



UNIVERSITE DE FIANARANTSOA

ECOLE NATIONALE D'INFORMATIQUE

MEMOIRE DE FIN D'ETUDES
POUR L'OBTENTION DU DIPLOME DE MASTER PROFESSIONNEL
Mention : Informatique
Parcours : Informatique Générale

Intitulé :

MISE EN PLACE D'UNE PLATEFORME E-LEARNING

Présenté le 1^{ER} décembre 2022

Par Mademoiselle RAJAONARISON Rojo Nirina

Membres du Jury :

Président : Monsieur RAKOTOASIMBAHOAKA Cyprien Robert, Maître de Conférences

Examinateur : Monsieur RATOVONDRAHONA Josué, Docteur en Informatique

Rapporteurs : Monsieur RATIARSON Venot, Maître de Conférences

Monsieur BAKARY Gaetan Jonathan, Gérant de l'entreprise iTS

CURRICULUM VITAE



RAJAONARISON Rojo Nirina

24 ans

Profil

Etudiante en deuxième année de formation en Master Professionnelle à l'Ecole Nationale d'Informatique, suivant le parcours : Informatique générale. Développeur Web expérimenté possédant des capacités de codage avancées et un enthousiasme pour les nouvelles améliorations.

Contact

📞 +261 34 86 470 79

✉️ Rojonirina3080@gmail.com

📍 Amboditsiry près Supermaki

Compétences en Informatique

- Technologies web :
- PHP, JSP, Angular, React, NodeJS, SailsJS, JSP (Java)
- Frameworks :
- CodeIgniter, Laravel
- Base de données :
- MySQL, PostgreSQL
- Language de programmation :
- C / C++, Java SE
- Méthodes et langage de conception:
- MERISE, UML , Agile

Centre d'intérêt

- Musique
- Photographie

EXPÉRIENCES PROFESSIONNELLES

iTeam-School (iTS) Antananarivo

Stagiaire | Juin 2022 – Novembre 2022

Missions réalisées :

- Conception et réalisation d'une plateforme e-Learning avec les frameworks VueJS et Django Rest Framework
- Codage des tests unitaires du Backend et testes fonctionnelles du Frontend

Nouvelle Institution de Microfinance – Antananarivo

Développeur freelance | Octobre 2021 – Décembre 2021

Missions réalisées :

Conception et réalisation d'une application de gestion du Parc Automobile avec le framework PHP Laravel

Ecole Nationale d'Informatique - Fianarantsoa

Développeur | 1 Avril 2021 - 1 Juillet 2021

Missions réalisées :

- Conception et réalisation du site de l'Ecole ainsi que l'espace Enseignants – Etudiants – Anciens élèves avec Angular comme frontend et SailsJs pour backend
- Déploiement du site

NOMERIKIA – Antananarivo

Stagiaire | 2 Décembre 2019 – 28 Février 2020

Missions réalisées :

Conception et réalisation de la plateforme Nomerikia.mg – côté Front-Office avec le framework PHP CodeIgniter

Ministère de l'Economie et des Finances – Antananarivo

Stagiaire | 26 Novembre 2018 - 1 Février 2019

Missions réalisées :

Conception et participation à la réalisation d'un application web de Gestion des activités du Personnel

PRIX ET DISTINCTIONS

Lauréate ICT EVENT 2021 organisé au sein de l'ENI | Première place

Lauréate Women Focused Newbies Contest organisé par AlgoMada | Cinquième place

DIPLOMES ET FORMATIONS

Master II Professionnel (en cours)

Ecole Nationale d'Informatique – Parcours : Informatique Générale | 2022

Licence Professionnelle

Ecole Nationale d'Informatique – Parcours : Informatique Générale | 2020

Baccalauréat série C – Mention : Bien

Lycée RANOHAVIMANANA Norbert Ambatondrazaka | 2016

SOMMAIRE

CURRICULUM VITAE	I
SOMMAIRE	II
REMERCIEMENTS	V
LISTE DES TABLEAUX.....	VI
LISTE DES FIGURES.....	VII
LISTE DES ABREVIATIONS	XI
INTRODUCTION GENERALE.....	1
PARTIE I : PRESENTATIONS	
Chapitre 1 : Présentation de l'Ecole.....	2
1.1. Information d'ordre générale	2
1.2. Missions et historique	2
1.3. Organigramme institutionnel de l'ENI	4
1.4. Domaine de spécialisation	5
1.5. Architecture des formations pédagogiques.....	5
1.6. Relations de l'ENI avec les entreprises et les organismes.....	8
1.7. Partenariat au niveau international.....	9
1.8. Débouchés professionnels avec des diplômés	11
1.9. Ressources humaines	13
Chapitre 2 : Présentation de i-Team School	14
2.1. Présentation générale	14
2.2. Historique.....	14
2.3. Services offerts par l'entreprise	15
2.4. Organigramme institutionnel de l'entreprise	15
Chapitre 3 : Description du projet.....	17
3.1. Formulation.....	17

3.2. Objectif et besoins de l'utilisateur	17
3.3. Moyens nécessaires à la réalisation du projet	18
3.4. Résultats attendus.....	21
3.5. Chronogramme de travail	21

PARTIE II : ANALYSE ET CONCEPTION

Chapitre 4 : Analyse préalable	22
4.1. Analyse de l'existant.....	22
4.2. . Critique de l'existant.....	23
4.3. Conception avant-projet.....	24
4.4. Méthode et outils utilisés	25
Chapitre 5 : Analyse conceptuelle	33
5.1. Dictionnaire de données.....	33
5.2. Règles de gestion	41
5.3. Désignation des rôles de l'équipe SCRUM	46
5.4. Etapes de l'élaboration du Product Backlog.....	46
5.5. Product Backlog.....	51
5.6. Sprint Backlog	55
5.7. Modèle du domaine.....	57
5.8. Diagramme de classe de conception pour chaque sprint	59
5.9. Diagramme de classe de conception globale	62
Chapitre 6 : Conception détaillé	64
6.1. Architecture globale du système	64
6.2. Diagramme de séquence de conception.....	65
6.1. Diagramme de paquetage.....	75
6.2. Diagramme de déploiement	76

PARTIE III : REALISATION

Chapitre 7 : Mise en place de l'environnement	78
7.1. Installation et mise en place des outils.....	78
7.2. Architecture de l'application.....	84
Chapitre 8 : Développement de l'application	86
8.1. Création de la base de données	86
8.2. Codage de l'application	87
8.3. Présentation de l'application.....	106
CONCLUSION	118
REFERENCES BIBLIOGRAPHIQUES	XII
REFERENCES WEBOGRAPHIQUES.....	XIII
GLOSSAIRE	XIV
TABLE DES MATIERES	XVI
RESUME.....	XXI
ABSTRACT	XXI

REMERCIEMENTS

Au terme de la rédaction de ce mémoire, c'est un devoir agréable d'exprimer en quelques lignes la reconnaissance que je dois à tous ceux qui ont contribué de loin ou de près à l'élaboration de ce travail. Mes remerciements sont adressés à :

- Dieu tout puissant pour m'avoir donné le courage, la force et la patience d'achever ce modeste travail ;
- Monsieur HAJALALAINA Aimé Richard, Professeur, Président de l'Université de Fianarantsoa, pour le bon déroulement de l'année universitaire ;
- Monsieur MAHATODY Thomas, Professeur titulaire et Directeur de l'Ecole Nationale d'Informatique, qui assure le bon fonctionnement de nos études au sein de l'ENI ;
- Monsieur BAKARY Gaetan Jonathan, directeur et co-fondateur d'i-Team School pour le partage de connaissances et d'expériences dans le milieu informatique, tout en m'accordant sa confiance dans l'exécution de missions valorisantes durant ce stage ;
- Monsieur RATIARSON Venot, Maître de Conférences et mon encadreur pédagogique, pour ses précieux conseils, son encadrement et ses orientations sur la réalisation de ce travail ;
- Tous les membres du jury pour faire honneur de leur présence à cette représentation ;
- Tous les enseignants de l'ENI, pour la qualité des enseignements fournis tout au long de ma scolarité ;
- Mes parents et toute ma famille pour leur soutien matériel, financier, affectif et moral.

LISTE DES TABLEAUX

Tableau 1: Organisation du système de formation pédagogique de l'Ecole.....	5
Tableau 2: Architecture des études correspondant au système LMD	6
Tableau 3: Formations existantes à l'ENI.....	7
Tableau 4: Débouchés professionnels éventuels des diplômés	12
Tableau 5: Résumé du statut juridique de l'entreprise.....	14
Tableau 6: Moyens en ressources humaines pour la mise en œuvre du projet	19
Tableau 7: Moyens matériels pour la réalisation du projet	20
Tableau 8: Moyens logiciels utilisés pour la réalisation du projet	20
Tableau 9: Chronogramme de travail.....	21
Tableau 10: Avantages et inconvénients du contexte actuel	23
Tableau 11 : Avantages et inconvénients de chaque solution	24
Tableau 12 : Tableau comparatif entre l'approche traditionnelle et l'approche Agile	26
Tableau 13: Tableau comparatif entre les outils de modélisation.....	28
Tableau 14 : Tableau comparatif entre Vuejs et Angular	29
Tableau 15: Tableau comparatif entre Nodejs et Django Rest Framework	30
Tableau 16 : Tableau comparatif entre Visual Studio Code et Sublime Text.....	31
Tableau 17: Comparaison entre MariaDB, MySQL et PostgreSQL	32
Tableau 18: Dictionnaire de données	33
Tableau 19: Nom des tables et leur signification	42
Tableau 20: Liste des acteurs	47
Tableau 21: Détails concernant les principaux thèmes	48
Tableau 22: Product Backlog du projet.....	52
Tableau 23: Type d'association et leur illustration.....	59
Tableau 24: Composants de base d'un diagramme de paquetage.....	75
Tableau 25: Composant de base d'un diagramme de déploiement.....	76

LISTE DES FIGURES

Figure 1: Organigramme de l'Ecole Nationale d'Informatique	4
Figure 2 : Organigramme de l'entreprise iTS	16
Figure 3 : Diagramme de flux de la méthode d'éducation utilisé par la plupart des établissements	22
Figure 4: Fonctionnement de la méthode SCRUM	27
Figure 5: Découpage d'un projet en sprint et en release.....	50
Figure 6: Construction du sprint backlog à partir du product backlog.....	55
Figure 7: Présentation du travail à faire	56
Figure 8: Modèle de domaine de la plateforme.....	58
Figure 9: Illustration d'une classe.....	59
Figure 10: Exemple général d'un diagramme de classe	60
Figure 11: Diagramme de classe de conception du SPRINT 1	60
Figure 12: Diagramme de classe de conception du SPRINT 2	61
Figure 13: Diagramme de classe de conception du SPRINT 3	61
Figure 14: Diagramme de classe de conception du SPRINT 5	61
Figure 15: Diagramme de classe de conception du SPRINT 6	62
Figure 16: Diagramme de classe de conception du SPRINT 7	62
Figure 17: Diagramme de classe de conception globale	63
Figure 18: Architecture client/serveur.....	64
Figure 19: Formalisme d'un diagramme de séquence	65
Figure 20: Diagramme de séquence de conception pour l'authentification	66
Figure 21: Diagramme de séquence sur la création d'un cours	67
Figure 22: Diagramme de séquence sur la création d'une partie	67
Figure 23: Diagramme de séquence de création d'un contenu	68
Figure 24: Diagramme de séquence pour la consultation de cours.....	69

Figure 25: Diagramme de séquence pour la gestion de l'interface utilisateur.....	70
Figure 26: Diagramme de séquence pour la gestion des catégories.....	70
Figure 27: Diagramme de conception pour la gestion d'un niveau	71
Figure 28: Diagramme de séquence de conception de la gestion des parcours	71
Figure 29: Diagramme de séquence de la gestion des types de cours.....	72
Figure 30: Diagramme de séquence de conception pour la gestion des utilisateurs	72
Figure 31: Diagramme de séquence de conception pour la modification du profil	73
Figure 32: Diagramme de séquence de conception pour la rédaction d'un quizz	74
Figure 33: Diagramme de séquence de conception pour l'effectuation d'un quizz	75
Figure 34: Diagramme de paquetage du projet	76
Figure 35: Diagramme de déploiement de la plateforme	77
Figure 36: Page d'accueil pour l'installation de Visual Paradigm	78
Figure 37: Fin de l'installation de Visual Paradigm	79
Figure 38: Début de l'installation de Visual Studio Code	79
Figure 39: Fin de l'installation de Visual Studio Code.....	80
Figure 40: Vérification de la version de Git installée	80
Figure 41: Structure du dossier Frontend	81
Figure 42: Arborescence du dossier du backend.....	83
Figure 43: Interface du logiciel HeidiSQL.....	84
Figure 44: Architecture de la plateforme	85
Figure 45: Extrait de la base de données	86
Figure 46: Contenu du dossier its.....	87
Figure 47: Extrait de code de settings.py	88
Figure 48: Extrait de code de urls.py	89
Figure 49: Contenu du dossier restapi.....	89
Figure 50: Extrait de code de models.py	90

Figure 51: Extrait de codes de serializers.py.....	91
Figure 52: Extrait de code de views.py	92
Figure 53: Extrait de code du fichier test_models.py.....	93
Figure 54: Extrait de code de test_views.py	94
Figure 55: Contenu de package.json	95
Figure 56: Extrait de code de create.js	96
Figure 57: Extrait de code de read.js.....	97
Figure 58: Extrait de codes de update.js	98
Figure 59: Extrait de delete.js	99
Figure 60: Extrait de code de router/index.js	100
Figure 61: Contenu de src/views/App.vue	101
Figure 62: Structure du dossier src/views	102
Figure 63: Extrait de code de CoursesListsPage.vue	103
Figure 64: Extrait de code de ProfileView.vue	104
Figure 65: Extrait de code de end_to_end_tests.py.....	105
Figure 66: Page de connexion	106
Figure 67: Modification du profil.....	107
Figure 68: Tableau de bord	107
Figure 69: Page d'administration de la plateforme - Partie 1	108
Figure 70: Page d'administration de la plateforme - Partie 2	108
Figure 71: Page dédiée à la gestion des enseignants	109
Figure 72: Page de création d'un enseignant.	109
Figure 73: Page dédiée à la gestion des étudiants	110
Figure 74: Page dédié à la création d'annonce	110
Figure 75: Création d'un cours	111
Figure 76: Un cours créé avec ses parties	111

Figure 77: Création d'un quizz	112
Figure 78: Consultation des notifications adressées à l'enseignant.....	112
Figure 79: Consultation des notifications adressées à l'étudiant.....	113
Figure 80: Consultation d'un cours - Partie 1	113
Figure 81: Consultation d'un cours - Partie 2.....	114
Figure 82: Section pour rendre un devoir.....	114
Figure 83: Page pour traiter un quizz dont les questions s'affichent d'un coup	115
Figure 84: Page pour traiter un quizz dont les questions s'affichent un à un	115
Figure 85: Premières interactions avec le Bot Messenger	116
Figure 86: Consultation d'un cours.....	117
Figure 87: Consultation des supports pédagogiques	117

LISTE DES ABREVIATIONS

API: Application Programming Interface

CLI : Command-Line Interface

DRF : Django Rest Framework

ENI : Ecole Nationale d'Informatique

iTS: i-Team School

JIRAMA: Jiro sy Rano Malagasy

LMD : Licence-Master-Doctorat

MAEP : Mécanisme Africain d'Evaluation par les Pairs

MEF : Metro Ethernet Forum 8

MEN : Ministère de l'Éducation Nationale

MFB : Ministère des Finances et du Budget

MIC : Madagascar Informatique Corporation

PRESUP : Programme de Renforcement en l'Enseignement Supérieur

RG : Règle de Gestion

SGBD : Système de Gestion de Base de Données

SQL : Structured Query Language

TIC : Technologies de l'Information et de la Communication

INTRODUCTION GENERALE

L'essor de l'apprentissage en ligne a constitué un changement majeur dans l'éducation au cours des deux dernières décennies. Avec l'avènement de l'internet et du haut débit, l'apprentissage en ligne est devenu une option plus viable et plus abordable pour les instituts. Selon le Corporate Digital Learning, le secteur de l'apprentissage virtuel a connu une croissance considérable de 900 % au cours des 22 dernières années. Cette évolution a permis aux étudiants d'apprendre de manière plus flexible et personnalisée, tout en offrant aux éducateurs de nouvelles façons d'impliquer les étudiants.

A Madagascar, de plus en plus d'établissements intègrent ce concept. En effet, c'est un choix judicieux pour les instituts de formations d'adhérer à l'éducation en ligne car elle permet à leurs étudiants d'accéder à une éducation de qualité quand et où ils veulent. D'ailleurs, la pandémie du Covid-19 a mis en exergue son efficacité quand certains établissements qui appliquaient l'apprentissage en ligne ont maintenu leur programme pédagogique tandis que d'autres ont dû prendre une pause de plusieurs mois.

Ce projet consiste à la création et la mise en place d'une plateforme e-Learning moins chère et facilement accessible pour les étudiants. Ainsi, la plateforme offre la possibilité aux lycées, universités ou même les centres de formations des échanges d'informations entre enseignants et étudiants d'une façon plus abordable.

Du point de vue méthodologique, pour élaborer ce projet, nous allons utiliser des langages de programmation, un langage de modélisation, une méthode de gestion de projet, ainsi qu'un système de gestion de base de données. Le résultat attendu de ce projet est la mise en place d'une plateforme d'apprentissage en ligne avec des interactions entre pédagogues, apprenants et ressources pédagogiques. Ainsi, la plateforme web se doit d'être intuitive, sécurisé et doit disposer des cours, des quizz, des tests et des résultats.

Ce document a pour objectif de présenter le travail réalisé lors du stage au sein de l'entreprise i-Team School. Pour cela, nous commencerons par les diverses présentations à savoir celle de notre Ecole puis de l'entreprise i-Team School qui a bien voulu nous accueillir. Par la suite, nous passerons par la description du projet dans ces différentes facettes puis nous expliquerons les différentes analyses faites pour aboutir à la réalisation de cette plateforme.

Chapitre 1 : Présentation de l'Ecole

1.1. 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. Son adresse électronique est eni@eni.mg et son site web : www.eni.mg.

1.2. Missions et historique

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 école supérieure peut être considérée aujourd'hui comme la vitrine et la pépinière des élites informaticiennes du pays. L'Ecole Nationale d'Informatique s'est constituée de façon progressive au sein du Centre Universitaire Régional (CUR) de Fianarantsoa.

De façon formelle, l'ENI était constituée et créée au sein du (CUR) par le décret N° 83185 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 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).

L'implantation de cette Ecole Supérieure de technologie de pointe dans un pays en développement et dans une Province (ou Faritany) à tissu économique et industriel faiblement développé ne l'a pourtant pas défavorisée, ni empêchée de former des spécialistes informaticiens de bon niveau, qui sont recherchés par les entreprises, les sociétés et les organismes publics et privés sur le marché de l'emploi.

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.

Dans le cadre du Programme de renforcement en l'Enseignement Supérieur (PRESUP), la filière de formation des Techniciens Supérieurs en Maintenance des Systèmes des informatiques a été mise en place en 1986 grâce à l'appui matériel et financier de la Mission Française de coopération auprès de l'Ambassade de France à Madagascar.

Une formation pour l'obtention de la certification CCNA et / ou NETWORK+ appelée « CISCO Networking Academy » a été créée à l'Ecole en 2002-2003 grâce au partenariat avec CISCO SYSTEM et l'Ecole Supérieure Polytechnique d'Antananarivo. Cependant, cette formation n'avait pas duré longtemps.

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. 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 options suivantes 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). Mais la filière de formation des Techniciens Supérieurs en Maintenance des Systèmes Informatiques a été gelée en 2009.

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'à l'Université de Toliara.

1.3. Organigramme institutionnel de l'ENI

L'organigramme de l'Ecole Nationale d'Informatique est inspiré des dispositions du décret N° 83-185 du 23 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 conseil d'école constitue l'organe délibérant de l'Ecole. Elle est dirigée par un directeur nommé par un décret adopté en conseil des Ministres. Sous la tutelle de la direction, on retrouve le collège des enseignants, le service pédagogique, la scolarité et le service administratif et financier.

La figure 1 présente l'organigramme actuel de l'Ecole[a].

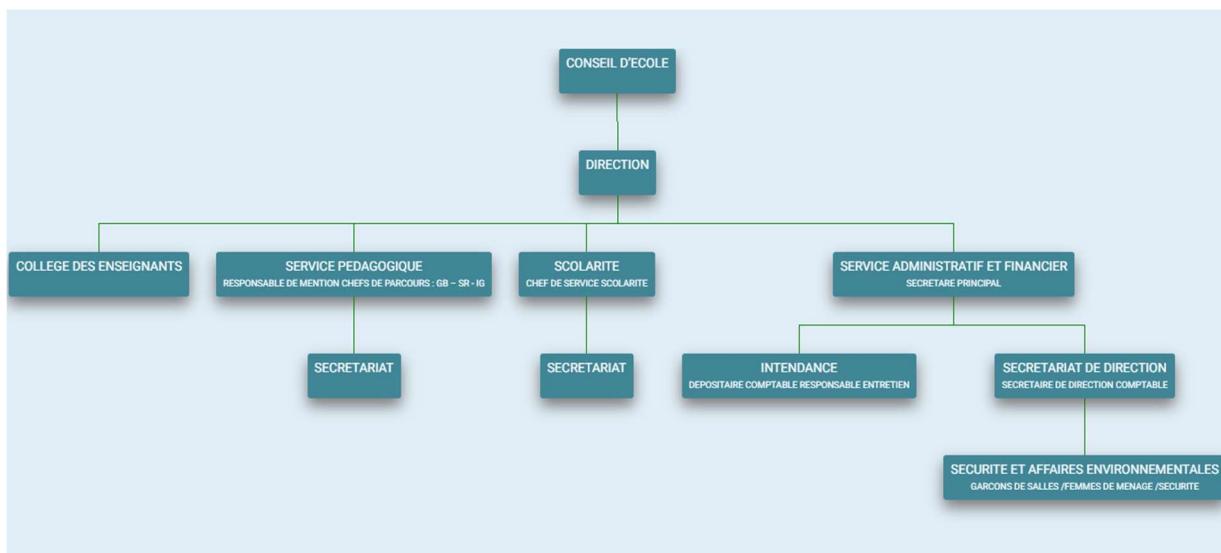


Figure 1: Organigramme de l'Ecole Nationale d'Informatique

Sur cet organigramme, l'Ecole placée sous la tutelle académique et administrative de l'Université de Fianarantsoa, et dirigée par un Directeur élu par les Enseignants – Chercheurs permanents de l'Etablissement et nommé par un décret pris en Conseil des ministres pour un mandat de 3 ans.

1.4. 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 ;

D'une manière plus générale, les programmes des formations sont basés sur l'informatique de gestion et sur l'informatique des Systèmes et Réseaux. Et les modules de formation intègrent aussi bien des éléments d'Informatique fondamentale que des éléments d'Informatique appliquée.

Le tableau 1 décrit l'organisation du système de formation pédagogique de l'Ecole Nationale d'Informatique.

Tableau 1: Organisation du système de formation pédagogique de l'Ecole

Formation théorique	Formation pratique
<ul style="list-style-type: none">- Enseignement théorique- Travaux dirigés- Travaux pratiques	<ul style="list-style-type: none">- Etude de cas- Travaux de réalisation- Projets / Projets tutorés- Voyage d'études- Stages

1.5. 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) auprès du Ministères de l'Enseignement Supérieur et de la Recherche Scientifique selon les dispositions de l'Arrêté N°31.174/2012MENS en date du 05 Décembre 2012.

Au sein de l'ENI, il existe une seule mention (INFORMATIQUE) et trois parcours :

- Génie logiciel et Base de Données ;
- Administration des Systèmes et Réseaux ;
- Informatique Générale.

L'architecture des études à trois niveaux conforment au système LMD permet les comparaisons et les équivalences académiques des diplômes au niveau international.

- L = Licence (Bac + 3) = L1, L2, L3 = 6 semestres S1 à S6;
- M = Master (Bac + 5) = M1, M2 = 4 semestres S7 à S10.

Le diplôme de licence est obtenu en 3 années des études après Baccalauréat. Et le diplôme de Master est obtenu en 2 ans après obtention du diplôme de LICENCE.

- Le MASTER PROFESSIONNEL est un diplôme destiné à la recherche d'emploi au terme des études ;
- Le MASTER RECHERCHE est un diplôme qui remplace l'ancien Diplôme d'Etudes Approfondies (DEA), et qui permet de s'inscrire directement dans une Ecole Doctorale au terme des études ;
- D = Doctorat (Bac +8) Le Doctorat est un diplôme qu'on peut obtenir en 3 ans après l'obtention du diplôme de MASTER RECHERCHE.

Le tableau 2 présente l'architecture des études correspondant au système LMD.

Tableau 2: Architecture des études correspondant au système LMD



Le tableau 3 liste les formations existantes à l'ENI.

Tableau 3: Formations existantes à l'ENI

FORMATION EN		
	LICENCE PROFESSIONNELLE ET HYBRIDE	MASTER
Condition d'admission	Par voie de concours Formation Professionnelle : 100 candidats Formation hybride : 150 candidats	
Condition d'accès	Bac de série C, D ou Technique	Être titulaire de licence professionnelle
Durée de formation	3 années	2 années
Diplôme délivrer	Diplôme de Licence Professionnelle en Informatique	Diplôme de Master Professionnel Diplôme de Master 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 Master Recherche permet à son titulaire de poursuivre directement des études en doctorat et de s'inscrire directement dans une Ecole Doctorale.

Les Ecoles Doctorales jouissent d'une autonomie de gestion par rapport aux Etablissements de formation universitaire.

Il convient de signaler que par arrêté ministériel N° 21.626/2012 – MESupRES publié le 9 Août 2012 par la Commission National d'habilitation (CNH), l'Ecole Doctorale «Modélisation – Informatique » a été habilitée pour l'Université de Fianarantsoa.

Depuis l'année universitaire 2010 - 2011, l'Ecole Nationale d'Informatique s'est mise à organiser des formations hybrides en informatique dans les différentes régions (Fianarantsoa, Toliara) en raison de l'insuffisance de la capacité d'accueil des infrastructures logistiques. En effet, le système de formation hybride semi-présentielle utilise la visioconférence pour la formation à distance.

Bien qu'il n'existe pas encore au niveau international de reconnaissance écrite et formelle des diplômes délivrés par l'ENI, les étudiants diplômés de l'Ecole sont plutôt bien accueillis dans les instituts universitaires étrangères (CANADA, Suisse, France...).

1.6. Relations de l'ENI avec les entreprises et les organismes

Les stages effectués chaque année par les étudiants mettent l'Ecole en rapport permanent avec plus de 300 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 techniciens supérieurs de niveau LICENCE et des ingénieurs de type généraliste 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.

Les principaux débouchés professionnels des diplômés de l'Ecole concernent les domaines suivants :

- Les technologies de l'information et de la communication (TIC)
- La sécurité informatique des réseaux
- L'administration des réseaux et des systèmes
- Les services bancaires et financiers, notamment le Mobile Banking
- Les télécommunications et la téléphonie mobile
- Les Big Data
- Le commerce, la vente et l'achat, le Marketing
- L'ingénierie informatique appliquée
- L'écologie et le développement durable

Parmi les sociétés, entreprises et organismes partenaires de l'Ecole, on peut citer : ACCENTURE Mauritius, Air Madagascar, Ambre Associates, Airtel, Agence Universitaire de la Francophonie (AUF), B2B, Banque Centrale, BFG-SG, BIANCO, BLUELINES, CNAPS, Bureau National de Gestion des Risques et des Catastrophes (BNGRC), CEDII-Fianarantsoa, Data Consulting, Central Test, Centre National Antiacridien, CNRE, CHU, CNRIT, COLAS,

Direction Générale des Douanes, DLC, DTS/Moov, FID, FTM, GNOSYS, IBERIA, INGENOSIA, INSTAT, IOGA, JIRAMA, JOUVE, MADADEV, MAEP, MEF, MEN, MESupRES, MFB, MIC, MNINTER, Min des postes/Télécommunications et du Développement Numérique, NEOV MAD, Ny Havana, Madagascar National Parks, OMNITEC, ORANGE, OTME, PRACCESS, QMM Fort-Dauphin, SMMC, SNEDADRS Antsirabe, Sénat, Société d'Exploitation du Port de Toamasina (SEPT), SOFTWELL, Strategy Consulting, TELMA, VIVETEC, Société LAZAN'I BETSILEO, WWF ...

L'organisation de stage en entreprise continue non seulement à renforcer la professionnalisation des formations dispensées, mais elle continue surtout à accroître de façon exceptionnelle les opportunités d'embauche pour les diplômés de l'Ecole.

1.7. Partenariat au niveau international

Entre 1996 et 1999, l'ENI avait bénéficié de l'assistance technique et financière de la Mission Française de Coopération et d'action culturelle dans le cadre du Programme de Renforcement de l'Enseignement Supérieur (PRESUP) consacré à l'Ecole a notamment porté sur :

- Une dotation en logiciels, micro-ordinateurs, équipements de laboratoire de maintenance et de matériels didactiques
- La réactualisation des programmes de formation assortie du renouvellement du fonds de la bibliothèque
- L'appui à la formation des formateurs
- L'affectation à l'Ecole d'Assistants techniques français

De 2000 à 2004, l'ENI avait fait partie des membres du bureau de la Conférence Internationale des Ecoles de formation d'Ingénieurs et Technicien d'Expression Française (CITEF).

Les Enseignants-Chercheurs de l'Ecole participent régulièrement aux activités organisées dans le cadre du Colloque Africain sur la Recherche en Informatique (CARI). L'ENI avait également signé un accord de coopération interuniversitaire avec l'Institut de Recherche en Mathématiques et Informatique Appliquées (IREMIA) de l'Université de la Réunion, l'Université de Rennes 1, l'INSA de Rennes, l'Institut National Polytechnique de Grenoble (INPG).

A partir du mois de Juillet 2001, l'ENI avait abrité le Centre de Réseau Opérationnel (Network Operating Center) du point d'accès à Internet de l'Ecole ainsi que de l'Université de Fianarantsoa. Grâce à ce projet américain qui a été financé par l'USAID Madagascar, l'ENI de l'Université de Fianarantsoa avait été dotées d'une ligne spécialisée d'accès permanent au réseau Internet.

L'ENI avait de même noué des relations de coopération avec l'Institut de Recherche pour le Développement (IRD). L'objet du projet de coopération avait porté sur la modélisation environnementale du Corridor forestier de Fandriana jusqu'à Vondrozo. Dans ce cadre, un atelier scientifique international avait été organisé à l'ENI en Septembre 2008. Cet atelier scientifique avait eu pour thème de modélisation des paysages.

Et dans le cadre du programme scientifique PARRUR, l'IRD avait financé depuis 2010 le projet intitulé « Forêts, Parcs et Pauvreté dans le Sud de Madagascar (FPPSM). Des étudiants en DEA et des Doctorants issus de l'ENI avaient participé à ce Programme.

Par ailleurs, depuis toujours la même année 2010, l'ENI de Fianarantsoa avait été sélectionnée pour faire partie des organismes partenaires de l'Université de Savoie dans le cadre du projet TICEVAL relatif à la certification des compétences en TIC. Le projet TICEVAL avait été financé par le Fonds Francophone des Inforoutes pour la période allant de 2010 à 2012, et il avait eu pour objectif de généraliser la certification des compétences en Informatique et Internet du type C2i2e et C2imi.

Dans le cadre du projet TICEVAL, une convention de coopération avec l'Université de Savoie avait été signée par les deux parties concernées. La mise en œuvre de la Convention de Coopération avait permis d'envoyer des étudiants de l'ENI à Chambéry pour poursuivre des études supérieures en Informatique.

Enfin et non des moindres, l'Ecole Nationale d'Informatique avait signé en Septembre 2009 un protocole de collaboration scientifique avec l'ESIROI – STIM de l'Université de la Réunion.

Comme l'ENI constitue une pépinière incubatrice de technologie de pointe, d'emplois et d'entreprises, elle peut très bien servir d'instrument efficace pour renforcer la croissance économique du pays, et pour lutter contre la Pauvreté.

De même que le statut de l'Ecole devrait permettre de renforcer la position concurrentielle de la Grande Ile sir l'orbite de la modélisation grâce au développement des nouvelles technologies.

1.8. Débouchés professionnels avec des diplômés

Le chômage des jeunes diplômés universitaires fait partie des maux qui gangrènent Madagascar. L'environnement socio-politique du pays depuis 2008 jusqu'à ce jour a fait que le chômage des diplômés est devenu massif par rapport aux établissements de formation supérieure existants. Cependant, 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 Nationale d'Informatique 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 34 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. Par conséquent, en raison de fait que l'équipe pédagogique de l'Ecole est expérimentée, les enseignants-chercheurs et les autres formateurs de l'Ecole sont dotés d'une grande expérience dans l'enseignement et dans le milieu professionnel.

L'Ecole est fière de collaborer de façon régulière avec un nombre croissant d'entreprises, de sociétés et d'organismes publics et privés à travers les stages des étudiants.

Les formations dispensées à l'Ecole sont ainsi orientées vers le besoin et les attentes des entreprises et des sociétés.

L'Ecole fournit à ses étudiants de niveau LICENCE et MASTER des compétences professionnelles et métiers indispensables pour les intégrer sur le marché du travail.

L'Ecole s'efforce de proposer à ses étudiants une double compétence à la fois technologique et managériale combinant l'informatique de gestion ainsi que l'administration des réseaux et systèmes.

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 professionnels éventuels des diplômés.

Tableau 4: Débouchés professionnels éventuels des diplômés

LICENCE	<ul style="list-style-type: none"> - 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 cybercafé
MASTER	<ul style="list-style-type: none"> - Administrateur de cybercafé - Administrateur de réseau et système - Architecture de système d'information - Développeur d'application /web /java/Python/ Android / IOS - Ingénieur réseau - Webmaster /web designer - Concepteur Réalisateur d'applications - Directeur du système de formation - Directeur de projet informatique - Chef de projet informatique - Responsable de sécurité informatique - Consultant fonctionnel ou freelance

1.9. Ressources humaines

- Directeur de l’Ecole : Monsieur MAHATODY Thomas
- Responsable de Mention : Monsieur RABETAFIKA Louis Haja
- Responsable de Parcours « Génie Logiciel et Base de Données » : Monsieur RALAIVAO Jean Christian
- Responsable de Parcours « Administration Systèmes et Réseaux » : Monsieur SIAKA
- Responsable de Parcours « Informatique Générale » : Monsieur Gilante GESAZAFY
- Nombre d’Enseignants permanents : 12 dont un (01) Professeurs Titulaire, deux (02) Professeurs, cinq (05) Maîtres de Conférences et quatre (04) Assistants d’Enseignement Supérieur et de Recherche
- Nombre d’Enseignants vacataires : 10
- Personnel Administratif : 23

Chapitre 2 : Présentation de i-Team School

2.1. Présentation générale

i-Team School , abrégé en iTS, est une startup créée officiellement en Avril 2022, sous la forme juridique SARLU, qui promeut un projet d'innovation technologique visant l'éducation à Madagascar pour que les plateformes e-learning puissent être plus proche et accessible à tous les étudiants.

Le tableau 5 présente le résumé du statut juridique de l'entreprise.

Tableau 5: Résumé du statut juridique de l'entreprise

Raison Sociale	i-Team School
Statut juridique	Société Anonyme à Responsabilité Limité Unipersonnelle (SARLU)
Siège	Immeuble JACARANDA 1 ^{ère} étage, bureau N°1 Ambatonakanga, Antananarivo
Téléphone	034 61 780 78
Email	contact@its.mg
NIF	4011741869
STAT	62012112022010392

2.2. Historique

Partie d'une idée de lier Moodle et un bot messenger, iTS a été élu 2^{ème} meilleur projet de POESAM 2021 en gagnant une incubation à Orange Fab du département Orange Digital Center.

Vers la fin de l'année 2021, la startup est incubée dans un accélérateur de startup « Zafitody » sis à Ampefiloha.

Aujourd'hui, iTS se place comme une plateforme d'accès pour les apprenants à des plateformes e-Learning comme Moodle, Google Classroom, et même sur une plateforme personnalisée.

2.3. Activités de l'entreprise

L'entreprise iTS est une entreprise qui souhaite contribuer efficacement à la mise en œuvre de l'enseignement en ligne.

L'objectif est d'offrir aux enseignants et aux étudiants une expérience positive et réussie. Pour cela, l'entreprise souhaite participer en améliorant le système d'éducation mise en place par l'institut de formation ou le remplacer si cela est nécessaire.

Les établissements cibles sont tous les instituts de formations, à savoir, les écoles, les universités, les centres de formations, ...

Les activités de l'entreprise se concentre sur l'offre de trois services : la mise en place d'une plateforme e-Learning, la mise en place d'un bot messenger, et accompagnement personnalisé et formation sur l'utilisation de la plateforme.

2.3.1. Mise en place d'une plateforme e-Learning

i-Team School déploie et s'occupe de la mise en ligne, la configuration de Moodle pour les instituts qui n'utilisent pas encore de plateforme e-learning. En outre, elle assure la sécurité des données.

2.3.2. Mise en place d'un bot messenger

Dans le cas où l'institut client possède déjà une plateforme d'enseignement en ligne, l'entreprise met en place pour chaque page par institut le bot messenger qui est lié à leur plateforme déjà existante.

2.3.3. Accompagnement personnalisé et formation sur l'utilisation de la plateforme

L'entreprise iTS peut offrir différents accompagnements pour tout le processus de la mise en place, de la formation aux insertions des contenus dans la plateforme.

2.4. Organigramme institutionnel de l'entreprise

La figure 2 montre l'organigramme institutionnel de l'entreprise iTS

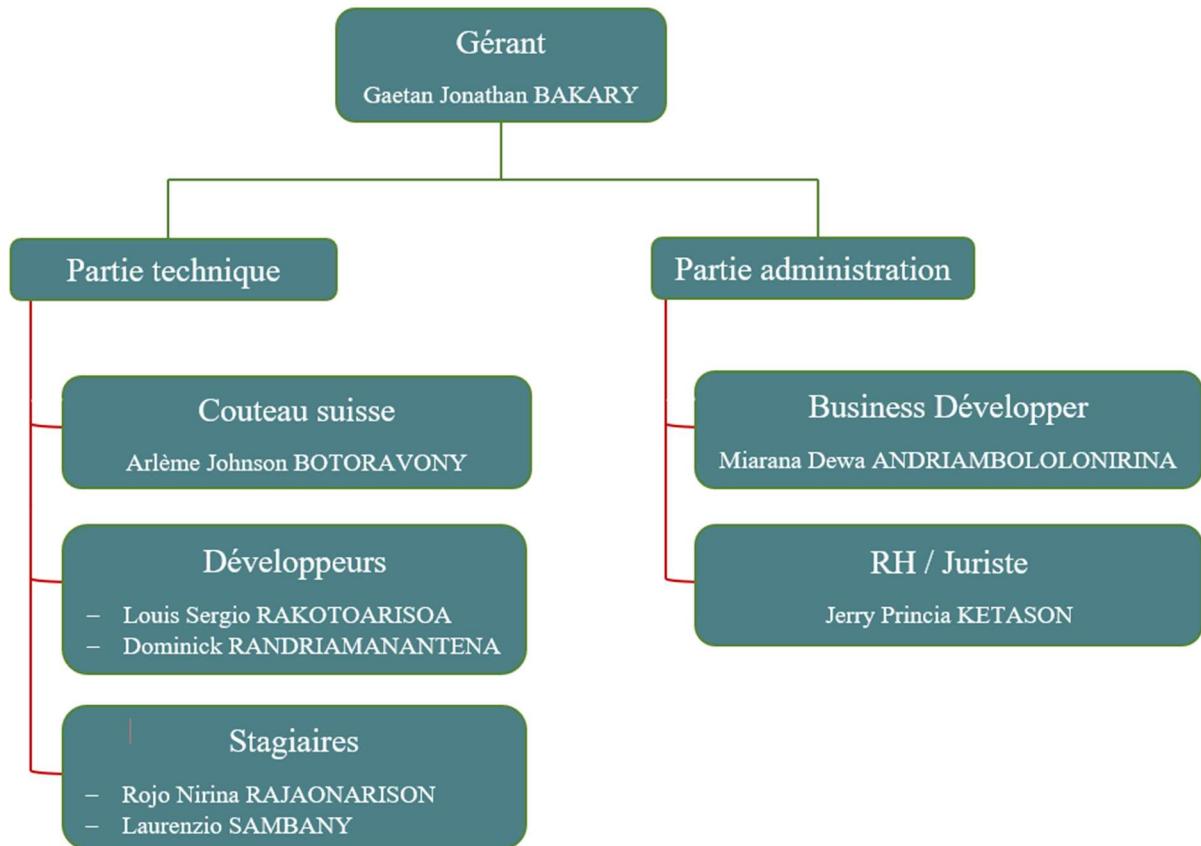


Figure 2 : Organigramme de l'entreprise iTS

Chapitre 3 : Description du projet

3.1. Formulation

Le temps où l'éducation se faisait uniquement en présentiel a révolu. A l'ère de l'internet, il est astucieux d'entamer dans la nouvelle norme qu'est l'enseignement en ligne. L'enseignement en ligne n'est pas forcément là pour écraser les cours en présentiel, mais également pour les améliorer, les faciliter et les rendre plus accessibles aux parties prenantes. Ainsi, nous nous intéressons à un projet qui prendra en compte le système déjà mis en place par les établissements et qui apportera sa grande contribution en optimisant l'accès pour tous et les procédures d'études.

Ce projet se nomme « iTS.mg ». Elle consiste à mettre en place une plateforme d'enseignement en ligne pour les établissements dans le but de permettre les échanges d'outils pédagogiques entre enseignants et étudiants.

3.2. Objectif et besoins de l'utilisateur

3.2.1. Objectif

L'objectif de ce projet est la « Conception et mise en place d'une plateforme e-Learning ». Il offre une solution numérique pour accompagner les instituts de formation dans l'établissement d'une plateforme d'apprentissage en ligne conforme à ses besoins, à la portée de main des étudiants où et quand ils le souhaitent, et plus abordable en matière de connexion.

3.2.2. Besoins de l'utilisateur

Les utilisateurs de la plateforme sont les administrateurs de la plateforme, les enseignants et les étudiants. Ils escomptent une page dédiée à la modification de leur profil.

Spécifiquement, les administrateurs s'attendent à :

- la création et la gestion des classes,
- la liste de tous les cours ,
- la création et la gestion des utilisateurs,
- la création et la gestion des annonces,
- la page d'administration des différents configurations de la plateforme.

Les enseignants comptent sur la plateforme pour :

- la liste de leur cours,
- la création et la gestion des cours qui comprend également:
 - la création et la gestion des parties intégrante d'un cours,
 - la création et la gestion des contenus de chaque partie,
 - la création de quizz pour les étudiants,
 - la consultation des comptes rendu du quizz,
 - la création et l'organisation de testes,
 - l'écrit d'un devoir,
 - la consultation des rapports d'un devoir,
- la création et la gestion des annonces.

Quant aux étudiants, ils aspirent à :

- la liste de leurs cours,
- la page pour consulter un cours qui comprend également:
 - la page pour rendre leur devoir,
 - la page pour récupérer les différents supports pédagogiques,
 - la page pour effectuer des quizz,
- la page pour consulter les annonces qui leurs sont adressées.

3.3. Moyens nécessaires à la réalisation du projet

Il est fondamental d'établir la liste des moyens nécessaires à la réalisation du projet si nous souhaitons la réussite de celle – ci.

3.3.1. Moyens humains

Les moyens humains fournissent les compétences et les connaissances nécessaires à l'exécution des tâches requises pour instaurer la plateforme. En outre, ils contribuent à garantir le bon fonctionnement et l'efficacité de la plateforme.

Dans notre cas, les moyens humains régissent aux règles d'une équipe agile composé de Product owner, de Scrum Master et d'une équipe de développement. Cependant, l'attribution des rôles s'ajuste à l'équipe existante.

Le tableau 6 représente les moyens en ressources humaines pour la création et la mise en place de la plateforme.

Tableau 6: Moyens en ressources humaines pour la mise en œuvre du projet

Poste	Effectif	Tâches
Chef de projet, également Product owner	1	<ul style="list-style-type: none"> – Attribution des tâches tirés des UserStories – Validation des résultats
Scrum Master	1	<ul style="list-style-type: none"> – Accompagnement de l'équipe suivant la méthodologie SCRUM – Travailler avec le product owner
Développeurs	2	<p>L'un : - Développement du Backend</p> <p>- Maintenance applicative serveur</p> <p>- Mise en place de la pipeline CI/CD</p> <p>L'autre : - Développement du bot</p>
Stagiaires développeurs	2, y compris moi – même	<ul style="list-style-type: none"> – Développement des tests unitaires côté Backend – Participation au développement du Backend – Développement du Frontend – Développement des tests de bout en bout côté Frontend – Rédaction de la documentation technique

3.3.2. Moyens matériels

Il s'agit de l'ensemble des matériels nécessaires employés pour mener à terme le projet, qu'ils soient d'usage commun ou individuel.

Le tableau 7 résume les matériels nécessaires utilisés pour la création et la mise en place de la plateforme.

Tableau 7: Moyens matériels pour la réalisation du projet

Désignation	OS	Utilisation
Cinq PC	Ubuntu 20.04.1 LTS sur WSL2 ou directement Ubuntu 20.04.1	<ul style="list-style-type: none"> – Développement des tests – Développement de la plateforme – Développement du bot – Développement du NLP
Serveur	Ubuntu 20.04.1 LTS	Serveur de préproduction pour héberger les résultats du développement continu
Serveur AWS	Ubuntu 20	Serveur de production pour héberger la plateforme

3.3.3. Moyens logiciels

Les moyens logiciels sont les outils employés conforme aux besoins de chaque membre de l'équipe en vue de mieux organiser le travail.

Le tableau 8 présente les moyens logiciels pour la réalisation du projet.

Tableau 8: Moyens logiciels utilisés pour la réalisation du projet

Désignation	Utilisations
Visual Paradigm	Outil de conception avec le langage de modélisation UML
Visual Studio Code	Outil de développement
HeidiSQL	Outil d'administration de la base de données
Figma	Conception des maquettes et designs
Messenger, Gmail	Outils de communication et de collaboration
Plateforme Github	<ul style="list-style-type: none"> – Suivie du projet et assignement des tâches avec Workflow – Hébergement de code pour le contrôle de version – Etablissement des backlog de projet avec Github Project – Automatisation des pipelines CI/CD avec Github Actions
AWS	Serveur d'hébergement de la production

3.4. Résultats attendus

A l'issue du travail, nous nous attendons à une plateforme :

- intuitive,
- performante,
- ergonomique,
- rapide dans les traitements,
- sécurisée,
- accessible via un bot messenger.

3.5. Chronogramme de travail

Le stage ayant débuté le 1^{er} juin 2022, le tableau 9 représente le chronogramme des travaux.

Tableau 9: Chronogramme de travail

Mois	Juin	Juillet	Aout	Septembre	Octobre	Novembre
Conception globale de la base de données et de l'application						
Création et mise en place de l'application						
Rédaction du mémoire de Master						

Chapitre 4 : Analyse préalable

L'analyse préalable est la première phase réelle du cycle de vie d'un projet. Elle concerne la première formalisation du besoin et de sa solution.

4.1. Analyse de l'existant

4.1.1. Organisation actuelle

Il existe aujourd'hui de nombreux types d'enseignement, mais l'une des formes les plus traditionnelles reste l'enseignement dans une salle de classe. Dans ce type d'enseignement, les étudiants sont généralement encadrés par un professeur dans un environnement en face à face et ils apprennent par le biais de cours magistraux, de discussions et de devoirs. Néanmoins, l'ère de l'internet a révolutionné de nombreux établissements scolaires et universitaires qui introduisent à présent à l'éducation en ligne via des plateformes comme Moodle. L'accès aux cours a alors connu une grande évolution, désormais, l'éducation peut se faire soit en présentiel, soit en ligne sur des plateformes accessibles dès lors qu'on a des datas.

La figure 3 expose le diagramme représentant la méthode d'éducation employé par de nombreux instituts de formation.

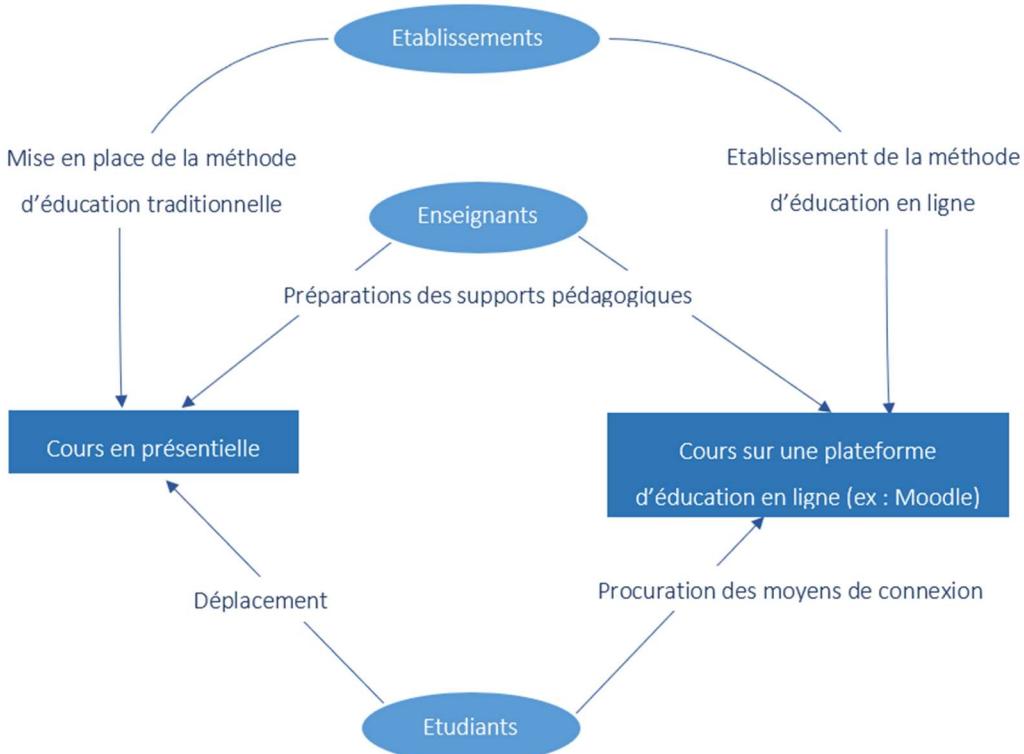


Figure 3 : Diagramme de flux de la méthode d'éducation utilisé par la plupart des établissements

4.1.1. Inventaires logiciels et matériels

En amont du projet, l'entreprise iTS dispose d'ordinateurs portables ainsi qu'un serveur.

4.2. . Critique de l'existant

Le tableau 10 résume les points forts et points faibles du contexte actuel.

Tableau 10: Avantages et inconvénients du contexte actuel

Points forts	Points faibles
<p><u>Enseignement traditionnel :</u></p> <ul style="list-style-type: none">- Les étudiants reçoivent une attention personnelle de la part de leurs enseignants.- L'environnement d'apprentissage est plus structuré, ce qui peut être bénéfique pour les étudiants qui ont besoin de ce type d'environnement pour réussir. <p><u>Enseignement en ligne mise en place progressivement :</u></p> <ul style="list-style-type: none">- Les étudiants ont la possibilité d'apprendre à son propre rythme, la commodité de pouvoir étudier de n'importe où et la variété des cours et des programmes disponibles.	<p><u>Enseignement traditionnel :</u></p> <ul style="list-style-type: none">- Les enseignants et étudiants font face à des difficultés car les études peuvent prendre beaucoup de temps.- Il peut également être difficile de retenir des informations lorsqu'on se contente de lire ou d'écouter des cours. <p><u>Enseignement en ligne mise en place progressivement :</u></p> <ul style="list-style-type: none">- Il y a encore de nombreux instituts qui n'ont pas encore de plateforme d'étude en ligne.- Les établissements utilisent la plupart du temps des plateformes communs au grand public qui ne comblent pas tout leur besoin et sont submergés de fonctionnalités que les utilisateurs ne comprennent pas et ne comptent pas utiliser.- Les enseignants et les étudiants se heurtent souvent à la difficulté d'accès à internet, qui est à Madagascar assez coûteux pour peu de connexion.

4.3. Conception avant-projet

4.3.1. Proposition de solutions

Face à ces inconvénients, l'entreprise iTS souhaite intervenir en proposant aux établissements scolaires une plateforme d'étude en ligne, qui soit adéquat à leurs besoins et qui soit plus accessible aux étudiants. Cette plateforme mettra en relation des enseignants et des étudiants de sorte à échanger et établir un suivi des cours.

Pour la mise en place de cette plateforme, trois solutions se posent :

- Solution 1 : Concevoir une plateforme de visioconférence qui mettra en relation les étudiants et les enseignants pour assister à un cours ;
- Solution 2 : Concevoir une plateforme web pour la mise en ligne de supports pédagogiques ;
- Solution 3 : Concevoir une plateforme web interactive dont les cours peuvent être accessibles via un bot messenger.

4.3.2. Avantages et inconvénients de chaque solution

Les avantages et les inconvénients de chaque solution s'affichent dans le tableau 11.

Tableau 11 : Avantages et inconvénients de chaque solution

Solutions	Avantages	Inconvénients
Solution 1	<ul style="list-style-type: none">– Une bonne alternative pour minimiser les cours en présentielles– Utile pour les cours dont les enseignants sont à l'étranger– Facilité sur les interactions directement durant le cours	<ul style="list-style-type: none">– Difficulté d'accès sur les cours synchrones– Non-assistance au cours en cas de problèmes techniques– Beaucoup de concurrences facilement accessibles car moins chers en data
Solution 2	<ul style="list-style-type: none">– Une bonne alternative pour minimiser les cours en présentielles– Utile pour les cours dont les enseignants sont à l'étranger	<ul style="list-style-type: none">– Complication pour interagir directement avec l'enseignant en cas de non compréhension du cours– Le partage se fait uniquement à un sens : enseignant partage des cours

	<ul style="list-style-type: none"> – Possibilité d'accéder aux cours où et quand on le peut 	
Solution 3	<ul style="list-style-type: none"> – Une bonne alternative pour minimiser les cours en présentielles – Utile pour les cours dont les enseignants sont à l'étranger – Possibilité d'accéder au cours où et quand on le peut – Facilité d'effectuer les tâches attribuées où et quand on le peut 	<ul style="list-style-type: none"> – Complication pour interagir directement avec l'enseignant en cas de non compréhension du cours

4.3.3. Choix de la solution et justification

La troisième solution est la solution retenue pour la mise en œuvre du projet car elle semble plus appropriée à notre problématique.

En effet, elle permet aux étudiants d'assister aux cours plus facilement et de manière asynchrone et plus abordable. Ces étudiants peuvent échanger des documents dans le cadre du cours, effectuer des devoirs ou des quizz de manière plus fluide. En parallèle, elle permet aux enseignants de poursuivre leur pédagogie à leur rythme.

4.4. Méthode et outils utilisés

Il est à souligner que la plupart des outils utilisés a été imposé par l'entreprise. Néanmoins, les raisons et justifications seront exposé dans ce rapport par le biais de comparaisons entre des méthodes et des outils.

4.4.1. Méthode de gestion de projet

Choisir une méthodologie pour conduire un projet permet à tous les acteurs de travailler efficacement ensemble, en suivant des règles clairement définies. Les méthodologies de management de projet vous aident à accomplir chaque étape de votre projet, de la planification à la mise en œuvre, dans un souci d'efficacité et rentabilité. [b]

Pour une bonne gestion du projet, le tableau 12 présente une comparaison entre l'approche traditionnelle et l'approche Agile [1].

Tableau 12 : Tableau comparatif entre l'approche traditionnelle et l'approche Agile

Thème	Approche traditionnelle	Approche Agile
Cycle de vie	En cascade ou en V, sans rétroaction possible, phases séquentielles	Interactif et incrémental
Planification	Préditive, caractérisée par des plans plus ou moins détaillés sur la base d'un périmètre et d'exigences définies et stables au début du projet.	Adaptative avec plusieurs niveaux de planification avec ajustements si nécessaires au fil de l'eau en fonction des changements survenus.
Equipe	Une équipe avec des ressources spécialisées, dirigées par un chef de projet.	Une équipe responsabilisée où l'initiative et la communication sont privilégiées, soutenue par le chef de projet.
Qualité	Contrôle qualité à la fin du cycle de développement. Le client découvre le produit fini.	Un contrôle qualité précoce et permanent, au niveau du produit et du processus. Le client visualise les résultats tôt et fréquemment.
Changement	Résistance voire opposition au changement. Procès sus lourds de gestion des changements acceptés.	Accueil favorable au changement inéluctable, intégré dans le processus.
Suivi de l'avancement	Mesure de la conformité aux plans initiaux. Analyse des écarts.	Un seul indicateur d'avancement : le nombre de fonctionnalités implémentées et le travail restant à faire.
Gestion des risques	Processus distinct, rigoureux, de gestion des risques.	Gestion des risques intégrée dans le processus global, avec responsabilisation de chacun dans l'identification et la résolution des risques. Pilotage par les risques.

Après analyse, le choix s'est porté sur la méthode Agile. En effet, elle s'appuie sur l'utilisation d'un cadre méthodologique léger mais suffisant centré sur l'humain et la communication. En ingénierie logicielle, les pratiques agiles mettent en avant la collaboration entre des équipes autoorganisées et pluridisciplinaires et leurs clients [2]. Elles préconisent une planification adaptative, un développement évolutif, une livraison précoce et une amélioration continue, et elles encouragent des réponses flexibles au changement [3].

Les méthodes Agiles disponibles sont nombreuses et peuvent être source de confusion.

Les méthodes Agiles les plus populaires en usage aujourd'hui sont : l'eXtreme Programming (XP), Scrum, Feature Driven Development (FDD), Lean Software Development, Agile Unified Process (Agile UP ou AUP), Crystal et Dynamic Systems Development Method (DSDM).

Scrum est la méthode choisie pour la mise en œuvre du projet. Elle est une méthode agile qui s'adapte aux projets Web, et permet aux développeurs d'être au plus près des besoins du marché, en leur évitant de développer des choses qui ne seront pas utilisées.

Les principaux acteurs sur Scrum sont :

- Le Product Owner, qui est l'expert qui collabore avec le client
- Le Scrum Master, qui est le guide de l'avancement du projet
- L'équipe qui est constitué des développeurs

La figure 4 résume le fonctionnement de la méthode Scrum. [c]

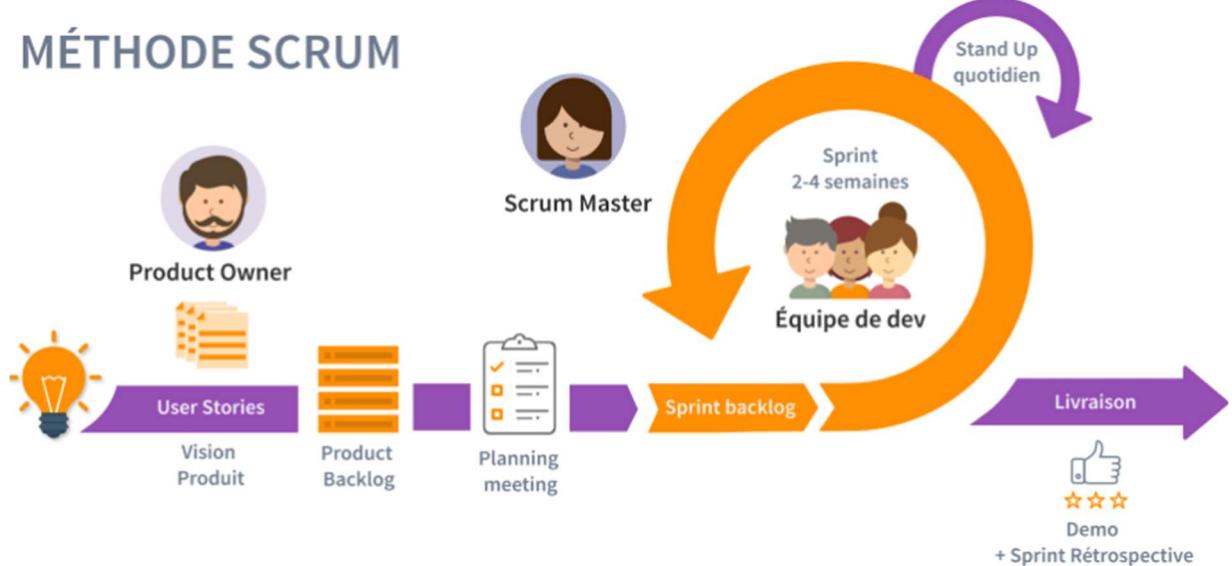


Figure 4: Fonctionnement de la méthode SCRUM

En premier lieu, le Product Owner rencontre le client et analyse son besoin. Il identifie toutes les fonctionnalités dont le produit devra être composé (les user stories) dans ce qui s'appelle le Product Backlog. En second lieu, on entame une répartition de chacune des tâches à faire. L'équipe trie les fonctionnalités et tâches qu'elle répartit dans des Sprint (durée de cycle de deux semaines). En troisième lieu, tous les vendredis qui closent un sprint on teste les bénéfices de la fonctionnalité avec le Product Owner.

4.4.2. Choix de l'outil de modélisation

Le langage de modélisation graphique choisi par l'équipe est le langage UML. En effet, il apporte une compréhension rapide du programme à d'autres développeurs externes en cas de reprise du logiciel et facilite sa maintenance.

Le tableau 13 résume une comparaison entre les outils de modélisation avec le langage UML[d].

Tableau 13: Tableau comparatif entre les outils de modélisation

Nom	ArgoUML	UML Designer	Visual Paradigm for UML
Intégré ou Autonome	Autonome	Intégrable dans Eclipse	Intégrable dans Eclipse, NetBeans et IntelliJ
Ergonomie	Facile d'utilisation pour débutant mais pas pour les professionnels	Ouverte	Dispose de différentes fonctions facilitant la conception des diagrammes aux débutants
Types de graphiques pris en charge	Cas d'utilisation, Classe, Etat, Activité, Séquence, Collaboration, Déploiement	Cas d'utilisation, Classe, Etat, Activité, Séquence, Collaboration, Déploiement, Composant	Cas d'utilisation, Classe, Objet, Etat, Activité, Séquence, Collaboration, Déploiement, Communication, Composant, Package, Structure composite

Nous avons choisi de modéliser avec l'outil Visual Paradigm for UML de par ces qualités et sa richesse en termes de graphiques pris en charge qui conviennent à la conception.

4.4.3. Choix des langages de programmation

Dans l'optique de produire une plateforme performante, nous avons séparé le frontend et l'API backend. De ce fait, ce sont des applications entièrement différentes qui utilisent leurs propres langages, frameworks et mécanismes de développement, et l'API backend sera disponible pour toutes les applications qui pourront l'utiliser.

4.4.3.1. Choix du langage pour le Frontend

En développement web, la notion de « frontend » fait référence à l'ensemble des éléments visibles et accessibles directement sur un site web.

Le tableau 14 illustre les comparaisons entre deux des plus célèbres frameworks d'aujourd'hui [e].

Tableau 14 : Tableau comparatif entre Vuejs et Angular

	Angular	Vuejs
Connu comme	Un framework JS complet	Un framework JS progressif
Langage source	TypeScript	Javascript (supporte aussi le TypeScript)
Taille	563 kB (minifié et compressé)	33.5kB (minifié et gzipé)
Performances	Rapide	Comparativement plus rapide
Etoiles Github	Rapide	Comparativement plus rapide
Courbe d'apprentissage	Abrupte	Plus facile par rapport à Angular
Rendu du navigateur	Shadow DOM	Virtual DOM

Le choix du langage pour le Frontend s'est porté sur Vuejs car elle est plus facile à prendre en main, légère et avec des performances élevées.

4.4.3.2. Choix du langage pour le Backend

En développement web, la notion de « backend » concerne toute la partie invisible de la conception d'un site internet, tel que l'accès à la base de données.

Le tableau 15 les comparaisons entre deux des frameworks Backend les plus connus [f].

Tableau 15: Tableau comparatif entre Nodejs et Django Rest Framework

	NodeJS	Django Rest Framework
Programmation en	Javascript	Python
Architecture	Modèle piloté par les évènements	Modèle : Modèle – Template -Vue
Sécurité	Moins sécurisé	Plus sécurisé
Performance	Pas aussi rapide que Django Rest Framework	Offre de meilleures performances
Efficacité	Moins efficace	Plus efficace
Documentation et templates	Pas de templates, nous devons construire l'application à partire de zéro	Templates intégrés
Scalabilité	Plus scalable	Moins scalable
Communauté	Plus grande communauté que Django Rest Framework	Plus petite communauté que Nodejs

Le choix du langage pour le backend sur Django Rest Framework pour sa facilité d'approche, sa puissance, son efficacité dans la création d'un projet web.

4.4.3.3. Choix du langage pour les tests

Le développement du projet est un développement piloté par les tests. Comme le choix des langages pour le frontend et du backend ont été effectué en amont, pour suivre la logique et garder l'uniformité, le langage pour les tests unitaires est le langage Python. En effet, notamment pour le Backend, DRF a des classes spéciales pour les tests unitaires, évidemment en Python.

Le développement des tests unitaires pour le backend est effectué avec le langage Python, avec les classes :

- TestCase : pour les tests concernant les modèles
- APITestCase : pour les tests relatifs aux vues

Le développement des tests de bout en bout du frontend est effectué avec le langage Python avec la librairie Playwright.

4.4.4. Choix de l'éditeur de texte

Une plateforme web est, pour une grande partie, composé de fichiers texte. Pour cette raison, afin de développer dans les meilleures conditions, nous devons choisir notre éditeur de texte soigneusement.

Un éditeur de texte peut avoir peu de fonctionnalités de base mais être extensible pour s'adapter à nos besoins.

Le tableau 16 résume un comparatif entre deux des éditeurs de texte les plus utilisés [g].

Tableau 16 : Tableau comparatif entre Visual Studio Code et Sublime Text

	Visual Studio Code	Sublime Text
Intégration Git	3/5	4/5
Gestion des codes	4/5	4/5
Achèvement du code	3/5	3/5
Débogage	4/5	2/5
Performance	3/5	5/5
Refactoring	1/5	0/5
Multiplateforme	5/5	3/5

Le choix de l'éditeur de texte se porte sur Visual Studio Code car il est plus facile à configurer pour le débogage et la création de projets de code, il s'exécute sur ARM et dispose d'une tonne de plugins et d'un support communautaire. Parallèlement, Visual Studio Code est gratuit tandis que Sublime Text est payant.

4.4.5. Choix du SGBD

L'acronyme SGBD, pour Système de Gestion de Base de Données, désigne un logiciel informatique permettant le stockage, la consultation, la mise à jour, la structuration ou encore le partage d'informations dans une base de données.

Le tableau 17 présente une comparaison de trois SGBD parmi les plus SGBD populaires sur les fonctionnalités, les prix, les déploiements et le support. [h]

Tableau 17: Comparaison entre MariaDB, MySQL et PostgreSQL

Comparaison	MariaDB	MySQL	PostgreSQL
Avis sur l'évaluation des fonctionnalités			
Rapport qualité-prix	4,8/5	4,6/5	4,6/5
Fonctionnalités	4,7/5	4,5/5	4,6/5
Simplicité	4,4/5	4,4/5	4,3/5
Support client	4,1/5	4,1/5	4,1/5
Prix			
Version gratuite	Oui	Non	Oui
Version d'essai gratuite	Oui	Oui	Oui
Déploiement			
Basé sur le cloud	Oui	Oui	Oui
Sur site	Oui	Non	Oui

Nous avons choisi d'utiliser MariaDB comme système de gestion de base de données pour ses performances et ses qualités comparés à MySQL et PostgreSQL.

Chapitre 5 : Analyse conceptuelle

L'analyse conceptuelle se situe au cœur de l'élaboration de la plateforme. Elle permet de comprendre, modéliser et concrétiser les besoins et les tâches qui emmènent des nécessités.

5.1. Dictionnaire de données

Le dictionnaire des données est un document qui regroupe toutes les données que nous aurons à conserver dans notre base de données (et qui figureront donc dans le diagramme de classe de conception globale). L'élaboration de ce dictionnaire des données est une étape qui a son importance, surtout si nous sommes plusieurs à travailler sur une même base de données, d'un volume conséquent.

Le tableau 18 nous expose les données utilisées par l'application.

Tableau 18: Dictionnaire de données

Rubrique	Description	Structure		Observation
		Type	Taille	
abbreviation	Abréviation du nom d'un institut	AN	10	
adress	Adresse d'un institut	AN	56	
answer	Nomenclature choisie par l'institut pour représenter la réponse	AN	16	
anwer_id	Identifiant d'une réponse	N		anwer_id est utilisé en tant que clé étrangère
attempt_id	Identifiant de l'effectuation d'un quizz, devoir ou test	N		attempt_id est utilisé en tant que clé étrangère
author_id	Identifiant de l'auteur d'une notification			author_id est utilisé en tant que clé étrangère
background	Chemin de l'image qui illustre l'image de fond d'un institut	AN	100	
category	Nomenclature choisie par l'institut pour représenter la catégorie d'un cours	AN	16	

category_id	Identifiant d'une catégorie d'un cours	N		category_id est utilisé en tant que clé étrangère
chapter	Nomenclature choisie par l'institut pour représenter le contenu	AN	16	
code	Code unique représentant un cours	AN	10	
color_1	Couleur primaire d'un institut	AN	8	
color_2	Couleur secondaire d'un institut	AN	8	
color_3	Couleur du mode d'affichage d'un institut	AN	8	Exemple : mode sombre ou mode lumineux
comment	Commentaire sur une question d'un quizz	AN	250	
content_id	Identifiant d'un contenu	N		content_id est utilisé en tant que clé étrangère
course	Nomenclature choisie par l'institut pour représenter le cours	AN	16	
course_id	Identifiant d'un cours	N		course_id est utilisé en tant que clé étrangère
created_at	Date de création d'une ligne dans la table	Date		<p>Cette table peut être une table :</p> <ul style="list-style-type: none"> - « Content » (pour un contenu d'une partie constituant un cours) - « Attempt » (pour l'effectuation d'un test, d'un devoir ou d'un quizz) - « Notification » (pour les notifications)

date	Pour la table Progress, date représente la date où un étudiant a atteint une partie. Pour la table Attempt, date représente la date où un étudiant a effectué un quizz, rendu un devoir ou passé un test	Date - Time		Par défaut, date est toujours affecté à la date de création de la ligne.
date_end	Date de fin d'un évènement enregistré dans le calendrier	Date - Time		
date_start	Date de début d'un évènement enregistré dans le calendrier	Date - Time		
description	Brève description de la table	AN	120	Cette table peut être une table : <ul style="list-style-type: none"> – « User » (pour un utilisateur) – « Schedule » (pour un calendrier associé à un cours) – « Course » (pour un cours) – « Part » (pour une partie constituant un cours) – « Institut » (pour l'institut en question)
disable_at	Date à laquelle la notification sera supprimée	Date - Time		

display_all	Pour un contenu de type quizz, display_all est un choix pour confirmer si l'affichage des questions se fait d'un coup ou par question	N	1	0 pour faux et 1 pour vrai
duration	Durée d'un évènement enregistré dans le calendrier	Time		
email	Adresse électronique unique rattaché à un utilisateur	AN	30	
exam	Nomenclature choisie par l'institut pour représenter l'examen	AN	16	
group	Nomenclature choisie par l'institut pour représenter la classe	AN	16	
group_id	Identifiant d'une classe	N		group_id est utilisé en tant que clé étrangère
id	Identifiant d'une table	N		Chaque table possède un identifiant id
institut_id	Identifiant d'un institut	N		institut_id est utilisé en tant que clé étrangère
image	Chemin de l'image qui illustre le contenu d'une ligne d'une table	AN	100	<p>Cette table peut être une table :</p> <ul style="list-style-type: none"> – « User » (pour un utilisateur) – « Course » (pour un cours) – « Question » (pour une question dans un quizz)
is_correct	Booléen pour noter si une réponse s'agit de la bonne réponse ou non	N	1	0 pour faux et 1 pour vrai

is_rendered	Booléen pour noter qu'un contenu s'agit d'un devoir à rendre ou non	N	1	0 pour faux et 1 pour vrai
is_test	Booléen pour noter qu'un contenu s'agit d'un test ou non	N	1	0 pour faux et 1 pour vrai
lesson	Nomenclature choisie par l'institut pour représenter la partie	AN	16	
level	Nomenclature choisie par l'institut pour représenter le niveau	AN	16	
level_id	Identifiant d'un niveau	N		level_id est utilisé en tant que clé étrangère
libelle	Libellé attaché à un cours	AN	25	
link	Lien externe qui sert de support pour un contenu de cours	AN	100	
logo	Chemin de l'image qui illustre le logo d'un institut	AN	100	
mark	La note attribuée à une question dans un quizz	N	1	
matricule	Numéro matricule unique désignant un utilisateur	AN	16	
name	Nom qui représente la table.	AN	56	<p>Cette table peut être une table :</p> <ul style="list-style-type: none"> - « Type » (pour type de cours) - « Parcours » (pour un parcours) - « Level » (pour un niveau)

				<ul style="list-style-type: none"> - « Category » (pour une catégorie de cours) - « Course » (pour un cours) - « Part » (pour une partie constituant un cours) - « Content » (pour un contenu d'une partie) - « Institut » (pour l'institut en question)
notification_id	Identifiant d'une notification	N		notification_id est utilisé en tant que clé étrangère
parcours	Nomenclature choisie par l'institut pour représenter le parcours	AN	16	
parcours_id	Identifiant d'un parcours	N		parcours_id est utilisé en tant que clé étrangère
part_id	Identifiant d'une partie d'un cours	N		part_id est utilisé en tant que clé étrangère
path	Chemin vers un fichier qui correspond à une ligne créée dans une table	AN	100	<p>Cette table peut être une table :</p> <ul style="list-style-type: none"> - « Contenu » (pour un contenu d'une partie constituant un cours) - « Attempt » (pour l'effectuation d'un test, d'un devoir ou d'un quizz)
phone	Numéro de téléphone de l'institut	AN	13	

recipient_id	Identifiant de l'utilisateur à laquelle une notification s'adresse	N		
ref	Identifiant du contenu auquel la notification est rattachée	N		Peut-être nulle
result	Nomenclature choisie par l'institut pour représenter le résultat	AN	16	
role	Role d'un utilisateur	AN	7	Il n'y a que trois possibilités : – ADMIN pour un administrateur – PROF pour un enseignant – STUDENT pour un étudiant
sequence	Ordre d'apparition d'une partie d'un cours lors de l'affichage du cours	N		
student	Nomenclature choisie par l'institut pour représenter l'étudiant	AN	16	
student_id	Identifiant d'un étudiant	N		student_id est utilisé en tant que clé étrangère
subject	Nomenclature choisie par l'institut pour représenter le sujet	AN	16	
teacher	Nomenclature choisie par l'institut pour représenter l'enseignant	AN	16	
teacher_id	Identifiant d'un enseignant	N		teacher_id est utilisé en tant que clé étrangère
title	Titre représentant un calendrier d'un cours	AN	56	

test	Nomenclature choisie par l'institut pour représenter le test	AN	16	
text	Texte correspondant à la ligne créée dans la table	AN	250	<p>Cette table peut être une table :</p> <ul style="list-style-type: none"> – « Content » (pour un contenu d'une partie constituant un cours) – « Question » (pour une question dans un quizz) – « Answer » (pour la réponse d'une question) – « Attempt » (pour l'effectuation d'un test, d'un devoir ou d'un quizz) – « Notification » (pour les notifications)
type	Type catégorisant une ligne de la table	7		<p>Cette table peut être une table :</p> <ul style="list-style-type: none"> – « Course » (pour un cours) – « Content » (pour un contenu d'une partie constituant un cours) – « Attempt » (pour l'effectuation d'un test, d'un devoir ou d'un quizz)

				– « Notification » (pour les notifications)
unread	Booléen pour déterminer si la notification est lue ou non	N	1	0 pour faux et 1 pour vrai
visible	Booléen pour faire en sorte que la ligne de la table soit visible ou pas lors d'un affichage	N	1	0 pour faux et 1 pour vrai
question	Nomenclature choisie par l'institut pour représenter la question	AN	16	
question_id	Identifiant d'une question	N		question_id est utilisé en tant que clé étrangère

Légende :

- AN : Alpha Numérique
- N : Numérique
- Date : Date
- Date - Time : Temps précis en date et en heure
- Time : temps précis

5.2. Règles de gestion

Les règles de gestion guident et documentent la création d'un modèle. Elles sont rédigées pour répondre aux besoins du client et respecter les contraintes qui régissent le fonctionnement global de la plateforme.

La mise en place de la plateforme iTS.mg est manié par les règles de gestion qui suivent.

Tout d'abord, les règles de gestion qui ont la portée sur la base de données :

- RG 1 : Les noms des tables et de leurs attributs sont en anglais.

Le tableau 19 résume les noms des tables et leur signification.

Tableau 19: Nom des tables et leur signification

Nom	Signification
Answer	Réponse
Attempt	Tentative
Category	Catégorie d'un cours
Color	Couleur
Content	Contenu
Course	Cours
Group	Classe
Institute	Institut
Level	Niveau
Nomenclature	Nomenclature
Notification	Notification
Parcours	Parcours
Part	Partie
Progress	Progression
Result	Résultat
Question	Question
Schedule	Planification horaire, utilisé pour stocker les programmes relatifs à un cours
Type	Type de cours
User	Utilisateur
UserNotificationState	Etat d'une notification par rapport à un utilisateur

- RG 2: Une classe est défini par un couple unique d'un niveau et d'un parcours.
- RG 3: Un cours est assigné à un ou plusieurs classes.
- RG 4: Un cours peut avoir aucun ou plusieurs programmes dans le calendrier.

- RG 5: Un cours est catégorisé par une seule catégorie et une seule.
- RG 6: Une catégorie ne catégorise aucun ou plusieurs cours.
- RG 7: Un cours est classé par un seul type et un seul.
- RG 8: Un type ne classe aucun ou plusieurs cours.
- RG 9: Un cours est composé d'aucun ou de plusieurs parties.
- RG 10: Une partie appartient à un cours et un seul.
- RG 11: Une partie comporte un ou plusieurs contenus.
- RG 12: Un contenu appartient à une partie et une seule.
- RG 13: Un contenu ne peut avoir que les types : texte, lien, devoir, vidéo, fichier et quizz.
- RG 14: Une notification peut s'agir d'un cours ou d'un contenu de type devoir ou quizz.
- RG 15: Une notification est assigné à un ou plusieurs utilisateurs.
- RG 16: Un contenu de type quizz peut avoir un ou plusieurs questions.
- RG 17: Une question est rattaché à un contenu de type quizz et un seul.
- RG 18: Une question peut avoir deux ou plusieurs réponses.
- RG 19: Une réponse tient à une question et une seule.
- RG 20: Une question peut avoir une ou plusieurs bonnes réponses.
- RG 21: Une tentative aboutit à un résultat et un seul.
- RG 22: Un résultat correspond à une tentative et une seule.
- RG 23: Une tentative est relié à une ou plusieurs tentatives à une réponse.
- RG 24: Une tentative à une réponse est associé à un seul tentative.
- RG 25: Un ensemble de couleur correspond à un institut et un seul.
- RG 26: Une question peut avoir une ou plusieurs bonnes réponses.
- RG 27: La progression d'un étudiant sur un cours peut être enregistré.
- RG 28: Un institut peut avoir une nomenclature et une seule.

- RG 29: Une nomenclature correspond à un institut et un seul.
- RG 30: Un institut désigne un ensemble de couleur et un seul.

Ensuite, les règles de gestion qui ont la portée sur l'affichage et la manipulation de la plateforme :

- RG 31 : Les informations sur l'institut sont affichées sur la page de connexion.
- RG 32 : La page d'accueil de tout utilisateur comporte la liste des cours.
- RG 33 : Chaque institut peut choisir ses propres couleurs et ses propres nomenclatures.
- RG 34: Un administrateur peut modifier tous les paramètres reliés à l'affichage de la plateforme de son institut.
- RG 35: Un administrateur se charge de la gestion de tous les paramètres reliés au cours.
- RG 36: Un administrateur est le seul qui peut gérer les enseignants et les étudiants.
- RG 37: Un administrateur peut créer un ou plusieurs annonces dédiés aux classes.
- RG 38: Un administrateur est le seul qui peut créer un ou plusieurs annonces dédiés aux enseignants.
- RG 39: Un administrateur peut accéder à la liste de tous les cours.
- RG 40: Un enseignant peut enseigner aucun ou plusieurs cours.
- RG 41: Un enseignant ne peut accéder qu'à la liste des cours qu'il a créé.
- RG 42: Un enseignant peut gérer ses cours et leur composants.
- RG 43: Un cours peut avoir deux types : soit où les parties s'affichent par rapport à la progression d'un élève, soit les parties s'affichent d'un coup.
- RG 44: La création d'un cours implique l'envoi d'une notification aux élèves des classes concernées.
- RG 45: Un enseignant peut récupérer les fichiers que les étudiants ont rendu.
- RG 46: La création d'un devoir implique l'envoi d'une notification aux élèves des classes concernées.

- RG 47: Un compte rendu sur un devoir doit pouvoir être accessible par son enseignant.
- RG 48: Un enseignant peut créer un ou plusieurs contenu de type quizz.
- RG 49: Un quizz peut être affiché soit de manière à afficher toutes les questions d'un coup, soit de manière à afficher les questions une à une.
- RG 50: Un rapport sur un quizz doit pouvoir être accessible par son enseignant.
- RG 51: La création d'un test pour un cours crée un évènement sur le calendrier de ce cours.
- RG 52: La création d'un test implique l'envoi d'une notification aux élèves des classes concernées.
- RG 53: Un enseignant peut créer aucun ou plusieurs notifications adressées aux classes.
- RG 54: Un enseignant peut consulter les notifications qui lui sont adressées par l'administrateur.
- RG 55: Un étudiant ne peut accéder qu'aux cours qui sont assignés à sa classe.
- RG 56 : La progression d'un élève par rapport à un cours peut être enregistré.
- RG 57 : La progression d'un élève sur un cours est à enregistrer si le cours est de type « à afficher par partie ».
- RG 58 : Un cours de type « à afficher par partie » implique que l'affichage du cours du côté étudiant ne retourne que les parties auxquelles correspondent sa progression.
- RG 59: Un étudiant peut consulter les notifications qui lui sont adressées.
- RG 60: Cliquer sur une notification doit mener au cours ou au contenu correspondant à la notification si ce n'est pas juste une simple annonce.
- RG 61: Une notification consultée et non consultée ne doit pas être affiché de la même façon.
- RG 62: Un étudiant peut effectuer aucun ou plusieurs fois un quizz.

- RG 63: Le résultat du quizz doit être affiché à chaque fois qu'un élève a soumis toutes ses réponses.
- RG 64: Un étudiant peut rendre aucun ou plusieurs fichiers pour un contenu de type devoir. Dans le cas de plusieurs fichiers, seul le dernier sera pris en compte.

5.3. Désignation des rôles de l'équipe SCRUM

L'équipe SCRUM est composée de plusieurs rôles clés, chacun ayant sa propre responsabilité dans le bon fonctionnement du processus.

Ces rôles sont classés en deux catégories :

- Le rôle de « cochon », qui désigne ceux qui sont impliqués dans le projet
- Le rôle de « poule », qui désigne ceux qui sont concernés par le projet

Dans le rôle de « cochon », nous distinguons :

- le Product Owner, attribué à une personne qui sera responsable de la vision du produit et de la priorité des fonctionnalités ;
- le Scrum Master, assigné à une personne qui sera responsable de veiller à ce que le processus soit suivi et que tout le monde soit sur la même longueur d'onde ;
- les membres de l'équipe de développement, dont moi-même, seront responsables du développement du produit et de sa livraison.

Dans le rôle de « poule », nous avons les clients de l'entreprise qui bénéficieront de la plateforme.

5.4. Etapes de l'élaboration du Product Backlog

Par définition, le « Product Backlog » est une liste de toutes les caractéristiques, fonctionnalités et améliorations qui doivent être apportées à un produit. Il est généralement tenu à jour par le Product Owner, et est utilisé par l'équipe de développement pour prioriser le travail et s'assurer que le produit va toujours de l'avant.

Les cinq étapes de l’élaboration du Product Backlog sont :

- la vision du projet
- la liste des acteurs
- les thèmes
- la première Release
- les Users stories

5.4.1. Vision du projet

La vision du projet est « mettre en relation les enseignants et les élèves d’un institut de formation dans le cadre pédagogique de manière le plus accessible possible et plus sécurisé ».

5.4.2. Liste des acteurs

Les acteurs sont les intervenants qui utilisent le système dans tous ses aspects.

Dans notre cas, les acteurs sont l’administrateur, l’enseignant et l’élève.

Le tableau 20 résume les informations sur les acteurs du projet.

Tableau 20: Liste des acteurs

Acteur	Description	NCT	NCM
Administrateur	<ul style="list-style-type: none">– Gestion des différents paramètres de la plateforme– Gestion des niveaux, parcours, classes et utilisateurs– Envoi de notifications aux élèves et aux enseignants	Elevé	Elevé
Enseignant	<ul style="list-style-type: none">– Gestion des cours et de ses composants– Consultation des rapports des quizz et des devoirs– Envoi de notifications aux élèves	Moyen	Elevé
Elève	<ul style="list-style-type: none">– Consultation des cours et des notifications adressés– Traitement des quizz et des tests– Envoi des fichiers pour les devoirs	Bas	Moyen

Légende :

- NCT : niveau de connaissance technologique
- NCM : niveau de connaissance métier

5.4.3. Thèmes

De la vision du produit, nous pouvons extraire les thèmes principaux. Ils forment l'objectif central du produit.

La liste des thèmes de la plateforme est :

- La connexion à la plateforme
- La gestion des différents paramètres de l'institut
- La gestion des utilisateurs
- L'envoi des notifications
- La réception des notifications
- La gestion des cours et de ses composants
- La consultation des cours et de ses composants
- La consultation des rapports des quizz et des devoirs

Le tableau 21 montre les détails concernant ces principaux thèmes.

Tableau 21: Détails concernant les principaux thèmes

Thème	Description	Valeur	Critères de satisfaction	Utilisateurs
Connexion à la plateforme	Accès à la page de connexion, se connecter et enfin accéder à la plateforme.	65	Connaître les erreurs si la connexion a échoué. Accéder au tableau de bord si la connexion a réussi.	Tous les utilisateurs
La gestion des différents paramètres de l'institut	Toutes les opérations sur le paramétrage de l'institut	80		Administrateur

La gestion des utilisateurs	Toutes les opérations sur les utilisateurs	70	Diviser séparément par classe	Administrateur
L'envoi des notifications	Création des notifications par les administrateurs et les enseignants	40	Un administrateur peut notifier les enseignants et les classes ; un enseignant peut notifier les classes	Administrateur Enseignant
Réception des notifications	Réception par les enseignants et les élèves, des notifications qui leur sont adressées	35		Enseignant Etudiant
La gestion des cours et de ses composants	Toutes les opérations sur les cours et leurs composants	65	Un enseignant n'a accès qu'aux cours qu'il a créé.	Enseignant
La consultation des cours et de ses composants	Toutes les opérations qui s'en suivent de la consultation des cours et des composants	60	L'utilisateur doit pouvoir effectuer toutes les opérations liées aux contenus	Etudiant
La consultation des rapports des quizz et des devoirs	Consulter les comptes rendus des quizz et des devoirs par rapport aux classes concernés	45		Enseignant

5.4.4. Vision de la première release

La figure 5 illustre le découpage en sprints et en releases d'un projet.

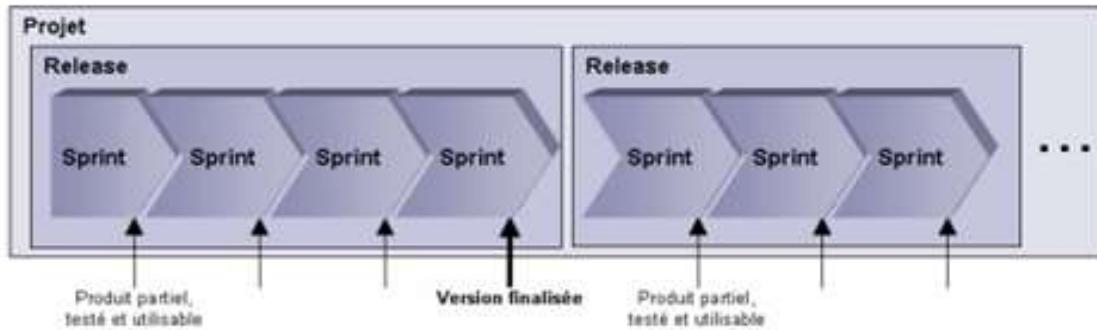


Figure 5: Découpage d'un projet en sprint et en release

La vision de la première release consiste donc à prioriser les thèmes, de ce fait, le premier thème le plus important sera la première release.

Pour notre part, la première release sera « l'administration des paramètres de l'institut ».

Les thèmes classés par priorité seront alors :

- La gestion des différents paramètres de l'institut,
- La gestion des utilisateurs,
- La connexion à la plateforme,
- La gestion des cours et de ses composants,
- La consultation des cours et de ses composants,
- La consultation des rapports des quizz et des devoirs,
- L'envoi des notifications,
- La réception des notifications.

5.4.5. User Story

Un user story est un élément du backlog. Il constitue les spécifications du projet présentées sous forme d'histoires utilisateurs qui décrivent les interactions de l'utilisateur avec le système de façon plus ou moins détaillées.

Un User Story respecte la syntaxe suivante : « En tant que ..., je veux ..., afin de ... ».

Les user stories de la plateforme sont représentés par la liste qui suit.

En tant qu'administrateur, je veux une page sur l'institut, afin d'effectuer des modifications sur les informations sur l'institut.

En tant qu'administrateur, je veux une page d'administration des différents paramètres propres à l'institut, afin de gérer les couleurs, les nomenclatures qui lui sont spéciales.

En tant qu'administrateur, je veux gérer les catégories de cours, afin de permettre la création de cours avec les différentes catégories disponibles.

En tant qu'administrateur, je veux gérer les utilisateurs, afin de pouvoir gérer les enseignants et les étudiants.

En tant qu'administrateur, je veux envoyer des notifications à des enseignants et des étudiants, afin de leur faire part des annonces de l'institut.

En tant qu'enseignant, je veux gérer les cours et leur composant, afin de partager des cours et des supports pédagogiques avec les étudiants de mes cours.

En tant qu'enseignant, je veux consulter les rapports des quiz et des devoirs, afin de voir qui ont effectué et qui n'ont pas effectué les quizz et les devoirs avec les détails de chaque classe.

En tant qu'enseignant, je veux envoyer des notifications à mes étudiants, afin de leur annoncer les nouvelles importantes.

En tant qu'étudiant, je veux recevoir des notifications sur les annonces, les nouveaux cours, les tests et les quizz, afin de m'organiser facilement pour les consulter et les traiter.

En tant qu'étudiant, je veux consulter les cours qui me sont adressé, afin de suivre parfaitement mes cours.

En tant qu'étudiant, je veux traiter des quizz et des tests, afin de savoir mon avancement sur un cours.

En tant qu'étudiant, je veux rendre des devoirs, afin d'obtenir des notes ou des remarques de mon enseignant.

5.5. Product Backlog

Le Product Backlog est une liste hiérarchisée de fonctionnalités ou de tâches qui doivent être complétées afin de créer et de publier un produit logiciel. Il est généralement géré par le Product Owner et est utilisé par l'équipe de développement pour planifier et estimer le travail à effectuer.

Le tableau 22 expose le Product Backlog du projet.

Tableau 22: Product Backlog du projet

ID	Titre	Importance	Estimation (en h)	Démonstration de la fonctionnalité	Commentaires
SPRINT 1					
1	Mise à jour des informations de l'institut	170	26	L'utilisateur se connecte à la plateforme. Il a accès à la page d'administration. Les informations actuelles de l'institut sont affichées et l'utilisateur peut les modifier.	L'utilisateur a un rôle d'administrateur.
2	Mise à jour des couleurs de la plateforme que l'institut utilisera	160	26	L'utilisateur se connecte à la plateforme. Il a accès à la page d'administration des couleurs de la plateforme. Les couleurs actuelles de l'institut sont affichées et l'utilisateur peut les modifier.	L'utilisateur a un rôle d'administrateur.
3	Mise à jour des nomenclatures que la plateforme utilisera	150	18	L'utilisateur se connecte à la plateforme. Il a accès à la page d'administration des nomenclatures. Les nomenclatures actuelles de l'institut sont affichées et l'utilisateur peut les modifier.	L'utilisateur a un rôle d'administrateur.
SPRINT 2					
4	Gestion des niveau	140	17	L'utilisateur se connecte à la plateforme. Il a accès à la page d'administration des niveaux. Il peut consulter la liste des niveaux, effectuer l'ajout de nouveau niveau, la modification et la suppression d'un niveau.	L'utilisateur a un rôle d'administrateur.

5	Gestion des parcours	130	17	L'utilisateur se connecte à la plateforme. Il a accès à la page d'administration des parcours. Il peut consulter la liste des parcours, effectuer l'ajout de nouveau parcours, la modification et la suppression d'un parcours.	L'utilisateur a un rôle d'administrateur.
6	Gestion des classes	120	17	L'utilisateur se connecte à la plateforme. Il a accès à la page d'administration des classes. Il peut consulter la liste des classes, effectuer l'ajout de nouvelle classe, la modification et la suppression d'une classe.	L'utilisateur a un rôle d'administrateur.
7	Gestion des catégories des cours	110	17	L'utilisateur se connecte à la plateforme. Il a accès à la page d'administration des catégories. Il peut consulter la liste des catégories, effectuer l'ajout de nouvelle catégorie, la modification et la suppression d'une catégorie.	L'utilisateur a un rôle d'administrateur.
SPRINT 3					
8	Gestion des enseignants	100	45	L'utilisateur se connecte à la plateforme. Il a accès à la section « enseignant ». Il peut consulter la liste des enseignants (tous ou par classe), ajout de nouveau enseignant, modification et suppression d'un enseignant.	L'utilisateur a un rôle d'administrateur.
9	Gestion des étudiants	90	60	L'utilisateur se connecte à la plateforme. Il a accès à la section « étudiant ». Il peut consulter la liste des étudiants, effectuer l'ajout de nouvel étudiant, la modification et la suppression d'un étudiant.	L'utilisateur a un rôle d'administrateur.

SPRINT 4						
10	Connexion à la plateforme	80	21	L'utilisateur a accès à la page de connexion, remplit le formulaire de la connexion, et accède à la page d'accueil et déconnexion.		
11	Modification du profil	70	14	Une fois que l'utilisateur est connecté, il a accès à ses informations actuelles et aux modifications de ces dernières.		
SPRINT 5						
12	Gestion des cours - côté enseignant	60	100	L'utilisateur se connecte à la plateforme. Il peut consulter les cours créés, procéder à la gestion des cours, des parties des cours, des contenus de ces parties avec affichage distincts des différents types de contenu.	L'utilisateur a un rôle d'enseignant.	
13	Consultation des cours – côté étudiant	50	40	L'utilisateur se connecte à la plateforme. Il peut consulter les cours et des différents composants des cours avec affichage distincts des différents types de contenus.	L'utilisateur a un rôle d'étudiant.	
SPRINT 6						
14	Section des quizz et des devoirs	40	90	L'utilisateur se connecte à la plateforme. Il peut effectuer la création des quiz, gestion des questions, traitement des quizz par les étudiants ; la création des contenus de types devoir, rendement des fichiers devoir par les étudiants.	L'enseignant s'occupe de la création et l'étudiant des traitements des contenus.	
15	Rapport des quizz et des devoirs	30	50	L'utilisateur se connecte à la plateforme. Il peut récupérer des fichiers devoirs, consulter des rapports pour les quizz et les devoirs.	L'utilisateur a un rôle d'enseignant.	

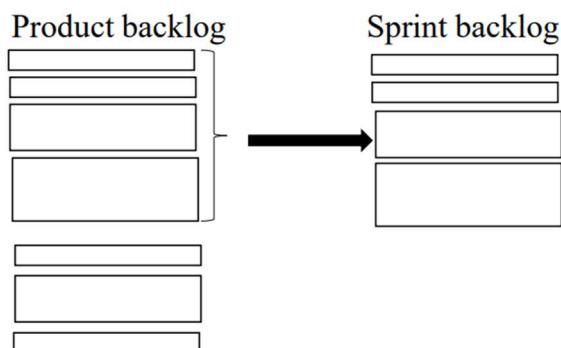
SPRINT 7						
16	Création et gestion des notifications	20	21	L'utilisateur se connecte à la plateforme. Il peut créer et modifier une annonce.	Une notification peut s'agir d'un nouveau contenu ou d'un nouveau cours	
17	Réception des notifications	10	21	L'utilisateur se connecte à la plateforme. Il peut consulter les notifications, qui sont filtrés de sorte à proposer les filtrages sur ceux créés par l'utilisateur, ceux reçus et ceux non-consultés.		

5.6. Sprint Backlog

La réunion de planification du sprint se déroule avec le product owner et l'équipe complète. Cette réunion doit durer au maximum une journée.

Le Product Owner présente à l'équipe les besoins qui ont la plus haute importance dans le Product Backlog. L'équipe sélectionne les besoins qu'elle pense pouvoir implémenter dans le délai imparti. La réunion est découpée en 2 parties : la présentation du travail à faire et la planification du sprint. [4]

La figure 6 montre la constitution du sprint backlog à partir du product backlog.



5.6.1. Présentation du travail à faire

Le Product backlog est une liste de tout le travail qui doit être fait pour mener à bien un projet. La release est un sous-ensemble du backlog de produit qui représente le travail qui doit être fait pour atteindre un objectif spécifique. Le sprint est un sous-ensemble de la release qui représente le travail qui doit être fait dans une période de temps spécifique.

Comme une image vaut mille mots, la figure 7 expose la présentation du travail à faire.



Figure 7: Présentation du travail à faire

5.6.2. Planification du sprint

Pour la planification du sprint, les besoins sont divisés en tâches. Pour chaque sprint, les tâches sont :

- Création de l'endpoint qui permet d'effectuer la tâche
- Codage de la fonction correspondant
- Codage des tests unitaires du backend
- Création des templates
- Création de l'interface et attachement avec le l'endpoint du backend
- Création des tests de bout en bout du frontend

5.7. Modèle du domaine

En génie logiciel, un modèle de domaine est le modèle conceptuel d'un domaine. Il comprend tant le comportement que les données. De ce fait, il est la première approche du diagramme de classe du projet.

La figure 8 illustre le modèle de domaine de la plateforme.

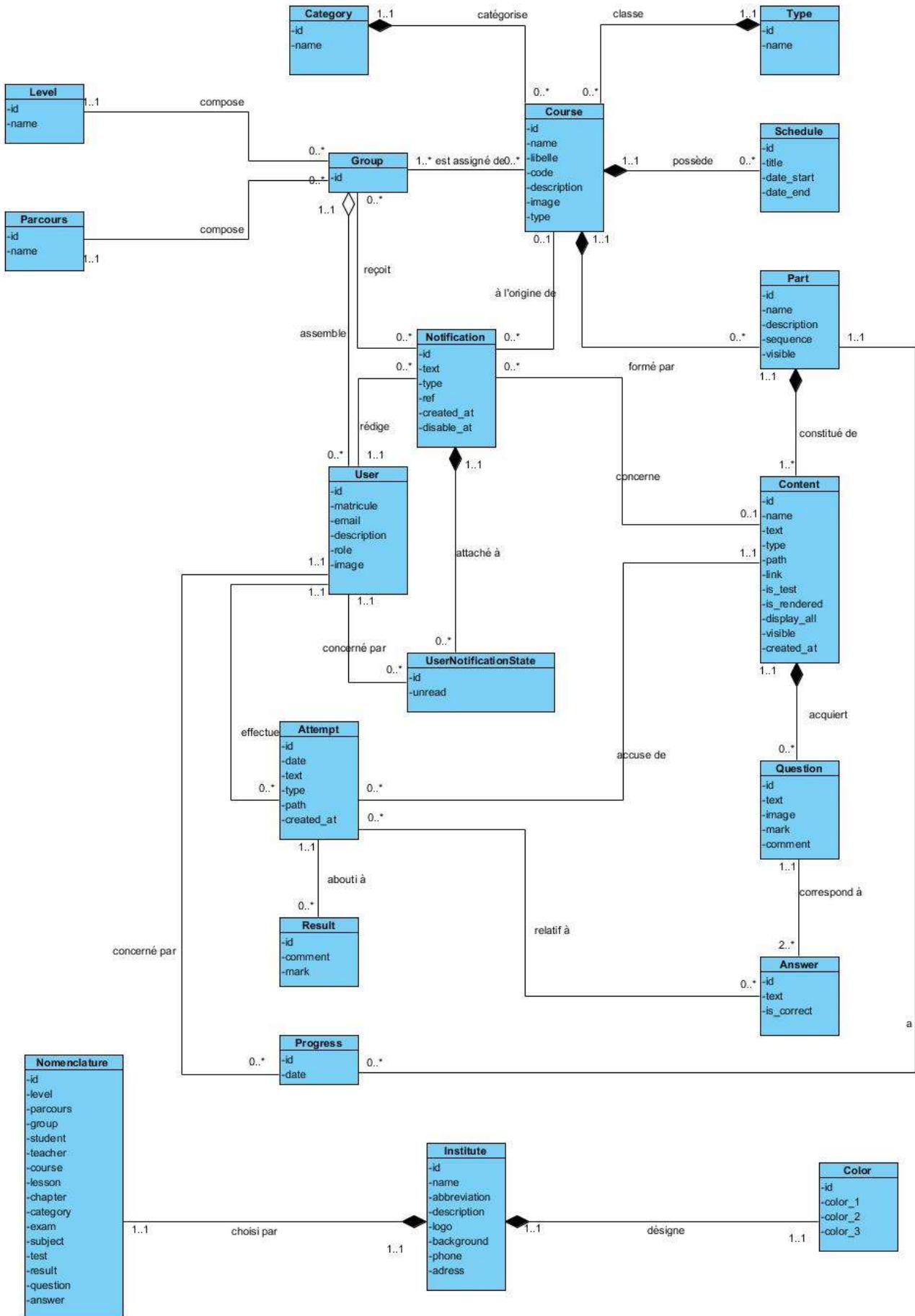


Figure 8: Modèle de domaine de la plateforme

5.8. Diagramme de classe de conception pour chaque sprint

La mise en œuvre du projet sera organisée en six sprint et nous allons détailler le diagramme de classe de conception de chacun d'eux.

Un diagramme de classes fournit une vue globale d'un système en présentant ses classes, interfaces et collaborations, et les relations entre elles. Les diagrammes de classes sont statiques : ils affichent ce qui interagit mais pas ce qui se passe pendant l'interaction. [i]

En notation UML, une classe est représentée sous la forme d'un rectangle divisé en plusieurs parties : le nom de la classe, les attributs (champs), les opérations (méthodes) et autres.

La figure 9 illustre la forme d'une classe.

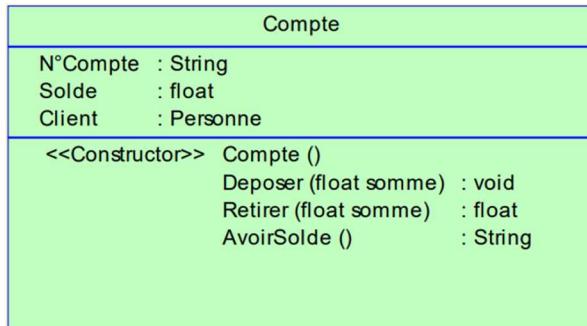


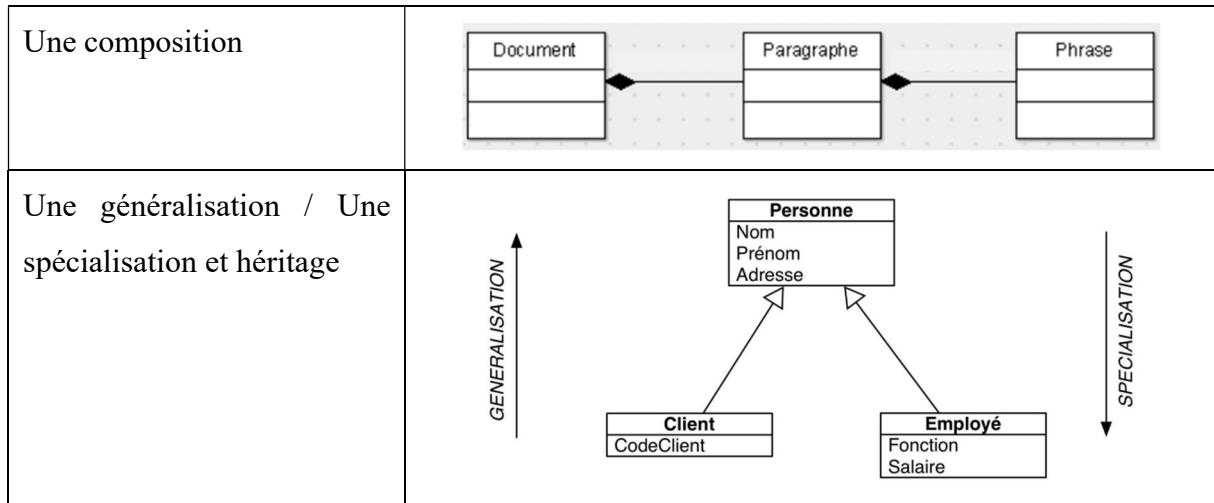
Figure 9: Illustration d'une classe

Les classes sont reliées entre elles par des associations. Cette association porte un nom et est représenté par une ligne rectiligne. Ceci dit, il y a plusieurs sortes d'associations entre les classes, et une association peut être réflexive.

Le tableau 23 résume les associations existantes.

Tableau 23: Type d'association et leur illustration

Type d'association	Illustration
Une simple association représentée par son nom	<p>A simple association diagram showing a line connecting two boxes labeled 'Client' and 'Compte'. Above the line, the text 'Est titulaire de' is written in bold black font.</p>
Une agrégation	<p>An aggregation diagram showing a line connecting two boxes labeled 'Pays' and 'Région'. The word 'Agrégat' is written above the 'Pays' box, and 'Agrégée' is written above the 'Région' box. A diamond symbol is placed on the line between the two boxes.</p>



Une association est présentée également avec sa multiplicité. Ce dernier est le nombre de participations d'une classe dans une association :

- indiquée à chaque extrémité d'une association
- sous la forme minimum..maximum
- les nombres acceptés sont 0, 1, * (sauf exception)

La figure 10 présente un exemple général d'un diagramme de classes : de deux classes avec leur association et ses multiplicités.



Figure 10: Exemple général d'un diagramme de classe

La figure 11 représente le diagramme de classe de conception du SPRINT 1 qui est sur la gestion des premiers paramètres de l'institut.

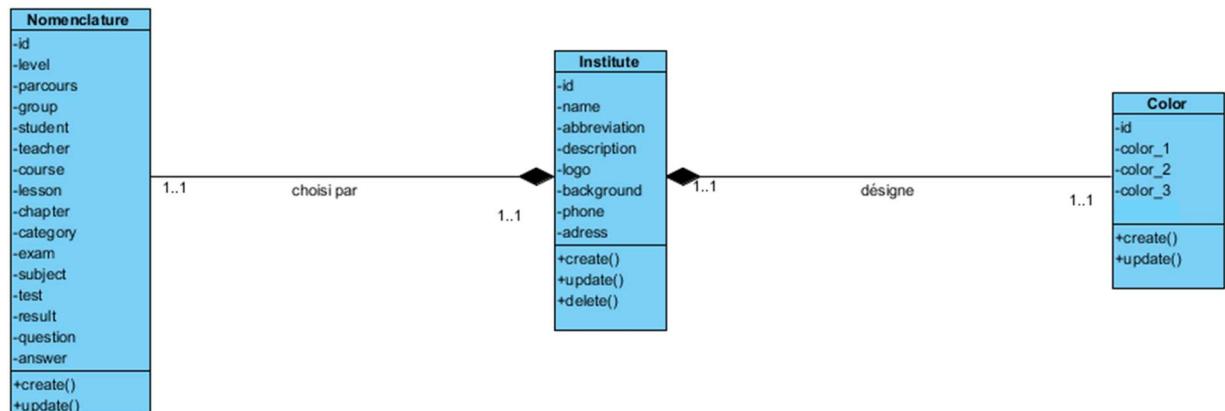


Figure 11: Diagramme de classe de conception du SPRINT 1

La figure 12 illustre le diagramme de classe de conception du SPRINT 2 qui est sur la gestion des autres paramètres de la plateforme et des classes des instituts.

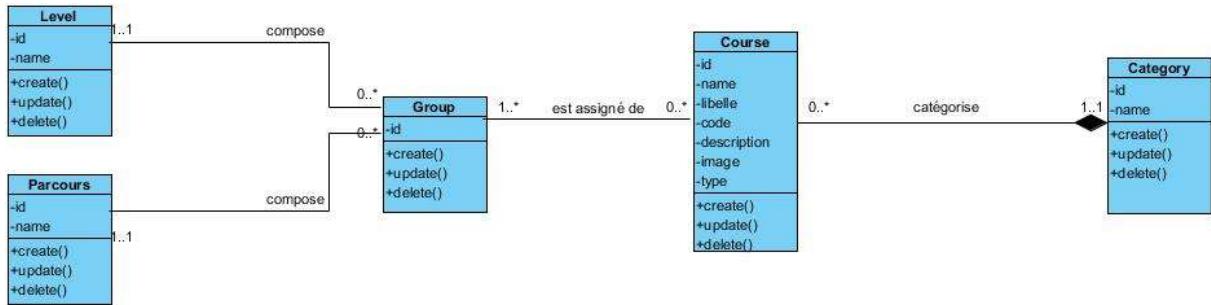


Figure 12: Diagramme de classe de conception du SPRINT 2

La figure 13 affiche le diagramme de classe de conception du SPRINT 3 qui concerne la gestion des utilisateurs de la plateforme.

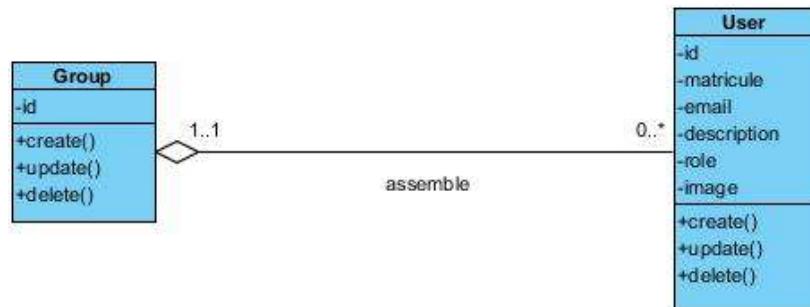


Figure 13: Diagramme de classe de conception du SPRINT 3

Le SPRINT 4 concerne la connexion vers la plateforme, nous n'avons pas besoins de diagramme spécifique. La figure 14 montre le diagramme de classe de conception du SPRINT 5 qui porte sur la gestion des cours et leurs composants.

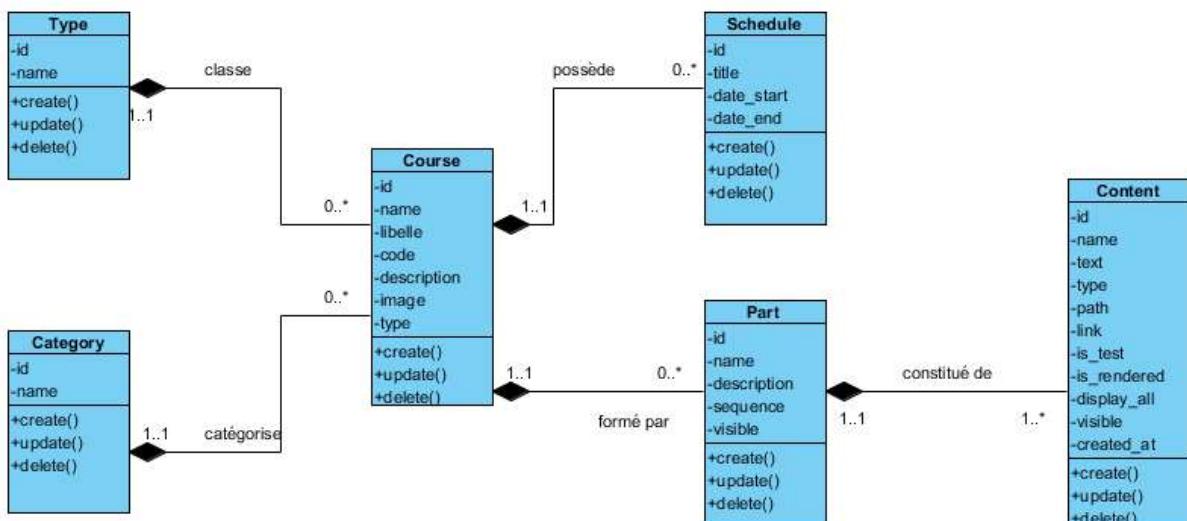


Figure 14: Diagramme de classe de conception du SPRINT 5

La figure 15 représente le diagramme de classe de conception du SPRINT 6 qui concerne les quizz.

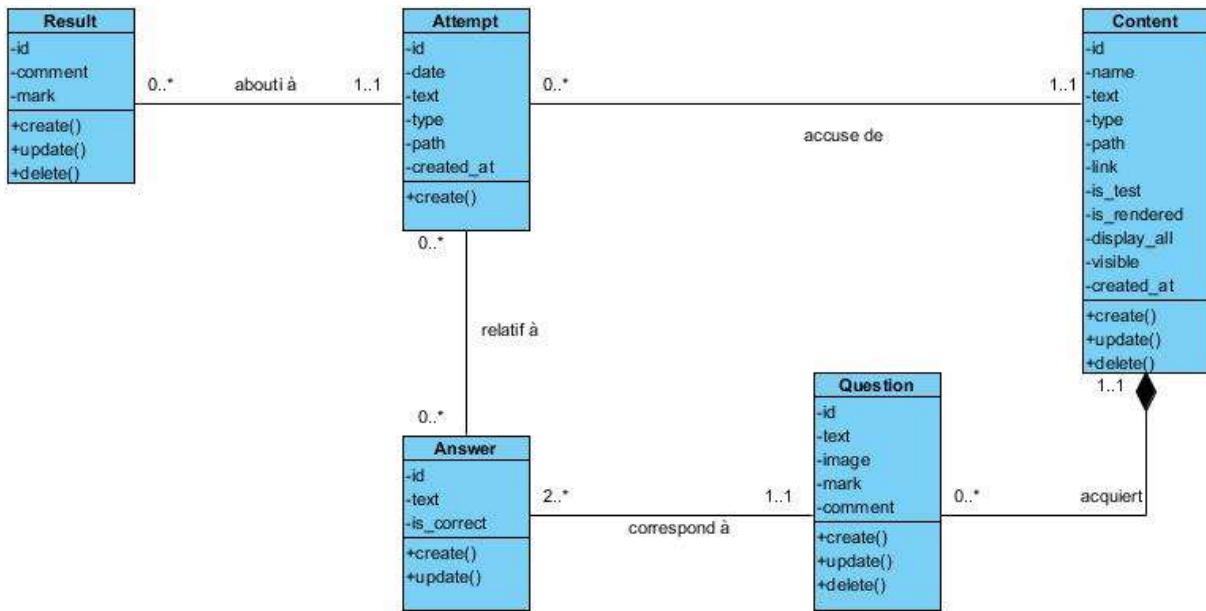


Figure 15: Diagramme de classe de conception du SPRINT 6

La figure 16 expose le diagramme de classe de conception du SPRINT 7 qui est sur les notifications.

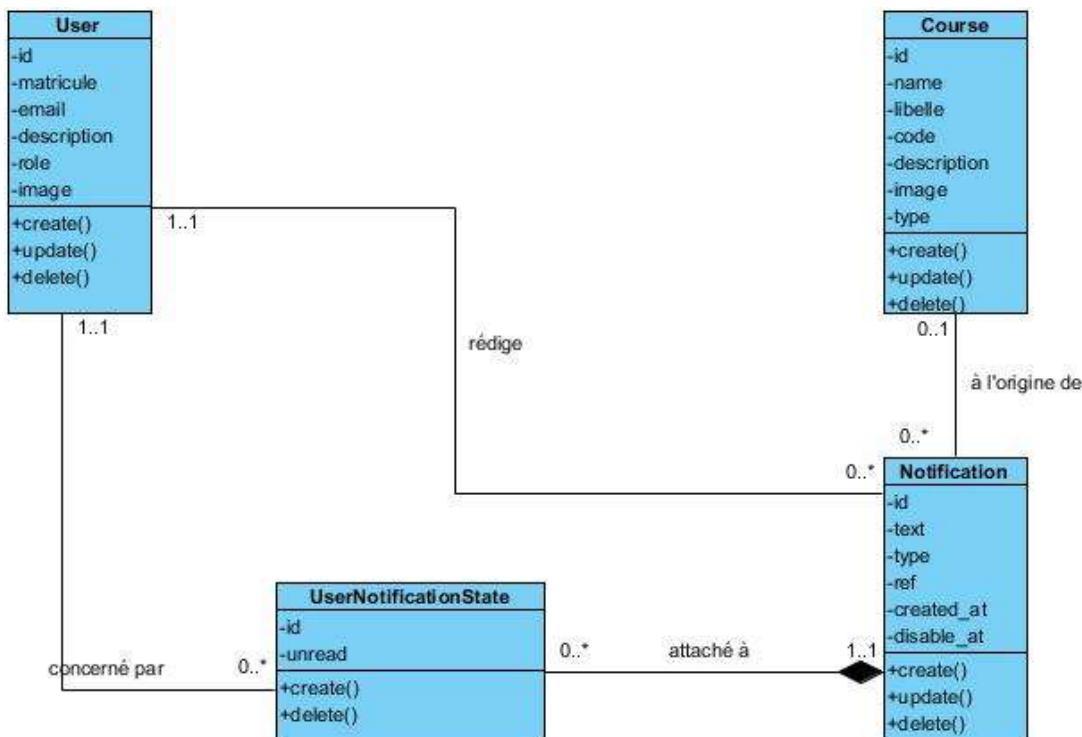


Figure 16: Diagramme de classe de conception du SPRINT 7

5.9. Diagramme de classe de conception globale

Les différentes étapes de cette partie de modélisation ont conduit à la synthèse de ce diagramme de classe de conception globale, illustré par la figure 17.

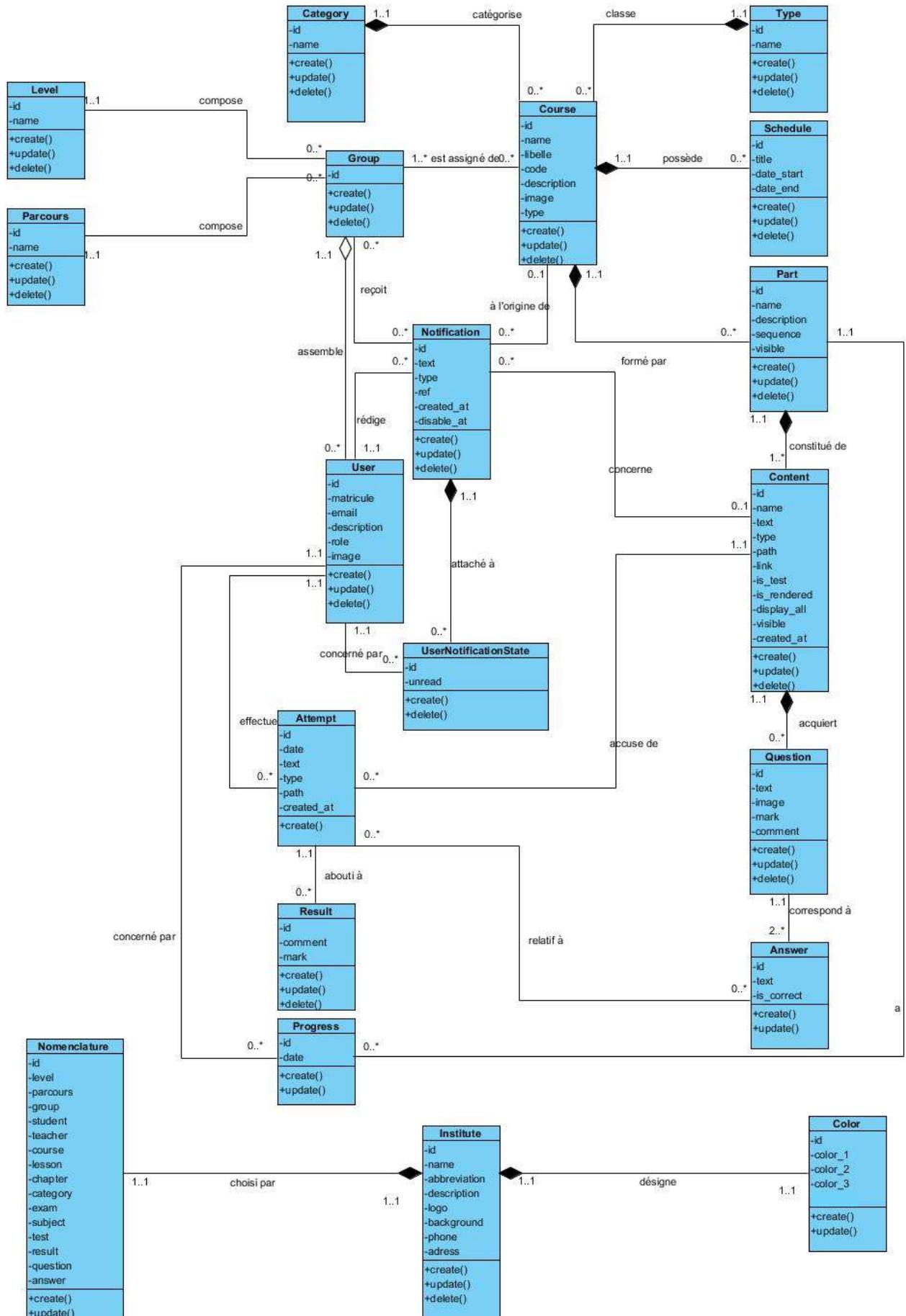


Figure 17: Diagramme de classe de conception globale

Chapitre 6 : Conception détaillé

6.1. Architecture globale du système

Le système adopte une architecture client/serveur.

La figure 18 représente cette architecture [5].

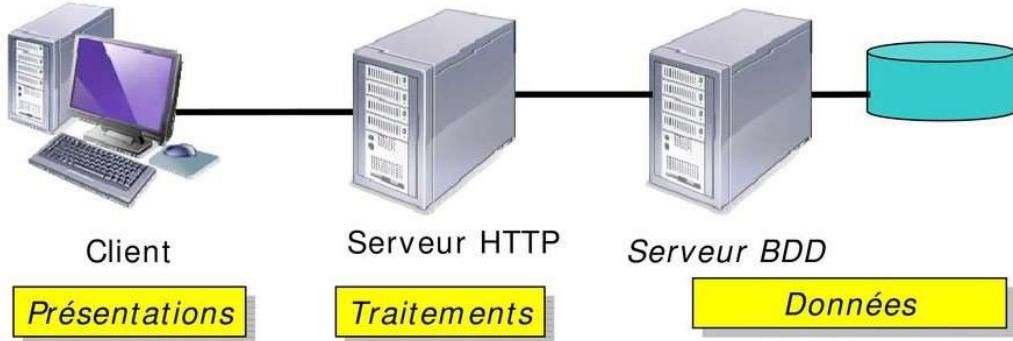


Figure 18: Architecture client/serveur

L'architecture client / serveur s'agit d'un modèle logique d'architecture applicative qui vise à modéliser une application comme un empilement de trois couches logicielles (ou niveaux, étages, tiers) dont le rôle est clairement défini :

- la présentation des données, correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur ;
- le traitement métier des données, correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative ;
- l'accès aux données persistantes : correspondant aux données qui sont destinées à être conservées sur la durée, voire de manière définitive.

Cette architecture a été choisi en vue de ses avantages dont le principal avantage est sa grande flexibilité :

- flexibilité des requêtes clients vers le serveur, qui permet à l'entreprise d'envisager l'introduction de toutes nouvelles technologies.
- flexibilité dans l'allocation des ressources, qui permet d'envisager une allocation et ou modification dynamique aux grés des besoins évolutifs au sein d'une entreprise.

6.2. Diagramme de séquence de conception

Un diagramme de séquence de conception est un type de diagramme d'interaction. Il représente les interactions entre objets en précisant la chronologie des échanges de message, montre sous forme de scénarios, la chronologie des envois de messages issus d'un cas d'utilisation.

Le diagramme de séquence fait ressortir :

- les acteurs
- les objets
- les messages

La figure 19 représente le formalisme d'un diagramme de séquence.

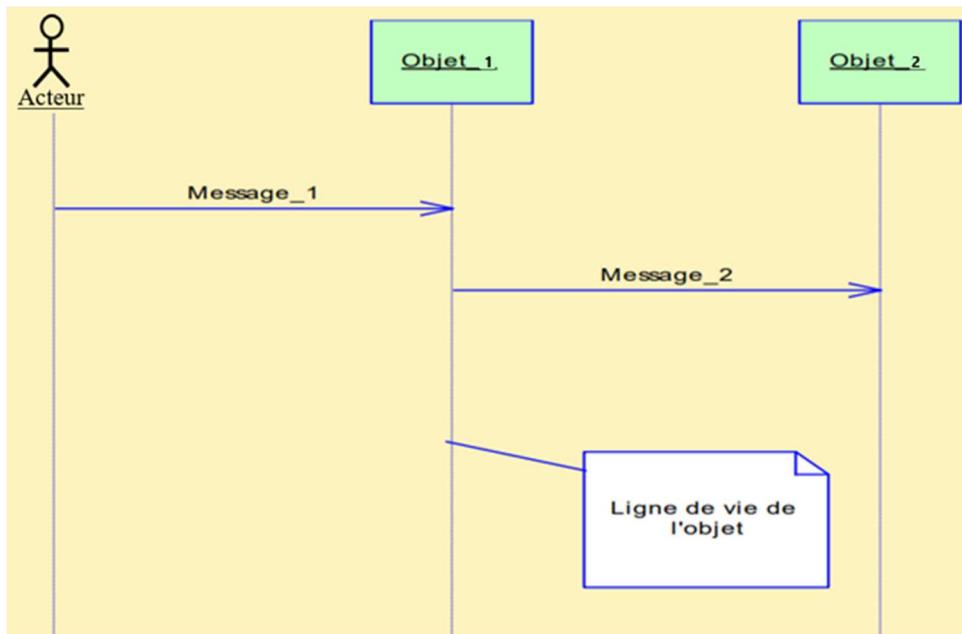


Figure 19: Formalisme d'un diagramme de séquence

- Un acteur est représenté par un dessin facilité d'humain. C'est un rôle joué par une entité externe qui agit sur le système (par exemple, les utilisateurs humains).
- Un objet est représenté par un rectangle et une ligne verticale (ligne de vie de l'objet). Les objets communiquent en échangeant des messages représentés par des flèches orientées de l'émetteur au récepteur.
- L'ordonnancement vertical des messages indique la chronologie.

La figure 20 présente le diagramme de séquence de conception pour l'authentification.

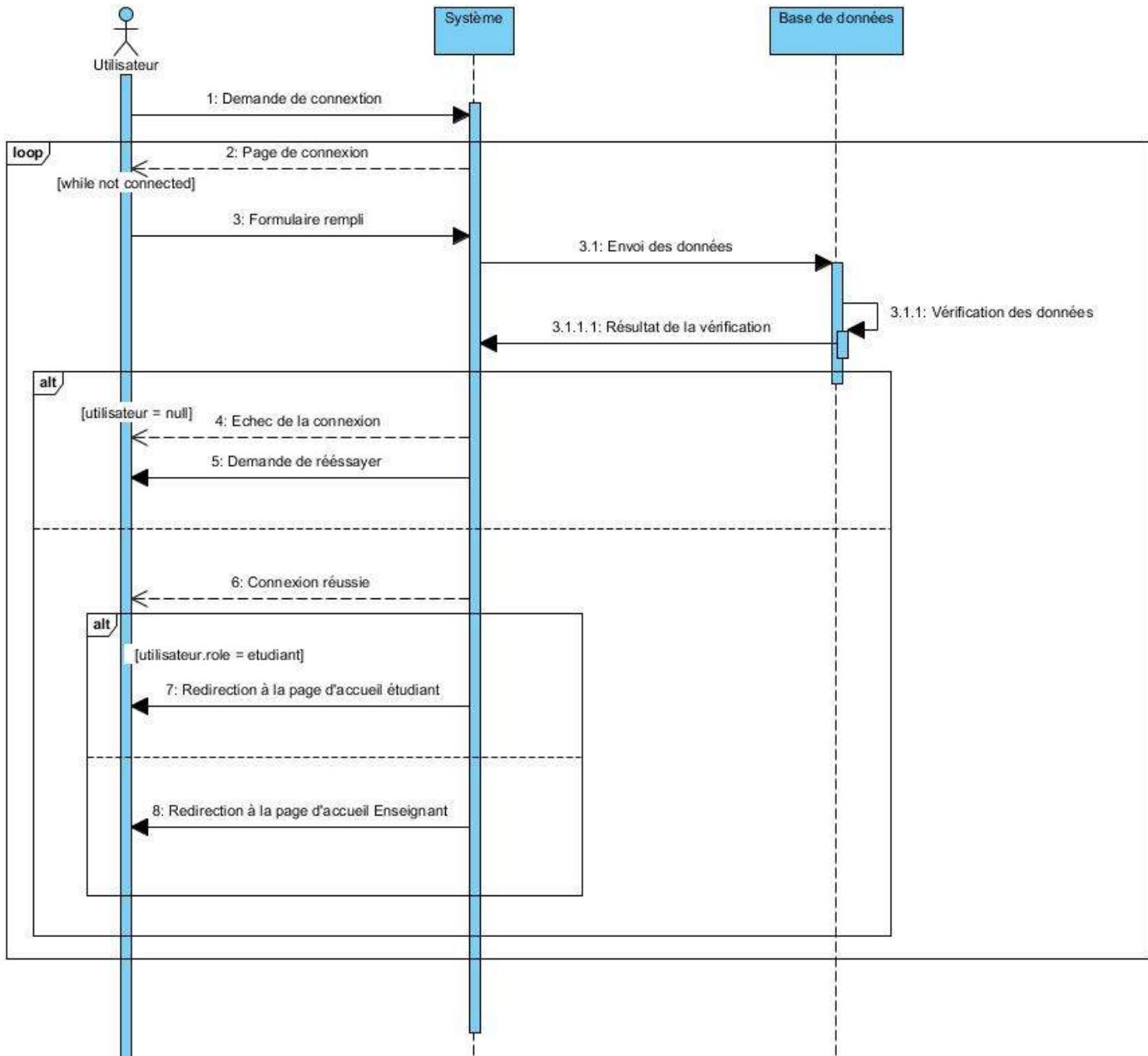


Figure 20: Diagramme de séquence de conception pour l'authentification

La figure 21 illustre le diagramme de séquence de conception pour la création d'un cours.

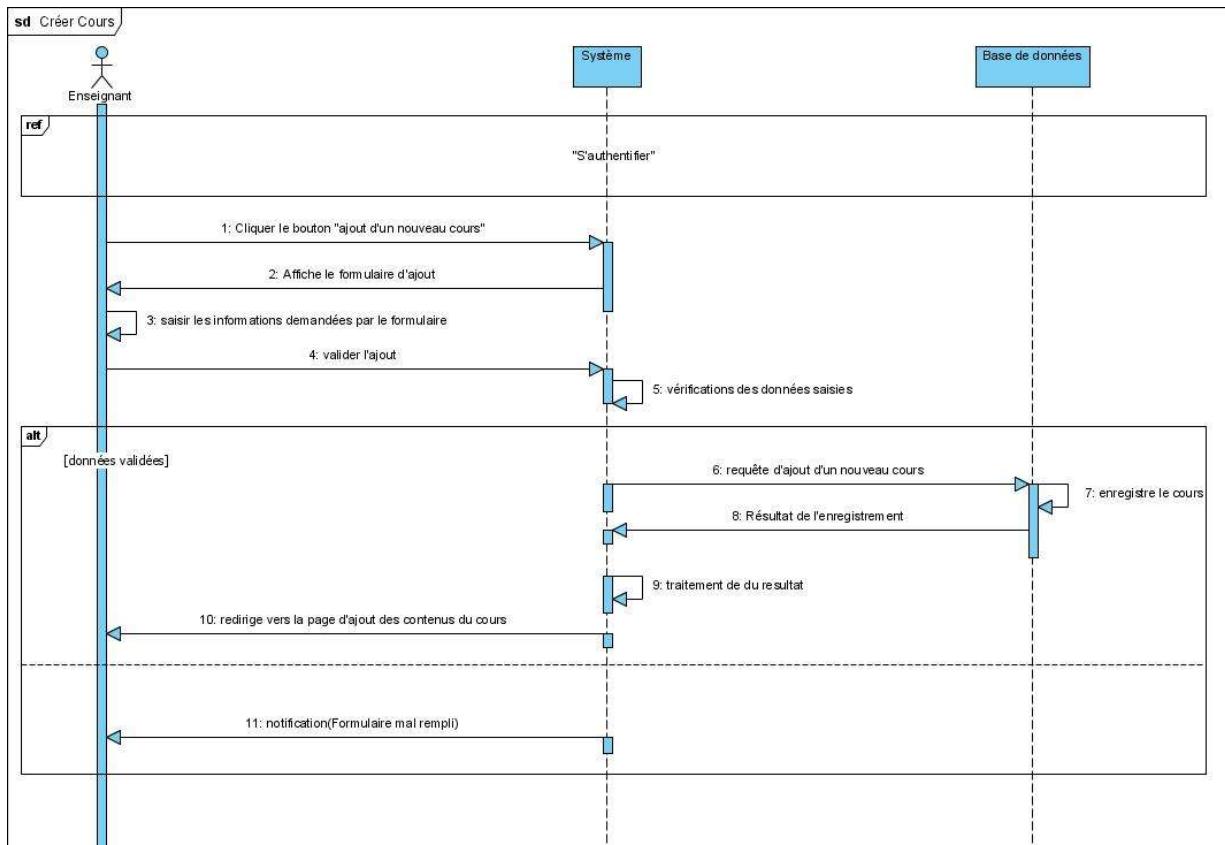


Figure 21: Diagramme de séquence sur la création d'un cours

La figure 22 montre le diagramme de séquence de conception pour la création d'une partie d'un cours.

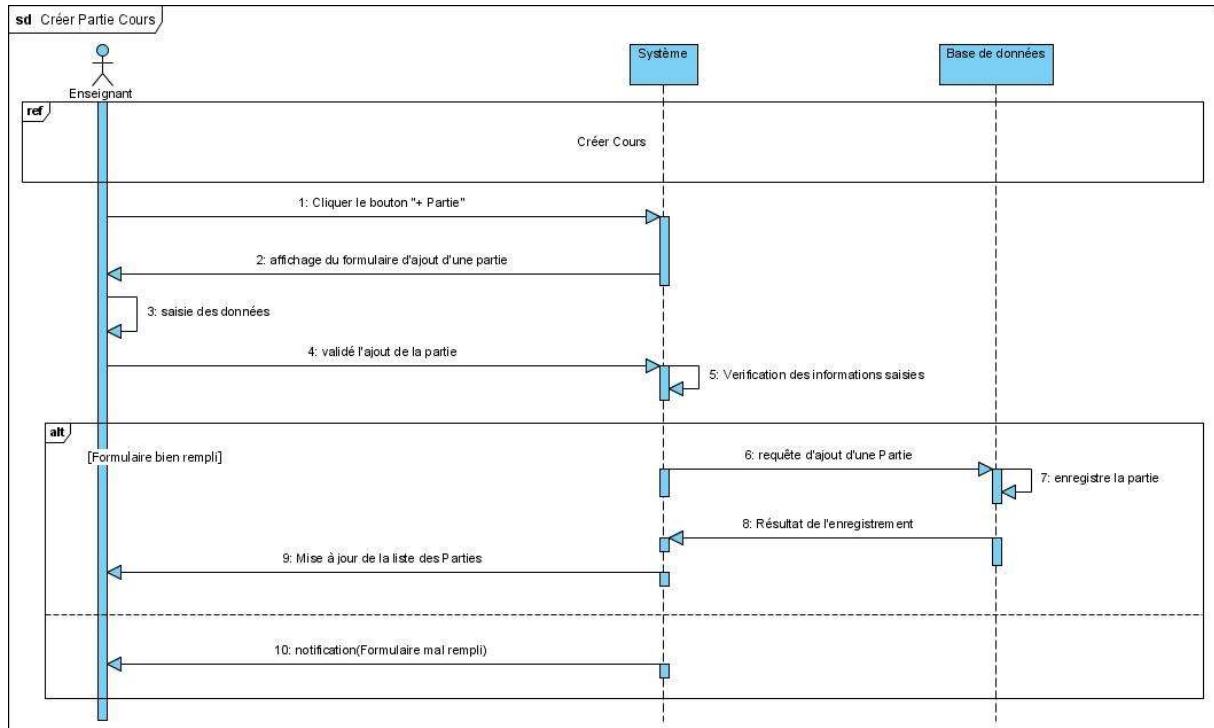


Figure 22: Diagramme de séquence sur la création d'une partie

La figure 23 présente le diagramme de séquence de conception pour la création d'un contenu de la partie.

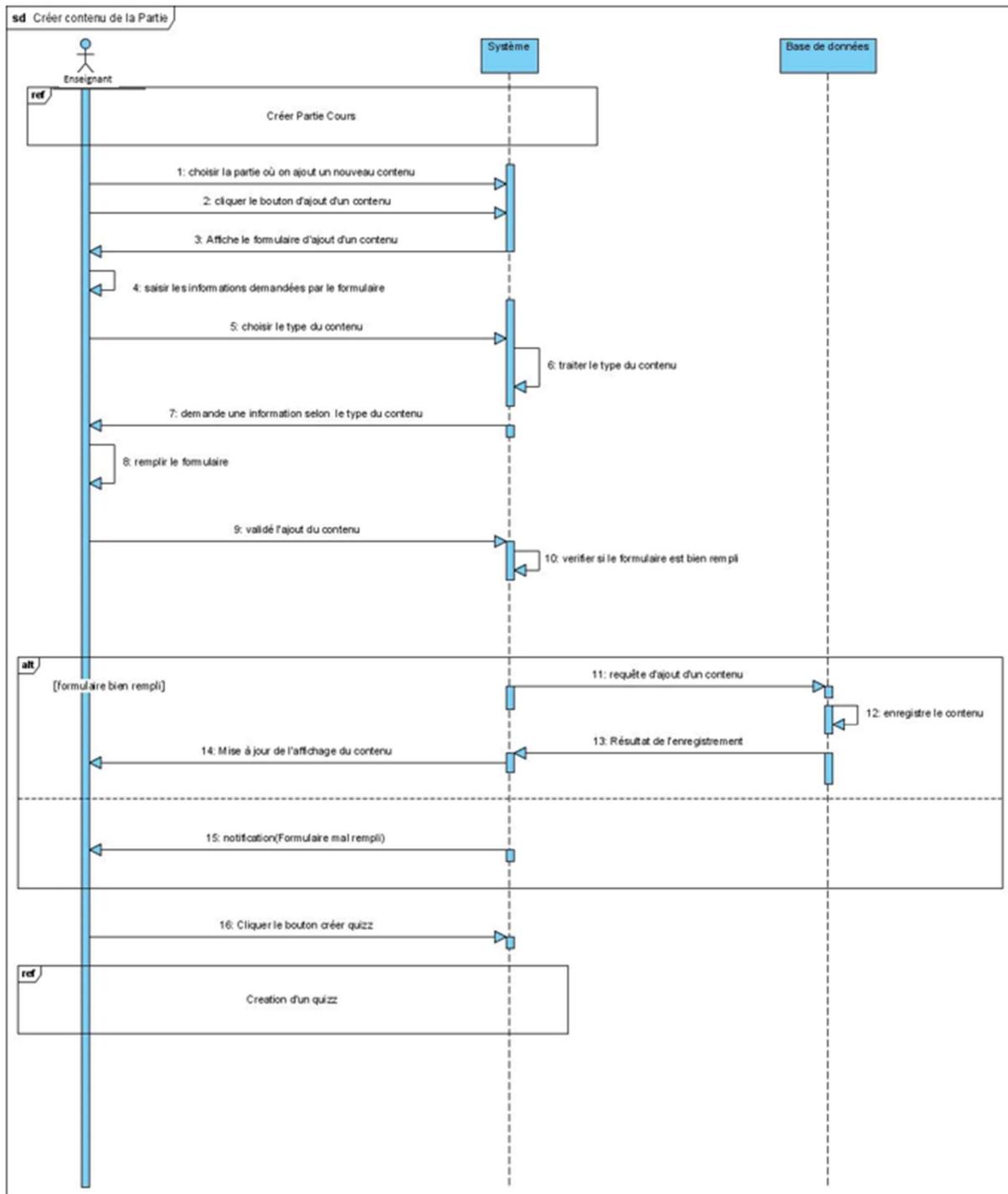


Figure 23: Diagramme de séquence de création d'un contenu

La figure 24 présente le diagramme de séquence de conception pour la consultation d'un cours par un utilisateur.

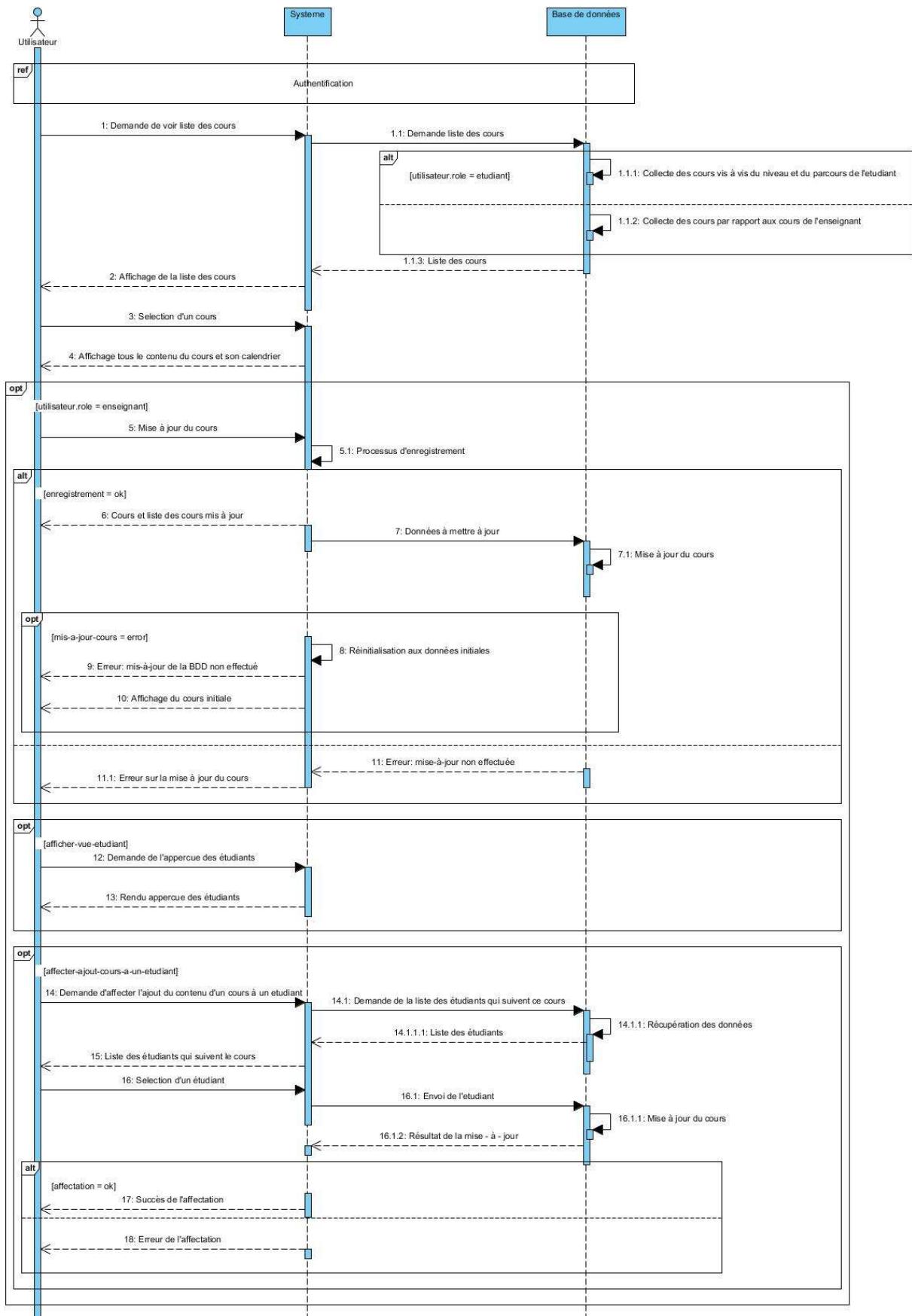


Figure 24: Diagramme de séquence pour la consultation de cours

La figure 25 expose le diagramme de séquence de conception pour la gestion de l'interface utilisateur.

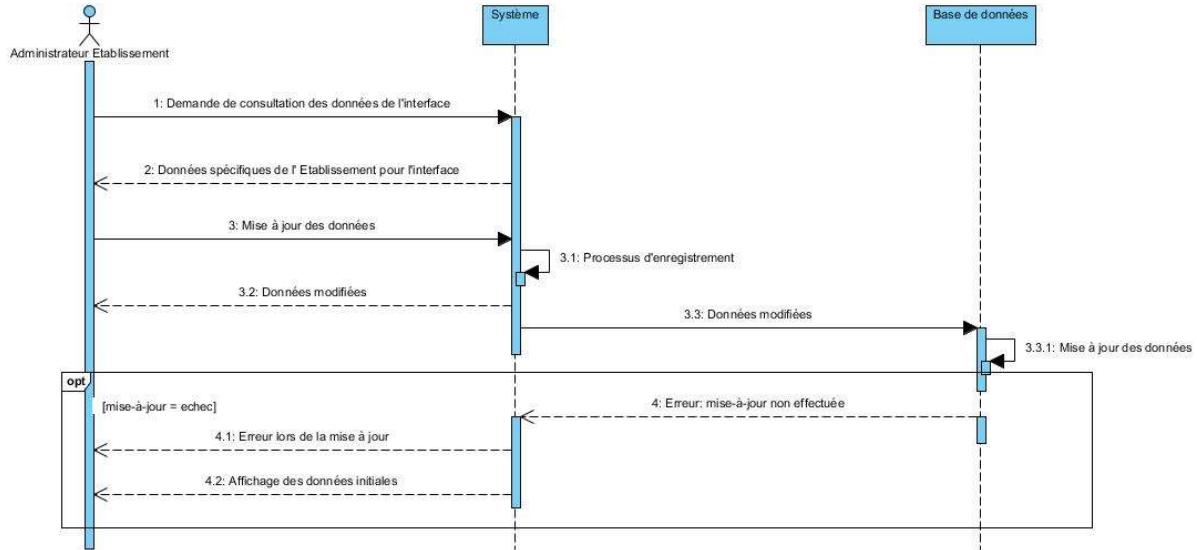


Figure 25: Diagramme de séquence pour la gestion de l'interface utilisateur

La figure 26 présente le diagramme de séquence de conception pour la gestion des catégories de cours.

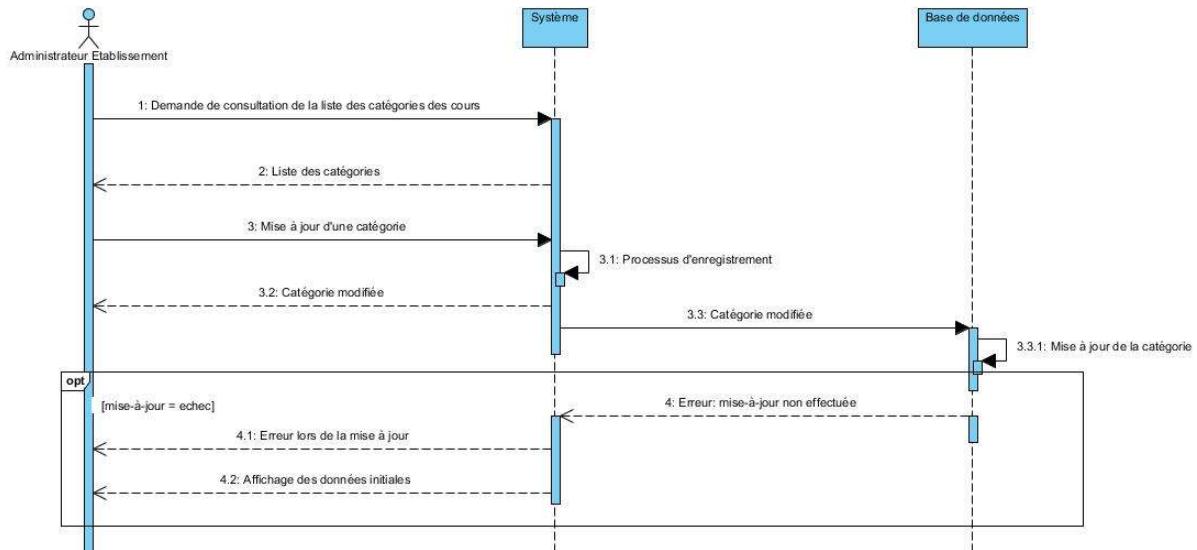


Figure 26: Diagramme de séquence pour la gestion des catégories

La figure 27 illustre le diagramme de séquence de conception pour la gestion des niveaux.

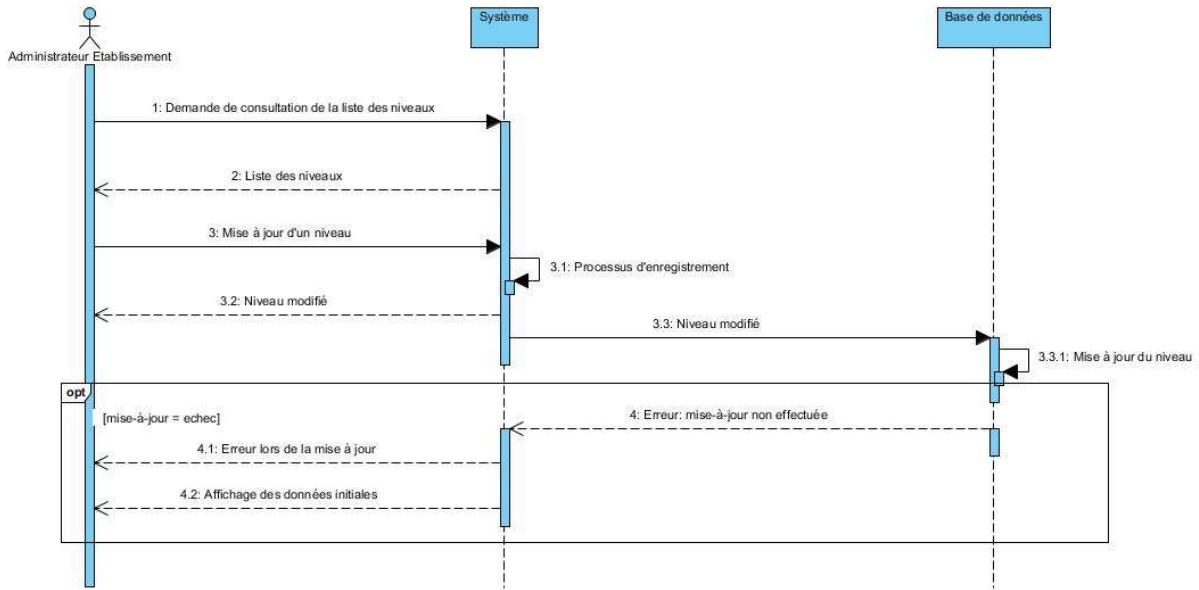


Figure 27: Diagramme de conception pour la gestion d'un niveau

La figure 28 montre le diagramme de séquence de conception pour la gestion des parcours.

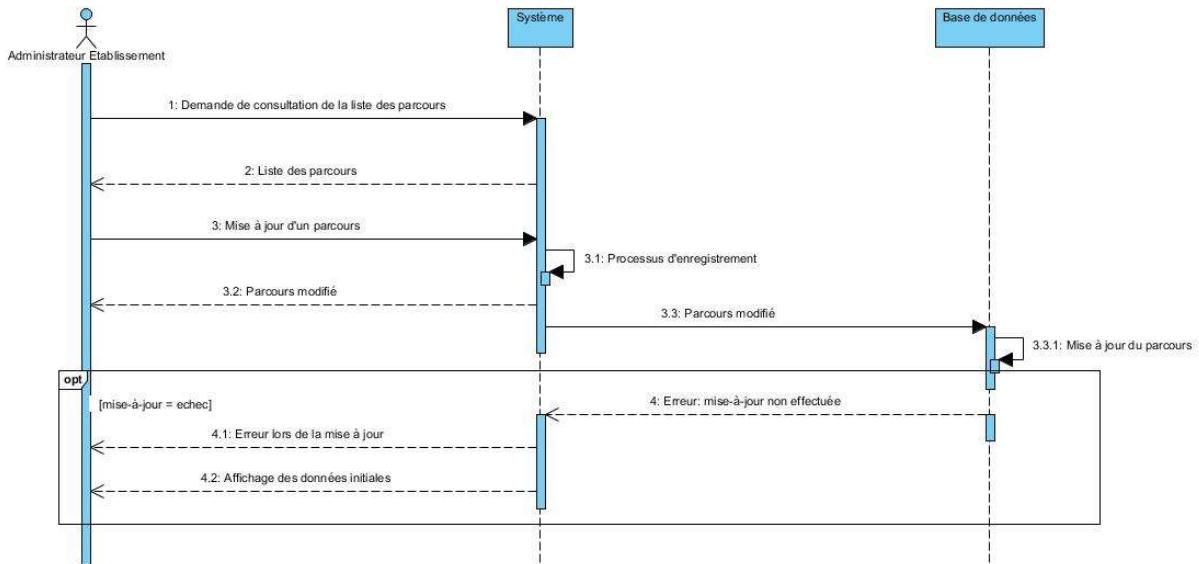


Figure 28: Diagramme de séquence de conception de la gestion des parcours

La figure 29 représente le diagramme de séquence de conception pour la gestion des types de cours.

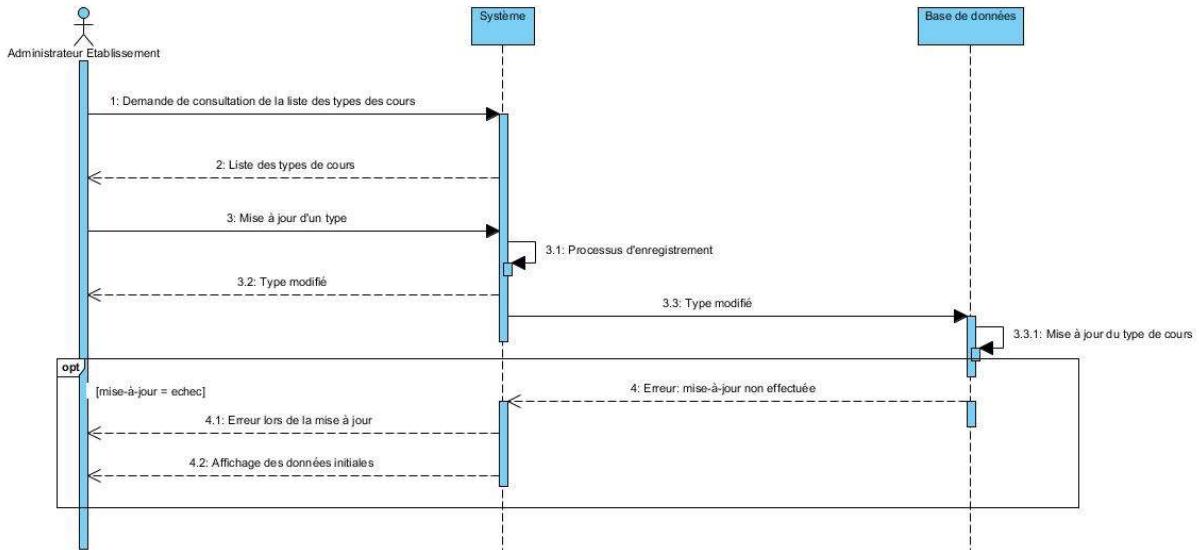


Figure 29: Diagramme de séquence de la gestion des types de cours

La figure 30 illustre le diagramme de séquence de conception pour la gestion des utilisateurs.

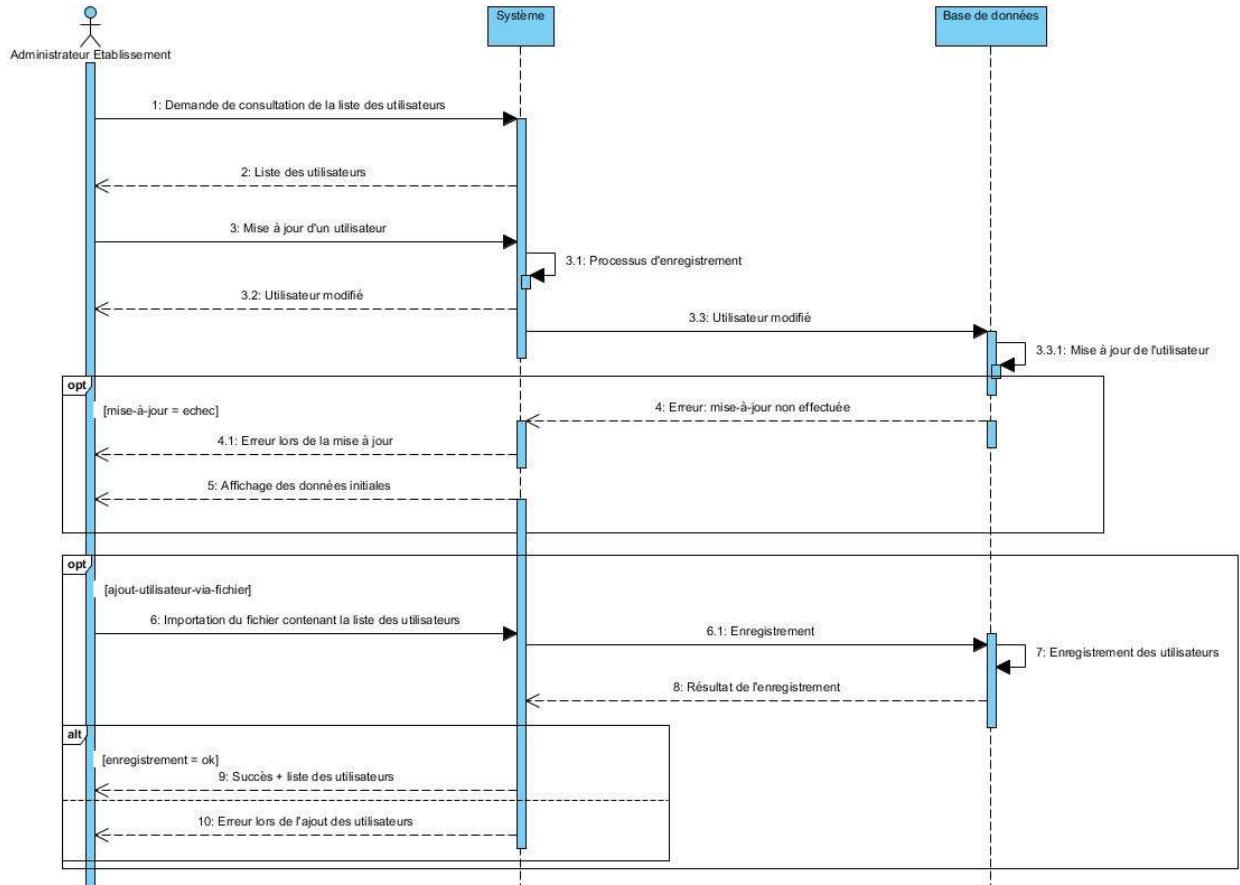


Figure 30: Diagramme de séquence de conception pour la gestion des utilisateurs

La figure 31 présente le diagramme de séquence de conception pour la modification d'un profil.

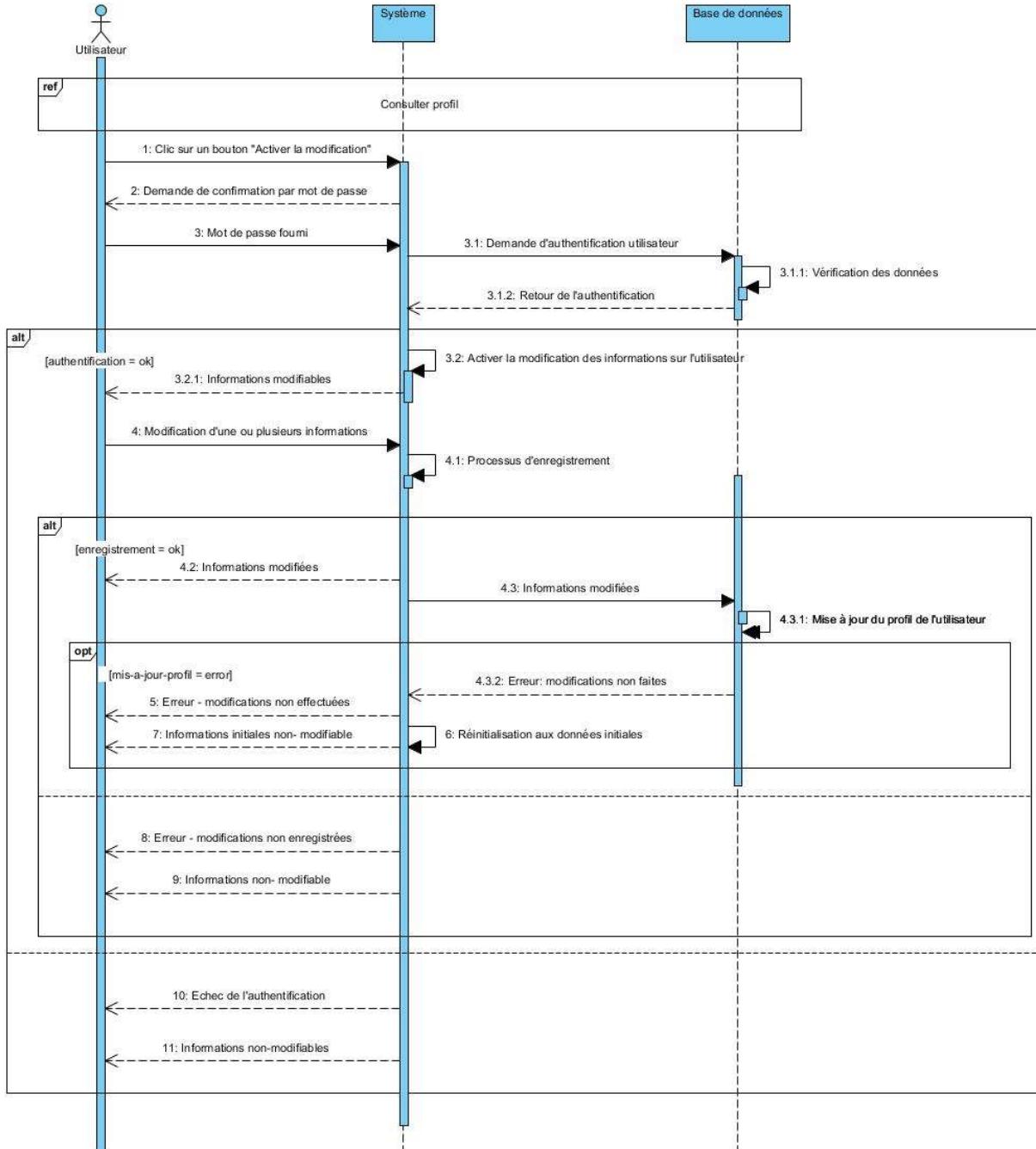


Figure 31: Diagramme de séquence de conception pour la modification du profil

La figure 32 expose le diagramme de séquence de conception pour la rédaction du quizz par un enseignant.

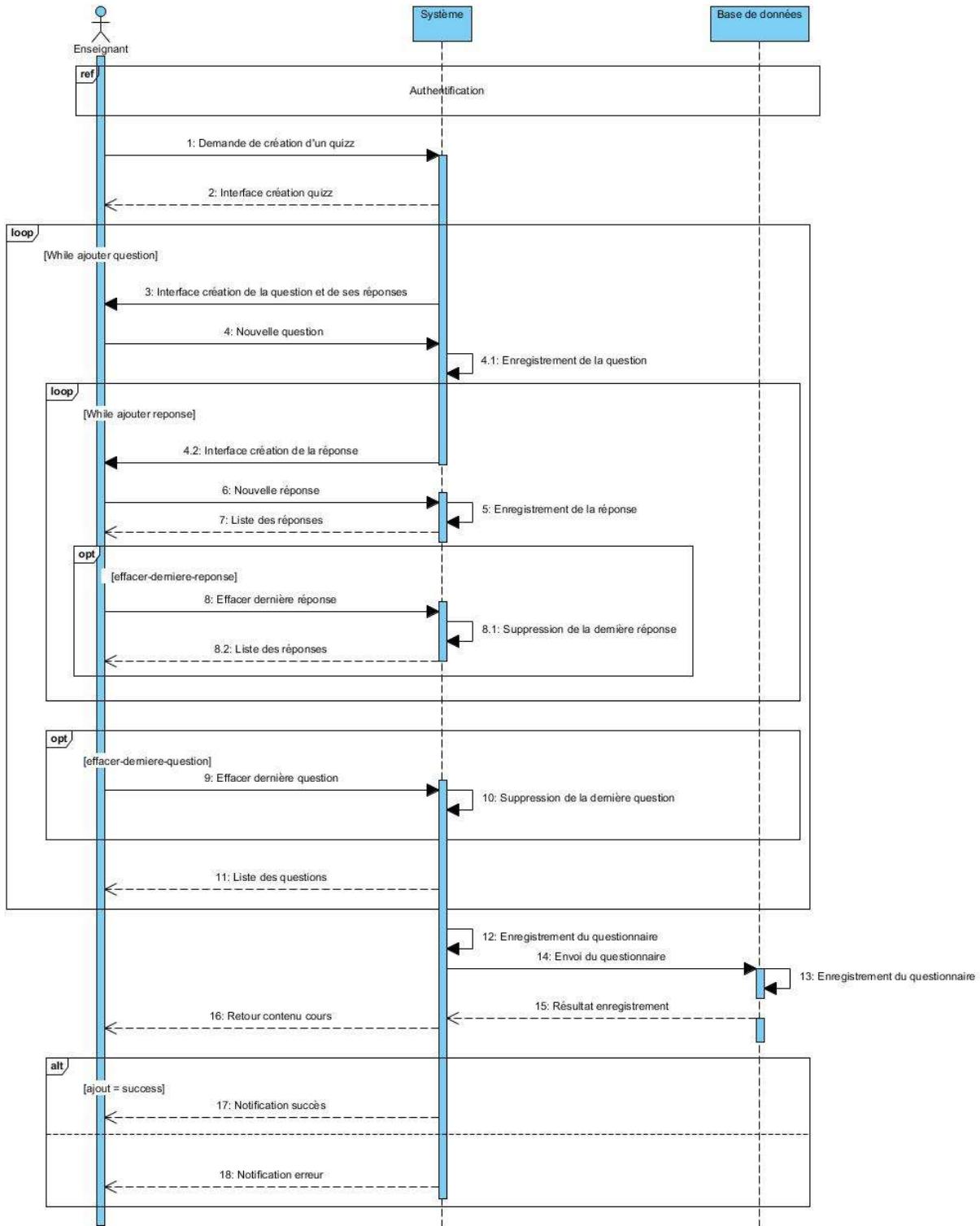


Figure 32: Diagramme de séquence de conception pour la rédaction d'un quizz

La figure 33 illustre le diagramme de séquence de conception pour l'effectuation d'un quizz par un étudiant.

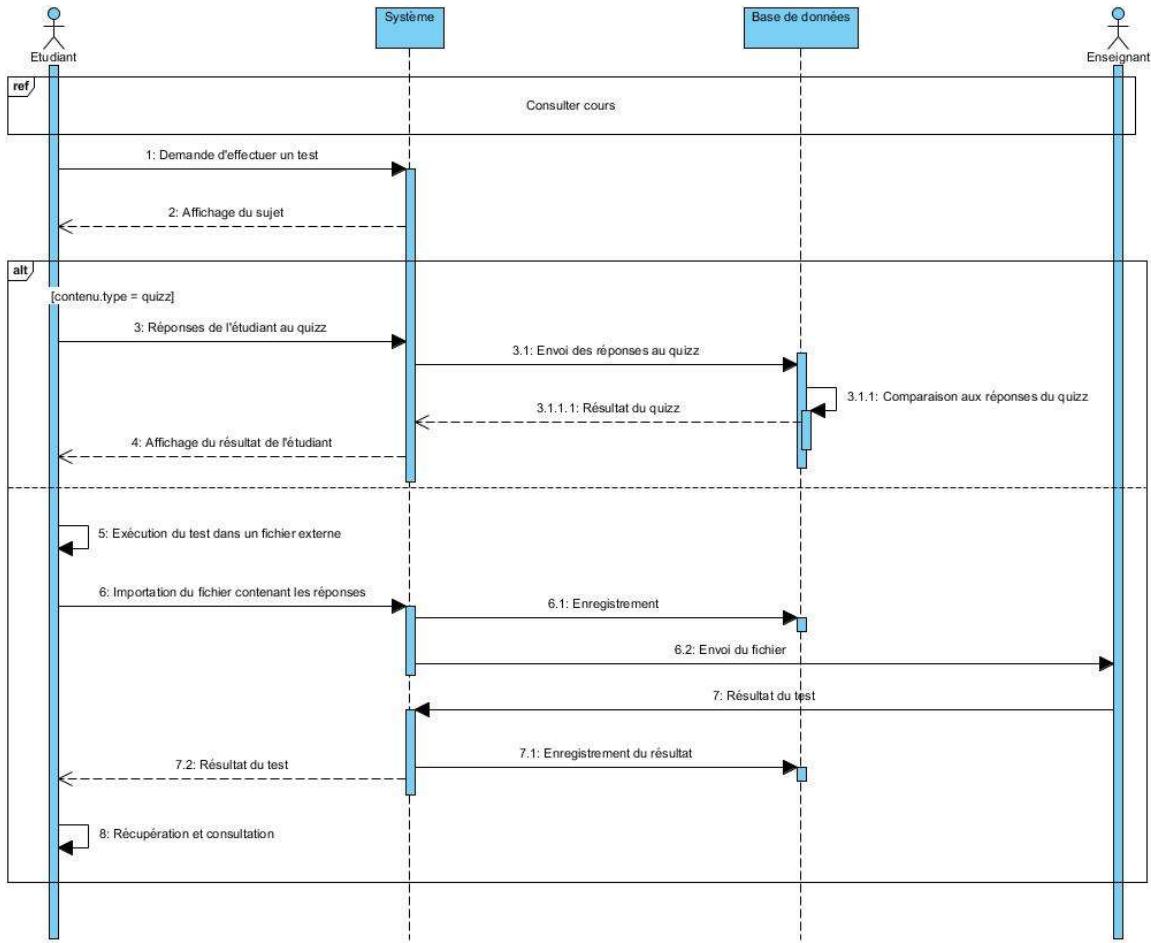


Figure 33: Diagramme de séquence de conception pour l'effectuation d'un quizz

6.1. Diagramme de paquetage

Un diagramme de paquetage est un diagramme structurel utilisé pour représenter l'organisation et la disposition de divers éléments du projet, modélisés sous forme de paquetage.

Le tableau 24 représente les composants de base d'un diagramme de paquetage.

Tableau 24: Composants de base d'un diagramme de paquetage

Symbole	Nom	Description
	Package	Regroupe des éléments apparentés en fonction des données, du comportement ou interaction avec les utilisateurs
	Interdépendance	Décrit la relation entre un élément et un autre

La figure 34 présente le diagramme de paquetage du projet.

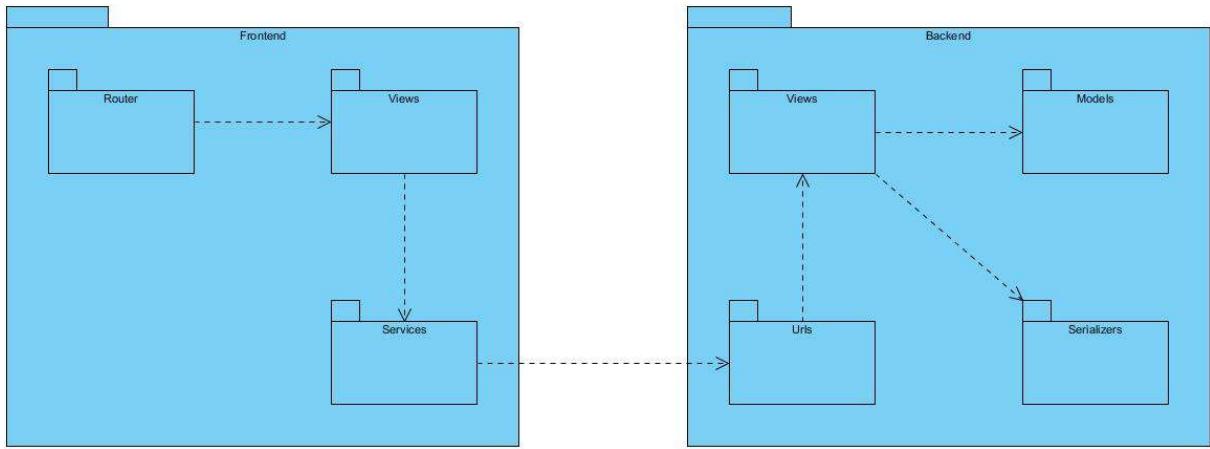


Figure 34: Diagramme de paquetage du projet

6.2. Diagramme de déploiement

Le diagramme de déploiement décrit le déploiement physique des informations générées par le logiciel sur des composants matériels.

Le tableau 25 représente les composants de base d'un diagramme de déploiement.

Tableau 25: Composant de base d'un diagramme de déploiement

Symbole	Nom	Description
	Noeud	Elément matériel ou logiciel
	Composant	Elément logiciel
	Association	Message ou tout autre type de communication entre deux nœuds
	Interface fournie et interface requise	Utilisation pour montrer qu'un composant fournit un service que l'autre demande.

La figure 35 présente le diagramme de déploiement de la plateforme.

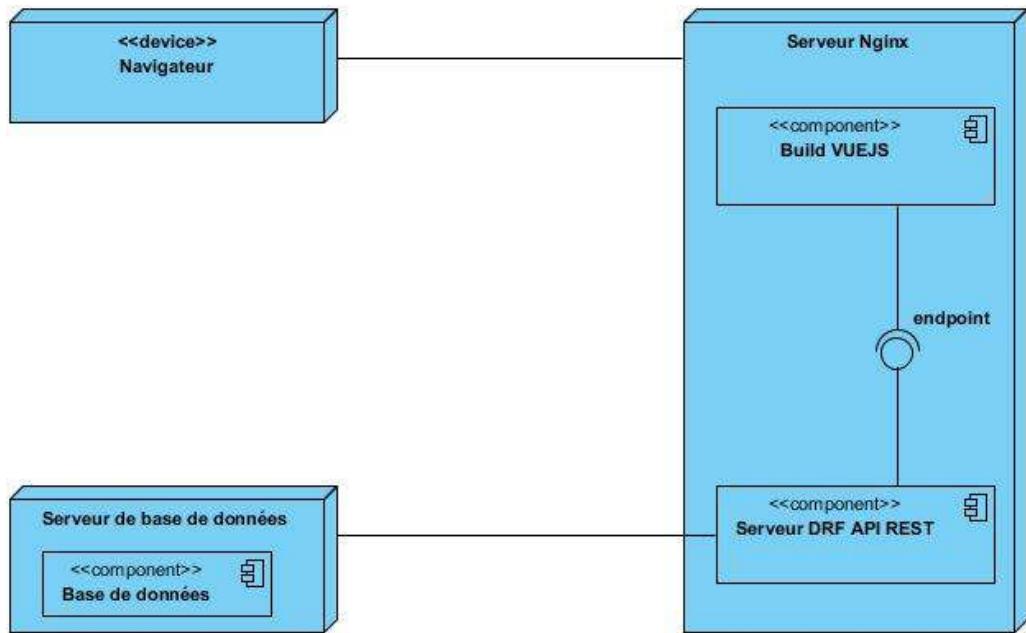


Figure 35: Diagramme de déploiement de la plateforme

Chapitre 7 : Mise en place de l'environnement

7.1. Installation et mise en place des outils

Dans cette section, nous allons nous concentrer sur les installations et configurations nécessaires afin de mener à bien la réalisation du projet.

7.1.1. Installation de Visual Paradigm

Visual Paradigm est un outil de développement logiciel qui aide les utilisateurs à créer des diagrammes UML, des organigrammes et d'autres types de diagrammes. Il est disponible en version gratuite et payante. La version gratuite est limitée en termes de caractéristiques et de fonctionnalités, mais elle reste un outil utile pour les petits projets. La version payante est plus complète et comprend des fonctions telles que la prise en charge de la rétro-ingénierie, la génération de code, etc.

Pour installer Visual Paradigm for UML8.0, il faut lancer le programme d'installation VP_suite 5.0. La figure 36 affiche la fenêtre de bienvenue.

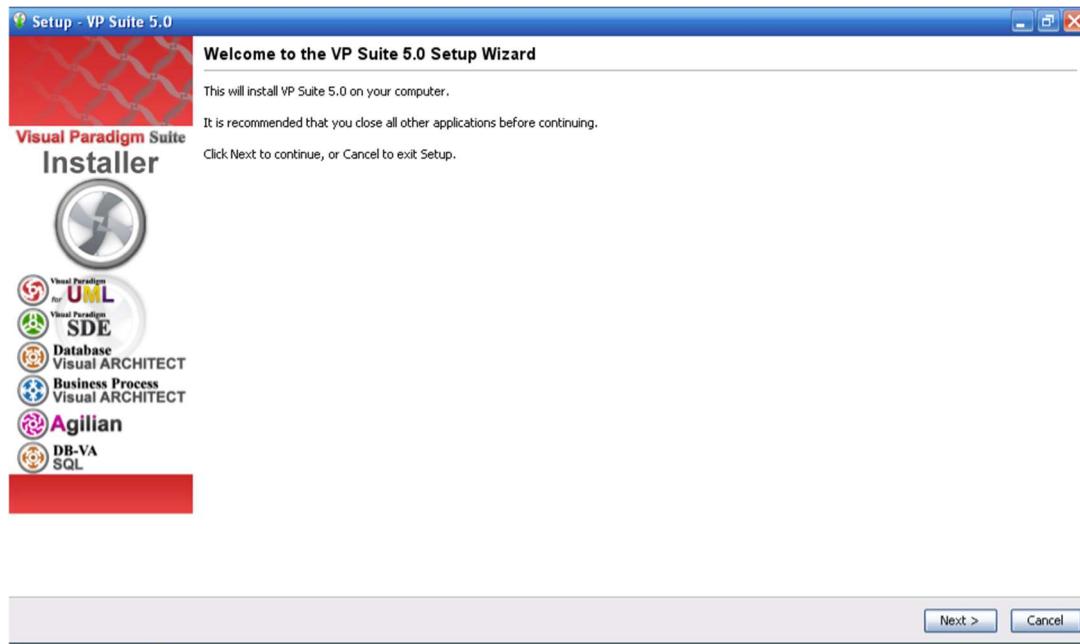


Figure 36: Page d'accueil pour l'installation de Visual Paradigm

Il suffit de lire et suivre attentivement les instructions et les configurations pour installer notre outil de modélisation graphique.

La figure 37 montre la dernière étape qui affiche la confirmation de l'installation.

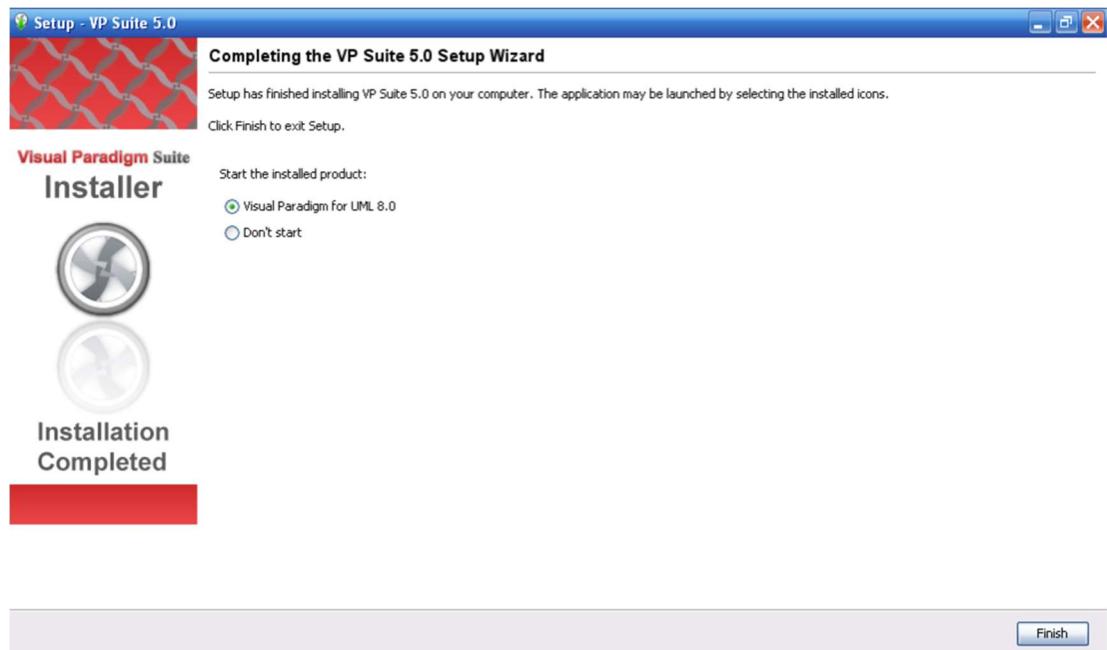


Figure 37: Fin de l'installation de Visual Paradigm

7.1.2. Installation de Visual Studio Code

Le téléchargement du logiciel se fait sur son site officiel et l'installation consiste à suivre les indications proposées. Puis, afin d'améliorer l'environnement de développement, certaines extensions ont été installées :

- « Python » pour le développement du backend
- « Vue Language Features (Volar) » pour le développement du frontend

La figure 38 montre le début de l'installation de Visual Studio Code

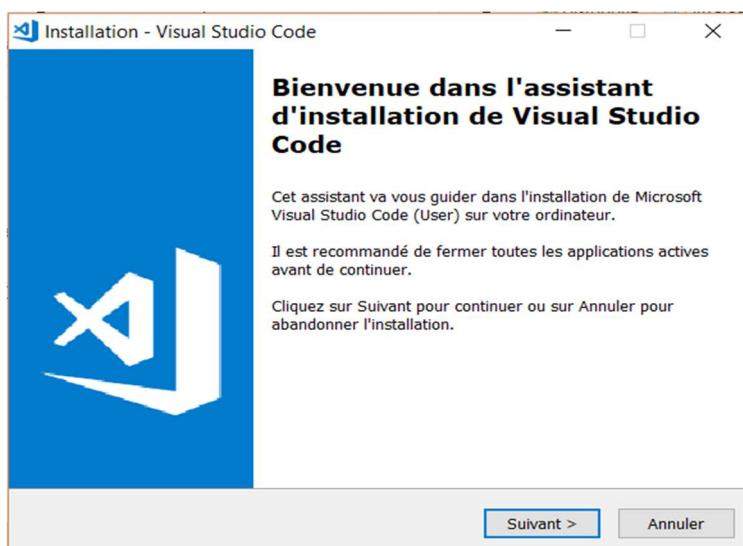


Figure 38: Début de l'installation de Visual Studio Code

Pour l'installation, il suffit de lire attentivement les directives de l'assistant d'installation et de choisir les configurations selon nos besoins.

La figure 39 présente la fin de l'installation de Visual Studio Code.



Figure 39: Fin de l'installation de Visual Studio Code

7.1.3. Installation de git

Pour débuter, il faut créer un compte Github sur son site officiel <https://github.com/>. Ensuite, nous devons installer Git sur la machine. Pour cela, il suffit de télécharger le fichier .exe d'installation depuis la page officielle <https://gitforwindows.org/>, double cliquer sur l'icone du fichier téléchargé et page après page, configurer l'installation de Git. Puis, nous pouvons le vérifier si l'installation s'est bien déroulée en demandant la version de Git installée. Pour cela, il faut lancer le terminal « git bash », installé en même tant que Git. Une fois le terminal lancé, il ne nous reste plus qu'à taper la commande «git version».

La figure 40 illustre une installation effectuée avec succès.

A screenshot of a terminal window titled "MINGW64:/c/Users/rojon". The command \$ git version is entered and the output is "git version 2.28.0.windows.1".

Figure 40: Vérification de la version de Git installée

Enfin, nous créons deux repository sur Github, l'un pour le frontend et l'un sur le backend.

7.1.4. Installation de Vuejs

Pour installer vue sur Ubuntu, nous allons tout d'abord devoir installer npm. Cela se fait via les commandes :

```
sudo apt-get update  
sudo apt-get install nodejs npm
```

Puis, installer vue-cli avec la commande :

```
npm install -g vue-cli
```

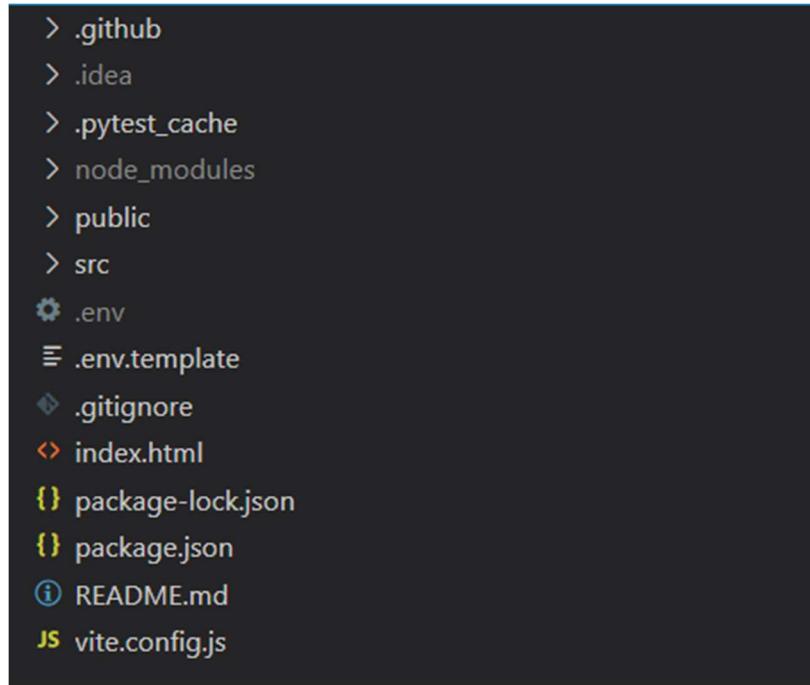
Enfin, installer vue avec la commande :

```
npm install vue
```

La création d'un projet VueJS se fait par la commande :

```
vue create its-frontend
```

La figure 41 présente l'arborescence de notre frontend.



```
> .github
> .idea
> .pytest_cache
> node_modules
> public
> src
⚙️ .env
☰ .env.template
❖ .gitignore
✖ index.html
{} package-lock.json
{} package.json
ⓘ README.md
JS vite.config.js
```

Figure 41: Structure du dossier Frontend

- Le dossier .github comporte les fichiers de notre CI / CD.
- .pytest_cache contient les caches des derniers tests fonctionnalités réalisés dans les tests de bout en bout avec Playwright.

- Le dossier `node_modules` stoque tous les modules nécessaires au projet.
- Le dossier `public` contient tout le contenu qui sera directement mis à la racine du serveur web, sans être passé par Webpack.
- Le fichier `.env` renferme tous les variables d'environnement comme présenté dans le fichier `.env.template`.
- Le fichier `package.json` contient les informations de package NPM du projet (version, dépendances, scripts etc.).
- Le fichier `vite.config.js` est le fichier de configuration pour Vite sur ce projet.
- Le fichier `index.html` sera le fichier destiné pour la production.
- Le dossier `src` va contenir notre code VUEJS, c'est nous qui définissons comment se structure notre application au sein de ce dossier.

Pour lancer le frontend, nous utilisons la commande :

```
npm run dev
```

7.1.5. Installation de Playwright

Pour les tests de bout en bout, nous utilisons la librairie Playwright avec le langage python. Pour installer le plugin de Pytest, nous lançons la commande :

```
pip install pytest-playwright
```

Pour installer les navigateurs requis, nous utilisons la commande :

```
playwright install
```

7.1.6. Installation de Django Rest Framework

Pour la création du Backend avec Django Rest Framework, nous allons créer un fichier qui contiendra toutes ses dépendances. Ensuite, pour les installer, nous devons suivre les commandes :

```
sudo apt-get update
sudo apt install python3-pip
pip install -r requirements.txt
```

La figure 42 représente l'arborescence du dossier du backend créé.

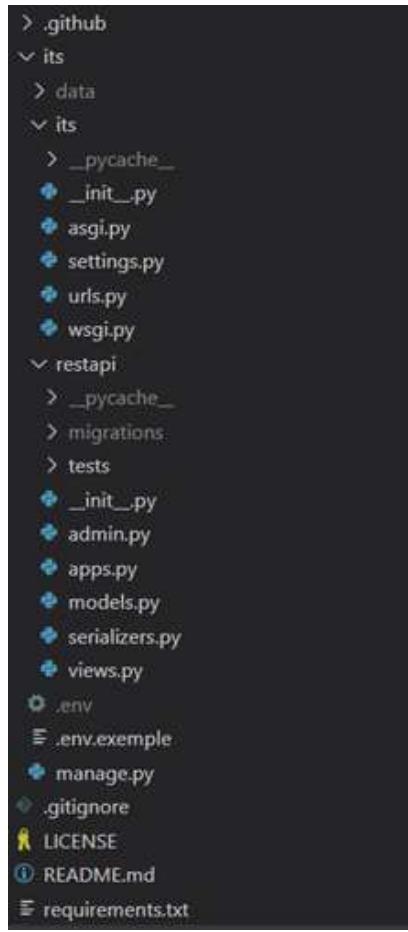


Figure 42: Arborescence du dossier du backend

- Le dossier .github contient le fichier pour le déploiement continu.
- Le dossier its/data comporte les fichiers correspondants aux enregistrements dans la base de données
- Le dossier restapi/tests contient les fichiers pour les tests unitaires du backend
- Les autres fichiers sont des fichiers préalablement créés et remplis suivant le template d'un projet DRF, il nous suffit de coder dedans en codant nos besoins.

Nous entrons dans le dossier du projet créé. Nous y créons un fichier .env pour contenir nos variables d'environnements. Pour effectuer la migration de des bases de données de Django, nous exécutons les commandes :

```
python manage.py makemigrations
python manage.py migrate
```

Enfin, nous lançons le backend avec la commande :

```
python manage.py runserver
```

7.1.7. Installation de HeidiSQL

HeidiSQL est un logiciel libre, et a pour objectif d'être facile à apprendre. "Heidi" nous permet de voir et de modifier des données et des structures à partir d'ordinateurs exécutant notre système de base de données MariaDB.

Dans un premier temps, nous devons le télécharger depuis le site <https://www.heidisql.com/download.php>. Dans un second temps, nous devons l'installer juste avec un double clic et en suivant soigneusement l'assistant d'installation du logiciel.

La figure 43 présente l'interface de HeidiSQL une fois l'installation achevée avec succès.

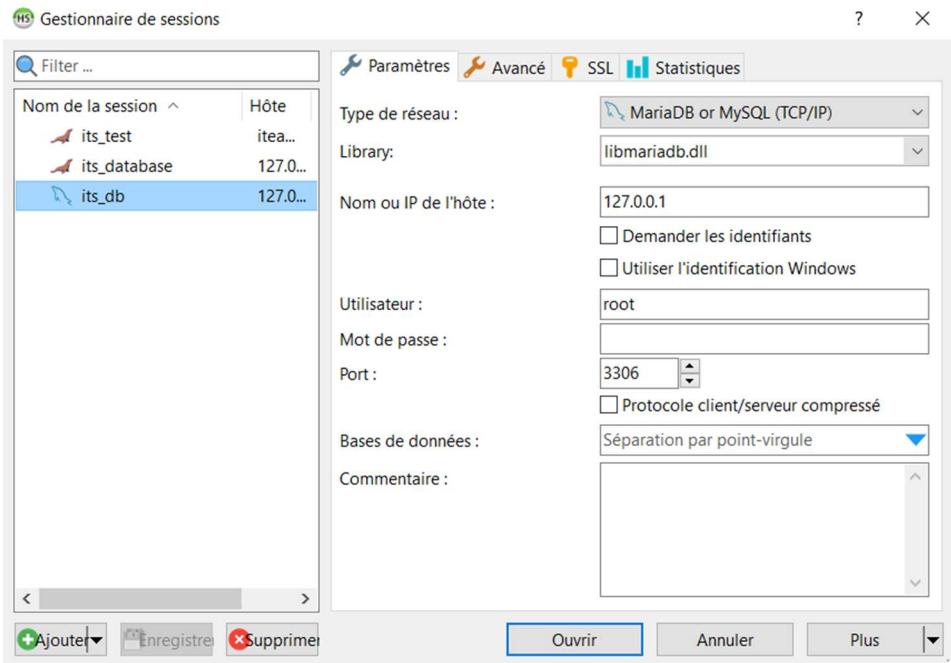


Figure 43: Interface du logiciel HeidiSQL

7.2. Architecture de l'application

Pour la mise en œuvre du projet ITS, nous avons choisi de découpler la plateforme. On parle alors du frontend : la partie visible avec laquelle le visiteur interagit ; et le backend, la partie qui est reliée à la base de données.

La figure 44 illustre l'architecture de la plateforme.

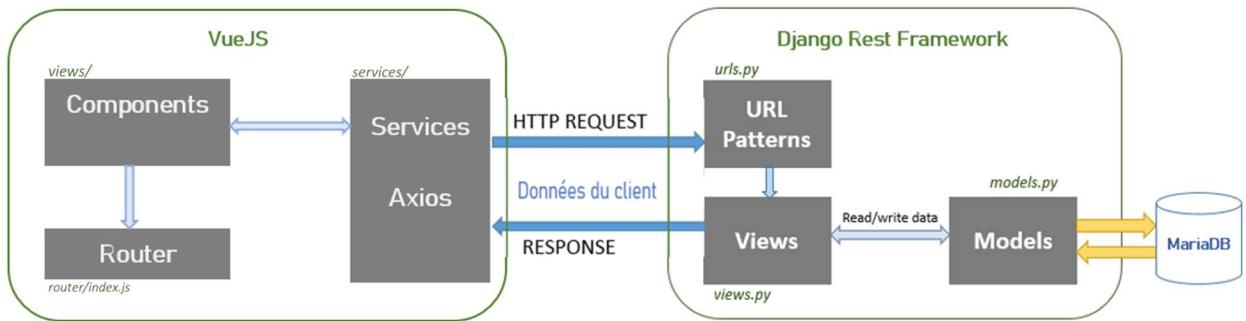


Figure 44: Architecture de la plateforme

D'une part, le côté Frontend développé avec le framework VueJS, il comprend :

- des components (dans le dossier views) : ils concernent les vues à afficher sur la plateforme ;
- le router (dans le fichier router/index.js) : il s'occupe de la navigation entre les urls et dans les vues ;
- des services (dans le dossiers services) : il s'agit de quatre fichiers qui s'occupent de la récupération et de l'envoie des données depuis le backend.

D'autre part, le côté Backend développé avec Django Rest Framework, il comprend :

- les urls (dans le fichier urls.py) : ce sont les endpoints fournies par notre API ;
- les views (dans le fichier views.py) : il s'agit de fonctions qui s'occupent des requêtes http reçues, effectuent des actions en correspondance et qui renvoient une réponse http adéquate ;
- les models (dans le fichier models.py) : chacun d'eux est une représentation d'une table dans la base de données.

Chapitre 8 : Développement de l'application

8.1. Crédation de la base de données

Les conceptions effectuées ont mené à la création d'une base de données que nous allons nommer « its_db ». Cette dernière est générée automatiquement à chaque fois que nous lançons la migration des bases de données depuis le backend.

La figure 45 présente un extrait du code SQL de notre base de données.

```
-- Listage des données de la table itsdb.django_content_type : ~26 rows (environ)
INSERT INTO `django_content_type` (`id`, `app_label`, `model`) VALUES
(1, 'admin', 'logentry'),
(3, 'auth', 'group'),
(2, 'auth', 'permission'),
(4, 'contenttypes', 'contenttype'),
(7, 'restapi', 'answer'),
(8, 'restapi', 'attempt'),
(9, 'restapi', 'category'),
(24, 'restapi', 'color'),
(10, 'restapi', 'content'),
(11, 'restapi', 'course'),
(23, 'restapi', 'group'),
(12, 'restapi', 'institute'),
(13, 'restapi', 'level'),
(22, 'restapi', 'nomenclature'),
(25, 'restapi', 'notification'),
(14, 'restapi', 'parcours'),
(15, 'restapi', 'part'),
(21, 'restapi', 'progress'),
(16, 'restapi', 'question'),
(20, 'restapi', 'questionattempt'),
(19, 'restapi', 'result'),
(18, 'restapi', 'schedule'),
(17, 'restapi', 'type'),
(6, 'restapi', 'user'),
(26, 'restapi', 'usernotification'),
(5, 'sessions', 'session');

-- Listage de la structure de table itsdb.restapi_answer
CREATE TABLE IF NOT EXISTS `restapi_answer` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `text` longtext NOT NULL,
  `is_correct` tinyint(1) NOT NULL,
  `question_id_id` bigint(20) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `restapi_answer_question_id_id_98a5cee1_fk_restapi_question_id` (`question_id_id`),
  CONSTRAINT `restapi_answer_question_id_id_98a5cee1_fk_restapi_question_id`
    FOREIGN KEY (`question_id_id`) REFERENCES `restapi_question` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=81 DEFAULT CHARSET=utf8mb3;
```

Figure 45: Extrait de la base de données

8.2. Codage de l'application

Comme le développement de la plateforme a été découplé de manière à séparer le backend et le frontend, les extraits de code sur le codage de la plateforme seront représentés en les séparant également.

8.2.1. Codage du backend

Le backend est développé avec Django Rest Framework. Ce dernier est conçu pour rendre le processus de développement plus facile et plus rapide.

DRF est livré avec de nombreuses fonctionnalités intégrées, telles qu'un système d'authentification des utilisateurs, un mappeur objet-relationnel (ORM) et un moteur de modèles.

Pour rappel, le dossier du backend contient deux dossiers principaux : its et restapi. Ces dossiers contiennent des fichiers livrés avec DRF que nous complétons au fur et à mesure de notre développement.

Dans un premier temps, nous allons nous focaliser sur le dossier its.

La figure 46 présente le contenu du dossier its.

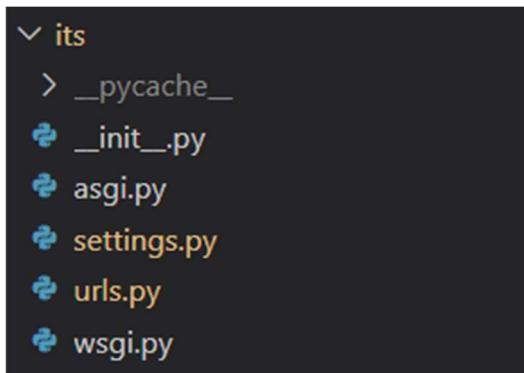


Figure 46: Contenu du dossier its

Pour l'ouvrage, nous allons approfondir seulement sur les fichiers les plus importants à la compréhension et non tous les fichiers.

Dans le dossier its, nous avons le fichier settings.py qui comporte les configurations principales pour tout le projet, à savoir la configuration des applications et modules utilisés, de la base de données, des tokens, ...

La figure 47 expose un extrait de code de settings.py .

```

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'rest_framework',
    'restapi',
    'drf_yasg',
    'corsheaders',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'corsheaders.middleware.CorsMiddleware',
    'django.middleware.common.CommonMiddleware',
    'corsheaders.middleware.CorsMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': env.get('DB_NAME'),
        'USER': env.get('DB_USER'),
        'HOST': env.get('DB_HOST'),
        'PASSWORD': env.get('DB_PASS'),
        'PORT': '3306',
        'OPTIONS': {
            'sql_mode': 'traditional',
        }
    }
}

```

Figure 47: Extrait de code de settings.py

Dans le dossier its, nous avons également le fichier urls.py. Au sein de ce fichier, nous configurons nos endpoints les reliant aux views correspondantes.

La figure 48 présente un extrait de code de urls.py.

```

router = routers.DefaultRouter()
router.register(r'user', UserViewSet)
router.register(r'type', TypeViewSet)
router.register(r'parcours', ParcoursViewSet)
router.register(r'level', LevelViewSet)
router.register(r'category', CategoryViewSet)
router.register(r'group', GroupViewSet)
router.register(r'course', CourseViewSet)
router.register(r'schedule', ScheduleViewSet)
router.register(r'part', PartViewSet)
router.register(r'progress', ProgressViewSet)
router.register(r'content', ContentViewSet)
router.register(r'question', QuestionViewSet)
router.register(r'answer', AnswerViewSet)
router.register(r'attempt', AttemptViewSet)
router.register(r'result', ResultViewSet)
router.register(r'question_attempt', QuestionAttemptViewSet)
router.register(r'institut', InstitutViewSet)
router.register(r'nomenclature', NomenclatureViewSet)
router.register(r'color', ColorViewSet)
router.register(r'notification', NotificationViewSet)
router.register(r'my-notification', MyNotificationViewSet)

urlpatterns = [
    path('admin/', admin.site.urls),

    path('api/v1/', include(router.urls)),
    path('api/v1/quizz/<int:id>', QuizzView.as_view(), name='quizz'),
    path('api/v1/finish-attempt/<int:id>', FinishAttemptView.as_view(), name='finish-attempt'),
    path('api/v1/next-part/', updateProgressViewSet.as_view(), name='next-part'),
    path('api/v1/file-attempt-report/<int:id>', FileAttemptReport.as_view(), name='file-attempt-report'),
    path('api/v1/quizz-report/<int:id>', QuizzReport.as_view(), name='quizz-report'),
    path('api/v1/question-report/<int:id>', QuestionReport.as_view(), name='question-report'),
    path('api/v2/finish-attempt/<int:id>', FinishAttemptView2.as_view(), name='finish-attempt-v2'),

    path('api/v1/login/', ObtainTokenPairView.as_view(), name='obtain_token_pair'),
    path('api/v1/refresh-token/', TokenRefreshView.as_view(), name='token_refresh'),
    path('api/v1/api-auth/', include('rest_framework.urls', namespace='rest_framework')),
    path('api/v1/register/', RegisterView.as_view(), name='auth_register'),
]

```

Figure 48: Extrait de code de urls.py

La figure 49 illustre le contenu du dossier restapi.

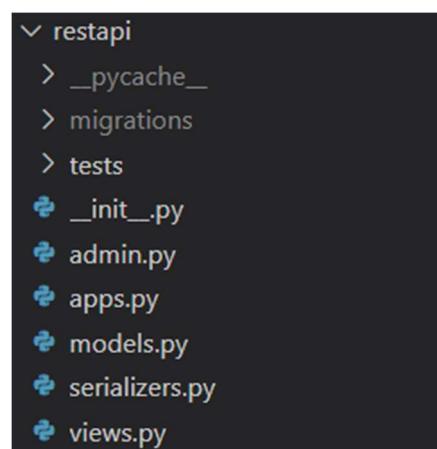


Figure 49: Contenu du dossier restapi

Dans le dossier restapi, nous pouvons voir le fichier models.py. Il contient les modèles de la plateforme. Par définition, un modèle est la source d'information unique et définitive à propos de vos données. Il contient les champs et le comportement essentiels des données que vous stockez. Généralement, chaque modèle correspond à une seule table de base de données.

Les bases :

- Chaque modèle est une classe Python qui hérite de django.db.models.Model.
- Chaque attribut du modèle représente un champ de base de données.
- Avec tout cela, DRF vous offre une API d'accès à la base de données générée automatiquement avec la commande « python manage.py migrate »

La figure 50 montre un extrait de codes de models.py.

```
class Level(models.Model):  
    name = models.CharField(_("Name"), max_length=56, unique=True)  
  
    def __str__(self):  
        return self.name  
  
  
class Category(models.Model):  
    name = models.CharField(_("Name"), max_length=56, unique=True)  
  
    def __str__(self):  
        return self.name  
  
  
class Group(models.Model):  
    parcours_id = models.ForeignKey(to=Parcours, on_delete=models.CASCADE, null=False)  
    level_id = models.ForeignKey(to=Level, on_delete=models.CASCADE, null=False)  
  
    def __str__(self):  
        return "%s %s" % (self.level_id.name, self.parcours_id.name)  
  
  
class User(AbstractUser):  
    matricule = models.CharField(max_length=16, blank=True, unique=True)  
    email = models.EmailField(unique=True)  
    description = models.CharField(max_length=120, blank=True)  
    role = models.CharField(max_length=16, choices=USER_ROLES, default="STUDENT")  
    group_id = models.ForeignKey(to=Group, verbose_name="Group", related_name="users",  
                                on_delete=models.SET_NULL, null=True, blank=True)  
    image = models.ImageField(upload_to='users', blank=True, null=True)  
  
    def __str__(self):  
        return "%s %s" % (self.first_name, self.last_name)
```

Figure 50: Extrait de code de models.py

Dans le dossier restapi, nous pouvons voir également le fichier serializers.py. Ce dernier contient les sérialiseurs du Backend. L'infrastructure de sérialisation de DRF fournit un mécanisme pour « traduire » les modèles Django Rest Framework en d'autres formats. En général, ces autres formats sont basés sur du texte et utilisés pour envoyer des données de DRF à travers le réseau, mais il est possible qu'un sérialiseur prenne en charge n'importe quel format. On y mentionne :

- les données retournées et la forme de ces données
- des fonctions sur la récupération, la création, la modification et la suppression de données

La figure 51 illustre un extrait de code du fichier serializers.py.

```

class GroupSerializer(serializers.ModelSerializer):
    name = serializers.SerializerMethodField()
    level_name = serializers.SerializerMethodField()
    parcours_name = serializers.SerializerMethodField()

    class Meta:
        model = Group
        fields = ['url', 'id', 'name', 'level_name', 'parcours_name', 'parcours_id', 'level_id']

    def get_name(self, obj):
        return obj.__str__()

    def get_level_name(self, obj):
        return obj.level_id.name

    def get_parcours_name(self, obj):
        return obj.parcours_id.name


class TypeSerializer(serializers.ModelSerializer):
    class Meta:
        model = Type
        fields = ['url', 'id', 'name']


class PartSerializer(serializers.ModelSerializer):

    contents = serializers.SerializerMethodField(read_only=True)

    def get_contents(self, part):
        user = self.context['request'].user
        query = Content.objects.filter(part_id=part)
        if user.role == "STUDENT":
            query = query.filter(visible=True)
        serializer = ContentSerializer(instance=query, many=True, context=self.context)
        return serializer.data

    class Meta:
        model = Part
        fields = ['url', 'id', 'name', 'description', 'course_id', 'sequence', 'visible', 'contents']

```

Figure 51: Extrait de codes de serializers.py

Un des fichiers les plus importants dans notre dossier restapi est également le fichier views.py qui contient les fonctions de vues du backend. Une fonction de vue, ou vue pour faire simple, est une fonction Python acceptant une requête web et renvoyant une réponse web. Cette réponse peut contenir le contenu HTML d'une page web, une redirection, une erreur 404, un document XML, une image... ou vraiment n'importe quoi d'autre. La vue elle-même contient la logique nécessaire pour renvoyer une réponse.

La figure 52 représente un extrait de code du fichier views.py.

```

class CourseViewSet(ModelViewSet):
    """
        API Endpoint that allows courses to be viewed or edited
    """
    queryset = Course.objects.all()
    serializer_class = CourseSerializer
    permission_classes = [IsAuthenticated]

    def get_queryset(self):
        """
            Optionally restricts the returned course list to a given user,
            by filtering against a `level_id` and `parcours_id` query parameter in the URL.
        """
        data = self.request.query_params
        queryset = Course.objects.all()

        parcours_id = data.get("parcours_id")
        level_id = data.get("level_id")
        if level_id and parcours_id:
            queryset = queryset.filter(groups_level_id__id__in=level_id,
                                      groups_parcours_id__id__in=parcours_id)

        if self.request.user.role == "PROF":
            queryset = queryset.filter(teacher=self.request.user)
        elif self.request.user.role == "STUDENT":
            queryset = queryset.filter(groups__in=[self.request.user.group_id])

        return queryset

class ScheduleViewSet(ModelViewSet):
    """
        API Endpoint that allows schedules to be viewed or edited
    """
    queryset = Schedule.objects.all()
    serializer_class = ScheduleSerializer
    permission_classes = [IsAuthenticated]

```

Figure 52: Extrait de code de views.py

Dans le dossier restapi, nous avons pu voir le dossier tests.

Ce dossier contient deux fichiers écrits en python. Ces fichiers sont des tests unitaires effectués l'un sur le model et l'un sur les vues. Pour rappel, les tests unitaires permettent de vérifier le bon fonctionnement d'une petite partie bien précise (unité ou module) d'une application. Ils s'assurent qu'une méthode exposée à la manipulation par un utilisateur fonctionne bien de la façon dont elle a été conçue.

La figure 53 expose un extrait de code du fichier test_models.py.

```
class LevelTest(TestCase):
    def setUp(self):
        Level.objects.create(name="L1")

    def test_level_name(self):
        level = Level.objects.get(name="L1")
        self.assertEqual(level.__str__(), "L1")

class CategoryTest(TestCase):
    def setUp(self):
        Category.objects.create(name="C1")

    def test_category_name(self):
        category = Category.objects.get(name="C1")
        self.assertEqual(category.__str__(), "C1")

class GroupTest(TestCase):
    def setUp(self):
        Parcours.objects.create(name="P1")
        Level.objects.create(name="L1")
        Group.objects.create(parcours_id=Parcours.objects.get(name="P1"), level_id=Level.objects.get(name="L1"))

    def test_group_name(self):
        group = Group.objects.get(parcours_id=Parcours.objects.get(name="P1"), level_id=Level.objects.get(name="L1"))
        self.assertEqual(group.__str__(), "L1 P1")

class UserTest(TestCase):
    def setUp(self):
        Parcours.objects.create(name="P1")
        Level.objects.create(name="L1")
        self.group = Group.objects.create(parcours_id=Parcours.objects.get(name="P1"), level_id=Level.objects.get(name="L1"))

    def test_user_name(self):
        user = User.objects.create(matricule="MAT", email="email@gmail.com", description="Description", first_name="First",
                                    last_name="Last", group_id=self.group, image="image.png")
        self.assertEqual(user.__str__(), "First Last")
```

Figure 53: Extrait de code du fichier test_models.py

Pour lancer les tests sur les models, nous lançons la commande : « python manage.py tests test_models.py»

Sur le terminal, nous obtenons un point « . » si le test est réussi et « F » si le test n'est pas réussi.

La figure 54 présente un extrait de code de test_views.py.

```

# Type
class CreateNewTypeTest(APITestCase):
    """ Test module for inserting a new type """

    def setUp(self):
        self.u2 = User.objects.create(username="U2", email="user2@test.com", password="123456", matricule="002")
        self.client.force_authenticate(user=self.u2)
        self.valid_payload = { 'name': 'Type 1'}
        self.invalid_payload = {'not_name': 'Type 1'}

    def test_create_valid_type(self):
        response = self.client.post('/api/v1/type/', data=json.dumps(self.valid_payload), content_type='application/json')
        self.assertEqual(response.status_code, status.HTTP_201_CREATED)

    def test_create_invalid_type(self):
        response = self.client.post('/api/v1/type/', data=json.dumps(self.invalid_payload), content_type='application/json')
        self.assertEqual(response.status_code, status.HTTP_400_BAD_REQUEST)

class GetTypeTest(APITestCase):
    """ Test module for GET types API """

    def setUp(self):
        self.u1 = User.objects.create(username="U1", email="user1@test.com", password="123456", matricule="001")
        self.t1 = Type.objects.create(name="Type 1")
        self.t2 = Type.objects.create(name="Type 2")
        self.t3 = Type.objects.create(name="Type 3")

    def test_get_all_type(self):
        # get API response
        self.client.force_authenticate(user=User.objects.get(username="U1"))
        response = self.client.get('/api/v1/type/')
        # get data from db
        types = Type.objects.all()
        serializer = TypeSerializer(types, many=True, context={'request': response.wsgi_request})
        self.assertEqual(response.data, serializer.data)
        self.assertEqual(response.status_code, status.HTTP_200_OK)

    def test_get_valid_single_type(self):
        self.client.force_authenticate(user=User.objects.get(username="U1"))
        response = self.client.get(f'/api/v1/type/{self.t3.pk}/')
        type = Type.objects.get(pk=self.t3.pk)
        serializer = TypeSerializer(type, context={'request': response.wsgi_request})
        self.assertEqual(response.data, serializer.data)
        self.assertEqual(response.status_code, status.HTTP_200_OK)

    def test_get_invalid_single_user(self):
        self.client.force_authenticate(user=User.objects.get(username="U1"))
        response = self.client.get('/api/v1/type/npq/')
        self.assertEqual(response.status_code, status.HTTP_404_NOT_FOUND)

class DeleteTypeTest(APITestCase):
    """ Test module for DELETE types API """

    def setUp(self):
        self.u1 = User.objects.create(username="U1", email="user1@test.com", password="123456", matricule="001")
        self.t1 = Type.objects.create(name="Type 1")

    def test_delete_type(self):
        self.client.force_authenticate(user=User.objects.get(username="U1"))
        response = self.client.delete(f'/api/v1/type/{self.t1.pk}/')
        self.assertEqual(response.status_code, status.HTTP_204_NO_CONTENT)

```

Figure 54: Extrait de code de test_views.py

Pour lancer les tests sur les models, nous lançons la commande : « python manage.py tests test_models.py ». Sur le terminal, nous obtenons un point si le test est réussi « . » et « F » si le test n'est pas réussi.

8.2.2. Codage du frontend

Nous avons vu précédemment l’arborescence de notre frontend développé avec VueJS. Vue (prononcé /vju:/, comme le terme anglais view) est un framework évolutif pour construire des interfaces utilisateur. [j]

A la racine du projet, nous pouvons voir le fichier package.json qui contient les dépendances utilisées par l’application.

La figure 55 résume le contenu de package.json.

```
{
  "name": "its",
  "private": true,
  "version": "0.0.0",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": [
    "@ckeditor/ckeditor5-build-classic": "^34.1.0",
    "@ckeditor/ckeditor5-vue": "^4.0.1",
    "@popperjs/core": "^2.11.6",
    "@vueform/multiselect": "^2.5.2",
    "axios": "^0.27.2",
    "chart.js": "^3.9.1",
    "jwt-decode": "^3.1.2",
    "moment": "^2.29.4",
    "mosha-vue-toastify": "^1.0.23",
    "sweetalert": "^2.1.2",
    "vue": "^3.2.25",
    "vue-chartjs": "^4.1.2",
    "vue-color-kit": "^1.0.5",
    "vue-router": "^4.0.15",
    "vue3-easy-data-table": "^1.4.2",
    "vue3-select2-component": "^0.1.5",
    "vuex": "^4.0.2"
  ],
  "devDependencies": {
    "@vitejs/plugin-vue": "^2.3.3",
    "bootstrap": "^5.1.3",
    "popper.js": "^1.16.1",
    "sass": "^1.52.1",
    "sass-loader": "^13.0.0",
    "vite": "^2.9.15"
  }
}
```

Figure 55: Contenu de package.json

Pour commencer l'immersion dans le codage du frontend, nous allons nous tourner dans la récupération des données venant de notre API. Les fichiers qui s'occupent des récupérations de ces données et de l'interaction avec le backend sont les fichiers qui se trouvent dans le dossier src/services. Ce sont les fichiers :

- create.js pour la création de données
- read.js pour la récupération de données
- update.js pour la modification de données
- delete.js pour la suppression de données

La figure 56 illustre un extrait de codes du contenu de create.js.

```
export default async function createData(uri, data, next) {
  const headers = {
    'Content-Type': 'application/json',
    'Authorization': `Bearer ${localStorage.getItem('access')}`
  };
  try {
    await axios.post(`${uri}`, data, {headers})
      .then((response)=>{
        if(response && response.status == 201) {
          return next(response);
        } else if (response && response.status == 200) {
          return next(response);
        }
      })
      .catch((error)=>{
        if (error.response) {
          if (error.response.status == 401) {
            notify('Attention','Vous n\'êtes pas autorisé à effectuer cette action.', 'warning');
            setTimeout(function(){
              router.replace({ path: '/login' });
              localStorage.removeItem('access');
              localStorage.removeItem('user');
            }, 3000);
          } else if (error.response.statusText == 'Forbidden') {
            notify('Attention','Vous n\'avez pas accès à cette action.', 'danger');
            setTimeout(function(){
              router.replace({ path: '/login' });
              localStorage.removeItem('access');
              localStorage.removeItem('user');
            }, 3000);
          } else if (error.response.statusText == 'Bad Request') {
            if (error.response.data) {
              notify(['Erreun',Object.values(error.response.data)[0][0], 'danger']);
            }
          } else {
            notify('Erreun',error.response.statusText, 'danger');
          }
        }
      });
  } catch (error) {
    console.log(error)
  }
}
```

Figure 56: Extrait de code de create.js

La figure 57 présente un extrait de codes du contenu de read.js.

```

export default async function readData(uri, next){
  const headers = {
    'Content-Type': 'application/json',
    'Authorization': `Bearer ${localStorage.getItem('access')}`
  };
  let message = 'Une erreur a survenue. Veuillez réessayer plus tard';
  let type = 'warning';
  try {
    await axios.get(`${uri}`, {headers})
      .then((response)=>{
        if(response) {
          if (response.status == 200) {

            return next(response);
          } else if (response.status == 204) {
            message = 'Il n\'y a aucun contenu à afficher .';
            notify('Erreur', message, 'danger');
          }
        }
      })
      .catch((error)=>{
        console.log(error)
        if( error.response) {
          if (error.response.statusText == 'Unauthorized') {
            message = 'Vous n\'êtes pas autorisé à accéder à ce contenu.';
            notify('Attention',message, type);
            setTimeout(function(){
              router.replace({ path: '/login' });
              localStorage.removeItem('access');
              localStorage.removeItem('user');
            }, 3000);
          } else if (error.response.statusText == 'Forbidden') {
            message = 'Vous n\'avez pas accès au contenu que vous tentez d'accéder.';
            notify('Attention',message, type);
            setTimeout(function(){
              router.replace({ path: '/login' });
              localStorage.removeItem('access');
              localStorage.removeItem('user');
            }, 3000);
            type = 'danger';
          } else {
            notify('Erreur',error.response.statusText, 'error');
          }
        }
      })
    } catch (error) {
      notify('Erreur',message, 'danger');
    }
}

```

Figure 57: Extrait de code de read.js

La figure 58 montre un extrait de code de update.js.

```

export default async function updateData(uri, data, next){
    const headers = {
        'Content-Type': 'application/json',
        'Authorization': `Bearer ${localStorage.getItem('access')}`
    };
    try {
        await axios.patch(`#${uri}`, data, {headers})
            .then((response)=> {
                return next(response)
            })
            .catch((error)=>{
                if (error.response.statusText == 'Unauthorized') {
                    notify('Attention','Vous n\'êtes pas autorisé à effectuer cette action.', 'warning');
                    setTimeout(function(){
                        router.replace({ path: '/login' });
                        localStorage.removeItem('access');
                        localStorage.removeItem('user');
                    }, 3000);
                } else if (error.response.statusText == 'Forbidden') {
                    notify('Attention','Vous n\'avez pas accès à cette action.', 'danger');
                    setTimeout(function(){
                        router.replace({ path: '/login' });
                        localStorage.removeItem('access');
                        localStorage.removeItem('user');
                    }, 3000);
                } else if (error.response.statusText == 'Bad Request') {
                    if (error.response.data) {
                        notify('Erreur',Object.values(error.response.data)[0][0], 'danger');
                    }
                }
                else {
                    notify('Erreur',error.response.statusText, 'error');
                }
            })
    } catch (error) {
        notify('Erreur','Une erreur a survenue. La modification n\'a pas pu être effectuée.', 'danger');
    }
}

```

Figure 58: Extrait de codes de update.js

La figure 59 affiche un extrait de code de delete.js.

```

export default async function deleteData(uri, next) {
  const headers = {
    'Content-Type': 'application/json',
    'Authorization': `Bearer ${localStorage.getItem('access')}`
  };
  try {
    await axios.delete(`#${uri}`, {headers})
      .then((response)=> {
        if(response) {
          return next(response)
        }
      })
      .catch((error)=>{
        if(error.response) {
          if (error.response.statusText == 'Unauthorized') {
            notify('Attention','Vous n\'êtes pas autorisé à effectuer cette action.', 'warning');
            setTimeout(function(){
              router.replace({ path: '/login' });
              localStorage.removeItem('access');
              localStorage.removeItem('user');

            }, 3000);
          } else if (error.response.statusText == 'Forbidden') {
            notify('Attention','Vous n\'avez pas accès à cette action.', 'danger');
            setTimeout(function(){
              router.replace({ path: '/login' });
              localStorage.removeItem('access');
              localStorage.removeItem('user');

            }, 3000);
          } else {
            notify('Erreur',error.response.statusText, 'error');
          }
        }
      })
    } catch (error) {
      notify('Erreur','Une erreur a survenu. La suppression n\'a pas pu être effectuée.', 'danger');
    }
}

```

Figure 59: Extrait de delete.js

Ensuite, nous allons nous pencher vers la navigation au sein de la plateforme. La navigation entre les pages sont orchestrés par le fichier index.js dans src/router, qui fait correspondre les composants de vue aux urls.

La figure 60 montre un extrait de code de router/index.js.

```

{
  path: '/courses/:parcours/',
  name: 'CoursesFilterByParcours',
  beforeEnter : [TeacherOrAdminGuard],
  component: CoursesListPage,
  children: [
    {
      path: ':level',
      name: 'CoursesFilterByLevel',
      component: CoursesListPage,
    }
  ],
},
{
  path: '/courses',
  name: 'CoursesListPage',
  component: CoursesListPage,
},
{
  path: '/',
  name: 'Login',
  component: LoginPage
},
{
  path: '/login',
  name: 'Login_Page',
  component: LoginPage
},
{
  path: '/app',
  name: 'AppPage',
  component: AppPage
},
{
  path: '/course',
  name: 'AddCoursePage',
  beforeEnter : [TeacherGuard],
  component: AddCoursePage,
  children: [
    {
      path: ':course',
      name: 'EditCoursePage',
      component: AddCoursePage,
    }
  ],
},
{
  path: '/course/add-content/:id',
  name: 'CoursePreview',
  beforeEnter : [TeacherGuard],
  component: CoursePreview
},

```

Figure 60: Extrait de code de router/index.js

Comme nous développons notre frontend avec VueJS, il suit ses fonctionnements. De ce fait, toute accès à la plateforme conduit au component src/views/App.vue.

La figure 61 présente le contenu de src/views/App.vue.

```

<template>
| <router-view/>
</template>

<script>
export default {
  beforeMount() {
    this.$store.dispatch('initializeToken');
  }
}
</script>

<style>
@import '../node_modules/bootstrap/dist/css/bootstrap.min.css';
@import "https://fonts.googleapis.com/css?family=Roboto:300,400,500,600,700";
@import url('https://fonts.googleapis.com/css2?family=Dancing+Script&family=IBM+Plex+Mono:wght@100&family=Poppins:wght@300&family=Rubik:wght@300,400,500,700');

#app, .modals {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  color: #2c3e50;
  background-color: #f9fbfd;
  font-family: 'Roboto', sans-serif;
  height: 100vh;
  overflow-x: hidden;
}

.col {
  padding: 0px;
}

</style>

```

Figure 61: Contenu de src/views/App.vue

<template></template> correspond à l'affichage à notre écran.

Grâce à ce code, nous voyons à l'intérieur de ce dernier <router-view />, cela signifie qu'il sera remplacé par le component qui correspond à l'url noté dans router/index.js.

Entre les balises <script></script>, nous pouvons voir la création de la vue. Si nous devons importer des fichiers ou des components, nous les insérons directement après la balise ouvrante. Puis, nous avons ici la syntaxe commune « export default {} » et à l'intérieur écrivons le nécessaire à la création de la vue, si nous avons besoin d'autres components, nous les citons dans « components : {} ».

Enfin, en bas page, nous avons la balise <style></style> qui contiendra le style de la page. Les styles écrit entre ces balises, dans ce fichier, est globale et valable pour toutes les autres pages qui sont ces enfants.

Dans le cas où nous voulons spécifier juste le style de ce component, nous modifions la balise comme <style scoped></style>.

A présent, nous allons voir les composants qui remplacent le contenu de <router-view/>. Ces composants sont organisés dans le dossier src/views.

La figure 62 expose la structure du dossier src/views.

```

    < views
      < components
        > admin
        > layouts
        > utils
        < ContentView.vue
        < CourseContent.vue
        < CourseView.vue
        < FooterView.vue
        < LoginView.vue
        < QuizzQuestion.vue
        < QuizzReponse.vue
      > errors
    < pages
      < AddCoursePage.vue
      < AdPage.vue
      < AttemptedFile.vue
      < ConsultAds.vue
      < CoursePreview.vue
      < CoursesDetails.vue
      < CoursesListPage.vue
      < LoginPage.vue
      < ProfileView.vue
      < QuestionReport.vue
      < QuizzBuilding.vue
      < QuizzMaking.vue
      < QuizzReport.vue
      < UserView.vue
    < AppPage.vue
  
```

Figure 62: Structure du dossier src/views

Le dossier pages comporte les composants de vue parents, le dossier components comprend les composants de vue enfants et le dossier errors comprend les pages d'erreurs.

La figure 63 illustre un extrait de code de CoursesListPage qui constitue la page d'accueil de chaque connexion. Il s'agit de la liste des cours adéquat au rôle de l'utilisateur connecté.

```

<template>
  <app-page>
    <template v-slot:content="slot_props">
      <div class="container mt-5" v-if="route.name !== 'CoursesListPage'>
        <div class="d-flex row">
          <div class="col-md-7">
            <span class="mt-3">Trier par {{slot_props.nomenclature.group}} :</span>
            <router-link :to="{name: 'CoursesFilterByLevel', params : {level: level.id}}"
              class="btn btn-secondary mx-2 mt-2" v-for="level in levels">{{ level.name }}</router-link>
          </div>
        </div>
      </div>
      <div class="courses-container">
        <h2 class="title"> {{slot_props.nomenclature.course}} disponibles</h2>
        <app-filter v-if="defaultConfig && defaultConfig.typeview"
          :data="courses" :config="defaultConfig" :school_color="slot_props.color.color_1"
          @updateView="(type)=> typeview = type"
          @updateDataFilter="(progress)=> filter = progress"
          @updateDataSearch="(search)=> userSearch = search"
        />

        <div class="grid-1" :class="typeview">
          <course v-for="course in visibleCourses" :color="slot_props.color"
            :course="course" @click="goTo(course.id)" :key="course.id" />
          <div v-if="!visibleCourses.length">
            <p :style="{textAlign:'center'}">
              Aucun <span class="text-lowercase" > {{ slot_props.nomenclature.course }} </span> à afficher
            </p>
          </div>
        </div>
        <app-pagination
          @updatePage="updatePage"
          :config="paginationConfig"
          :color="slot_props.color.color_1"
          v-show="courses.length >= 3"
        />
      </div>
    </template>
  </app-page>
</template>

<script>
import {ref, watch, onBeforeMount} from "vue";
export default{
  name: 'CoursesListPage',
  components:{...},
  data() {...},
  setup(){
    const router = useRouter();
    const route = useRoute();
    const store = useStore();
    let typeview = ref('');
    let levels = ref([]);
    let defaultConfig = ref({})
    let courses = ref([])

```

Figure 63: Extrait de code de CoursesListsPage.vue

La figure 64 montre un extrait de code de ProfileView.vue. Il s'agit de la page qui est dédié à la modification du profil d'un utilisateur.

```

<template>
  <app-page>
    <template v-slot:content="slot_props">
      <div class="container px-5">
        <form @submit.prevent="save" v-if="user">
          <div class="my-5 row">
            <label for="username" class="col-sm-4 col-form-label">Nom d'utilisateur</label>
            <div class="col-sm-8">
              <input type="text" readonly class="form-control" id="username" v-model="user.username">
            </div>
          </div>
          <div class="my-5 row">
            <label for="last_name" class="col-sm-4 col-form-label">Nom</label>
            <div class="col-sm-8">
              <input type="text" class="form-control" id="last_name" v-model="user.last_name">
            </div>
          </div>
          <div class="my-5 row">
            <label for="first_name" class="col-sm-4 col-form-label">Prénom(s)</label>
            <div class="col-sm-8">
              <input type="text" class="form-control" id="first_name" v-model="user.first_name">
            </div>
          </div>
          <div class="my-5 row">
            <label for="email" class="col-sm-4 col-form-label">E-mail</label>
            <div class="col-sm-8">
              <input type="text" class="form-control" id="email" v-model="user.email">
            </div>
            <button type="submit" class="btn btn-secondary" id="save">Enregistrer</button>
          </form>
        </div>
      </template>
    </app-page>
</template>

<script>
  import AppPage from "../AppPage.vue"

  export default {
    name: 'ProfileView',
    components: {
      'app-page': AppPage,
    },
    data() {
      return ...
    }
  },
  beforeMount() {
    var user_id = jwt_decode(localStorage.getItem('access')).user_id;
    readData(`api/v1/user/${user_id}/`, (response) => {
      if (response.data) {
        this.user = response.data
      }
    });
  },
  methods: {
    save: function () {
      delete this.user.image;
      updateData(`api/v1/user/${this.user.id}/`, this.user, async (response) => {
        if (response.status == 200) {
          notify('Succès', 'Modification effectuée avec succès', 'success');
          this.router.go();
        }
      });
    }
  }
}
</script>

```

Figure 64: Extrait de code de ProfileView.vue

Dans src, nous pouvons également voir le tests qui comporte le fichier end_to_end_tests.py. Ce fichier comporte les tests de bout en bout du frontend. Par définition, les tests de bout en bout sont une technique utilisée pour vérifier si notre plateforme se comporte comme prévu du début à la fin. Le testeur doit se mettre dans la peau d'un utilisateur et dérouler les tests comme s'il utilisait véritablement l'outil mis à sa disposition. Cette technique permet de valider le fonctionnement du front, mais aussi de vérifier son intégration avec le back-office. Les tests que nous effectuons sont développés en utilisant l'outil playwright et le langage python.

La figure 65 présente un extrait de code du test de bout en bout du frontend.

```

import os
import re
from time import sleep
from dotenv import load_dotenv
from playwright.sync_api import Page, expect

load_dotenv()

> def customInput(page: Page, name, text): ...

def test_1_connexion_valid_login(page: Page, mode = 'ADMIN'):
    if mode == 'ADMIN':
        valid_data = { 'username': os.getenv('ADMIN_USERNAME'), 'password': os.getenv('ADMIN_PASSWORD') }
    elif mode == 'STUDENT':
        valid_data = { 'username': os.getenv('STUDENT_USERNAME'), 'password': os.getenv('STUDENT_PASSWORD') }
    elif mode == 'TEACHER':
        valid_data = { 'username': os.getenv('TEACHER_USERNAME'), 'password': os.getenv('TEACHER_PASSWORD') }
    page.goto(f"{os.getenv('VITE_BASEURL')}/login")

    expect(page).to_have_title(re.compile("iTS e-learning"))
    expect(page).to_have_url(re.compile(".*login"))

    page.query_selector("input[name='username']").fill(valid_data['username'])
    page.query_selector("input[name='password']").fill(valid_data['password'])
    page.get_by_text('Connexion').click()

    expect(page).to_have_url(re.compile(".*courses"))
    sleep(2)

> def test_1_connexion_invalid_login(page: Page): ...

> def test_1_connexion_logout(page: Page): ...

def test_2_edit_profile(page: Page):
    test_1_connexion_valid_login(page)

    page.query_selector("a[id='params']").click()
    page.query_selector("a[id='showProfile']").click()

    expect(page).to_have_url(re.compile(".*profile"))
    page.locator("input#last_name").fill('RAKOTO')
    page.locator("input#first_name").fill('Vony')
    page.locator("input#email").fill('email@gmail.com')
    page.locator("button#save").click()

    page.get_by_text('Vony RAKOTO')

```

Figure 65: Extrait de code de end_to_end_tests.py

8.3. Présentation de l'application

La plateforme its.mg est une plateforme qui va contribuer fortement à la mise en ligne des enseignements des instituts. Pour cela, elle dispose de plusieurs fonctionnalités pour chaque type d'utilisateur qui sont :

- les administrateurs ;
- les enseignants ;
- les étudiants.

Les pages communs aux trois utilisateurs seront mis en avant en premier, ensuite, les pages spécifiques à chaque utilisateur et enfin, l'accès au bot messenger.

Il est important de noter que chaque institut peut personnaliser l'interface de sa plateforme, que de par les couleurs, les images, ou même les nomenclatures.

La figure 66 affiche la page de connexion à la plateforme.

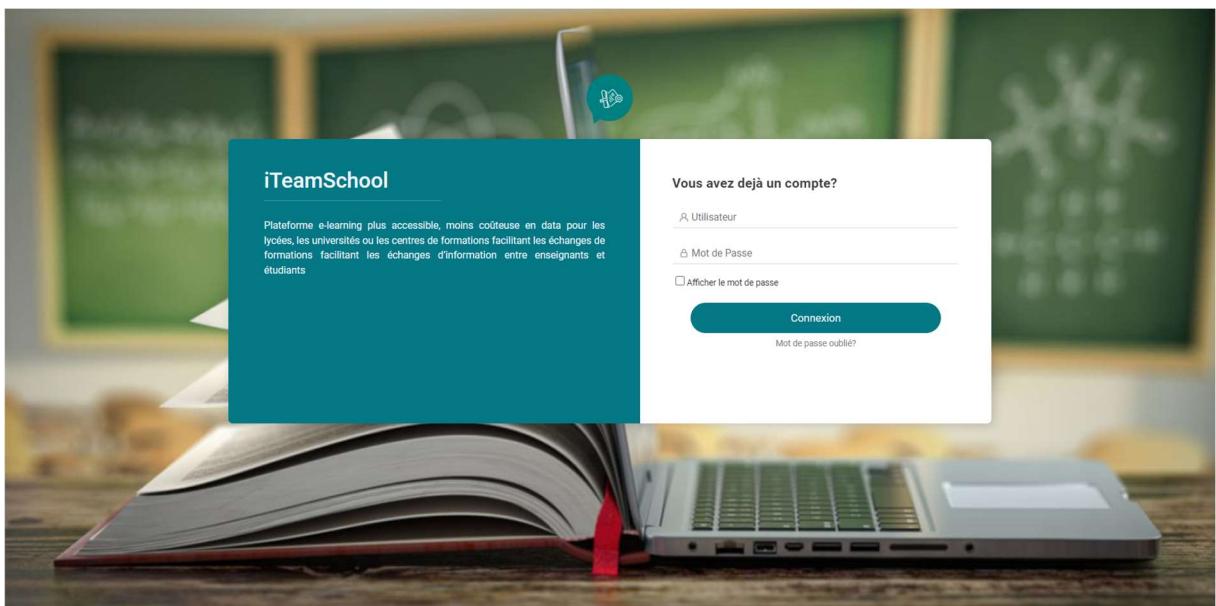


Figure 66: Page de connexion

La figure 67 présente la modification du profil d'un utilisateur.

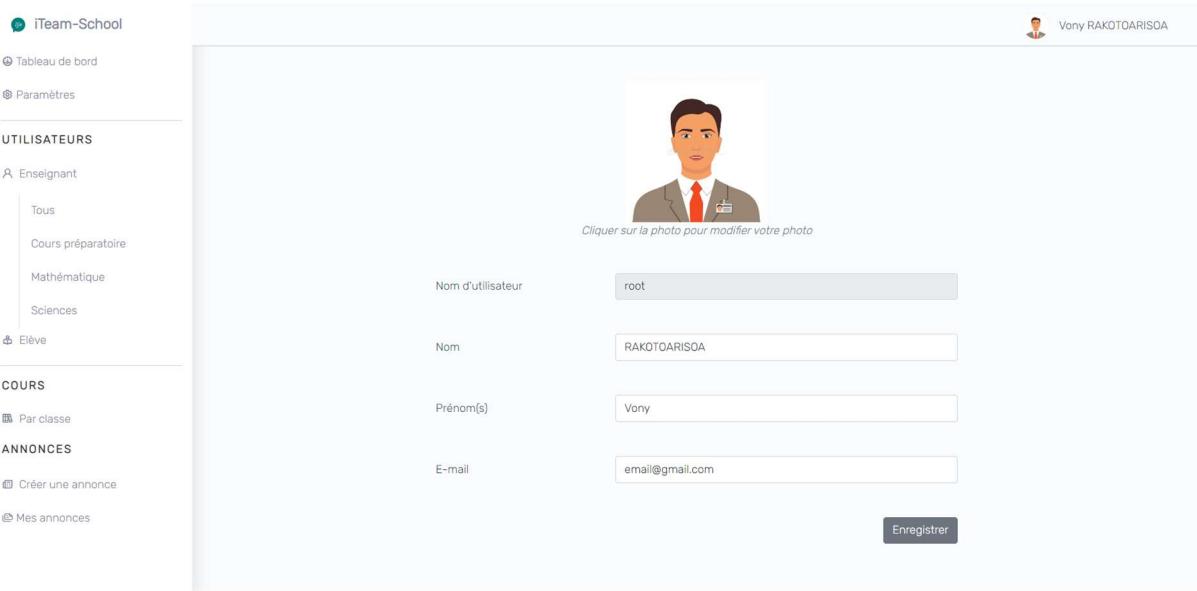


Figure 67: Modification du profil

La figure 68 illustre la page d'accueil de tous les utilisateurs qui est la liste des cours, cette liste varie en fonction des rôles de l'utilisateur.

The screenshot shows the 'Tableau de bord' (Dashboard) of the iTeam-School application. The sidebar includes links for Tableau de bord, Paramètres, UTILISATEURS (Enseignant, Elève), COURS (Par classe), and ANNONCES (Créer une annonce, Mes annonces). The main content area is titled 'Cours disponibles' (Available Courses) and lists three courses: 'Notre Institut APPRENTISSAGE' (with a brief description about creating iTeam School), 'Javascript DÉVELOPPEMENT' (with a brief description about creating programs), and 'Français APPRENTISSAGE' (with a brief description about French being the most widely learned language).

Figure 68: Tableau de bord

8.3.1. Pour les administrateurs

La figure 69 et la figure 70 montrent la page pour l'administration d'une plateforme d'un institut.

Administration
Ici vous pouvez faire la configuration entière de la plateforme

Veuillez sélectionner pour faire la modification de l'institut!

Informations générales

LOGO	Personnalisez votre plateforme en ajoutant une photo
NOM DE L'INSTITUT	iTeam-School
ABBREVIATION	ITS
DESCRIPTION	Plateforme e-learning plus accessible, moins coûteuse en data pour les lycées, les universités ou les centres de formations facilitant les échanges de formations facilitant les échanges d'information entre enseignants et étudiants.

Coordonnées

PHONE	0322025810
-------	------------

Figure 69: Page d'administration de la plateforme - Partie 1

Autres

NOMENCLATURES	>
COULEURS	>

Paramétrage de la plateforme

PARCOURS	>
NIVEAU	>
GROUP	>
CATÉGORIE	>

Figure 70: Page d'administration de la plateforme - Partie 2

La figure 71 présente la page dédiée à la gestion des enseignants.

Figure 71: Page dédiée à la gestion des enseignants

La figure 72 illustre la création d'un enseignant.

Figure 72: Page de création d'un enseignant.

La figure 73 représente la page dédiée à la gestion des élèves.

La gestion des élèves se fait par classe.

The screenshot shows a user interface for managing students. At the top right, there is a profile picture of 'Vony RAKOTOARISOA'. On the left, a sidebar menu includes 'Paramètres', 'UTILISATEURS' (with 'Enseignant' and 'Elève' sections), 'COURS' (with 'Par classe'), and 'ANNONCES' (with 'Créer une annonce' and 'Mes annonces'). The main content area displays a list of students under 'COURS PRÉPARATOIRE / 1ÈRE ANNÉE'. Each student entry includes a profile picture, name, class, email, and three buttons: 'Activer la modification', 'Enregistrer', and 'Effacer'. The students listed are:

- RABENANTOANINA Narindra (@ET004) - Nouvelle étudiante en cours préparatoire pour première année
- RAJAONARISON Nirina (@ET001) - Etudiante studieuse en cours préparatoire pour première année
- RABE Antsa (@rabAntsa) (@ET009)

Figure 73: Page dédiée à la gestion des étudiants

La page 74 présente la page pour la création d'une annonce, pour un administrateur, il peut l'adresser à une ou plusieurs classes, ou à une ou plusieurs enseignants. Pour un enseignant, il ne peut l'adresser qu'à une ou plusieurs classes.

The screenshot shows a form for creating a new announcement. At the top right, there is a profile picture of 'Vony RAKOTOARISOA'. On the left, a sidebar menu includes 'Tableau de bord', 'Paramètres', 'UTILISATEURS' (with 'Enseignant' and 'Elève'), 'COURS' (with 'Par classe'), and 'ANNONCES' (with 'Créer une annonce' and 'Mes annonces'). The main content area is titled 'Nouvelle annonce' and contains the following fields:

- Texte de l'annonce:** A text input field containing: "Qu'avez vous à annoncer ? A tous les élèves en première année. Une assemblée générale pour les nouveaux étudiants aura lieu le lundi 5 décembre 2022 à 8:00. Votre présence est obligatoire. PS: tenue correcte exigée".
- L'audience:** A dropdown menu where the option "Adressé aux Groups" is selected.
- Catégories:** A dropdown menu showing selected categories: "1ère année Cours préparatoire", "Mathématiques", and "Génie logiciel et base de données".
- Date de publication:** A date input field set to "2022-12-05T08:10".
- Action buttons:** "Annuler" and "Publier ➡".

Figure 74: Page dédié à la création d'annonce

8.3.2. Pour les enseignants

La figure 75 affiche la page pour la création d'un cours.

Figure 75: Création d'un cours

La page 76 présente un cours créé avec ses parties.

Cette page peut conduire à :

- la modification du cours
- la consultation des parties du cours et de leurs contenus
- l'ajout de parties du cours,
- l'ajout de contenus des parties.

Figure 76: Un cours créé avec ses parties

La figure 77 présente la création d'un quizz.

The screenshot shows the iTeam-School interface. On the left, there's a sidebar with 'iTeam-School' logo, 'Tableau de bord', 'COURS' (Create course, By class), 'ANNONCES' (Create announcement, My announcements), and a 'Nouveau quizz' button. The main area is titled 'Votre questionnaire' with a subtitle 'Évaluez vos acquis grâce à un quizz'. It has fields for 'Titre' (Title) containing 'Évaluez vos acquis grâce à un quizz' and 'Description' (Description) containing 'Vous avez bien avancez, procédez au quizz. Vos réponses et votre résultat seront enregistrés.'. There's also a checked checkbox for 'Affichage de tous les questions d'un coup' (Display all questions at once). A 'Valider le titre et la description' (Validate title and description) button is at the bottom right. Below this, a section for 'Question N° 1' is shown with the question 'À quoi servent les directives de préprocesseur, vous savez, ces lignes qui commencent par #include ?' and four options: 'Elles demandent d'inclure des fichiers au projet, c'est-à-dire d'ajouter des fichiers pour la compilation.', 'Elles représentent des fonctions et rassemblent plusieurs commandes demandées à l'ordinateur.', 'Elles permettent d'allouer la mémoire nécessaire pour le programme lors de la compilation.', and 'Elles permettent d'inclure le compilateur dans le projet afin de créer un exécutable.' The first option is marked as 'Bonne réponse' (Correct answer).

Figure 77: Création d'un quizz

L'enseignant dispose d'une page consacrée à la consultation des notifications. Il y a deux sections, l'un pour les notifications qui lui sont adressé et l'autre pour les notifications que l'enseignant a créées. Les notifications dont le font est gris sont les notifications non consultées.

La page 78 présente l'affichage des notifications qui sont adressé à l'enseignant.

The screenshot shows the 'Mes annonces' (My announcements) section of the iTeam-School interface. It lists three notifications from 'Véronique RAKOTOARTISOA': 1. 'Réunions des enseignants le lundi 5 décembre à 14h.' published on '27 novembre 2022 à 19:22'. 2. 'A tous les enseignants. Veuillez envoyer des notifications de bienvenue à tous les premières années.' published on '27 novembre 2022 à 19:21'. 3. Another notification from the same user, which is partially visible. The notifications are displayed in green boxes, while the recipient notifications are in grey boxes.

Figure 78: Consultation des notifications adressées à l'enseignant

8.3.3. Pour les étudiants

La figure 79 présente la consultation des notifications par les étudiants. Il y a deux sections, l'un pour l'affichage de toutes les notifications et l'autre pour les notifications non-consultés. Les notifications dont le font est gris sont les notifications non consultées. Il y a plusieurs types de notifications : pour les nouveaux cours, les devoirs, les tests, et les annonces.

The screenshot shows the 'Mes annonces' section of the iTeam-School platform. At the top, there are two buttons: 'Tout afficher' (Show all) and 'Afficher les non-consultés' (Show unconsulted). Below these, there are three notification cards:

- Announce**: Published by Onja RABE on November 27, 2022, at 16:17. Content: Bonjour à tous, Je tiens à vous souhaiter la bienvenue à l'Ecole. Je suis honoré de faire partie de votre corps enseignant qui va vous apporter les connaissances nécessaires en terme de développement.
- Cours**: Published by Onja RABE on November 27, 2022, at 16:13. Content: Nouveau cours intitulé Développement Web.
- Cours**: This card is partially visible below the second one.

Figure 79: Consultation des notifications adressées à l'étudiant

La figure 80 et la figure 81 présentent la page pour la consultation d'un cours.

The screenshot shows the course details for 'Langage C'. The course is presented by Vony RAKOTOARISOA, an administrator of the platform. The course is divided into 3 parts. The main content area features a large blue hexagon with a white 'C' logo, a laptop displaying code, and a coffee cup.

Langage C
Présenté par :
Vony RAKOTOARISOA
Administrateur de la plateforme
Ce Cours est divisé en 3 Partie composées par des chapitres.

Installez les outils nécessaires pour programmer
Trois outils sont essentiels pour pouvoir programmer: un éditeur de texte, un compilateur et un débogueur.

Strict minimum

Figure 80: Consultation d'un cours - Partie I

The screenshot shows a course consultation interface. On the left, there's a sidebar with 'iTTeam-School' logo, 'Tableau de bord', 'ANNONCES', and 'Mes annonces'. The main content area has a header 'Installez les outils nécessaires pour programmer' with a note about essential tools: text editor, compiler, debugger. It includes sections for 'Strict minimum', 'Lien pour les téléchargements', and 'Installons ensemble'. Below this is 'Ecrivez votre premier programme' with steps: 'Tapez un minimum de code', 'Commentez votre code', 'Renforcez vos acquis', and 'Testez vos compétences'. The final section is 'Découvrez les variables' with a note about variables being the most important concepts in programming. A 'Déclarer les variables' step is shown.

Figure 81: Consultation d'un cours - Partie 2

La page 82 présente la section pour rendre un devoir.

This screenshot shows a 'Testez vos compétences' section with a note to test skills with a task and submit the file for progress. It also shows a 'Votre devoir' section with a 'Ajouter un fichier' button. The interface features a Microsoft Word icon and the text 'Testez vos compétences'.

Figure 82: Section pour rendre un devoir

L'étudiant peut accéder à une page dédiée au quizz d'un cours.

Il y a deux sortes d'affichage :

- l'un où toutes les questions s'affichent d'un coup,
- l'autre où les questions s'affichent un à un.

La figure 83 illustre la page dédiée au quizz de type afficher toutes les questions d'un coup.

Question N° 1

À quoi servent les directives de préprocesseur, vous savez, ces lignes qui commencent par #include ?

- Elles demandent d'inclure des fichiers au projet, c'est-à-dire d'ajouter des fichiers pour la compilation.
- Elles représentent des fonctions et rassemblent plusieurs commandes demandées à l'ordinateur.
- Elles permettent d'allouer la mémoire nécessaire pour le programme lors de la compilation.
- Elles permettent d'inclure le compilateur dans le projet afin de créer un exécutable.

Question N° 2

Dans le programme ci-dessous, comment appelle-t-on les lignes qui se trouvent entre les accolades {} ?

```
int main()
{
    printf("Hello world!\n");
    return 0;
}
```

- Ce sont des variables
- Ce sont les directives de préprocesseur.

Figure 83: Page pour traiter un quizz dont les questions s'affichent d'un coup

La figure 84 expose la page dédiée au quizz dont le type est d'afficher les questions un à un.

À quoi servent les directives de préprocesseur, vous savez, ces lignes qui commencent par #include ?

- Elles demandent d'inclure des fichiers au projet, c'est-à-dire d'ajouter des fichiers pour la compilation.
- Elles représentent des fonctions et rassemblent plusieurs commandes demandées à l'ordinateur.
- Elles permettent d'allouer la mémoire nécessaire pour le programme lors de la compilation.
- Elles permettent d'inclure le compilateur dans le projet afin de créer un exécutable.

Suivant →

Figure 84: Page pour traiter un quizz dont les questions s'affichent un à un

8.3.4. Accessibilité via Bot Messenger

Nous avons parlé plutôt de la nécessité de faciliter l'accès des étudiants à la plateforme, surtout en termes de connexion. De ce fait, les cours pourront être accessibles aux étudiants via bot Messenger.

La figure 85 présente les premières interactions avec le bot. Ces interactions conduiront à la connexion avec un compte étudiant existant.

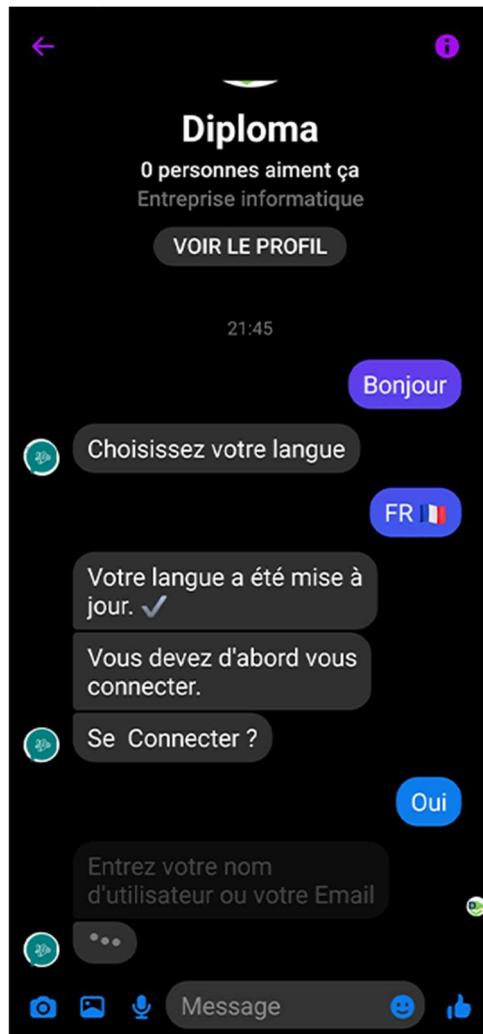


Figure 85: Premières interactions avec le bot messenger

Une fois la connexion réussie, la suite logique est la consultation des cours.

Pour cela, le bot affiche d'abord la liste des cours. Après avoir choisi le cours que nous souhaitons consulter, nous pouvons le parcourir car au fur et à mesure, il nous fait choisir entre les parties et les cours que nous désirons consulter.

La figure 86 et la figure 87 illustrent la consultation d'un cours.

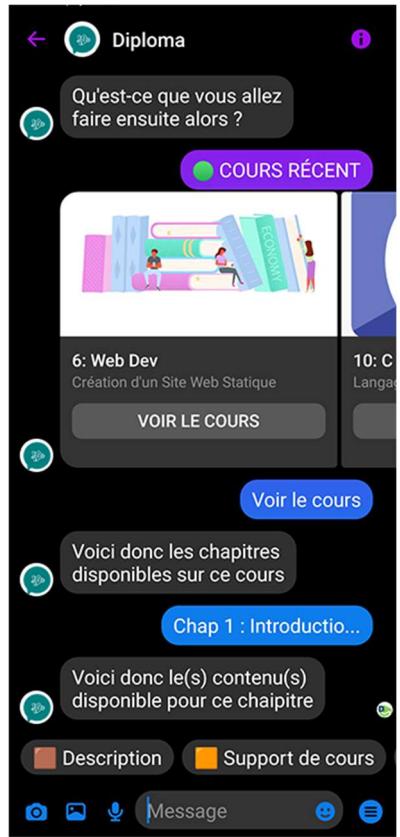


Figure 86: Consultation d'un cours

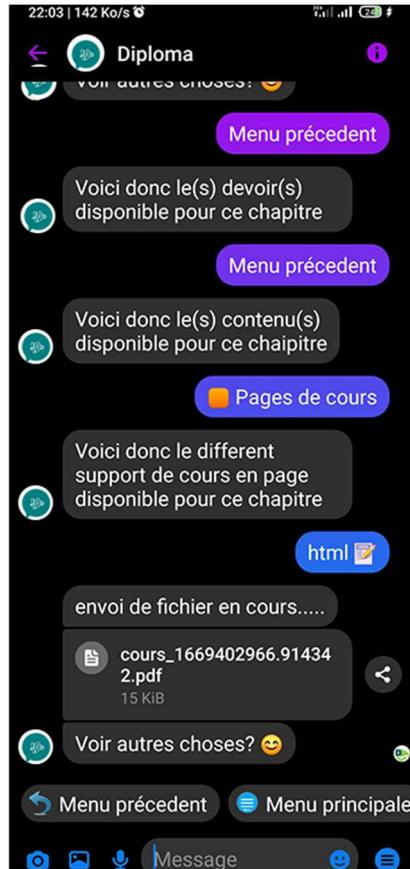


Figure 87: Consultation des supports pédagogiques

CONCLUSION

Cet ouvrage est le fruit d'un stage qui s'est déroulé dans une entreprise nommée « i-Team School », abrégé en ITS, dont l'objectif principal est de prendre part activement à la mise en place de l'enseignement en ligne à Madagascar. Pour cela, une plateforme a été conçue afin de mettre en ligne une plateforme d'apprentissage en ligne pour chaque institut, ainsi qu'un bot messenger par lesquels les étudiants pourront accéder à ces différents documents pédagogiques. Ainsi, le projet consiste à fournir une plateforme intuitive, puissante, sécurisée et bien référencié. Pour la réalisation de ce projet, nous avons utilisé la méthode de gestion de projet Agile, le langage de modélisation UML, les frameworks Vuejs et Django, le SGBD Mariadb et l'environnement de développement Visual Studio Code.

La plateforme web permet aux enseignants de partager des cours, des quizz, des devoirs et d'organiser des tests mais également d'analyser l'assiduité des étudiants et leur niveau. Elle permet aux étudiants de récupérer les cours, consulter ses calendriers, d'effectuer des quizz et de rendre des fichiers. Quant aux administrateurs de la plateforme, ils auront la main sur les différentes configurations propres à chaque institut. La plateforme est actuellement opérationnelle et à disposition de certains instituts. Néanmoins, dans la perspective, on envisage de l'améliorer en y intégrant d'autres applications comme une application de gestion de l'institut.

Ce stage nous a permis de familiariser avec la vie d'entreprise et d'approfondir nos connaissances pratiques en matière de développement web et en gestion de projet. En effet, j'ai pu mettre en pratique mes connaissances théoriques acquises durant ma formation et me suis confronté au du monde du travail et au travail d'équipe.

REFERENCES BIBLIOGRAPHIQUES

- [1] RAKOTOASIMBAHOAKA Cyprien Robert, *Méthodes Agiles – SCRUM* , p. 15
- [2] (en) Ken Collier, *Agile analytics : a value-driven approach to business intelligence and data warehousing*, Addison-Wesley, 2012 (ISBN 978-0-321-66954-4 et 0-321-66954-1, OCLC 748597334, lire en ligne [archive]), p. 121
- [3] (en) Alistair Cockburn, *Agile software development : the cooperative game*, Addison-Wesley, 2007 (ISBN 0-321-48275-1, 978-0-321-48275-4 et 978-0-321-48275-4, OCLC 7086 , lire en ligne [archive]), p. 34
- [4] RAKOTOASIMBAHOAKA Cyprien Robert, *Méthodes Agiles – SCRUM* , p. 36
- [5] Développement d'application client/serveur, 2010, Zouhair ELHADARI, p 38

REFERENCES WEBOGRAPHIQUES

[a] <https://www.eni.mg>, consulté le 19 octobre 2022

[b] <https://www.planzone.fr/blog/methodologies-gestion-projet>, consulté le 19 octobre 2022

[c] <https://www.bocasay.com/fr/methode-scrum-benefices-developpements-web/>, consulté le 6 novembre 2022

[d] <https://manurenaux.wp.imt.fr/2013/09/26/comparatif-vous-cherchez-un-modeleur-uml/>, consulté le 13 novembre 2022

[e] <https://www.simform.com/blog/angular-vs-vue/>, consulté le 6 novembre 2022

[f] <https://intellipaat.com/blog/django-rest-framework-vs-nodejs-difference/>, consulté le 8 novembre 2022

[g] <https://www.capterra.lu/compare/179535/186634/sublime-text/vs/visual-studio-code>, consulté le 24 octobre 2022

[h] [https://www.softwareadvice.fr/compare/108300/108159/mariadb/vs/mysql?vs\[\]>=95663](https://www.softwareadvice.fr/compare/108300/108159/mariadb/vs/mysql?vs[]>=95663), consulté le 24 octobre 2022

[i] https://docwiki.embarcadero.com/RADStudio/Sydney/fr/D%C3%A9finition_des_diagrammes_de_classes_UML_1.5 , consulté le 27 octobre 2022

[j] <https://fr.vuejs.org/v2/guide/index.html>, consulté le 4 novembre 2022

GLOSSAIRE

Administrateur : utilisateur chargé de la gestion globale des paramètres de la plateforme de son institut.

Backend : la partie de l'application immergée à l'utilisateur qui est responsable du bon fonctionnement des fonctionnalités de l'application et de la connexion avec la base de données.

Bot Messenger : un chatbot intégré à Facebook Messenger, capable de discuter avec les étudiants afin de leur partager les cours et les supports de cours.

Compte rendu des quizz/devoirs : affichage des listes des élèves qui ont effectué le quizz/ qui ont rendu le devoir. Ce rapport peut s'agir d'une liste par classe ou d'une liste de tous les étudiants qui ont participé.

Frontend : la partie visible de l'application avec laquelle le visiteur interagit.

Pédagogie : ensemble des méthodes et pratiques d'enseignement requises pour transmettre les connaissances et compétences.

Plateforme e-Learning : plateforme sur le type de formation en ligne et à distance qui utilise Internet et les nouvelles technologies digitales, dans le but d'améliorer les processus d'apprentissage.

Supports pédagogiques : les liens, les fichiers et les devoirs mise en ligne par les enseignants dans le cadre de son cours.

Test de bout en bout : une technique utilisée pour vérifier si une application se comporte comme prévu. Le testeur se met à tester les fonctionnalités de l'application avec la manière où nous pourrions manipuler les navigateurs. Il est utilisé pour assurer la non régression de l'application lors des ajouts de nouvelles fonctionnalités.

Test unitaire : une méthode permettant de vérifier le bon fonctionnement d'une partie de l'application. Il est codé en premier lieu et défini les besoins dans le fonctionnement d'une fonction. De ce fait, nous pourrions vérifier si le module codé par la suite fonctionne comme nous l'avons conçue.

Utilisateur : un internaute qui se connecte à la plateforme.

Quizz : questionnaire permettant de tester des compétences de l'étudiant.

TABLE DES MATIERES

CURRICULUM VITAE	I
SOMMAIRE	II
REMERCIEMENTS	V
LISTE DES TABLEAUX.....	VI
LISTE DES FIGURES.....	VII
LISTE DES ABREVIATIONS	XI
INTRODUCTION GENERALE.....	1
PARTIE I : PRESENTATIONS	
Chapitre 1 : Présentation de l'Ecole.....	2
1.1. Information d'ordre générale	2
1.2. Missions et historique	2
1.3. Organigramme institutionnel de l'ENI	4
1.4. Domaine de spécialisation	5
1.5. Architecture des formations pédagogiques.....	5
1.6. Relations de l'ENI avec les entreprises et les organismes.....	8
1.7. Partenariat au niveau international.....	9
1.8. Débouchés professionnels avec des diplômés	11
1.9. Ressources humaines	13
Chapitre 2 : Présentation de i-Team School	14
2.1. Présentation générale	14
2.2. Historique.....	14
2.3. Services offerts par l'entreprise	15
2.3.1. Mise en place d'une plateforme Moodle	15
2.3.2. Mise en place d'un bot messenger	15

2.3.3. Accompagnement personnalisé et formation sur l'utilisation de la plateforme	15
2.4. Organigramme institutionnel de l'entreprise	15
Chapitre 3 : Description du projet.....	17
3.1. Formulation.....	17
3.2. Objectif et besoins de l'utilisateur	17
3.2.1. Objectif	17
3.2.2. Besoins de l'utilisateur.....	17
3.3. Moyens nécessaires à la réalisation du projet.....	18
3.3.1. Moyens humains	18
3.3.2. Moyens matériels	19
3.3.1. Moyens logiciels	20
3.4. Résultats attendus.....	21
3.5. Chronogramme de travail	21

PARTIE II : ANALYSE ET CONCEPTION

Chapitre 4 : Analyse préalable	22
4.1. Analyse de l'existant.....	22
4.1.1. Organisation actuelle	22
4.1.1. Inventaires logiciels et matériels.....	23
4.2. Critique de l'existant.....	23
4.3. Conception avant-projet.....	24
4.3.1. Proposition de solutions.....	24
4.3.2. Avantages et inconvénients de chaque solution.....	24
4.3.3. Choix de la solution et justification	25
4.4. Méthode et outils utilisés	25
4.4.1. Méthode de gestion de projet.....	25

4.4.2. Choix de l'outil de modélisation.....	28
4.4.3. Choix des langages de programmation.....	29
4.4.4. Choix de l'éditeur de texte.....	31
4.4.5. Choix du SGBD	31
Chapitre 5 : Analyse conceptuelle	33
5.1. Dictionnaire de données.....	33
5.2. Règles de gestion	41
5.3. Désignation des rôles de l'équipe SCRUM	46
5.4. Etapes de l'élaboration du Product Backlog.....	46
5.4.1. Vision du projet.....	47
5.4.2. Liste des acteurs.....	47
5.4.3. Thèmes.....	48
5.4.4. Vision de la première release.....	49
5.4.5. User Story	50
5.5. Product Backlog.....	51
5.6. Sprint Backlog	55
5.6.1. Présentation du travail à faire	56
5.6.2. Planification du sprint.....	57
5.7. Modèle du domaine.....	57
5.8. Diagramme de classe de conception pour chaque sprint	59
5.9. Diagramme de classe de conception globale	62
Chapitre 6 : Conception détaillé	64
6.1. Architecture globale du système	64
6.2. Diagramme de séquence de conception.....	65
6.1. Diagramme de paquetage.....	75
6.2. Diagramme de déploiement	76

PARTIE III : REALISATION

Chapitre 7 : Mise en place de l'environnement	78
7.1. Installation et mise en place des outils.....	78
7.1.1. Installation de Visual Paradigm.....	78
7.1.2. Installation de Visual Studio Code	79
7.1.3. Installation de git.....	80
7.1.4. Installation de Vuejs	81
7.1.5. Installation de Django Rest Framework	82
7.1.6. Installation de HeidiSQL	84
7.2. Architecture de l'application.....	84
Chapitre 8 : Développement de l'application	86
8.1. Création de la base de données	86
8.2. Codage de l'application	87
8.2.1. Codage du backend.....	87
8.2.2. Codage du frontend.....	95
8.3. Présentation de l'application.....	106
8.3.1. Pour les administrateurs.....	107
8.3.2. Pour les enseignants.....	110
8.3.3. Pour les étudiants	113
8.3.4. Accessibilité via Bot Messenger.....	115
CONCLUSION	118
REFERENCES BIBLIOGRAPHIQUES	XII
REFERENCES WEBOGRAPHIQUES.....	XIII
GLOSSAIRE	XIV
TABLE DES MATIERES	XVI

RESUME

Succinctement, le projet consiste à créer et mettre en place une plateforme e-Learning qui mettra en relation enseignants et étudiants dans le but d'améliorer les méthodes d'apprentissage en offrant une souplesse, une commodité mais surtout une grande accessibilité à l'éducation. Pour cela, la méthode de gestion de projet appliquée est la méthode Agile, accompagné par la méthode de conception UML, les frameworks VueJS et Django Rest Framework dont le développement est dirigé par des tests, et le système de gestion de bases de données est Mariadb. Les principales fonctionnalités de la plateforme sont la création et de gestion de leurs propres cours par les enseignants et l'accès aux différents supports pédagogiques par les étudiants. Pour la rédaction de cet ouvrage, la phase d'analyse et de conception a été détaillé ; puis, la modélisation des besoins des utilisateurs ciblés, et enfin le développement de l'application.

Mots clés : apprentissage, Django, enseignants, étudiants, plateforme e-Learning, supports pédagogiques, VueJS

ABSTRACT

Briefly, the project consists of creating and implementing an e-Learning platform that will connect teachers and students with the aim of improving learning methods by offering flexibility, convenience but above all accessibility to education. For this, the project management method applied is the Agile method, accompanied by the UML design method, the VueJS and Django Rest Framework whose development is directed by tests, and the database management system is Mariadb. The main features of the platform are the creation and management of their own courses by teachers and the access to the different teaching materials by students. For the writing of this work, the analysis and design phase was detailed; then, the modeling of the needs of the targeted users, and finally the development of the application.

Key words: Django, e-Learning platform, learning, students, teachers, teaching materials, VueJS