



**Université de Gafsa**

***Institut Supérieure des Sciences Appliquées et Technologie de  
GAFSA***

***Département : Informatique & Télécommunication***

***Conception et implémentation d'un système de gestion  
de scolarité d'un établissement d'enseignement  
supérieur***

Présenté et soutenu par :

**BEN AMMAR Safa**

En vue de l'obtention de

**Licence Appliquée en Ingénierie des Systèmes Informatiques**

Sous la direction de :

**Mr. TELLI Mounir**

**Mme. SLIMANI Nawal**

Soutenu le 05/06/2023

Devant le jury composé de :

**Président : BOUKHARI Kabil**

**Rapporteur : KOTTI Ali**

**Membres : TELLI Mounir**

**2022/2023**

# *Dédicaces*

*Tout d'abord merci à mon DIEU de m'avoir donné la force pour  
terminer ce travail.*

*Je dédie ce travail à tous ceux qui m'ont aidé à réaliser ce  
travail.*

*❖ A mon père*

*En témoignage de ma sincère gratitude pour ses sacrifices qu'il a  
consentis pour moi et pour toute la famille.*

*❖ A ma mère*

*J'espère qu'elle le trouve récompensant de tous les sacrifices  
qu'elle consenti pour moi*

*❖ A mes frères et sœurs*

*En témoignage de mon amitié et mon attachement*

*Ben Ammar Safa*

# Remerciement

*D'abord, je tiens à remercier les membres de ma famille particulièrement mes parents pour tout le soutien et toute l'aide qu'ils m'ont apportés durant mes études.*

*Mes remerciements les plus distingués sont aussi accordés à mon encadreur: Monsieur **Telli Mounir** qui m'est beaucoup aidé et guidé tout au long de mon projet.*

*Je tiens à remercier également les membres du jury qui ont accepté d'évaluer mon mémoire. Leurs observations et commentaires me furent très précieux.*

*Un grand merci aussi à tous mes enseignants qui m'ont offert l'occasion de pouvoir réaliser mes études dans les meilleures conditions au sien de l'ISSAT.*

*C'est avec grande plaisir que je présente un grand remerciement à **Mme Slimani Nawel** pour m'avoir accordé l'opportunité d'effectuer cette stage dans leur Ecole Supérieure d'Ingénieurs Privée de Gafsa.*

# SOMMAIRE

INTRODUCTION GÉNÉRALE.....	1
<b>Chapitre 1 :.....</b>	<b>2</b>
Etat de l’art.....	2
1.1.    Introduction.....	3
1.2.    Cadre du projet.....	3
1.2.1.    Présentation ESIP .....	3
1.2.1.1.    Services et Départements .....	3
1.2.1.2.    Formation .....	4
1.3.    Contexte .....	5
1.3.1.    Problématique .....	5
1.3.2.    Résultats .....	5
1.4.    Notions de Base .....	6
1.4.1.    Qu'est-ce qu'un système de scolarité .....	6
1.4.2.    Fonctionnalités du système de gestion de scolarité .....	6
1.4.2.1.    Gestion de l'information scolaire.....	6
1.4.2.2.    Gestion de campus.....	6
1.4.2.3.    Gestion des étudiants.....	6
1.4.2.4.    Gestion des examens .....	7
1.4.3.    L’objectif de système de scolarité.....	7
1.5.    Etude de l’existant.....	7
1.5.1.    Les déferents logiciels.....	7
1.5.1.1.    Alma .....	7
1.5.1.2.    Schoolbic .....	8
1.5.1.3.    Classter .....	8
1.5.2.    Critique de l’existant .....	9
1.6.    Spécification des besoin.....	9
1.6.1.    Les besoins fonctionnels .....	10
1.6.2.    Les besoins non fonctionnels .....	10
1.7.    Solution proposée.....	10
1.7.1.    Choix de l’environnement du développement .....	12
1.7.2.    Choix de l’architecture de développement.....	12

1.8.	Conclusion .....	14
<b>Chapitre 2 :</b>	.....	15
2.1.	Introduction.....	16
2.2.	Méthodologie de conception adoptée .....	16
2.2.1.	Définition UML .....	16
2.2.2.	Différents diagrammes .....	16
2.2.2.1.	Les diagrammes de cas d'utilisation .....	16
2.2.2.2.	Le diagramme de séquence .....	16
2.2.2.3.	Diagramme d'activité.....	17
2.2.2.4.	Diagramme de Classe .....	17
2.2.3.	Pourquoi utiliser UML .....	18
2.3.	Conception de notre application .....	18
2.3.1.	Les diagrammes de cas d'utilisation .....	18
2.3.1.1.	Identification des acteurs.....	18
2.3.1.2.	Diagramme de cas d'utilisation globale .....	19
2.3.1.3.	Diagramme de cas d'utilisation pour l'administrateur .....	20
2.3.2.	Les diagrammes de séquences et activités .....	21
2.3.2.1.	Diagramme de séquences de cas d'utilisation " s'authentifier " .....	21
2.3.2.2.	Diagramme de séquence de cas d'utilisation " Gérer Enseignant " .....	23
2.3.2.3.	Diagramme de séquence "gérer Résultat " .....	26
2.3.2.4.	Diagramme de cas d'utilisation coté Enseignant .....	29
2.3.3.	Le diagramme de classes.....	32
2.4.	Conclusion .....	33
<b>Chapitre 3 :</b>	.....	34
Réalisation.....	.....	34
3.1.	Introduction.....	35
3.2.	Environnement logiciel et matériel .....	35
3.2.1.	Environnement matériel .....	35
3.2.2.	Environnement logiciel .....	35
3.3.	Présentation des pages web de l'application : .....	37
3.3.1.	L'authentification .....	38
3.3.2.	Fenêtre « Ajouter Enseignant ».....	38
3.3.3.	Fenêtre « Liste des Enseignant ».....	39
3.3.4.	Fenêtre « Supprimer Enseignant» .....	40
3.1.1.	Page Gérer Résultat:.....	40
3.1.2.	Liste des absences .....	41

3.1.3. Fenêtre « gérer les Séances» .....	41
3.4. Conclusion .....	41
Conclusion & perspectives .....	42

# ***LISTE DES FIGURES***

Figure 1. 1: Logo d'école .....	3
Figure 1. 2: Logiciel Alma.....	8
Figure 1. 3: Système Schoolbic .....	8
Figure 1. 4: Application Classter .....	9
Figure 1. 5: L'architecture générale de notre application .....	11
Figure 1. 6:L'architecture MVC .....	13
Figure 2. 1: Acteurs de notre Application.....	19
Figure 2. 2: Diagramme de cas d'utilisation global.....	20
Figure 2. 3: Diagramme de cas d'utilisation pour l'administrateur.....	20
Figure 2. 4: Diagramme de cas d'utilisation "S'authentifier" .....	21
Figure 2. 5: Diagramme du séquence de cas d'utilisation " S'authentifier " .....	22
Figure 2. 6: Diagramme d'activité de la fonction " Connexion " .....	22
Figure 2. 7: Diagramme de séquence de cas d'utilisation " Enregistrer Enseignant " .....	23
Figure 2. 8: Diagramme d'activité "Enregistrer Enseignant " .....	24
Figure 2. 9: Diagramme de séquence de cas d'utilisation " Modifier Etudiants " .....	25
Figure 2. 10: Diagramme d'activité " Modifier Etudiants " .....	25
Figure 2. 11: diagramme de séquence de cas d'utilisation " Enregistrer Résultat" .....	26
Figure 2. 12: Diagramme de séquence de cas d'utilisation " Modifier Résultat " .....	27
Figure 2. 13: Diagramme de séquence de cas d'utilisation " Supprimer Résultat " .....	28
Figure 2. 14:Diagramme de cas d'utilisation Coté enseignant .....	29
Figure 2. 15: Diagramme de sequence de cas d'utilisation " Enregistrer Absence" .....	29
Figure 2. 16: diagramme d'activité « Enregistrer Absence » .....	30
Figure 2. 17: Diagramme de sequence de cas d'utilisation " Modifier Absence " .....	31
Figure 2. 18: Diagramme de sequence de cas d'utilisation " Supprimer Absence " .....	32
Figure 2. 19: diagramme de classes .....	33
Figure3. 1: Notepad++.....	36
Figure3. 2: Xampp .....	36
Figure3. 3: Logo MySQL .....	37
Figure3. 4: SigleStar UML .....	37
Figure3. 5: Fenetre d'authentification .....	38
Figure3. 6: Fenêtre « Ajouter Enseignant » .....	39
Figure3. 7: Liste des Enseignants .....	39
Figure3. 8: Supprimer Enseignant .....	40
Figure3. 9: Gestion Résultat .....	40
Figure3. 10: Fenêtre liste des absences coté Enseignant .....	41
Figure3. 11:Fenêtre « Gérer les Séance » .....	41

## ***LISTE DES TABLEAUX***

Tableau 2. 1: Description de cas d'utilisation " S'authentifier " .....	21
Tableau 2. 2: Description de cas d'utilisation " Enregistrer Enseignant" .....	23
Tableau 2. 3: Description de cas d'utilisation " Modifier Etudiant " .....	24
Tableau 2. 4: Description de cas d'utilisation " Enregistrer Résultat " .....	26
Tableau 2. 5: Description de cas d'utilisation " Modifier Résultat " .....	27
Tableau 2. 6: Description de cas d'utilisation " Supprimer Résultat" .....	28
Tableau 2. 7:Description de cas d'utilisation " Enregistrer Absence " .....	29
Tableau 2. 8: Description de cas d'utilisation " Modifier Absence " .....	30
Tableau 2. 10: Description de cas d'utilisation " Supprimer Absence" .....	31
 Tableau 3. 1: Environnement matériel.....	 35



## ***INTRODUCTION GÉNÉRALE***

Il ne fait plus aucun doute que la technologie de l'information représente la révolution la plus importante et la plus innovante qui ait caractérisé la vie humaine au cours du siècle dernier. En effet, loin d'être un phénomène de mode éphémère, ou une tendance passagère, l'informatique vient nous apporter de multiples commodités à notre mode de vie. Aucun domaine n'a été épargné par cette stratégie qui offre de nombreux services à la fois à l'entreprise ou à la direction et aux employés.

Mais, au-delà de l'utilisation individuelle de l'informatique, c'est surtout la mise en communication des ordinateurs, qui a permis de révolutionner les méthodes de travail. Ainsi, on a assisté à l'émergence des réseaux. Ce nouveau progrès offre aux utilisateurs de nouveaux outils de travail et leur permet d'améliorer leur rentabilité et leur productivité.

La révolution des nouvelles technologies s'impose tant dans la vie domestique que professionnelle ; elle touche pratiquement tous les secteurs de la société, de l'économie,...etc. Cette évolution a aussi touchée l'univers des études et en plus particulier le monde de l'université.

Dans cette vision, nous inscrivons notre présent travail pour n'aborder que le problème lié à la Conception et implémentation d'un système de gestion de scolarité d'un établissement d'enseignement supérieur afin résoudre certaines difficultés y afférentes que rencontre l'école supérieure d'ingénieurs privée de Gafsa (ESIP) dans sa Gestion quotidienne.

Dans ce rapport sera devisé en trois chapitres respectivement :

- ❖ Le premier chapitre intitulé "Etat de l'art" présentera étude de l'existant, en suit la spécification des besoins.
- ❖ Le deuxième chapitre "Etude conceptuelle" va donner une vision plus détaillée, architecture, structure de la base de données.
- ❖ Enfin et au niveau du troisième chapitre "Réalisation " il s'agit d'aboutir à un produit final, environnement de travail, choix technique, phase d'implémentation et tests de validation

# ***Chapitre 1 :***

## ***Etat de l'art***

## **1.1. Introduction**

Dans ce chapitre, nous présenterons premièrement l'organisme d'accueil et que ce qu'un système de gestion de scolarité d'un établissement d'enseignement supérieur, son processus et ses objectifs. Dans la deuxième partie nous proposerons notre projet.

## **1.2. Cadre du projet**

### **1.2.1. Présentation ESIP**

**L'ESIP** est une **Ecole Supérieure Privée d'Ingénieurs de Gafsa** créée à l'initiative de nos investisseurs et partenaires avec comme objectif majeur de former des ingénieurs de terrain de très haut niveau scientifique et managérial au service de la société et de l'économie.

**L'ESIP a été agréée par l'état en octobre 2013 (Agrément n° 2013-05).**

Nos partenaires industriels sont fortement impliqués dans le fonctionnement et la gestion de l'école (Conseil d'administration, Conseil de Perfectionnement, Conseil scientifique, etc.).

**L'ESIP** a édifié son projet pédagogique en forte corrélation avec le monde de l'entreprise. Notre école d'ingénieurs vise la parfaite adéquation de sa formation aux besoins du marché et garantit à ses étudiants un taux d'employabilité exceptionnel.



**FIGURE 1. 1: LOGO D'ECOLE**

#### **1.2.1.1. Services et Départements**

On peut citer deux ressources d'informations :

- ✓ **ESIP** possède un staff pédagogique de haut niveau composé d'enseignants chercheurs et d'intervenants experts du monde industriel qui assurent à la

fois une solide base théorique et une riche expérience pratique au jeune ingénieur lui permettant de maîtriser rapidement son futur métier.

- ✓ **ESIP** met à votre disposition des formations certifiantes sur des applications et des logiciels et technologie de pointe dans les domaines de l'informatique, du contrôle, de la certification et de la sécurité des données (comme : SLQ expert, Linux (système d'exploitation), JAVA, JAVA-JEE, CAO-DAO, Certification en sécurité, CISCO, PHP et autre...).

### **1.2.1.2. Formation**

#### **✓ Cycle Préparatoire**

La Tunisie a des besoins croissants en ingénieurs pour soutenir son développement industriel : avec l'ouverture internationale des marchés, seules les entreprises compétitives, c'est à dire celles qui maîtrisent la technologie grâce à leurs ingénieurs, pourront soutenir la concurrence des pays industrialisés. C'est pourquoi le métier d'ingénieur est plein d'avenir et que la carrière d'ingénieur est une carrière diverse et passionnante, où les jeunes scientifiques peuvent donner la grande mesure de leurs talents.

Le premier but de l'**Ecole Supérieure d'Ingénieurs Privée de Gafsa** est de participer à former les futurs ingénieurs de qualité, ayant la maîtrise de la théorie et de la pratique. En effet, à tous les niveaux de formation, l'objectif est d'assurer une maîtrise de la science actuelle et de concevoir les enseignements de telle sorte que les futurs cadres aient la capacité de s'adapter aux changements mais aussi d'en être les initiateurs.

#### **✓ Cycle d'Ingénieurs**

Cette formation d'ingénieur est ouverte aux étudiants ayant réussi le cycle préparatoire aux études d'ingénieurs, ou les titulaires d'une licence dans une spécialité scientifique ou technique. Les étudiants titulaires d'un master 1 dans une branche scientifique ou technique accèdent directement à la 2ème année de ce cycle après l'étude de son dossier.

La formation d'ingénieur s'étale sur 3 années organisées en 6 semestres. La formation d'ingénieur se distingue par sa capacité à évoluer avec le développement technologique et à répondre aux besoins du monde de travail en termes de cadres professionnellement qualifiés et directement opérationnels.

## **1.3. Contexte**

La présentation du contexte de l'étude consiste à énoncer la problématique et à spécifier les résultats attendus.

### **1.3.1. Problématique**

La problématique de ce travail réside dans la manipulation des activités réalisées par Agent de scolarité ; elles sont liées aux problèmes tels que : la perte de données, difficultés d'établir des rapports, le contrôle ne pas assurer, manque de système informatique et manque de communication entre l'enseignant et les étudiants.

D'où il s'avère indispensable de formuler notre problématique en différentes questions suivantes permettant de résoudre ce problème:

- Est-ce que l'outil Informatique est-il indispensable à l'ESIP pour résoudre le problème de la gestion de scolarité ?

Pour trouver une solution à ces difficultés, il convient de mener une analyse pour aboutir à la mise en place d'un système informatique de gestion de scolarité d'où l'intérêt notre thème : «**Conception et implémentation d'un système de gestion de scolarité d'un établissement d'enseignement supérieur**».

### **1.3.2. Résultats**

Après analyse des problèmes énoncés précédemment, une application de base de données s'avère nécessaire. Le système à mettre en place devra résoudre les problèmes rencontrés dans la gestion de la scolarité tout en prenant en compte les besoins des utilisateurs (Agent de scolarité) et les perspectives d'évolution. Pour ce faire, notre travail consiste à réaliser un système dont les fonctionnalités s'articulent autour des points suivants :

- ✓ L'enregistrement d'un Etudiant
- ✓ Enregistrement des séances de cours
- ✓ L'accès aux informations (Enseignant, Etudiant et Résultat)
- ✓ L'archivage, la sécurité et la confidentialité des données.

## **1.4. Notions de Base**

### **1.4.1. Qu'est-ce qu'un système de scolarité**

Le Système de gestion scolaire est un outil précieux qui permet de rationaliser l'administration des écoles, des collèges et d'autres établissements d'enseignement. Il propose différents modules pour aider les membres du personnel et les enseignants à conserver les antécédents scolaires et les dossiers essentiels des étudiants.

De plus, il unifie les différents départements et fonctions de l'école que les administrateurs, comptables, les registraires et les gestionnaires de bureau peuvent accéder à tout moment et de n'importe où.

Le logiciel de gestion scolaire permet également d'automatiser les admissions, la planification des cours, la présence, etc. Il s'intègre à vos outils existants tels qu'un outil de facturation, un outil de carnet de notes et un logiciel de gestion de classe pour offrir une automatisation puissante.

### **1.4.2. fonctionnalités du système de gestion de scolarité**

Certains des modules et fonctionnalités standard des logiciels de gestion scolaire incluent :

#### **1.4.2.1. Gestion de l'information scolaire**

Il aide les écoles à gérer les admissions, la division des classes, les inscriptions des étudiants, la génération et l'attribution des cartes d'admission aux examens, etc., en calculant le nombre de facultés, d'étudiants et de salles de classe.

#### **1.4.2.2. Gestion de campus**

Ce module permet d'améliorer workflows en automatisant les tâches répétitives. Il vous permet également de gérer l'administration de l'école, de générer des laissez-passer, de gérer la sécurité et les visiteurs et de planifier des rendez-vous.

#### **1.4.2.3. Gestion des étudiants**

Il fournit des informations complètes sur les étudiants de l'admission à leur statut actuel. Cela peut inclure leurs profils, photographies, données de tuteur et de famille, rapports, etc.

#### **1.4.2.4. Gestion des examens**

Il calcule l'admissibilité à l'examen, crée des questions, attribue des salles de classe, évalue les scores et génère des rapports.

#### **1.4.3. L'objectif de système de scolarité**

La gestion d'une école peut sembler facile de l'extérieur, mais elle ne l'est pas de l'intérieur. C'est un domaine sensible qui nécessite une administration prudente du flux de travail et de la sécurité de l'école, en trouvant un rythme correct entre les étudiants, le personnel et les professeurs tout en garantissant un enseignement de qualité.

Bien que le système traditionnel permette de gérer différents aspects de la gestion scolaire, le monde moderne a besoin de plus de fonctionnalités. C'est crucial, en particulier dans les événements comme covid-19 qui ont fermé des écoles et d'autres établissements d'enseignement dans le monde.

Un logiciel de gestion scolaire peut les aider à mener correctement les activités éducatives, peu importe où se trouvent les élèves, le personnel ou les enseignants.

### **1.5. Etude de l'existant**

#### **1.5.1. Les différents logiciels**

##### **1.5.1.1. Alma**

Alma est un logiciel qui peut aider les enseignants à dialoguer avec les parents et les élèves, et leur offrir un système flexible qui élimine les doubles saisies et les erreurs.

La solution de gestion scolaire basée sur le cloud peut fonctionner pour des entreprises de toutes tailles dans le secteur de l'enseignement. Parmi ses principales fonctionnalités, elle aide les utilisateurs à gérer l'administration des écoles, à générer des rapports personnalisés et à gérer les inscriptions.

Les utilisateurs peuvent suivre les itinéraires des bus, diffuser des notifications d'urgence, et bien plus encore. Plus important encore pour les enseignants, il est possible de suivre les présences et de personnaliser les carnets de notes.

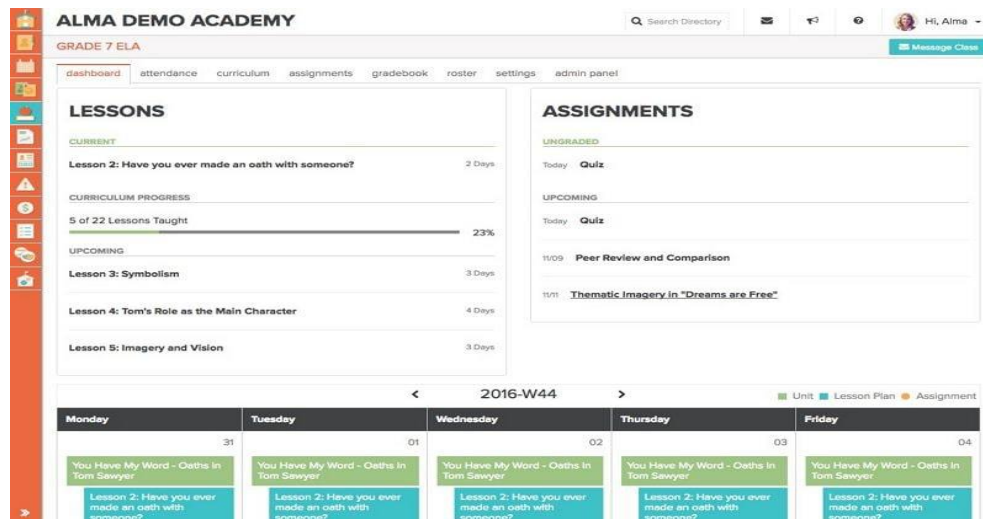


FIGURE 1. 2: LOGICIEL ALMA

### 1.5.1.2. Schoolbic

Cette application innovante a été conçue pour la collaboration dans les écoles et les établissements d'enseignement. Vous pouvez l'utiliser pour tout automatiser. Elle est idéale pour susciter l'engagement des élèves, des parents, des employés et des responsables d'écoles, ainsi que pour planifier et partager les programmes et les leçons. Ce logiciel de gestion scolaire offre une visibilité de qualité à 360° pour tous les types d'administration dans le secteur de l'éducation.

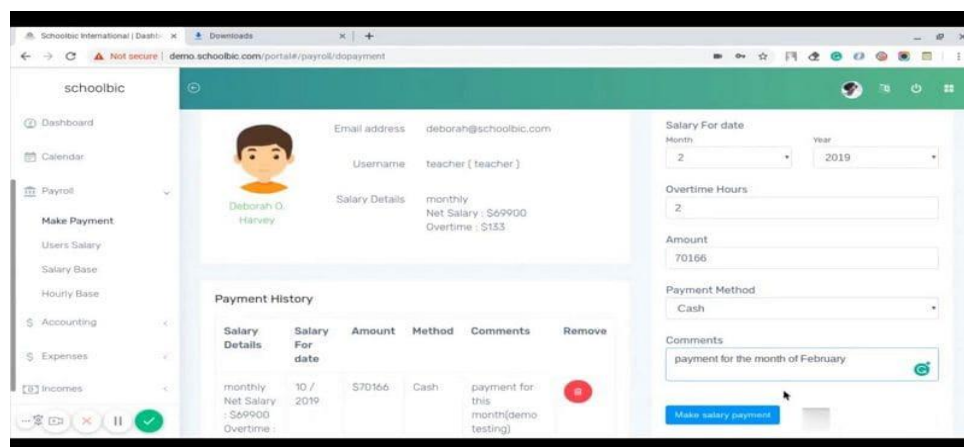


FIGURE 1. 3: SYSTEME SCHOOLBIC

### 1.5.1.3. Classter

Classter est une suite d'applications de gestion scolaire qui permet aux établissements de gérer les documents, les évaluations, les inscriptions, les réclamations et



les notifications. Parmi les principales fonctionnalités, citons la facturation, la réservation, le développement de programmes et la gestion des admissions.

Grâce à son module d'inscription en ligne, Classter permet aux étudiants et aux parents de s'inscrire automatiquement aux cours. Le logiciel permet également l'intégration de solutions tierces, notamment WordPress, WebEx, G Suite et Moodle.

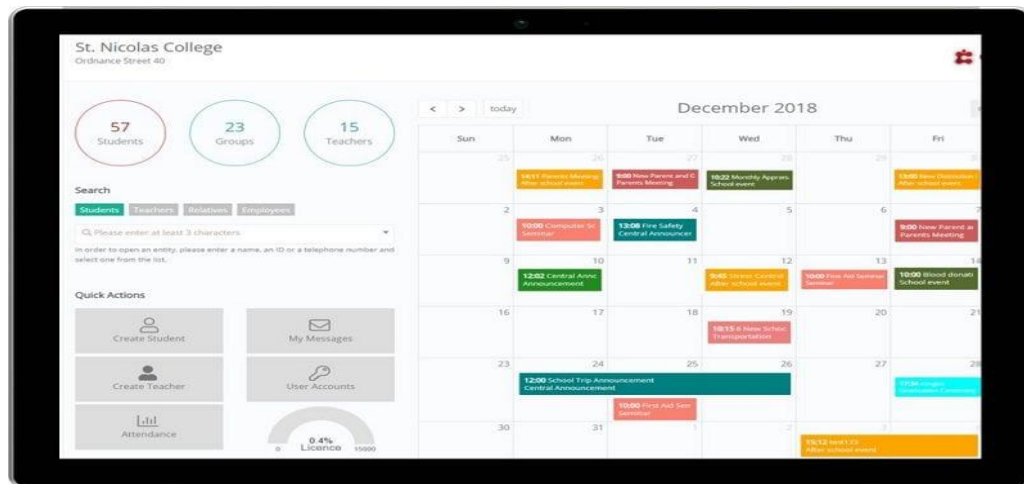


FIGURE 1. 4: APPLICATION CLASSTER

### 1.5.2. Critique de l'existant

La gestion du stock est un enjeu prioritaire au sein de l'administration. L'agent du bureau d'ordre en effet amenés à traiter au quotidien des volumes importants de matériel informatique, qui doivent être triés, consultés, imprimés, archivés, etc. Toute une organisation qui entraîne bien souvent :

- ✓ Des pertes ou des oublis provoquant des retards,
- ✓ Des erreurs de classement des documents,
- ✓ Perte du temps due aux opérations manuelles de traitement, de tri et de distribution du stock,
- ✓ Difficultés de recherche du matériel,
- ✓ Risque de perte de la traçabilité du matériel...

## 1.6. Spécification des besoins

L'étude de l'existant a permis de mettre en évidence les points positifs et les points de dysfonctionnement du système étudié. Il s'agira dans les paragraphes suivants de livrer les spécifications des besoins afin de répertorier les contraintes à prendre en compte dans la conception de la solution.

### **1.6.1. Les besoins fonctionnels**

Les besoins fonctionnels expriment une action que doit effectuer le système en réponse à une demande (sorties qui sont produite pour un ensemble donné d'entrées).

Avant d'imposer une solution, il faut se tourner vers le demandeur, pour aboutir de manière structurée à la solution. En effet, le but du projet est de satisfaire le besoin. Il faut exprimer clairement les objectifs à atteindre du projet, afin d'éviter toute confusion entre nous et le demandeur.

Le système proposé doit permettre de :

- ✓ Gérer séances,
- ✓ Consulter les Résultat,
- ✓ Gérer les enseignants,
- ✓ Rechercher (enseignant, résultat, séances et Etudiant)

### **1.6.2. Les besoins non fonctionnels**

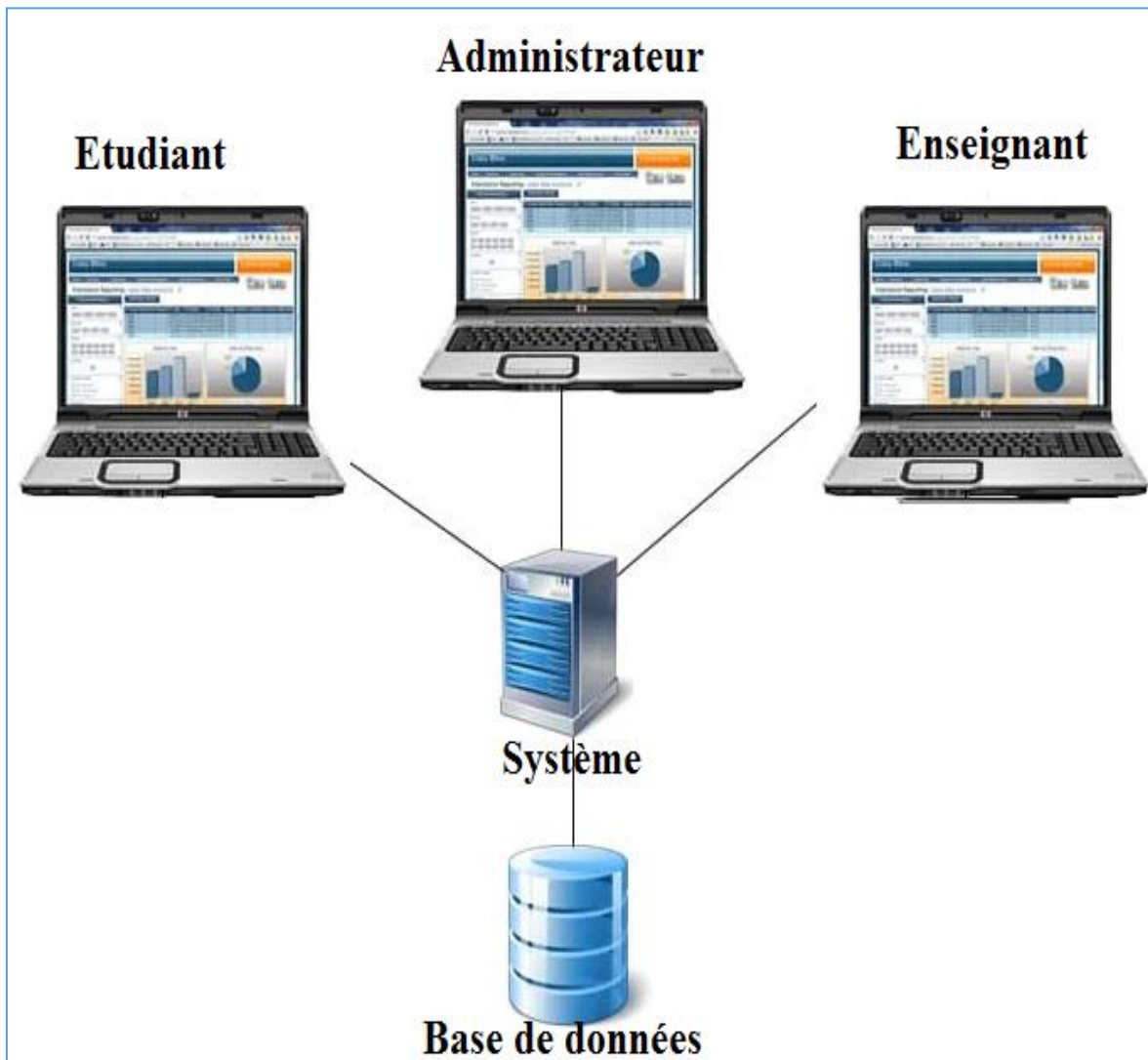
Les besoins non fonctionnels sont importants car ils agissent de façon indirecte sur le résultat et sur le rendement de l'utilisateur. Notre application doit répondre à ces besoins qui sont nécessaires pour atteindre la perfection et la bonne qualité du logiciel.

- ✓ **Fiabilité** : l'application doit fonctionner de façon cohérente sans erreurs,
- ✓ **Efficacité** : l'application doit permettre l'accomplissement des tâches avec le minimum de manipulations,
- ✓ **Sécurité** : l'application doit être sécurisée au niveau des données : authentification et contrôle d'accès,
- ✓ **Performance** : l'application doit être performante c'est-à-dire qu'elle doit répondre à travers ses fonctionnalités à toutes les exigences des utilisateurs d'une manière optimale.

## **1.7. Solution proposée**

Pour faire face aux problèmes qui existent dans les processus de gérer scolarité d'un établissement d'enseignement supérieur pour l'école supérieur d'ingénieur privé de Gafsa, nous sommes besoin d'un système qui automatise toutes les taches d'un bon système de gestion. A ce niveau nous proposons une solution améliorée et fiable qui prend en compte tous les exigences actuelles pour gérer les données et qui permettra de répondre aux futures besoins des utilisateurs.

La figure suivante représente l'architecture générale de notre application :



**FIGURE 1. 5: L'ARCHITECTURE GENERALE DE NOTRE APPLICATION**

Notre application est composée de :

- Un site ouvert et évolutif qui répond aux besoins des utilisateurs car il est créé par les dernières technologies de du développement web.
- Un système qui gère un large volume des données.
- Un système qui Facilite l'accès à l'information (les absences, résultat).
- Système plus sécurisé (authentification)

### **1.7.1. Choix de l'environnement du développement**

Pour la réalisation de notre application, nous allons définir un environnement de développement qui facilite à l'utilisateur l'accès aux différentes interfaces pour consulter, mettre à jour ou bien sauvegarder des informations. Pour cela, nous précisons :

- Le système d'exploitation : Windows.
- Le système de gestion de base des données : MySQL.
- Les outils de développement y compris les matériels, les langages et les logiciels : PHP, HTML, JavaScript, PHP, Sublim text.

### **1.7.2. Choix de l'architecture de développement**

Il existe plusieurs types d'architectures client/serveur, les plus répandus sont : Le client/serveur à 2 niveaux, le client/serveur à 3 niveaux et le MVC. Le choix de l'architecture dépend de notre application web qui est généralement conforme au modèle d'architecture MVC (modèle-vue-contrôleur). Le MVC est un motif de conception (design pattern) qui propose une solution générale au problème de la structuration d'une application. Le MVC définit des règles qui déterminent dans quelle couche de l'architecture, et dans quelle classe (orientée-objet) de cette couche, doit être intégrée une fonctionnalité spécifique. Une application conforme à ces règles est plus facile à comprendre, à gérer et à modifier. Ces règles sont issues d'un processus d'expérimentation et de mise au point de bonnes pratiques qui a abouti à une architecture standard.

L'objectif global du MVC est de séparer les aspects traitement, données et présentation, et de définir les interactions entre ces trois aspects. En simplifiant, les données sont gérées par le modèle, la présentation par la vue, les traitements par des actions et l'ensemble est coordonné par les contrôleurs. La figure Architecture MVC donne un aperçu de l'architecture obtenue, en nous plaçant d'emblée dans le cadre spécifique d'une application Web.

Cette architecture est décrite par ce schéma :

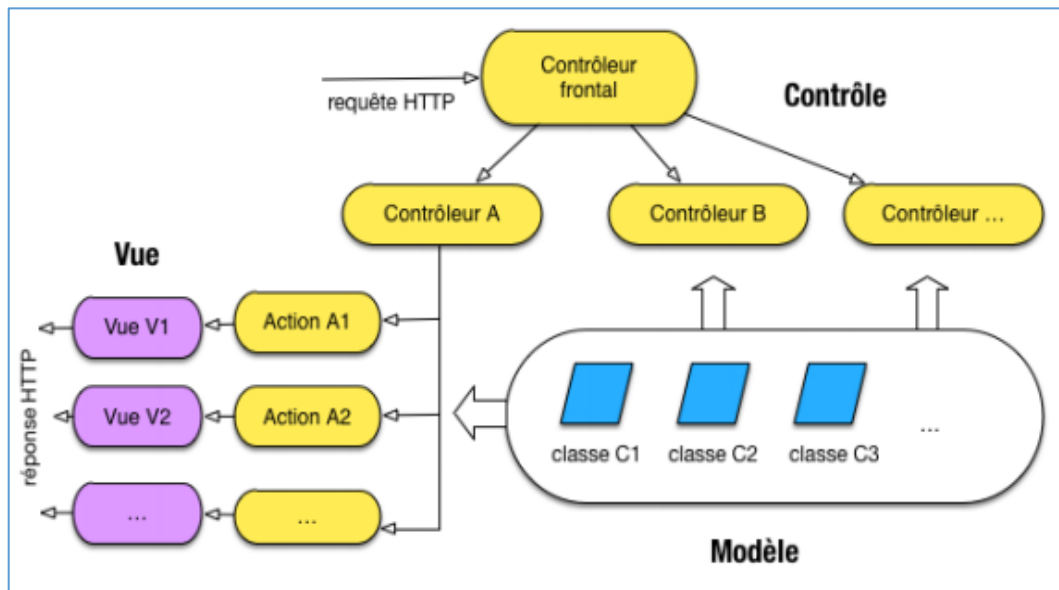


FIGURE 1. 6:L'ARCHITECTURE MVC

La figure montre une application constituée de plusieurs contrôleurs, chaque contrôleur étant lui-même constitué d'un ensemble d'actions. La première caractéristique de cette organisation est donc de structurer hiérarchiquement une application. Dans les cas simples, un seul contrôleur suffit, contenant l'ensemble des actions qui constituent l'application Web. Chaque requête HTTP est analysée par le framework qui détermine alors quel sont le contrôleur et l'action concernés. Il existe un contrôleur frontal (intégré au framework et donc transparent pour le programmeur), chargé de recevoir les requêtes HTTP, qui exécute l'action en lui passant les paramètres HTTP. Il se base notamment sur les mappings qui associent des URLs à des servlets. [5]

### Le modèle

Le modèle implante les fonctionnalités de l'application, indépendamment des aspects interactifs. Le modèle est également responsable de la préservation de l'état d'une application entre deux requêtes HTTP, ainsi que des fonctionnalités qui s'appliquent à cet état. Toute donnée persistante doit être gérée par la couche modèle, mais des objets métiers non persistant implantant une fonctionnalité particulière (un calcul, un service) sont également dans cette couche. Le modèle gère les données de session (le panier dans un site de commerce électronique par exemple) ou les informations contenues dans la base de données (le catalogue des produits en vente, pour rester dans le même exemple). Cela

comprend également les règles, contraintes et traitements qui s'appliquent à ces données, souvent désignées collectivement par l'expression logique métier de l'application.

### **La vue**

La vue est responsable de l'interface, ce qui recouvre essentiellement dans notre cas les fragments HTML qui sont assemblés pour constituer les pages du site. La vue est également responsable de la mise en forme des données (pour formater une date par exemple) et doit d'ailleurs se limiter à cette tâche. Il faut prendre garde à éviter d'y introduire des traitements complexes qui relèvent du logique métier, car ces traitements ne seraient alors pas réutilisables dans un autre contexte. En principe la vue ne devrait pas accéder au modèle et obtenir ses données uniquement de l'action (mais il s'agit d'une variante possible du MVC). [5]

### **Contrôleurs et actions**

Le rôle des contrôleurs est de récupérer les données utilisateurs, de les filtrer et de les contrôler, de déclencher le traitement approprié (via le modèle), et finalement de déléguer la production du document de sortie à la vue. Comme nous l'avons indiqué précédemment, l'utilisation de contrôleurs a également pour effet de donner une structure hiérarchique à l'application, ce qui facilite la compréhension du code et l'accès rapide aux parties à modifier. Indirectement, la structuration logique d'une application MVC en contrôleurs et actions induit donc une organisation normalisée. Les contrôleurs, comme leur nom l'indique, ont pour rôle essentiel de coordonner les séquences d'actions/réactions d'une application web. Ils reçoivent des requêtes, déclenchent une logique à base d'objets métiers, et choisissent la vue qui constitue la réponse à la requête. [5]

## **1.8. Conclusion**

L'analyse des besoins et des spécifications est une phase extrêmement importante pour le cycle de vie d'un projet. Au cours de ce chapitre nous avons discuté les exigences et les problèmes à résoudre de point de vue des besoins de l'utilisateur. Cette phase nous a permis de préparer la conception de l'application qui est notre but dans le chapitre suivant.

## ***Chapitre 2 :***

# ***Etude conceptuelle***

## **2.1. Introduction**

Dans ce chapitre, nous présenterons la technique élaborée pour réaliser notre système d'information. Nous montrerons les diagrammes de cas d'utilisation générale de notre projet ainsi que les diagrammes des séquences et le diagramme de classe.

## **2.2. Méthodologie de conception adoptée**

### **2.2.1. Définition UML**

UML (en anglais Unified Modeling Language ou langage de modélisation unifié) est un langage de modélisation graphique à base de pictogrammes. Il est apparu dans le monde du génie logiciel, dans le cadre de la "conception orientée objet". Couramment utilisé dans les projets logiciels, il peut être appliqué à toutes sortes de systèmes ne se limitant pas au domaine informatique.

### **2.2.2. Différents diagrammes**

UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux vues. Une vue est constituée d'un ou plusieurs diagrammes. On distingue deux types de vues :

#### **2.2.2.1. Les diagrammes de cas d'utilisation**

Ces diagrammes sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs, ils interagissent avec les cas d'utilisation. Les cas d'utilisation permettent de traduire les spécifications fonctionnelles d'utilisation du système. Ils sont décrits sous la forme d'actions et de réactions le comportement du système du point de vue d'un utilisateur et représentent un ensemble de séquences d'action réalisées par le système et produisant un résultat observable pour un acteur particulier.

#### **2.2.2.2. Le diagramme de séquence**

C'est la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation Unified Modeling Language. On montre ces interactions dans le cadre d'un scénario d'un Diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les



acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

Le diagramme de séquences est un diagramme d'interaction UML. Il représente les échanges des messages entre les objets du point de vue temporel et permet de détailler un diagramme de cas d'utilisation en décrivant un scénario. Il se base sur les concepts suivants :

- Objet : description d'un objet du monde réel (une personne ou une chose).
- Message : c'est le véhicule de la communication entre les objets.

#### **2.2.2.3. Diagramme d'activité**

Un diagramme d'activité permet de modéliser un processus interactif, global ou partiel pour un système donné (logiciel, système d'information). Il est recommandable pour exprimer une dimension temporelle sur une partie du modèle, à partir de diagrammes de classes ou de cas d'utilisation, par exemple.

Le diagramme d'activités est une représentation proche de l'organigramme ; la description d'un cas d'utilisation par un diagramme d'activités correspond à sa traduction algorithmique.

Une activité est l'exécution d'une partie du cas d'utilisation, elle est représentée par un rectangle aux bords arrondis.

Le diagramme d'activité est principalement un organigramme qui montre le flot de contrôle d'une activité à l'autre. On l'utilise pour modéliser les aspects dynamiques d'un système.

#### **2.2.2.4. Diagramme de Classe**

Les diagrammes de classes sont la représentation de la structure statique en termes de classes et de relations. Les objets sont les instances des classes et les liens les instances des relations. Le diagramme de classes est considéré comme le plus important de la modélisation orientée objet, il est le seul obligatoire lors d'une telle modélisation. Alors que le diagramme de cas d'utilisation montre un système du point de vue des acteurs, le diagramme de classes en montre la structure interne. Il permet de fournir une représentation abstraite des objets du système qui vont interagir ensemble pour réaliser les cas d'utilisation. Il est important de noter qu'un même objet peut très bien intervenir dans la réalisation de plusieurs cas d'utilisation. Les cas d'utilisation ne réalisent donc pas une

partition1 des classes du diagramme de classes. Un diagramme de classes n'est donc pas adapté (sauf cas particulier) pour détailler, décomposer, ou illustrer la réalisation d'un cas d'utilisation particulier.

### **2.2.3. Pourquoi utiliser UML**

De nos jours, les outils de modélisation de processus métier s'étoffent chaque année et les suites logicielles sont de plus en plus nombreuses. L'usage et les fonctionnalités d'UML diffèrent d'un périmètre à un autre, selon les besoins des clients et des fournisseurs d'applications. Dans notre application, le recours à la modélisation UML procure de nombreux avantages qui agissent sur :

- La modularité.
- L'abstraction.
- La structuration cohérente des fonctionnalités et des données.
- De bien définir les besoins clients.
- D'éviter des surcoûts liés à la livraison d'un logicielle qui ne satisfait pas le client.
- Vulgariser les aspects liés à la conception et à l'architecture, propres au logiciel, au client
- Elle apporte une compréhension rapide du programme à d'autres développeurs externes en cas de reprise du logiciel et facilite sa maintenance.

## **2.3. Conception de notre application**

### **2.3.1. Les diagrammes de cas d'utilisation**

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel.

#### **2.3.1.1. Identification des acteurs**

Un acteur représente une personne, un matériel ou un logiciel qui interagit directement avec le système en question. Nous avons identifié principalement pour notre application trois acteurs comme il est décrit dans la figure ci-dessous :

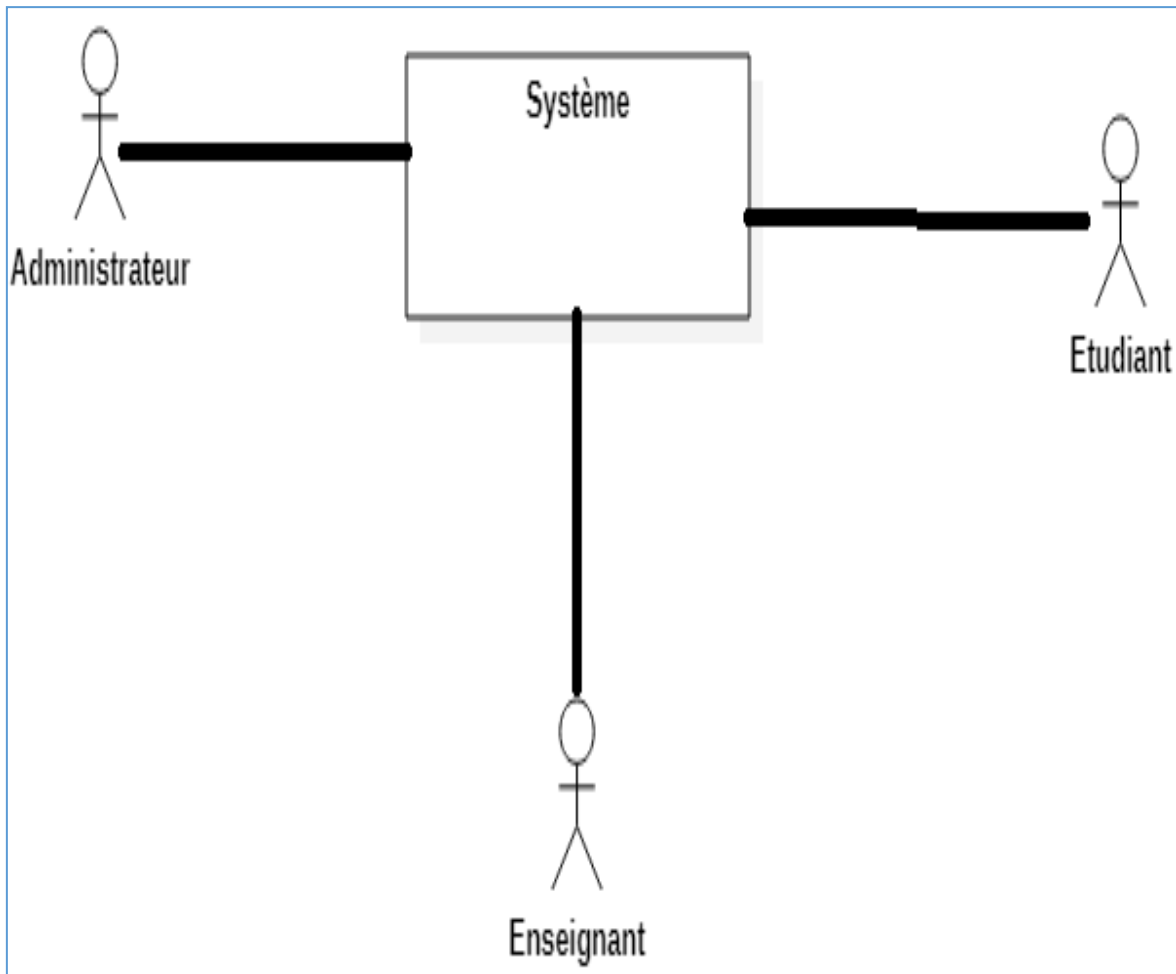


FIGURE 2. 1: ACTEURS DE NOTRE APPLICATION

### 2.3.1.2. Diagramme de cas d'utilisation globale

Les cas d'utilisation représentent une technique puissante pour illustrer le comportement détaillé d'un système en se basant sur les besoins des utilisateurs, ils sont des outils formels qui permettent d'exprimer les interactions et les dialogues des utilisateurs nommés acteurs du système. Un cas d'utilisation décrit un scénario particulier dans lequel le système fournit une sortie observable à la suite d'entrées fournies par un acteur particulier. Il est nécessaire d'adjoindre à chaque cas d'utilisation une description détaillée. Cette description est parfois textuelle et composée de plusieurs rubriques dont les importantes sont :

- Le scénario nominal : enchaînement d'actions typiques dans le cas où les choses se passent comme prévu.
- Les enchaînements alternatifs : enchaînements dans des cas particuliers.

Ce diagramme, qui est représenté par la figure ci-dessous, englobe les principales fonctionnalités offertes par notre application :

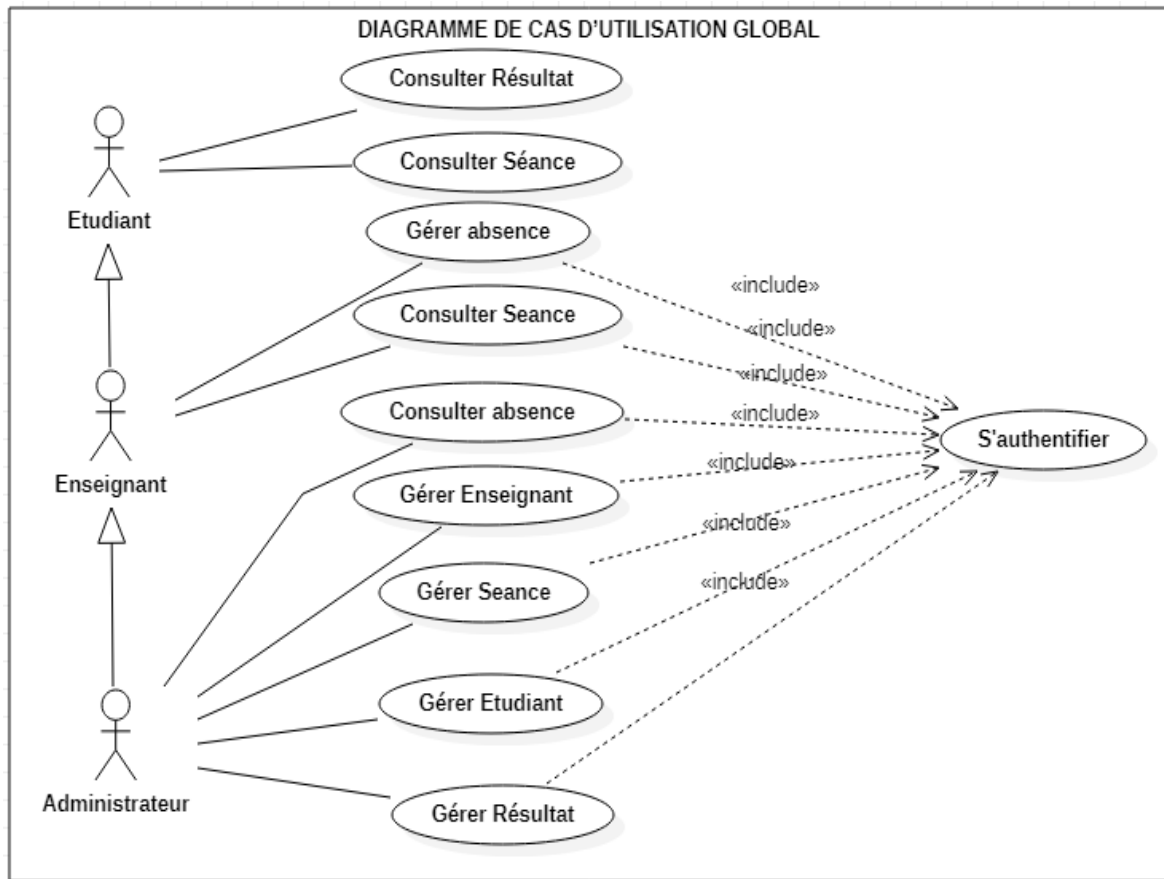


FIGURE 2. 2: DIAGRAMME DE CAS D'UTILISATION GLOBAL

### 2.3.1.3. Diagramme de cas d'utilisation pour l'administrateur

Le schéma suivant représente le diagramme des cas d'utilisation illustrant toutes les tâches principales pour l'administrateur sur l'application web.

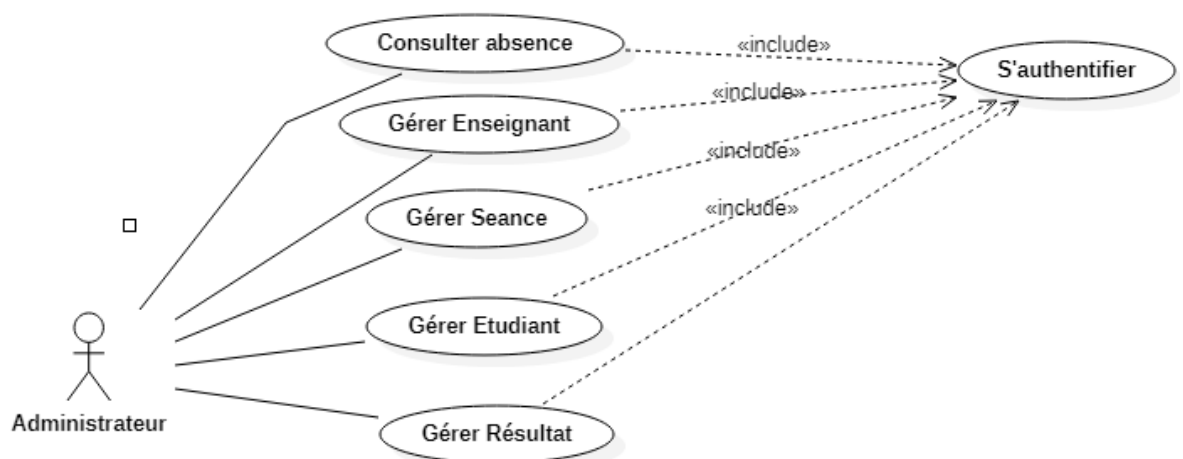


FIGURE 2. 3: DIAGRAMME DE CAS D'UTILISATION POUR L'ADMINISTRATEUR

Notre application permet à l'Administrateur d'effectuer plusieurs tâches telles que gérer Etudiant, gérer Enseignant, gérer séance le résultat.

### Diagramme de cas d'utilisation authentification

Dans un souci de sécurité de notre système, et pour la sécurisation des informations, avant d'effectuer une tâche quelconque, tous les utilisateurs du système doivent s'authentifier en saisissant leurs identifiants (Login, mot de passe). La figure suivante illustre le scenario d'authentification :

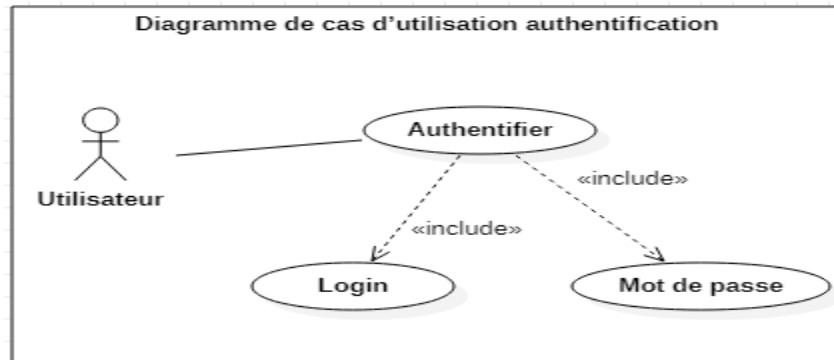


FIGURE 2. 4: DIAGRAMME DE CAS D'UTILISATION "S'AUTHTIFIER"

### 2.3.2. Les diagrammes de séquences et activités

Le diagramme de séquence représente une interaction entre plusieurs éléments qui interagissent par l'envoi des messages. Selon un ordre chronologique.

#### 2.3.2.1. Diagramme de séquences de cas d'utilisation " s'authentifier "

Il s'agit d'un tableau qui présente une description de cas d'utilisation " s'authentifier " :

TABEAU 2. 1: DESCRIPTION DE CAS D'UTILISATION " S'AUTHTIFIER "

Cas d'utilisation	Scenario Nominal	Scenario Alternatif	Acteur
" S'authentifier "	1. L'utilisateur demande fenêtre d'authentification 2. Le système affiche la fenêtre d'authentification 3. L'utilisateur saisit Login et mot de passe 4. Le système vérifie les données d'authentification. 5. le système affiche espace spécifique.	Si login et/ou mot de passe incorrect : 6. Le système affiche un message d'erreur ➤ Le scenario 3 se répète	➤ <b>Administrateur</b> ➤ <b>Enseignant</b>

Les deux figures suivantes représentent le diagramme de séquence ainsi que celui d'activité de scénarios « connexion » qui appartient à l'utilisateur (Administrateur/Agent Commerciale) :

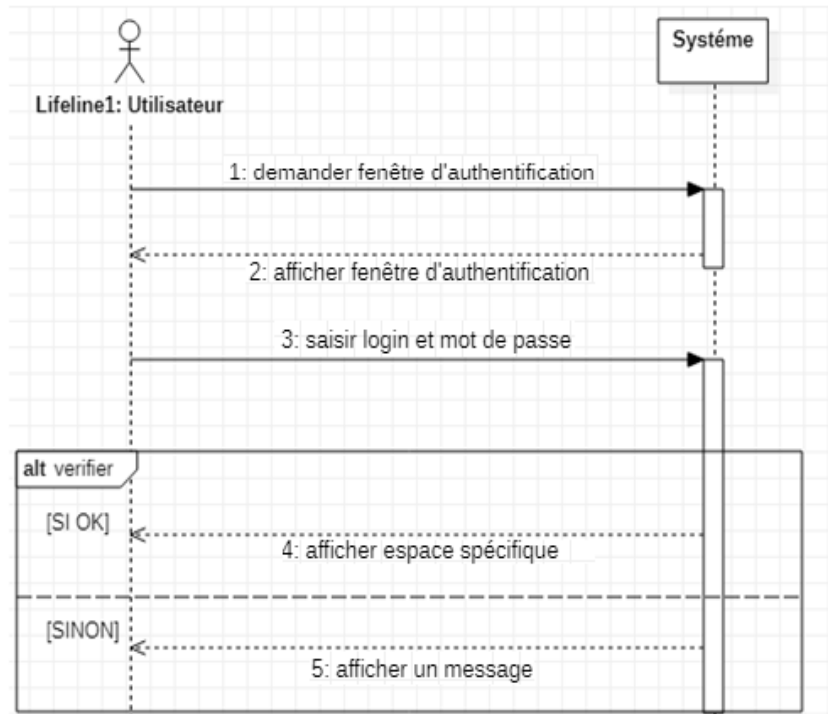


FIGURE 2. 5: DIAGRAMME DU SEQUENCE DE CAS D'UTILISATION " S'AUTHTENTIFIER "

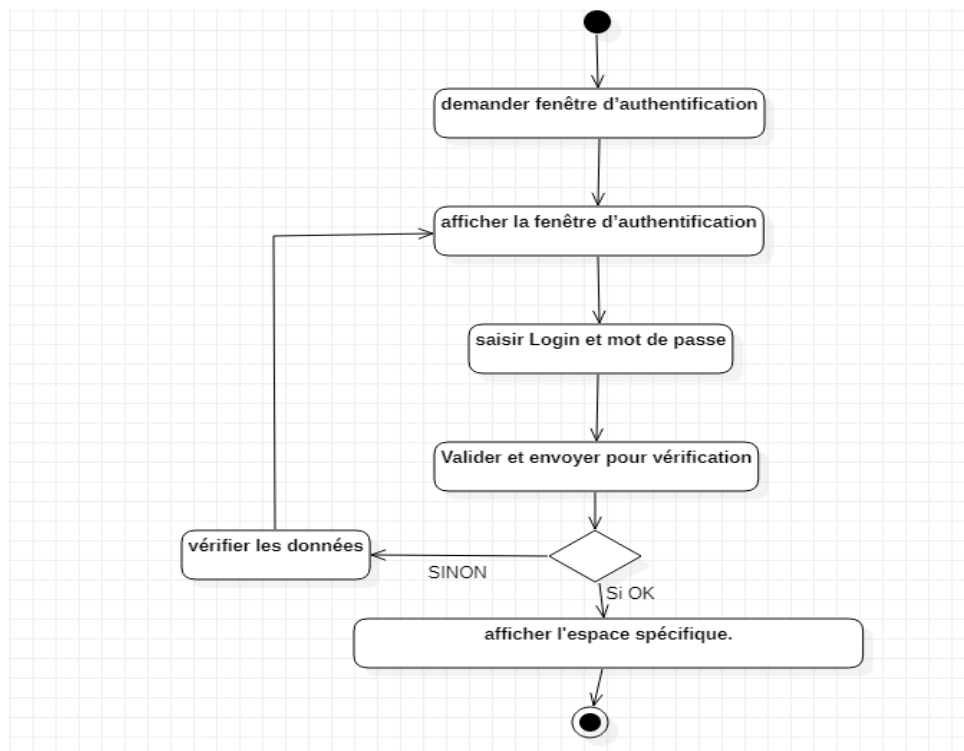


FIGURE 2. 6: DIAGRAMME D'ACTIVITE DE LA FONCTION " CONNEXION "

### 2.3.2.2. Diagramme de séquence de cas d'utilisation " Gérer Enseignant "

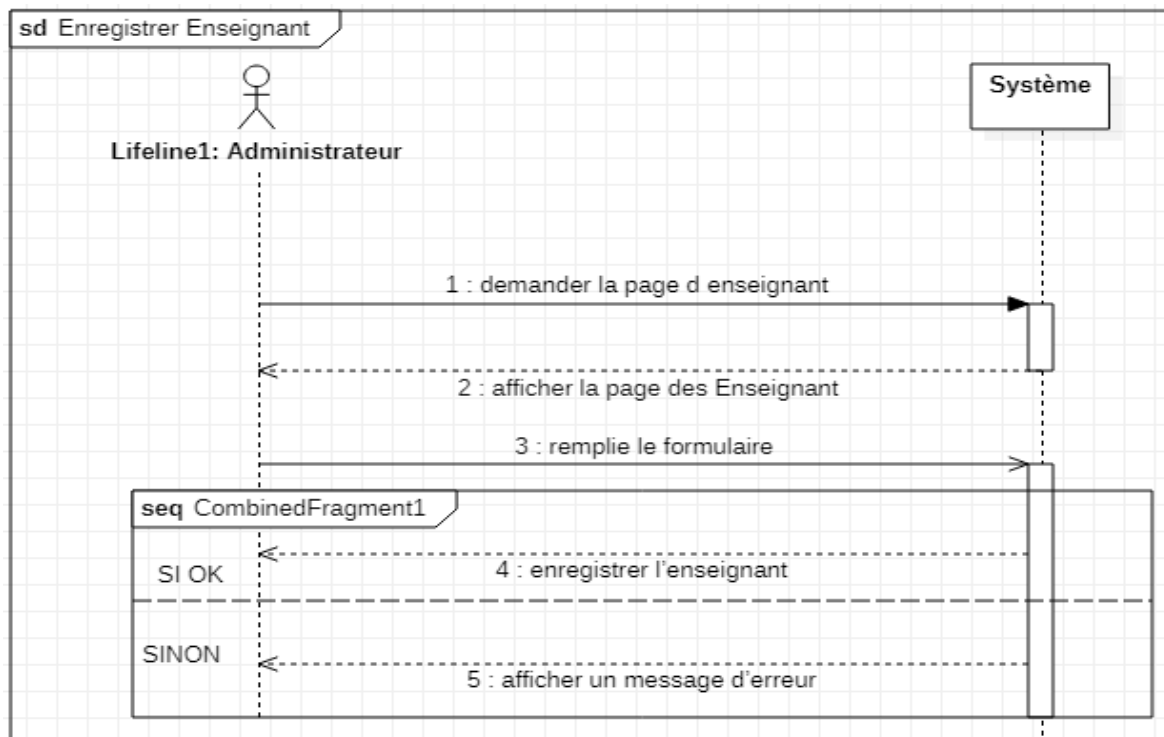
#### ✓ *Enregistrer Enseignant*

Il s'agit d'un tableau qui présente une description de cas d'utilisation" enregistrer Enseignant " :

**TABEAU 2. 2: DESCRIPTION DE CAS D'UTILISATION " ENREGISTRER ENSEIGNANT"**

Cas d'utilisation	Scenario Nominal	Scenario Alternatif	Acteur
« Enregistrer Enseignant»	1. L'administrateur demande la page des Enseignant. 2. Le système affiche la page des Enseignant. 3. L'administrateur remplit le formulaire. 4. le système valide l'ajout.	5. le système vérifie les champs de formulaire. → Si ok enregistrer l'enseignant. →Sinon afficher un message d'erreur.	➤ <b>Administrateur</b>

Les deux figures suivantes représentent le diagramme de séquence ainsi que celui d'activité de scénarios " Enregistrer Enseignant" qui appartient à l'administrateur :



**FIGURE 2. 7: DIAGRAMME DE SEQUENCE DE CAS D'UTILISATION " ENREGISTRER ENSEIGNANT "**

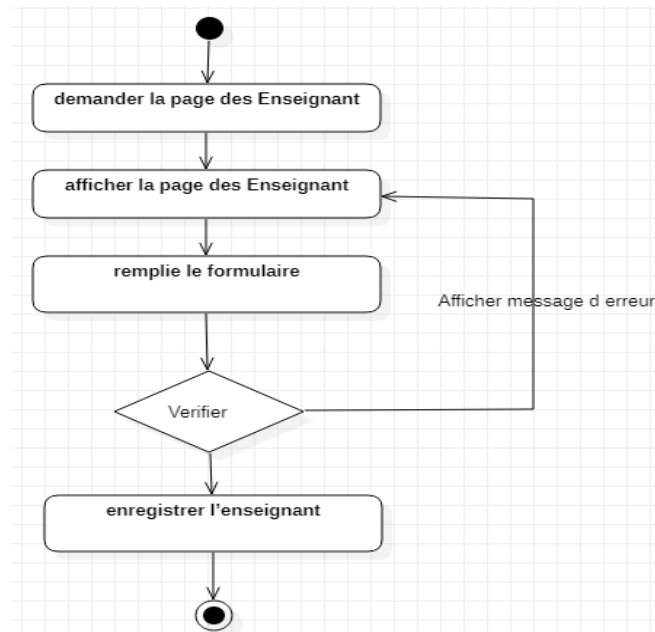


FIGURE 2. 8: DIAGRAMME D'ACTIVITE "ENREGISTRER ENSEIGNANT "

✓ *Modifier Etudiant*

Il s'agit d'un tableau qui présente une description de cas d'utilisation "modifier Etudiant " :

TABLEAU 2. 3: DESCRIPTION DE CAS D'UTILISATION " MODIFIER ETUDIANT "

Cas d'utilisation	Scenario Nominal	Scenario Alternatif	Acteur
"Modifier Etudiant "	1. L'administrateur demande la liste des Etudiants. 2. Le système affiche la liste des Etudiants. 3. L'administrateur choisit un Etudiant et demande la page de modification. 4. L'administrateur modifier le formulaire. 5. le système valide la modification	6. Le système vérifie le champ de formulaire. Si oui afficher la nouvelle liste des Etudiants Sinon afficher message d'erreur et le scénario 4 se répète.	Administrateur



Les deux figures suivantes représentent le diagramme de séquence ainsi que celui d'activité de scénarios "modifier Etudiant " qui appartient à l'administrateur :

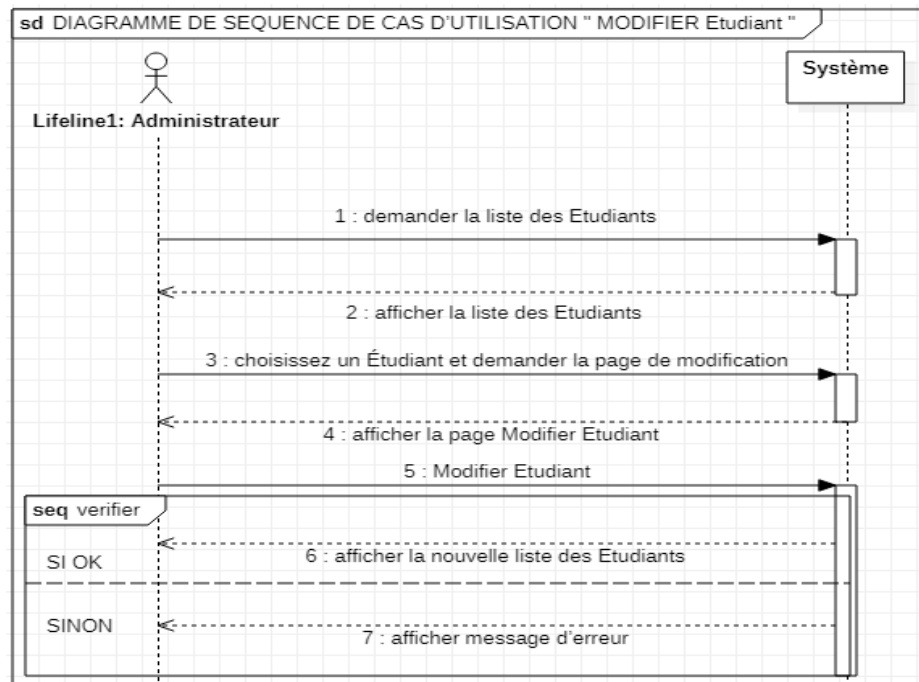


FIGURE 2. 9: DIAGRAMME DE SEQUENCE DE CAS D'UTILISATION " MODIFIER ETUDIANTS "

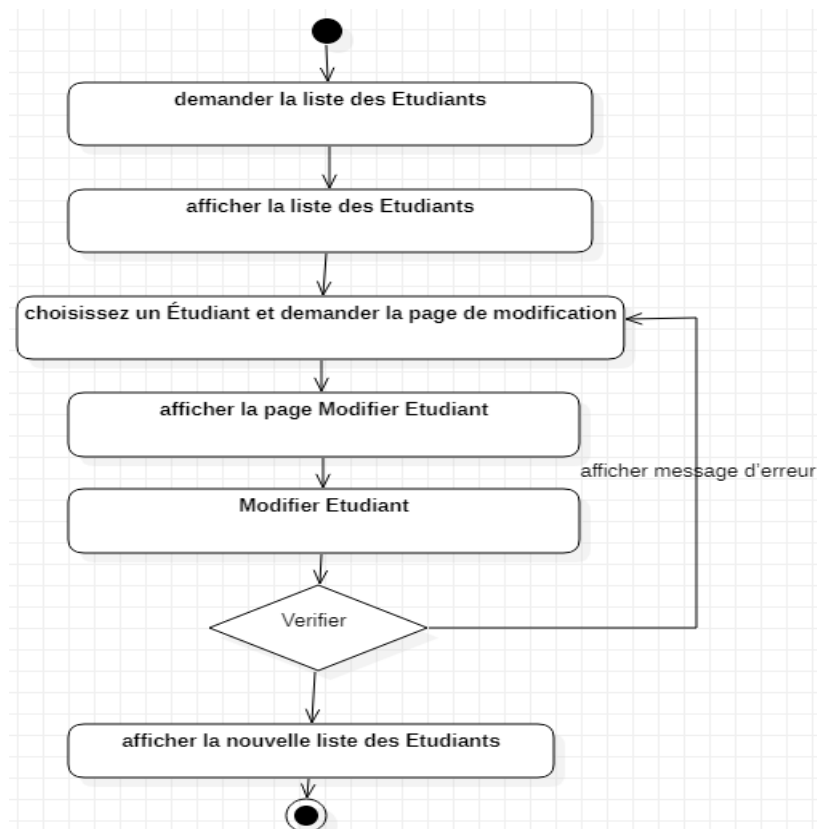


FIGURE 2. 10: DIAGRAMME D'ACTIVITE " MODIFIER ETUDIANTS "

### 2.3.2.3. Diagramme de séquence "gérer Résultat "

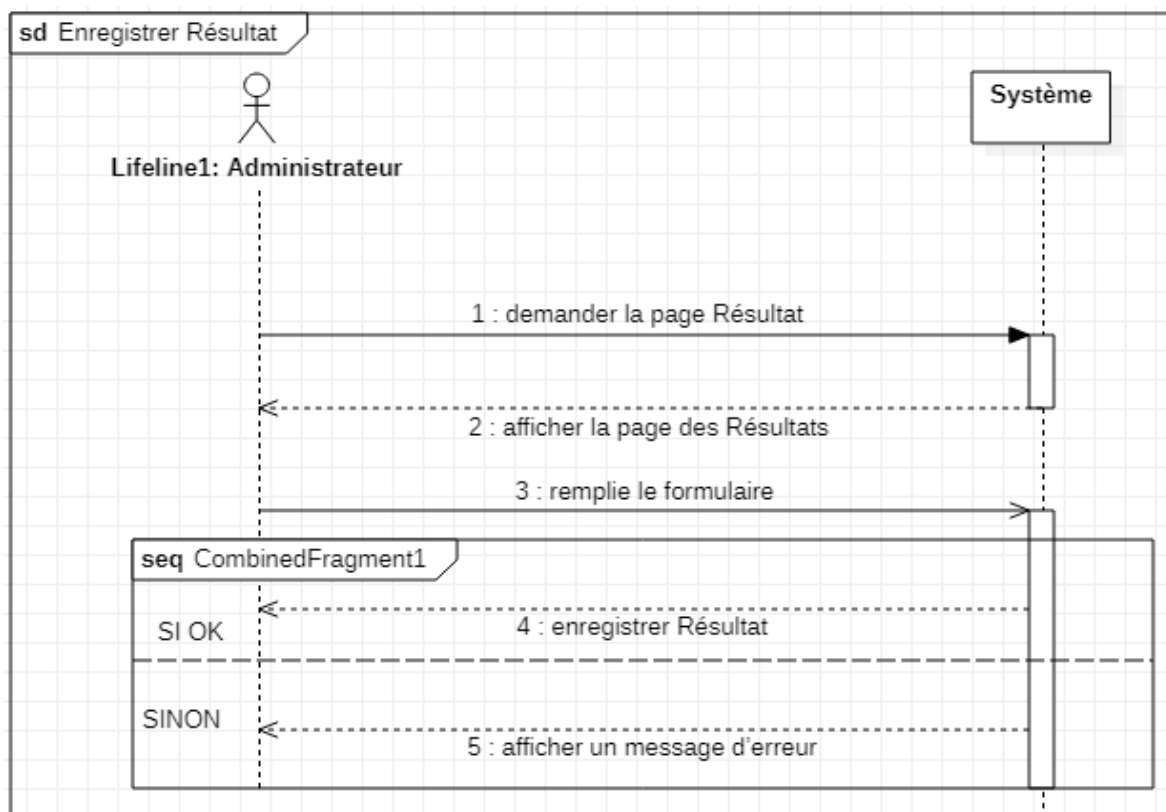
#### ➤ Enregistrer Résultat

Il s'agit d'un tableau qui présente une description de cas d'utilisation "enregistrer Résultat " :

**TABLEAU 2. 4: DESCRIPTION DE CAS D'UTILISATION " ENREGISTRER RESULTAT "**

Cas d'utilisation	Scenario Nominal	Scenario Alternatif	Acteur
"Gérer Résultat "	1. L'administrateur demande la page des Résultat. 2. Le système affiche la page des Résultat. 3. L'administrateur remplit le formulaire. 4. le système valide l'ajout.	4. le système vérifie les champs Résultat. → Si ok enregistrer Résultat. → Si non ok afficher un message d'erreur.	<b>Administrateur</b>

La figure suivante représente le diagramme de séquence ainsi que celui d'activité de scénarios "enregistrer Résultat " qui appartient à l'administrateur :



**FIGURE 2. 11: DIAGRAMME DE SEQUENCE DE CAS D'UTILISATION " ENREGISTRER RESULTAT "**

✓ *Modifier Résultat*

Il s'agit d'un tableau qui présente une description de cas d'utilisation "modifier Résultat " :

TABLEAU 2. 5: DESCRIPTION DE CAS D'UTILISATION " MODIFIER RESULTAT "

Cas d'utilisation	Scenario Nominal	Scenario Alternatif	Acteur
"Modifier Résultat"	1. L'administrateur demande la liste des Résultats. 2. Le système affiche la liste des Résultat. 3. L'administrateur choisit un Résultat et demande la page de modification. 4. L'administrateur modifier le formulaire. 5. le système valide la modification	5. Le système vérifie le champ de formulaire. → Si oui afficher la nouvelle liste résultats → Si non afficher message d'erreur et le scénario 4 se répète.	Administrateur

La figure suivante représente le diagramme de séquence ainsi que celui d'activité de scénarios "modifier résultat " qui appartient à l'administrateur :

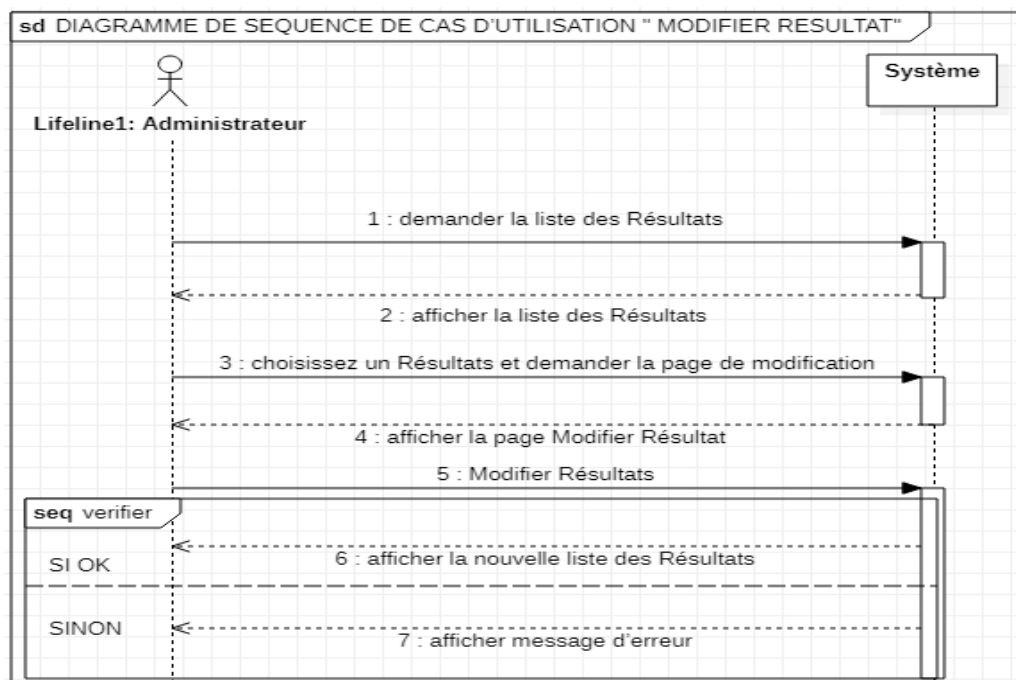


FIGURE 2. 12: DIAGRAMME DE SEQUENCE DE CAS D'UTILISATION " MODIFIER RESULTAT "

✓ *Supprimer Résultat*

Il s'agit d'un tableau qui présente une description de cas d'utilisation "supprimer Résultat " :

TABLEAU 2. 6: DESCRIPTION DE CAS D'UTILISATION " SUPPRIMER RESULTAT"

Cas d'utilisation	Scenario Nominal	Scenario Alternatif	Acteur
"Supprimer Résultat"	1. L'administrateur demande la liste des Résultats. 2. Le système affiche la liste des Résultats. 3. L'administrateur choisie un Résultat et confirmer suppression. 4. le système valide la suppression.		<b>Administrateur</b>

La figure suivante représente le diagramme de séquence ainsi que celui d'activité de scénarios "supprimer Résultat " qui appartient à l'administrateur :

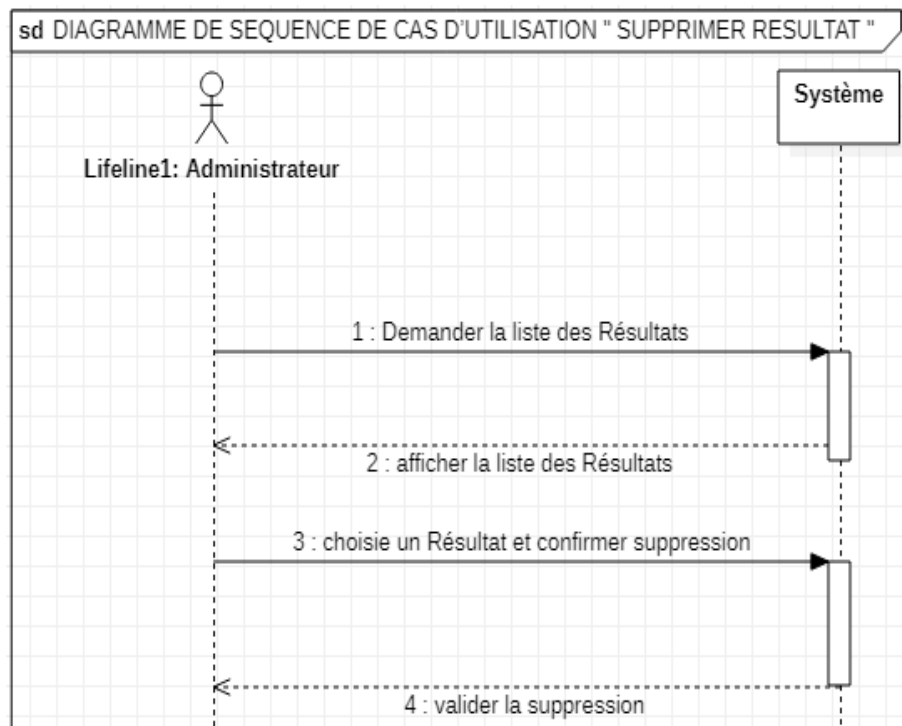


FIGURE 2. 13: DIAGRAMME DE SEQUENCE DE CAS D'UTILISATION " SUPPRIMER RESULTAT "

#### 2.3.2.4. Diagramme de cas d'utilisation coté Enseignant

Le schéma suivant représente le diagramme des cas d'utilisation illustrant toutes les tâches principales pour l'enseignant sur l'application web.



FIGURE 2. 14:DIAGRAMME DE CAS D'UTILISATION COTE ENSEIGNANT

##### ➤ *Enregistrer absence*

Il s'agit d'un tableau qui présente une description de cas d'utilisation "enregistrer absence " :

TABLEAU 2. 7:DESCRIPTION DE CAS D'UTILISATION " ENREGISTRER ABSENCE "

Cas d'utilisation	Scenario Nominal	Scenario Alternatif	Acteur
"Gérer absence"	1. L'enseignant demande la page des absences. 2. Le système affiche la page des absences. 3. L'enseignant remplit le formulaire. 4. le système valide l'ajout.	4. le système vérifie les champs d'utilisateur. → Si ok enregistrer absence. →Si non ok afficher un message d'erreur.	Enseignant

La figure suivante représente le diagramme de séquence ainsi que celui d'activité de scénarios "enregistrer absence " qui appartient à l'enseignant:

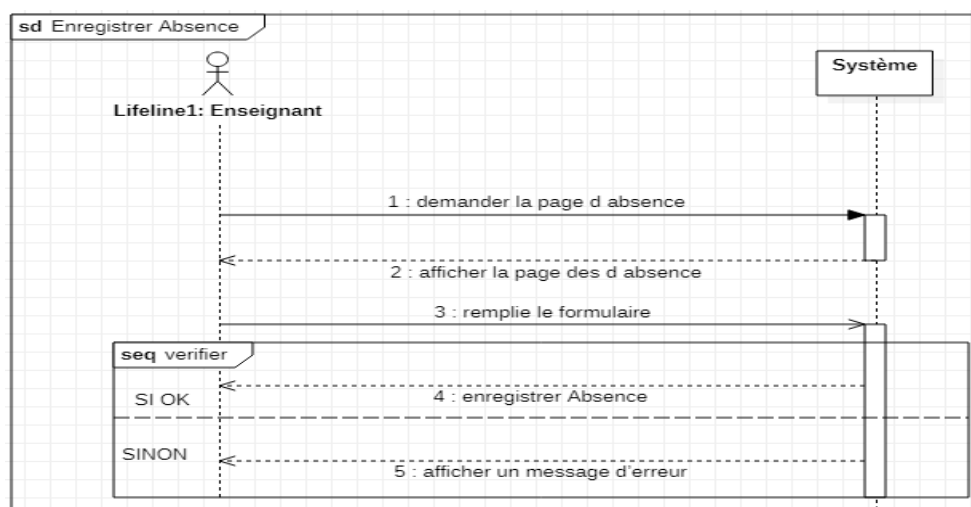


FIGURE 2. 15: DIAGRAMME DE SEQUENCE DE CAS D'UTILISATION " ENREGISTRER ABSENCE "

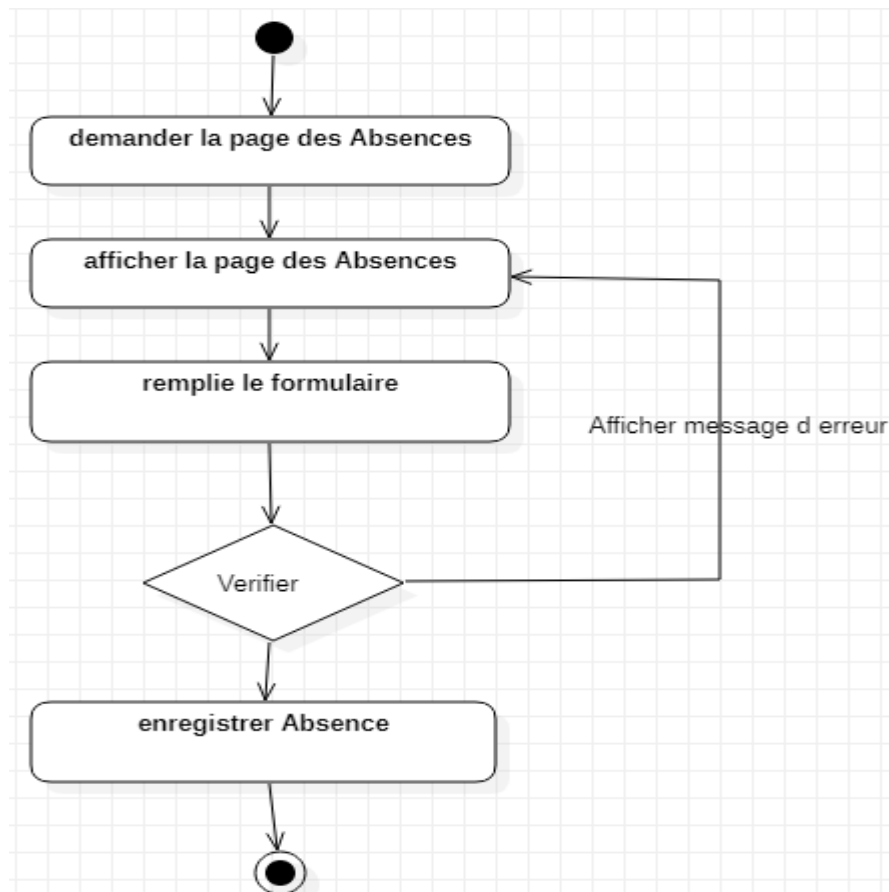


FIGURE 2. 16: DIAGRAMME D'ACTIVITE ENREGISTRER ABSENCE

✓ *Modifier Absence*

Il s'agit d'un tableau qui présente une description de cas d'utilisation "modifier Absence " :

TABLEAU 2. 8: DESCRIPTION DE CAS D'UTILISATION " MODIFIER ABSENCE "

Cas d'utilisation	Scenario Nominal	Scenario Alternatif	Acteur
"Modifier Absence"	1. L'enseignant demande la liste des Absences. 2. Le système affiche la liste des Absences. 3. L'enseignant choisit un Absence et demande la page de modification. 4. L'enseignant modifier le formulaire. 5. le système valide la modification	5. Le système vérifie le champ de formulaire. → Si oui afficher la nouvelle liste des Absences → Si non afficher message d'erreur et le scénario 4 se répète.	Enseignant

La figure suivante représente le diagramme de séquence ainsi que celui d'activité de scénarios "modifier Absence « qui appartient à L'enseignant:

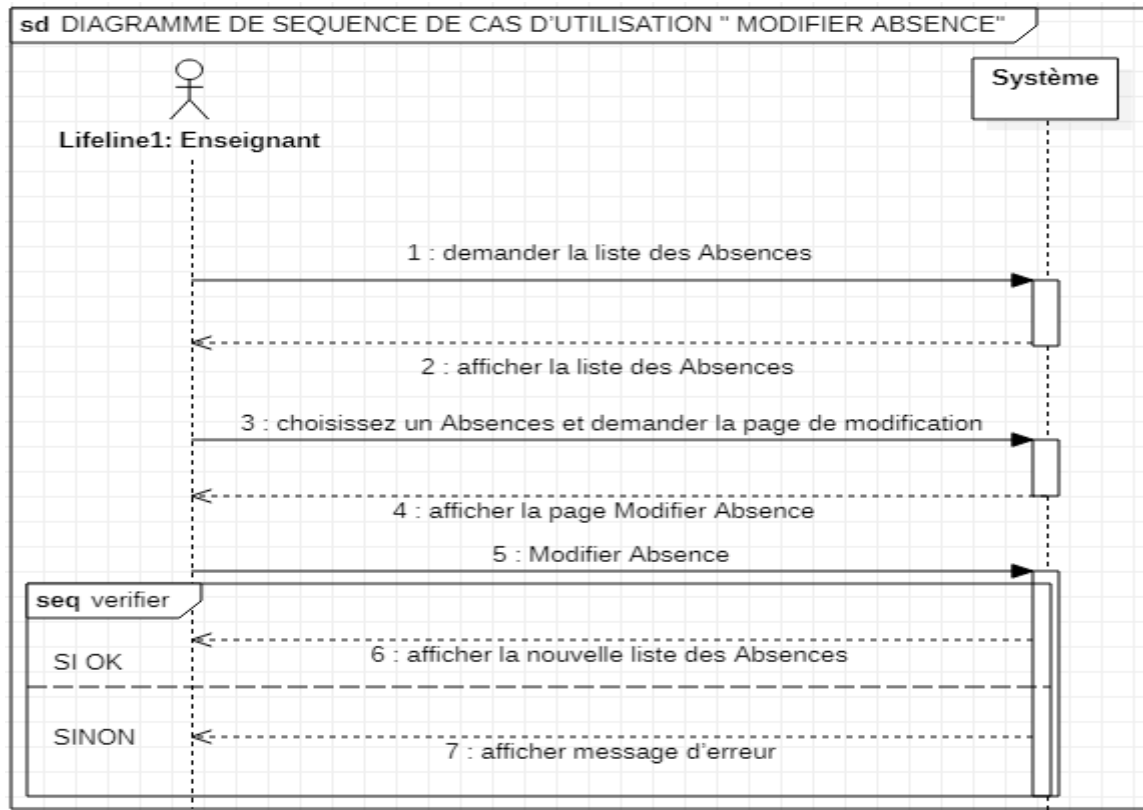


FIGURE 2. 17: DIAGRAMME DE SEQUENCE DE CAS D'UTILISATION " MODIFIER ABSENCE "

✓ *Supprimer Absence*

Il s'agit d'un tableau qui présente une description de cas d'utilisation" supprimer Absence " :

TABLEAU 2. 9: DESCRIPTION DE CAS D'UTILISATION " SUPPRIMER ABSENCE"

Cas d'utilisation	Scenario Nominal	Scenario Alternatif	Acteur
"Supprimer Absence"	1. L'enseignant demande la liste des absences. 2. Le système affiche la liste des absences. 3. L'enseignant choisie un Absence et confirmer suppression. 4. le système valide la suppression.		Enseignant

La figure suivante représente le diagramme de séquence de scénarios "supprimer absence " qui appartient à l'enseignant:

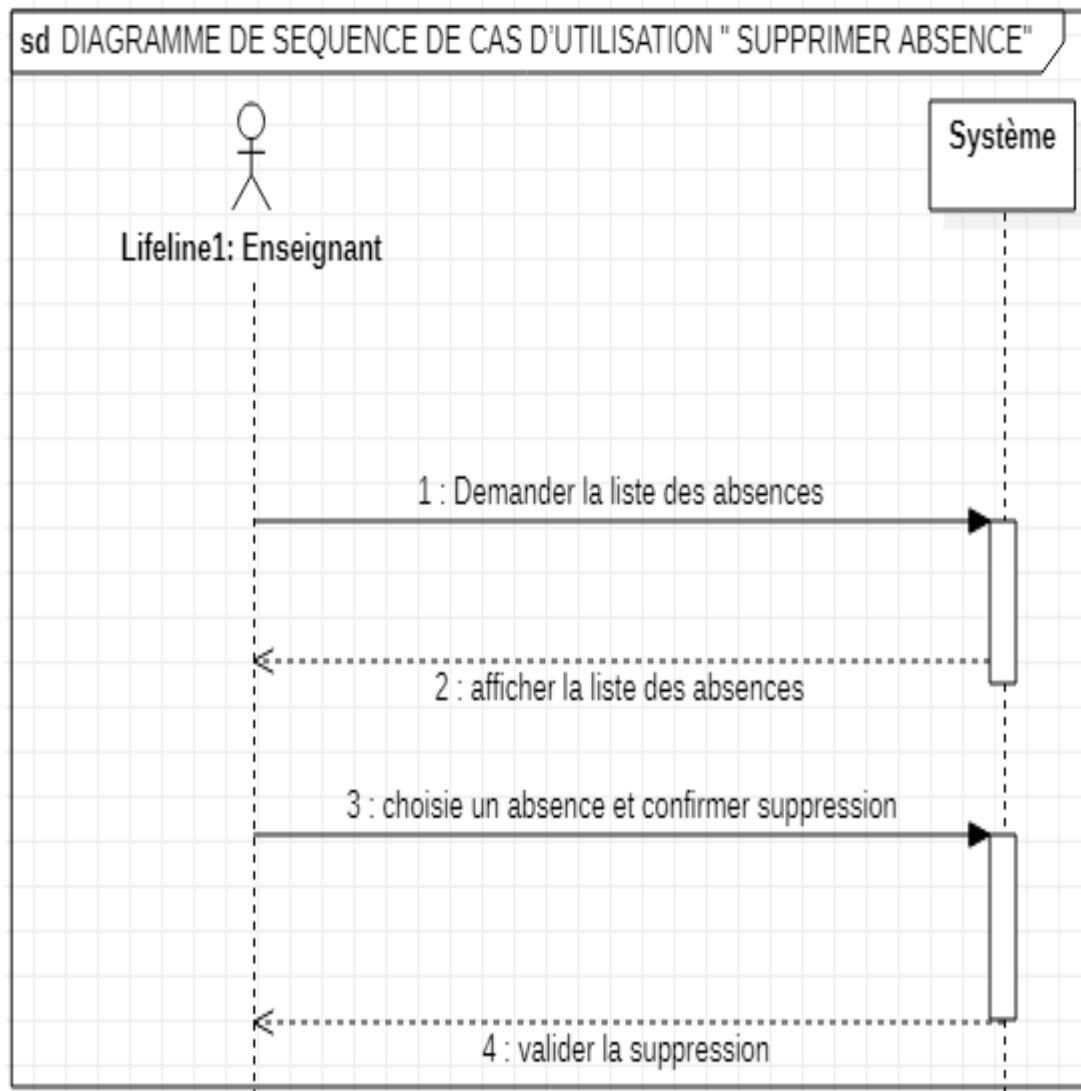


FIGURE 2. 18: DIAGRAMME DE SEQUENCE DE CAS D'UTILISATION " SUPPRIMER ABSENCE "

### 2.3.3. Le diagramme de classes

Le diagramme de classes représente les classes intervenant dans le système. Il est une représentation statique des éléments qui composent un système et de leurs relations.



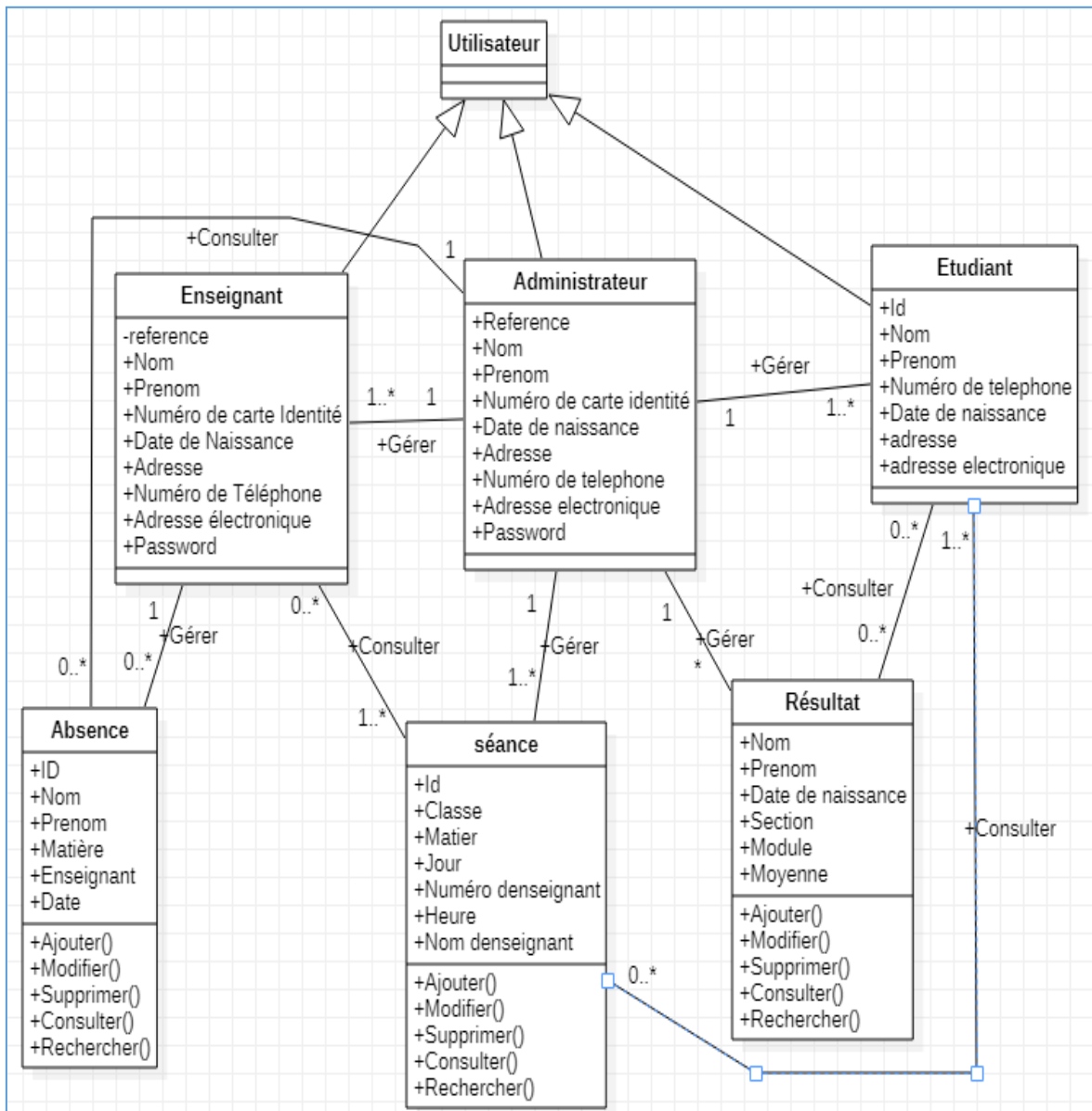


FIGURE 2. 19: DIAGRAMME DE CLASSES

## 2.4. Conclusion

Dans ce chapitre nous avons présenté le diagramme de cas d'utilisation générale de notre projet ainsi que les diagrammes des séquences et le diagramme de class. Par la suite, on doit chercher les solutions possibles pour développer l'application, ce que nous voyons présenter dans le chapitre suivant.

# ***Chapitre 3 :***

## ***Réalisation***

### 3.1. Introduction

Ce chapitre va nous servir à représenter, en premier lieu, l'environnement de travail de notre application qui comprend l'environnement matériel et l'environnement logiciel ainsi que la conception de la base de données. Et il permet, en deuxième lieu, de représenter les différentes interfaces homme machine élaborées au cours de ce projet.

### 3.2. Environnement logiciel et matériel

#### 3.2.1. Environnement matériel

Durant ce présent projet de fin d'études, tout le travail a été réalisé sur deux ordinateurs qui ont les caractéristiques techniques présentés dans le tableau suivant :

**TABLEAU 3. 1: ENVIRONNEMENT MATERIEL**

	Ordinateur portable
Processeur	pentium ® duel-Core CPU 2.13GHz
Mémoire (RAM)	3.00 Go
Carte graphique	Intel® HD Graphics
Type de système	Windows 7

#### 3.2.2. Environnement logiciel

Au cours de développement de notre application nous avons utilisé les outils logiciels suivants :

##### ➤ Notepad++

Notepad++ est un éditeur de texte libre générique, fonctionnant sous Windows, codé en C++, qui intègre la coloration syntaxique de code source pour les langages et fichiers C, C++, Java, C#, XML, ...

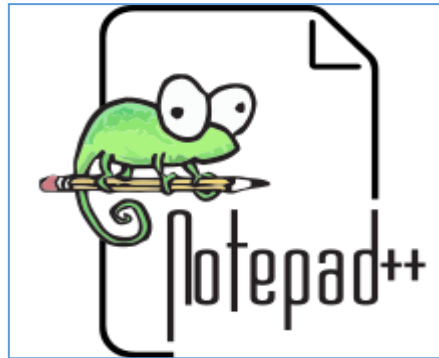


FIGURE3. 1: NOTEPAD++

### ➤ Xampp

XAMPP est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide.

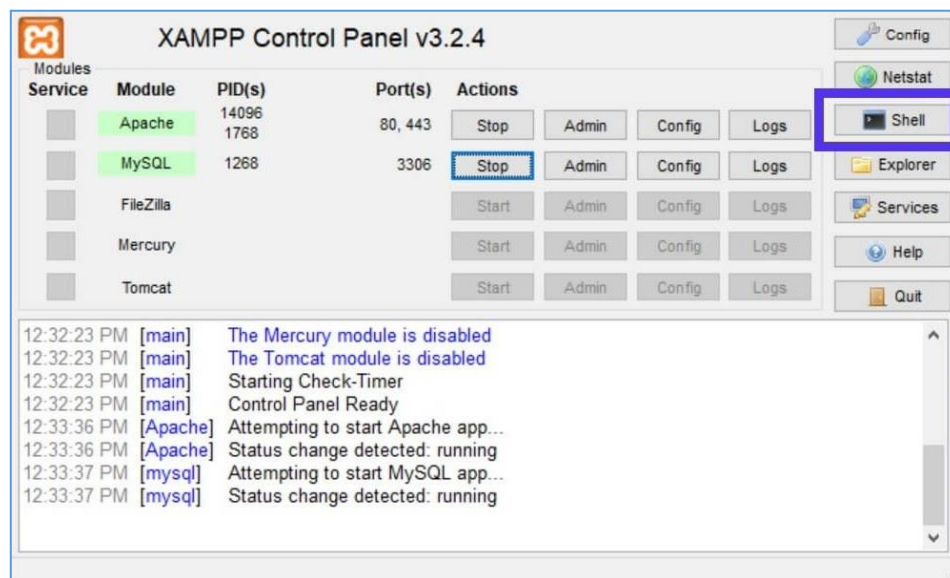


FIGURE3. 2: XAMPP

### ➤ MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle,

Informix et Microsoft SQL Server. Son nom vient du prénom de la fille du Co créateur Michael Widenius, MySQL fait allusion au Structured Query Language, le langage de requête utilisé.



FIGURE3. 3: LOGO MYSQL

### ➤ Star UML

StarUML est un logiciel de modélisation UML, cédé comme open source par son éditeur, à la fin de son exploitation commerciale.

Star UML gère la plupart des diagrammes spécifiés.



FIGURE3. 4: SIGLESTAR UML

## 3.3. Présentation des pages web de l'application :

Les interfaces graphiques de l'application sont très importantes, car elles permettent de faciliter le dialogue entre l'homme et la machine ainsi que d'améliorer les performances de l'application. Dans la conception des interfaces de notre application nous avons respecté un ensemble des choix ergonomiques comme la lisibilité, la compréhensibilité, etc. Dans ce qui suit une présentation des captures écrans des plus importantes interfaces de l'application.

### 3.3.1. L'authentification

La fenêtre de la figure ci-dessous (figure 3.5) permet authentifier des acteurs (Administrateur/Enseignant et Etudiant), si le login et le mot de passe est correct l'application les redirige vers leurs vues associées. Sinon recharge la page d'authentification.

Notre application est gérée par Trois principaux utilisateurs et chacun peut consulter son propre espace. Pour se connecter, un utilisateur doit remplir les champs Nom et Mot de passe ensuite, il doit cliquer sur le bouton Connexion. Si tout va bien, l'interface spécifique pour chaque utilisateur sera affichée. Sinon un message d'erreur s'affiche.

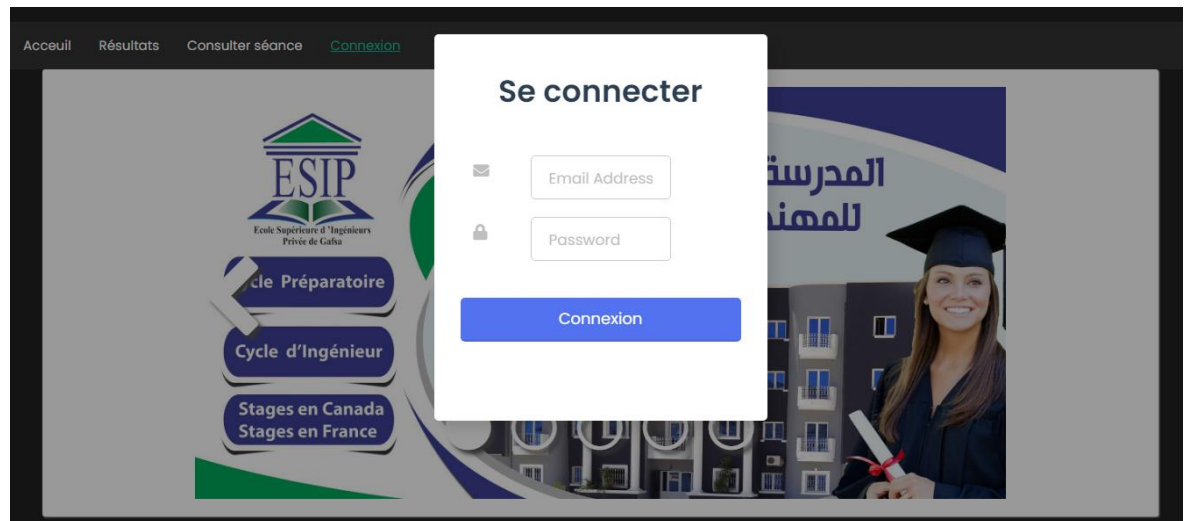


FIGURE3. 5: FENETRE D'AUTHENTIFICATION

### 3.3.2. Fenêtre « Ajouter Enseignant »

La figure ci-dessous présente l'interface d'inscription à remplir par L'administrateur. Dans cette interface l'administrateur doit remplir tous les champs puis Enregistrer pour être un bénéficiaire de l'application. Le bouton "ENREGISTRER" permet d'enregistrer le numéro de matricule, le nom, numéro de carte d'identité, date de naissance, adresse, numéro de téléphone, adresse électronique et leur mot de passe de chaque enseignant dans la base de données en remplissant le formulaire et en cliquant sur le bouton Enregistrer.

FIGURE3. 6: FENETRE « AJOUTER ENSEIGNANT »

### 3.3.3. Fenêtre « Liste des Enseignant »

Cette interface représente Liste des Enseignants où l'administrateur peut ajouter, modifier, supprimer un Enseignant.





No	Nom utilisateur	Mot de pass	Numéro de téléphone	Adresse electronique	Operation
1	admin	adminadmin	2147483647	ikram@gmail.com	 
2	Safa	safasafa	22765567	Safa@safa.com	 

FIGURE3. 7: LISTE DES ENSEIGNANTS

### 3.3.4. Fenêtre « Supprimer Enseignant »

Pour supprimer un enregistrement dans la liste des enseignants (figure 3.7), on choisit un Enseignant et confirmer suppression de l'enseignant. Le système s'affiche la nouvelle liste des enseignants enregistrés dans la base de données.

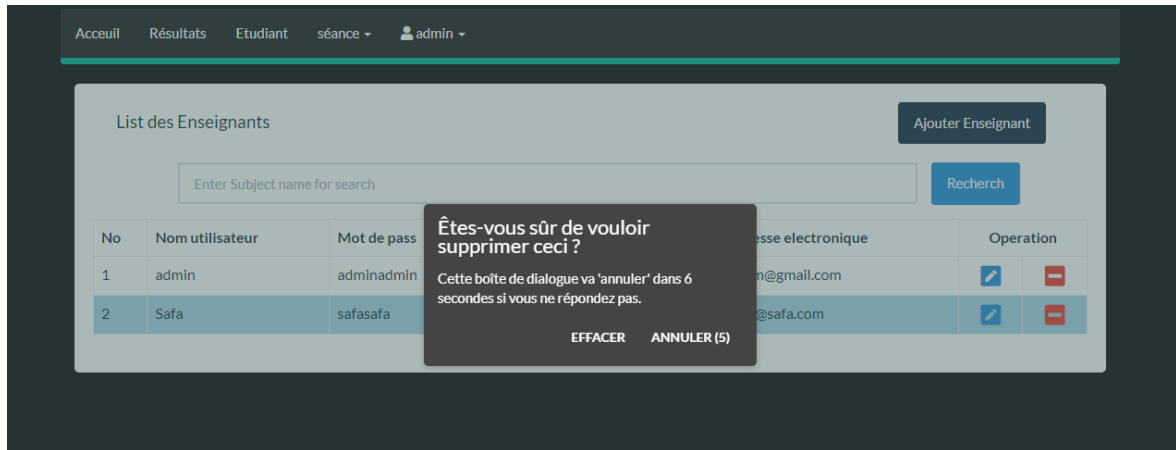


FIGURE3. 8: SUPPRIMER ENSEIGNANT

### 3.1.1. Page Gérer Résultat :

La figure ci-dessous est la première interface qui sera affichée lors de la gestion des résultats. Cette interface comprend tous les niveaux d'école ESIP

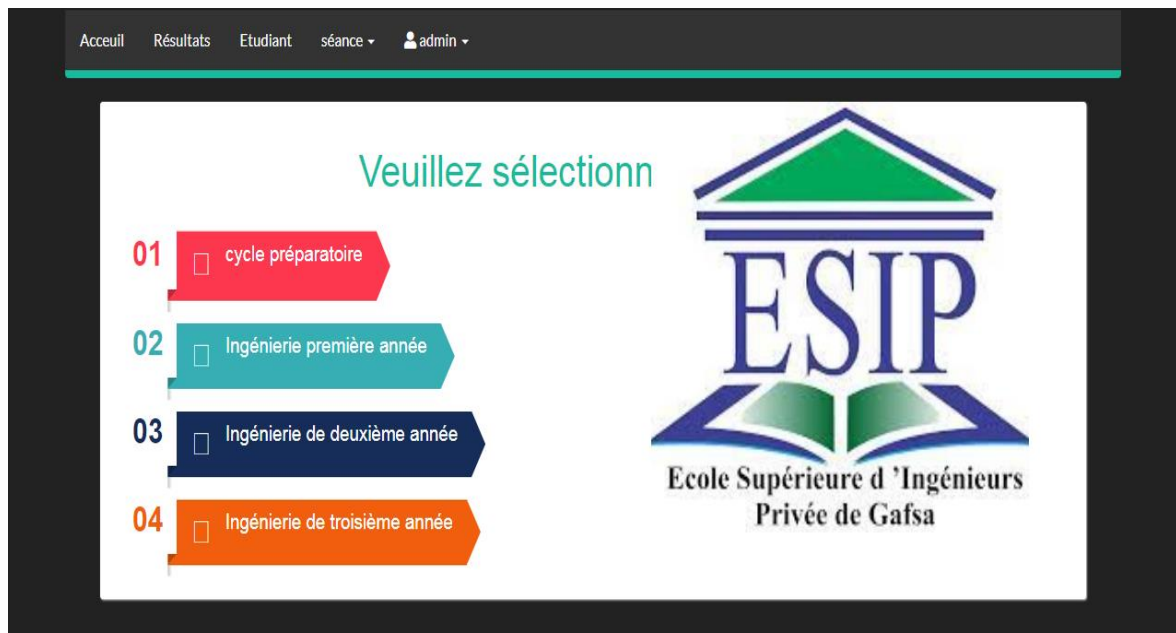


FIGURE3. 9: GESTION RESULTAT



### 3.1.2. Liste des absences

Cette interface représente Liste des Absences où l'enseignant peut ajouter, modifié, supprimer une absence.

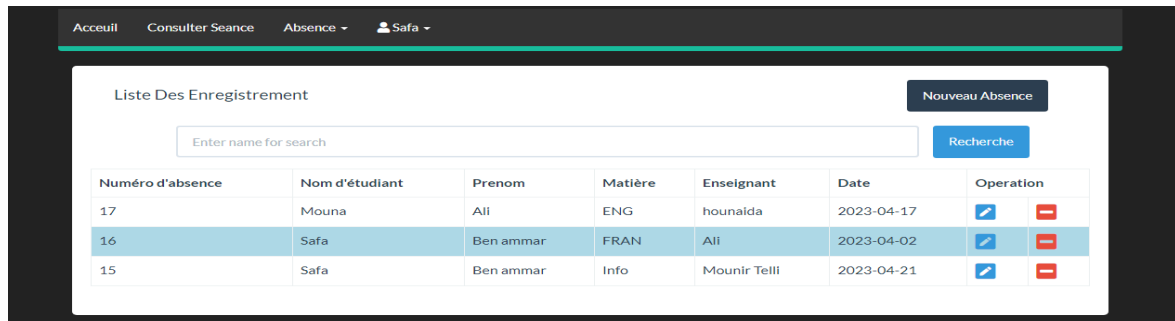


FIGURE3. 10: FENETRE LISTE DES ABSENCES COTE ENSEIGNANT

### 3.1.3. Fenêtre « gérer les Séances »

Cette interface représente les sous menus où l'administrateur peut ajouter nouvelle Séances.

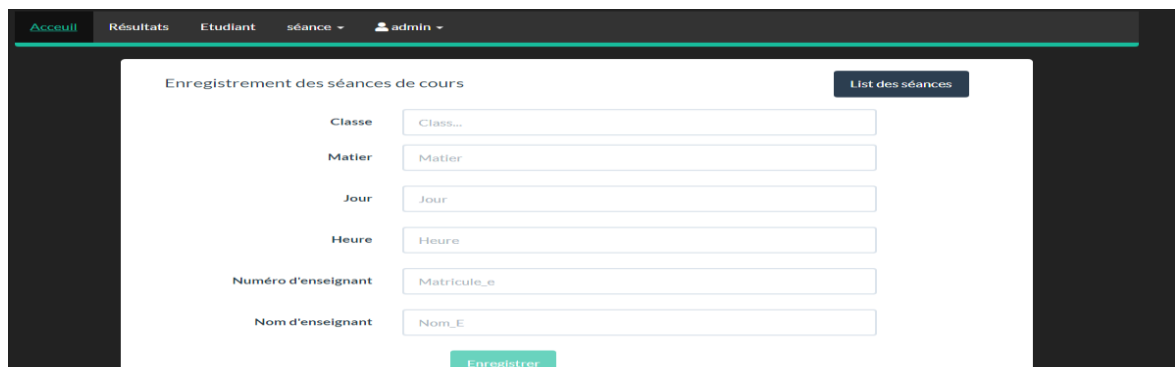


FIGURE3. 11:FENETRE « GERER LES SEANCE »

- ✓ A partir de cette fenêtre l'administrateur peut remplir les champs (Classe, Matière, Jour, Heure Numéro d'enseignant leur nom), ensuite, il doit cliquer sur le bouton enregistrer pour valider l'ajout.
- ✓ Apre savoir valider l'ajout, un message de validation sera affiché.

## 3.2. Conclusion

Dans ce chapitre, nous avons passé de la spécification de l'environnement matériel et logiciel à la présentation de quelques interfaces graphiques de notre application, qui expliquent les fonctionnalités fondamentales de notre travail.

## *Conclusion & perspectives*

L'objectif de notre projet de fin d'études était de concevoir et implémenter une application web de gestion scolaire destiné à l'Ecole Supérieure d'Ingénieurs Privée de Gafsa. Cette phase pratique de notre formation consacrée à l'analyse, à la conception et à la programmation nous a conduits à partir des problèmes constatés de concevoir une application. Certes, il existe une solution mais elle peut être amélioré davantage pour mieux répondre aux attentes. Nous avons utilisé le langage UML pour modéliser le système, le Visual Paradigme pour la conception en suivant les différentes étapes, MySQL comme SGBD et le langage PHP pour implémenter pour le développement de notre application. L'ensemble de ces outils et langage choisis nous ont permis de venir au bout de ce projet notamment le langage PHP nous facilitait du fait qu'il soit un langage flexible, l'application sera opérationnelle.

Nous avons essayé, dans ce rapport de présenter tout ce qui s'avère indispensable pour décrire clairement toutes étapes du projet : « Etat de l'art », «Etude conceptuelle » et « réalisation ».

La réalisation de ce projet, m'a permis d'acquérir et d'élargir mes connaissances dans les domaines des logiciels dont nous n'avions d'ambler rencontrer lors de notre formation. Nous avons aussi pu améliorer mes aptitudes en conception. Dans ce projet, les plus importants des points demandés ont été réalisés. Cependant, les quelques points qui restent à développeront seront effectuées à savoir :

- Gestion d'emplois de temps
- La Gestion des Bulletins
- La Gestion des Paiements

Plusieurs améliorations restent envisageables dans ce travail, ces améliorations touchent essentiellement l'extensibilité de notre application pour prendre en charge d'autres fonctionnalités.

Ajoutons que nous comptons développer la version mobile de cette application en mode consultation.

## LISTE DES RÉFÉRENCES

- [1] site officiel <https://www.erplain.com/fr>
- [2] site officiel <https://www.openconcerto.org/fr/index.html>
- [3] site officiel <https://www.gestion-stocks.com/fr/>
- [4] site officiel <https://openflex.cloud/fr/>
- [5] <http://orm.bdpedia.fr/mvc.html>

## **RÉSUMÉ**

Ce travail s'inscrit dans le cadre du projet de fin d'études en vue de l'obtention d'un diplôme de licence en Réseaux Informatique à Institut Supérieur des Sciences Appliquées et de Technologies de Gafsa. Notre travail est d'assurer une partie importante et indispensable pour la mise en place d'une application web pour la gestion de scolarité de l'école supérieur d'ingénieur privé de Gafsa. Notre problématique de départ était de répondre au mieux aux attentes des différents utilisateurs : l'étudiant, l'enseignant, l'administration. Par ailleurs, il fallait faciliter l'accès à l'information et éviter les traitements manuels source d'erreurs importantes. Et pour cela on a essayé d'apporter des solutions à cette problématique comme : la consultation des notes, l'enregistrement et la modification des notes, l'édition des absences .... La réalisation de ce travail nous a permis d'acquérir de nouvelles connaissances sur les technologies web. La conception du présent projet est accomplie en se fondant sur UML. Concernant la mise en œuvre, elle est effectuée grâce à divers outils de développement tel que Notepad++, et en se servant du système de gestion de base de données MySQL.

**Mots clés :** gestion de scolarité, ESIP, MySQL et PHP.

## **ABSTRACT**

This work is part of the end-of-studies project to obtain a bachelor's degree in computer networks at the Higher Institute of Applied Sciences and Technologies of Gafsa. Our work is to ensure an important and essential part for the implementation of a web application for the management of schooling of the higher school of private engineer of Gafsa. Our initial problem was to best meet the expectations of the different users: the student, the teacher, the administration. In addition, it was necessary to facilitate access to information and avoid manual processing, a source of major errors. And for this we have tried to provide solutions to this problem such as: consulting notes, recording, and modifying notes, editing absences.... The realization of this work allowed us to acquire new knowledge on web technologies. The design of this project is accomplished based on UML. Regarding the implementation, it is carried out using various development tools such as Notepad++, and using the MySQL database management system.

**Mots clés :** Education, ESIP, MySQL and PHP.