

PROJET SYMFONY

plateforme educative



Membres :

Tony SARRE

Khady Faye

Ibrahima N'diaye

Serigne Soukoum Foncéka

31 JANVIER 2021

TONY SARRE

sarre.tony@ugb.edu.sn

Sommaire

Introduction

I. DESCRIPTION DU PROJET

1. Exigences fonctionnelles
2. Exigences non fonctionnelles

II. ANALYSE ET CONCEPTION

1. Spécification des exigences : les cas d'utilisations
2. Analyse du domaine
 - Le model conceptuel de donnée
 - Le diagramme de classe
 - Le model logique de donnée

III. REALISATION TECHNIQUE

1. Etude technique
 - Matériel de base
 - Outils de développement
 - Langage de développement
2. Architecture générale de l'application
3. Présentation des interfaces
 - Scenario « Administrateur »
 - Enseignant
 - Etudiant

IV. ORGANISATION ET BILAN DU PROJET

1. Organisation du groupe : méthode de travail
2. Résultat
 - Apports
 - Bilan personnel

INTRODUCTION :

L'Internet a révolutionné le monde des ordinateurs et des communications comme rien d'autre auparavant. En effet, elle est à la fois une capacité de diffusion dans le monde entier, un mécanisme de distribution de l'information et un moyen de collaboration et d'interaction entre les individus et leurs ordinateurs, peu importe l'emplacement géographique.

Par ailleurs, l'explosion des médias sociaux modifie profondément internet et ses usages.

S'il est courant pour des adultes d'utiliser à titre personnel ou professionnel les médias sociaux, il en va autrement dans les milieux éducatifs, la formation continue ou professionnelle. Et pour servir mieux encore l'organisation, elle se doit d'être à la hauteur des ambitions qu'elle porte : rapidité, flexibilité, efficacité et rentabilité. Pour y faire face, le système traditionnel de formation par son approche présentielle, conditionné par l'unité de temps et de l'espace semble être insuffisant pour répondre à ce grand défi d'où la nécessité d'une nouvelle approche qui est le social Learning.

Les enseignants, n'ont plus le monopole de la transmission « L'apprenant 2.0 » ne se limite pas à utiliser Internet pour consulter des pages web ; s'il a commencé par échanger du courriel et des messages textes, il exploite maintenant les possibilités qu'offre le « World Wide Web » dans les deux sens, consultation et diffusion.

Bien que l'enseignement par internet (ou télé-enseignement ou e-Learning) souffre de carences principalement liées à l'absence relative de l'enseignant et donc à la difficulté d'adaptation de l'enseignement au niveau et au comportement de l'apprenant, il constitue un des moyens pédagogiques actuels et prometteurs.

Le social Learning devrait venir compléter l'offre existante de formations classiques, présentiels et e-learning. L'un des problèmes du e-learning est bien souvent, l'arrêt en cours de formation. Les apprenants ne vont pas au bout de leur apprentissage. En effet, seuls face à leurs écrans, ils ne trouvent pas la motivation pour s'améliorer. Le social Learning permet aux apprenants de discuter, d'échanger à propos de la formation et de progresser ensemble. L'important est donc de donner des outils pour lier les gens les uns aux autres afin qu'ils se parlent et s'auto-forment.

Le présent rapport s'articule autour de quatre points. Nous commencerons par la « Description du projet » qui localise le contexte général du notre projet et met l'accent sur le champ d'étude de notre application qui est l'apprentissage social. Le deuxième point « Analyse et spécification des besoins » dédié à l'étude de l'existant où nous présenterons une synthèse des solutions existantes sur le marché en discutant les avantages et les inconvénients de chacune d'elles. Puis, nous formulerons les principales fonctionnalités de la solution à réaliser. A la lumière de cette analyse, nous entamerons

la conception, dans lequel nous présenterons la notation de modélisation utilisée ainsi que l'ensemble des diagrammes conçus. A le troisième point, la « Réalisation technique » qui sera consacré à l'étude technique où nous détaillerons notre environnement de travail suivis d'une présentation des différentes fonctionnalités de notre plateforme « Social Learning » à travers des capture d'écran. Et pour conclure nous ferons le bilan du projet.

I. DESCRIPTION DU PROJET

Le système consiste à mettre en place une plateforme sociale Learning pour le compte des personnels de l'état qui souhaitent bénéficier d'un système perspicace.

Dans un secteur où les contraintes de formation sont importantes (disponibilité, respect des règlements intérieurs, etc.), le concept d'une plateforme offre des réponses complémentaires et une plus grande flexibilité à l'apprenant qui pourra se former à son rythme en fonction de ses disponibilités.

Notre objectif consiste à créer une plateforme du social Learning qui répond au besoin du l'enseignant et des apprenants à la fois et de manière à :

- Favoriser une pédagogie socioconstructiviste (activités, transparence, etc.),
- Garantir plus de souplesse pour la planification des formations à l'aide d'une interface ergonomique, légère, efficace, compatible -sans nécessité d'un navigateur récent- et simple à manipuler pour les différents utilisateurs,
- Faciliter l'accès aux ressources et services,
- Offrir aux apprenants un espace d'apprentissage individualisé,
- Faciliter le partage, la production, la communication.

1. Exigences Fonctionnelles

Les besoins fonctionnels représentent les attentes de chaque acteur de la future plate-forme. Toute solution conceptuelle doit satisfaire, préalablement, à des besoins fonctionnels afin de délimiter le périmètre fonctionnel de l'application et surveiller la traçabilité des besoins lors de la phase de développement.

La présente plateforme doit satisfaire les besoins fonctionnels suivants :

- La gestion des utilisateurs,
- La gestion des emplois du temps,
- La gestion des cours,
- La gestion des notes.

2. Exigences non fonctionnelles

Un besoin non fonctionnel est une restriction ou une contrainte qui pèse sur un service du système, telle les contraintes liées à l'environnement et à l'implémentation et les exigences en matière de performances.

Les besoins non fonctionnels :

- Le code doit être clair pour permettre des futures évolutions ou améliorations,
- La plateforme doit fournir un accès rapide aux informations, et doit faire la mise à jour en temps réel,
- La plateforme doit être portable, extensible, réutilisable et fiable,
- La plateforme offre une interface conviviale et facile à utiliser,
- La plateforme doit garantir la confidentialité, l'intégrité et la cohérence des données.

II. ANALYSE ET CONCEPTION

Dans le cadre de notre projet, nous avons opté pour le langage UML comme un langage de conception. UML (Unified Modeling Language, traduisez « langage de modélisation objet unifié ») est né de la fusion des trois méthodes qui ont le plus influencé la modélisation objet au milieu des années 90 : OMT, Booch et OOSE [15].

1. SPECIFICATION DES EXIGENCES : les cas d'utilisations

Un cas d'utilisation définit une manière d'utiliser le système et permet d'en décrire les exigences fonctionnelles. Chaque cas d'utilisation contient un ou plusieurs scénarios qui définissent comment le système devrait interagir avec les utilisateurs (appelés acteurs) pour atteindre un but ou une fonction spécifique d'un travail. Un acteur d'un cas d'utilisation peut être un humain ou un autre système externe à celui que l'on tente de définir. Il permet d'adopter le langage de l'utilisateur final ou de l'expert du domaine. Chaque cas d'utilisation est représenté au sein d'un diagramme de cas d'utilisation.

Acteurs	Cas d'utilisation
Administrateur	<ul style="list-style-type: none">• Se connecter (saisir nom d'utilisateur, saisir mot de passe, changer nom d'utilisateur, changer mot de passe, ajouter photo)• Gérer la plate-forme (ajouter classe, ajouter matière, les composants, emploi du temps, etc.)• Gérer les utilisateurs (Ajouter, supprimer, gérer les droits d'accès, etc.)

	<ul style="list-style-type: none"> • Gérer les notes (Ajouter, supprimer, etc.)
Enseignant	<ul style="list-style-type: none"> • Se connecter (saisir nom d'utilisateur, saisir mot de passe, changer nom d'utilisateur, changer mot de passe, ajouter photo) • Gérer les supports de cours (ajouter, modifier, gérer leurs droits d'accès, etc.) • Consulter la liste des étudiants (noter présence et absence) • Gérer l'espace de dépôt (déposer, modifier, gérer les droits d'accès, gérer la date d'ouverture et date limite)
Etudiant	<ul style="list-style-type: none"> • Se connecter (saisir nom d'utilisateur, saisir mot de passe, changer nom d'utilisateur, changer mot de passe, ajouter photo) • Consulter support de cours (télécharger, consulter) • Consulter note (faire son bulletin semestriel) • Consulter ponctualité (présence, absence)

Tableau 1 : Identification des cas d'utilisation

Une fois les différents cas d'utilisation identifiés, nous pouvons élaborer le diagramme des cas d'utilisation qui met en relation les acteurs et les cas d'utilisation qu'ils utilisent.

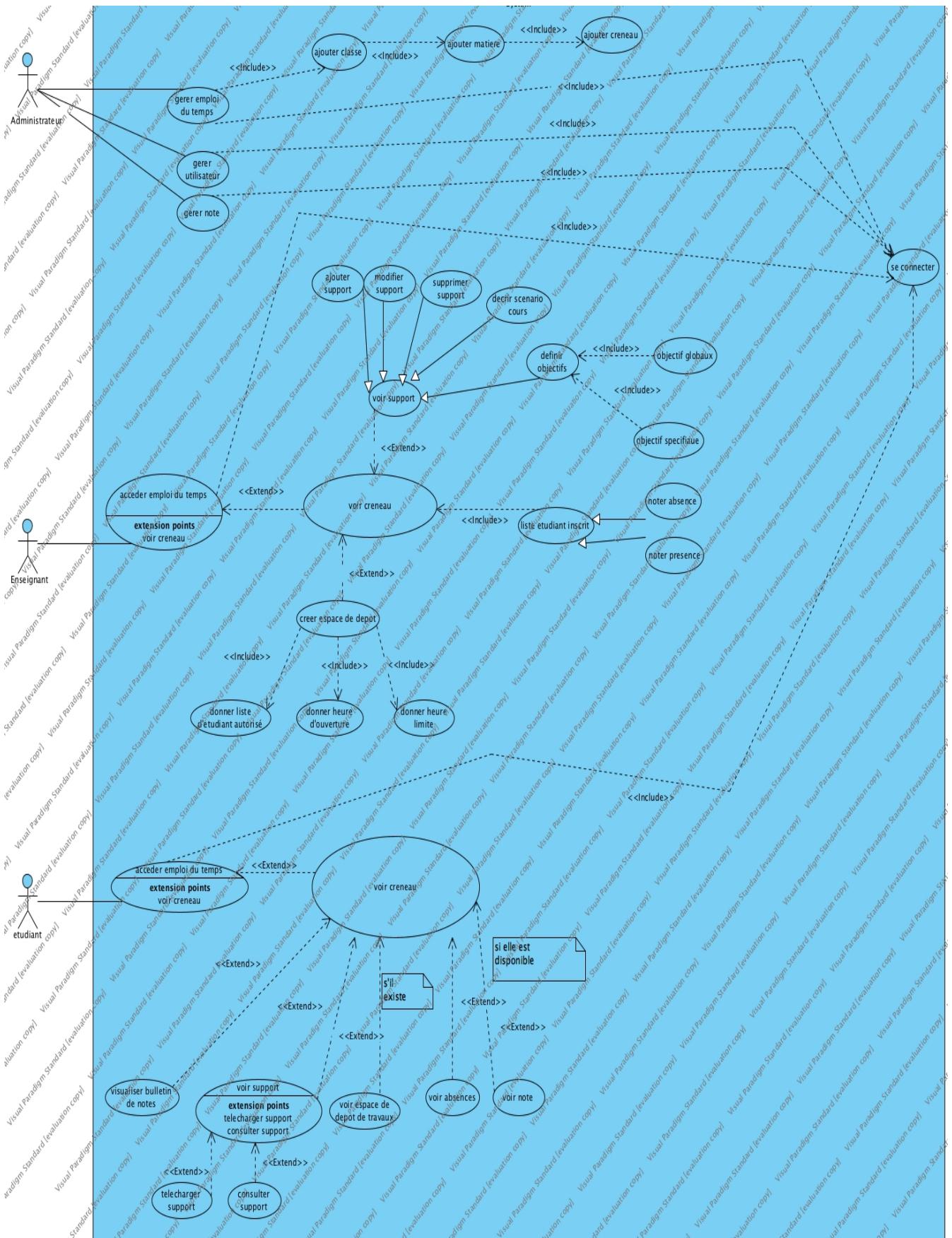


Figure1 : Diagramme global des cas d'utilisation

Risque :

Afin d'assurer le bon fonctionnement du projet, il faut prendre en considération un certain nombre de contraintes :

- Non maîtrise des outils de développement : le fait de ne pas maîtriser quelques outils de développement nous amène à passer un temps important pour apprendre et maîtriser ces outils.
- Risque de créer une architecture inadaptée

a. Raffinement des cas d'utilisation

Dans le but de mieux comprendre notre système et les interactions avec les utilisateurs, dans cette partie nous allons détailler les scenarios de principaux cas d'utilisation

- Cas d'utilisation : Se connecter :

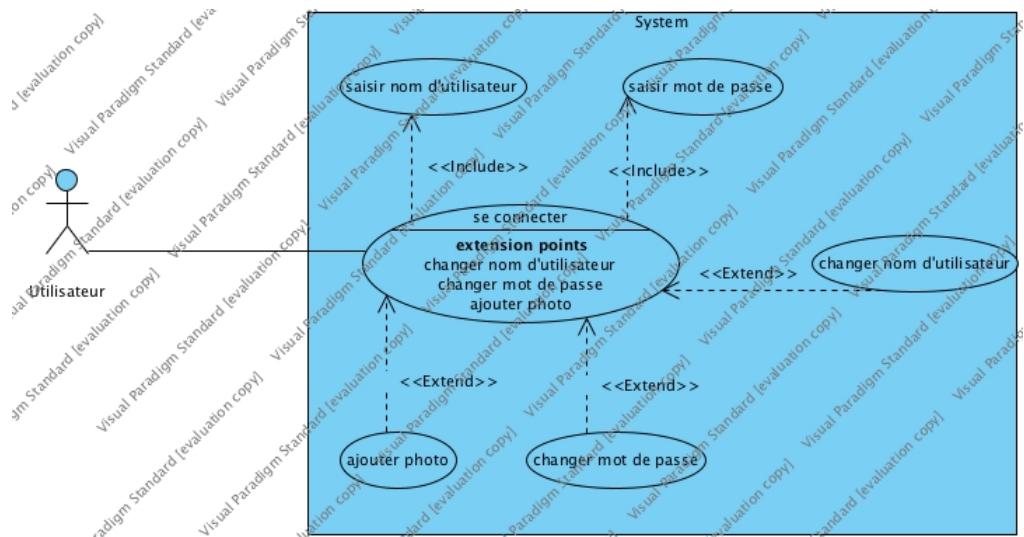


Figure 2 : Cas d'utilisation « Se connecter »

CU1 : S'authentifier
Résumé : Ce CU1 permet à l'acteur d'accéder à son espace personnel
Acteurs : administrateur, tuteur, apprenant
Précondition : L'acteur doit être un membre enregistré dans la plateforme
Post condition : le cas démarre après le point 02 de l'enchaînement nominal.

Scénario nominal :

« DEBUT »

01 : Le système affiche un formulaire de connexion à l'acteur.

02 : l'acteur saisit son nom ainsi que son mot de passe.

03 : Le système vérifie les paramètres.

04 : Le système ouvre l'espace de travail correspondant au profil.

« FIN »

Scénario alternative :

Le login ou le mot de passe est incorrect : ce scénario commence au point 03 du scénario nominal.

01 : Le système informe l'acteur que les données saisies sont erronées et lui demande s'il veut changer son mot de passe.

Diagramme de séquence :

➤ Connection de l'administrateur

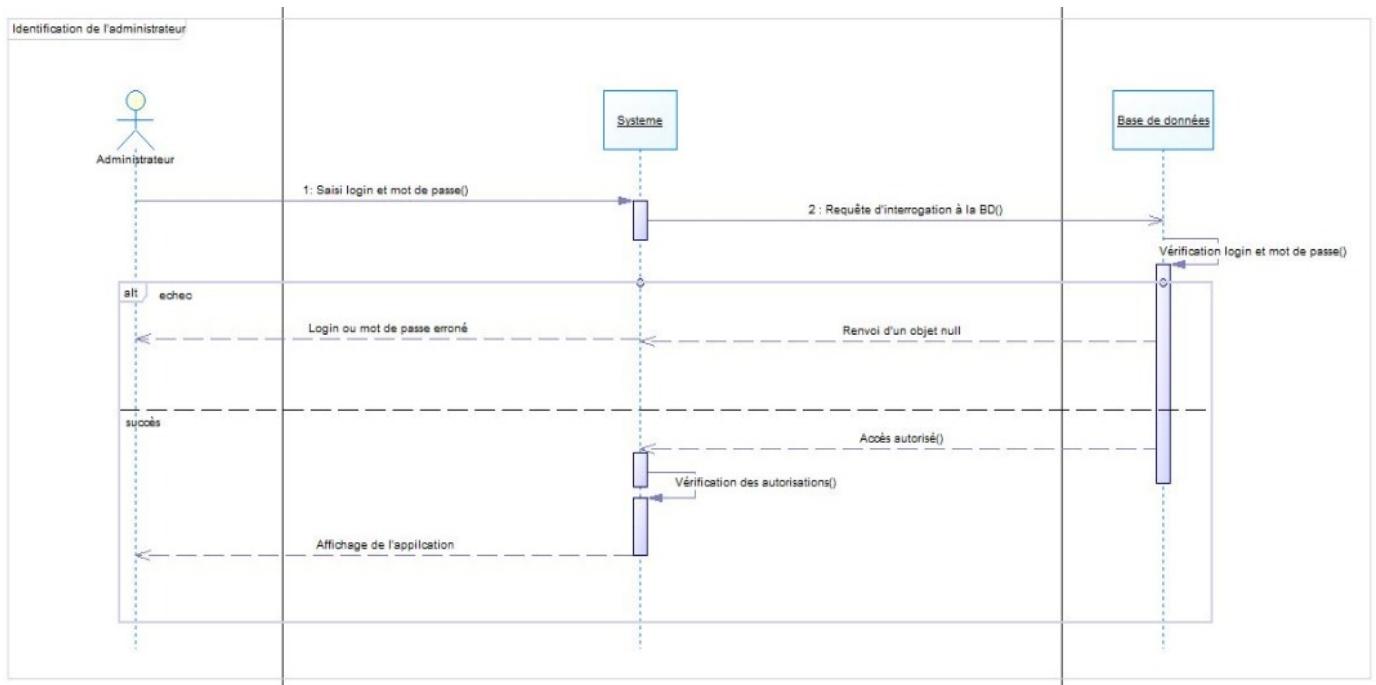


Figure 3 : « Connection de l'administrateur »

➤ Connection de l'enseignant

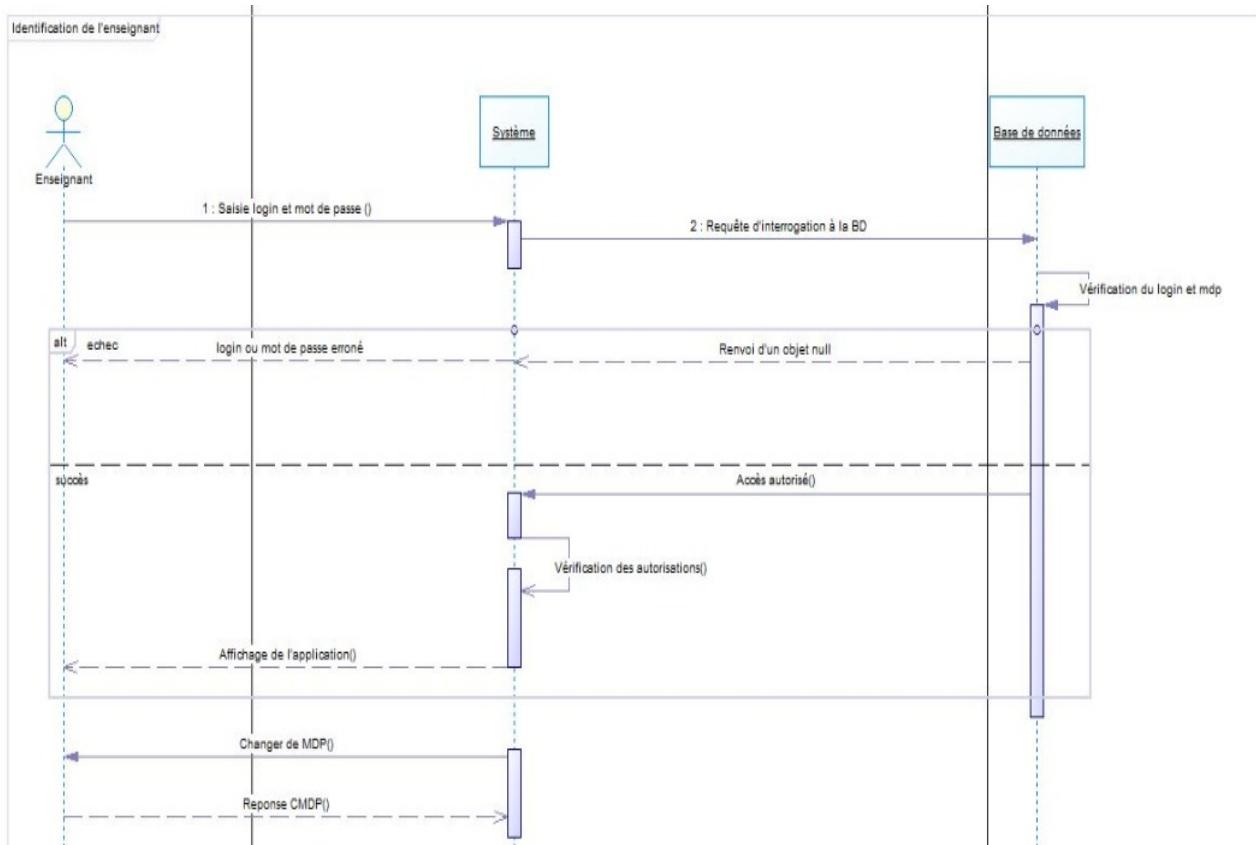


Figure 4 : « Connection de l'enseignant »

➤ Connection de l'étudiant

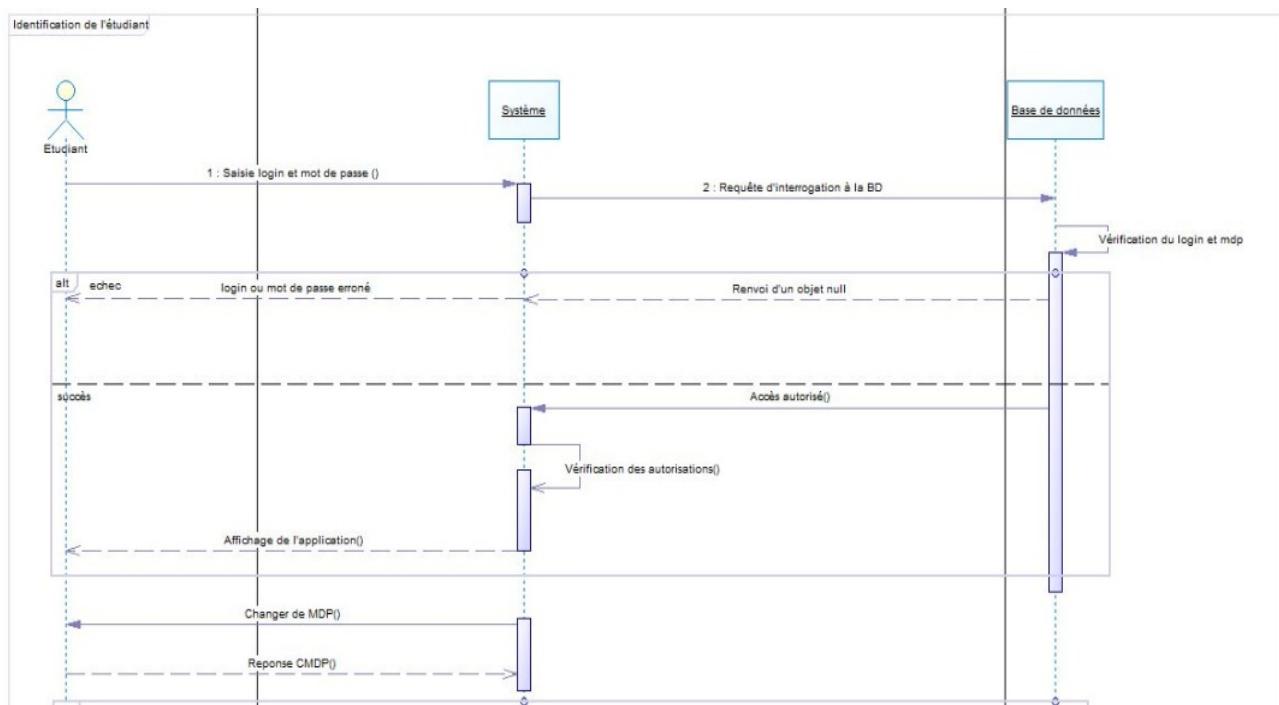


Figure 5 : « Connection de l'étudiant »

➤ Modifier mot de passe

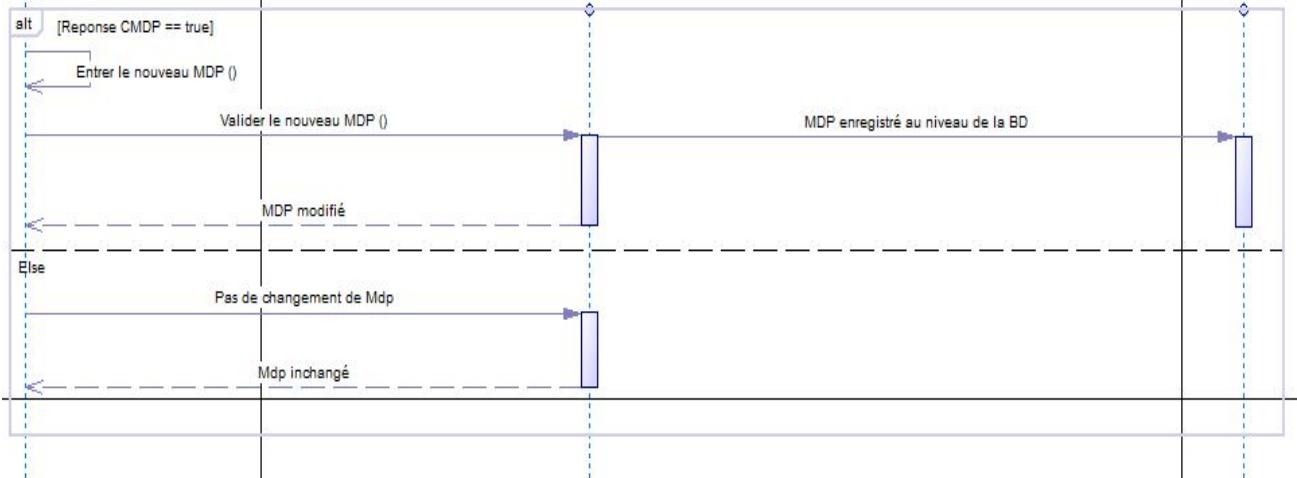


Figure 6 : « Modifier mot de passe »

- Cas d'utilisation : Gérer Emploi du temps :

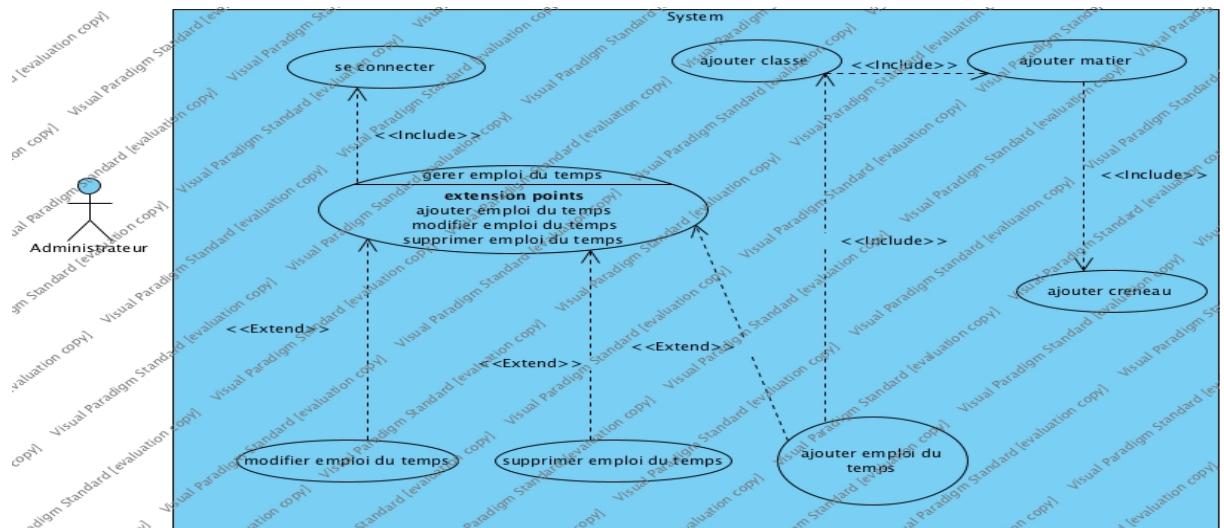


Figure 7 : Cas d'utilisation « Gérer emploi du temps »

CU2 : Gérer emploi du temps
Résumé : Ce CU2 permet à l'administrateur de modifier les compositions internes et externes de l'emploi du temps
Acteurs : administrateur
Précondition : L'acteur doit être connecté

Post condition : le cas démarre après le point 02 de l'enchainement nominal.

Scénario nominal :

« DEBUT »

01 : Le système affiche l'état actuel de la plateforme.

02 : l'acteur met à jour la plateforme.

03 : Le système vérifie la validité des mis à jour.

04 : Le système enregistre les mis à jour dans la base de données.

05 : le système notifie l'acteur du bon déroulement de mis à jour de l'emploi du temps

« FIN »

Scénario alternative :

L'emploi du temps est incorrect : ce scénario commence au point 03 du scénario nominal.

01 : Le système informe l'acteur que les mis à jour sont erronées et lui demande de ressaisir les données.

Diagramme de séquence :

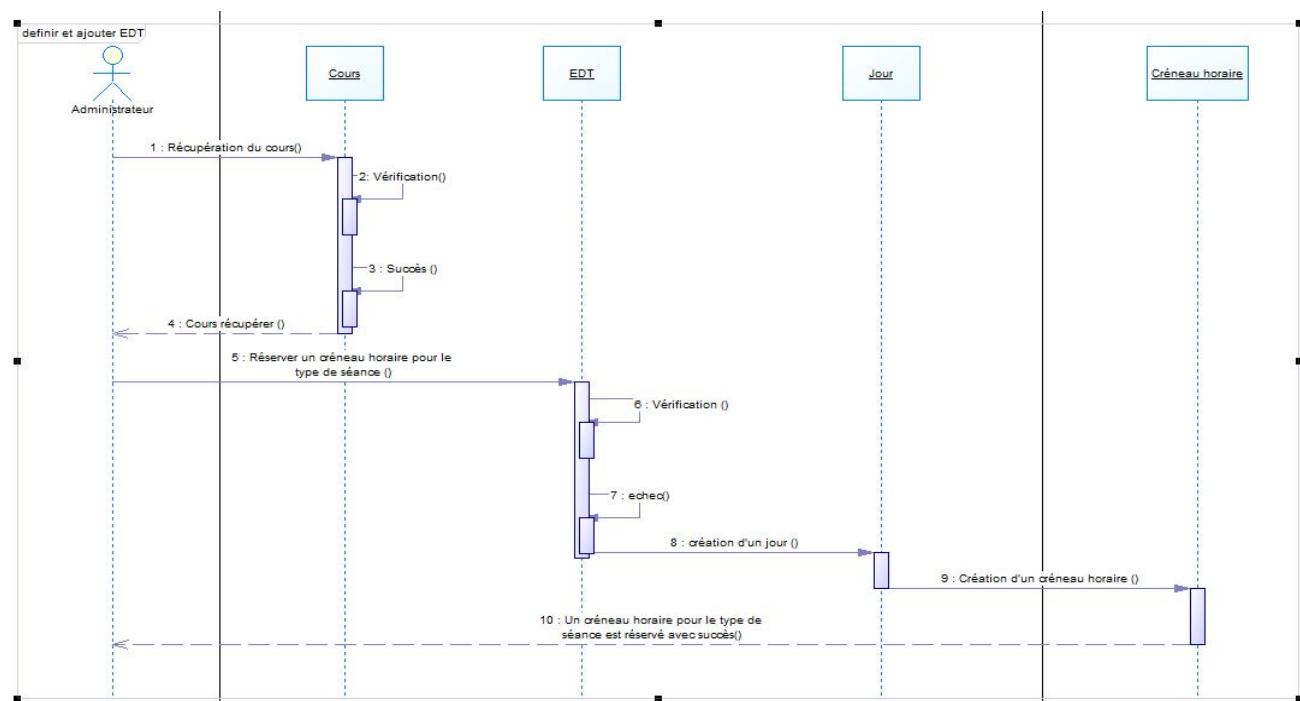


Figure 8 : « Définir et ajouter emploi du temps »

- Cas d'utilisation : Gérer les utilisateurs :

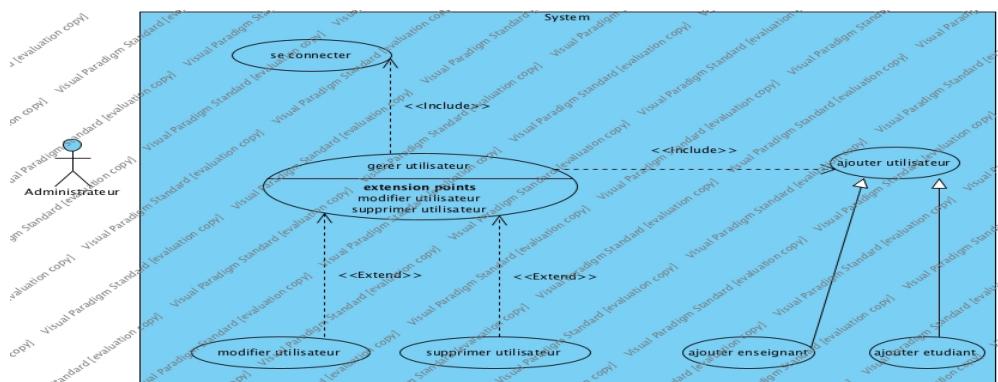


Figure 9 : Cas d'utilisation « Gérer les utilisateurs »

CU3 : Gérer les utilisateurs
Résumé : Ce CU3 permet à l'administrateur d'ajouter, de modifier et de supprimer un utilisateur
Acteurs : administrateur
Précondition : L'acteur doit être connecté
Post condition : le cas démarre après le point 02 de l'enchaînement nominal.
<p>Scénario nominal : Ajouter un utilisateur</p> <p>« DEBUT »</p> <p>01 : Le système affiche une page de connexion.</p> <p>02 : l'acteur saisit les informations du nouvel utilisateur.</p> <p>03 : Le système vérifie la validité des informations saisies.</p> <p>04 : Le système enregistre ces informations dans la base de données.</p> <p>05 : le système notifie l'acteur du bon déroulement de la connexion.</p> <p>« FIN »</p>
<p>Scenario alternative :</p> <p>Les informations sont manquantes ou incorrectes : ce scénario commence au point 03 du scénario nominal.</p>

01 : Le système informe l'acteur que les données saisies sont erronées et le scénario reprend au point 02 du scénario nominal.

Diagramme de séquence : Ajouter Utilisateur

➤ Ajouter enseignant

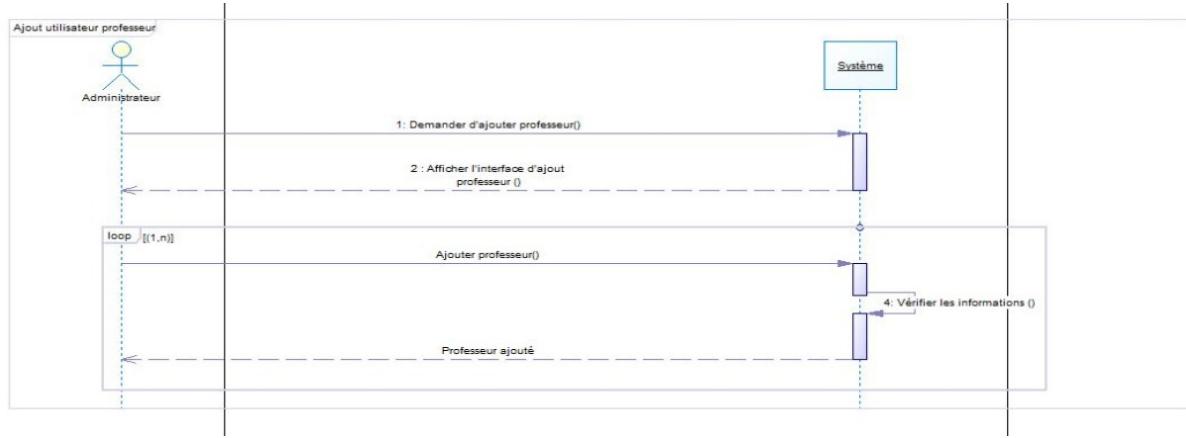


Figure 10 : « Ajouter enseignant »

➤ Ajouter étudiant

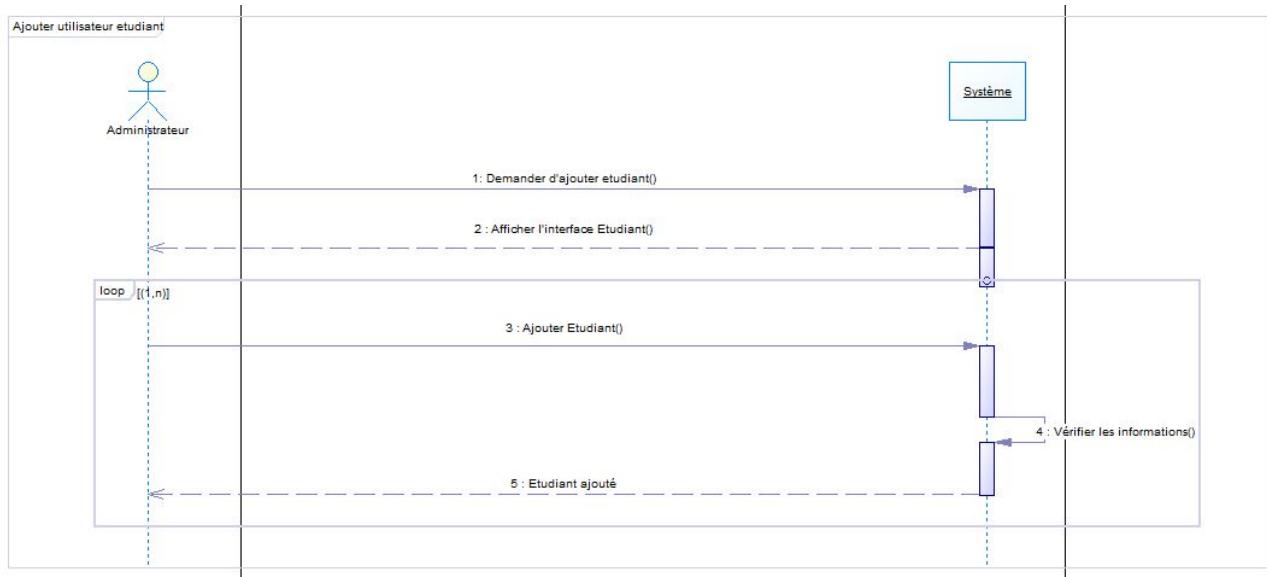


Figure 11 : « Ajouter étudiant »

Diagramme de séquence : Modifier Utilisateur

➤ Modifier enseignant

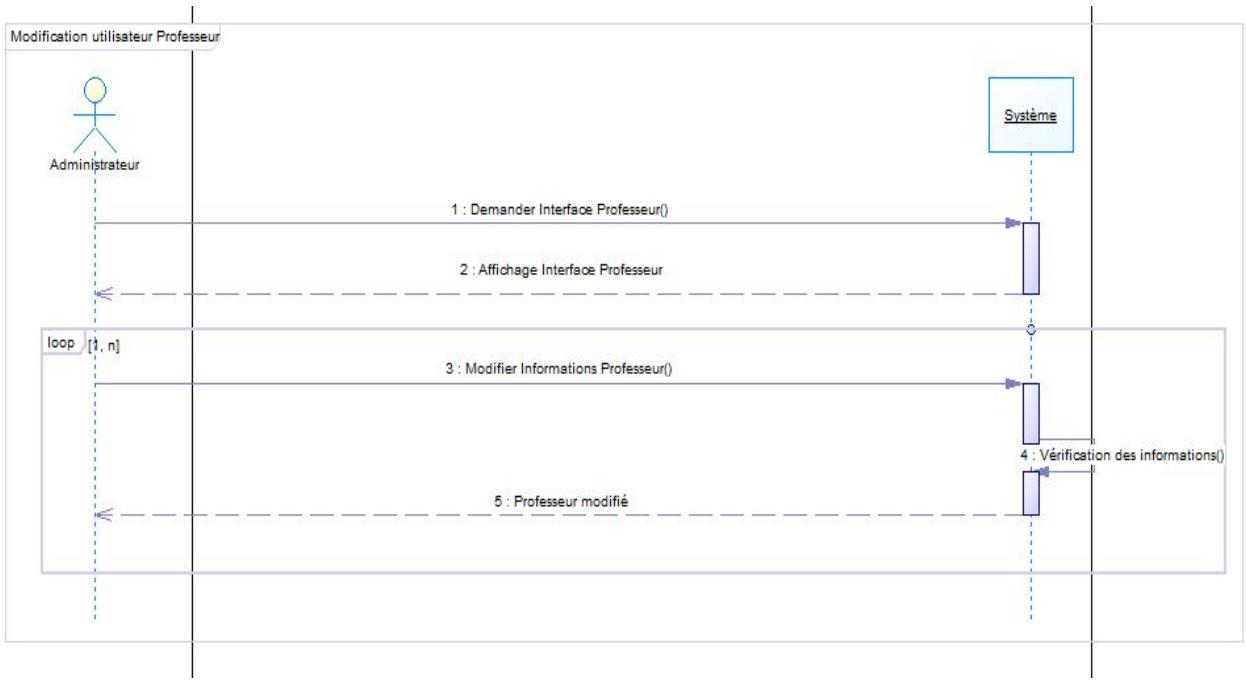


Figure 12 : « Modifier enseignant »

➤ Modifier étudiant

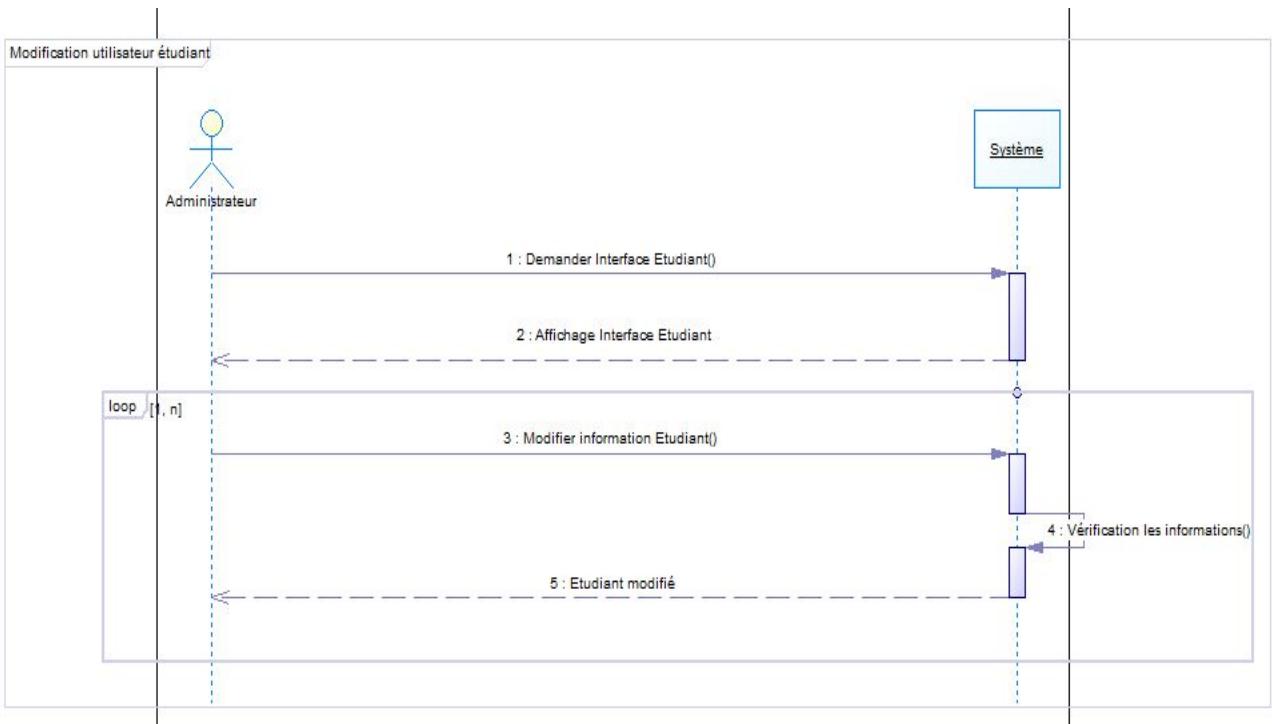


Figure 13 : « Modifier étudiant »

Diagramme de séquence : Supprimer Utilisateur

➤ Supprimer enseignant



Figure 14 : « Supprimer enseignant »

➤ Supprimer étudiant

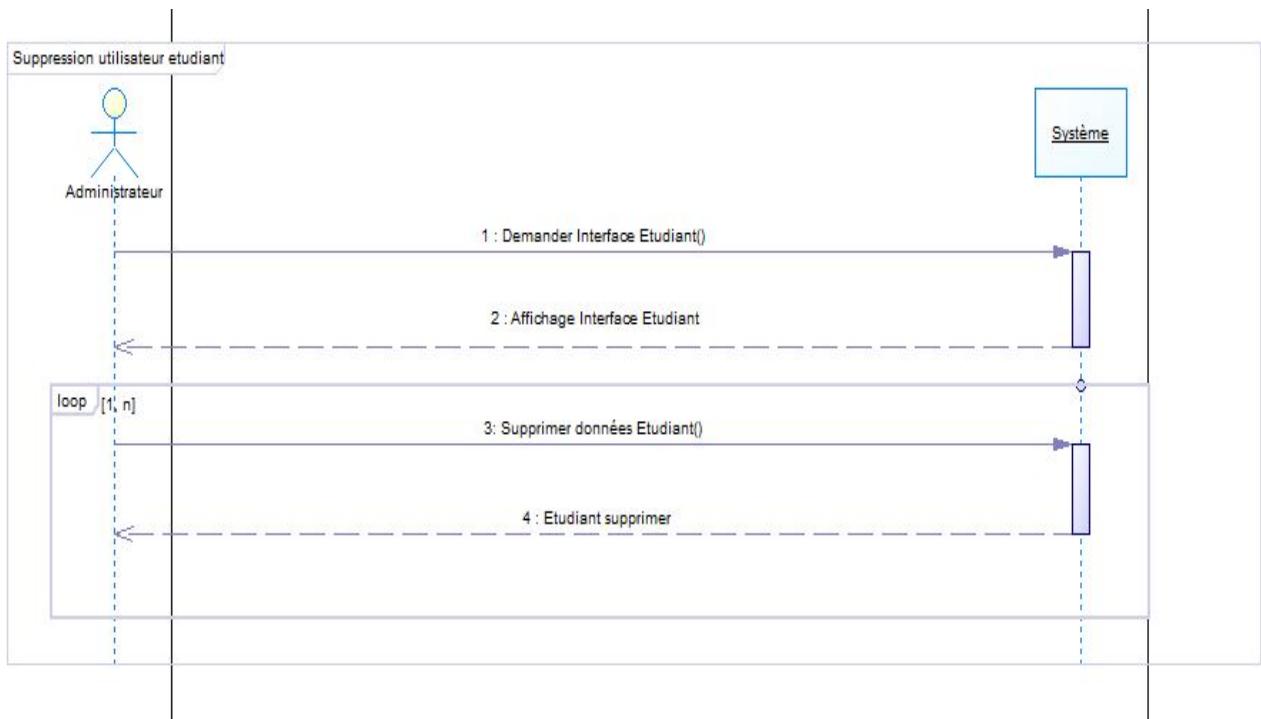


Figure 15 : « Supprimer étudiant »

- Cas d'utilisation : Gérer note :

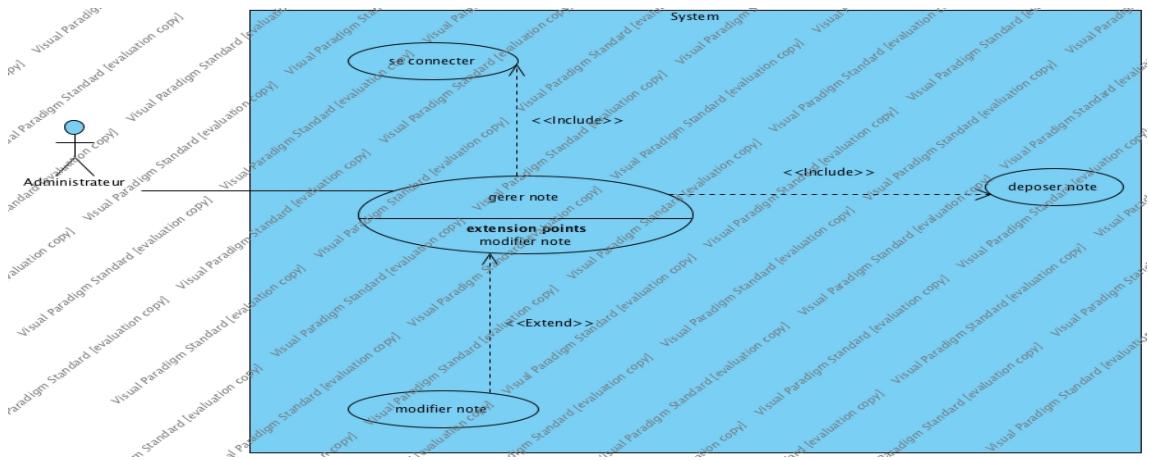


Figure 16 : Cas d'utilisation « Gérer les notes »

Diagramme de séquence :

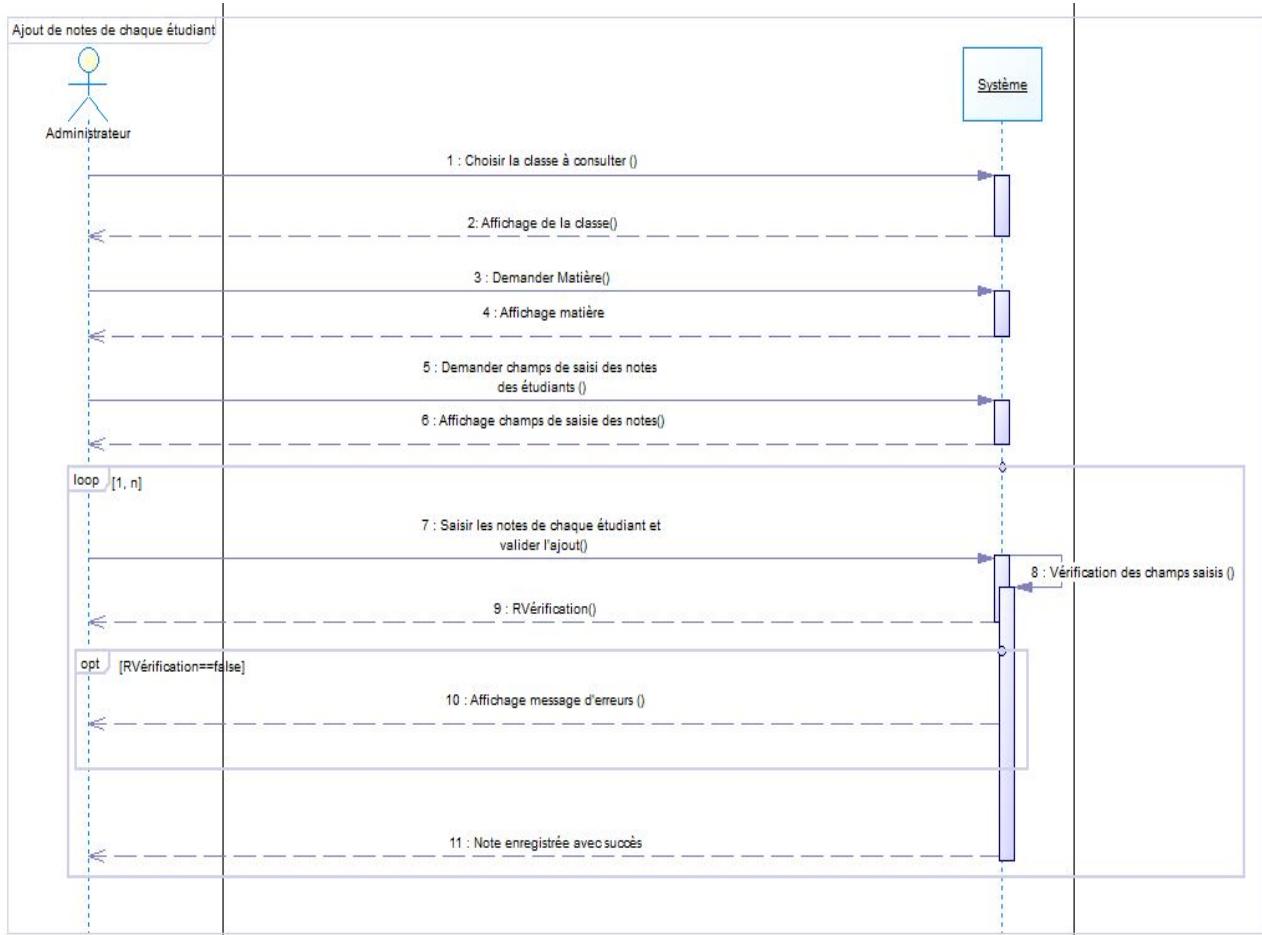


Figure 17 : « note des étudiants »

- Cas d'utilisation : voir créneau :

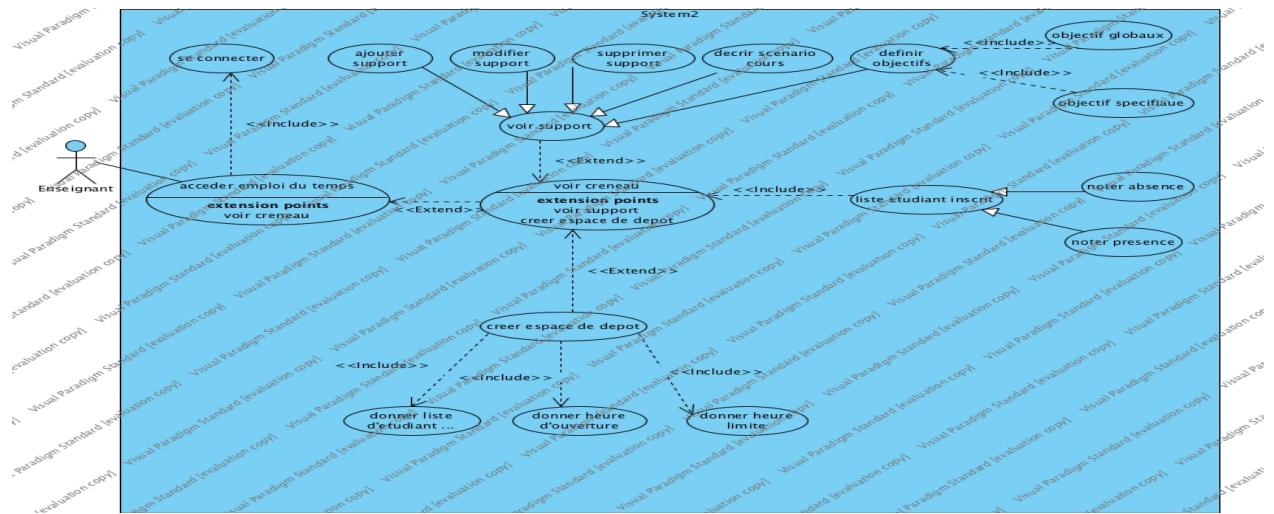


Figure 18 : Cas d'utilisation « Voir créneau »

CU4 : Voir créneau

Résumé : Ce CU4 permet à l'acteur d'ajouter, de supprimer et modifier un support cours. Il permet aussi de décrire un scenario, définir un objectif. il permet également de voir la liste des étudiants inscrits et aussi de créer un espace de dépôt

Acteurs : enseignant

Précondition : L'acteur doit être connecté.

Scénario nominal d'ajout d'un cours :

« DEBUT »

01 : Le système affiche deux méthodes d'ajout d'un cours.

<Méthode1 : déposer support de cours>

02 : L'acteur saisit le contenu du cours.

<Méthode2 : création de l'espace de dépôt>

02 : L'acteur crée un espace de dépôt.

03 : L'acteur configure les droits d'accès à l'espace (liste des étudiants autorisés, heure de début, heure limite).

04 : L'acteur note les présences et absences des étudiants inscrits à ce cours.

« FIN »

Diagramme de séquence :



Figure 19 : créneau enseignant

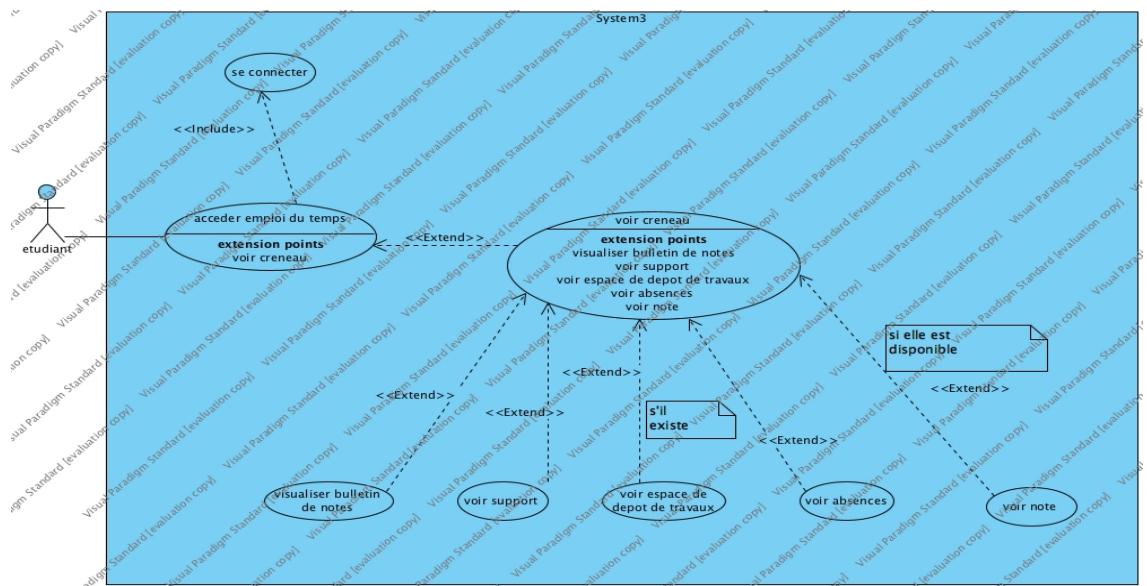


Figure 20 : Cas d'utilisation « Voir crenau »

CU5 : Voir créneau

Résumé : Ce CU5 permet à l'acteur de visualiser ces notes, voir le support, voir espace de dépôt.il peut également faire son bulletin semestriel.

Acteurs : Etudiant
Précondition : L'acteur doit être connecté.
Scénario nominal d'ajout d'un cours :
<p>« DEBUT »</p> <p>01 : Le système affiche deux méthodes d'aperçu du créneau.</p> <p><Méthode1 : voir support de cours></p> <p>02 : L'acteur consulte ou télécharge support de cours.</p> <p><Méthode2 : voir note></p> <p>02 : L'acteur peut éventuellement créer son bulletin.</p> <p>03 : L'acteur peut accéder à l'espace de dépôt s'il est autorisé.</p> <p>04 : L'acteur note ses présences et absences par rapport au cours.</p> <p>« FIN »</p>

Diagramme de séquence :

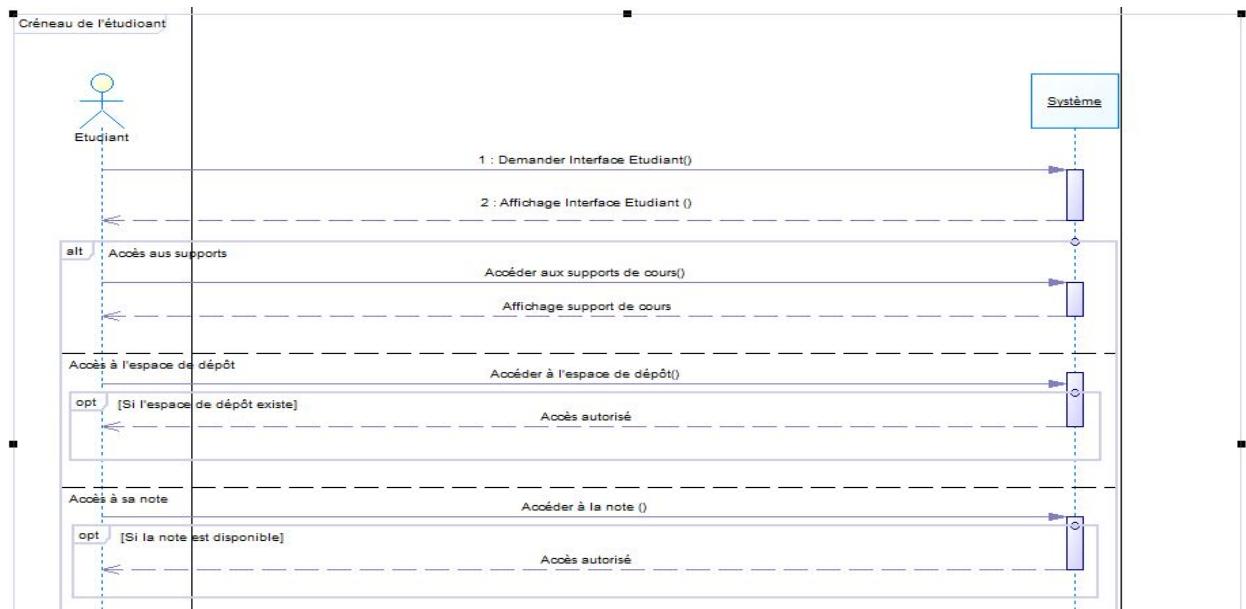


Figure 21 : « Créneau étudiant »

2. ANALYSE DU DOMAINE :

- Le model conceptuel de donnée

La modélisation conceptuelle des données (MCD) donne une vue statique du système permettant de décrire le système d'information à l'aide des concepts proposés par le modèle UML. Il s'agit d'une représentation facilement compréhensible permettant de décrire le système d'information à l'aide d'entités. Ainsi ce modèle conceptuel de données sera régi par certaines orientations et contraintes qui permettront de distinguer les différentes entités qui sont en relation.

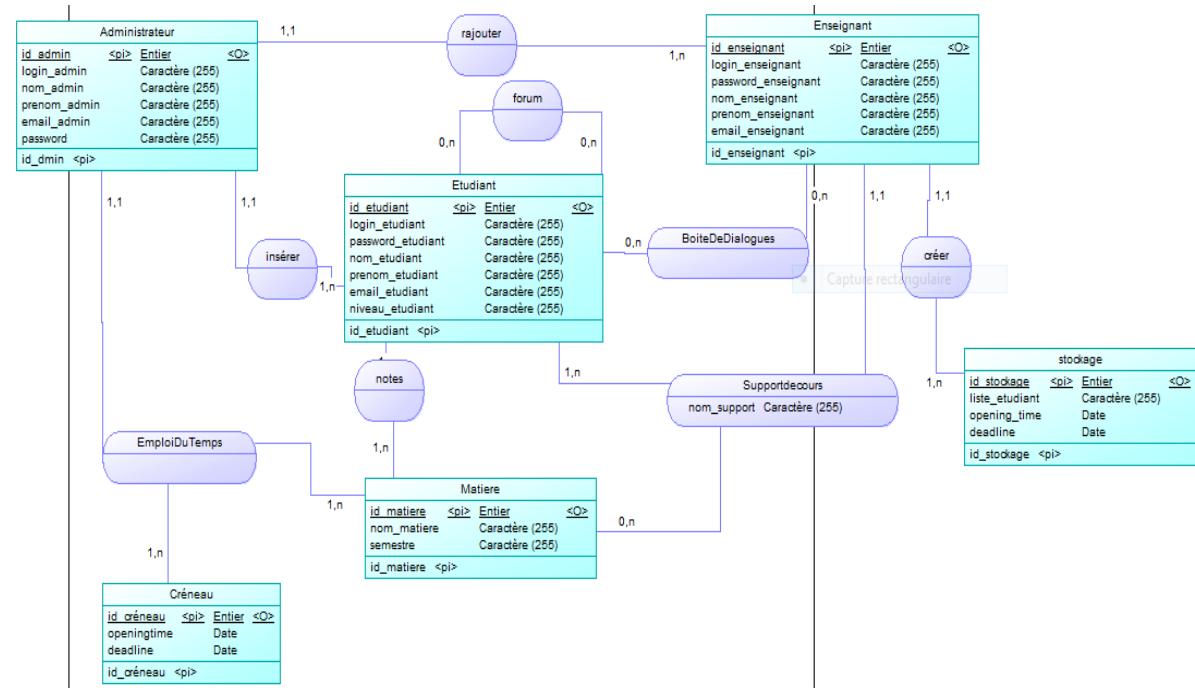


Figure 22 : « Modèle conceptuel de données »

➤ Le diagramme de classe

Le diagramme de classes est le plus important des diagrammes UML, c'est le seul qui soit obligatoire lors de la modélisation objet d'un système. Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que leurs relations. Ce diagramme fait partie de la partie statique d'UML, ne s'intéressant pas aux aspects temporels et dynamiques. Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

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

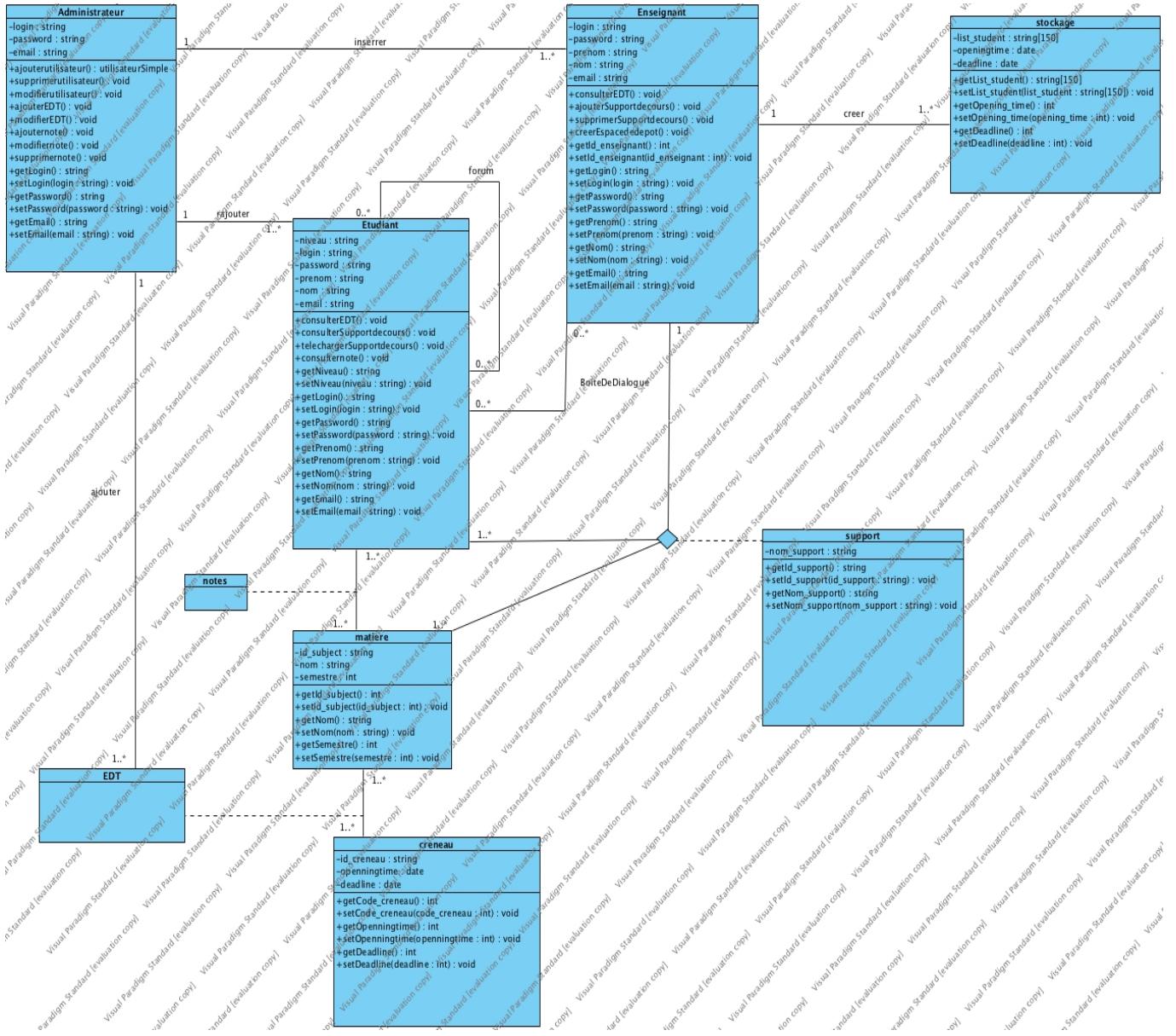


Figure 23 : « Diagramme de classe »

➤ Modèle Logique de données

Un modèle logique de données (MLD) est la représentation des données d'un système d'information. Les données sont représentées en prenant en compte le modèle technologique qui sera utilisée pour leur gestion. Le (MLD) est généré par le MCD en se transformant en un formalisme entité/association en troisième forme normale. Grace à ses multiples fonctionnalités le logiciel PowerAMC est en mesure de nous proposer directement la schématisation de ce modèle qui est la suivante :

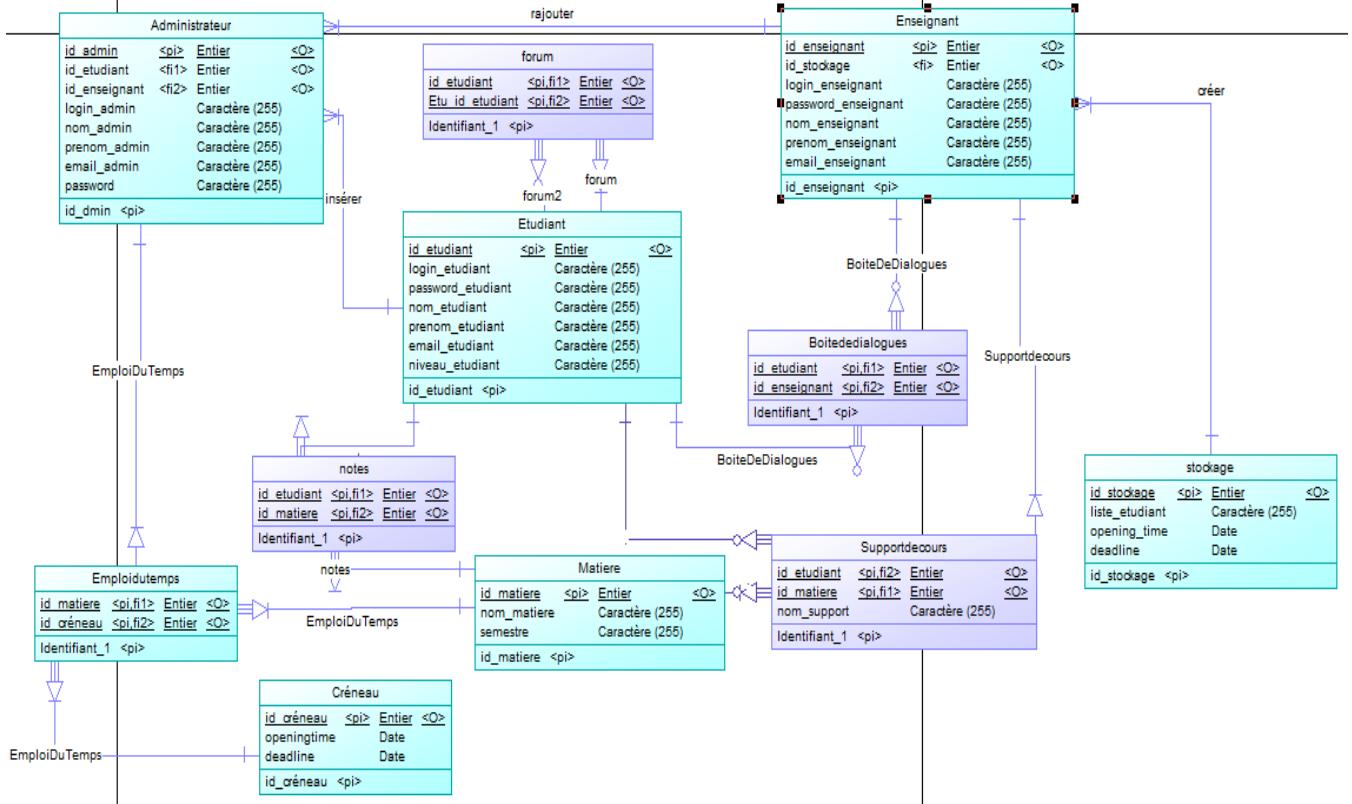


Figure 24 : « Modelé logique de données »

III. REALISATION TECHNIQUE

Après avoir élaboré la conception de notre application, nous abordons ici le dernier volet de ce rapport, qui a pour objectif d'exposer la phase de réalisation.

Nous menons tout d'abord une étude technique où nous décrivons les ressources logicielles et matérielles utilisées dans le développement de notre projet. Puis nous détaillons l'architecture, aussi nous présentons quelques interfaces réalisées pour illustrer le fonctionnement de quelques activités du système.

1. ETUDE TECHNIQUE

L'étude technique est une phase d'adaptation de conception à l'architecture technique. Cette étude, qui suit l'étude détaillée, constitue le complément de spécification informatique nécessaire pour assurer la réalisation du futur système.

➤ Matériel de base

Le développement de l'application est réalisé via deux ordinateurs portable ayant les caractéristiques suivantes :

Caractéristiques	Toshiba
Marque	Toshiba
Processeur	1,60hz
Ram	4giga
Disque dur	500giga
Système d'exploitation	Microsoft Windows 10

Tableau 2 : Matériel de base

Caractéristiques	MacOs 13
Marque	MacBook Pro
Processeur	2,5 GHz Intel Core i5
Ram	8Go
Disque dur	500Go
Système d'exploitation	Ubuntu

Tableau 3 : Matériel de base

➤ Outils de développement

Les principaux outils de développements utilisés pour l'élaboration de notre application sont présentés comme suit :

Wamp v3.2.3

WampServer (anciennement WAMP5) est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans avoir à se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant trois serveurs (Apache, MySQL et Maria DB), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL. Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un tray icône (icône près de l'horloge de Windows). La grande nouveauté de WampServer 3 réside dans la possibilité d'y installer et d'utiliser n'importe quelle version de PHP, Apache, MySQL ou Maria DB en un clic. Ainsi, chaque développeur peut reproduire fidèlement son serveur de production sur sa machine locale.



VISUAL STUDIO CODE



Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et MacOs. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la factorisation du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires. Le code source de Visual Studio Code provient du projet logiciel libre et open source VS Code de Microsoft publié sous la licence MIT permissive, mais les binaires compilés sont des logiciels gratuits pour toute utilisation. Dans le Stack Overflow 2019 Developer Survey, Visual Studio Code a été classé comme l'outil d'environnement de

développement le plus populaire, avec 50,7 % des

87 317 répondants déclarant l'utiliser.

VISUAL PARADIGM

Visual Paradigm (VP-UML) is a UML CASE Tool supporting UML 2, SysML and Business Process Modeling Notation (BPMN) from the Object Management Group (OMG). In addition to modeling support, it provides report generation and code engineering capabilities including code generation. It can reverse engineer diagrams from code, and provide round-trip engineering for various programming languages.



PowerAMC



Power Designer (anciennement PowerAMC) est un logiciel de conception créé par la société SAP, qui permet de modéliser les traitements informatiques et leurs bases de données associées. Il a été créé par SDP sous le nom AMC*Designer, racheté par Power soft qui lui-même a été racheté par Sybase en 1995.

Depuis 2010 Sybase appartient à l'éditeur allemand SAP. Avant mars 2016, la version française était commercialisée par SAP sous la marque PowerAMC, jusqu'à la fusion avec la version internationale sous le nom Power Designer depuis la version 16.6. Power Designer est disponible sous forme d'application native Microsoft Windows ou comme plugin éclipse. Par défaut, Power Designer stocke ses modèles sous forme de fichiers, dont l'extension dépend du type de modèle : bpm (pour business process model), cdm (pour conceptual data model) ... La structure interne du fichier peut être du XML ou du binaire compressé. Power Designer peut aussi stocker ses modèles dans un Référentiel.

GANTT project

L'outil Gantt Project permet de planifier un projet à travers la réalisation de diagrammes de Gantt ainsi que des diagrammes de ressources et des réseaux PERT. Voici la liste des fonctionnalités principales du projet : description de la structure de découpage du projet (en anglais "Work Breakdown Structure"). C'est une décomposition hiérarchique, axée sur les tâches, du travail que l'équipe de projet doit exécuter pour atteindre les objectifs du projet et produire les livrables voulus. liens multiples de précédences entre les tâches (prédécesseurs/successeurs) ; visualisation du chemin critique ; sauvegarde des données au format XML (fichiers .gan) ; compatibilité avec Microsoft Project (importation et exportation au format MPX (*.mpx) et MSPDI (*.xml)(format d'échange de donnée basé sur XML depuis Microsoft Project 2002)); gestion des jours fériés et des vacances pour les ressources ; édition de rapports au format HTML ou PDF ; exportation du diagramme en image PNG ou JPEG ; compatible avec un serveur WebDAV ; sauvegarde d'états du projet et comparaison ; Support de plus de 20 langages.



Microsoft WORD



Un logiciel de traitement de texte couvre deux notions, assez différentes en pratique : un éditeur de textes interactif et un compilateur pour un langage de mise en forme de textes (notions qui sont précisées dans Traitement de texte). Au cours de son évolution, Word a intégré l'outil de dessin qui permet d'effectuer des opérations de publication, comme l'ajout de graphiques (diagrammes, graphiques économiques, formes géométriques, illustrations, équations) aux documents. Microsoft Word 2007 intègre un système de menus d'un nouveau genre où les sous-menus n'apparaissent pas sous forme de texte mais sous forme de barre d'icône changeant de contenu. Ce jeu d'onglets se retrouve dans Microsoft Word 2010 et remplace intégralement les menus. Word 2013 apporte l'usage tactile et un mode lecture. Word 2016 intègre de nouvelles fonctionnalités de collaboration en temps réel et de partage et simplifie la création de forme

COMPOSER

Composer est un logiciel gestionnaire de dépendances libre écrit en PHP. Il permet à ses utilisateurs de déclarer et d'installer les bibliothèques dont le projet principal a besoin. Le développement a débuté en avril 2011 et a donné lieu à une première version sortie le 1 er mars 2012. Développé au début par Nils Adermann et Jordi Boggiano (qui continuent encore aujourd'hui à le maintenir), le projet est maintenant disponible sur la plateforme GitHub. Il est ainsi développé par toute une communauté. Le logiciel Composer est à l'initiative d'un portage en PHP du logiciel libzypp satsolver d'open Suse. Le logiciel Composer est fortement inspiré du logiciel npm pour node.js Et de bundle pour Ruby.



➤ LANGUAGE DE DEVELOPPEMENT

Les principaux langages de développements utilisés pour l'élaboration de notre application sont présentés comme suit :

SYMFONY



Symfony est un ensemble de composants PHP ainsi qu'un Framework MVC libre écrit en PHP. Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d'accélérer le développement d'un site web. Symfony 1.x, 2.x et 3.x utilisent la version 5 de PHP (PHP \geq 5.2.4 pour SF1, PHP \geq 5.3.3 pour SF2 et PHP \geq 5.5.9 pour SF3). La version 7 de PHP est quant à elle utilisée pour Symfony 4.x (PHP \geq 7.1) et Symfony 5.x (PHP \geq 7.2.5). Symfony (1.x) propose entre autres : une séparation du code en trois couches, selon le modèle MVC, pour une plus grande maintenabilité et évolutivité ; des performances optimisées et un système de cache afin d'assurer des temps de réponse optimaux ; une gestion des URL parlante, permettant à une page d'avoir une URL distincte de sa position dans l'arborescence ; un système de configuration en cascade utilisant pleinement le langage de description YAML ; un générateur de back-office et un lanceur de module (scaffolding) ; l'internationalisation native ; le support d'AJAX ; une architecture extensible permettant créations et utilisations de plugins. Symfony fournit une interface en ligne de commande pour améliorer la productivité en créant un code de base modifiable à volonté. La version 5.0 est sortie en novembre 2019.

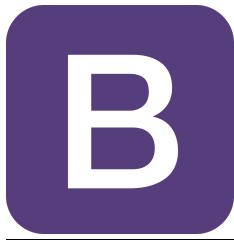
PHP v7.4.9

PHP (HyperText Preprocessor) est un langage de programmation qui s'intègre dans vos pages HTML. Il permet entre autres de rendre automatiques des tâches répétitives,

notamment grâce à la communication avec une base de données (utilisation la plus courante de PHP).



BOOTSTRAP



Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement Github.

2. Architecture générale de l'application

Le système se compose d'un ou plusieurs ordinateurs reliés à l'internet. L'architecture générale de la plateforme permet de donner une vision de haut niveau du système.

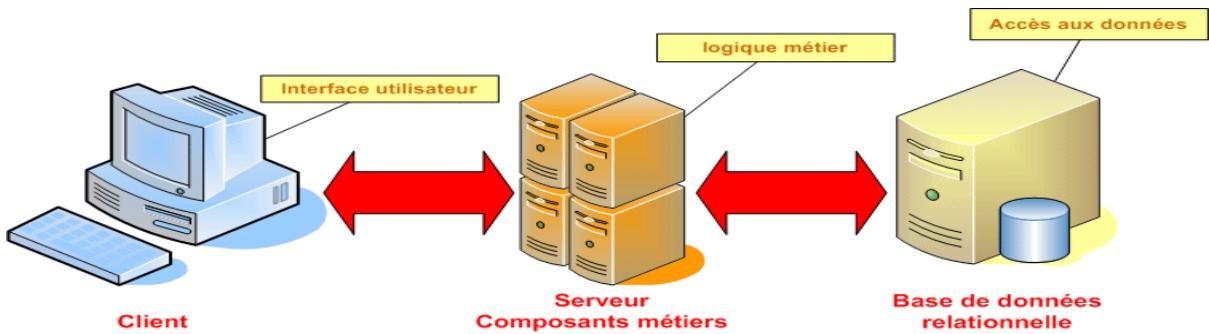


Figure 25 : Architecture 3-tiers

Les deux premiers tiers seront constitués à l'aide de la suite WAMP qui est composée :

- ✓ du moteur de base de données mySQL,
- ✓ du serveur Web Apache,
- ✓ du préprocesseur PHP,
- ✓ de l'outil d'administration phpMyAdmin.

Le troisième tiers, le client, sera un navigateur standard (Microsoft edge, Chrome, IE, etc.) :

- ✓ recevant des pages HTML,
- ✓ exécutant des feuilles de style (CSS),
- ✓ exécutant des scripts JavaScript.

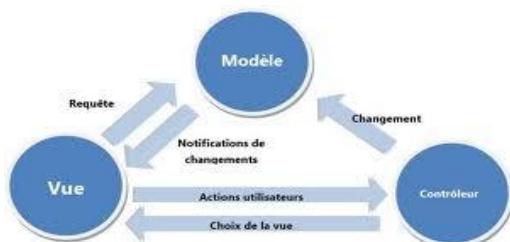


Figure 26 : Architecture MVC

Ce modèle de développement distingue 3 fonctionnalités :

- La vue (V)

Elle représente ce qui est exposé à l'utilisateur, en général il s'agit de HTML statique ou généré par du PHP ; il y a deux sortes de vue :

- ✓ Les pages d'information navigables grâce à des liens

- ✓ Les formulaires de saisies d'informations ; ces formulaires peuvent être présentées à plusieurs reprises pour confirmation ou signalement d'erreurs.

- Le contrôleur (C)

Ce sont les contrôleurs qui vont être à l'écoute des requêtes de l'utilisateur et fournir ainsi la vue externe correspondante. Pour cela, il faudra à tout moment connaître l'état de l'application c'est dire le contexte de la demande : « la page demandée fait suite à quelle action précise de l'utilisateur ? » C'est au contrôle de connaître l'état applicatif en testant une variable qui sera nommée Saction, provenant d'une requête POST ou GET.

- Le modèle (M)

C'est la couche (bibliothèque de fonction ou de classe) qui accède à la base de données.

3. Présentation des interfaces

Après les phases d'étude de l'existant, la conception et la modélisation fonctionnelle et organisationnelle nous avons développé les interfaces de notre application. Nous nous proposons dans cette section de présenter des scénarios d'utilisation illustrés par des interfaces graphiques relatives aux différents acteurs de notre plateforme.



Figure 27 : Page d'accueille

À partir de cette première interface, un utilisateur (administrateur, enseignant et étudiant) pourra se connecter. Il suffit cliquer sur le bouton connexion et ils accèderont à la page login qui suit :

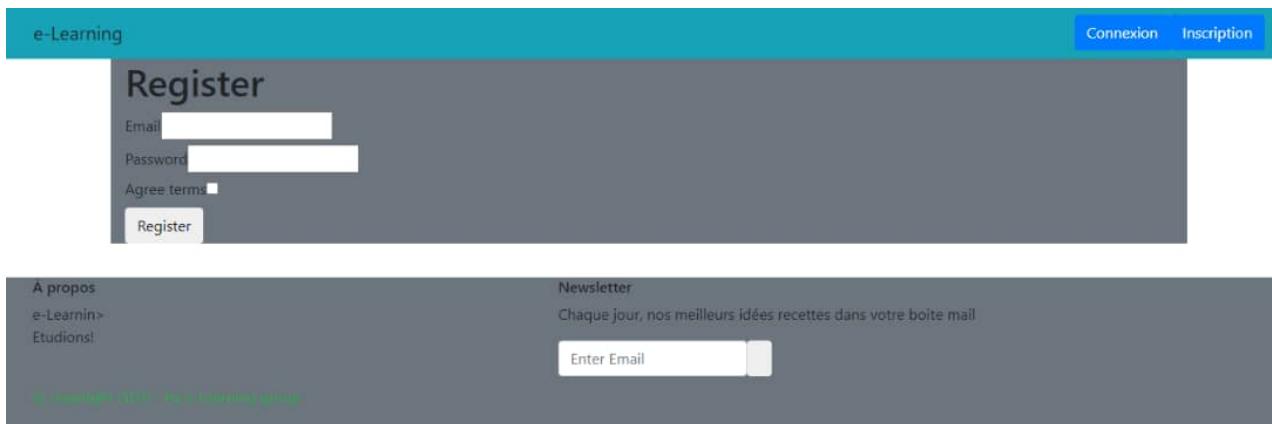


Figure 28 : page de connexion

De là, il suffira d'entrer un Email et un mot de passe et cliquer sur le bouton « register » pour ouvrir sa session.

➤ Scénario « Administrateur »

Soit le scénario suivant :

- 1) Ajouter un composant à la plateforme et le configurer.
- 2) Créer emploi du temps
- 3) Ajouter un utilisateur.
- 4) Configurer les permissions des utilisateurs.
- 5) Ajouter notes

E Learning

- [Dashboard](#)
- [Etudiant](#)
- [Enseignant](#)
- [Boitededialogue](#)
- [Creneau](#)
- [Emploidutemps](#)
- [Forum](#)
- [Matiere](#)
- [Notes](#)
- [Utilisateur](#)

Create Emploidutemps

[Create and add another](#) [Create](#)

Figure 29 : page administrateur « créer emplois du temps »

Cette interface permet à l'administrateur de créer un emploi du temps pour les utilisateurs.

E Learning

- [Dashboard](#)
- [Etudiant](#)
- [Enseignant](#)
- [Boitededialogue](#)
- [Cneau](#)
- [Emploidutemps](#)
- [Forum](#)
- [Matiere](#)
- [Notes](#)
- [Utilisateur](#)

Create Enseignant

Nom

Prenom

Login

Password

Email

Figure 30 : page administrateur « ajouter enseignant »

Cette interface permet à l'administrateur d'ajouter un enseignant.

E Learning

- [Dashboard](#)
- [Etudiant](#)
- [Enseignant](#)
- [Boitededialogue](#)
- [Creneau](#)
- [Emploidutemps](#)
- [Forum](#)
- [Matiere](#)
- [Notes](#)
- [Utilisateur](#)

Create Etudiant

Nom

Prenom

Email

Login

Password

Niveau

Figure 31 : page administrateur « ajouter étudiant »

Cette interface permet à l'administrateur d'ajouter un étudiant.

E Learning

- [Dashboard](#)
- [Etudiant](#)
- [Enseignant](#)
- [Boitededialogue](#)
- [Creneau](#)
- [Emploidutemps](#)
- [Forum](#)
- [Matiere](#)
- [Notes](#)
- [Utilisateur](#)

Create Notes

Figure 32 : page administrateur « ajouter note »

Cette interface permet à l'administrateur d'ajouter une note.

➤ Enseignant

Soit le scénario suivant :

- 1) Consulter emplois de temps
- 2) Ajouter support de cours.

- 3) Créer espace de dépôt
- 4) Discuter avec un étudiant via le module boites de dialogue.

e-Learning Connexion Inscription

Espace Enseignant

Loren ipsum lorem ipsum Lorem ipsum lorem ipsum Lorem ipsum lorem ipsum Loren ipsum lorem ipsum

Support de cours

Emploi du temps

Espace de depot

Boite de dialogue

Figure 33 : page enseignant

Chaque bouton correspond à l'un des cas du scenario.

➤ Etudiant

Soit le scénario suivant :

- 1) Consulter emploi du temps
- 2) Communiquer avec son tuteur via boite de dialogue
- 3) Consulter les supports des cours.
- 4) Discuter entre étudiant sur un forum .

Espace Etudiant

Loren ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum Loren ipsum lorem ipsum

- [Support de cours](#)
- [Emploi du temps](#)
- [Voir Notes](#)

- [Accéder au Forum](#)
- [Boîte de dialogue](#)

Figure 34 : page étudiant

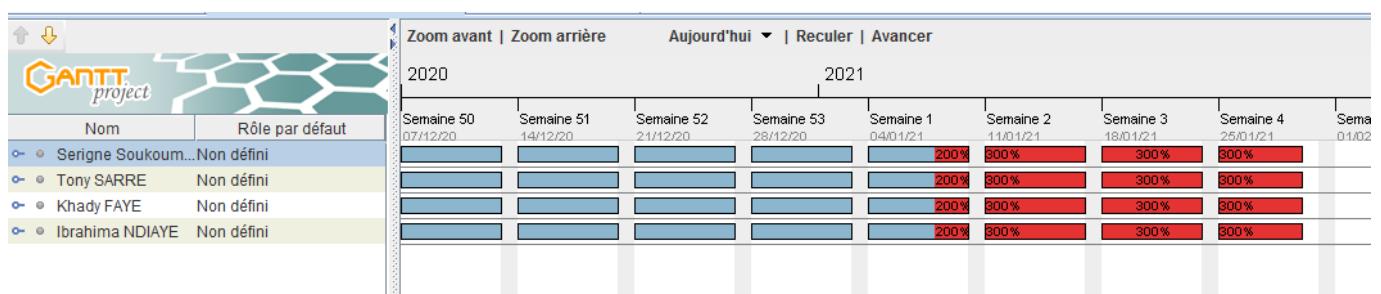
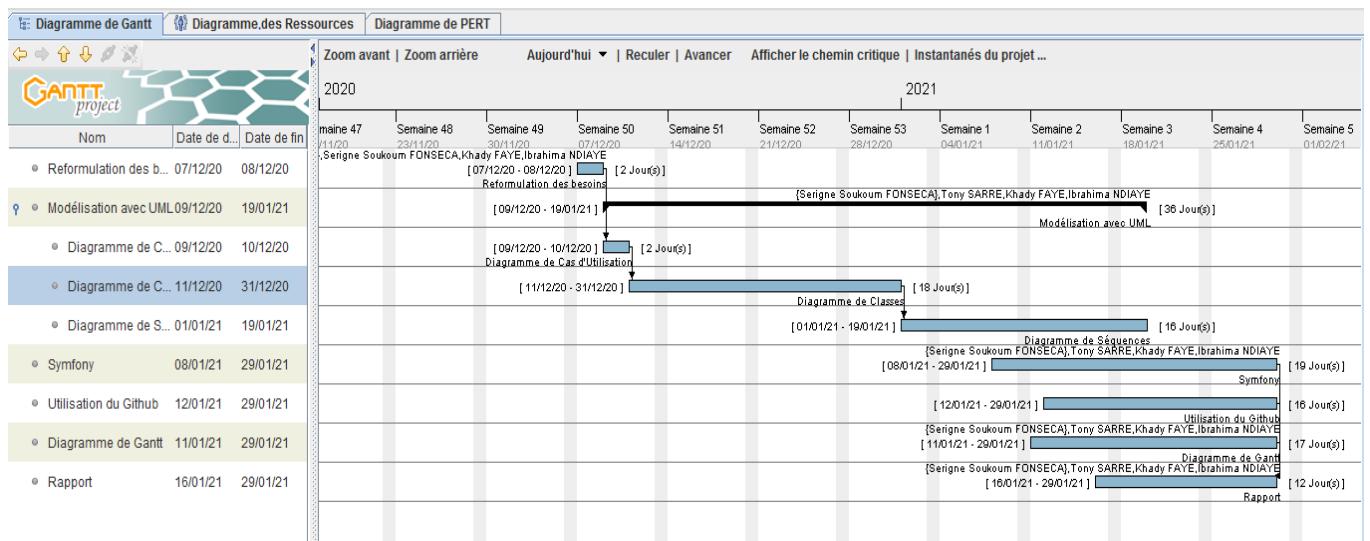
Chaque bouton correspond à l'un des cas du scenario.

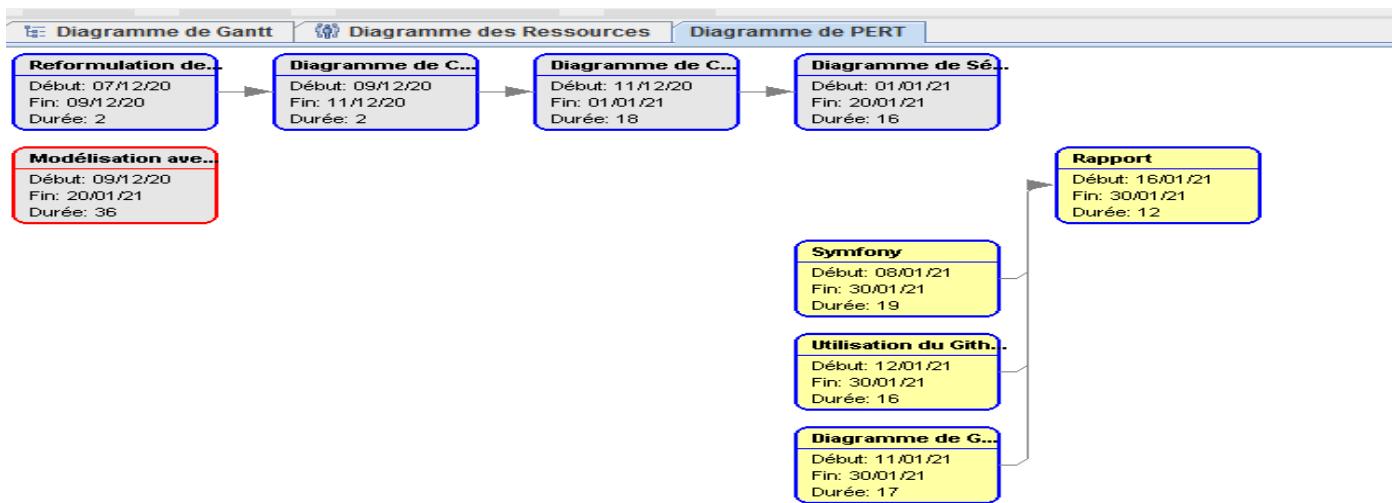
IV. ORGANISATION ET BILAN DU PROJET

1. Organisation du groupe : méthode de travail

Pour réaliser ce projet, nous avons travaillé en groupe du début jusqu'à la fin.

Ce diagramme de Gantt ci-dessous résume toute la procédure de notre travaille :





2. Résultat

➤ Apports

Ce projet nous a permis de se familiariser avec un certain nombre de concepts tout en se basant sur nos connaissances acquises au cours de notre formation.

- Apport au niveau des connaissances techniques

Ce projet de fin d'étude nous a permis de :

- ⊕ Mettre en œuvre les notions et les connaissances acquises au sein des modules « Framework et IDE ».
- ⊕ Manipuler les logiciels WAMP, Composer, Visual Studio code, Visual Paradigme, PowerAMC au cours du développement de l'application.
- ⊕ Améliorer nos connaissances théoriques concernant la communication client/serveur.
- ⊕ Etudier les architectures multicouches et réaliser l'importance du travail sur ce type d'application dans le monde de programmation.
- ⊕ Apprendre à gérer un projet.

- Apports au niveau de la conception et du développement

Au niveau de la conception et du développement, cette étude nous a permis de :

- ⊕ Mener une conception orientée objet représentée avec le langage UML
- ⊕ Apprendre à maîtriser la méthodologie de conception et de développement des applications clientes.
- ⊕ Comprendre en tant soit peu le Framework symfony
- ⊕ Maîtriser les étapes de développement et de conception d'une application Web.

➤ **Bilan Personnel :**

- ⊕ **Tony SARRE :** Ce projet a été une grande aventure pour moi. Il m'a permis de découvrir symfony et d'alimenter ma soif de développement web. Passionnée du développement, j'ai pu évoluer mes efforts et mon enthousiasme. J'ai failli me décourager lorsque fonceka s'en est aller avec notre projet mais cet esprit de groupe que nous avions m'a permis de tenir le coup jusqu'au bout.
- ⊕ **Ibrahima N'diaye :** Grace à ce projet, j'ai appris ce qu'est l'esprit de groupe et surtout j'ai découvert le Framework symfony avec toute sa complexité. J'avoue que cela était une grande expérience.
- ⊕ **Serigne Soukoum Fonceka :** Ce projet a été une grande aventure et une expérience sans faille. J'ai découvert ce qu'est le travail de groupe. Ce dernier m'a beaucoup stressé rajouté au stress personnel que j'ai reçu dernièrement, j'ai causé du tort à mes amis et camarades, oui je me suis rendu compte que ce sont mes amis car ils ne m'ont lâché en aucun cas et ne m'ont jamais jugé. Je suis très reconnaissant et m'excuse de mon acte. Mis à part, j'ai découvert symfony et j'ai aimé.
- ⊕ **Khady FAYE :** ce projet m'a fait découvrir de nouvelle chose et m'a fait vivre plein de chose au prêt de mes camarades. Symfony est un Framework complexe mais je me suis surpassé et j'ai vraiment compris plein de chose. Même si Fonceka nous avait causé du tort, nous autres avons pu nous relever avec l'aide de l'autre groupe. Je leur remercie vraiment. Cela était une grande aventure.

CONCLUSION :

Pour aboutir à notre but, nous avons procédé en trois principales étapes. Nous avons commencé par une étude des besoins. Nous avons présenté les besoins fonctionnels et non fonctionnels, ainsi qu'un diagramme global des cas d'utilisation. Nous avons passé par la suite à l'étude conceptuelle et organisationnelle. Nous avons modélisé l'aspect dynamique de notre système à l'aide des diagrammes de séquences et d'autres d'activités. L'aspect statique en utilisant le diagramme de classes. Finalement, nous avons présenté l'architecture générale de l'application et nous avons déployé différentes interfaces graphiques de la solution que nous avons implémentées. Les objectifs du projet n'ont pas tous étaient atteints. En effet, les fonctionnalités mentionnées dans le cahier des charges ont été implémentées et sont fonctionnels. L'amélioration de notre plateforme pourrait se rapporter à l'ajout d'autres fonctionnalités et améliorer le designer.