moins que React |
| **Performance** | Excellente grâce au Virtual DOM | Bonne, mais peut être moins performant que React dans certains cas |
| **Idéal pour** | Applications à grande échelle, interfaces utilisateur complexes, applications mobiles | Grandes applications d'entreprise, applications web complexes |

Table 15: Comparaison entre React JS et Angular JS

Le choix pour le ReactJS a ete facile puisqu’en plus d’etre flexible et simple, il est performant et correspond aux besoins des utilisateurs

* **Backend**

NodeJS propose plusieurs Frameworks comme Express, Nest, Koa,Hapi,...

|  |  |  |
| --- | --- | --- |
| **Caractéristique** | **Express.js** | **NestJS** |
| **Nature** | Microframework minimaliste et flexible | Framework modulaire et opinionné |
| **Langage** | JavaScript (TypeScript en option) | TypeScript (principalement) |
| **Architecture** | Middleware | Modulaire, inspiré d'Angular |
| **Performance** | Très performant | Légèrement moins performant qu'Express.js (mais reste très bon) |
| **Scalabilité** | Très bonne, grâce à son architecture modulaire | Excellente, conçue pour les applications à grande échelle |
| **Idéal pour** | Projets de petite à moyenne taille, API REST, applications simples | Projets complexes, applications d'entreprise, équipes importantes |
| **Forces** | Flexibilité, simplicité, performance | Structure, typage fort, testabilité, évolutivité |
| **Faiblesses** | Moins de structure pour les grands projets, moins de fonctionnalités intégrées | Courbe d'apprentissage plus raide |

Table 16: Comparaison entre ExpressJS et NestJS

**Express.js** est un excellent choix pour notre projet puisque nous recherchons :

* **La simplicité:** un framework minimaliste et facile à prendre en main.
* **La flexibilité:** besoin d'une grande liberté dans la conception de notre application.
* **Les performances:** besoin d'une application rapide et réactive.

**Systeme de gestion de base de donnees**

|  |  |  |
| --- | --- | --- |
| **Caractéristique** | **MySQL** | **MongoDB** |
| **Modèle de données** | Relationnel (tables, colonnes, lignes) | NoSQL (documents JSON) |
| **Langage de requête** | SQL (Structured Query Language) | MongoDB Query Language (MQL) |
| **Scalabilité** | Bonne scalabilité verticale, peut nécessiter un sharding pour une scalabilité horizontale importante | Excellente scalabilité horizontale grâce à sa nature distribuée |
| **Transactions** | Transactions ACID (Atomicité, Cohérence, Isolation, Durabilité) | Soutien limité aux transactions |
| **Relations entre les données** | Relations bien définies entre les tables via des clés étrangères | Relations plus flexibles, basées sur des références ou des imbrications |
| **Utilisation typique** | Applications d'entreprise, e-commerce, systèmes de gestion de contenu (CMS) | Applications web modernes, stockage de données non structurées, Big Data |

Table 17: Comparaison entre MySQL et MongoDB

**MySQL** est un excellent choix pour notre projet si nous recherchons :

* **Un modèle de données relationnel solide:** besoin d'une structure de données claire et bien définie.
* **Des transactions ACID:** L'intégrité des données est essentielle pour notre application.
* **Une communauté active et un support important:** besoin d'une base de données mature et fiable.

**Outils de prototypage**

|  |  |  |
| --- | --- | --- |
| **Caractéristique** | **Figma** | **Adobe XD** |
| **Nature** | Outil de conception UI/UX basé sur le web | Outil de conception UI/UX principalement desktop (avec une version web) |
| **Fonctionnalités principales** | Création de maquettes, prototypes interactifs, collaboration en temps réel, conception de systèmes de design, gestion de versions | Création de maquettes, prototypes interactifs, intégration avec d'autres outils Adobe |
| **Collaboration** | Collaboration en temps réel très avancée, commentaires, historique des versions | Collaboration en temps réel, mais moins fluide que Figma |
| **Plateforme** | Basé sur le web (fonctionne sur tous les systèmes d'exploitation) | Principalement desktop (Windows, macOS), version web disponible |
| **Système de design** | Gestion avancée des bibliothèques de composants, création de systèmes de design complets | Gestion des composants, mais moins complète que Figma |
| **Prototypage** | Prototypes interactifs riches, micro-interactions, transitions | Prototypes interactifs, mais moins de flexibilité que Figma |
| **Prix** | Gratuit pour les équipes de moins de 2 personnes, tarifs abordables pour les équipes plus grandes | Abonnements payants, inclus dans la suite Adobe Creative Cloud |
| **Communauté** | Très grande et active, nombreux plugins et ressources | Grande communauté, mais moins active que Figma |

Table 18: Comparaison entre Figma et Adobe XD

Figma se distingue par sa collaboration en temps réel fluide, son système de design puissant et sa flexibilité. C'est un outil idéal pour les designers, les développeurs et les équipes qui souhaitent travailler de manière efficace et collaborative sur leurs projets de design.

**En choisissant Figma, nous optons pour un outil moderne, performant et adapté aux besoins des équipes de conception actuelles.**

**Outils de developpement**

|  |  |  |
| --- | --- | --- |
| **Caractéristique** | **Visual Studio Code** | **WebStorm** |
| **Nature** | Éditeur de code source extensible | Environnement de développement intégré (IDE) complet |
| **Fonctionnalités principales** | Édition de code, débogage, intégration Git, personnalisation via extensions | Édition de code avancée, débogage, refactoring, tests unitaires, intégration avec d'autres outils JetBrains, prise en charge native de nombreux frameworks |
| **Langages supportés** | Large éventail de langages grâce aux extensions (JavaScript, TypeScript, Python, etc.) | Spécialisé dans le développement web (JavaScript, TypeScript, HTML, CSS, etc.) avec une prise en charge approfondie des frameworks populaires (React, Angular, Vue.js) |
| **Communauté** | Très grande et active, communauté open-source | Grande communauté, axée sur le développement web et les outils JetBrains |
| **Prix** | Gratuit | Payant (offre une période d'essai gratuite sauf si on est inscrit au Github Student qui nous offre une licence pour les produits Jetbrains) |

Table 19: Comparaison entre VS Code et Webstorm

WebStorm est un choix judicieux pour les développeurs web qui recherchent un environnement de développement complet, productif et spécialisé. Bien que VS Code soit un excellent éditeur de code, il ne peut pas égaler les fonctionnalités avancées et la productivité de WebStorm, en particulier pour les projets web complexes. Et voila la raison pour laquelle on a choisi Webstorm pour le developpement de l’application

## CHAPITRE 5: ANALYSE CONCEPTUELLE

#### 5.1 Presentation de la methode 2TUP

La méthode **2TUP** (Two-Tiered Unified Process) est un modèle de développement logiciel qui repose sur des itérations courtes pour la conception et la livraison de produits. Ce processus combine une planification rigoureuse avec une exécution itérative, en distinguant clairement les phases de **conception** et de **réalisation**, afin de garantir une gestion optimisée du projet.

Nous avons appliqué cette approche, qui est à la fois **itérative** et **incrémentale**, en se basant sur une architecture solide et orientée par les cas d’utilisation. Le processus est piloté par l'utilisation de **UML** (Unified Modeling Language) pour la modélisation. Pour élaborer les différents diagrammes UML nécessaires, un **dictionnaire des données** et un ensemble de **règles de gestion** sont indispensables.

#### 5.2 Dictionnaire des donnees

#### 5.3 Regle de gestion

Une règle de gestion représente un principe appliqué par le système. Cela peut correspondre à une exigence légale, une directive émanant d'un responsable, ou encore une disposition interne spécifique à l'application. Suite à l'analyse de la situation actuelle, les règles de gestion suivantes sont jugées essentielles :

**RG1** : Un compte est attribué à un et un seul utilisateur et doit être créé par l’administrateur du plateforme

**RG2** : Chaque utilisateur peut envoyer des messages et des fichiers à d’autres utilisateurs. Surtout les professeurs peut envoyer des documents a ses étudiants.

**RG3** : Chaque utilisateur peut consulter l’historique de ses activités (messages envoyés, fichiers partagés).

**RG4** : L’administrateur peut surveiller toutes les interactions (publications, messages, fichiers) des utilisateurs sur la plateforme pour assurer la conformité des échanges.

**RG5** : Seul l’administrateur peut désactiver, supprimer ou restaurer un compte utilisateur.

**RG6** : Un utilisateur peut mettre à jour son profil (nom, fonction, département), sauf son adresse e-mail et son mot de passe qui est définie par l’institution et ne peut être modifiée.

**RG7** : Seul l’administrateur peut créer des comptes d'utilisateurs ayant des droits spéciaux (modérateurs ou administrateurs).

**RG8** : Les publications peuvent être supprimées par leur auteur, par un modérateur ou un administrateur, mais l’historique de modification reste consultable.

**RG9** : Les utilisateurs peuvent signaler des comportements inappropriés ou des contenus offensants à un modérateur ou à l’administrateur via une fonction de signalement.

**RG10**: Un système de notifications en temps réel informe les utilisateurs des nouveaux messages, ou des réponses à leurs publications.

**RG11** : L’administrateur peut suspendre temporairement un utilisateur pour non-respect des règles d’utilisation de la plateforme..

#### 5.4 Representation et specification des besoins

Dans le cadre de la mise en place d’un système informatique performant, l'accent est mis sur l’identification des besoins des utilisateurs. Ces besoins sont analysés à travers les spécifications fonctionnelles et non fonctionnelles afin de concevoir une application de qualité répondant aux attentes des utilisateurs.

##### 5.4.1 Diagramme des cas d’utilisation

Le **diagramme de cas d’utilisation** illustre les fonctionnalités essentielles dont les utilisateurs ont besoin, en présentant les interactions sous forme d'actions et de réactions.

Ce diagramme joue un rôle crucial dans l'analyse du système, car il permet de :

* Délimiter le périmètre du système et ses interactions avec son environnement,
* Identifier et comprendre les attentes des utilisateurs,
* Modéliser les aspects dynamiques du système,
* Mettre en évidence les acteurs et les fonctionnalités proposées par le système.

**Diagramme des cas d’utilsiation du systeme**

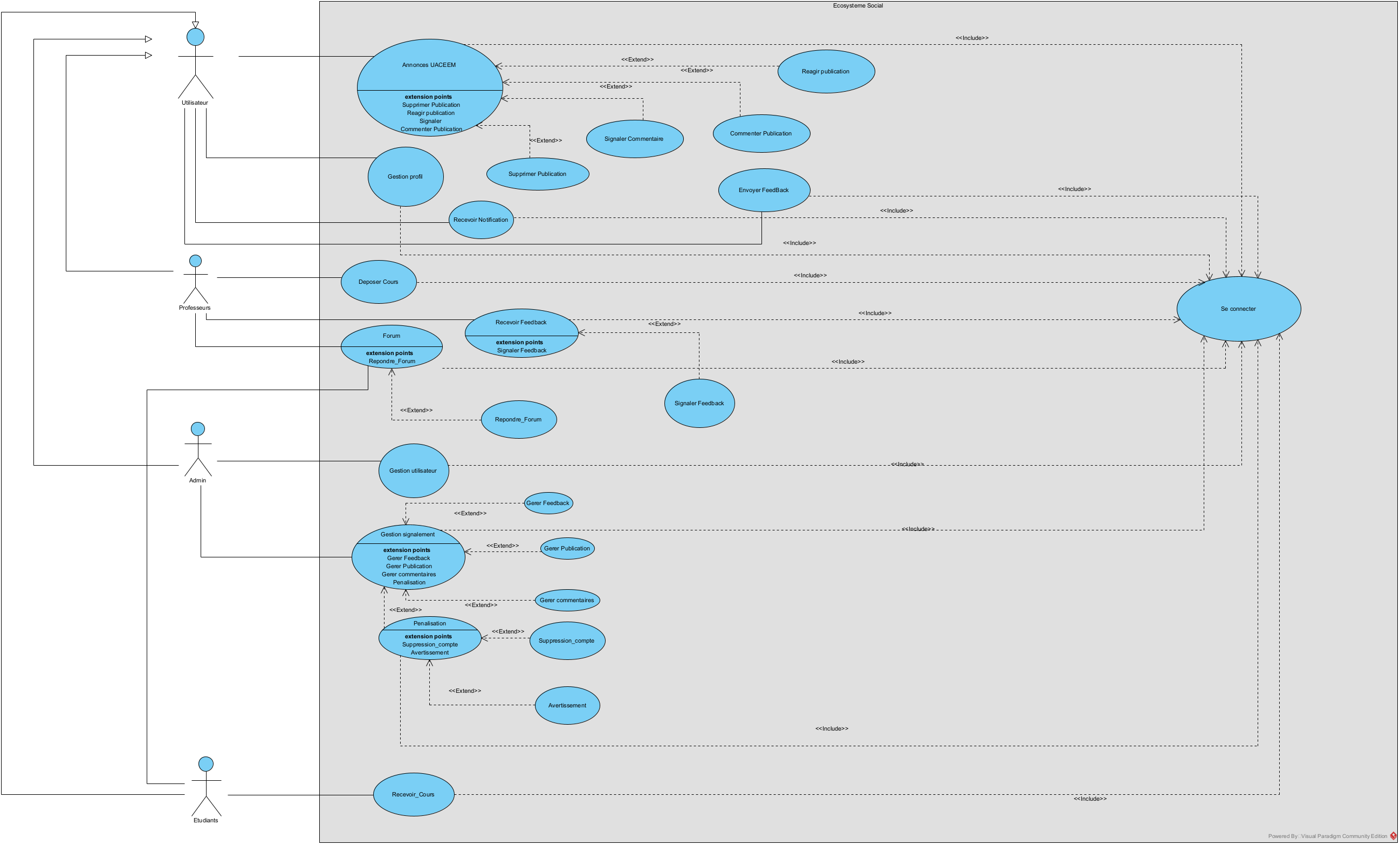


Figure 7: Diagramme des cas d'utilsiation

##### 5.4.2 Description textuelle pour chaque cas d’utilisation

Un **cas d'utilisation** est constitué d'un ensemble d'interactions, c’est-à-dire des échanges entre une action initiée par un acteur et la réponse du système. Décrire un cas d’utilisation revient à spécifier :

* Chaque interaction de manière séquentielle, en précisant les règles métier appliquées par le système (aussi appelées CU) ;
* L'événement déclencheur du cas d’utilisation (initié par les acteurs) ;
* Les conditions préalables que le système doit satisfaire pour permettre l'exécution du cas d'utilisation (préconditions) ;
* L'état du système après l'exécution du cas d’utilisation (postconditions).

| **Rubrique** | **Description** |
| --- | --- |
| Cas d'utilisation | Se connecter |
| Acteur | Utilisateurs (Admin, Professeurs, Étudiants) |
| Précondition | - L'utilisateur dispose d'un compte valide dans le système  - L'utilisateur a accès à internet |
| Scénario nominal | 1. L'utilisateur demande l'accès à la page de connexion 2. Le système affiche le formulaire de connexion 3. L'utilisateur saisit son identifiant et son mot de passe 4. Le système vérifie les informations d'identification 5. Le système authentifie l'utilisateur   6. Le système redirige l'utilisateur vers son tableau de bord |
| Scénario alternatif | 1. Identifiants incorrects:   - Le système affiche un message d'erreur  - Retour à l'étape 2 du scénario nominal   1. Compte bloqué:   - Le système affiche un message approprié  - Proposition de contacter l'administrateur |
| Scénario d'exception | - Panne du serveur d'authentification  - Problème de connexion réseau  - Base de données inaccessible |
| Postcondition | - L'utilisateur est connecté à son espace personnel  - Les droits d'accès sont appliqués selon son profil |

Table 20: Description textuelle de "Se connecter"

| **Rubrique** | **Description** |
| --- | --- |
| Cas d'utilisation | Déposer Cours |
| Acteur | Professeur |
| Précondition | - Le professeur est authentifié  - Le professeur a les droits de dépôt de cours |
| Scénario nominal | 1. Le professeur accède à l'interface de dépôt de cours 2. Le système affiche le formulaire de dépôt 3. Le professeur renseigne les métadonnées du cours (titre, description, niveau) 4. Le professeur sélectionne et téléverse les fichiers du cours 5. Le système vérifie et valide les fichiers   6. Le système enregistre le cours et notifie le succès |
| Scénario alternatif | 1. Format de fichier non supporté:   - Message d'erreur spécifiant les formats acceptés   1. Taille de fichier excessive:   - Suggestion de compression ou de fractionnement |
| Scénario d'exception | - Échec du téléversement  - Espace de stockage insuffisant  - Erreur de traitement des fichiers |
| Postcondition | - Le cours est enregistré dans le système  - Les étudiants concernés sont notifiés  - Le cours apparaît dans la liste des cours disponibles |

Table 21: Description textuelle "Déposer Cours"

| **Rubrique** | **Description** |
| --- | --- |
| Cas d'utilisation | Gestion utilisateur |
| Acteur | Admin |
| Précondition | - L'administrateur est authentifié  - L'administrateur possède les droits de gestion des utilisateurs |
| Scénario nominal | 1. L'admin accède au module de gestion des utilisateurs 2. Le système affiche la liste des utilisateurs avec options de filtrage 3. L'admin peut:   - Créer un nouvel utilisateur  - Modifier un utilisateur existant  - Désactiver/réactiver un compte  - Gérer les roles  4. Le système applique les modifications |
| Scénario alternatif | 1. Création d'utilisateur avec email existant:   - Message d'erreur de doublon   1. Modification de droits non autorisée:   - Message d'erreur de permission |
| Scénario d'exception | - Erreur de synchronisation avec la base de données  - Conflit de modification simultanée |
| Postcondition | - Les modifications sont enregistrées  - Les utilisateurs concernés sont notifiés des changements |

Table 22: Description textuelle "Gestion utilisateur"

| **Rubrique** | **Description** |
| --- | --- |
| Cas d'utilisation | Accès/voir KONEKTEA |
| Acteur | Tous les utilisateurs |
| Précondition | - Accès à internet |
| Scénario nominal | 1. L'utilisateur accède à la page principale de KONEKTEA 2. Le système affiche:   - Les informations générales  - Les actualités  3. L'utilisateur peut naviguer entre les différentes sections |
| Scénario alternatif | 1. Accès à du contenu restreint:   - Redirection vers la page de connexion   1. Contenu temporairement indisponible:   - Message d'information |
| Scénario d'exception | - Panne du serveur  - Maintenance du site |
| Postcondition | - L'utilisateur a accès aux informations publiques |

Table 23: Description textuelle "Acces/voir KONEKTEA"

| **Rubrique** | **Description** |
| --- | --- |
| Cas d'utilisation | Gestion profil |
| Acteur | Tous les utilisateurs authentifiés |
| Précondition | - L'utilisateur est connecté au système  - Le profil existe dans la base de données |
| Scénario nominal | 1. L'utilisateur accède à son espace profil 2. Le système affiche les informations actuelles:   - Informations personnelles  - Photo de profil  - Préférences de notification  - Paramètres de confidentialité   1. L'utilisateur modifie les informations souhaitées sauf l’adresse email et le mot de passe   4. Le système valide et enregistre les modifications |
| Scénario alternatif | 1. Format de photo non valide:   - Message d'erreur avec formats acceptés |
| Scénario d'exception | - Erreur lors de l'enregistrement des modifications  - Problème de chargement des données |
| Postcondition | - Les modifications sont enregistrées  - Le profil est mis à jour |

Table 24: Description textuelle "Gestion de profil"

| **Rubrique** | **Description** |
| --- | --- |
| Cas d'utilisation | Recevoir Notification |
| Acteur | Tous les utilisateurs authentifiés |
| Précondition | - L'utilisateur est inscrit au système  - Les notifications sont activées |
| Scénario nominal | 1. Un événement déclencheur se produit:   - Nouveau cours disponible  - Message reçu   1. Le système génère la notification 2. L'utilisateur reçoit la notification 3. L'utilisateur peut:   - Consulter la notification  - Marquer comme lue  - Supprimer |
| Scénario alternatif | 1. Notifications désactivées   2. Notification non délivrée |
| Scénario d'exception | - Erreur de génération de notification  - Problème de livraison |
| Postcondition | - La notification est marquée comme délivrée  - L'historique des notifications est mis à jour |

Table 25: Description textuelle "Recevoir Notification"

| **Rubrique** | **Description** |
| --- | --- |
| Cas d'utilisation | Envoyer Feedback / Recevoir Feedback |
| Acteur | Étudiant, Professeur |
| Précondition | - L'utilisateur est authentifié  - L'élément concerné existe (cours, fonctionnalité, publications, commentaires) |
| Scénario nominal | 1. L'utilisateur accède au formulaire de feedback   2. Le système affiche le formulaire avec:  - Sélection du type de feedback  - Champ de description   1. - Option de pièce jointe   3. L'utilisateur soumet son feedback  4. Le système confirme la réception |
| Scénario alternatif | 1. Formulaire incomplet:   - Message d'erreur  - Champs requis mis en évidence  2. Pièce jointe trop volumineuse |
| Scénario d'exception | - Erreur lors de la soumission  - Service temporairement indisponible |
| Postcondition | - Le feedback est enregistré  - Les administrateurs sont notifiés |

Table 26: Description textuelle "Envoyer Feedback / Recevoir Feedback"

| **Rubrique** | **Description** |
| --- | --- |
| Cas d'utilisation | Publier forum |
| Acteur | Tous les utilisateurs authentifiés (Professeur, Étudiant) |
| Précondition | - L'utilisateur est connecté  - L'utilisateur a les droits de publication dans le forum  - Le forum est actif |
| Scénario nominal | 1. L'utilisateur accède au forum 2. Le système affiche:   - Liste des catégories de discussion  - Sujets existants  - Bouton "Nouveau sujet"   1. L'utilisateur peut:   - Créer un nouveau sujet  - Répondre à un sujet existant   1. Pour une nouvelle publication:   - Remplir le titre (si nouveau sujet)  - Rédiger le contenu  - Ajouter des pièces jointes (optionnel)  - Ajouter des tags/catégories   1. Le système vérifie le contenu   6. La publication est postée |
| Scénario alternatif | 1. Contenu non conforme aux règles:   - Message d'avertissement  - Demande de modification   1. Pièce jointe non autorisée:   - Message d'erreur sur les types acceptés |
| Scénario d'exception | - Forum en maintenance  - Erreur de sauvegarde  - Timeout de session  - Problème de connexion pendant la publication |
| Postcondition | - La publication est visible dans le forum  - Les abonnés au sujet sont notifiés  - Le compteur de messages est mis à jour  - L'historique de l'utilisateur est actualisé |

Table 27: Description textuelle "Publier Forum"

| **Rubrique** | **Description** |
| --- | --- |
| Cas d'utilisation | Gestion de signalement |
| Acteur | Primaire: Admin  Secondaire: Utilisateurs (pour effectuer les signalements) |
| Précondition | - L'administrateur est authentifié  - Des signalements sont en attente de traitement  - Le système de signalement est actif |
| Scénario nominal | 1. Le modérateur accède au tableau de bord des signalements 2. Le système affiche:   - Liste des signalements en attente  - Historique des signalements traités   1. Pour chaque signalement:   - Visualisation du contenu signalé  - Motif du signalement  - Historique de l'utilisateur signalé   1. L’administrateur peut:   - Examiner le signalement  - Prendre une décision  - Appliquer une sanction  - Ajouter un commentaire de modération  5. Le système applique la décision |
| Scénario alternatif | 1. Signalement nécessitant une expertise supérieure:   - Escalade vers un administrateur  - Mise en attente avec note   1. Signalement abusif:   - Sanction possible du signaleur   1. Contestation de décision:   - Processus de révision  - Seconde analyse |
| Scénario d'exception | - Contenu déjà supprimé  - Utilisateur déjà banni  - Erreur système lors de l'application des sanctions |
| Postcondition | - Le signalement est traité et archivé  - Les parties concernées sont notifiées  - Les sanctions sont appliquées si nécessaire |

Table 28: Description textuelle "Gestion de signalement"

##### 5.4.3 Priorisation des cas d’utilisation

La priorité des cas d’utilisation est définie en fonction des besoins des utilisateurs : les plus prioritaires sont ceux qui assurent le bon fonctionnement du logiciel.

Pour hiérarchiser les cas d’utilisation, plusieurs critères sont pris en compte :

* L’importance de la fonctionnalité pour l'utilisateur,
* L’impact sur l’architecture technique,
* La complexité des fonctionnalités mises en œuvre,
* Les exigences et contraintes à respecter.

Les cas d’utilisation sont ensuite classés par ordre d’importance pour chaque acteur, ce qui permet d'établir une priorité pour leur implémentation. Dans le cadre de ce projet, les cas d’utilisation sont priorisés en fonction de deux indices :

* **Indice 1** : Cas d’utilisation à implémenter en priorité. Il s’agit des fonctionnalités les plus essentielles, constituant les bases du système.
* **Indice 2** : Cas d’utilisation important mais de moindre priorité, servant à soutenir les fonctionnalités principales.

| **Cas d'utilisation** | **Acteur(s)** | **Priorité** | **Justification** |
| --- | --- | --- | --- |
| **Se connecter** | Utilisateurs (Admin, Professeurs, Étudiants) | 1 | Fonctionnalité essentielle permettant l'accès à toutes les autres fonctionnalités. Elle a un impact direct sur l'architecture et est nécessaire au bon fonctionnement du système. |
| **Déposer Cours** | Professeur | 1 | Fonction cruciale pour les professeurs afin de mettre à disposition les ressources éducatives. Impact direct sur l'utilisation par les étudiants. |
| **Gestion utilisateur** | Admin | 1 | Nécessaire pour la gestion des comptes et des droits d'accès, essentielle pour maintenir le bon fonctionnement du système. |
| **Accès/voir KONEKTEA** | Tous les utilisateurs | 1 | Fonction clé pour permettre à tous les utilisateurs d'accéder aux informations générales et aux actualités. |
| **Recevoir Notification** | Tous les utilisateurs authentifiés | 1 | Important pour informer les utilisateurs en temps réel des nouveaux événements, messages ou contenus disponibles. |
| **Gestion profil** | Tous les utilisateurs authentifiés | 2 | Bien que nécessaire, cette fonctionnalité est moins prioritaire car elle n'impacte pas directement les autres fonctionnalités essentielles du système. |
| **Envoyer Feedback / Recevoir Feedback** | Étudiant, Professeur | 2 | Utile pour améliorer l'expérience utilisateur, mais pas essentiel au fonctionnement de base du système. |
| **Publier Forum** | Tous les utilisateurs authentifiés | 2 | Fonction utile pour la collaboration et les échanges, mais moins prioritaire que la gestion des cours ou des notifications. |
| **Gestion de signalement** | Admin, Utilisateurs (signalements) | 2 | Fonction importante pour la modération, mais elle peut être implémentée après les principales fonctionnalités. |

Table 29: Tableau de priorisation des cas d'utilsiation

##### 5.4.4 Diagramme de sequence systeme pour chaque cas d’utilisation

Le diagramme de séquence illustre les interactions entre les acteurs et le système, en mettant en évidence de manière chronologique les échanges de messages entre eux. Il fournit une explication détaillée d'un cas d'utilisation spécifique. Les éléments clés d'un diagramme de séquence incluent les messages échangés entre les lignes de vie, présentés dans l'ordre temporel. Chaque cas d'utilisation possède ainsi son propre diagramme associé.

La figure 8 represente le diagramme de sequence du systeme pour le cas d’utilisation «Authentification»

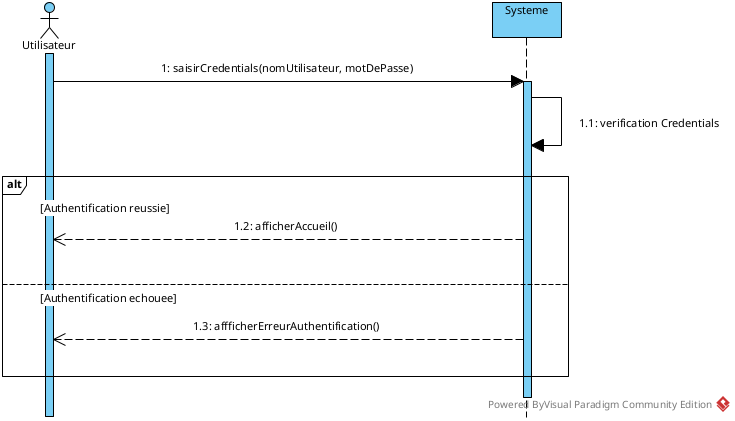


Figure 8: Diagramme de sequence de systeme de cas d'utilisation "Authentification"

La figure 9 represente le diagramme de sequence du systeme pour le cas d’utilisation «Gestion de publication»

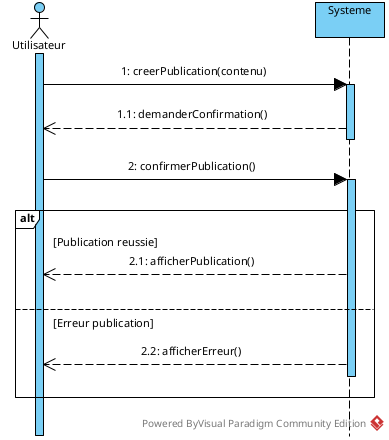


Figure 9: Diagramme de sequence de systeme pour le cas d'utilisation "Gestion de publication"

La figure 10 represente le diagramme de sequence systeme pour le cas d’utilisation «Reagir publication».

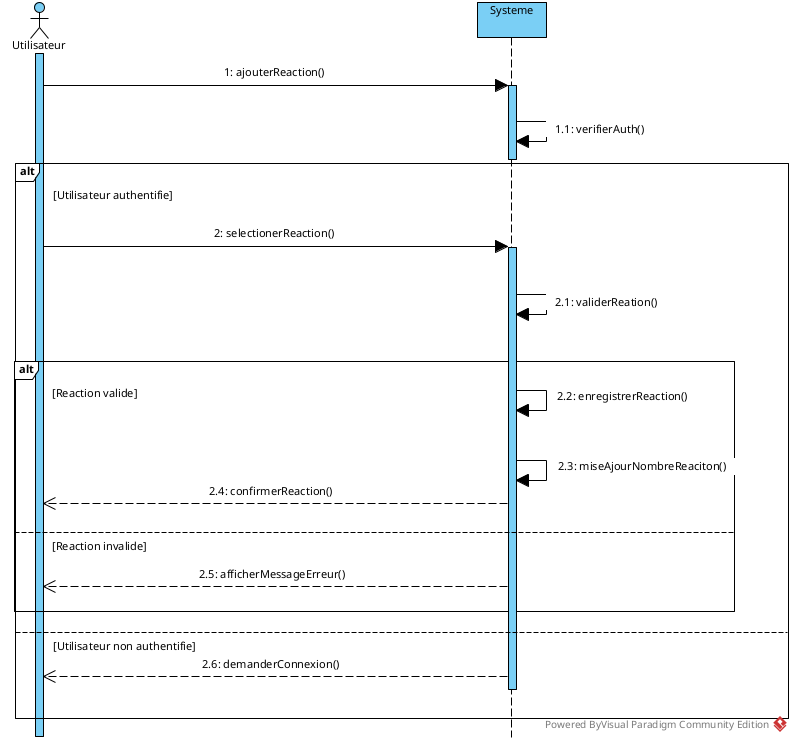


Figure 10: Diagramme de sequence de systeme pour le cas d’utilisation «Reagir Publication»

La figure 11 represente le diagramme de sequence systeme pour le cas d’utilisation «Commenter»

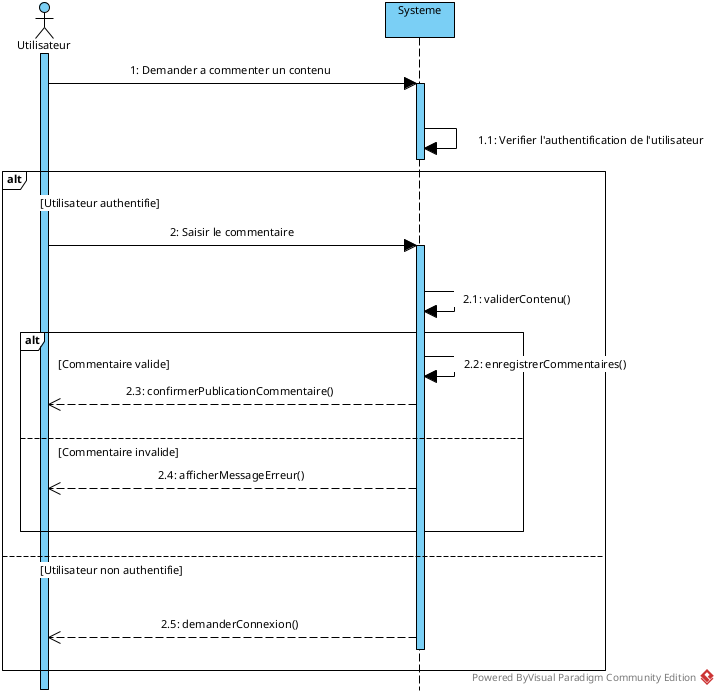


Figure 11: Diagramme de sequence systeme pour le cas d'utilisation "Commenter"

La figure 12 represente le diagramme de sequence systeme pour le cas d’utilisation «Depot de cours»

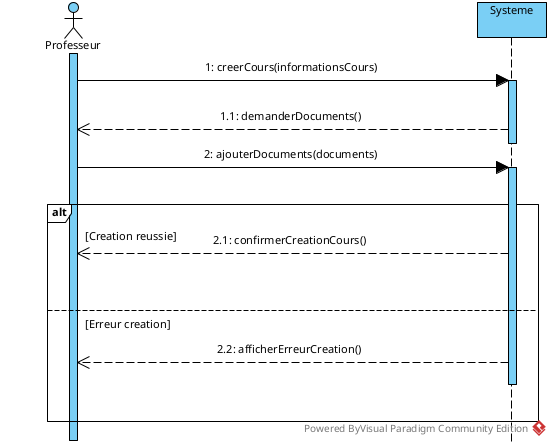


Figure 12: Diagramme de sequence systeme pour le cas d'utilisation "Depot de cours"

La figure 13 represente le diagramme de sequence systeme pour le cas d’utilisation «Gestion de signalement»

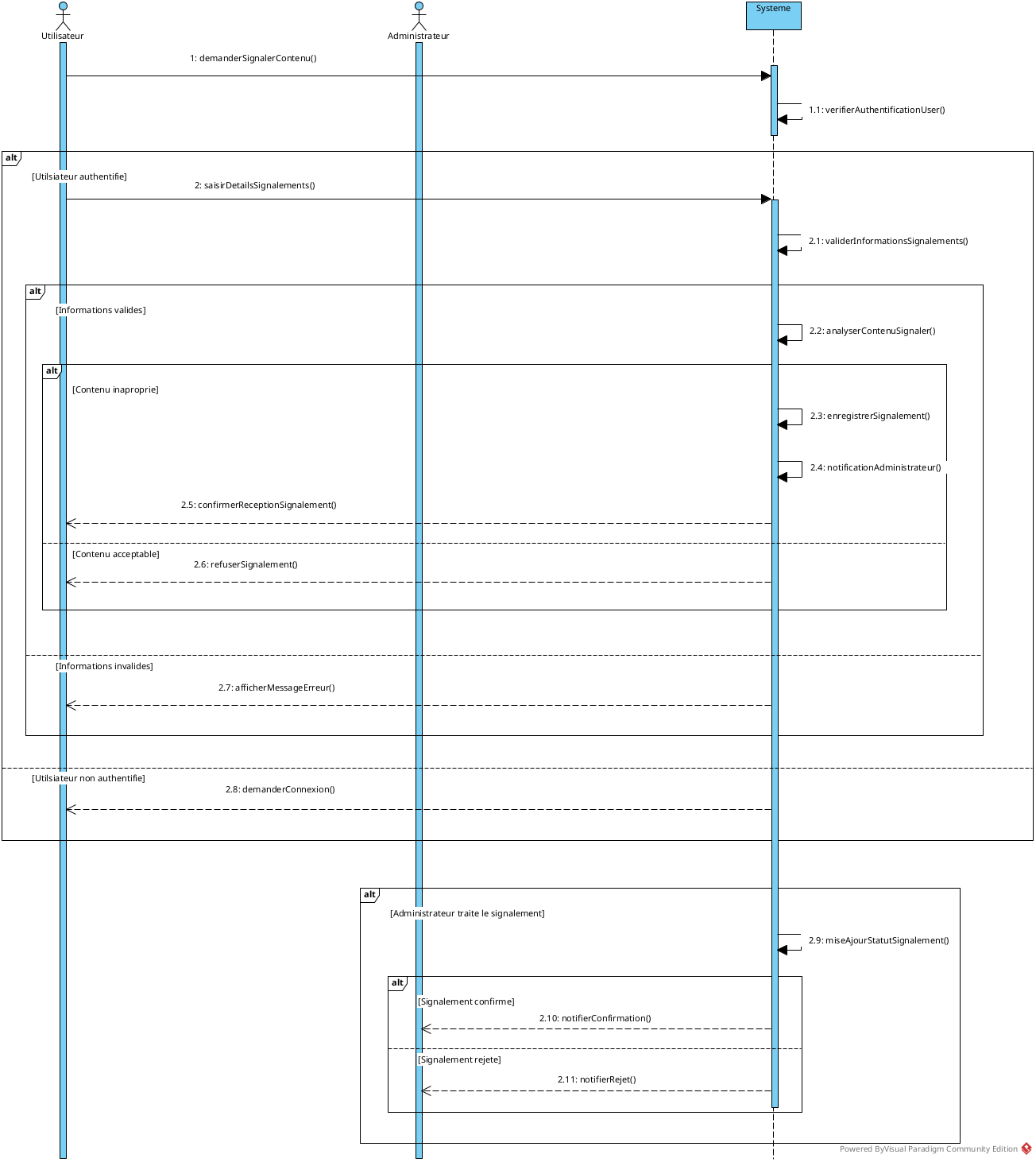


Figure 13: Diagramme de sequence systeme pour le cas d'utilisation "Gestion signalement"

La figure 14 represente le diagramme de sequence systeme pour le cas d’utilisation «Gestion Emploi du temps»

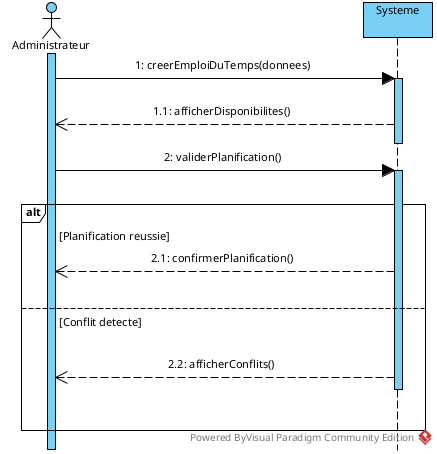


Figure 14: Diagramme de sequence systeme pour le cas d'utilisation "Gestion Emploi Du Temps"

La figure 15 represente le diagramme de sequence systeme pour le cas d’utilisation «Messagerie»

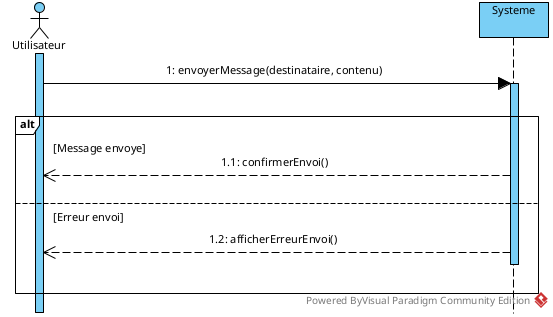


Figure 15: Diagramme de sequence systeme pour le cas d'utilisation "Messagerie"

La figure 16 represente le diagramme de sequence systeeme pour le cas d’utilisation «Gestion des archives»

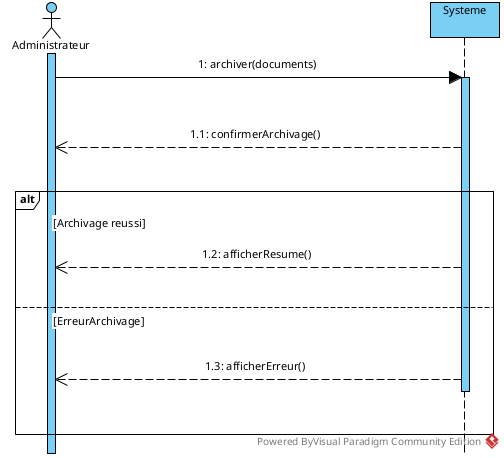


Figure 16: Diagramme de sequence systeme pour le cas d'utilisation "Gestion des archives"

La figure 17 represente le diagramme de sequence systeme pour le cas d’utilisation «Gestion Forum»

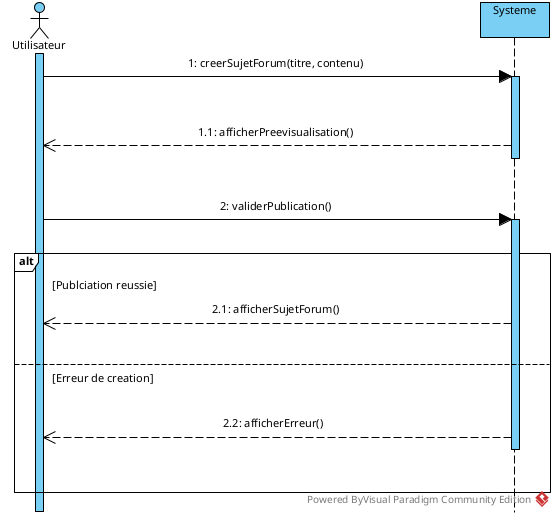


Figure 17: Diagramme de sequence systeme pour le cas d'utilisation "Gestion Forum"

La figure 18 represente le diagramme de sequence systeme pour le cas d’utilisation «Gestion Feedback»

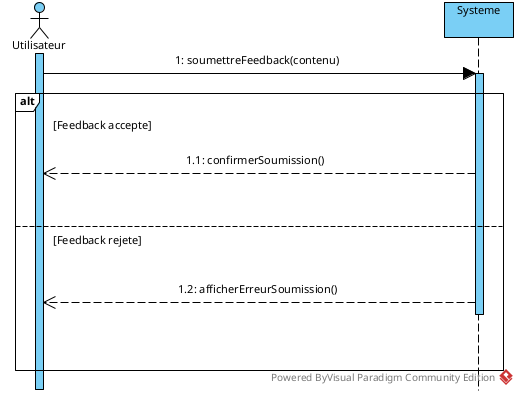


Figure 18: Diagramme de sequence systeme pour le cas d'utilisation "Gestion Feedback"

#### 5.5 Specification des besoins techniques

Pour realiser ce projet, on a besoin:

* d’un outil de modelisation: Visual Paradigm pour la modelisation UML
* d’un outil de prototypage: Figma pour les maquettes et les prototypes
* d’un environnement de developpement: Webstorm
* d’un langage de programmation; Javascript (React cote frontend et NodeJS cote backend)
* du framework Express
* d’un systeme de base de donnees MySQL

#### 5.6 Modelisation du domaine

Un modèle du domaine est une représentation visuelle des concepts liés à un domaine spécifique du monde réel. La création d’un modèle de classes pour ce domaine facilite la transition vers une modélisation orientée objet. L'analyse du domaine est distincte de celle des besoins et peut être réalisée avant, en parallèle ou après cette dernière. Cette étape d’analyse permet de développer une première version du diagramme de classes, également appelée modèle du domaine. Ce modèle vise à définir les classes représentant les entités ou concepts du domaine concerné. Il s'agit donc de créer une représentation des objets réels pertinents dans un domaine spécifique. Ces entités et concepts peuvent être identifiés à partir de la connaissance du domaine ou à travers des échanges avec des experts.

Pour construire ce diagramme, plusieurs étapes doivent être suivies :

* Identifier les entités ou concepts du domaine ;
* Déterminer et ajouter les associations et attributs pertinents ;
* Structurer et simplifier le modèle en supprimant les classes redondantes et en recourant à l'héritage.

La figure 19 represente le modele de domaine

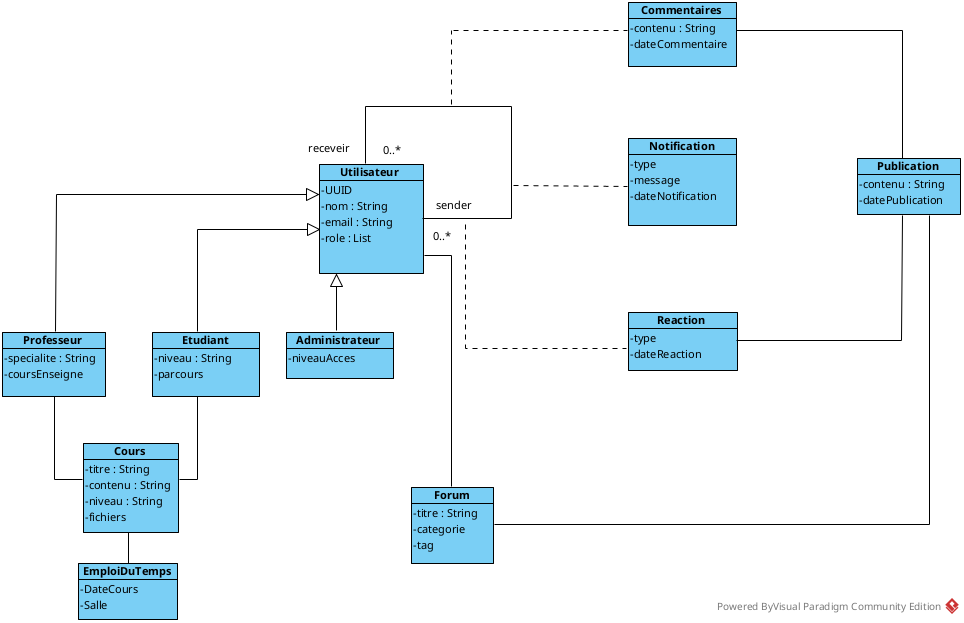


Figure 19: Modele de domaine

## CHAPITRE 6: CONCEPTION DETAILLEE

#### 6.1 Architecture du systeme

Il s'agit d'une phase du développement où les spécifications fonctionnelles et les exigences techniques sont transformées en une architecture logicielle détaillée. Cette étape implique la conception de la structure des données, le choix des algorithmes, la définition des interfaces utilisateurs et la préparation du plan de mise en œuvre technique.

**ARCHITECTURE CLIENT-SERVEUR**

L'architecture client-serveur est un modèle de conception qui divise une application en deux parties principales : le client et le serveur. Chaque composant joue un rôle spécifique dans le traitement et l’échange de données.

Ainsi, l'application se divise en deux composants essentiels :

**Le client** : Il représente l'interface avec laquelle l'utilisateur interagit. Le client est responsable de la présentation des données, de la gestion des entrées utilisateur, et de l'affichage des résultats obtenus après traitement par le serveur.

**Le serveur** : Il est chargé du traitement des requêtes envoyées par le client. Le serveur gère la logique métier, la persistance des données (accès aux bases de données) et renvoie les résultats nécessaires au client. Le serveur est le cœur du système, car il exécute les opérations complexes et gère l’intégrité des données.

Dans cette architecture, le client envoie des requêtes au serveur, qui les traite et retourne les informations ou les résultats demandés. Cette séparation permet d’améliorer la performance, la sécurité et la scalabilité de l’application.

La figure 20 represente l’architecture du systeme de l’application

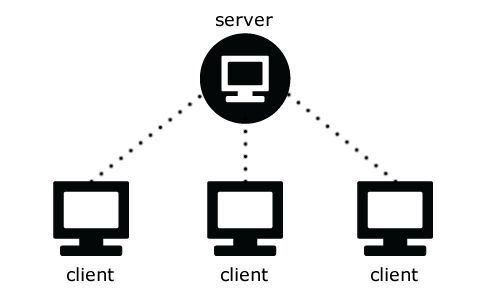


Figure 20: Architecture su systeme de l'application

Dans cette architecture, le client envoie des requêtes au serveur, qui les traite et retourne les informations ou les résultats demandés. Cette séparation permet d’améliorer la performance, la sécurité et la scalabilité de l’application.

#### 6.2 Diagramme de sequence de conception pour chaque cas d’utilisation

Les diagrammes de séquences de conception mettent en évidence les objets qui participent à la réalisation de chaque cas d’utilisation, voir les figures 21,22, 23, 24, 25, 26, 27.

**Diagramme de sequence de conception du cas d’utilisation «Authentification»**

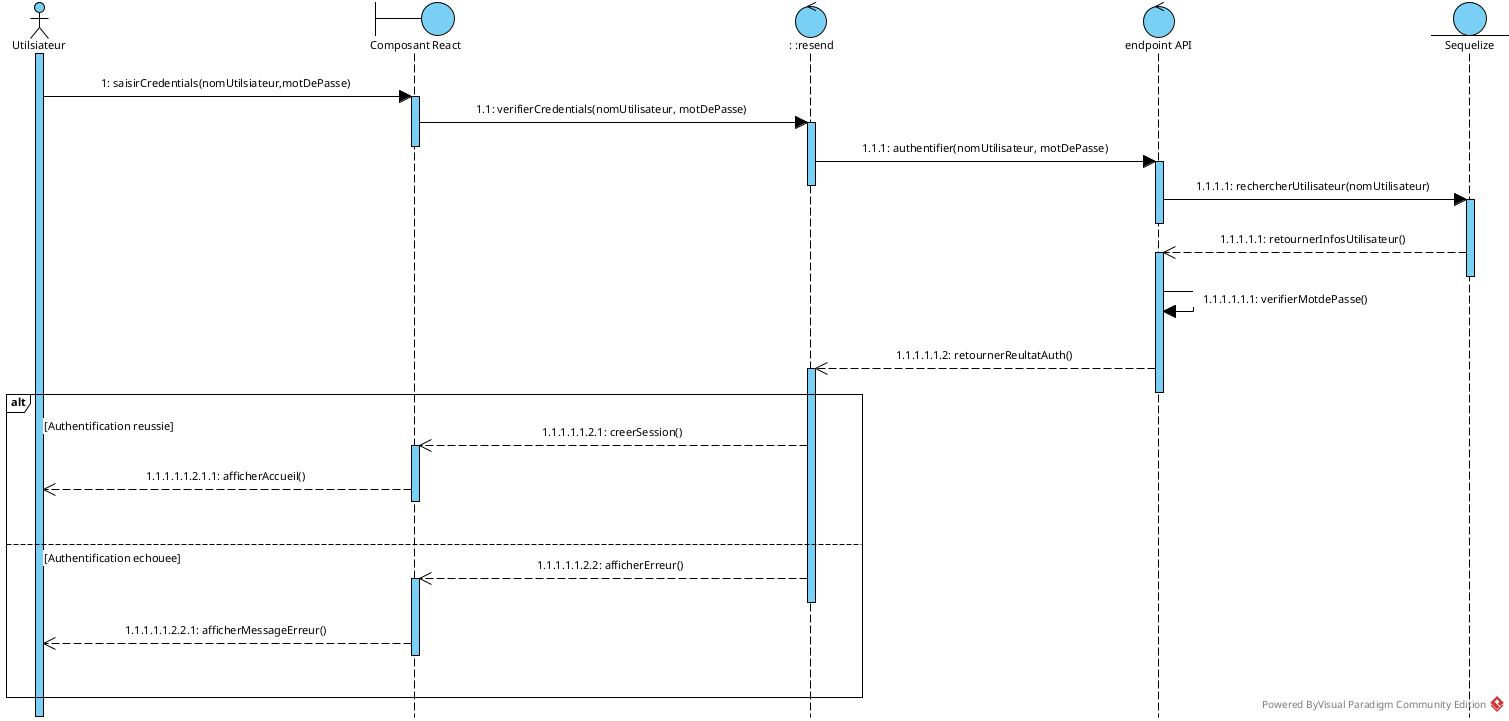


Figure 21: Diagramme de sequence de conception pour le cas d'utilisation "Authentification"

**Diagramme de sequence de conception du cas d’utilisation «Gestion signalement»**

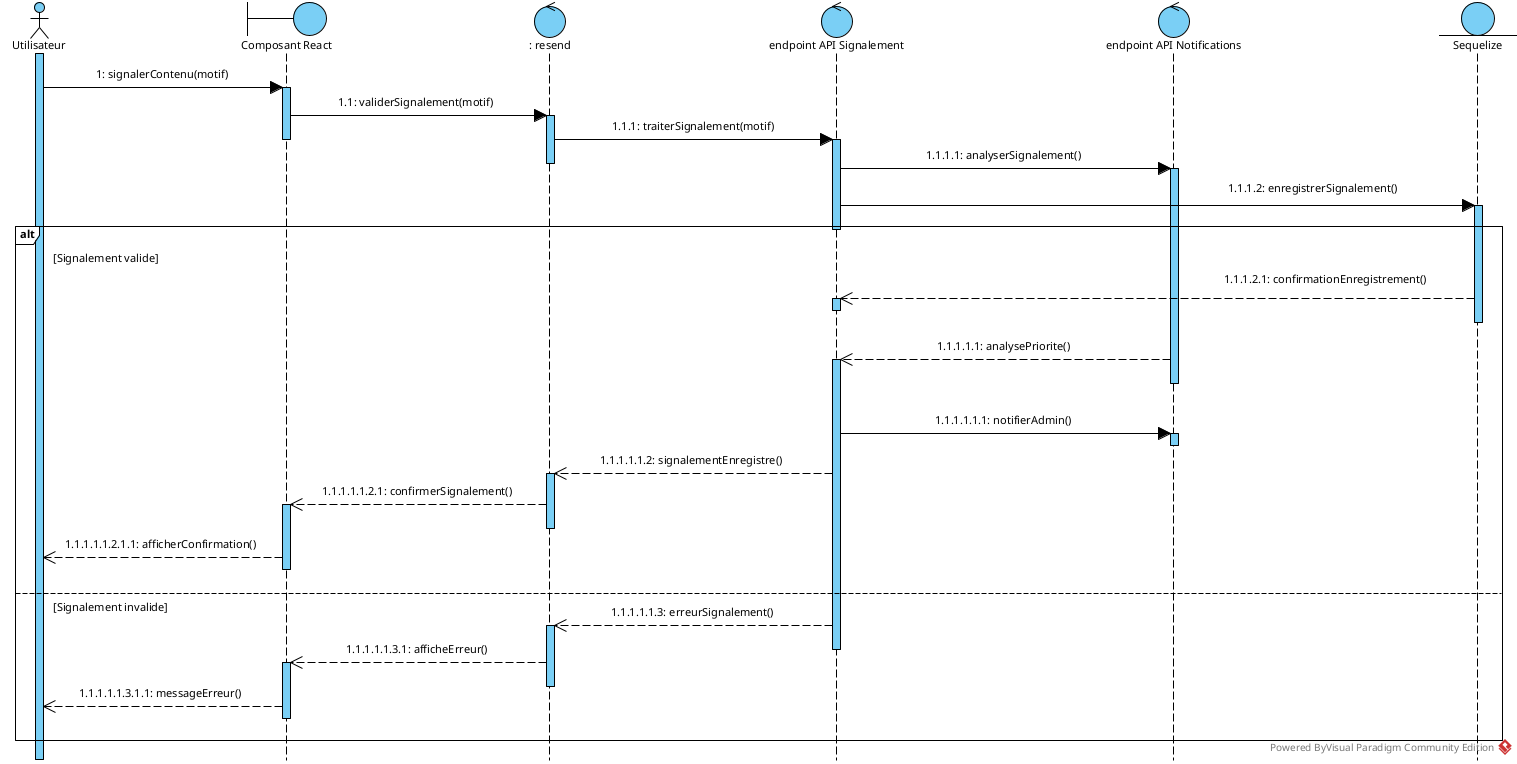


Figure 22: Diagramme de sequence de conception de cas d'utilisation "Gestion signalement"

**Diagramme de sequence de conception du cas d’utilisation «Gestion des cours - professeurs »**

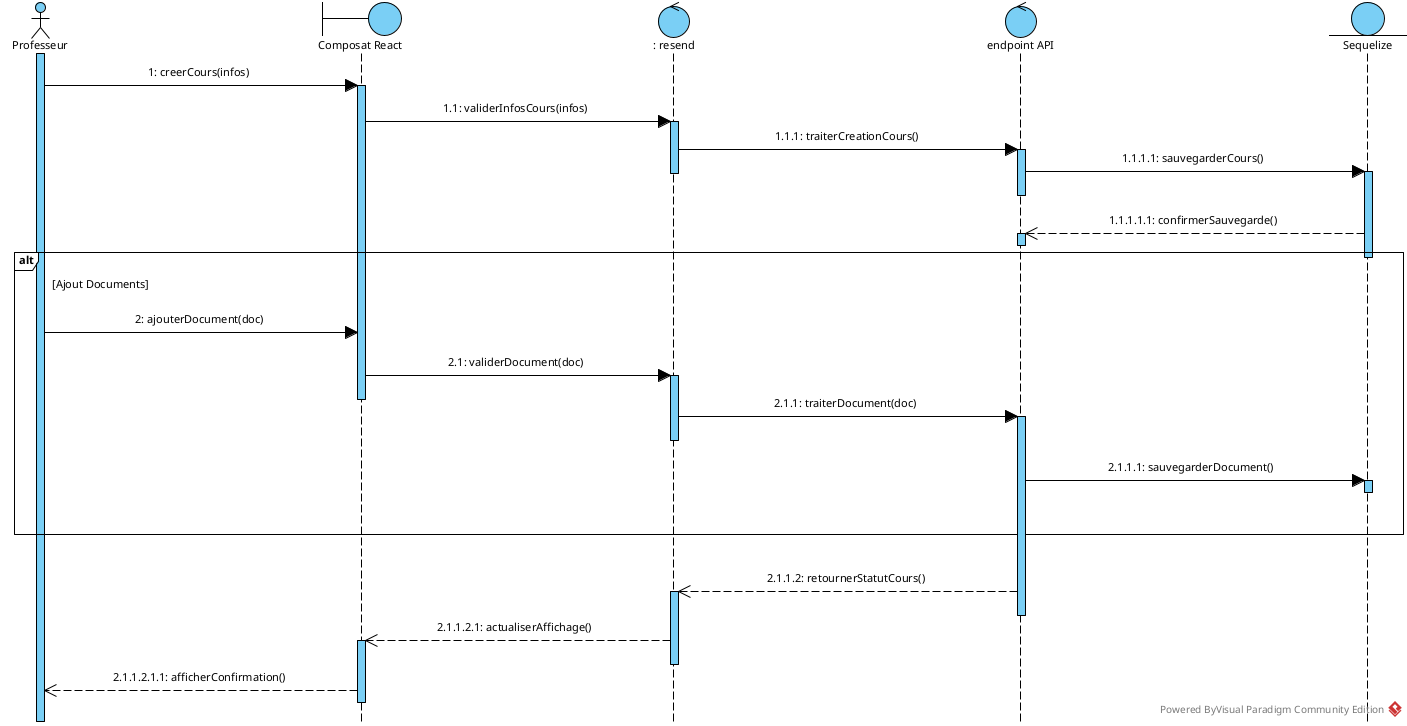


Figure 23: Diagramme de sequence de conception de cas d'utilisation "Gestion des cours - professeurs"

**Diagramme de sequence de conception du cas d’utilisation «Publier contenu »**

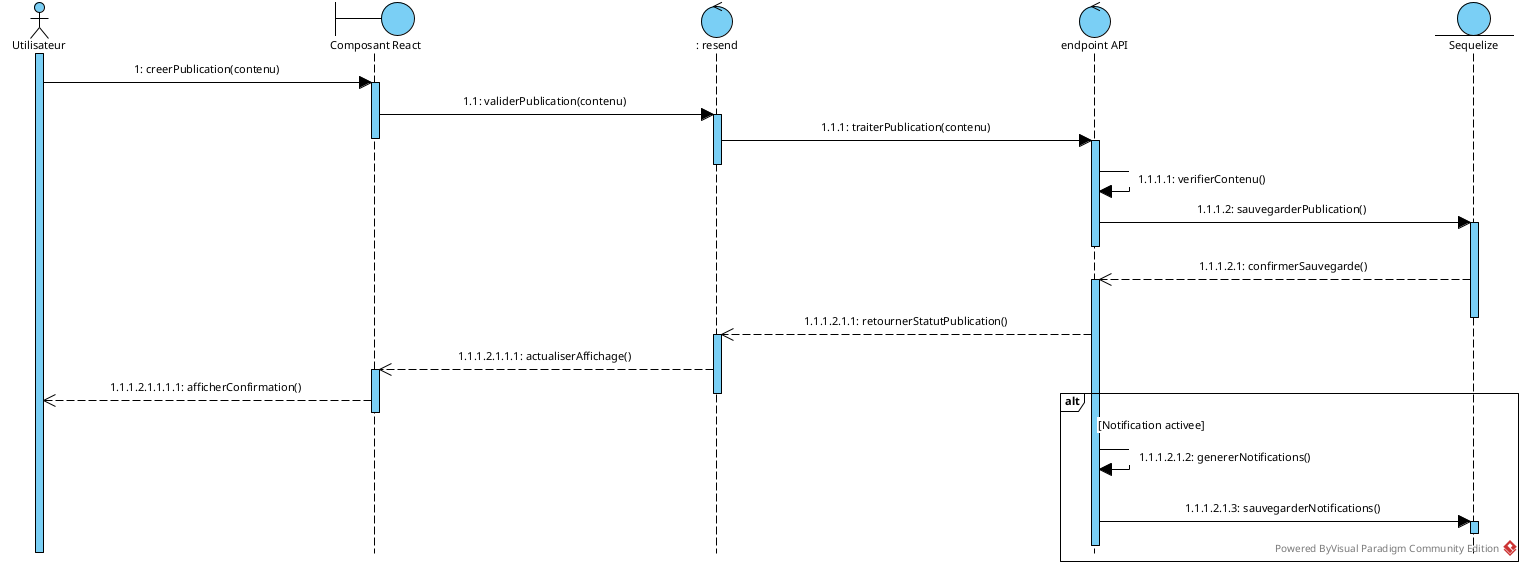


Figure 24: Diagramme de sequence de conception du cas d'utilisation "Publier contenu"

**Diagramme de sequence de conception du cas d’utilisation «Messagerie »**

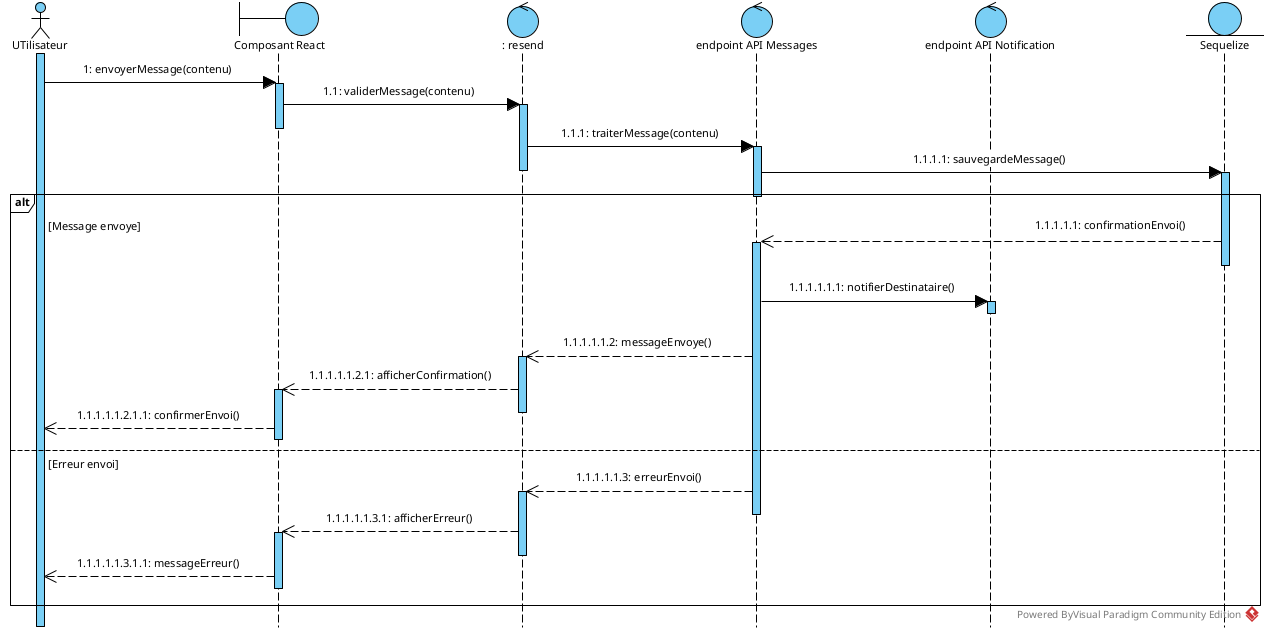


Figure 25: Diagramme de sequence de conception du cas d'utilisation "Messagerie"

**Diagramme de sequence de conception du cas d’utilisation «Gerer emploi du temps»**

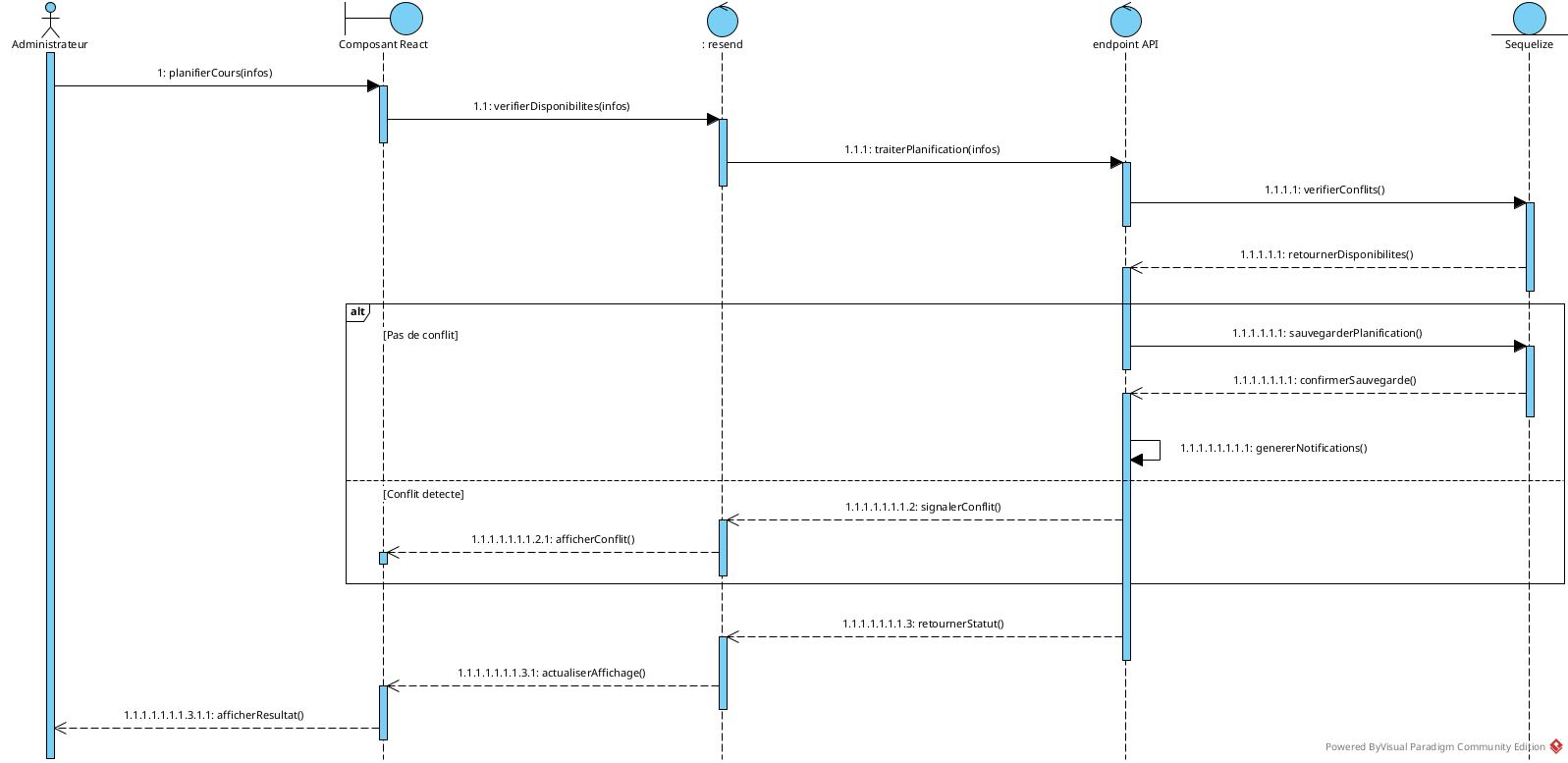


Figure 26: Diagramme de sequence de conception du cas d'utilisation "Gerer emploi du temps"

**Diagramme de sequence de conception du cas d’utilisation «Reagir Contenu»**

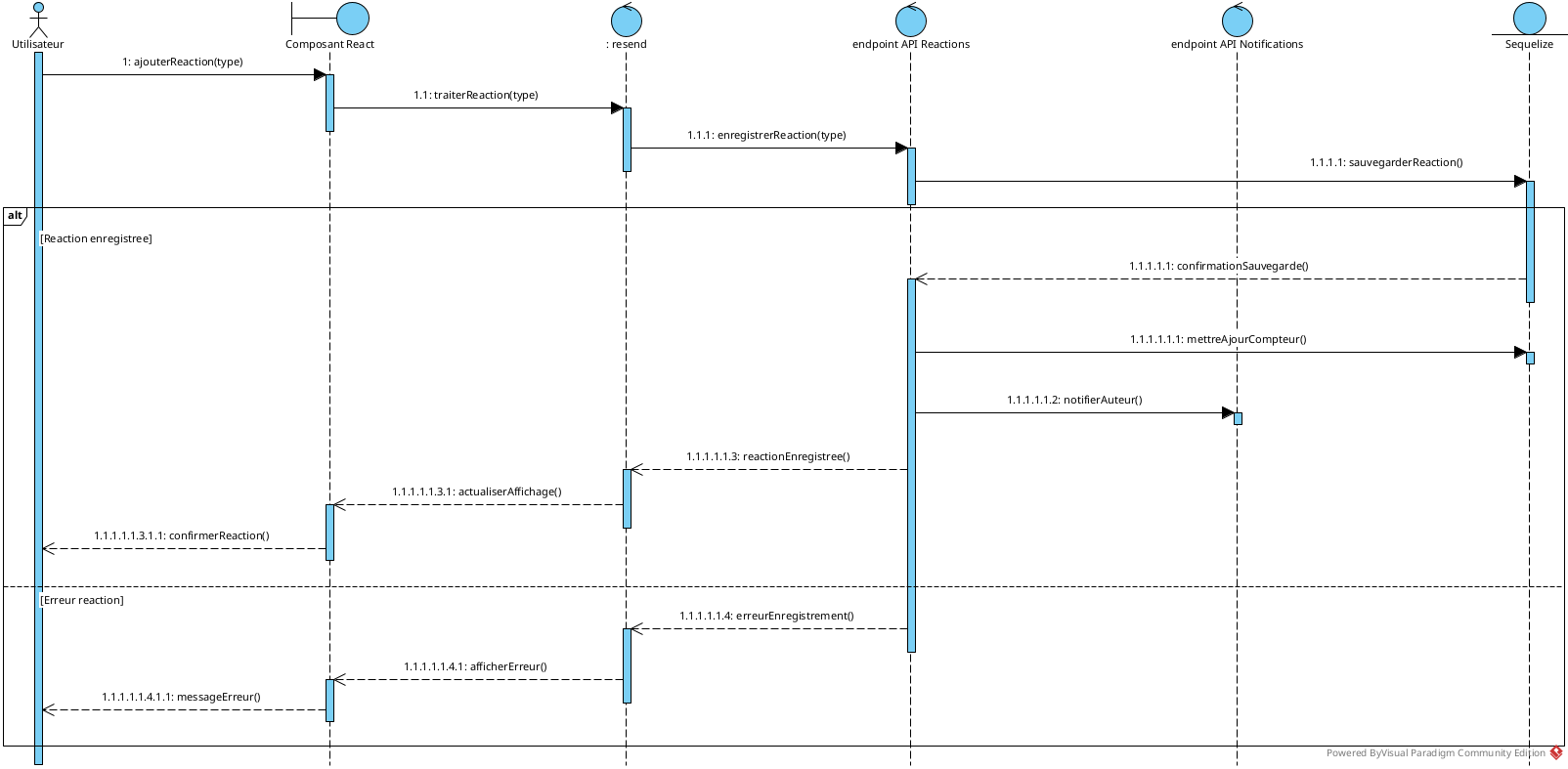


Figure 27: Diagramme de sequence de conception du cas d'utilisation "Reagir Contenu"

#### 6.3 Diagramme de classe de conception pour chaque cas d’utilisation

Le diagramme de classes illustre la structure statique d'un système en termes de classes et des relations qui existent entre elles. Il permet de modéliser les entités d'un système d'information (S.I.), en mettant en avant les informations essentielles gérées par le domaine. Ces informations sont organisées en classes, et le diagramme fait ressortir les relations possibles entre ces classes.

Ce diagramme repose sur plusieurs concepts clés : l'objet, la classe (avec ses attributs et ses méthodes), ainsi que les différents types de relations entre les classes.

Voici les principaux éléments que l'on peut retrouver dans un diagramme de classes :

* **Classe** : Représente une entité du système, incluant ses attributs et ses opérations.
* **Attribut** : Ce sont les propriétés ou caractéristiques d'une classe, décrivant l'état de ses objets.
* **Opération** : Les actions que les objets d'une classe peuvent réaliser.
* **Relation** : Décrit les connexions entre les classes, en illustrant leurs associations. Les types courants de relations incluent l'association, l'agrégation, la composition et l'héritage.
* **Multiplicité** : Précise combien d'instances d'une classe peuvent être associées à des instances d'une autre classe.
* **Visibilité** : Détermine l'accessibilité des attributs et des opérations (par exemple, public, privé, protégé).

La figure 28 represente le diagramme de classe de conception du cas d’utilisation «Authentification»

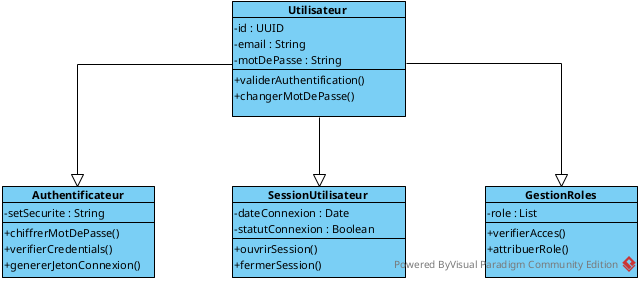


Figure 28: Diagramme de classe de conception du cas d'utilisation "Authentification"

La figure 29 represente le diagramme de classe de conception du cas d’utilisation «Gestion de cours»

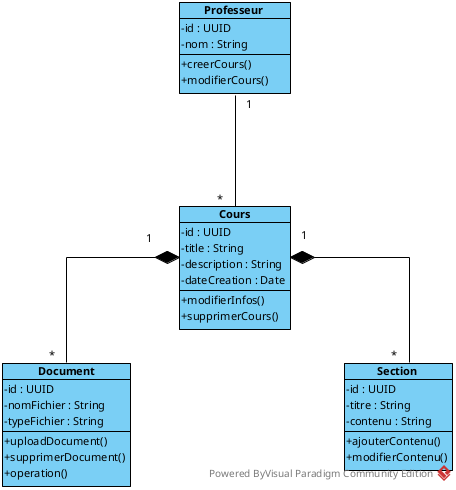


Figure 29: Diagramme de classe de conception du cas d'utilisation "Gestion de cours"

La figure 30 represente le diagramme de classe de conception du cas d’utilisation «Gestion de l’emploi du temps»

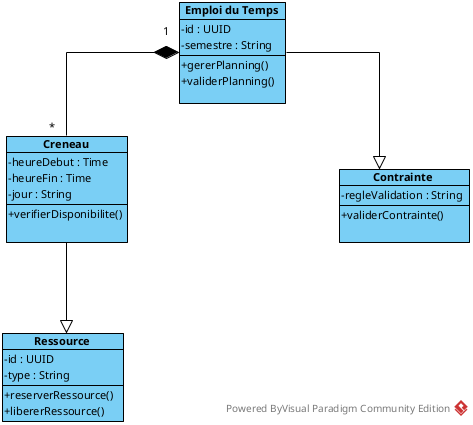


Figure 30: Diagramme de classe de conception du cas d'utilisation "Gestion emploi du temps"

La figure 31 represente le diagramme de classe de conception du cas d’utilisation «Gestion de Feedback»

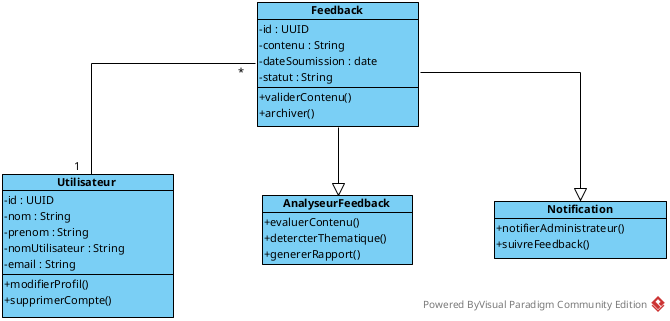


Figure 31: Diagramme de classe de conception du cas d'utilisation "Gestion de Feedback"

La figure 32 represente le diagramme de classe de conception du cas d’utilisation «Gestion d’utilisateurs»

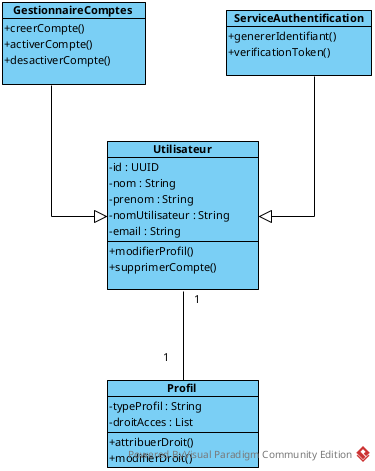


Figure 32: Diagramme de classe de conception du cas d'utilisation "Gestion utilisateurs"

La figure 33 represente le diagramme de classe de conception du cas d’utilisation «Messageries»

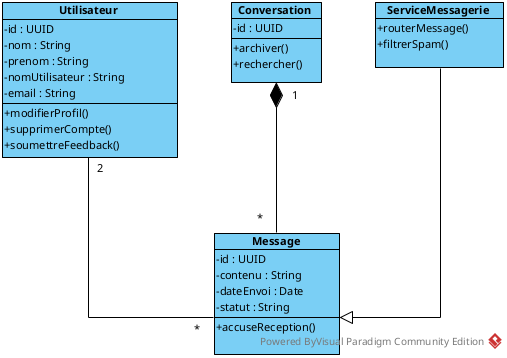


Figure 33: Diagramme de classe de conception du cas d'utilisation "Messageries"

#### 6.4 Diagramme de classe de conception globale

La figure 3 represente le diagramme de classe de conception globale

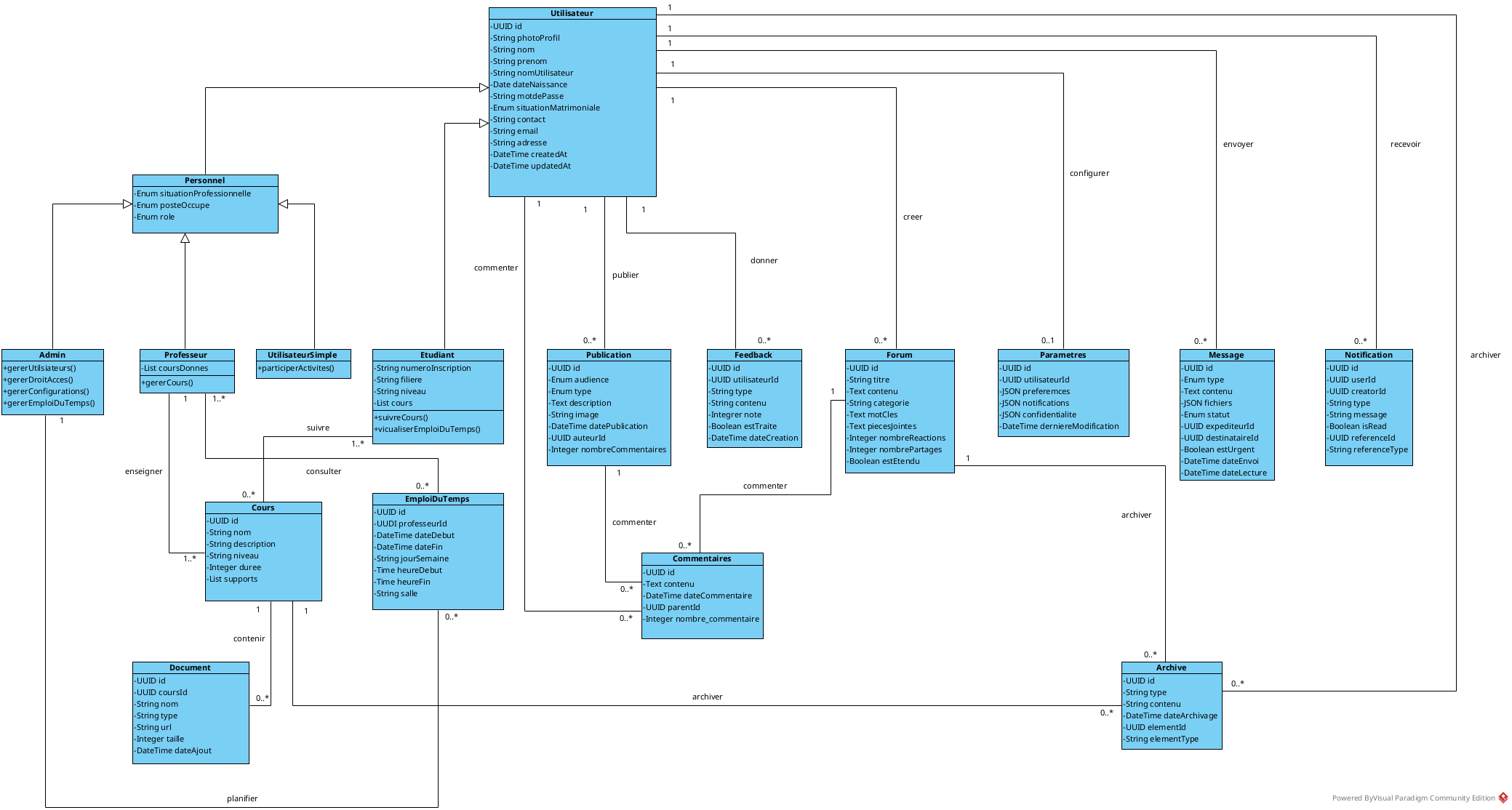


Figure 34: Diagramme de classe globale

#### 6.5 Diagramme de paquettages

Un paquetage rassemble différents éléments unitaires d'UML. Les diagrammes de paquetages illustrent ce regroupement d'éléments (comme des classes, par exemple), ainsi que les dépendances qui peuvent exister entre eux.

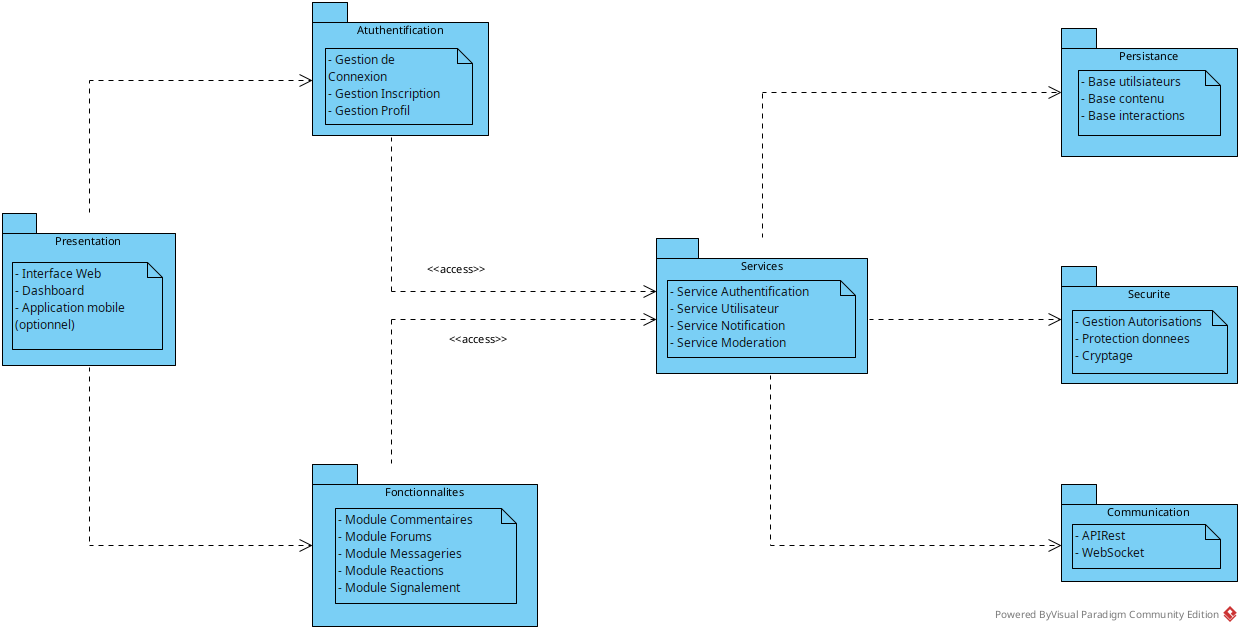
La figure 35 represente le diagramme de paquettage de l’application

Figure 35: Diagramme de paquettage de l'application

#### 6.6 Diagramme de deploiement

Le diagramme de déploiement tient compte des divers composants matériels (comme les ordinateurs, réseaux, périphériques, systèmes de stockage, etc.) et illustre leur interaction ainsi que leur répartition.

La figure 36 represente le diagramme de paquettage de l’application

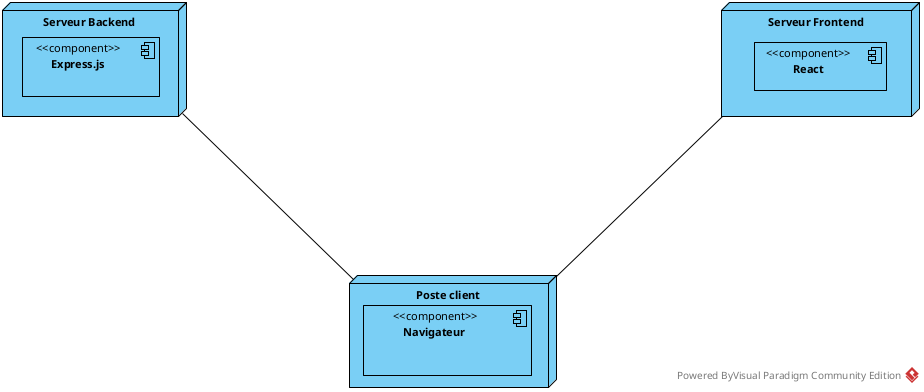


Figure 36: Diagramme de deploiement de l'application

## CHAPITRE 7: CHARTES GRAPHIQUES

Ce chapitre mettra à l’avant les critères dans un charte graphiques

#### **7.1. Définition**

La charte graphique est un véritable guide de style visuel pour une entreprise. Elle contient un ensemble de règles précises sur l'utilisation du logo, des couleurs, des typographies et d'autres éléments graphiques. Son objectif ? Garantir une identité visuelle forte et reconnaissable sur tous les supports de communication, du site web aux brochures en passant par les réseaux sociaux.

#### **7.2. Nécessité dans la réalisation d’une application**

La charte graphique est essentielle pour assurer une cohérence visuelle sur l'ensemble des supports de communication, y compris dans le développement d'une application. Elle contribue à préserver une identité visuelle distinctive et facilement reconnaissable, garantissant une expérience utilisateur uniforme. Cela permet également de renforcer la visibilité de la marque et d'accroître sa notoriété.

#### **7.3.Eléments mise en valeur**

##### **7.3.1. Nom de l’application**

Choisir le nom d'une application, c'est un peu comme choisir un surnom : il doit parfaitement refléter sa personnalité ! Pour notre plateforme dédiée à la vie universitaire, nous avons opté pour **"Konektea"**. Ce nom fait clairement écho à l'idée de connexion, mais il porte un sens plus profond. Il symbolise une passerelle vers une communauté dynamique où étudiants, enseignants et passionnés se rassemblent, échangent et collaborent.

Le nom **"Konektea"** s'inspire du verbe **"konekti"** en créole, qui signifie **"se connecter"**, et de son équivalent en français **"connecter"**. Le suffixe **"-a"** ajoute une touche dynamique, souvent retrouvée dans les langues romanes comme l'espagnol et l'italien, ce qui donne au nom une dimension active et engageante.

##### **7.3.2. Logo**

Dans l'univers du design, il existe différents types de logos, mais on peut généralement les classer en sept grandes catégories :

* Monogrammes
* Mots-symboles (ou logotypes)
* Symboles et pictogrammes
* Logos abstraits
* Mascottes
* Combinaisons d'éléments divers
* Emblèmes

Suite aux besoins identifiés et aux résultats des études menées, le choix s'est porté sur un logo de type **Mots-symboles** (ou logotype). Ce type de logo repose principalement sur l'utilisation du nom de la marque ou de l'application, souvent stylisé, pour créer une identité visuelle forte et mémorable.

**Pourquoi avoir choisi un logo de type Mots-symboles ?**

Les logotypes se distinguent par leur utilisation du nom de l'entreprise, ce qui en fait une forme de logo particulièrement efficace pour renforcer l'identité de la marque. Associé à une typographie distinctive, un logotype peut devenir instantanément reconnaissable et difficile à oublier. Le choix de la police de caractères est donc crucial, car il reflète directement l'image de l'entreprise. Par exemple, les marques de mode privilégient souvent des typographies épurées et élégantes pour véhiculer une image de luxe, tandis que les institutions juridiques ou gouvernementales optent pour des polices plus traditionnelles et sérieuses, pour inspirer confiance et stabilité.

C’est dans cette logique que le choix d’un logotype s'est imposé pour ce projet. En créant un logo typographique, il est possible de véhiculer efficacement l'identité de la plateforme tout en restant fidèle à l'image que l'on souhaite transmettre. Le style du logotype sera donc à la fois représentatif de la plateforme et suffisamment distinctif pour marquer les esprits.



Figure 37: Logo de Konektea selon le theme de la plateforme

##### 7.3.3 Police d’écriture

La typographie utilisée dans l'application joue un rôle essentiel dans l'établissement du ton et du positionnement de l'entreprise, tout en communiquant ses valeurs et son secteur d'activité. En plus de refléter l'identité de la marque, la police d'écriture doit être appropriée pour le public cible de l'application. Dans ce contexte, le choix de la typographie a été fait en tenant compte du fait que les principaux utilisateurs sont des personnes âgées.

La police sélectionnée est **Poppins**, une typographie sans empattement crée par les designers de **Indian Type Foundry**. Ce choix est motivé par son design contemporain et épuré, qui offre une excellente lisibilité, tout en étant particulièrement adaptée aux besoins d'accessibilité des utilisateurs. Sa clarté et simplicité contribuent à une expérience utilisateur fluide et agréable, essentielle pour ce public.

##### 7.3.4 Les couleurs

L'une des principales difficultés dans l'élaboration d'une charte graphique réside dans le choix et l'association des couleurs. Des recherches suggèrent qu'il est préférable de limiter le nombre de couleurs utilisées à trois pour assurer une cohérence visuelle optimale. Dans le cadre de l'UACEEM, les couleurs choisies s'inspirent principalement de celles du logo de l'organisation, en privilégiant des nuances de bleu sombre, auxquelles un accent de jaune a été ajouté pour apporter de la profondeur et de la nuance, tout en s'harmonisant avec le jaune du logo du CIP. Cette combinaison permet de maintenir une identité visuelle forte tout en restant fidèle à l'image de l'UACEEM.



Figure 38: Logo UACEEM Manakambahiny et logo CIP

***Choix des couleurs***

La combinaison de ces trois couleurs est utilisee pour le design de l’interface de l’application

**RGB: 78 - 78 - 78 ( #4E4E4E )**



**RGB: 237 - 182 - 64 ( #EDB640 )**

**RGB: 36 - 48 - 99 ( #243063 )**

Figure 39: Palette de couleur utilise dans l'integration de la plateformes

|  |  |
| --- | --- |
| **RGB: 78 - 78 - 78 ( #4E4E4E )** | Utilise pour les textes et les titres |
| **RGB: 237 - 182 - 64 ( #EDB640 )** | Utilise comme couleur secondaire |
| **RGB: 36 - 48 - 99 ( #243063 )** | Utilise comme couleur primaire |

Table 30: Utilisation des couleurs

# PARTIE III: REALISATION

## CHAPITRE 8: MISE EN PLACE DE L’ENVIRONNEMENT DE DEVELOPPEMENT

#### 8.1 Installation et configuration des outils

**Figma: Outil de prototypage**

Figma est une application web et de bureau qui offre un espace de travail unifié pour les designers, les développeurs et les autres membres d'une équipe. Il permet de concevoir des maquettes, des prototypes interactifs et des systèmes de design complets, tout en favorisant la communication et la collaboration en temps réel.

C’est aussi un outil puissant qui combine les avantages du design vectoriel avec les fonctionnalites de prototypage. Il permet de creer des interfaces haute fidelite, d’animer des elements et de tester l’experience utilisateur avant le developpement.

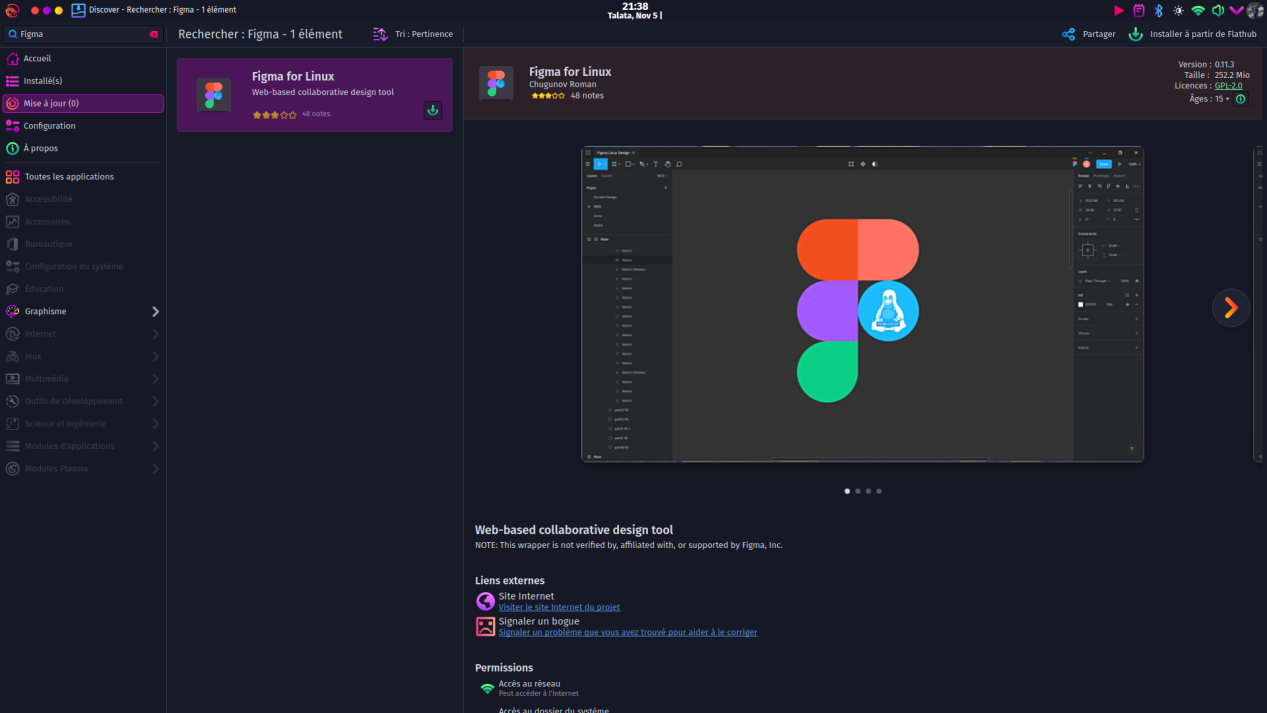


Figure 40: Installation de Figma-Linux

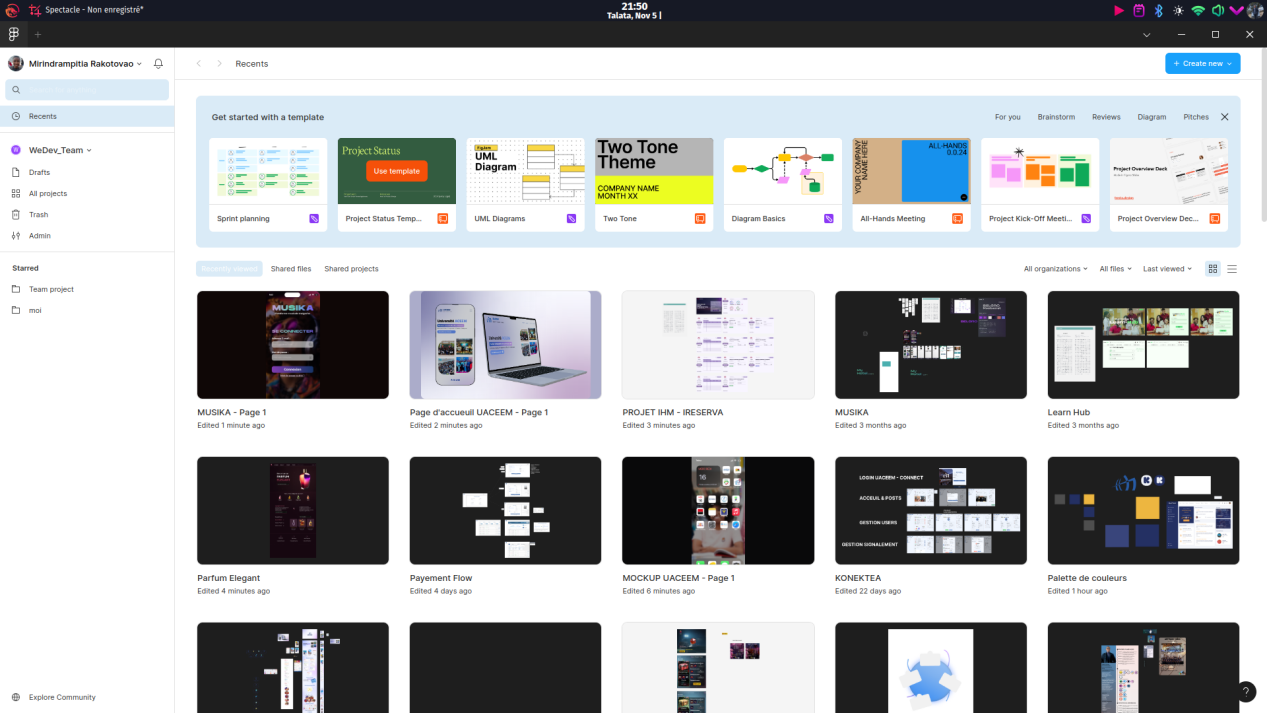


Figure 41: Interface d'acceuil Figma-Linux

**Webstorm: Outil de developpement**

WebStorm est un logiciel qui offre un environnement de travail complet pour les développeurs web. Il inclut un éditeur de code intelligent avec des fonctionnalités comme la complétion automatique, la refactorisation, la détection d'erreurs en temps réel, et la navigation facile dans le code. WebStorm prend en charge un large éventail de technologies web modernes et est particulièrement apprécié pour sa productivité et sa personnalisation.

Les principales fonctionnalités de WebStorm :

* **Éditeur de code intelligent :** Complétion automatique, refactorisation, débogage...
* **Intégration de version control :** Git, SVN, Mercurial...
* **Support de nombreux frameworks :** React, Angular, Vue, Node.js...

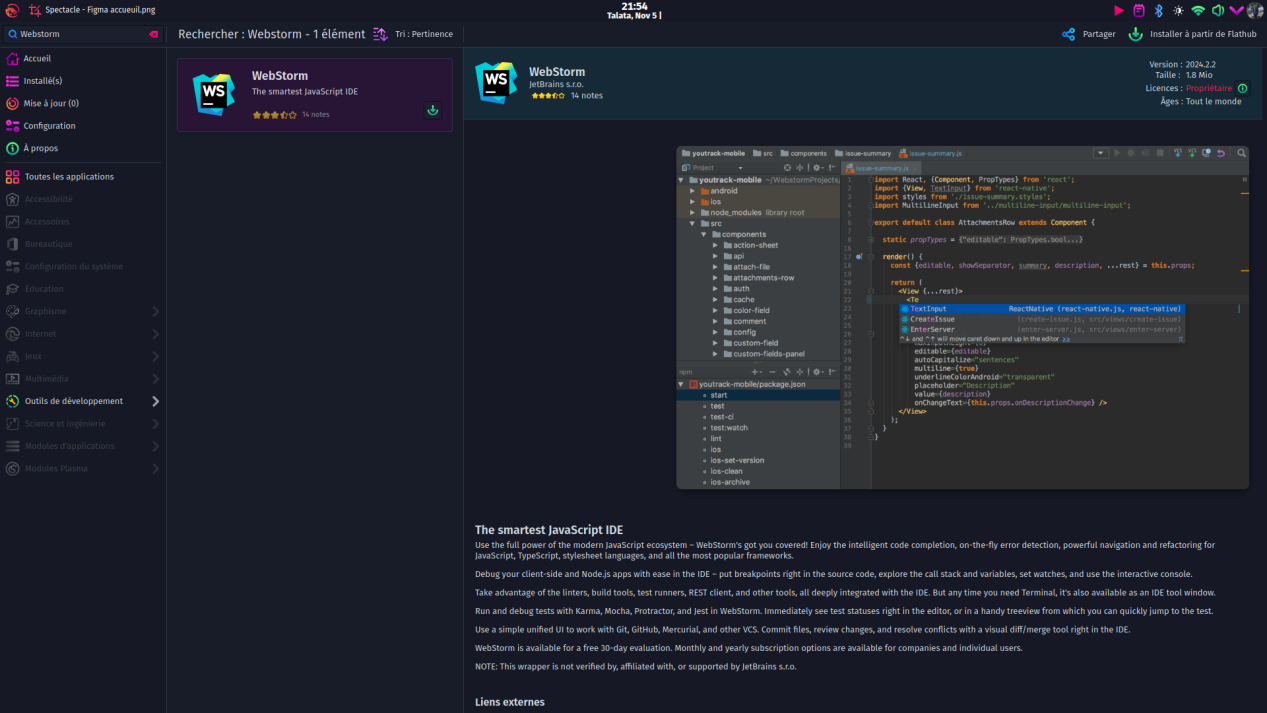


Figure 42: Installation de Webstorm

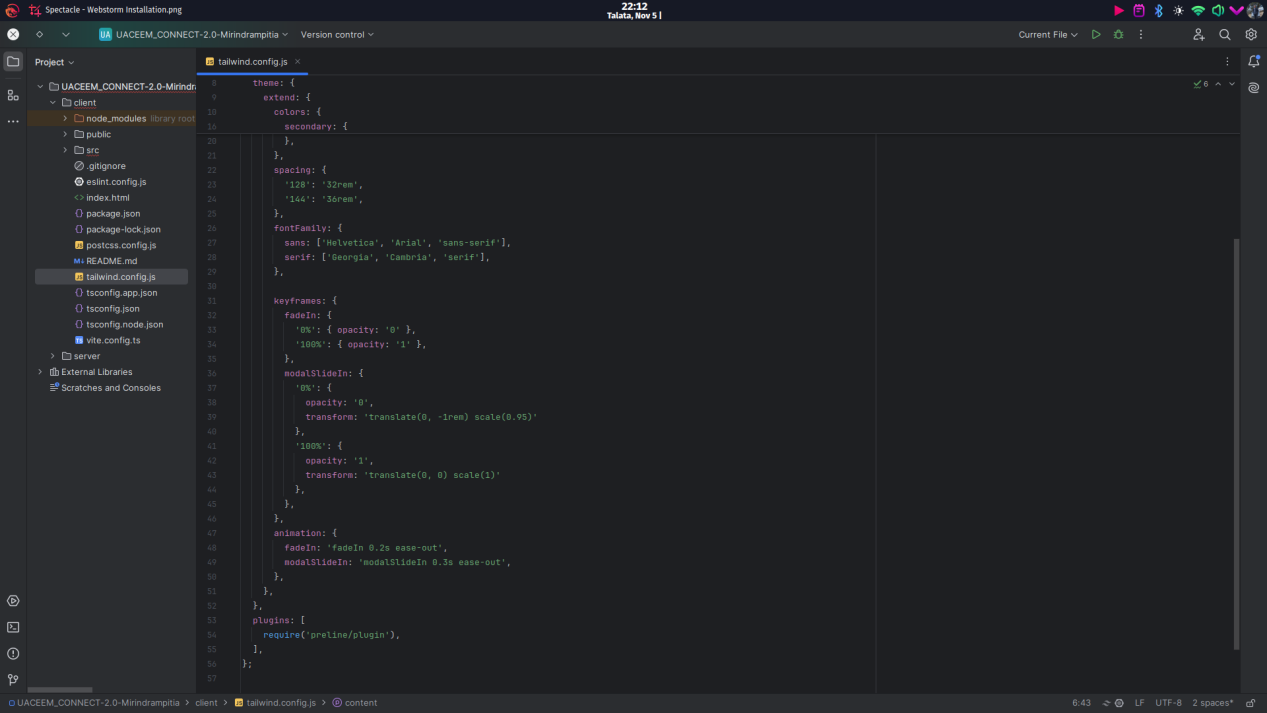


Figure 43: Interface Webstorm

**Visual Paradigm: outil de modelisation**

**Visual Paradigm** est un outil puissant qui offre une gamme complète de fonctionnalités pour la modélisation UML (Unified Modeling Language). Il permet de créer des diagrammes de classe, de séquence, d'état, d'activité et bien d'autres, facilitant ainsi la conception, l'analyse et la documentation de systèmes logiciels. En outre, Visual Paradigm intègre des fonctionnalités de gestion de projet, de génération de code et d'ingénierie inverse.

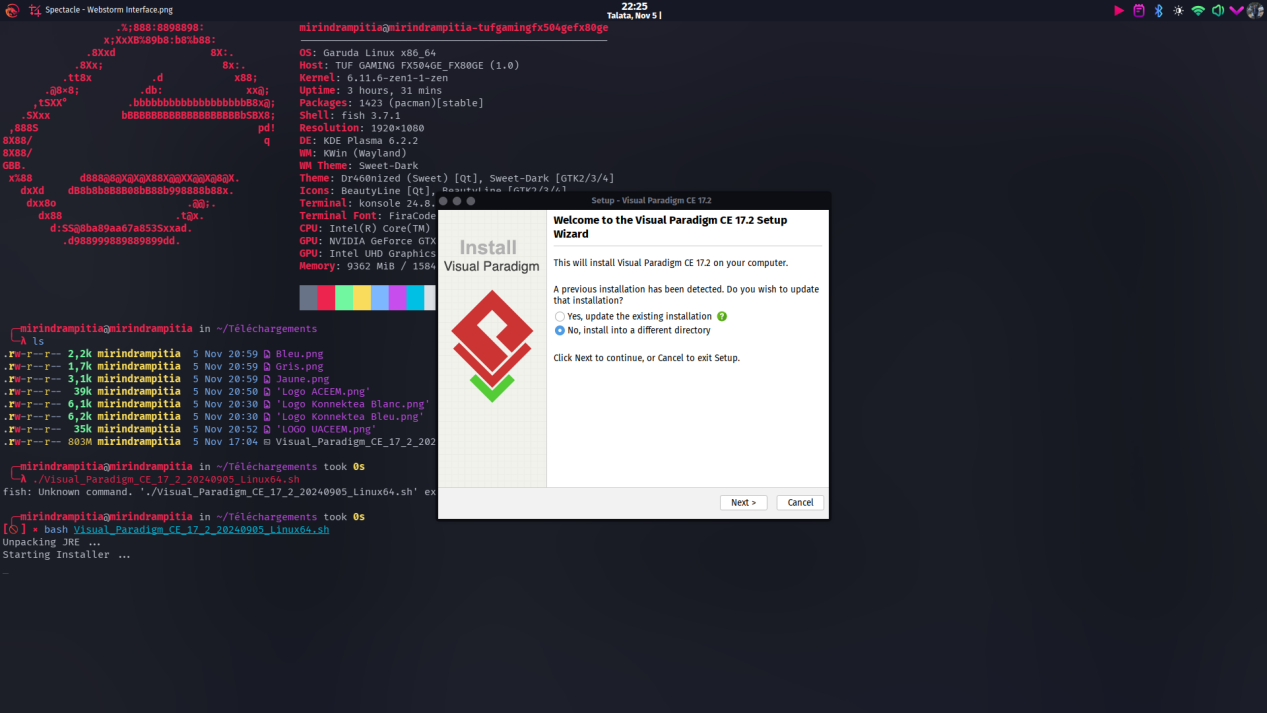


Figure 44: Installation Visual Paradigm 17.2

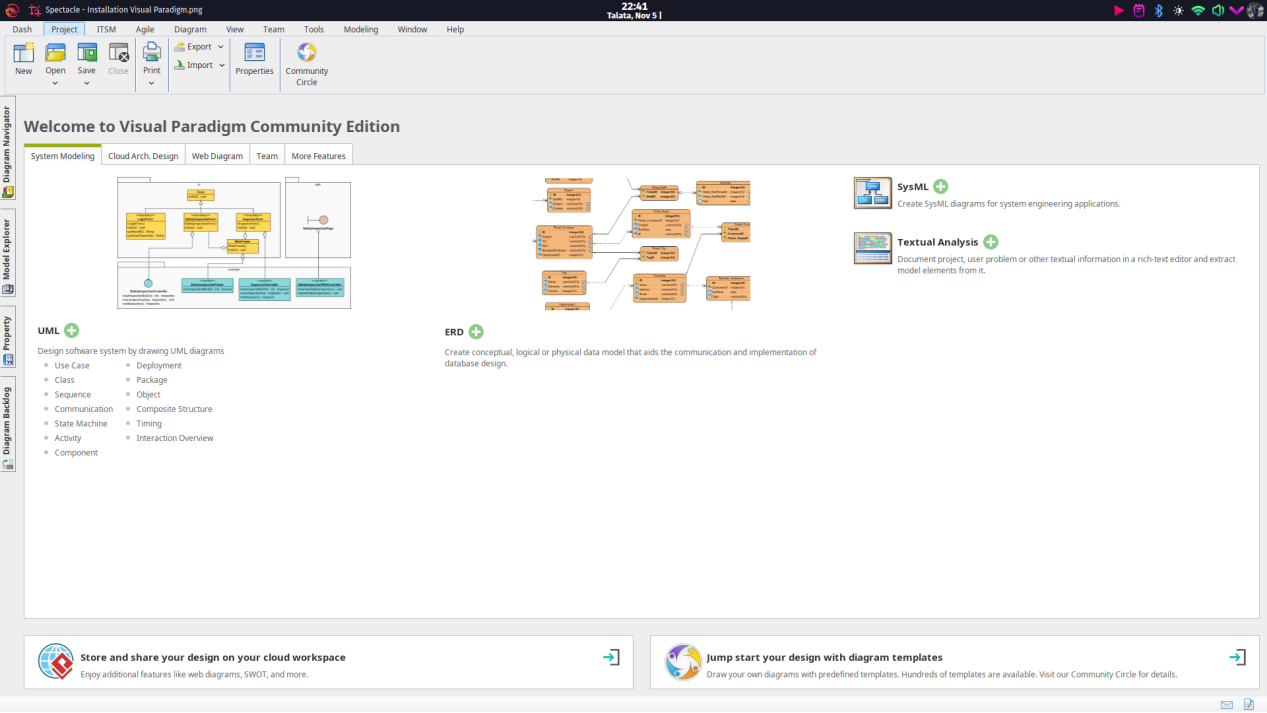


Figure 45: Page d'acceuil Visual Paradigm

#### 8.2 Architecture de l’application

On a vu dernierement que notre appplication suit l’architecture client-serveur.

L’architecture client-serveur est une architecture moderne, ou le front-end (dans notre cas React) et le backend (dans notre cas Node.js/Express) sont clairement separes.

L’application se compose en 5 couches:

- **La couche presentation (Frontend):** la partie interface utilisateur ou client de l’application, geree par React.js. Elle a pour responsabilite d’afficher les donnees aux utilisateurs et gerer l’interaction utilisateur. Elle appelle les services d’API REST pour obtenir ou envoyer des donnees au serveur. Et elle maintient l’etat de l’application cote client avec des hooks, ou des outils de gestion d’etat comme Redux.

- **La couche API/Communication (Backend):** le serveur (backend) qui est gere par Node.js et Express.js. Elle gere les routes API qui recoivent les requetes du frontend. Elle renvoie aussi les reponses au frontend au format JSON, avec les donnees ou les resultats de l’operation. Et elle interprete les requetes HTTP (GET,POST,PUT,DELETE) envoyees par le frontend.

**- La couche logique metier (Services/Controlleurs):** C’est dans cette couche que les controlleurs sont definis dans une application Express.. Elle contient la logique metier ou les services.Elle gere la ccordination entre la couche API et la couche d’acces aux donnees

**- La couche d’acces aux donnees (Modeles / ORM):** Cette couche est geree par Sequelize dans cette architecture. Elle interagit directement avec la base de donnees. Elle maniple les modeles qui representent les tables dans la base de donnees et gere les relations entre les differentes tables.

**- La couche de persistance** **(Base de donnees):** Elle stocke les donnees de maniere permanente et fournit des mecanismes de requette pour extraire, inserer, mettre a jour ou supprimer les donnees.

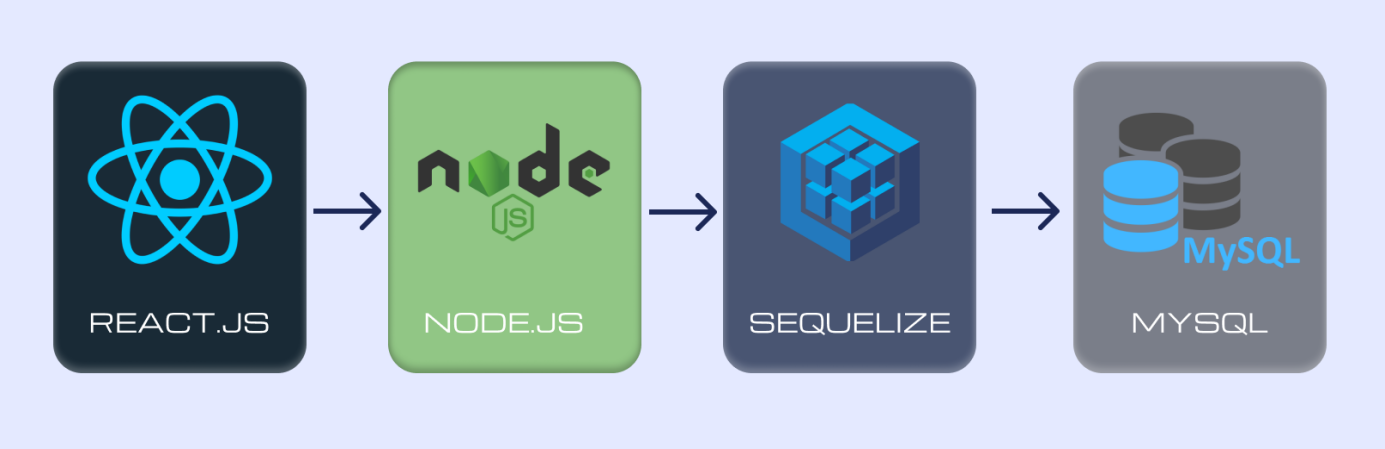


Figure 46: Architecture de l'application

## CHAPITRE 9: DEVELOPPEMENT DE L’APPLICATION

#### 9.1 Creation de la base de donnees

Pour la création de la base de données, nous avons opté pour un ORM (mapping objet-relationnel) nommé **Sequelize**. Il s'agit d'une solution d'accès aux données pour Node.js qui simplifie les interactions entre une application et sa base de données en offrant une interface de programmation conviviale. Sequelize permet aux développeurs de définir des modèles en JavaScript qui représentent les tables de la base de données, puis génère automatiquement les requêtes SQL en fonction des opérations définies dans le code. De plus, il intègre des outils de migration qui facilitent la gestion des modifications du schéma de la base de données, tout en assurant la conservation des données existantes et le suivi des versions des changements effectués.



Modele Sequelize Migraation Sequelize Requete Sequelize

Figure 47: Concept de Squelize

#### 9.2 Codage de l’application

Dans cette section, nous mettons en avant plusieurs extraits de code source de l'application. Ces exemples ont pour objectif de fournir un aperçu du fonctionnement interne de l'application, en illustrant certains aspects clés de son architecture et de sa logique de développement. Ils permettent de mieux comprendre les choix techniques effectués, ainsi que la manière dont les fonctionnalités sont implémentées pour offrir une expérience utilisateur fluide et efficace.

La figure 48 nous montre la creation d’un message prive entre les utilisateurs avec Socket.IO

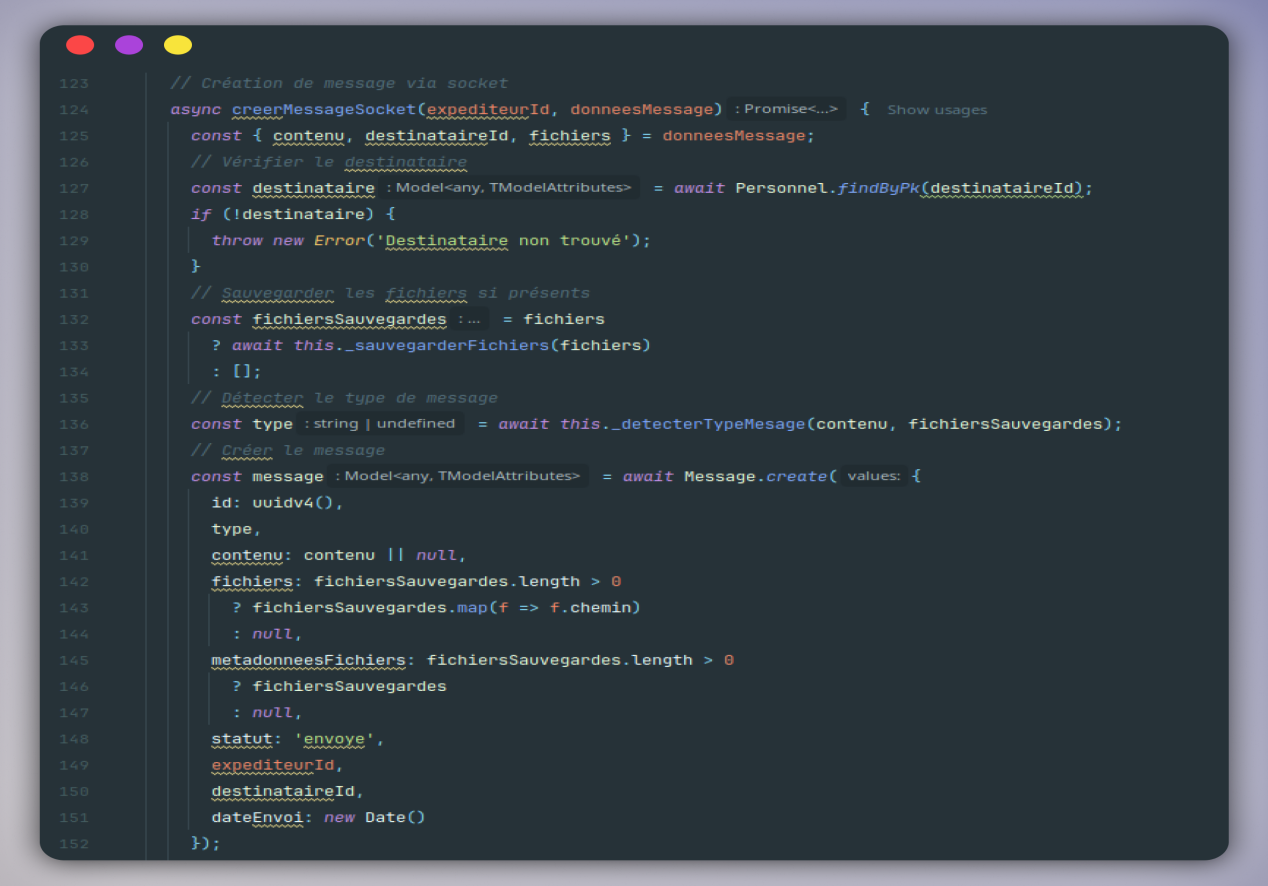


Figure 48: Message prive entre utilisateurs avec Socket.IO

La figure 49 nous montre la creation d’un forum

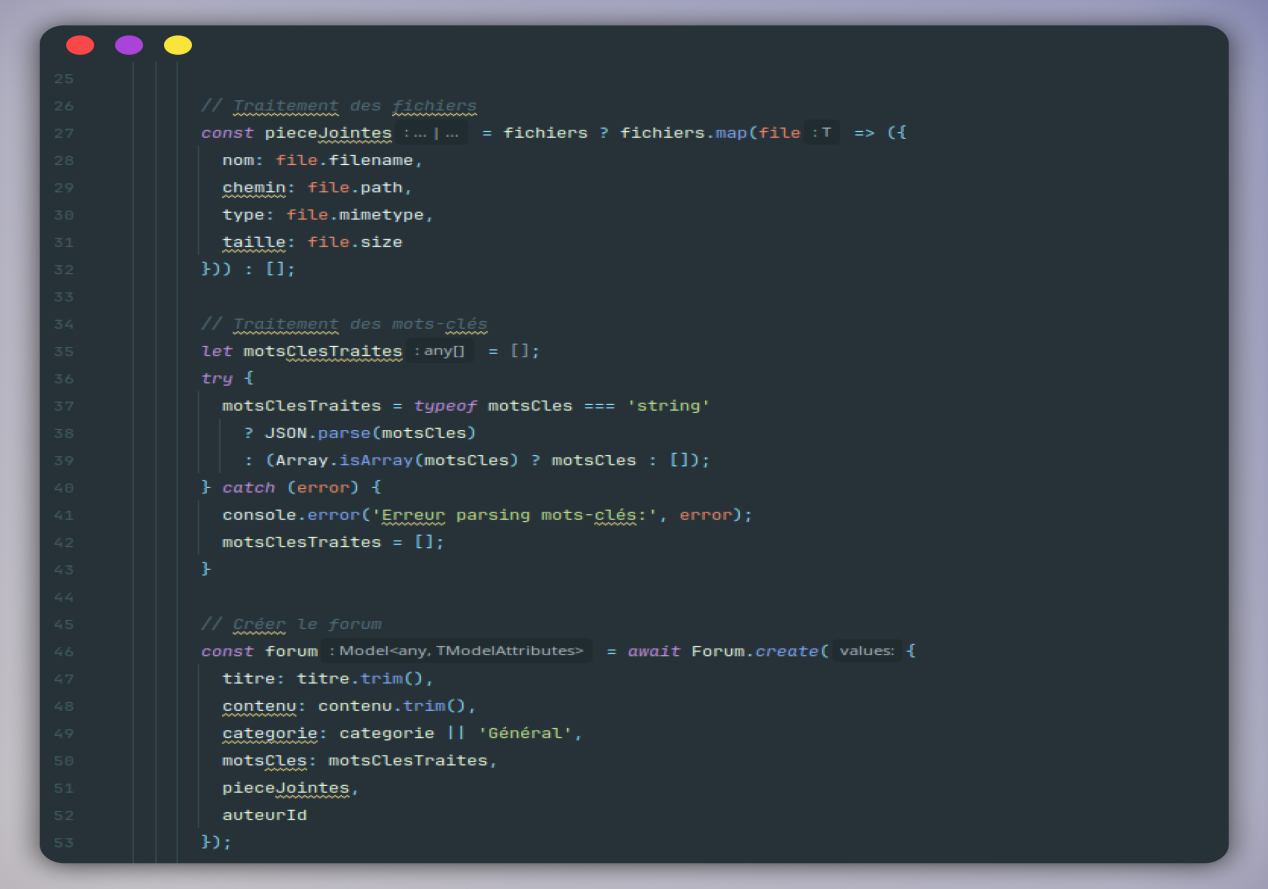


Figure 49: Creation d'un forum

#### 9.3 Presentation de l’application

Pour illustrer efficacement la présentation, nous avons choisi d'inclure plusieurs captures d'écran. Ces visuels permettront de démontrer clairement les fonctionnalités et les performances de l'application. En complément, nous présenterons également une capture du prototype de l'application, afin de mettre en avant l'évolution du design et la manière dont les concepts initiaux ont été traduits en une interface fonctionnelle. Cela offrira une vision complète du parcours de développement, depuis la phase de prototypage jusqu'à la version finale.

La figure 50 montre la phase de prototypage de l’application a partir de Figma

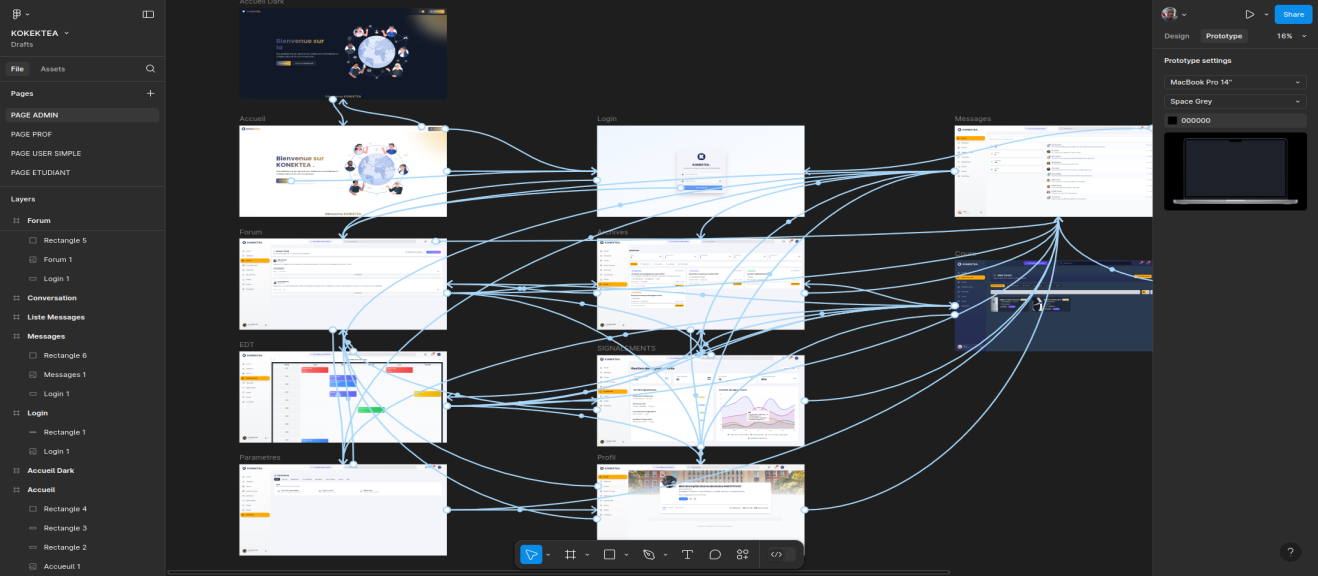


Figure 50: Prototypage & Maquettage de l'application

La figure 51 montre la page d’accueil de la plateforme KONEKTEA



Figure 51: Page d'accueil KONEKTEA