

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique
Université Abderhmane Mira - Bejaia -
Faculté des Sciences Exactes
Departement d'Informatique



*Mémoire de fin de Cycle en vue de l'obtention d'un Master professionnelle
Option : Génie Logiciel - Administration et Sécurité des Réseaux*

Thème

Conception et réalisation d'une application web 3 tiers pour
la gestion de scolarité, cas d'études
«département de technologie»

Réalisé par :

M^{lle} IKENE Fahima
M^{lle} MOUDOUB Lila

Le jury composé de :

Présidente M^{me} Amal TIAB.
Examinatrice M^{me} Feriel BESSAAD.
Encadreur M^r Redouane OUZEGGANE.

Promotion 2019-2020.

- Remerciement -

Avant tout, nous aimerions remercier le bon Dieu tout puissant qui nous a accordé
santé et courage pour réaliser ce travail.

Nous remercions notre encadreur Monsieur Redouane OUZEGGANE pour son
assistance, disponibilité, orientations et conseils.

Nous tenons également à remercier l'ensemble des membres du jury qui me font
l'honneur de juger ce travail.

Nos vifs remerciements s'adressent également tous les enseignants qui ont contribué
à notre formation.

Enfin, nous remercions aussi tous nos amis et nos familles qui nous ont soutenu et
tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

- Dedicaces -

*Au meilleur des pères
A ma très chère maman
Qu'ils trouvent en moi la source de leur fierté
A qui je dois tout
A mes sœurs et frères
A qui je souhaite un avenir radieux plein de réussite
A ma grand-mère Baya
A mes Amis
A tous ceux qui me sont chers*

LILA

- Dedicaces -

A mes très chers parents

*Pour tous leurs sacrifices, leur amour
et leur soutien tout au long de mes études*

A mes chers frères et sœurs

A toute ma famille

A mes chers Amis

FAHIMA

Table des matières

Table des matières	i
Tables des figures	iv
Liste des tableaux	vi
Liste des abréviations	viii
Introduction générale	1
1 Présentation de l'organisme d'accueil	2
1.1 Introduction	2
1.2 Présentation de l'organisme d'accueil	2
1.3 Contexte et problématique	3
1.3.1 Contexte	3
1.3.2 Problématique	3
1.3.3 Solution proposée	4
1.4 recueil des besoins	4
1.4.1 Besoins fonctionnels	4
1.4.2 Besoins non fonctionnels (Techniques)	5
1.5 Conclusion	5
2 Analyse des besoins	6
2.1 Introduction	6
2.2 Méthodes d'analyse	6
2.3 Identifications des acteurs	7
2.3.0.1 Diagramme de contexte	7
2.4 Identification des cas d'utilisations	8
2.4.1 Diagramme de cas d'utilisation	10
2.4.2 Description textuelle par cas d'utilisation	12
2.4.2.1 Description textuelle de cas d'utilisation « s'authentifier »	12
2.4.2.2 Description textuelle de cas d'utilisation « Insérer les étudiants »	13

2.4.2.3	Description textuelle de cas d'utilisation « créer unité d'enseignement »	14
2.5	Diagrammes de sequence	14
2.5.1	Diagramme de sequence de cas d'utilisation « s'authentifier »	15
2.5.2	Diagramme de sequence de cas d'utilisation « insérer les étudiants »	16
2.5.2.1	diagramme de sequence de cas d'utilisation « inscrire manuellement un étudiant »	17
2.5.2.2	diagramme de sequence de cas d'utilisation « importer des étudiants »	17
2.5.3	Diagramme de sequence de cas d'utilisation « creer unités d'enseignements »	18
2.6	Conclusion	19
3	Conception	20
3.1	Introduction	20
3.2	Diagramme de séquence d'interaction	20
3.2.1	Diagramme de séquence d'interaction « s'authentifier »	21
3.2.2	Diagramme de séquence d'interaction « insérer des étudiants »	22
3.2.3	Diagramme de séquence d'interaction « créer unités d'enseignements »	24
3.3	Le diagramme de classes du domaine	24
3.3.1	Diagramme de classes	24
3.3.1.1	les éléments de base d'un diagramme de classe	25
3.3.2	Dictionnaire de données	27
3.3.3	Schéma relationnel	28
3.3.3.1	Règles de passage du diagramme de classes au modèle relationnel	28
3.4	Conclusion	30
4	Réalisation	31
4.1	Introduction	31
4.2	Les outils et langages utilisés	31
4.2.1	Eclipse	31
4.2.2	Java	31
4.2.3	JEE	32
4.2.4	HTML	32
4.2.5	CSS	32
4.2.6	JavaScript	32
4.2.7	AJAX	33
4.2.8	Visuel Paradigm	33
4.2.9	Serveur Tomcat	33
4.2.10	Bootstrap	33
4.3	Schéma Physique de la Base de donnée	33
4.4	Diagramme de Déploiement	35

4.5	Présentation de quelques interfaces de l'application	35
4.5.1	Interface d'authentification	35
4.5.2	Interface de technicien	36
4.5.3	Interface d'ajout d'un étudiant	37
4.5.4	Interface d'ajout d'un enseignant	38
4.6	Conclusion	38
Conclusion Générale et perspectives		39
Annexe A		40
A.1	Méthodes d'analyse	40
A.1.1	La méthode agile	40
A.1.2	SCRUM	40
A.1.3	UML (Unified Modeling Language)	41
A.1.3.1	Les diagrammes UML	43
Annexe B		44
B.1	Description textuelle par cas d'utilisation	44
B.1.1	Description textuelle de cas d'utilisation « gestion des matières »	44
B.1.1.1	Description textuelle de cas d'utilisation « Ajouter matière »	44
B.1.1.2	Description textuelle de cas d'utilisation « Modifier matière »	45
B.1.1.3	Description textuelle de cas d'utilisation « Supprimer matière »	45
B.1.1.4	Description textuelle de cas d'utilisation « Attribuer une formule à une matière »	46
B.1.2	Description textuelle de cas d'utilisation « gestion des groupes »	46
B.1.3	Description textuelle de cas d'utilisation « gestion des enseignants »	47
B.1.4	Description textuelle de cas d'utilisation « Saisir les notes »	47
B.1.4.1	Description textuelle de cas « Saisir manuellement les notes »	47
B.1.4.2	Description textuelle de cas « Importer les notes »	48
B.1.5	Description textuelle de cas d'utilisation « consulter les notes »	48
B.2	Diagrammes de sequence	49
B.2.1	Diagramme de sequence de cas d'utilisation « gestion des matières »	49
B.2.2	Diagramme de sequence de cas « ajouter matière »	49
B.2.3	Diagramme de sequence de cas « modifier matière »	50
B.2.4	Diagramme de sequence de cas « supprimer matière »	51
B.2.5	Diagramme de sequence de cas « attribuer formule »	52
B.2.6	Diagramme de sequence de cas d'utilisation « gestion des groupes »	53
B.2.7	Diagramme de sequence de cas d'utilisation « gestion des enseignants »	54
B.2.8	Diagramme de sequence de cas d'utilisation « saisir l'état d'avancement »	55
B.2.9	Diagramme de sequence de cas d'utilisation « consulter les notes »	56

Annexes C	57
C.1 Diagrammes de séquence d'interaction	57
C.1.1 Diagramme de séquence d'interaction « gestion des matières »	57
C.1.1.1 Diagramme de séquence d'interaction de cas « ajouter matière »	58
C.1.1.2 Diagramme de séquence d'interaction de cas « modifier matière »	59
C.1.1.3 Diagramme de séquence d'interaction de cas « supprimer matière »	60
C.1.1.4 Diagramme de séquence d'interaction de cas « attribuer formule »	61
C.1.2 Diagramme de séquence d'interaction « gestion des enseignants »	62
C.1.2.1 Diagramme de séquence d'interaction de cas « ajouter enseignant »	62
C.1.2.2 Diagramme de séquence d'interaction de cas « modifier enseignant »	63
C.1.2.3 Diagramme de séquence d'interaction de cas « supprimer enseignant »	64
C.1.3 Diagramme de séquence d'interaction « gérer groupes »	65
C.1.4 Diagramme de séquence d'interaction « saisir les notes »	66
C.1.4.1 Diagramme de séquence d'interaction de cas : « importer les notes »	66
C.1.4.2 Diagramme de séquence d'interaction de cas : « saisir manuellement les notes »	67
C.1.5 Diagramme de séquence d'interaction « saisir l'état d'avancement »	68
C.1.6 Diagramme de séquence d'interaction « consulter les notes »	69
Bibliographie	70

Table des figures

2.1	Relation entre les acteurs du système	7
2.2	Diagramme de contexte statique	8
2.3	Diagramme global de cas d'utilisation	11
2.4	Diagramme sequence de cas d'utilisation « s'authentifier »	15
2.5	Diagramme sequence cas d'utilisation « insérer les étudiants »	16
2.6	Diagramme sequence cas d'utilisation « inscrire manuellement »	17
2.7	Diagramme sequence cas d'utilisation « importer les étudiants »	17
2.8	Diagramme sequence cas d'utilisation « Créer unités d'enseignements » .	18
3.1	Diagramme sequence d'interaction « s'authentifier »	21
3.2	Diagramme sequence d'interaction « insérer des étudiants »	23
3.3	Diagramme sequence d'interaction « créer unités d'enseignements » . . .	24
3.4	Diagramme de classe	26
4.1	Schéma Physique de la Base de donnée	34
4.2	Diagramme de Déploiement	35
4.3	Page authentification	36
4.4	Page de technicien	36
4.5	Page d'ajout d'un étudiant	37
4.6	Page d'ajout d'un enseignant	38
A.1.7	Diagrammes UML	43
B.2.1	Diagramme sequence cas d'utilisation « ajouter matière »	49
B.2.2	Diagramme sequence cas d'utilisation « modifier matière »	50
B.2.3	Diagramme sequence cas d'utilisation « supprimer matière »	51
B.2.4	Diagramme sequence cas d'utilisation « attribuer formule »	52
B.2.5	Diagramme sequence cas d'utilisation « gestion des groupes »	53
B.2.6	Diagramme sequence cas d'utilisation « gestion des enseignants »	54
B.2.7	Diagramme sequence cas d'utilisation « saisir l'état d'avancement » . . .	55
B.2.8	Diagramme sequence cas d'utilisation « consulter les notes »	56
C.1.1	Diagramme sequence d'interaction « ajouter une matière »	58
C.1.2	Diagramme sequence d'interaction « modifier une matière »	59
C.1.3	Diagramme sequence d'interaction « supprimer une matière »	60
C.1.4	Diagramme sequence d'interaction « attribuer formule »	61

C.1.5	Diagramme sequence d'interaction « ajouter un enseignant »	62
C.1.6	Diagramme sequence d'interaction « modifier un enseignant »	63
C.1.7	Diagramme sequence d'interaction « supprimer un enseignant »	64
C.1.8	Diagramme sequence d'interaction « gérer groupes »	65
C.1.9	Diagramme sequence d'interaction « importer les notes »	66
C.1.10	Diagramme sequence d'interaction « saisir les notes manuellement » . . .	67
C.1.11	Diagramme sequence d'interaction « saisir l'état d'avancement »	68
C.1.12	Diagramme sequence d'interaction « consulter les notes »	69

Liste des tableaux

2.1	Tableau représentant les cas d'utilisation	9
2.2	Description textuelle « s'authentifier »	12
2.3	Description textuelle « saisir manuellement »	13
2.4	Description textuelle « importer des étudiants »	13
2.5	franch	14
3.1	Dictionnaire de données	28
1	Description textuelle Ajouter une matière	44
2	Description textuelle Modifier une matière	45
3	Description textuelle Supprimer une matière	45
4	Description textuelle Attribuer une formule	46
5	Description textuelle Gérer les groupes	46
6	Description textuelle gestion des enseignants	47
7	Description textuelle Saisir les notes	47
8	Description textuelle importer les notes	48
9	Description textuelle consulter les notes	48

Liste des abréviations

AJAX	A synchronous J ava S cript X ML
BDD	B ase D e D onnées
CSS	C ascading S tyles S heet
HTML5	H yper T ext M arkup L anguage
JEE	J ava E ntreprise E dition
LMD	L icence M aster D octorat
MVC	M odèles V ue C ontrol
PV	P rocés- V erbal
RAD	R apid A pplication D evelopment
SGBD	S ystème G estion B ase D onnée
SQL	S tructured Q uery L anguage
UML	U nified M odeling L Language
XML	e Xtensible M arkup L anguages
XP	e Xtreme P rogramming

Introduction Générale

Actuellement, le monde connaît une avancé technologique considérable dans tous les secteurs et cela grâce a l'informatique qui est une science qui étudie les techniques de traitement automatique de l'information. Elle joue un rôle important dans le développement des entreprises et d'autre établissements.

Aujourd'hui et de plus en plus, il est possible de s'abonner a des applications web qui désigne un logiciel applicatif hébergé sur un serveur et accessible via un navigateur web. Contrairement à un logiciel traditionnel, l'utilisateur d'une application web n'a pas besoin de l'installer sur son ordinateur. Il lui suffit de se connecter à l'application à l'aide de son navigateur favori. La tendance actuelle est d'offrir une expérience utilisateur et des fonctionnalités équivalentes aux logiciels directement installés sur l'ordinateurs.

Notre projet de fin d'étude consiste à la réalisation d'une application web pour la gestion de la scolarité du département de technologie. Afin d'atteindre notre objectif, nous avons adopté une conception basée sur UML et guidé par la méthode de conception de projet SCRUM, ainsi que les différentes technologies comme JEE, HTML5, CSS3, JQuery et Mysql pour la réalisation de l'application. Pour bien présenter notre travail nous l'avons structuré en quatre chapitres :

Le chapitre un est consacré a la présentation de l'organisme d'accueil ainsi que la capture et la spécification des besoins ceci nous permettra de délimiter notre problématique et les objectifs à atteindre. Ce chapitre aboutira à un cahier de charge.

Le second chapitre porte sur l'analyse des besoins en utilisant principalement les diagrammes de cas d'utilisation système, ceci permettra d'appréhender et bien comprendre le domaine d'application ainsi que des fonctionnalités du futur système.

On ce qui concerne le chapitre trois, il permettra le passage a l'étape de conception en spécifiant avec plus de détaille, les différents objets intervenants dans la réalisation de chaque cas d'utilisateur système.

Enfin le dernier chapitre sera consacré à la phase de réalisations de notre application ou nous avons utilisé différents langages de programmation.

Chapitre 1

Présentation de l'organisme d'accueil

1.1 Introduction

Avant toute conception d'un système informatique, il est important d'effectuer une étude préliminaire du système dans le but d'identifier l'existant et de délimiter les anomalies et manques. Dans ce premier chapitre nous allons présenter l'organisme d'accueil, à savoir : le département de Technologie de l'université de Bejaia, son historique et fonctionnement.

1.2 Présentation de l'organisme d'accueil

La faculté de technologie, est issue de la restructuration de la faculté des sciences et des sciences de l'ingénierie, scindée en deux en vertu du décret exécutif 07/271 du 11/09/2007. Elle est effective à compter du 01/01/2008.

Cette nouvelle faculté compte 9252 étudiants en graduation dont 6216 inscrits en Licence et 2936 inscrits en Master, et 365 inscrits en post graduation. Le nombre d'enseignants chercheurs s'élève à 360, dont 36 professeurs, 110 maîtres de conférences, 212 maîtres assistants et 01 assistant à qui s'ajoutent 99 ATS permanents.

La faculté est actuellement structurée en 08 départements pédagogiques (Technologie, Mines et Géologie, Hydraulique, Architecture, Génie civil, Génie des procédés, Génie électrique et Génie mécanique) qui assurent des formations de Licence, de Master et de Doctorat dans le nouveau système LMD, ainsi que des formations de Doctorat dans le système classique qui est en voie d'extinction.

Elle comporte aussi en son sein 12 laboratoires de recherche (Labo de Technologie Industrielle et de l'Information, Labo de Maîtrise des Énergies Renouvelables, Labo de Génie de l'Environnement, Labo d'Electrochimie, de Corrosion et de Valorisation Énergétique, Labo d'Hydraulique Appliquée et Environnement, Labo des Procédés Membranaires et des Techniques de Séparation et de Récupération, Labo de Technologie des Matériaux et de Génie des Procédés, Labo de Génie de la Construction et Architecture, Labo des Matériaux polymères Avancés, Labo de Mécanique, Matériaux et Energétique, Labo des Matériaux Organiques et Labo de Génie Electrique) qui contribuent de manière significative à la promotion de la recherche scientifique et à la formation par la

recherche[10].

Notre travail étant centré sur le département de Sciences Technologies qui est un composant de la faculté des Sciences Technologies, en première année est commune à toutes les filières. En S3 (Semestre 3), l'étudiant choisit la matière dont il veut se spécialiser, par exemple Génie Électrique, Génie Civil..etc.

1.3 Contexte et problématique

Dans le but de palier les problèmes rencontrés par le personnel dans la gestion de scolarité du département de Sciences Technologies, l'objectif de notre travail consiste à concevoir et réaliser une application web pour la gestion de scolarité pour le département de sciences technologies de l'université de Bejaia.

1.3.1 Contexte

La gestion de scolarité est une tâche très complexe qui fait intervenir plusieurs acteurs à différents niveaux de hiérarchie. Lors de notre stage, nous avons repéré trois niveaux de gestion de la scolarité, à savoir :

- Niveau de Département.
- Niveau de Faculté.
- Niveau de Vice-Rectorat chargée de la pédagogie.

Au niveau de département, on s'occupe de l'inscription des étudiants au niveau des groupes et de sections, la saisie des notes des différentes évaluations et délibérations semestrielles et annuelles. Au niveau de la faculté, on archive les différents PV de délibération ainsi que les différents documents (Attestation de scolarité, Relevés des notes, ...) Au niveau de Vice-Rectorat chargée de la pédagogie, souvent, c'est des données décisionnelles sous formes de statistiques (Visualisation graphiques, listes, ...)

1.3.2 Problématique

Lors de notre stage au niveau du département de Sciences Technologies, nous avons constaté la complexité dans la gestion de la scolarité, nous avons pu recenser les insuffisances suivantes :

- Inexistence d'un système informatique pour la gestion de scolarité (PVs, Modules, Enseignants..etc.).
- Risque de perte de données(notes, fiches..etc.) et/ou modifications inopinées .
- L'usage des logiciels classiques (Excel) pour la gestion de la scolarité provoque des anomalies et ambiguïtés dans l'établissement de document comme les relevés de notes et PVs.

- L'application utilisée est mono-poste non reliée au réseau, par conséquent des problèmes liés aux transfères et distributions d'informations vont surgir,..etc.

1.3.3 Solution proposée

Pour remédier aux problèmes cités ci-dessus, nous avons proposé une solution qui consiste à développer une application web dédiée à la gestion de la scolarité du département ST, cette dernière vise à :

- Faciliter : la saisie des notes, l'insertion des étudiants.
- Avoir un tableau de bord pour la suivie des saisies : notes, état d'avancement des cours..etc.

1.4 recueil des besoins

Dans cette section, nous présenterons les différents besoins que l'application doit répondre. Nous avons deux types de besoins, besoins fonctionnels et besoins non-fonctionnels.

1.4.1 Besoins fonctionnels

En ce qui concerne les fonctionnalités du futur système, nous pouvons mettre l'accent sur les points suivants :

1. **La gestion de chef de département :(avoir un tableau de bords)** : le chef de département peut consulter :
 - Etat d'avancement des cours pour chaque matière.
 - Les notes(Examens, TD, TP).
 - L'opération de saisie des notes.
2. **La gestion de l'enseignant** : l'enseignant peut :
 - Saisir l'état d'avancement(TPs,TDs cours).
 - Saisir les notes : saisie manuelle sur formulaire ou par exportation/importation.
 - Renseigner les différents encadrements.
3. **La gestion de technicien** : le technicien peut :
 - Créer les unités d'enseignements.
 - Gérer :
 - La gestion des matières(ajouter, modifier, supprimer et attribuer une formule de calcul à une matière).
 - La gestion des enseignants(ajouter, modifier, supprimer).
 - La gestion des sections et groupes(ajouter, modifier, supprimer).
 - Insérer les étudiants(manuellement ou importation).
 - Générer les PVs.

1.4.2 Besoins non fonctionnels (Techniques)

L'application doit répondre aux besoins techniques suivants :

- Sécurité : Protéger l'accès à l'application par un identifiant et un mot de passe.
- L'application doit permettre de gérer les accès des utilisateurs selon un privilège.
- L'interface de cette application doit être ergonomique, conviviale et voir même apte à aider l'utilisateur à mieux gérer son espace de travail.
- Le temps de réponse de l'application doit être acceptable.

1.5 Conclusion

Dans ce chapitre, nous avons entamé notre travail par la présentation de l'organisme d'accueil. Nous avons soulevé quelques problématiques par rapport à la gestion de la scolarité. Enfin nous avons proposé une solution qui répond aux besoins du système. Dans le prochain chapitre analyse des besoins, nous allons présenter la démarche à suivre pour mener à bien notre projet.

Chapitre 2

Analyse des besoins

2.1 Introduction

Dans la phase d'analyse des besoins, nous présenterons d'abord la méthodologie suivit pour la réalisation de notre projet,ensuite nous décrirons les interactions des différents acteurs avec le système afin de réaliser les fonctionnalités attendues qui sont modélisé sous forme de cas d'utilisation. Pour mener à bien cette phase, nous commençons par la présentation des acteurs, la communication entre les acteurs et l'application, les cas d'utilisation pour chaque acteur, la description des cas d'utilisation en utilisant les diagrammes de séquences système.

2.2 Méthodes d'analyse

En ingénierie, une méthode d'analyse et de conception est un procédé qui a pour objectif de permettre de formaliser les étapes préliminaires de développement d'un système afin de rendre ce développement plus facile aux besoins du clients.

Pour ce faire,on part d'un énoncé informel (le besoin tel qu'il est exprimé par le client, complété par des recherches d'informations auprès des experts du domaine fonctionnel, comme les futurs utilisateurs du logiciel).

On cites les plus utilisées : les Méthodes systémiques, les méthodes objets et les méthodes agiles (XP,SCRUM, RAD..ect).voir Annexe A page 40

2.3 Identifications des acteurs

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Un acteur peut consulter et /ou modifier directement l'état du système, en émettant et /ou en recevant des messages susceptibles d'être porteur de données. En ce qui concerne notre système, nous avons pu identifier les acteurs suivants :

Chef de département : Accéder aux différentes statistiques : Taux d'avancement des cours, TDs et TPs – Taux d'avancement opération de saisie des notes, Les résultats des différentes évaluations (T.D., T.P.s, Examens)..

Enseignant :Saisie des états d'avancement (TPs, TDs et cours)..

Technicien :Gérer les enseignant, étudiants , pv's ..

En plus de ces trois acteurs, nous ajoutons l'acteur générique « Utilisateur » qui représente toute entité non encore identifié par le système. Ainsi pour cet acteur, la seule fonctionnalité accessible est l'authentification. Le diagramme suivant illustre la relation entre les quatres acteurs.

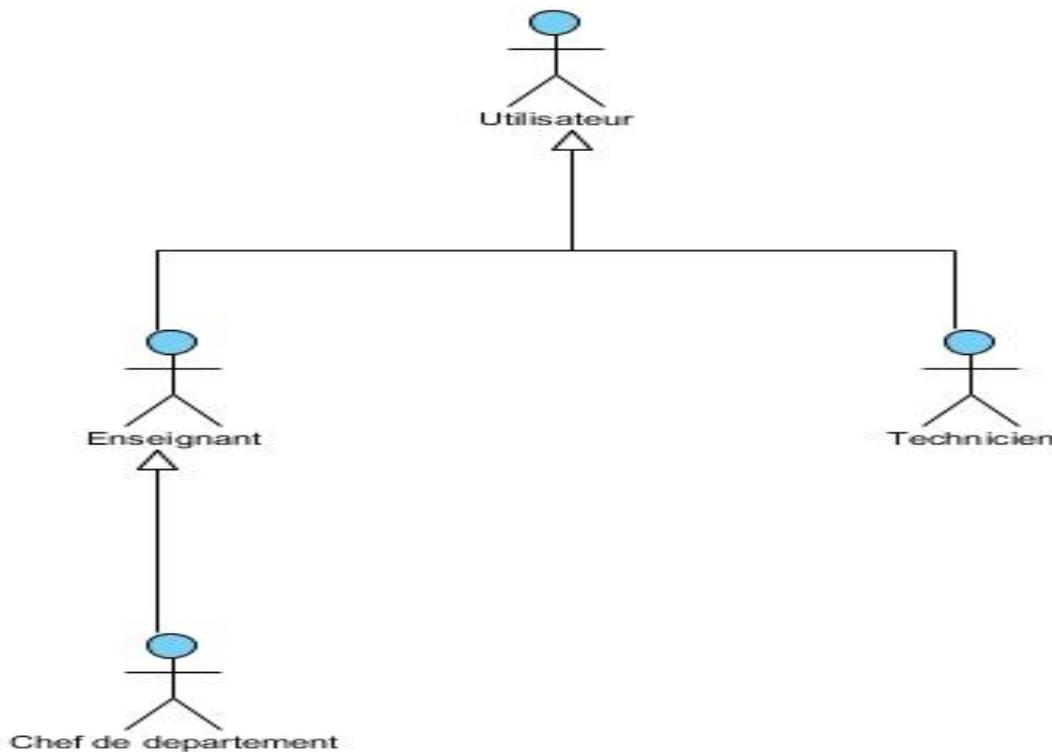


FIGURE 2.1: Relation entre les acteurs du système

2.3.0.1 Diagramme de contexte

Ce diagramme exprime l'environnement du système dans une situation donnée. Le diagramme de contexte permet donc de définir les frontières de l'étude et de préciser la phase

du cycle de vie dans laquelle on situe l'étude (généralement la phase d'utilisation). Il permet de répondre à la question : Quels sont les acteurs et éléments environnants dans mon système ?

Il peut se préparer sous forme de carte mentale pour ensuite être modéliser à l'aide d'un diagramme de définition de bloc BDD (base de données).

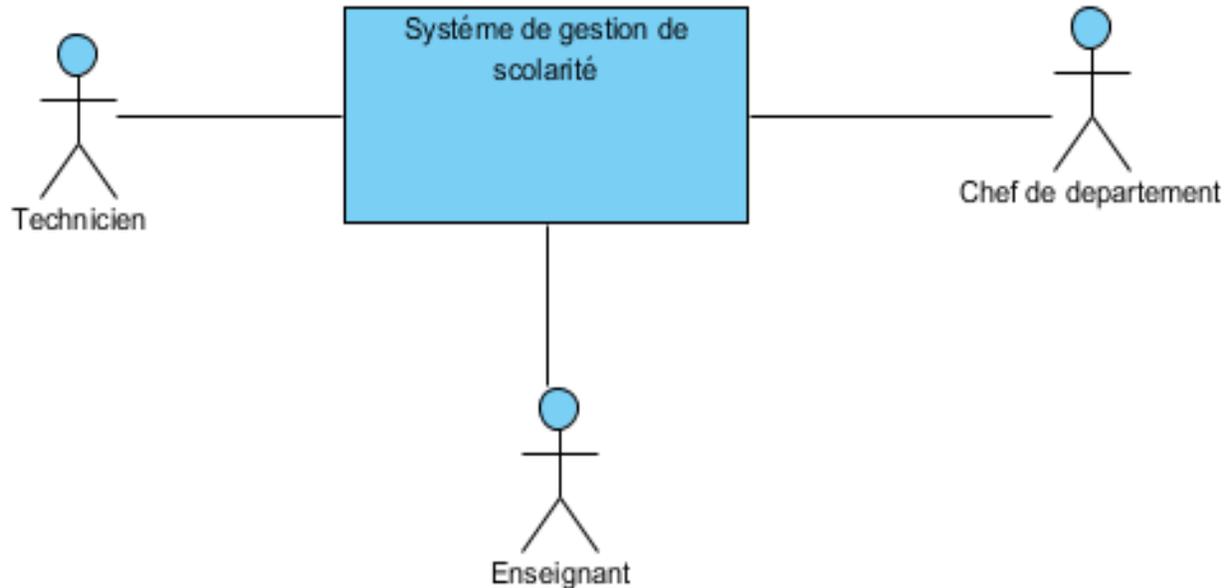


FIGURE 2.2: Diagramme de contexte statique

2.4 Identification des cas d'utilisations

En langage UML, les diagrammes de cas d'utilisation modélisent le comportement d'un système et permettent de capturer les exigences du système. Les diagrammes de cas d'utilisation décrivent les fonctions générales et la portée d'un système. Ces diagrammes identifient également les interactions entre le système et ses acteurs. Les cas d'utilisation et les acteurs dans les diagrammes de cas d'utilisation décrivent ce que le système fait et comment les acteurs l'utilisent, mais ne montrent pas comment le système fonctionne en interne.

Nous avons déterminé principalement les cas d'utilisations, présentés dans le tableau suivant :

Acteur	Cas d'utilisation	
Enseignant Chef de département Technicien	S'authentifier	
Enseignant	Saisir l'état d'avancement	
	Renseigner les encadrements	
	Saisir les notes (manuellement ou par importation/exportation)	
Chef de département	Consulter	l'état d'avancement
		l'opération de saisie des notes
		les notes
Technicien	Insérer les enseignants	
	Créer les unités d'enseignement	
	Gère la gestion des matières	
	Insérer les étudiants (manuellement ou par importation/exportation)	
	Créer les sections/groupes	
	Consulter l'historique des modifications des notes	
	Générer les PV	

TABLE 2.1: Tableau représentant les cas d'utilisation

2.4.1 Diagramme de cas d'utilisation

Ce diagramme est destiné à représenter les besoins des utilisateurs par rapport au système. il met en jeu trois concepts :

- L'acteur : Un acteur représente un rôle joué par une entité externe qui interagit directement avec le système étudié.
- Le cas d'utilisation : correspond à un certain nombre d'actions que le système devra exécuter en réponse à un besoin d'un acteur.
- L'interaction : Une interaction permet de décrire les échanges entre un acteur et un cas d'utilisation.

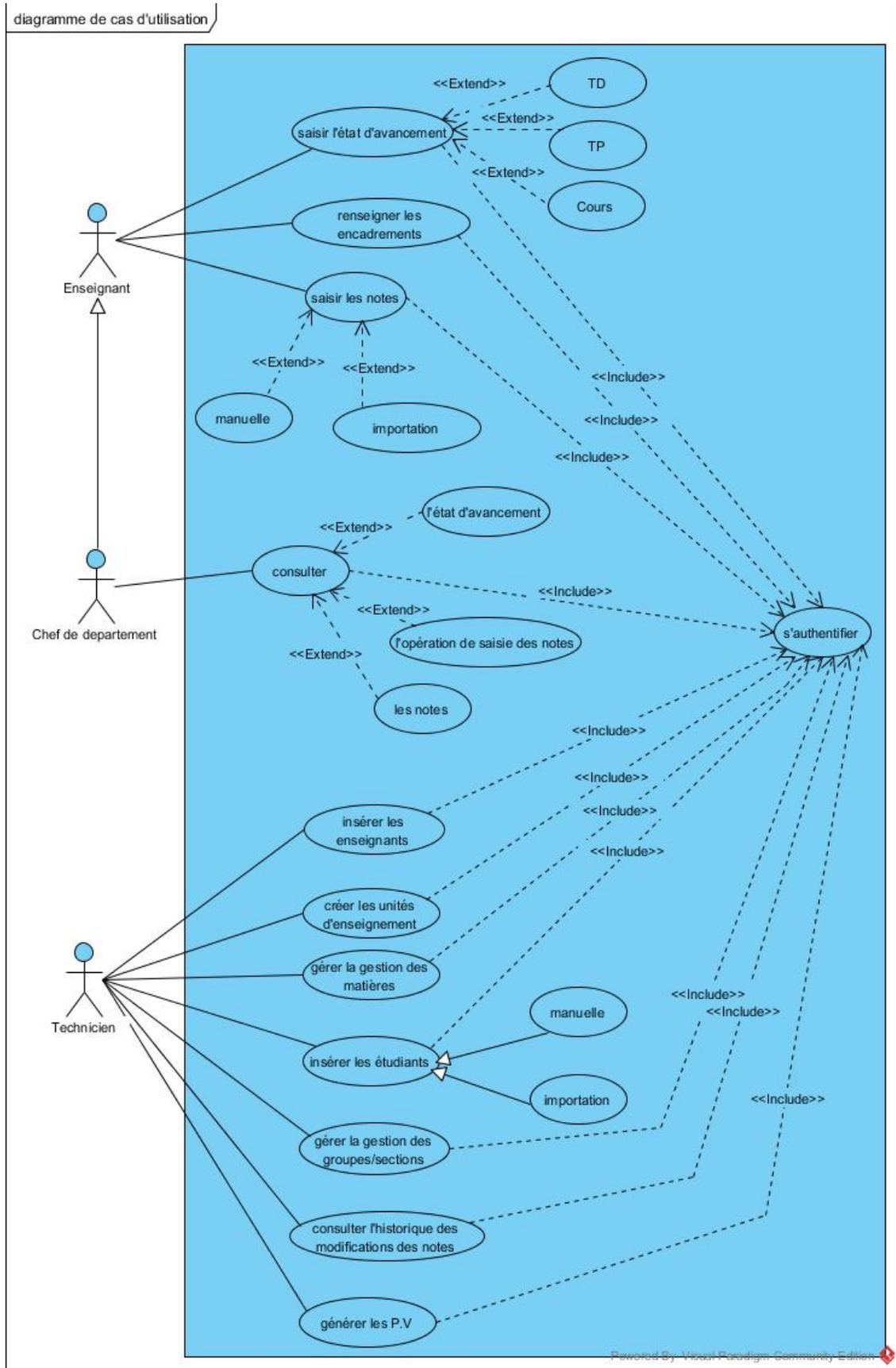


FIGURE 2.3: Diagramme global de cas d'utilisation

2.4.2 Description textuelle par cas d'utilisation

Nous allons maintenant décrire quelques cas d'utilisation énuméré dans le diagramme de cas d'utilisation précédent en identifiant les acteurs et les différents scénarios possible.

2.4.2.1 Description textuelle de cas d'utilisation « s'authentifier »

Le tableau suivant représente la discription textuelle de cas « s'authentifier » :

Cas d'utilisation	S'authentifier
Acteur	Utilisateur
Pré-condition	Néant
Post-condition	L'utilisateur accède à son espace privé
Scénario nominal	1-L'utilisateur introduire son login et son mot de passe 2-Le système vérifie les données de saisie 3-Le système redirige l'utilisateur vers son espace privé
Scénario alternatif	2-1-l'utilisateur saisir les données manquantes 2-1-1-le système affiche un message d'erreur 2-1-2-reprise de l'étape 1 de scenario nominal 2-2-les données de saisie sont incorrectes 2-2-1-le système affiche un message d'erreur de type « les données saisies sont invalides » 2-2-2-reprise de l'étape 1 de scénario nominal

TABLE 2.2: Description textuelle « s'authentifier »

2.4.2.2 Description textuelle de cas d'utilisation « Insérer les étudiants »

Les deux tableaux suivants représentent la description textuelle de cas « Insérer les étudiants » :

cas : manuellement et importation.

Cas d'utilisation	Manuellement
Acteur	Technicien
Pré-condition	Authentification préalable Le technicien clique sur la gestion des étudiants, un écran est affiché L'option saisir les étudiants manuellement est choisit
Post-condition	Les étudiants ont été bien insérés
Scénario nominal	1-Le technicien saisir les étudiants et valide 2-le système vérifie la saisie 3-le système enregistre les étudiants et affiche un message de succès
Scénario alternatif	2-1-le technicien oublie les champs manquants 2-1-1-le système affiche un message d'erreur 2-1-2-reprise de l'étape 1 de scénario

TABLE 2.3: Description textuelle « saisir manuellement »

Cas d'utilisation	Importation
Acteur	Technicien
Pré-condition	Authentification préalable Le technicien clique sur la gestion des étudiants, un écran est affiché L'option importer des étudiants est choisit
Post-condition	Les étudiants ont été bien importés
Scénario nominal	1-Le technicien choisir u fichier et il le télécharge 2-le système vérifie les données de fichier 3-le système enregistre les étudiants et affiche un message de succès
Scénario alternatif	2-1-le fichier et invalide 2-1-1-le système affiche un message d'erreur 2-1-2-reprise de l'étape 1 de scénario

TABLE 2.4: Description textuelle « importer des étudiants »

2.4.2.3 Description textuelle de cas d'utilisation « créer unité d'enseignement »

Le tableau suivant représente la description textuelle de cas « créer l'unité d'enseignement » :

Cas d'utilisation	Créer l'unité d'enseignement
Acteur	Technicien
Pré-condition	Authentification préalable
Post-condition	Une unité d'enseignement a été bien créée
Scénario nominal	1-Le technicien demande de créer l'unité d'enseignement 2-le système affiche la page « unité d'enseignement » 3-le technicien sélectionne les matières et saisir le nom de l'unité à créée et validée 4-le système créé l'unité et affiche un message de succès

TABLE 2.5: Description textuelle « Créer les unités d'enseignements »

Pour les descriptions textuelles des autres cas d'utilisation voir Annexe B page 44.

2.5 Diagrammes de sequence

Dans cette section nous allons présenter quelques diagrammes de séquences de notre système. Le diagramme de séquence est la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML.

2.5.1 Diagramme de sequence de cas d'utilisation « s'authentifier »

La figure suivante représente le diagramme de sequence de cas d'utilisation « s'authentifier » :

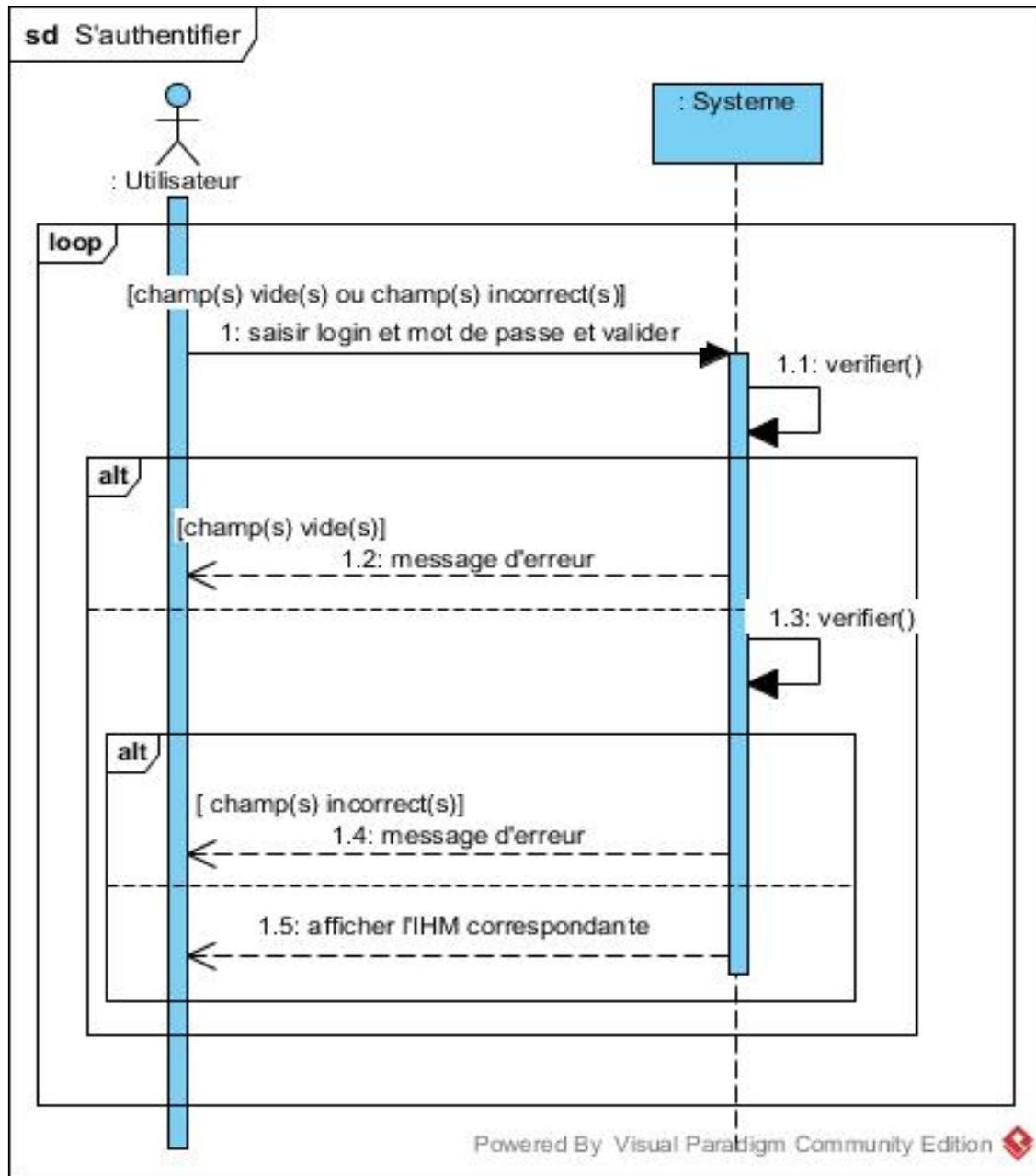


FIGURE 2.4: Diagramme sequence de cas d'utilisation « s'authentifier »

2.5.2 Diagramme de sequence de cas d'utilisation « insérer les étudiants »

Les figures suivantes représentent les diagrammes de sequence de cas d'utilisation « insérer les étudiants » :

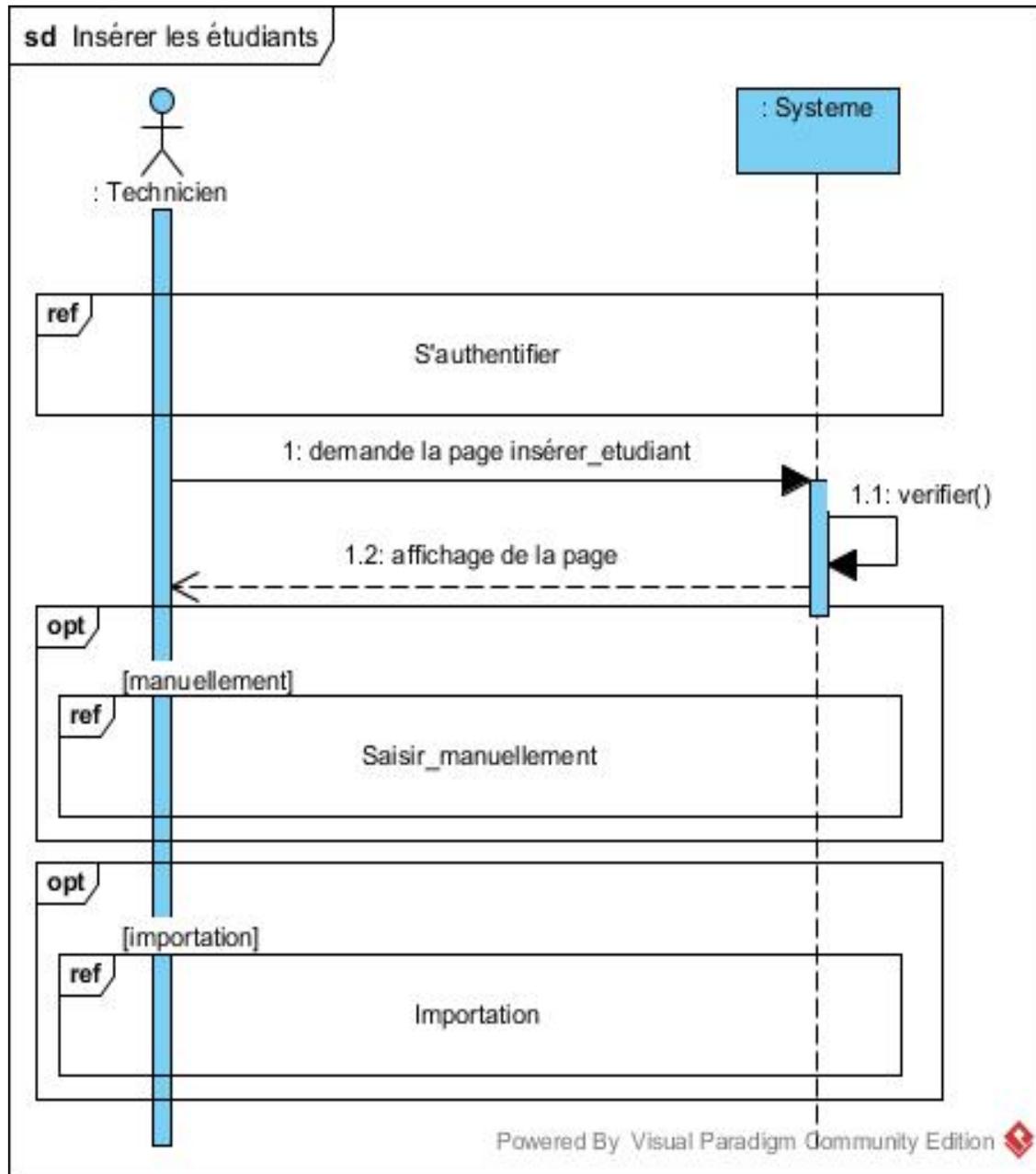


FIGURE 2.5: Diagramme sequence cas d'utilisation « insérer les étudiants »

2.5.2.1 diagramme de sequence de cas d'utilisation « inscrire manuellement un étudiant »

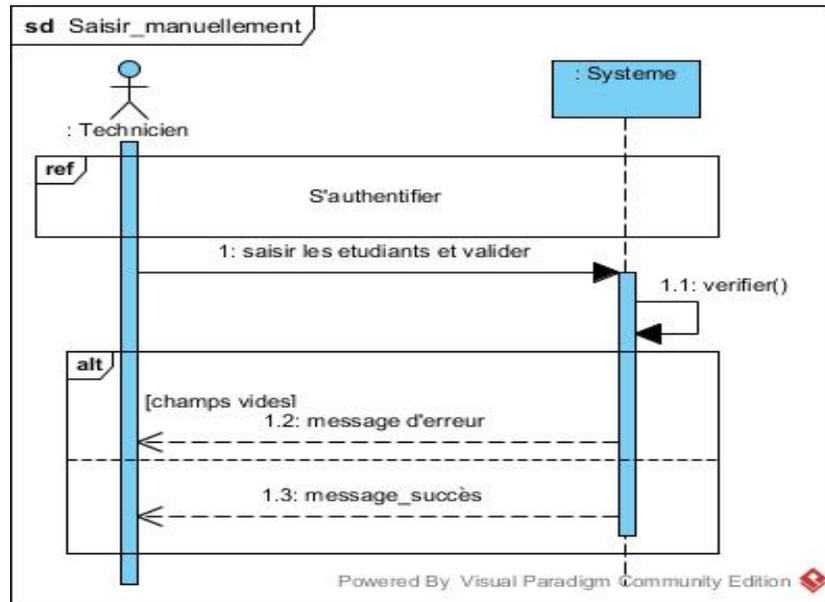


FIGURE 2.6: Diagramme sequence cas d'utilisation « inscrire manuellement »

2.5.2.2 diagramme de sequence de cas d'utilisation « importer des étudiants »

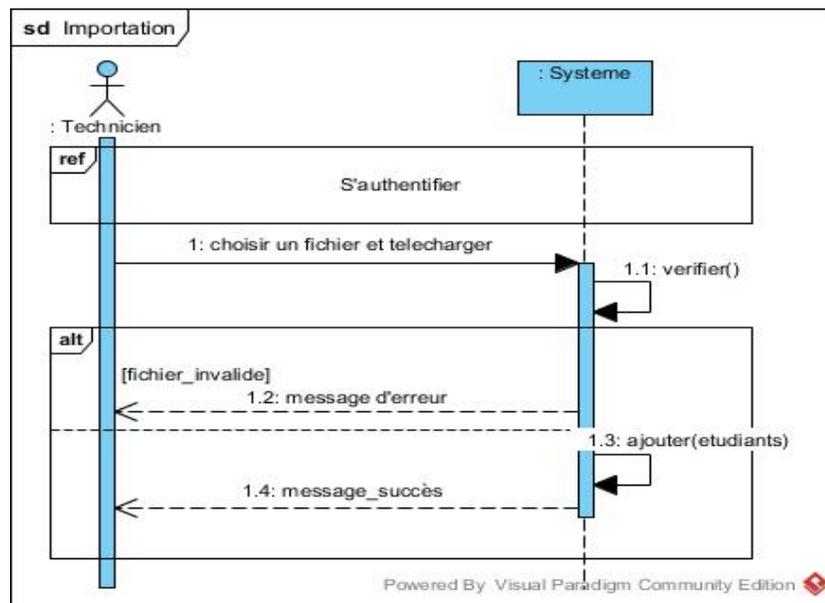


FIGURE 2.7: Diagramme sequence cas d'utilisation « importer les étudiants »

2.5.3 Diagramme de sequence de cas d'utilisation « creer unités d'enseignements »

La figure suivante représente le diagramme de sequence de cas d'utilisation « creer unités d'enseignements » :

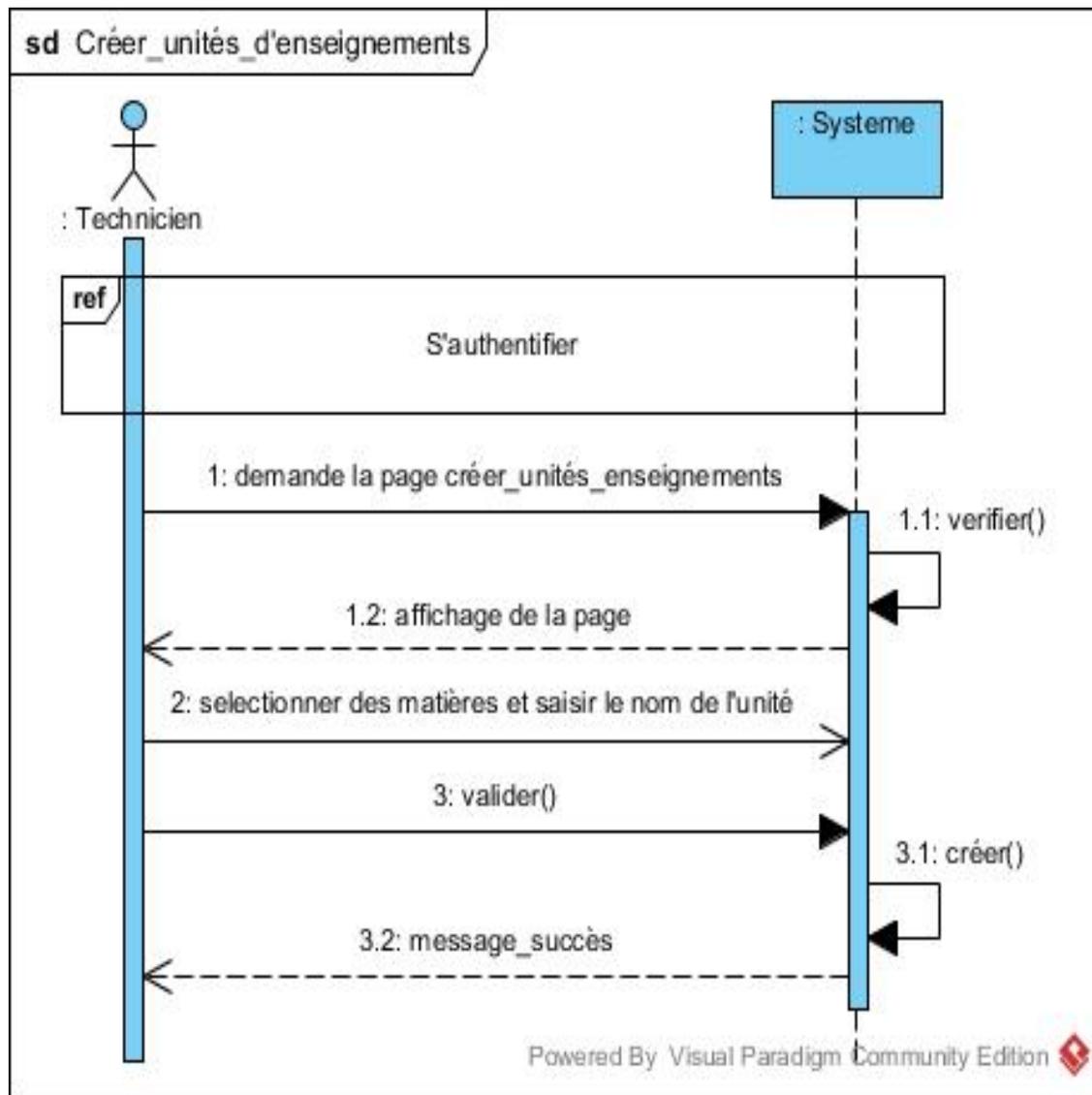


FIGURE 2.8: Diagramme sequence cas d'utilisation « Créer unités d'enseignements »

Pour les autres diagrammes de sequence voir Annexe B page 49.

2.6 Conclusion

Dans ce chapitre nous avons présenté la méthodologie suivie et identifié les acteurs de notre système aussi décrits les cas d'utilisations éventuels. dans le chapitre suivant nous mettrons l'accent sur la conceptions de notre système en se basant sur l'analyse faite dans le chapitre deux.

Chapitre 3

Conception

3.1 Introduction

Après avoir effectué l'analyse des besoins, grâce aux diagrammes de cas d'utilisation et aux diagrammes de séquence système, nous entamons la phase de conception de notre application. Cette phase sera concrétisée à l'aide des diagrammes de séquence détaillés ou d'interaction qui mettent l'accent sur la communication entre les différents objets du système. Ceci, nous mènera à déduire les différentes classes (objets interfaces, contrôleurs et objets entités) constituant l'application à réaliser. Par la suite, nous passerons à la modélisation de la base de données en appliquant les règles de passage du diagramme de classe vers le schéma relationnel.

3.2 Diagramme de séquence d'interaction

Le deuxième niveau d'utilisation des diagrammes de séquence est celui de l'analyse du domaine (conception), dans le quel le système sera éclaté en objets intervenant a la réalisation d'un cas d'utilisation. Ces objets appartiennent a trois type de classes : les objets d'interfaces (boundary object), les objets de contrôle (control object) et les objets entités (Entity object). Selon[8], il existe des règles précises sur les interactions possibles entres les instances de ces trois type de classe d'analyse :

1. les acteurs ne peuvent interagir qu'avec les classes d'interface.
2. Les classes d'interface peuvent interagir avec les classes de contrôle ou, exceptionnellement avec d'autres classes d'interface.
3. Les classes de contrôles peuvent interagir avec les interfaces, les entités ou bien avec d'autre contrôle.
4. Les classes entités ne peuvent interagir qu'avec les classes entités.

Dans ce qui suit, nous allons présenter les diagrammes de séquence d'interaction pour quelques cas d'utilisations vus dans le chapitre précédent.

3.2.1 Diagramme de séquence d'interaction « s'authentifier »

Pour bénéficier des services de notre application, l'utilisateur passe en premier lieu par un formulaire d'authentification où il saisit son identifiant et son mot de passe. Ces données sont transférées vers la classe contrôle « C_authentification » qui sera chargée de vérifier l'existence de l'utilisateur dans l'entité « Compte ». Par la suite, si les identifiants sont valides l'utilisateur sera redigé vers sa page correspondante sinon un message d'erreur sera affiché.

La figure suivante illustre le diagramme de séquence d'interaction pour le cas d'utilisation « s'authentifier » :

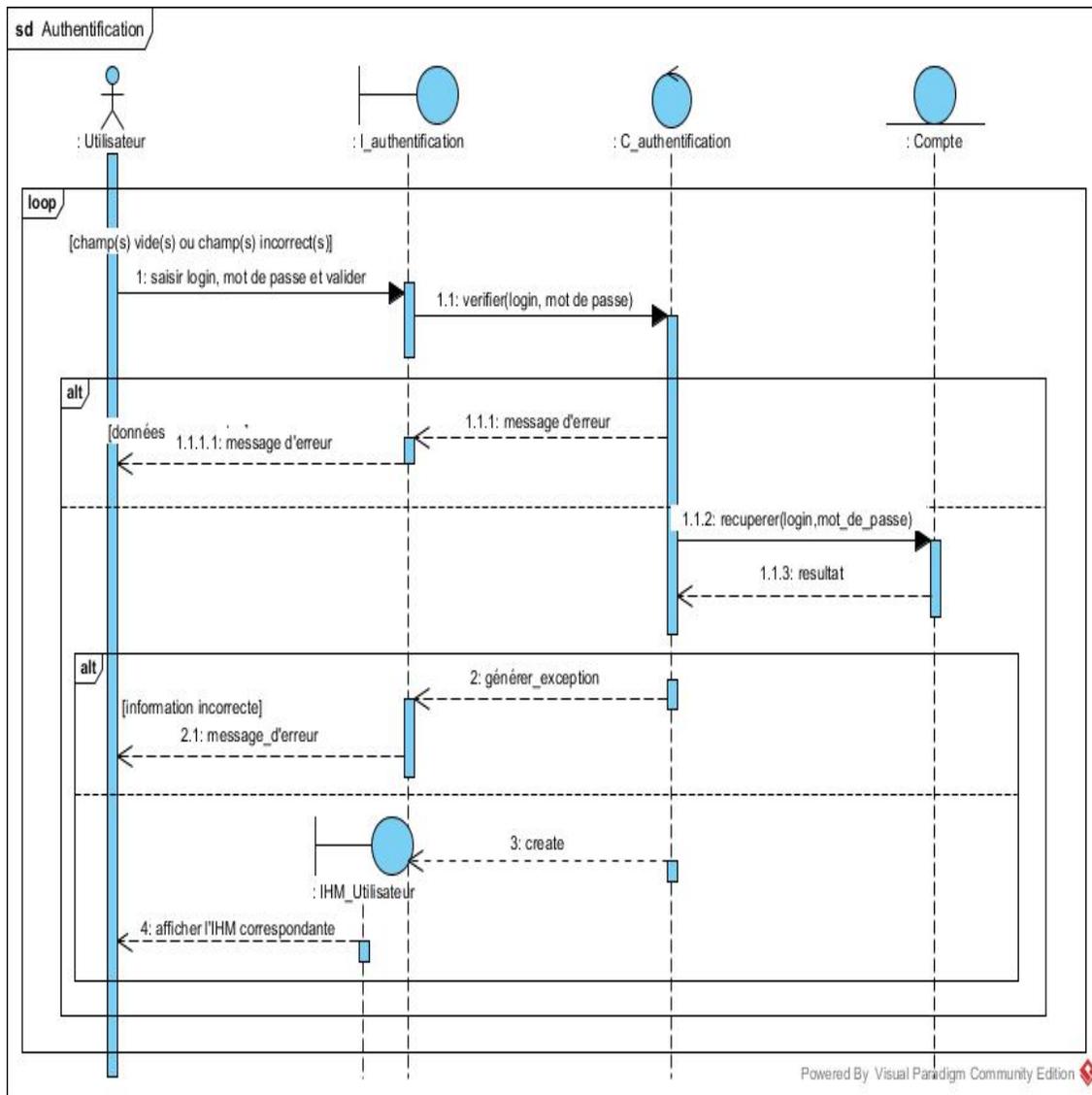


FIGURE 3.1: Diagramme sequence d'interaction « s'authentifier »

3.2.2 Diagramme de séquence d'interaction « insérer des étudiants »

Pour l'insertion des étudiants, le technicien choisit entre l'insertion manuelle ou l'importation de la liste des étudiants.

Insérer manuellement : le technicien remplit le formulaire des étudiants via la classe dialogue « : I_Gestion_etudiants », lorsque le technicien envoie les données pour les sauvegarder, ces données seront transférées en premier lieu vers le contrôle « : C_Gestion_etudiants » afin de les ajouter dans l'entité « : etudiant ».

Importer les étudiants : le technicien choisit le fichier à importer et télécharge le fichier via l'interface « : I_Gestion_etudiants ». Une fois les données seront transmises vers la classe « : C_Gestion_etudiants », elle sera chargée de vérifier la validité du fichier téléchargé. Si toutes les informations sont valides, la classe « : C_Gestion_etudiants » parcourt toutes les lignes du fichier et ajoute les nouveaux étudiants dans l'entité « : etudiant ».

La figure suivante illustre le diagramme de séquence d'interaction pour le cas d'utilisation « insérer des étudiants » :

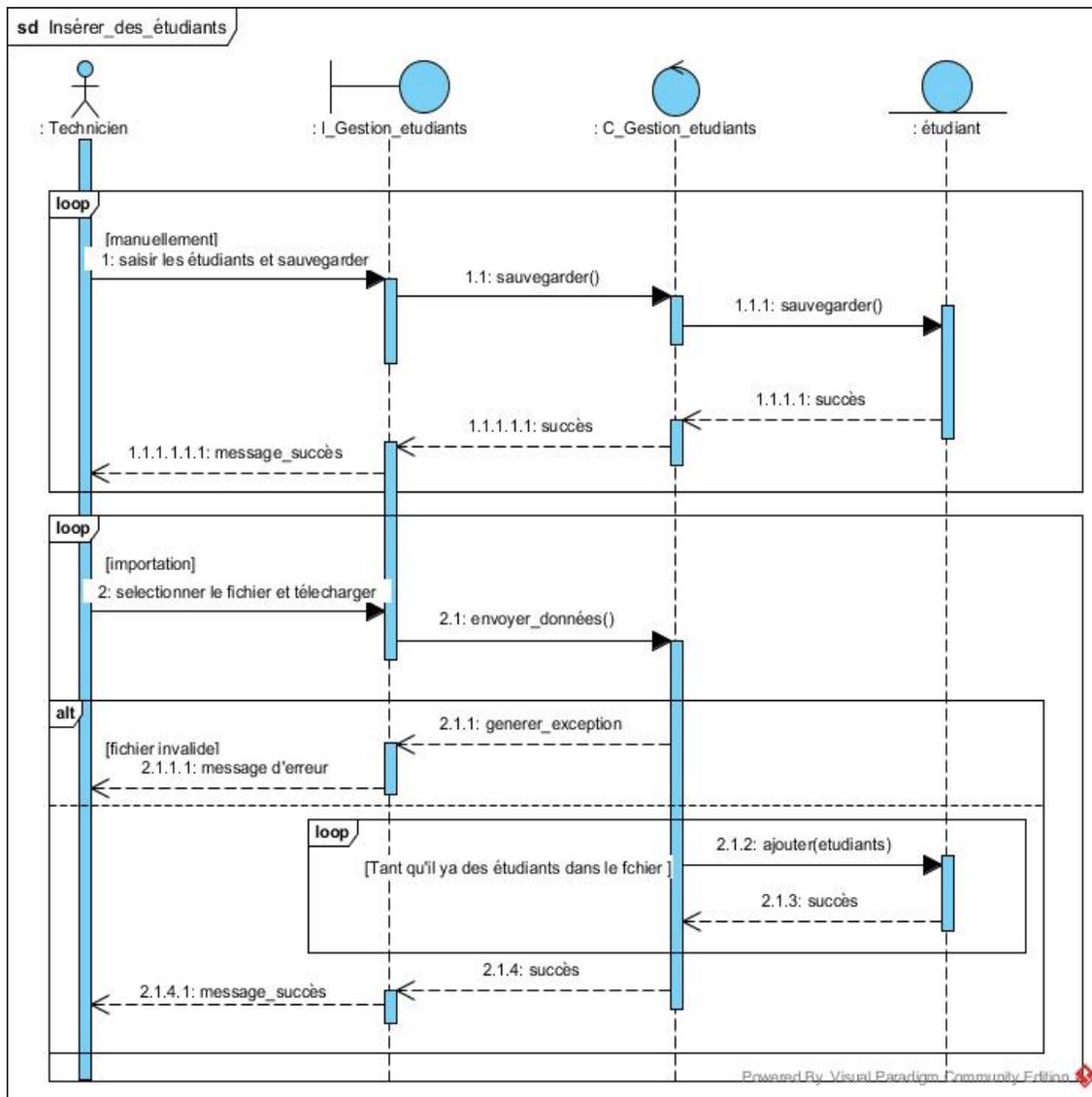


FIGURE 3.2: Diagramme séquence d'interaction « insérer des étudiants »

3.2.3 Diagramme de séquence d'interaction « créer unités d'enseignements »

La figure suivante illustre le diagramme de séquence d'interaction pour le cas d'utilisation « créer unités d'enseignements » :

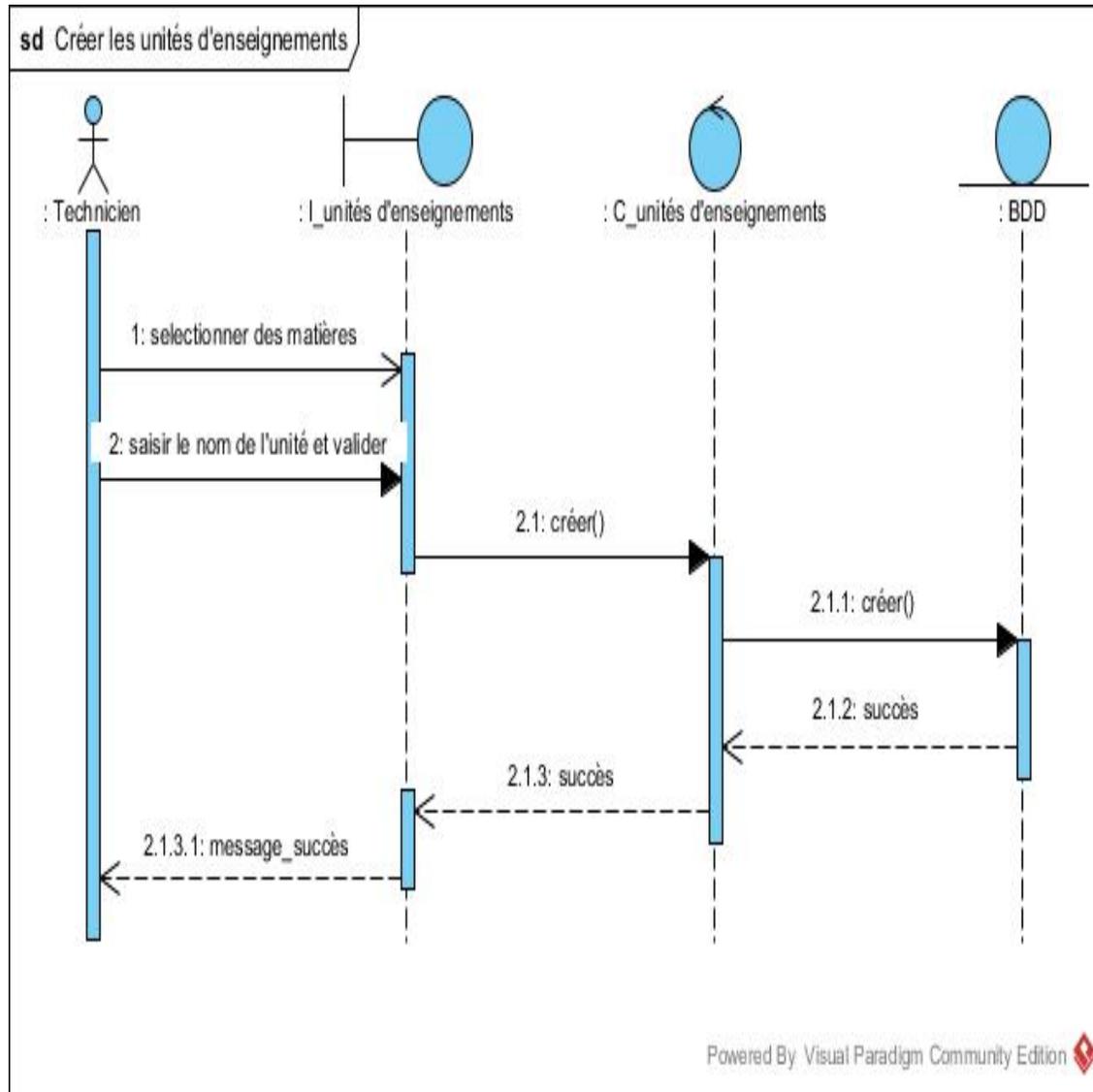


FIGURE 3.3: Diagramme séquence d'interaction « créer unités d'enseignements »

Pour les restes des diagrammes d'interaction voir Annexe C page 57.

3.3 Le diagramme de classes du domaine

3.3.1 Diagramme de classes

Le diagramme de classe est l'un des diagrammes statiques d'UML. Il permet de décrire la structure d'un système informatique tout en montrant les différentes classes, et leurs attributs,

leurs méthodes ainsi que les relations entre eux.

3.3.1.1 les éléments de base d'un diagramme de classe

Dans UML un diagramme de classe est constitué des éléments suivants :

Classe : est la description formelle d'un ensemble d'objets ayant une sémantique et des caractéristiques communs.

Objet : est une instance d'une classe. C'est une entité discrète dotée d'une identité, d'un état et d'un comportement que l'on peut invoquer. Les objets sont des éléments individuels d'un système en cours d'exécution.

Attributs : définissent les informations d'une classe ou un objet connaitre. Ils représente des données en capsules dans les objets de cette classe. Chacune initialisée. Le nom d'un attribut doit être unique dans une classe.

Opération : Dans une classe, une opération (même nom et mêmes types de paramètres) doit être unique. Quand le nom d'une opération apparaît plusieurs fois avec des paramètres différents, on dit que l'opération est surchargée. En revanche, il est impossible que deux opérations ne se distinguent que par leur valeur retournée. La déclaration d'une opération contient les types des paramètres et le type de la valeur de retour.

Classe d'association : Permet de décrire soit des attributs, soit des opérations propres à l'association.

Relations : Ces relations ne sont pas exclusives au diagramme de classes, elles peuvent également s'appliquer à l'ensemble des diagrammes statiques. ;

- Association : L'association est une connexion sémantique entre deux classes (relation logique). Une association peut être nommée. L'invocation d'une méthode est une association. Elle peut être binaire, dans ce cas elle est représentée par un simple trait, ou n-aire, les classes sont reliées à un losange par des traits simples. Ces relations peuvent être nommées. L'association n'est utilisée que dans les diagrammes de classe.
- Composition : La composition est une agrégation avec cycle de vie dépendant : la classe composée est détruite lorsque la classe mère disparaît. L'origine de cette association est représentée par un losange plein.
- Héritage : L'héritage est un principe de division par généralisation et spécialisation, représenté par un trait reliant les deux classes et dont l'extrémité du côté de la classe mère comporte un triangle.

La classe fille hérite de tous les attributs et méthodes, qu'ils soient publics, protégés ou privés. Cependant, elle ne peut pas utiliser directement les attributs et méthodes privés (que ce soit en lecture ou en écriture), sauf par l'intermédiaire d'une méthode héritée (publique ou protégée).

Le diagramme de classe de la figure suivante fournit une description détaillée des différentes classes de l'application ainsi que les différentes associations qui les relient :

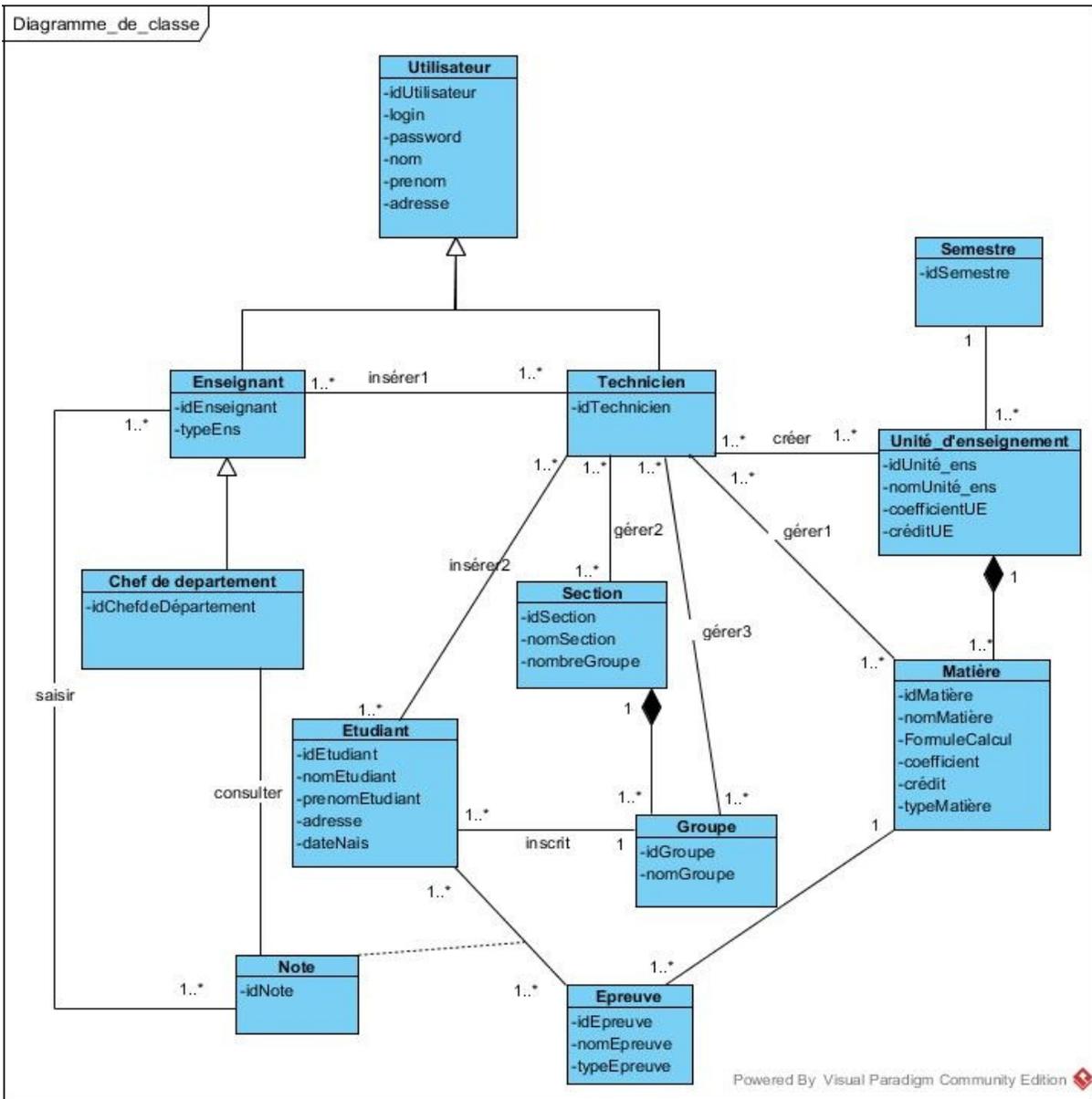


FIGURE 3.4: Diagramme de classe

3.3.2 Dictionnaire de données

Le dictionnaire de données répertorie tous les termes et leurs définitions, il est donné par le tableau suivant :

Classe Utilisateur		
Attribut	Description	Type
idUtilisateur	Identifiant de l'utilisateur	Numérique
Login	L'identifiant d'accès de l'utilisateur	Alphanumérique
Password	Mot de passe de l'utilisateur	Chaîne de Caractères
Nom	Nom de l'utilisateur	Alphabétique
Prénom	Prénom de l'utilisateur	Alphabétique
Adresse	Adresse de l'utilisateur	Alphanumérique
Classe Enseignant		
Attribut	Description	Type
idUtilisateur	Identifiant de l'enseignant	Numérique
Classe Technicien		
Attribut	Description	Type
idTechnicien	Identifiant du technicien	Numérique
Classe Matière		
Attribut	Description	Type
idMatière	Identifiant de la matière	Numérique
NomMatière	Nom de la matière	Alphabétique
FormuleCalcul	Formule de calcul de la moyenne de la matière	Chaîne de Caractères
Coefficient	Coefficient de la matière	Numérique
Crédit	Crédit de la matière	Numérique
Classe Unité d'enseignement		
Attribut	Description	Type
idUnité_ens	Identifiant de l'unité d'enseignement	Numérique
NomUnité_ens	Nom de l'unité d'enseignement	Alphabétique
CoefficientUE	La somme des coefficients des matières d'une UE	Numérique
CréditUE	La somme des crédits des matières d'une UE	Numérique
Classe Groupe		
Attribut	Description	Type
idGroupe	Identifiant de groupe	Numérique
NomGroupe	Nom de groupe	Alphabétique
Classe Section		
Attribut	Description	Type
idSection	Identifiant de la section	Numérique
NomSection	Nom de la section	Alphabétique
NombreGroupe	Nombre de groupe a la section	Numérique

Classe Etudiant		
Attribut	Description	Type
idEtudiant	Identifiant de l'étudiant	Numérique
NomEtudiant	Nom de l'étudiant	Alphabétique
PrenomEtudiant	Prénom de l'étudiant	Alphabétique
dateNaisEtudiant	Date de naissance de l'étudiant	Date
adresseEtudiant	Adresse de l'étudiant	Alphanumérique
Classe Note		
Attribut	Description	Type
idNote	Identifiant de note	Numérique
Classe Semestre		
Attribut	Description	Type
idSemestre	Identifiant de semestre	Numérique
Classe Epreuve		
Attribut	Description	Type
idEpreuve	Identifiant de l'épreuve	Numérique
NomEpreuve	Nom de l'épreuve	Alphabétique
typeEpreuve	Type de l'épreuve	Chaine de Caractères

TABLE 3.1: Dictionnaire de données

3.3.3 Schéma relationnel

3.3.3.1 Règles de passage du diagramme de classes au modèle relationnel

Le schéma de base de données relationnelle est déduit à partir du diagramme de classes entités. Le passage du diagramme de classes vers la base de données relationnelle se fait selon les règles de passage décrites ci-dessous [7] :

Règle 1 : Transformation des classes Chaque classe devient une relation. L'identifiant (respectivement les attributs) de la classe devient la clé primaire (respectivement des attributs) de la relation.

Règle 2 : Association un-à-plusieurs Il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.

Règle 3 : Association plusieurs-à-plusieurs l'association devient une relation dont la clé primaire est composée par la concaténation des identifiants des classes connectées à l'association.

Règle 4 : Association un-à-un Il faut ajouter un attribut clé étrangère, dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de la classe connectée à l'association.

Règle 5 : Transformation de l'héritage Trois décompositions sont possibles pour traduire une association d'héritage en fonction des contraintes existantes :

1. Décomposition par distinction : Il faut transformer chaque sous-classe en une relation. La clé primaire de la surclasse, migre dans la (les) relation(s) issue(s) de la (des) sous-classe(s) et devient à la fois clé primaire et clé étrangère.
2. Décomposition descendante : S'il existe une contrainte de totalité ou de partition sur l'association d'héritage, il est possible de ne pas traduire la relation issue de la surclasse. Il faut alors faire migrer tout ses attributs dans la(les) relation(s) issue(s) de la (des) sous-classe(s).
3. Décomposition ascendante : Il faut supprimer la relation issue de la sous-classe et faire migrer les attributs dans la relation issue de la surclasse.

En appliquant les règles citées ci-dessus, nous avons généré le modèle relationnel suivant :

Utilisateur(idUtilisateur, login, password, nom, prenom, adresse).

Enseignant(idEnseignant, typeEns, #idUtilisateur).

Chef de département(idChefdeDepartement, #idEnseignant).

Technicien(idTechnicien, #idUtilisateur).

Unité d'enseignement(idUnité_ens, nomUnité_ens, coefficientUE, ceditUE, #idSemestre).

Matière(idMatière, nomMatière, FormulCalcul, coefficient, crédit, #idUnité_ens).

Section(idSection, nomSection, nombreGroupe).

Groupe(idGroupe, nomGroupe, #idSection).

Etudiant(idEtudiant, nomEtudiant, prenomEtudiant, dateNais, adresse, #idGroupe).

Epreuve(idEpreuve, nomEpreuve, typeEpreuve, #idMatière).

Note(idNote, #idEpreuve, #idEtudiant).

Semestre(idSemestre).

Insérer1(#idTechnicien#idEnseignant).

Insérer2(#idTechnicien#idEtudiant).

Créer(#idTechnicien#idUnité_ens).

Saisir(#idEnseignant#idNote).

Gérer1(#idTechnicien#idMatière).

Gérer2(#idTechnicien#idSection).

Gérer3(#idTechnicien#idGroupe).

3.4 Conclusion

A travers ce chapitre, nous avons présenté notre conception de l'application. Nous avons exposé, dans un premier lieu, les différentes interactions entre les objets du système grâce aux diagrammes de séquence détaillés. Ensuite, nous avons pu déduire le schéma relationnelle à partir de diagramme de classe en respectant les règles de passage en model relationnelle. A présent, nous somme capable d'entamer la partie réalisation.

Chapitre 4

Réalisation

4.1 Introduction

Dans ce chapitre nous allons traité la partie réalisation dans laquelle nous citons l'environnement de développement mis en place (Eclipse) ensuite nous donnons un aperçu des outils et les langages de programmation utilisés. nous présenterons le diagramme de déploiement de l'application et en montrant et expliquant quelques unes de nos interfaces.

4.2 Les outils et langages utilisées

4.2.1 Eclipse

Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la fondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java .

Son objectif est de produire et fournir des outils pour la réalisation de logiciels, englobant les activités de programmation (notamment environnement de développement intégré et frameworks) mais aussi d'AGL recouvrant modélisation, conception, test, gestion de configuration, reporting... Son EDI, partie intégrante du projet, vise notamment à supporter tout langage de programmation à l'instar de Microsoft Visual Studio.

4.2.2 Java

Java est un langage de programmation et une plate-forme informatique qui ont été créés par Sun Microsystems en 1995. Beaucoup d'applications et de sites Web ne fonctionnent pas si Java n'est pas installé et leur nombre ne cesse de croître chaque jour. Java est rapide, sécurisé et fiable.

Les objectifs de Java sont d'être multiplateformes et d'assurer la sécurité aussi bien pendant le développement que pendant l'utilisation d'un programme Java. Son point fort qui le démarque des autres est sa portabilité due a sa bibliothèque de classes indépendances de la plateforme ce

qui est le point essentiel de la programmation sur internet ou plusieurs machines dissemblables sont inter-connectées [11].

4.2.3 JEE

Java EE (Enterprise Edition) est une plate-forme largement utilisée contenant un ensemble de technologies qui réduisent le coût et la complexité du développement, Java EE s'appuie sur la plate-forme Java SE et fournit des moyens standard pour traiter les transactions via Java Transaction API (JTA), les messages via Java Message Service (JMS) ou la persistance via Java Persistence API (JPA). Il peut donc être considéré comme une extension de Java SE destinée à faciliter le développement d'applications distribuées, robustes, puissantes et à haute disponibilité [1].

4.2.4 HTML

HTML5 (HyperText Markup Language 5) est la dernière révision majeure du HTML (format de données conçu pour représenter les pages web). Cette version a été finalisée le 28 octobre 2014. HTML5 spécifie deux syntaxes d'un modèle abstrait défini en termes de DOM : HTML5 et XHTML5. Le langage comprend également une couche application avec de nombreuses API, ainsi qu'un algorithme afin de pouvoir traiter les documents à la syntaxe non conforme. Le travail a été repris par le W3C en mars 2007 après avoir été lancé par le WHATWG. Les deux organisations travaillent en parallèle sur le même document afin de maintenir une version unique de la technologie.

Le html est un langage permettant d'écrire des documents hypertexte pour le web. C'est un langage très simple, basé sur des balises. Les balises servent à mettre en forme le texte, à réaliser des liens vers d'autres pages, à insérer des images...etc [13].

4.2.5 CSS

Le langage web CSS (Les feuilles de style en cascade) est un langage qui permet de gérer la présentation d'une page Web. Le langage CSS est une recommandation du World Wide Web Consortium (W3C), au même titre que HTML ou XML [14].

4.2.6 JavaScript

JavaScript est un langage de script orienté objet utilisé dans les interfaces côté client pour les applications Web. Les boîtes à outils JavaScript peuvent être utilisées pour implémenter des

composants et des fonctionnalités compatibles Ajax dans les pages Web [14].

4.2.7 AJAX

Ajax (Asynchronous JavaScript and XML) est un ensemble de techniques de développement web permettant de créer des applications web interactives. Grâce à lui, les applications récupèrent de façon asynchrone des portions de données à partir du serveur sans interférer avec la charge et le comportement de la page en cours de consultation [14].

4.2.8 Visuel Paradigm

Visual Paradigm est une suite logicielle de gestion d'entreprise et de développement logiciel primée dans le monde entier, qui fournit toutes les fonctionnalités nécessaires pour l'architecture d'entreprise, la gestion de projet, le développement logiciel et la collaboration d'équipe dans une solution à guichet unique [9].

4.2.9 Serveur Tomcat

Tomcat est un serveur HTTP à part entière. De plus, il gère les servlets et les JSP (par un compilateur Jasper compilant les pages JSP pour en faire des servlets). Tomcat a été écrit en langage Java. Il peut donc s'exécuter via la machine virtuelle Java sur n'importe quel système d'exploitation la supportant.

4.2.10 Bootstrap

Bootstrap, un framework CSS, fait partie de ces outils qui accélèrent grandement un projet web. Il étend les possibilités du langage CSS en ajoutant plein de fonctionnalités et de composants. Je dirais plutôt, on peut avoir le même résultat que du CSS en utilisant seulement Bootstrap. Même si c'est un framework CSS, il ajoute également un plus au HTML et au Javascript [14].

4.3 Schéma Physique de la Base de donnée

Un schéma de base de données physique expose la manière dont les données sont stockées physiquement sur un système de stockage en termes de fichiers et d'index.

La figure suivante représente le schéma physique de la base de données :

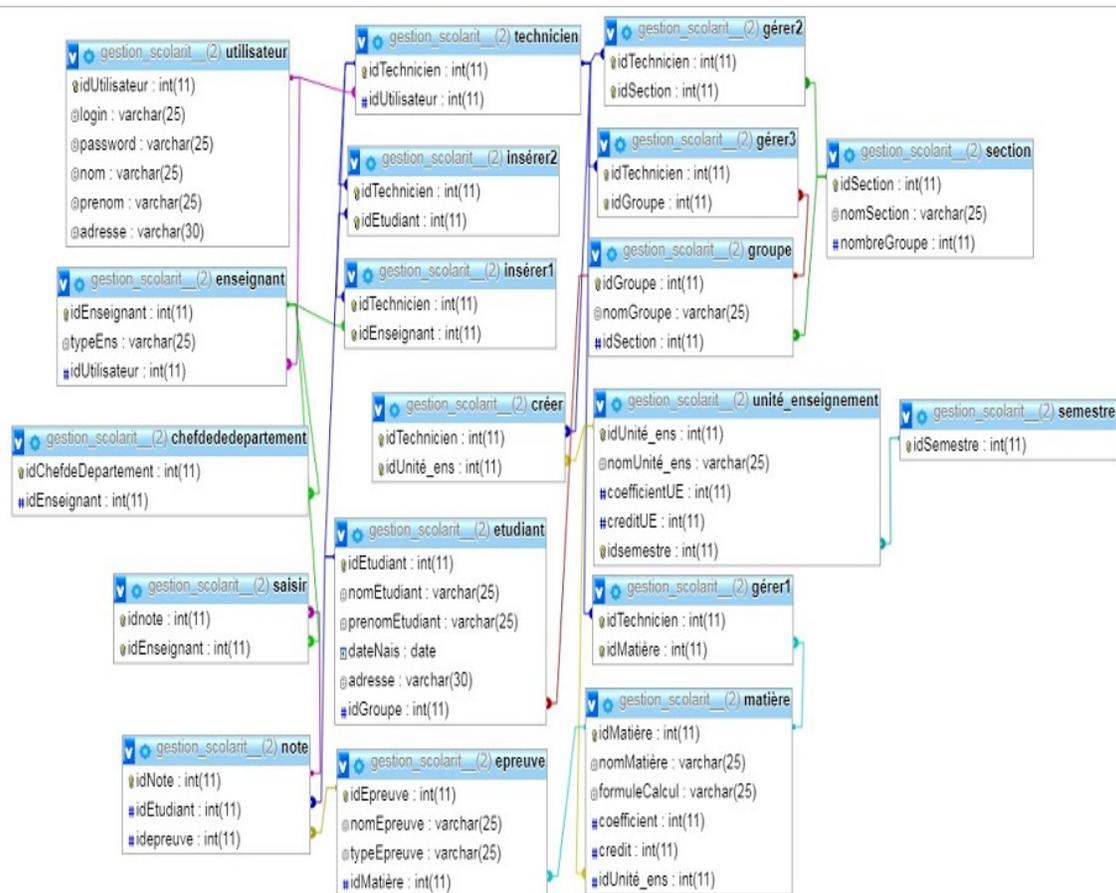


FIGURE 4.1: Schéma Physique de la Base de donnée

4.4 Diagramme de Déploiement

Le diagramme de déploiement du système que nous avons réalisé, est illustré par la figure suivante :

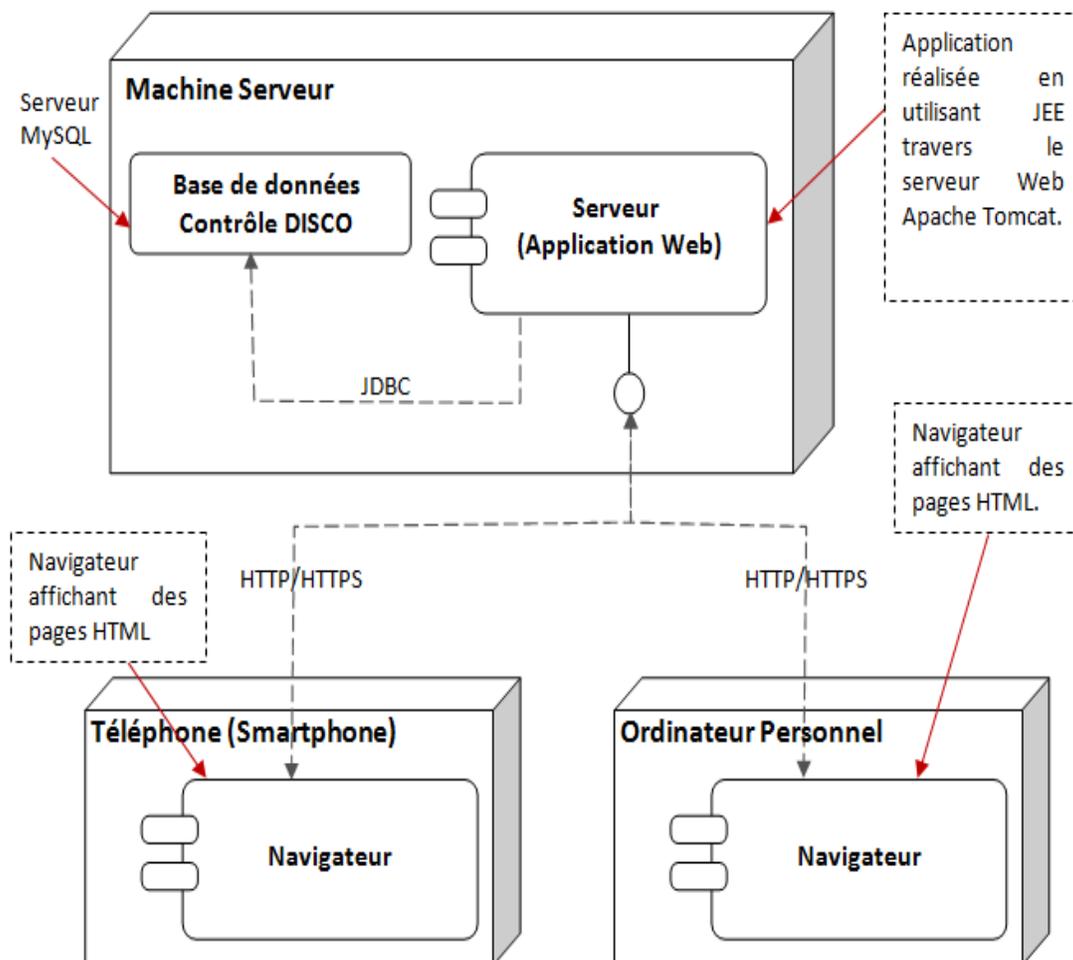


FIGURE 4.2: Diagramme de Déploiement

4.5 Présentation de quelques interfaces de l'application

Nous allons présenter, dans ce qui suit, quelques exemples représentatifs de l'IHM de notre application.

4.5.1 Interface d'authentification

L'utilisateur doit remplir les deux champs d'authentification en saisissant les coordonnées correctes afin qu'il puisse accéder à son espace.

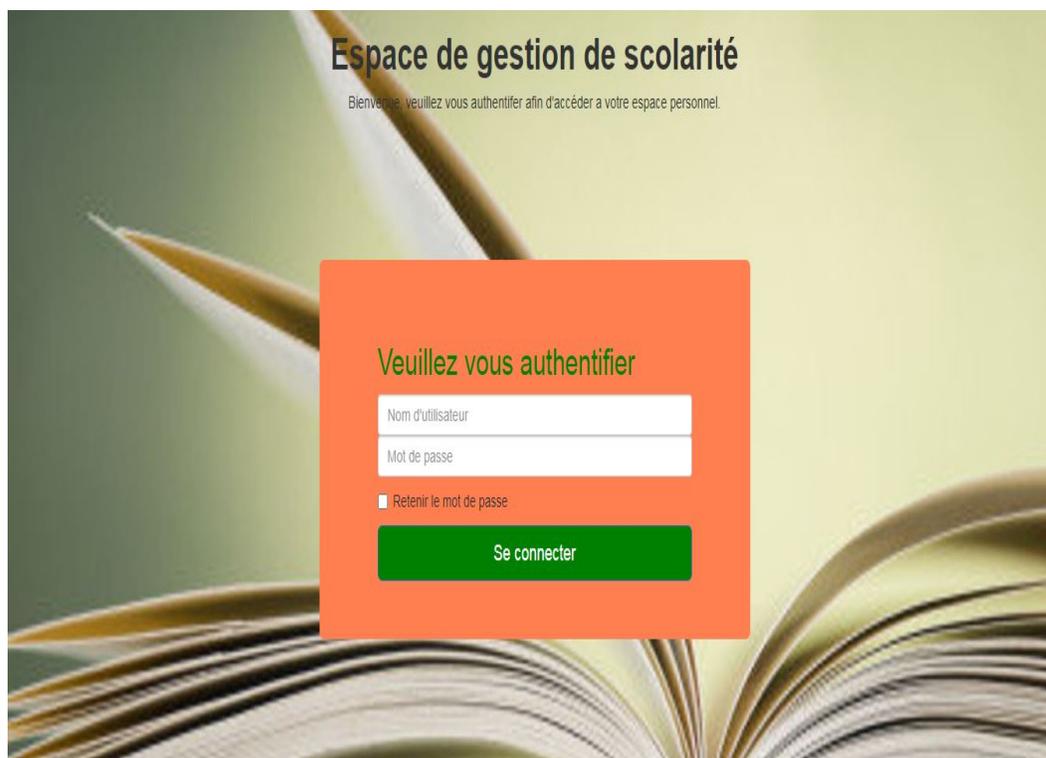


FIGURE 4.3: Page authentification

4.5.2 Interface de technicien

Après que le technicien saisit ses informations, il peut accéder à son espace.



FIGURE 4.4: Page de technicien

4.5.3 Interface d'ajout d'un étudiant

Le technicien peut ajouter des étudiants manuellement ou par l'importation excel.

Espace technicien deconnexion

Espace d'ajout d'etudiant

Informations sur l'etudiant

Nom:

Prenom:

Date de naissance: 01/06/2017

adresse:

[Ajouter](#)

• [Importation excel](#)

FIGURE 4.5: Page d'ajout d'un étudiant

4.5.4 Interface d'ajout d'un enseignant

Le technicien remplit le formulaire de l'enseignant et il l'ajoute.

Espace technicien deconnexion

Espace d'ajout d'enseignant

Informations sur l'enseignant

Nom:

Prenom:

Type:

Affiliation: oui non

Module:

login:

mot de passe:

FIGURE 4.6: Page d'ajout d'un enseignant

4.6 Conclusion

Ce dernier chapitre représente une récapitulation de tout le travail élaboré pendant ce projet de conception et de développement ainsi qu'une représentation des résultats atteints. En effet, nous avons décrit les environnements matériels et logiciels sur lesquels nous avons construit notre application. Nous avons ensuite présenté le schéma physique de la base de donnée, Suivi du diagramme de déploiement associé à notre système. Enfin, nous avons clôturé ce chapitre par la présentation de quelques interfaces de l'application.

Conclusion Générale et perspectives

Dans ce rapport nous avons présenté les différentes étapes de la conception et de la mise en œuvre de notre application gestion de scolarité destiné au département de technologie.

Afin de pouvoir atteindre cet objectif nous avons abordé le problème en s'appuyant sur la méthode SCRUM.

Cette dernière est loin d'être une démarche classique, et avec SCRUM on a pas eu a gérer le cahier des charges traditionnel, le recueil des besoins fonctionnels ce sont fais au fur et a mesure de l'avancement de la réalisation.

Nous avons fais la modélisation graphique du système a l'aide de diagramme UML et pour la réalisation nous avons opté pour JEE.

L'objectif premier de notre projet était de faciliter la gestion de la scolarité du département de technologie. Ce travail a abouti à la création d'une application web 3 tiers qui s'occupe de différentes tâches et activités reliées a cette scolarité dont nous pouvons citer : saisie des états d'avancement (TPs, TDs et cours), insérer les enseignants intervenant au niveau de département (Affiliés ou non-affiliés) et leurs types : permanents, associés ou vacataires et gérer les gestion des matières, groupes et sections, ...

Au cours de ce projet nous avons exploité toutes nos connaissances et consacré tout le temps nécessaire pour pouvoir créer le meilleur, dans les limites de nos espérances.

Cependant nous aurions souhaité avoir plus de temps pour améliorer notre travail et apporter plus de solutions.

Comme perspectives, nous tenons a améliorer notre application, du coté interface nous voudrions présenter notre application en utilisant la technologie single page. Et nous pouvons l'enrichir aussi par d'autres fonctionnalités : notamment un système pour la gestion des emplois du temps et qui se base principalement sur la gestion des éléments d'enseignement et des enseignants qui sont déjà déployés dans notre application.

Annexe A

A.1 Méthodes d'analyse

A.1.1 La méthode agile

La méthode agile est une approche itératif et incrémentale, qui est mené dans un esprit collaboratif avec juste ce qui faut de formalisme. Elle génère un produit de haut qualité tout en prenant en compte les besoins du client [12].

Origine et valeurs des méthodes agiles

Le mouvement des méthodes agiles est né en 2001 aux États-Unis. Devant l'observation faite du taux important d'échec des projets, dix-sept experts en développement logiciel, se sont réunis afin d'échanger et de trouver un socle commun de valeurs et de bonnes pratiques. Le résultat de cette réflexion a abouti au Manifeste pour le développement logiciel agile et la création de l'Agile Alliance. Les quatre valeurs du Manifeste :

- Les individus et leurs interactions avant les processus et les outils.
- Des fonctionnalités opérationnelles avant la documentation.
- Collaboration avec le client plutôt que contractualisation des relations.
- Acceptation du changement plutôt que conformité aux plans.

A.1.2 SCRUM

SCRUM est une méthode agile dédiée à la gestion de projet . Cette méthode de gestion, ou plutôt ce Framework de management de projet, à pour objectif d'améliorer la productivité de son équipe dans les meilleurs délais [2].

SCRUM permet de produire la plus grande valeur métier en une durée la plus courte. Elle est caractérisée par le fait que :

- Des règles et rôles plutôt que des notions.

- Des réunions quotidiennes appelées "cérémonies agiles" ou daily scrum.

Avantage de SCRUM

Répond aux problématiques des autres méthodes traditionnelles :

- Manque de communication.
- La mauvaise compréhension des besoins.
- La difficulté a prendre en compte les changement au cours du projet.
- L'insuffisance des testes et le suivi de la réalisation.

Pilier de SCRUM

SCRUM repose sur trois piliers fondamentaux. Il s'agit de la transparence, l'inspection et l'adaptation. Ces trois piliers sont étroitement liés les uns aux autres. Pour les appliquer, Nous utilisons une méthode empirique. **Transparence** : tous les indicateurs relatifs au développement sont visibles de tous.

Inspection : se remettre en question a chaque sprint "itérations de quelques semaines" quotidiennement.

Adaptation :étape qui vient après l'inspection où l'équipe de développement s'adapte aux changements .

Les rôles dans SCRUM

- Product owner : C'est le représentant du client ou l'utilisateur lui même, il doit être en mesure d'expliquer au Scrum master et à l'équipe de développement l'objectif de l'application, il est considéré comme responsable de succès de projet.
- Scrum Master : Son objectif principal est de protéger l'équipe de développement des perturbations extérieures.
- Development team : L'équipe est responsable de développer le produit logiciel commandé par le gestionnaire de produit (product owner). les membres de l'équipe doivent être pluridisciplinaire, constituée de 4 à 10 personnes ayant l'expertise nécessaire pour assurer la réalisation du produit.

Ergonomie

L'ergonomie est l'étude scientifique de la relation entre l'homme et ses moyens, méthodes et milieux de travail. Elle est caractérisée par sa vocation à adapter les outils à l'homme.

L'ergonomie Web consiste a adapter un site web a ses utilisateurs , analyser ce dont ils ont besoins et mettre en œuvre des méthodes pour que l'interface Finale réponde à ces besoins. On dit qu'un site web est ergonomique si il est utile et utilisable, c'est a dire si il satisfait l'utilisateur et si il lui permet d'accomplir ses objectifs rapidement et avec le moins d'erreurs possibles [4].

A.1.3 UML (Unified Modeling Language)

UML, se définit comme un langage de modélisation graphique et textuel destiné à comprendre et à définir des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles,

concevoir des solutions et communiquer des points de vue. UML modélise l'ensemble des données et des traitements en élaborant des différents diagrammes. En clair, il ne faut pas désigner UML en tant que méthode (Il y manque la démarche) mais plutôt comme une boîte d'outils qui sert à améliorer les méthodes de travail [5].

A.1.3.1 Les diagrammes UML

Les diagrammes sont répartis en deux grandes catégories présenter dans la figure qui suit [6] :

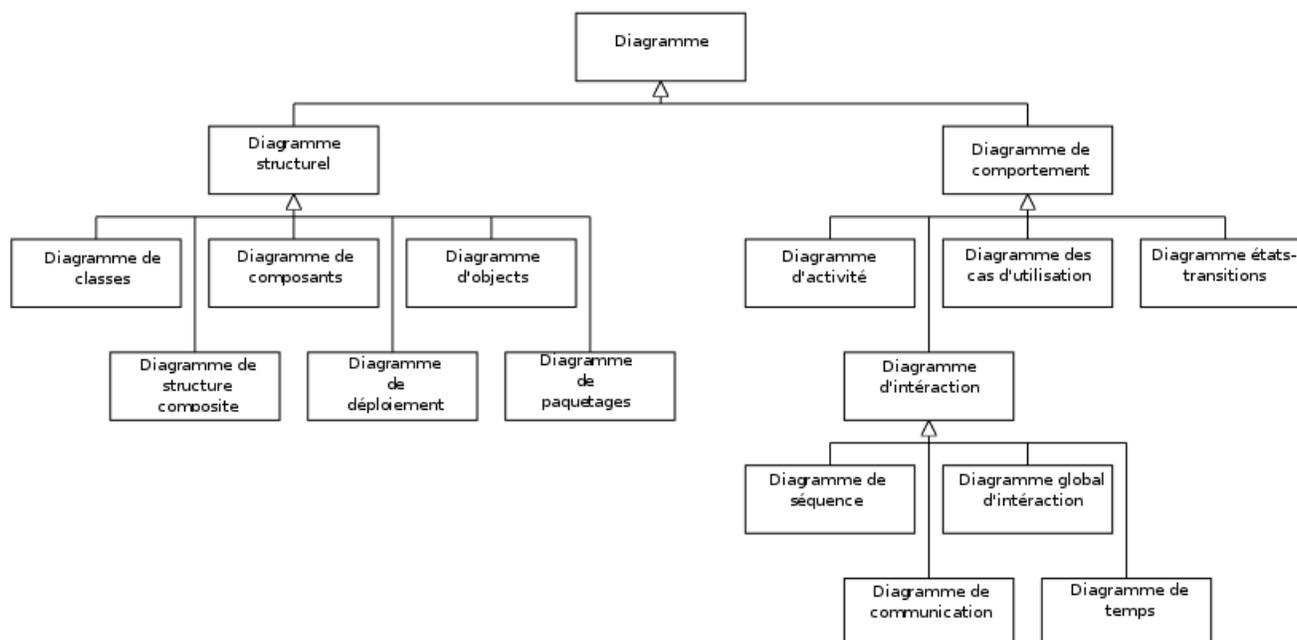


FIGURE A.1.7: Diagrammes UML

Annexe B

B.1 Description textuelle par cas d'utilisation

B.1.1 Description textuelle de cas d'utilisation « gestion des matières »

les tableaux suivants représentent la description textuelle de cas d'utilisation « gérer les matières » :

B.1.1.1 Description textuelle de cas d'utilisation « Ajouter matière »

Cas d'utilisation	Ajouter une matière
Acteur	Technicien
Pré-condition	Authentification préalable Le technicien clique sur le menu matière, un écran est affiché, L'option ajouter matière est choisit
Post-condition	Une matière a été bien ajoutée
Scénario nominal	1-Le technicien saisir les informations et valide 2-le système vérifier les informations saisies 3-le système enregistre la nouvelle matière et affiche un message de succès
Scénario alternatif	2-1-le technicien saisit des informations manquantes 2-1-1-le système affiche un message d'erreur 2-1-2-reprise de l'étape 1 de scénario

TABLE 1: Description textuelle Ajouter une matière

B.1.1.2 Description textuelle de cas d'utilisation « Modifier matière »

Cas d'utilisation	Modifier une matière
Acteur	Technicien
Pré-condition	Authentification préalable Le technicien clique sur le menu matière, un écran est affiché L'option modifier matière est choisit
Post-condition	Les informations ont été bien modifiées
Scénario nominal	1-le technicien choisit une matière 2-le technicien modifie les informations et valide 3- le système vérifier les informations modifiées 4-le système enregistre les informations et affiche un message de succès
Scénario alternatif	3-1-le technicien oublie de remplir l'un des champs 3-1-1-le système affiche un message d'erreur 3-1-2-reprise de l'étape 2 de scénario

TABLE 2: Description textuelle Modifier une matière

B.1.1.3 Description textuelle de cas d'utilisation « Supprimer matière »

Cas d'utilisation	Supprimer une matière
Acteur	Technicien
Pré-condition	Authentification préalable Le technicien clique sur le menu matière, un écran est affiché L'option supprimer une matière est choisit
Post-condition	Une matière a été bien supprimée
Scénario nominal	1-Le technicien sélectionne une matière et supprime 2-le système affiche le message de confirmation 3-le technicien confirme la suppression 4-le système supprime la matière et affiche un message de succès
Scénario alternatif	3-1-le technicien annule son choix 3-1-1-le système annule la suppression

TABLE 3: Description textuelle Supprimer une matière

B.1.1.4 Description textuelle de cas d'utilisation « Attribuer une formule à une matière »

Cas d'utilisation	Attribuer une formule
Acteur	Technicien
Pré-condition	Authentification préalable Le technicien clique sur le menu matière, un écran est affiché L'option attribuer formule est choisit
Post-condition	Une formule a été bien enregistrée
Scénario nominal	1-Le technicien sélectionne une matière et ouvrir 2-le système affiche l'écran de saisie 3-le technicien saisir la formule est sauvegarde 4-le système enregistre la formule et affiche un message de succès

TABLE 4: Description textuelle Attribuer une formule

B.1.2 Description textuelle de cas d'utilisation « gestion des groupes »

Le tableau suivant représente la description textuelle de cas d'utilisation « gérer les groupes » :

Cas d'utilisation	Gérer les groupes
Acteur	Technicien
Pré-condition	Authentification préalable
Post-condition	Le technicien gère les groupes
Scénario nominal	1-le technicien demande de gérer les groupes 2-le système affiche la page « gestion des groupes » 3-le technicien choisit l'une des options « ajouter, modifier, supprimer » 4-le système exécute l'option choisie

TABLE 5: Description textuelle Gérer les groupes

B.1.3 Description textuelle de cas d'utilisation « gestion des enseignants »

Le tableau suivant représente la description textuelle de cas d'utilisation « gestion des enseignants » :

Cas d'utilisation	Gestion des enseignants
Acteur	Technicien
Pré-condition	Authentification préalable
Post-condition	Le technicien gère la gestion des enseignants
Scénario nominal	1-le technicien demande la page de gestion des enseignants 2-le système affiche la page « gestion des enseignants » 3-le technicien choisit l'une des options « ajouter, modifier, supprimer » 4-le système exécute l'option choisie

TABLE 6: Description textuelle gestion des enseignants

B.1.4 Description textuelle de cas d'utilisation « Saisir les notes »

Les figures suivantes représentent la description textuelle de cas d'utilisation « Saisir les notes » :

B.1.4.1 Description textuelle de cas « Saisir manuellement les notes »

Cas d'utilisation	Manuellement
Acteur	Enseignant
Pré-condition	Authentification préalable Le technicien clique sur la gestion des notes, un écran est affiché L'option saisir les notes manuellement est choisit
Post-condition	Les notes ont été bien enregistrées
Scénario nominal	1-l'enseignant saisit les notes et sauvegarde 2-le système enregistre les notes et affiche un message de succès

TABLE 7: Description textuelle Saisir les notes

B.1.4.2 Description textuelle de cas « Importer les notes »

Cas d'utilisation	Importation
Acteur	Enseignant
Pré-condition	Authentification préalable Le technicien clique sur la gestion des notes, un écran est affiché L'option importer les notes est choisit
Post-condition	Les notes ont été bien importées
Scénario nominal	1-l'enseignant choisit un fichier et le télécharge 2-le système enregistre les notes et affiche un message de succès

TABLE 8: Description textuelle importer les notes

B.1.5 Description textuelle de cas d'utilisation « consulter les notes »

La figure suivante représente la description textuelle de cas d'utilisation « consulter les notes » :

Cas d'utilisation	consulter les notes
Acteur	Chef de département
Pré-condition	Authentification préalable
Post-condition	La liste des notes est affichée sur l'écran
Scénario nominal	1-le chef de département demande de consulter les notes 2-le système affiche la liste des notes

TABLE 9: Description textuelle consulter les notes

B.2 Diagrammes de sequence

B.2.1 Diagramme de sequence de cas d'utilisation « gestion des matières »

»

Les figures suivantes représentent les diagrammes de sequence de cas d'utilisation « gestion des matières » :

B.2.2 Diagramme de sequence de cas « ajouter matière »

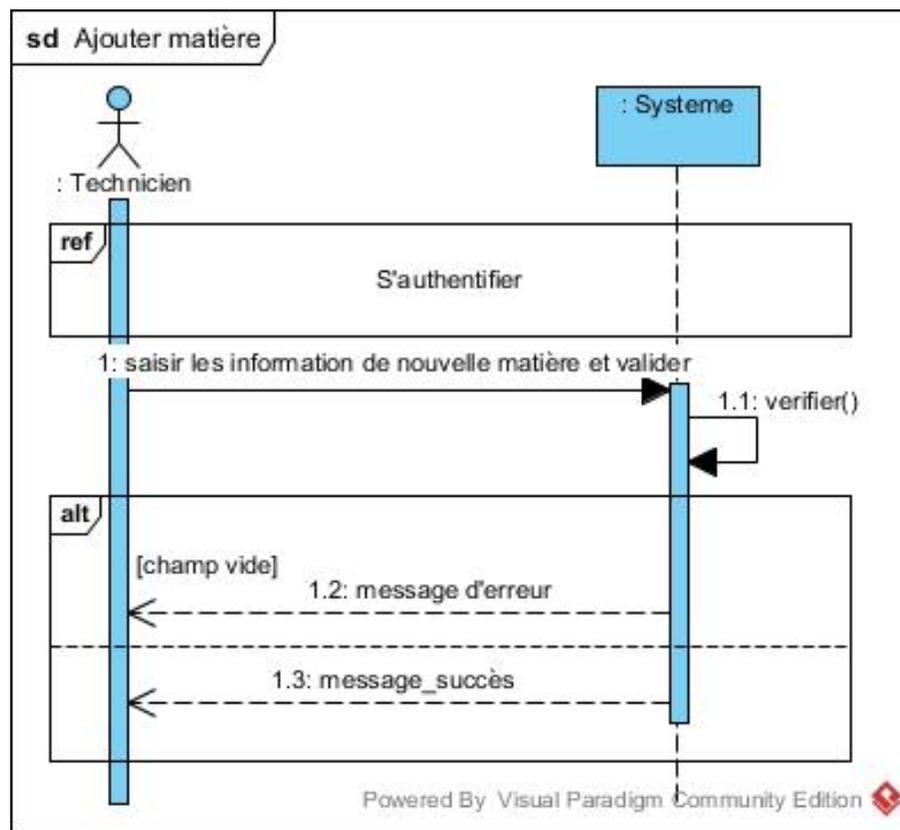


FIGURE B.2.1: Diagramme sequence cas d'utilisation « ajouter matière »

B.2.3 Diagramme de sequence de cas « modifier matière »

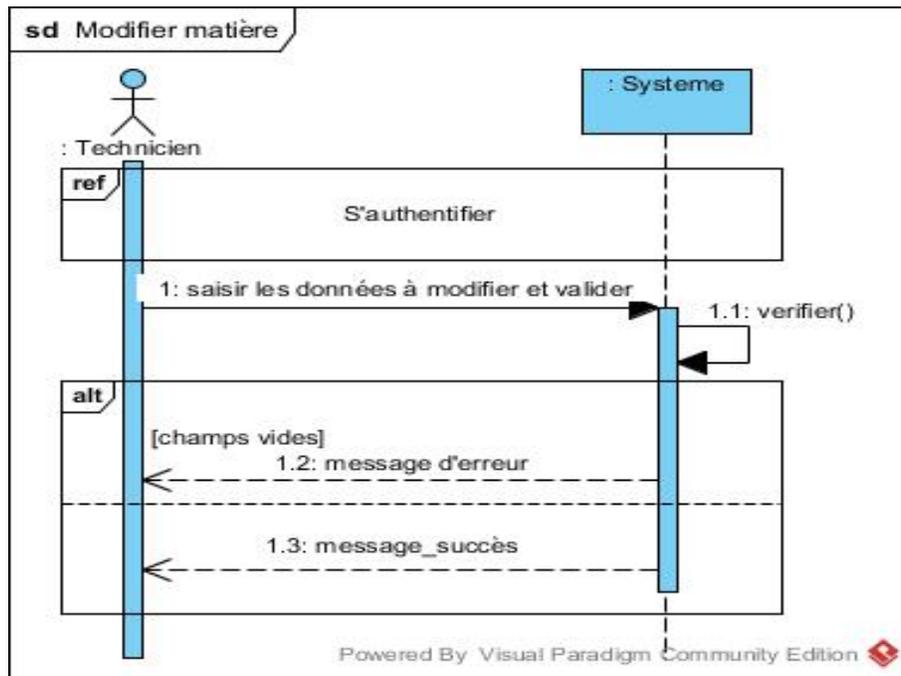


FIGURE B.2.2: Diagramme sequence cas d'utilisation « modifier matière »

B.2.4 Diagramme de sequence de cas « supprimer matière »

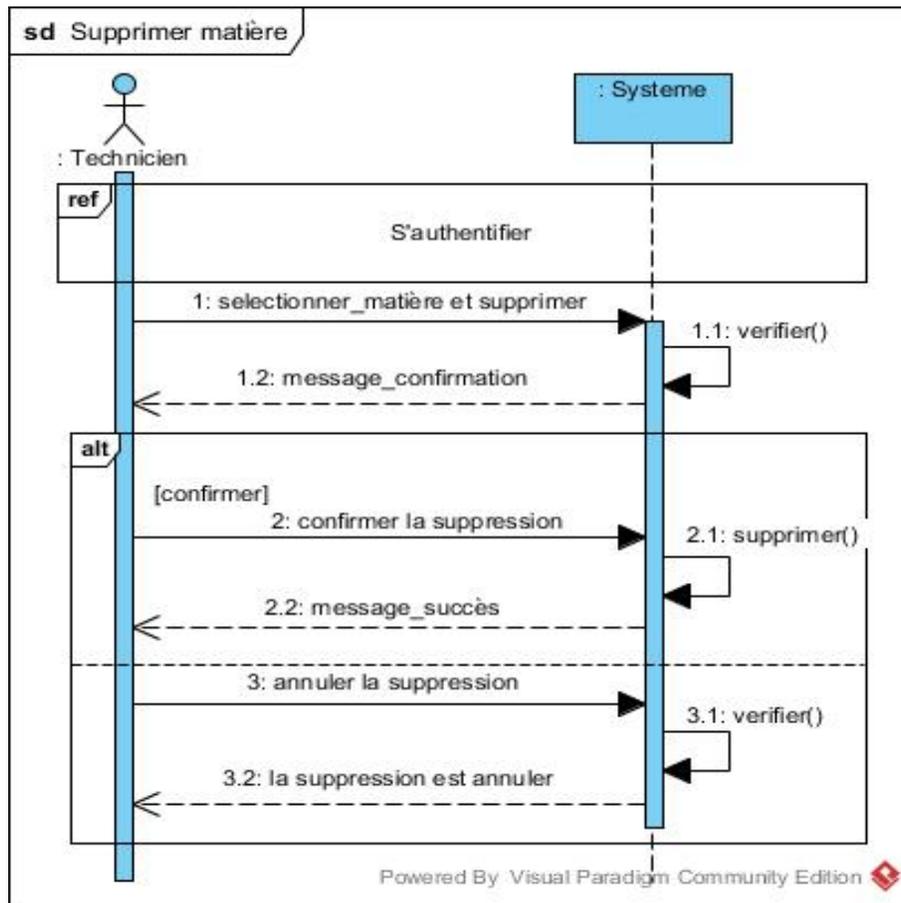


FIGURE B.2.3: Diagramme sequence cas d'utilisation « supprimer matière »

B.2.5 Diagramme de sequence de cas « attribuer formule »

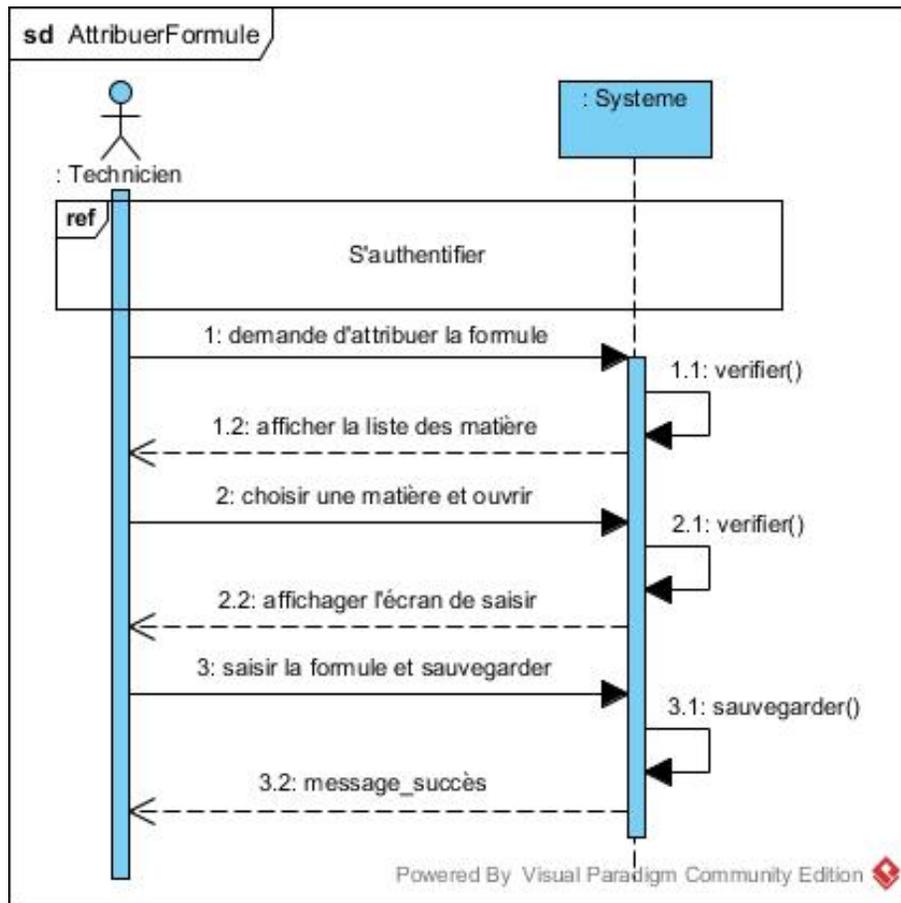


FIGURE B.2.4: Diagramme sequence cas d'utilisation « attribuer formule »

B.2.6 Diagramme de sequence de cas d'utilisation « gestion des groupes »

»

La figure suivante représente le diagramme de sequence de cas d'utilisation « gestion des groupes » :

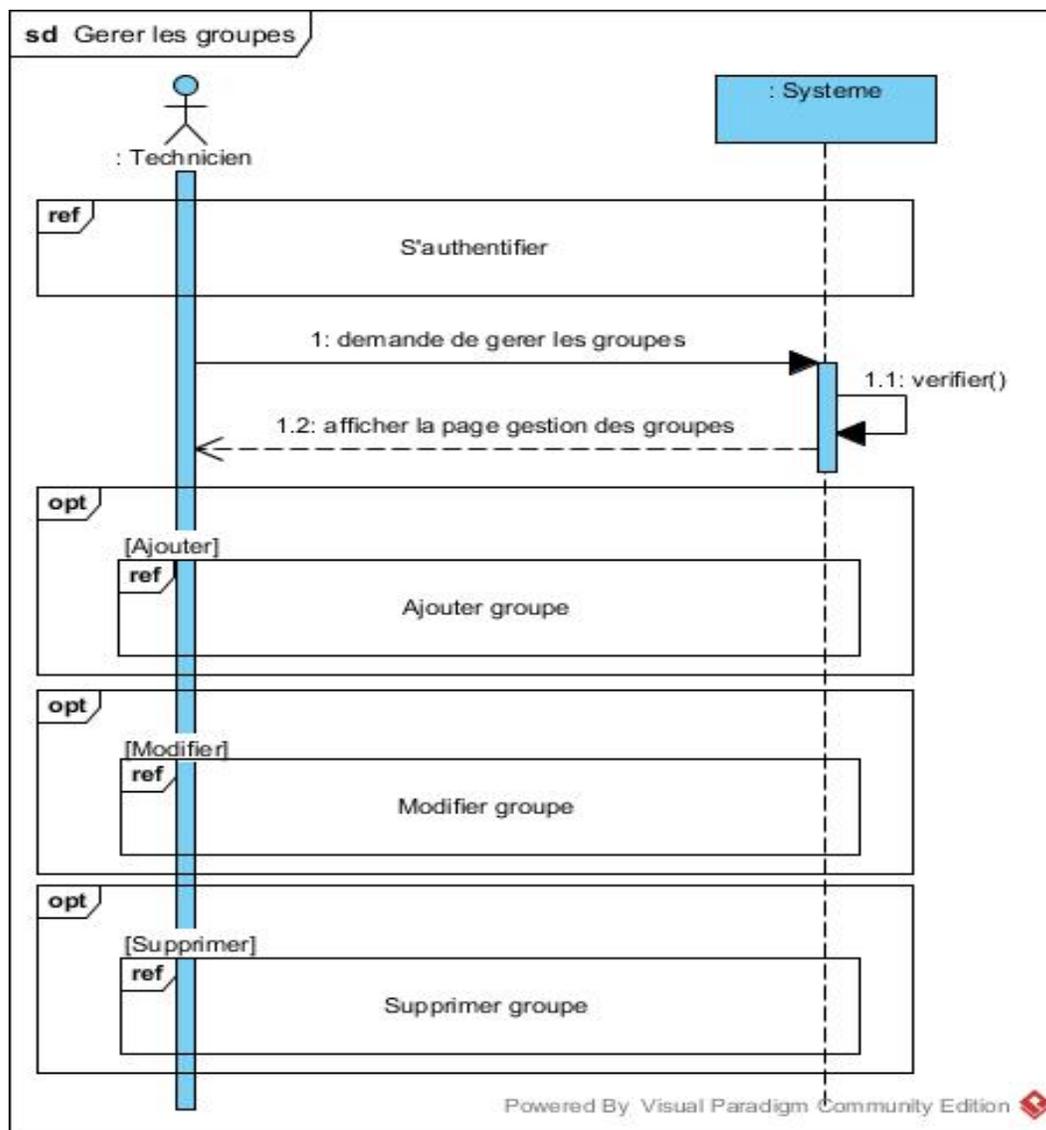


FIGURE B.2.5: Diagramme sequence cas d'utilisation « gestion des groupes »

B.2.7 Diagramme de sequence de cas d'utilisation « gestion des enseignants »

La figure suivante représente le diagramme de sequence de cas d'utilisation « gestion des enseignants » :

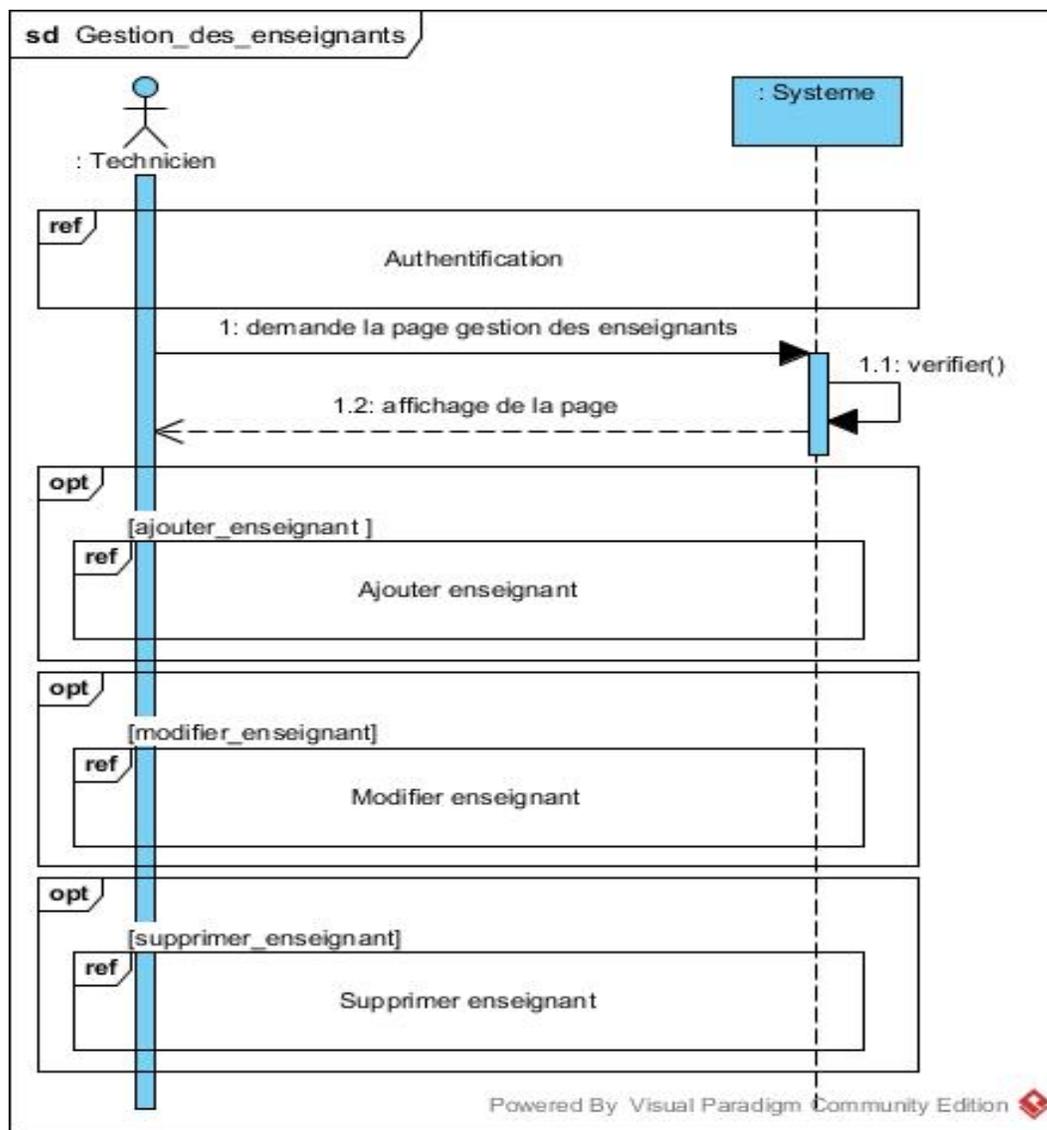


FIGURE B.2.6: Diagramme sequence cas d'utilisation « gestion des enseignants »

B.2.9 Diagramme de sequence de cas d'utilisation « consulter les notes »

La figure suivante représente le diagramme de sequence de cas d'utilisation « consulter les notes » :

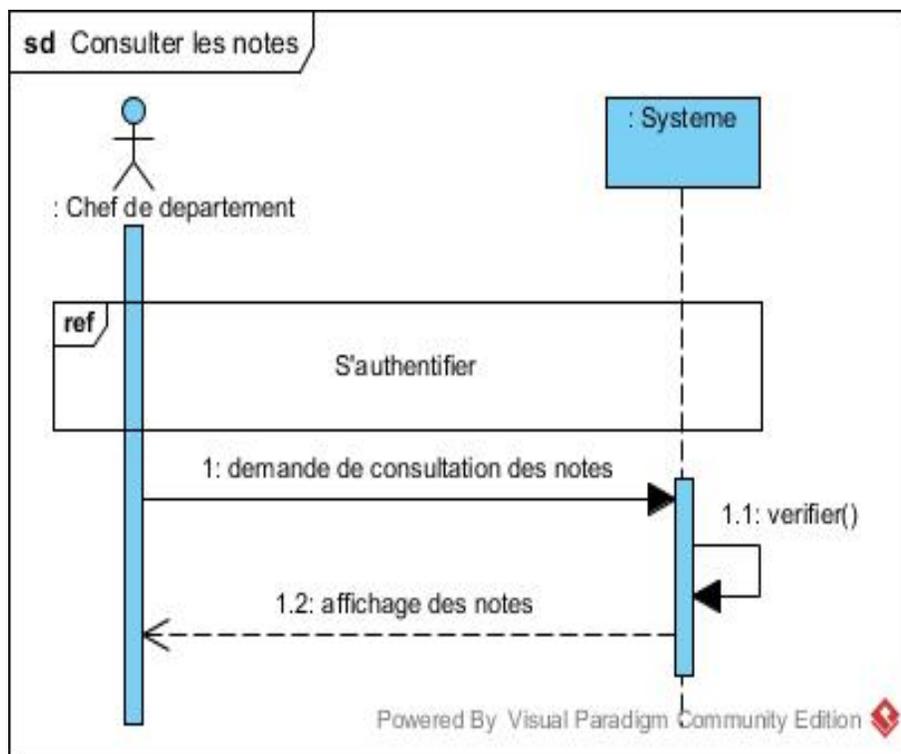


FIGURE B.2.8: Diagramme sequence cas d'utilisation « consulter les notes »

Annexes C

C.1 Diagrammes de séquence d'interaction

C.1.1 Diagramme de séquence d'interaction « gestion des matières »

Les figures suivantes illustrent les diagrammes de séquence d'interaction pour le cas d'utilisation « gestion des matières » :

C.1.1.1 Diagramme de séquence d'interaction de cas « ajouter matière »

La figure suivante représente le diagramme de séquence de cas d'utilisation « ajouter matière » :

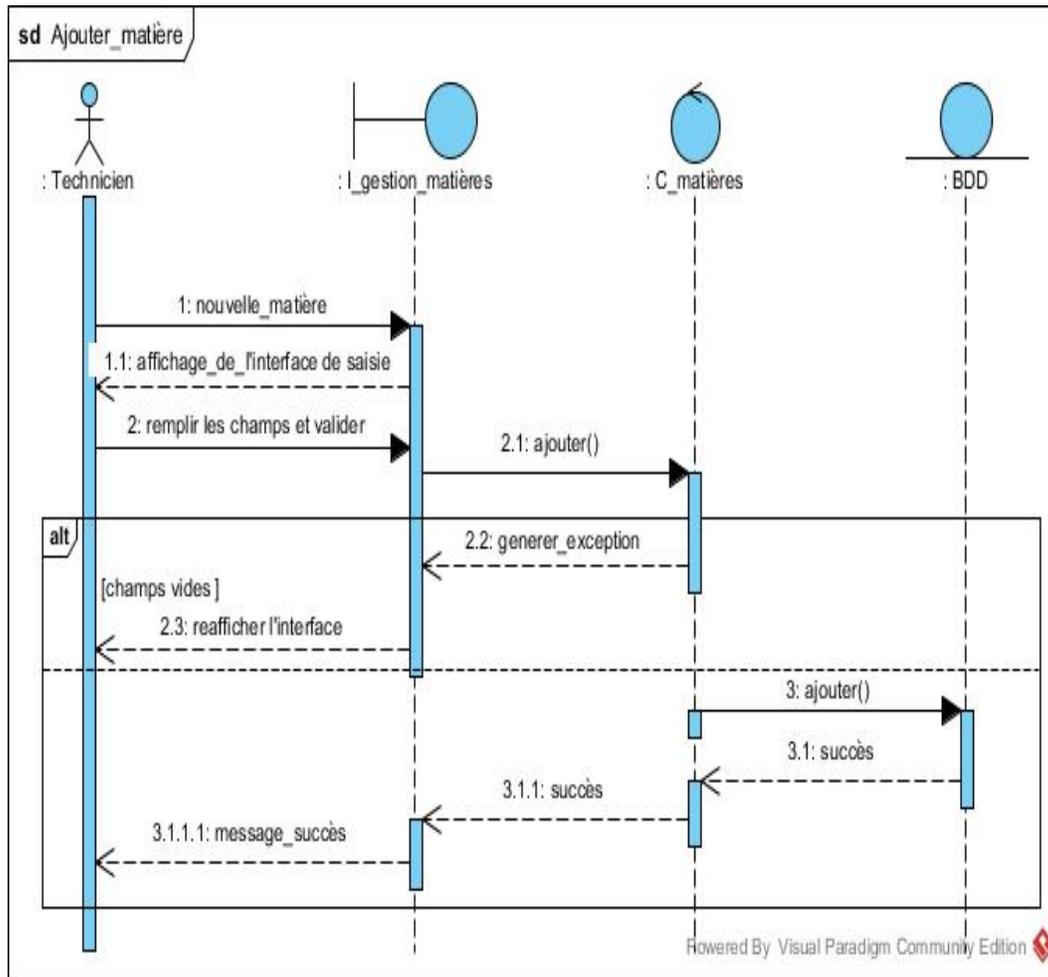


FIGURE C.1.1: Diagramme séquence d'interaction « ajouter une matière »

C.1.1.2 Diagramme de séquence d'interaction de cas « modifier matière »

La figure suivante représente le diagramme de séquence de cas d'utilisation « modifier matière » :

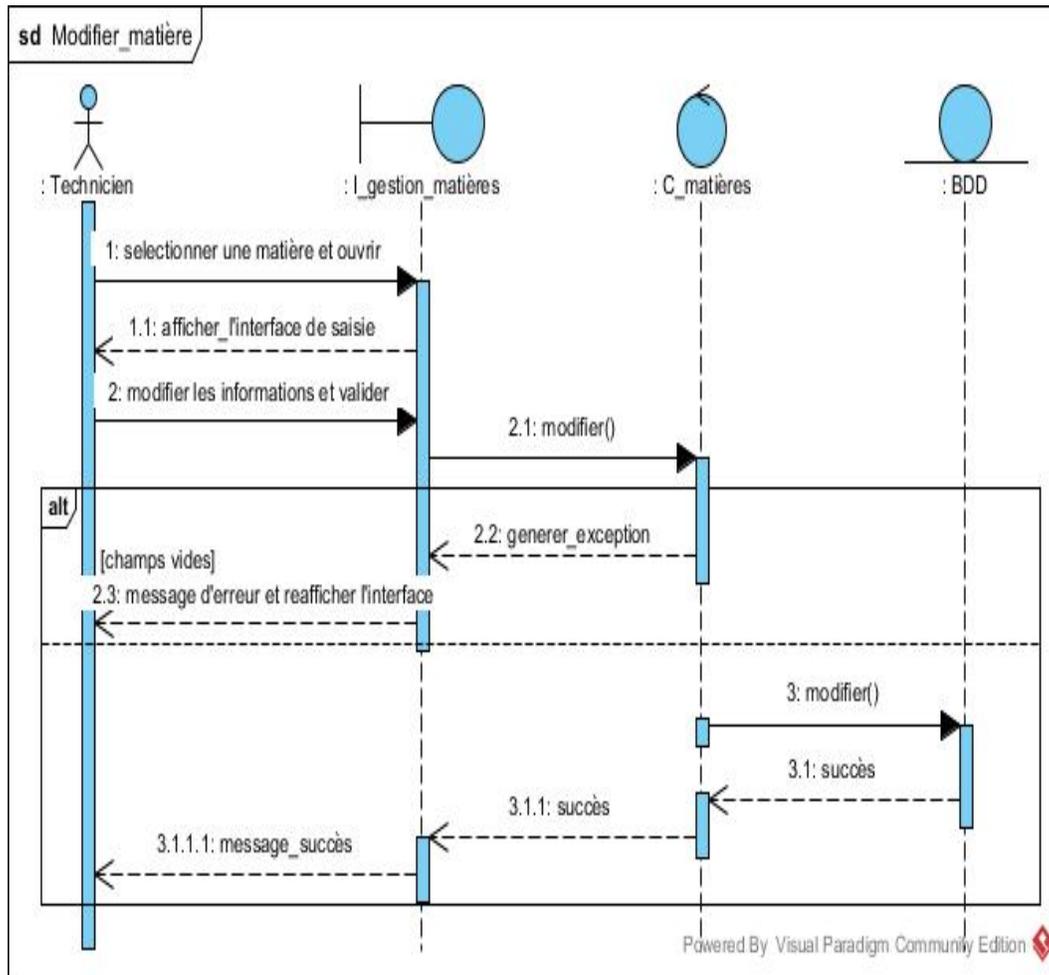


FIGURE C.1.2: Diagramme séquence d'interaction « modifier une matière »

C.1.1.3 Diagramme de séquence d'interaction de cas « supprimer matière »

La figure suivante représente le diagramme de séquence de cas d'utilisation « supprimer matière » :

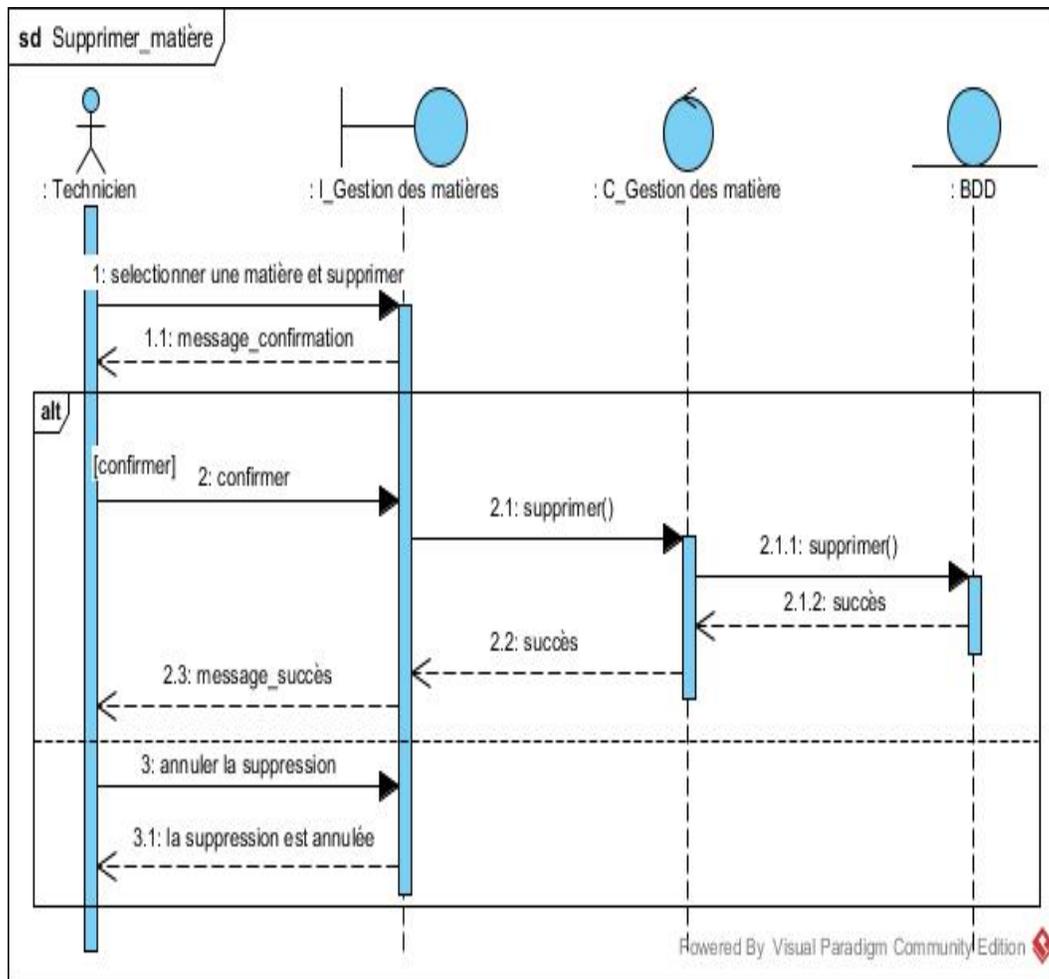


FIGURE C.1.3: Diagramme séquence d'interaction « supprimer une matière »

C.1.1.4 Diagramme de séquence d'interaction de cas « attribuer formule »

La figure suivante représente le diagramme de séquence de cas d'utilisation « attribuer formule » :

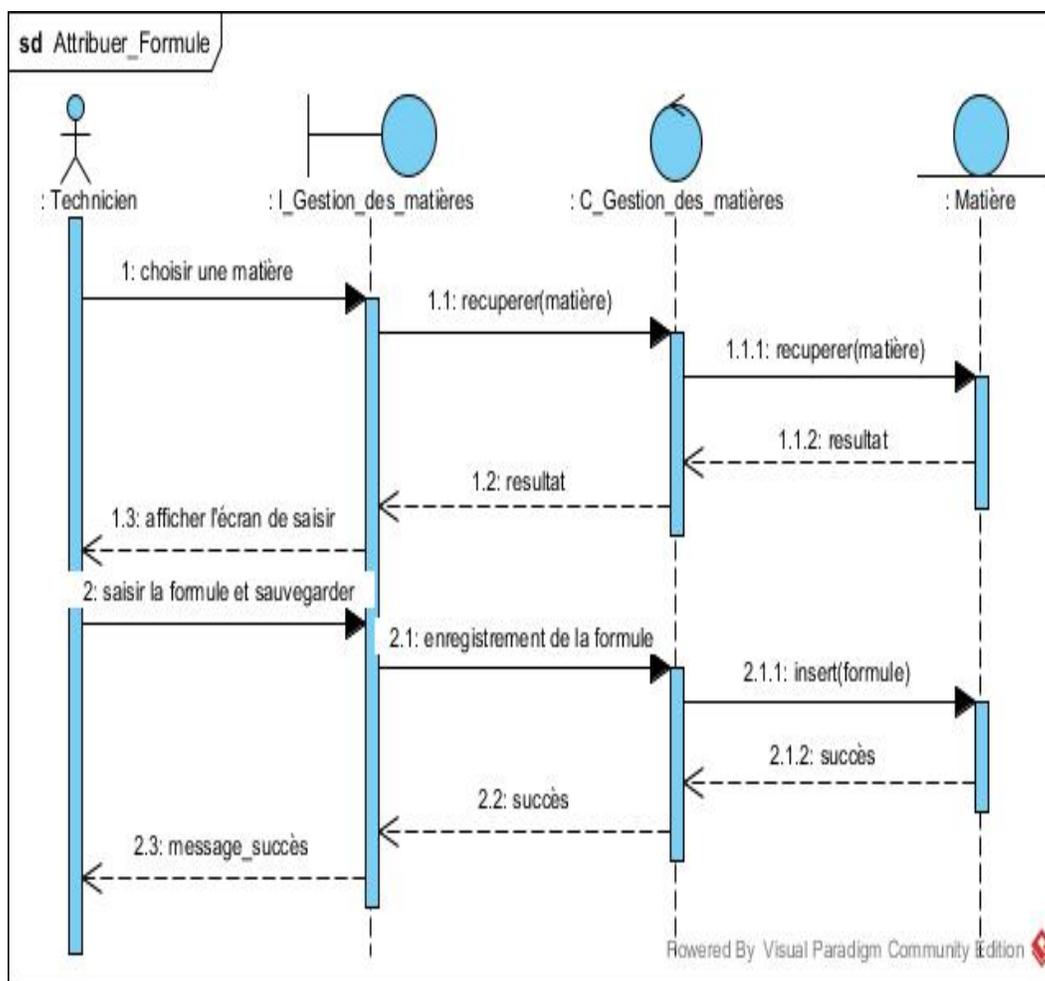


FIGURE C.1.4: Diagramme séquence d'interaction « attribuer formule »

C.1.2 Diagramme de séquence d'interaction « gestion des enseignants »

Les figures suivantes illustrent les diagrammes de séquence d'interaction pour le cas d'utilisation "gestion des enseignants"

C.1.2.1 Diagramme de séquence d'interaction de cas « ajouter enseignant »

La figure suivante représente le diagramme de séquence de cas d'utilisation « ajouter enseignant » :

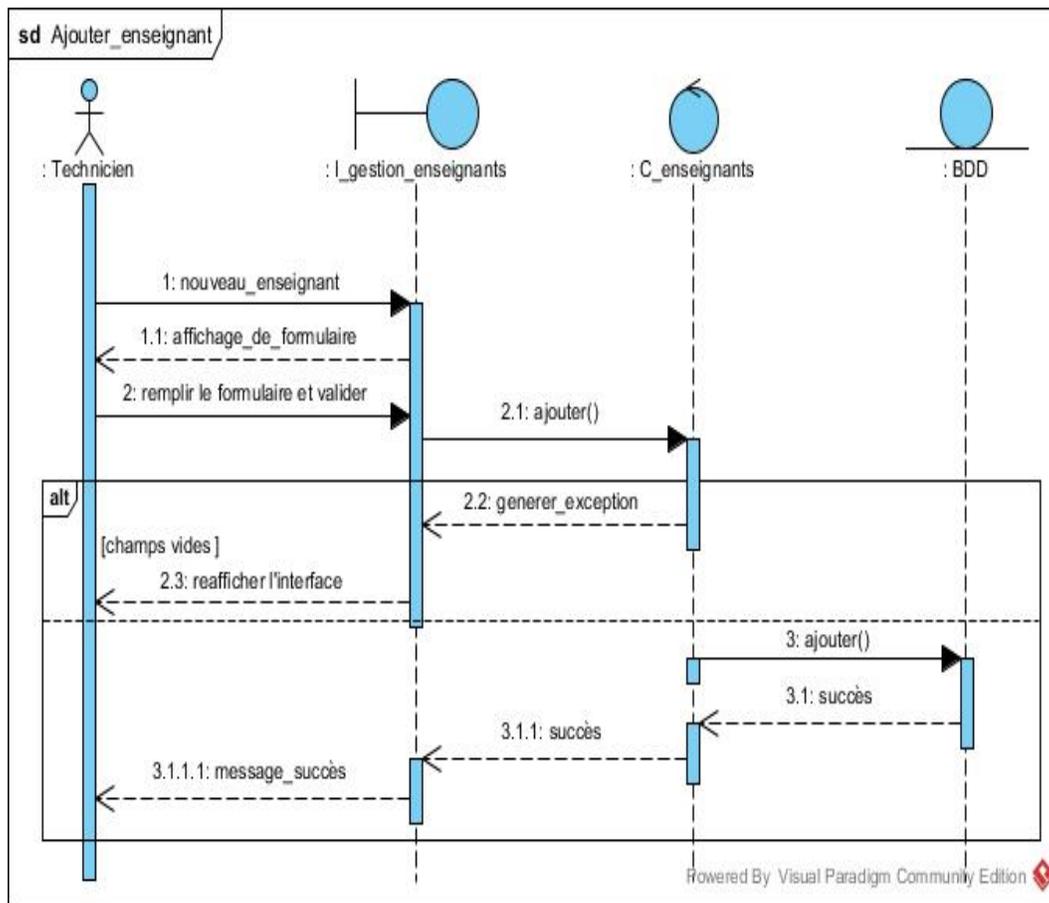


FIGURE C.1.5: Diagramme séquence d'interaction « ajouter un enseignant »

C.1.2.2 Diagramme de séquence de cas « modifier enseignant »

La figure suivante représente le diagramme de séquence de cas d'utilisation « modifier enseignant » :

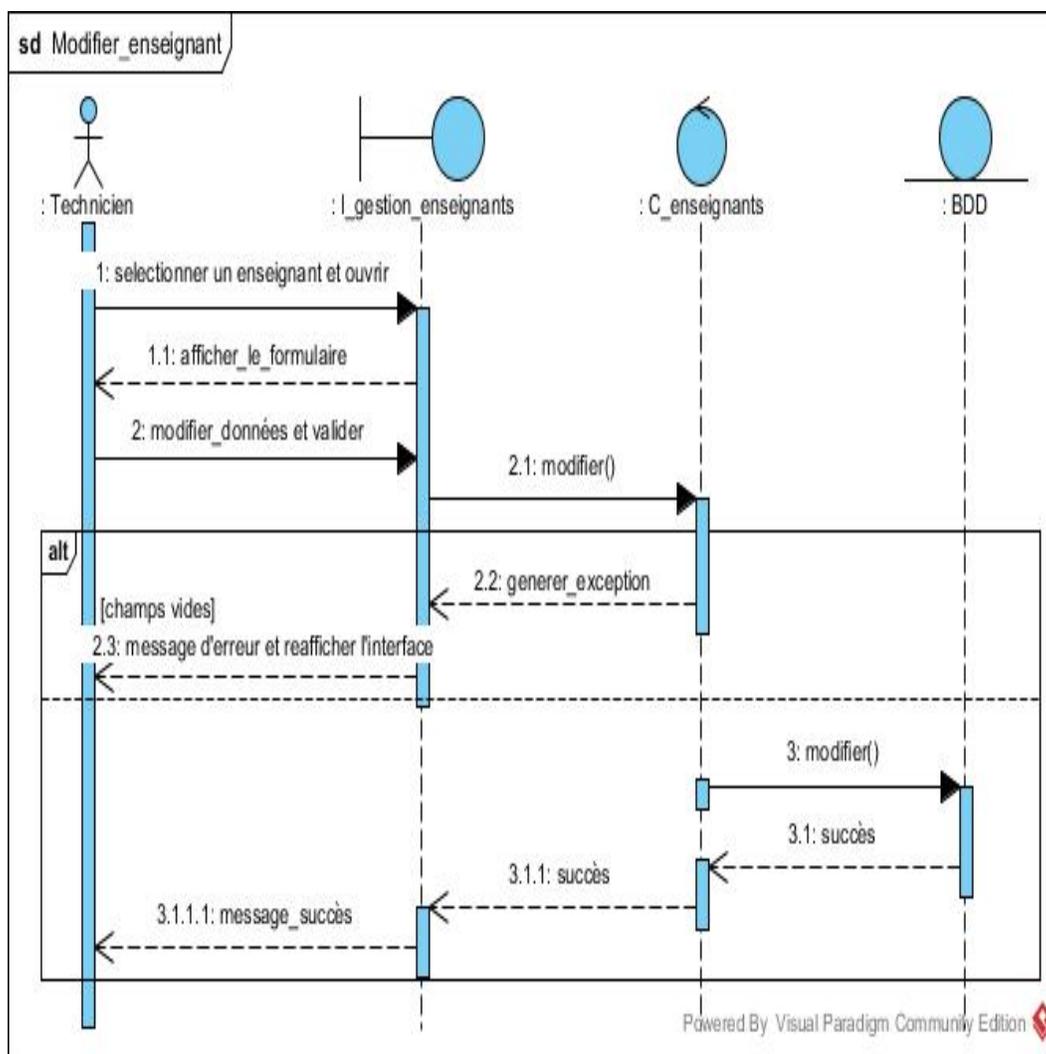


FIGURE C.1.6: Diagramme séquence d'interaction « modifier un enseignant »

C.1.2.3 Diagramme de séquence d'interaction de cas « supprimer enseignant »

La figure suivante représente le diagramme de séquence de cas d'utilisation « supprimer enseignant » :

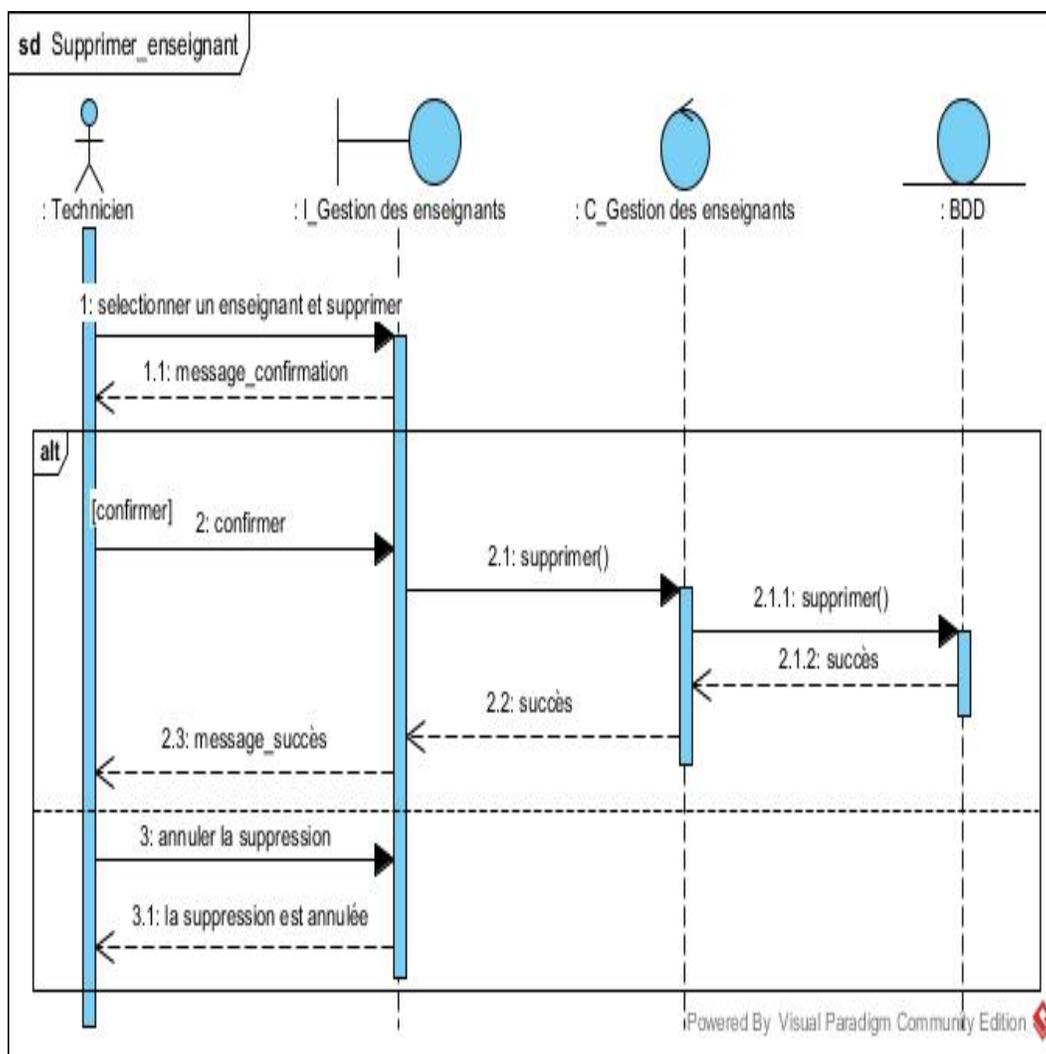


FIGURE C.1.7: Diagramme séquence d'interaction « supprimer un enseignant »

C.1.3 Diagramme de séquence d'interaction « gérer groupes »

La figure suivante illustre le diagramme de séquence d'interaction pour le cas d'utilisation « gérer groupes » :

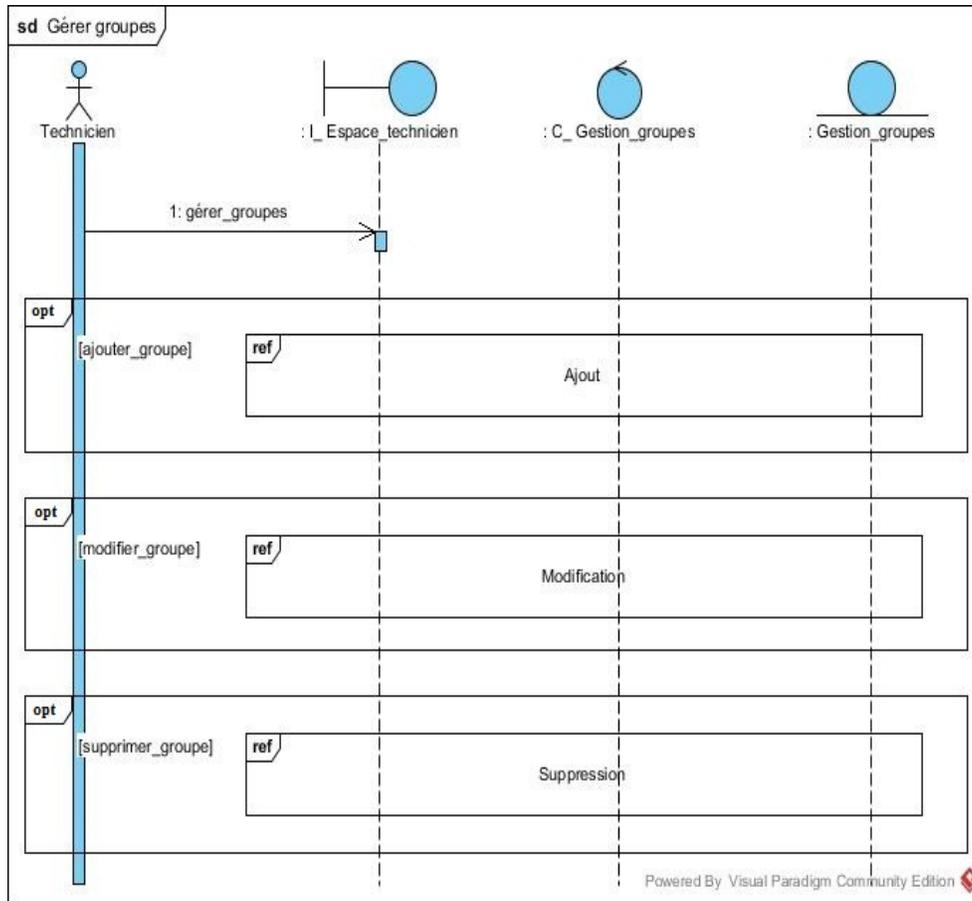


FIGURE C.1.8: Diagramme sequence d'interaction « gérer groupes »

C.1.4 Diagramme de séquence d'interaction « saisir les notes »

Les figures suivantes illustrent les diagrammes de séquence d'interaction pour le cas d'utilisation « saisir les notes » :

C.1.4.1 Diagramme de séquence d'interaction de cas : « importer les notes »

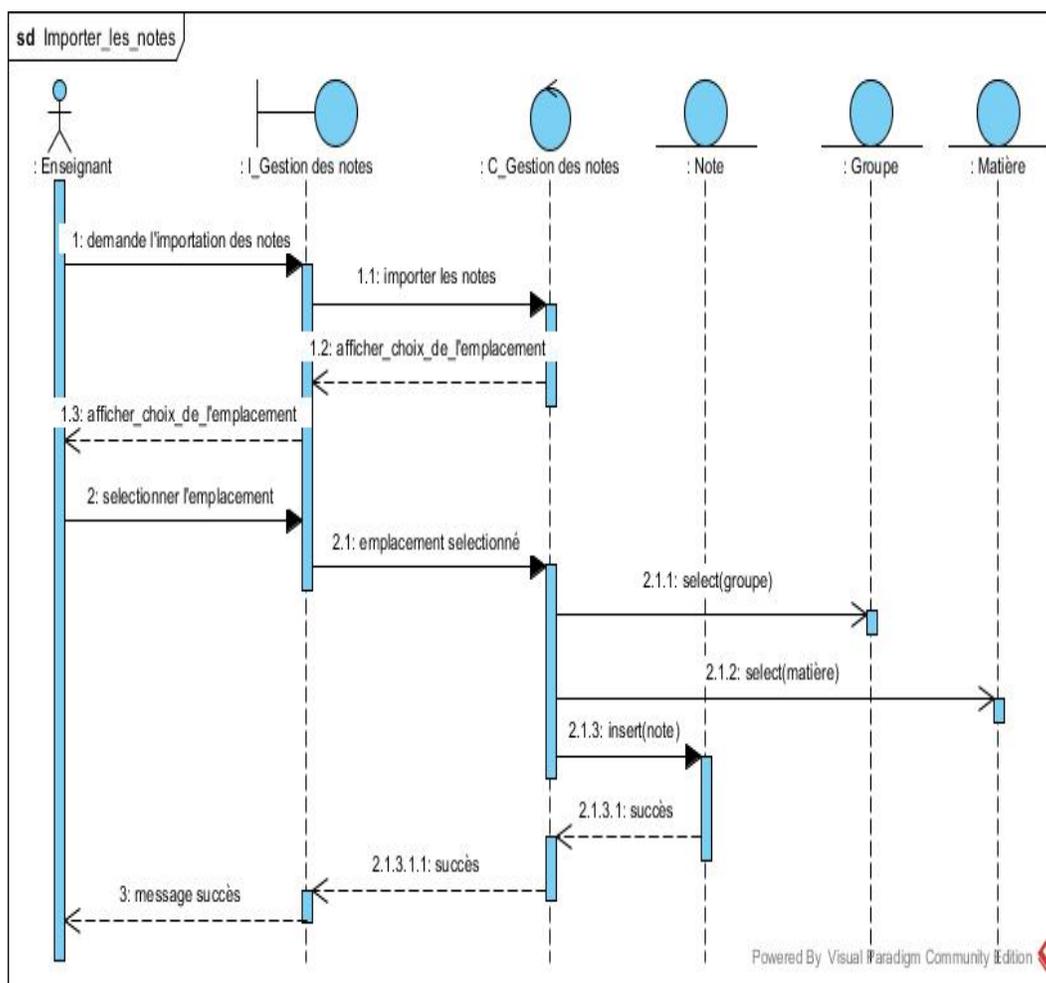


FIGURE C.1.9: Diagramme sequence d'interaction « importer les notes »

C.1.4.2 Diagramme de séquence d'interaction de cas : « saisir manuellement les notes »

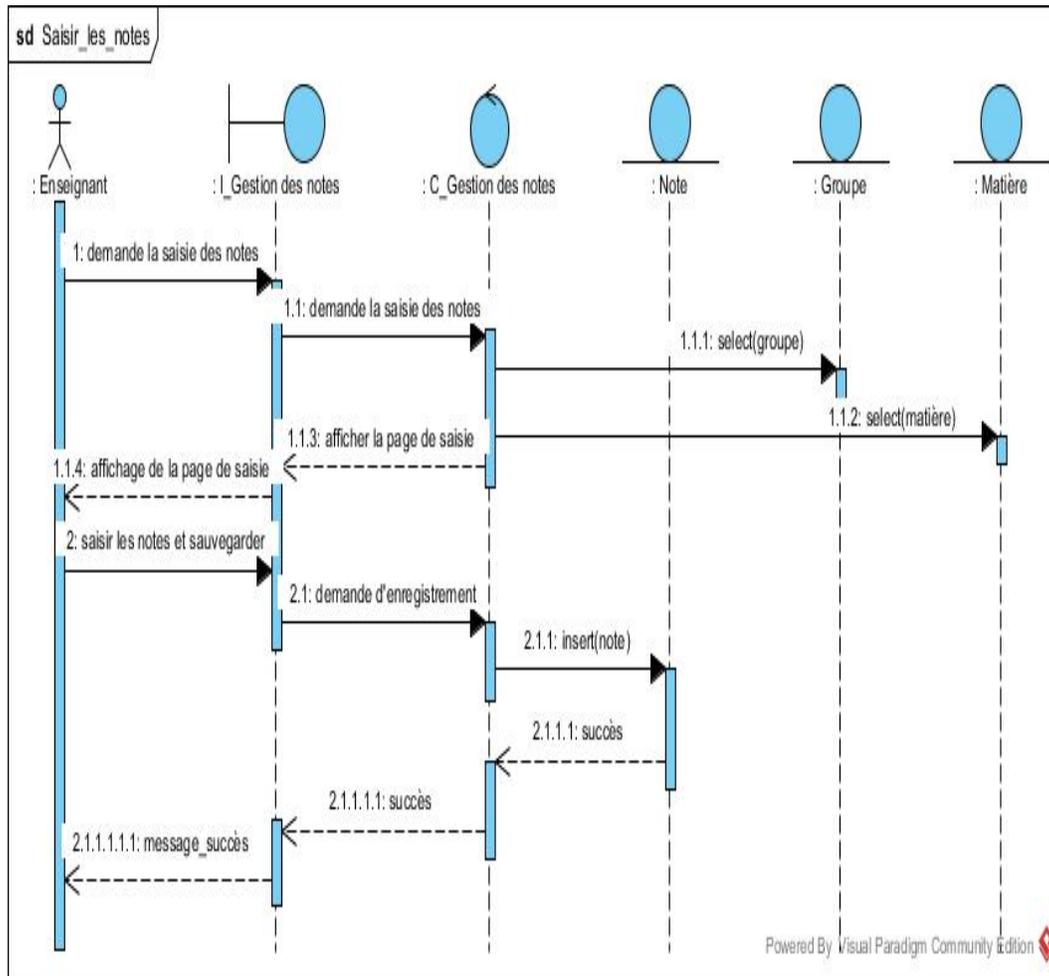


FIGURE C.1.10: Diagramme séquence d'interaction « saisir les notes manuellement »

C.1.5 Diagramme de séquence d'interaction « saisir l'état d'avancement »

»

La figure suivante illustre le diagramme de séquence d'interaction pour le cas d'utilisation « saisir l'état d'avancement » :

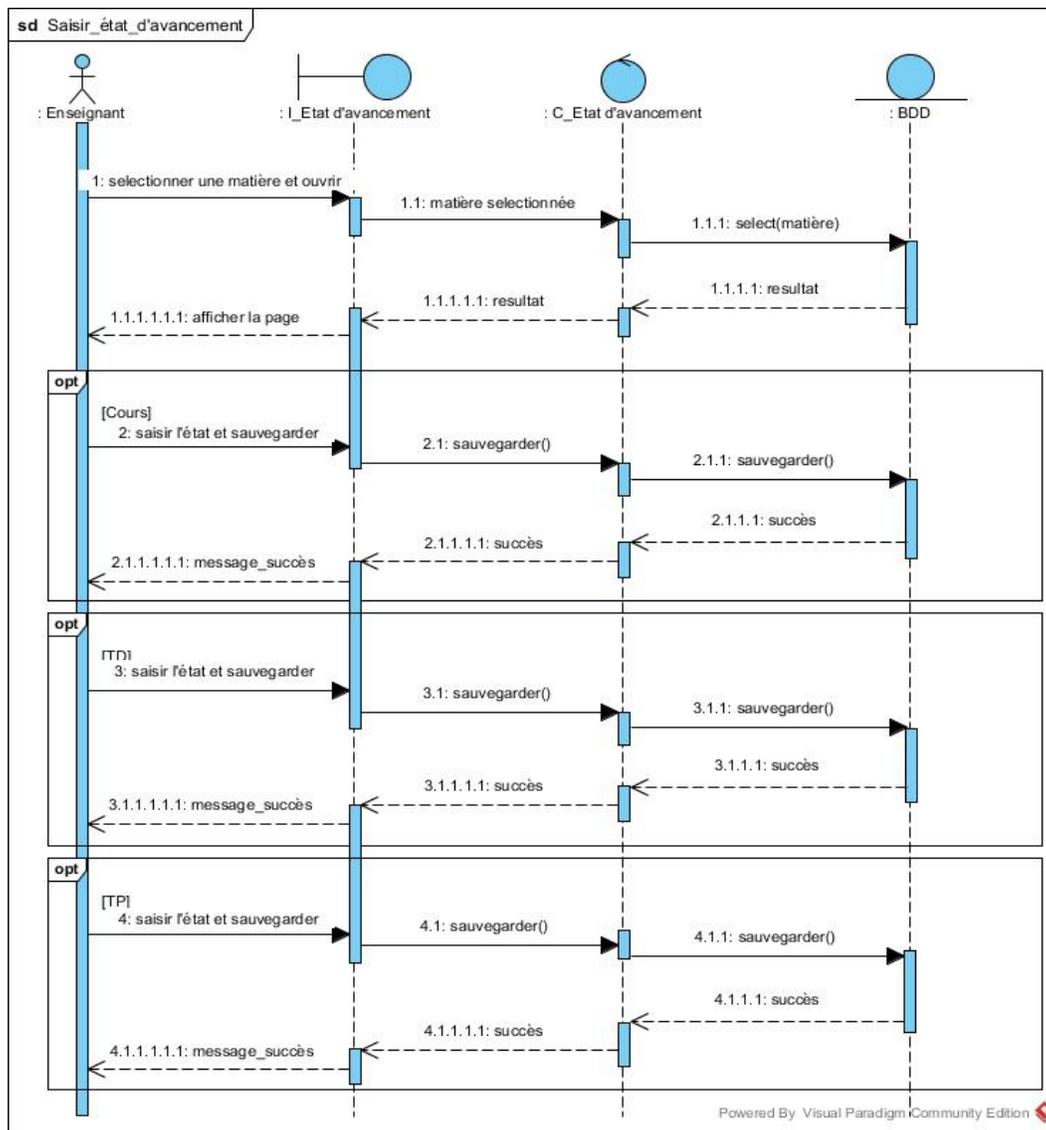


FIGURE C.1.11: Diagramme sequence d'interaction « saisir l'état d'avancement »

C.1.6 Diagramme de séquence d'interaction « consulter les notes »

La figure suivante illustre le diagramme de séquence d'interaction pour le cas d'utilisation « consulter les notes »

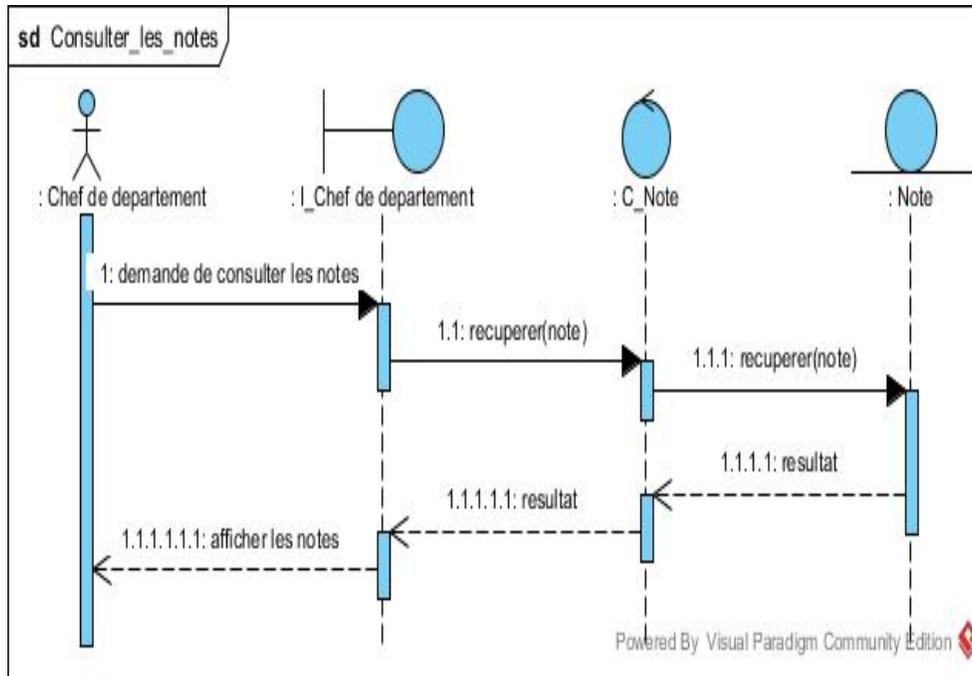


FIGURE C.1.12: Diagramme sequence d'interaction « consulter les notes »

Bibliographie

- [1] A.GONCALVES, Java EE 6 et GlassFish 3 , PEARSON, 2009.
- [2] C.Aubry, Scrum, DUNOD , 4e Edition, 2015.
- [3] G. Kurzawa, Agilité en pratique : SCRUM, Sorbonne université, Juin 2017.
- [4] A.Boucher, Ergonomie Web, EYROLLES , 2e Edition, 2009.
- [5] O.Mawloud, La modélisation avec UML, Université de Bejaia, 2017/2018.
- [6] F. Juliard, UML Unified Method Language, Journal Université de Bretagne Sud UFR SSI-IUP Vannes, 2001-2002.
- [7] Philippe RIGAUX. Cours de bases de données. juin 2001.

- [8] F.Vallée P.Roques. UML 2 en action de l'analyse des besoins a la conception. 2e Edition Eyrolles, 2007.
- [9] **site web** <https://www.visual-paradigm.com/features/> **consulter le** 30/08/2020.
- [10] **site web** http://www.univ-bejaia.dz/Fac_Technologie/presentation/ **consulter le** 02/03/2020.
- [11] **site web** <https://netbeans.org/kb/trails/java-ee.html>. **consulter le** 12/05/2020.
- [12] **site web** <https://www.agiliste.fr/exemple-dorganisation-projet-agile/> **consulter le** 05/06/2020.
- [13] **site web** [https://www.HTMLM5.com/tout_savoir.html/](https://www.HTMLM5.com/tout_savoir.html) **consulter le** 14/09/2020.
- [14] **site web** <https://www.w3schools.com/> **consulter le** 10/08/2020.

RÉSUMÉ

La gestion de la scolarité au niveau du département de Technologie de l'université Abderahmane MIRA de Bejaia rencontre de nombreux problèmes. Afin de pallier à ces problèmes, nous avons procédé à la conception et la réalisation d'une application web qui gère la scolarité du département de Technologie. Afin d'atteindre cet objectif, nous avons choisi de modéliser avec le formalisme UML et la méthode de gestion de projet SCRUM. Lors de la réalisation, nous avons choisi d'appliquer le design pattern MVC pour séparer les trois couches de l'application : La partie interface en utilisant HTML5, JQuery CSS et le framework bootstrap, la partie Contrôleur en utilisant les servlets, et enfin la partie modèle composée des objets métiers ainsi que l'accès aux données

mot clé : Gestion, ST, JEE, web, UML

ABSTRACT

The management of schooling at the level of the Technology Department of the Abderahmane MIRA University of Bejaia is encountering many problems. In order to overcome these problems, we have designed and built a web application that manages the education of the Technology department. In order to achieve this objective, we have chosen to model with the UML formalism and the SCRUM project management method. During the realization, we chose to apply the MVC design pattern to separate the three layers of the application : The interface part using HTML5, JQuery CSS and the bootstrap framework, the Controller homeland using the servlets, and finally the model part composed of business objects as well as data access

key words : Management, ST, JEE, web, UML