



Université Abderrahmane Mira de Béjaïa

Faculté des sciences Exactes

Département D'Informatique

Mémoire de Fin de Cycle

En vue de l'obtention du diplôme de Master Professionnel en Génie Logiciel

Thème

**Conception & Réalisation d'une application web pour
la gestion des emplois du temps**

Cas d'étude département d'informatique

Réalisé par :

MEBARKI Cilia

BRAHIMI Meriem

Soutenu devant le jury composé de :

Présidente : KESSIRA Dalila. MAA

Examinatrice : CHERIFI Feriel. MCB

Encadrante : SAAD Narimane

Promotion2022/2023

Remerciements

Tout d'abord, nous tenons à remercier Dieu, de nous avoir donné la santé, la volonté et la patience à terme notre formation de master et pouvoir réaliser ce travail.

*Nous tenons à remercier notre promotrice **Madame SAAD NARIMANE** pour la confiance et l'intérêt qu'elle nous a témoigné durant toute la période de travail, pour son aide, ses précieux conseils, sa patience. Nous sommes très honorées de la participation de **Madame KESSIRA DALILA** et **Madame CHERIFI FERIEL** dans notre jury de soutenance. On les remercie vivement pour avoir accepté de juger ce travail.*

Nous tenons à remercier tous nos enseignantes qu'on a eu le plaisir de côtoyer pendant la période de notre formation à l'université de Bejaia.

Un grand merci à nos familles pour leurs soutiens aussi bien moral que financier et pour leurs sacrifices.

Dédicace

Je dédie ce travail

A mes très chers parents qui mon toujours

soutenu et cru en moi,

A mon cher fiancé et sa famille

A mes chères grands-parents

A mes chers frères et chères sœurs

A mon binôme Cilia et sa famille

A tous mes enseignants

Meriem

Dédicace

Je dédie ce travail

A mes très chers parents qui mon toujours

soutenu et cru en moi,

A mes chères grands-parents

A mes chers frères et chères sœurs

A mon binôme Meriem et sa famille

A tous mes enseignants

Cifia

Table des matières

Liste des abréviations	I
Liste des figures	II
Liste des tableaux	III
Introduction générale	1
Chapitre I : Etude de l'existant.....	3
Introduction.....	3
I.1. Présentation du département d'informatique	3
I.2. Mission du département.....	4
I.3. Problématique	4
I.4. Etude du système pédagogique	4
I.4.1. L'activité pédagogique.....	5
I.4.2. Les ressources	5
I.5. Description des problèmes à résoudre	6
I.5.1. Objectifs	7
I.5.2. Les contraintes.....	7
Conclusion	8
Chapitre II : Planification d'horaire de travail et processus de modélisation	9
Introduction.....	9
II.1. Problématique de la planification d'horaires de travail	9
II.2. Définition de planification et planning	9
II.2.1. Planification.....	9
II.2.2. Planning.....	9
II.3. Utilité d'un planning.....	10
II.4. Elaboration d'un planning	10
II.5. Comment est évalué un planning ?.....	10
II.6. Processus utilisés.....	11
II.6.1 UML	11
II.6.2 Processus Unifié	12
II.6.2.1 Phases de UP.....	13
II.6.2.2 Activités du processus unifié	13
II.7. Avantage UML & UP	15

Conclusion	15
Chapitre III : : Analyse et conception	16
Introduction	16
III.1. Spécification des besoins	16
III.1.1. Besoins fonctionnels	16
III.1.2. Besoins non fonctionnels	17
III.2. Analyse des besoins	17
III.2.1. Identification des acteurs	17
III.2.2. Diagramme de contexte	18
III.3. Diagramme de cas d'utilisation	19
III.4. Description des cas d'utilisations	20
III.5. Conception	28
III.5.1. Diagramme de séquence	28
III.5.1.1. Diagramme de séquence « Authentification »	28
III.5.1.2. Diagramme de séquence « Ajouter formation »	29
III.5.1.3. Diagramme de séquence « modifier un étudiant »	30
III.5.1.4. Diagramme de séquence « supprimer un enseignant »	31
III.5.1.5. Diagramme de séquence « créer un EDT »	32
III.5.1.6. Diagramme de séquence « consulter et imprimer l'EDT »	33
III.5.2. Diagramme de classe	34
III.5.2.1. Dictionnaire de données	34
III.5.3. Modèle relationnel	37
III.5.3.1. Les règles de passage au modèle relationnel à partir d'un diagramme de classes	37
III.5.3.2. Le passage au modèle relationnel	37
Conclusion	38
Chapitre IV : Implémentation et résultats	39
Introduction	39
IV.1. Application web	39
IV.2. Environnements de développement	39
IV.2.1. Visual studio code	39
IV.2.2. StarUML	40
IV.3. Langages et outils du développement	40
IV.3.1. HTML	40

IV.3.2. CSS	40
IV.3.3. Python	41
IV.3.4. Django	41
IV.3.5. JavaScript	41
IV.4. Les fonctionnalités de l'application.....	42
IV.4.1. Connexion	42
IV.4.2. Espace gestionnaire.....	43
IV.4.2.1. Tableau de bord	43
IV.4.2.2. Ajouter formation	43
IV.4.2.3. Modifier un enseignant	44
IV.4.2.4. Affecter module a l'enseignant	45
IV.4.2.5. Recherche d'un enseignant	46
IV.4.2.6. Supprimer un enseignant.....	47
IV.4.2.7. Génération d'un emploi du temps	47
IV.4.3. Espace enseignant	48
IV.4.3.1. Consultation d'un emploi du temps.....	48
IV.4.4. Espace étudiant	48
IV.4.4.1. Consultation d'un emploi du temps.....	48
IV.4.4.2. Imprimer un emploi du temps	49
Conclusion	49
Conclusion générale.....	50
Bibliographie	51

Liste des abréviations

CSS:Cascading Style Sheets

DEUA : Diplôme d'Etudes Universitaires Appliqué

EDT : Emploi de temps

FSE:La faculté des Sciences Exactes

HTML: HyperText Markup Language

IDE: Integrated Development Environment

LMD: (Licence, Master et Doctorat)

MI: Mathématique Informatique

OS : Operating System

RN : Recrutement nationale

SM: Séance de matière

TD: Travaux dirigées

TP: Travaux pratiques

UML: Unified Modeling Language

UP: Unified Process

UAMB: Université Abberahmane Mira de Bejaia

XML: eXtensible Markup Language

Liste des figures

FIGURE 1: DEROULEMENT DU PROCESSUS UNIFIE	12
FIGURE 2: DESCRIPTION DU PROCESSUS UNIFIE	14
FIGURE 3: DIAGRAMME DE CONTEXTE	18
FIGURE 4: DIAGRAMME DE CAS D'UTILISATION « GLOBALE ».....	19
FIGURE 5: DIAGRAMME DE CAS D'UTILISATION « AUTHENTIFICATION »	20
FIGURE 6: DIAGRAMME DE CAS D'UTILISATION « GERER LES ETUDIANTS »	21
FIGURE 7: DIAGRAMME DE CAS D'UTILISATION « GERER LES ENSEIGNANTS »	22
FIGURE 8: DIAGRAMME DE CAS D'UTILISATION « CONSULTER UN EDT »	24
FIGURE 9: DIAGRAMME DE CAS D'UTILISATION « GERER LES SECTIONS »	25
FIGURE 10 : DIAGRAMME DE CAS D'UTILISATION « GERER LES FORMATIONS »	26
FIGURE 11: DIAGRAMME DE SEQUENCE « AUTHENTIFICATION »	28
FIGURE 12: DIAGRAMME DE SEQUENCE « AJOUTER FORMATION »	29
FIGURE 13 : DIAGRAMME DE SEQUENCE « MODIFIER ETUDIANT ».....	30
FIGURE 14: DIAGRAMME DE SEQUENCE « SUPPRIMER ENSEIGNANT »	31
FIGURE 15: DIAGRAMME DE SEQUENCE « CREER UN EDT »	32
FIGURE 16: DIAGRAMME DE SEQUENCE « CONSULTER ET IMPRIMER L'EDT ».....	33
FIGURE 17 : DIAGRAMME DE CLASSE	36
FIGURE 18 : INTERFACE DE CONNEXION	42
FIGURE 19 : TABLEAU DE BORD	43
FIGURE 20 : AJOUTER FORMATION	44
FIGURE 21 : MODIFIER UN ENSEIGNANT	44
FIGURE 22 : LA LISTE DES ENSEIGNANTS	45
FIGURE 23 : LA LISTE DES MODULES	45
FIGURE 24 : AFFECTER MODULE A L'ENSEIGNANT	46
FIGURE 25 : LA RECHERCHE D'UN ENSEIGNANT.....	46
FIGURE 26 : SUPPRESSION D'UN ENSEIGNANT.....	47
FIGURE 27 : CREER UN EMPLOI DU TEMPS	47
FIGURE 28 : INTERFACE DE CONSULTATION D'UN EDT	48
FIGURE 29 : CONSULTER LES EDT	48
FIGURE 30 : IMPRIMER UN EMPLOI DU TEMPS	49

Liste des tableaux

TABLEAU 1 : STRUCTURE DE LA TABLE UTILISATEUR.....	34
TABLEAU 2: STRUCTURE DE LA TABLE ENSEIGNANT.....	34
TABLEAU 3: STRUCTURE DE LA TABLE FORMATION.....	34
TABLEAU 4 : STRUCTURE DE LA TABLE GROUPE.....	35
TABLEAU 5 : STRUCTURE DE LA TABLE SEMESTRE.....	35
TABLEAU 6 : STRUCTURE DE LA TABLE SECTION.....	35
TABLEAU 7 : STRUCTURE DE LA TABLE MODULES.....	35
TABLEAU 8 : STRUCTURE DE LA TABLE SALLE.....	36
TABLEAU 9 : STRUCTURE DE LA TABLE SEANCE.....	36

Introduction générale

Au cours des dernières années, la révolution de l'informatique a balayé tous les pays du monde, cela inclus notre pays l'Algérie qui a connu un fort développement incontestable dans ce domaine, et cette dernière a pris son véritable envol avec l'apparition des différents réseaux sociaux et applications.

En effet, l'informatique est devenue l'outil indispensable de gestion, parlons sur la gestion d'emploi du temps dans une université pour garantir le bon déroulement des programmes d'études et la réussite des étudiants.

Ainsi, cette gestion consiste à organiser et à coordonner les horaires des cours, des examens, des travaux pratiques, des conférences et des autres activités académiques et extra-académiques. C'est un processus complexe qui nécessite une planification minutieuse et une communication efficace entre les différentes parties prenantes, notamment les enseignants, les étudiants, les responsables administratifs et les services de soutien.

La difficulté de cette tâche fastidieuse et répétitive, qui fait généralement intervenir de nombreux éléments d'information, est liée à la nature des contraintes qu'il s'agit de satisfaire. A celle-ci s'ajoutent des caractéristiques bien particulières de l'institution ou l'organisation concernée. Parce qu'il y a plusieurs modèles du problème, des contraintes qui changent, et des caractéristiques particulières pour chaque institution ou organisation, il devient de plus en plus difficile de trouver une solution générale pour les problèmes d'emploi du temps, et c'est pour cela que ce domaine a besoin de plus en plus de la recherche.

Dans ce contexte, le problème connu sous le nom d'emploi du temps consiste à organiser des rencontres entre des enseignants et des groupes d'étudiants en définissant un lieu et une heure de rencontre tout en satisfaisant un ensemble de contraintes. Il existe différentes versions de ce dernier mais la plupart peuvent être classifié en deux grandes catégories qui sont : emploi du temps pour examen et emploi du temps universitaire. [1]

Dans ce cadre s'inscrit notre projet de fin d'étude qui consiste à réaliser une application web pour la gestion des emplois du temps à l'université « A/Mira de Bejaia » (UAMB).

Pour une bonne élaboration nous avons structuré notre mémoire en quatre chapitres :

Chapitre I «Etude de l'existant » nous avons présenté une vue globale du département informatique de l'université UAMB ainsi que son étude du système pédagogique.

Le Chapitre II, intitulé « Planification d'horaire de travail et processus de modélisation » explore divers aspects de la planification, notamment l'élaboration, l'évaluation et la mise en œuvre de plannings. Il examine également les processus de modélisation, tels qu'Unified Modeling Language (UML) et du Processus Unifié (UP), avec leurs avantages.

Chapitre III « Analyse et conception » est dédié à la présentation de modélisation UML et la démarche de développement UP. Les acteurs qui interagissent dans le système. Nous avons également décrit les besoins de chaque acteur sous forme de diagramme de cas d'utilisation. Chaque cas d'utilisation a un diagramme de séquence. Le passage au modèle relationnel est extrait par le diagramme de classe, dont l'objectif est d'implémenter notre base de données.

Le Chapitre IV est la partie dédiée à l'implémentation et à la réalisation de notre application, où nous présentons l'environnement de développement, la structure de notre application, ainsi que quelques-unes de ses interfaces.

Nous concluons ce rapport par une conclusion générale résumant les principales fonctionnalités réalisées et proposant quelques perspectives en vue de terminer (d'élargir et d'améliorer) ce travail.

Chapitre I : Etude de l'existant

Introduction

La présentation et l'analyse de l'organisme d'accueil constituent la phase de base d'une étude de système. Il s'agit de comprendre les difficultés qui ont été rencontrées dans l'organisation pour faciliter les étapes à venir. Par conséquent, l'organisme d'accueil doit nous fournir toutes les informations nécessaires pour établir une bonne conception et proposer des solutions efficaces. Pour commencer ce chapitre, nous présenterons dans un premier temps le département informatique de l'université "A/Mira de Bejaia" de manière succincte, puis nous nous concentrerons sur l'étude du système pédagogique de ce département qui sera notre sujet d'analyse et en exposant la problématique et les objectifs de notre projet, dans le but d'assurer une meilleure gestion.

I.1. Présentation du département d'informatique

La faculté des Sciences Exactes (FSE), de l'université A. Mira de Bejaia se compose de cinq départements (mathématiques, informatique, physique et SM, chimie et recherche opérationnelle). Le service pédagogique de département d'informatique sera le terrain de notre étude.

Depuis sa création à la fin de l'année 2002, le département d'informatique de l'université A. Mira de Bejaia a offert trois types de formations différentes.

Le premier était le système classique à deux cycles de graduation. En plus de cela il y avait un programme de cycle court (2003-2006) menant au diplôme de DEUA (Diplôme d'Etudes Universitaires Appliqué) en informatique, ainsi qu'un programme de cycle long (2003-2012) menant au diplôme d'ingénieur d'état en informatique [2].

Le département d'informatique a adopté la formation de système LMD (Licence, Master et Doctorat) depuis l'année 2003.

I.2. Mission du département

Le département a pour mission d'assurer un suivi pédagogique des cycles de graduation et de post-graduation, ainsi que de gérer la scolarité des étudiants notamment leurs inscriptions, évaluation, présence aux enseignements, absences et sanction. Il est également chargé de la gestion des enseignants incluant les matières enseignées, l'affectation des modules, le volume horaire, les emplois du temps, le planning des examens, la gestion des soutenances, la gestion des absences, la saisie des notes et les délibérations [2].

Les niveaux d'étude ouverts au sein du département sont :

- Licence informatique (MI, RN)
- Master professionnelle Génie Logiciel
- Master professionnelle Administration et sécurité des réseaux
- Master professionnelle Intelligence Artificielle
- Master recherche Réseau et sécurité
- Master recherche système information avancé
- Doctorat

I.3. Problématique

Notre département doit résoudre la problématique de la planification des emplois du temps. Pour y parvenir, nous devons organiser les étudiants en formations, sections et groupes en fonction de leur niveau d'études. Chaque groupe a un emploi du temps hebdomadaire, comprenant des séances de cours, TP ou TD pour chaque module qui est assuré par un enseignant. Au début de chaque semestre, nous devons répartir ces locaux pour chaque groupe.

I.4. Etude du système pédagogique

Ce système universitaire n'a pas de caractéristique pédagogique spécifique et présente de nombreuses similitudes avec le fonctionnement des autres universités. L'UAMB propose des formations variées dans le cadre du système LMD, et les étudiants s'inscrivent au début de chaque année universitaire, généralement en septembre, mais cela peut varier selon les formations. Le programme pédagogique de chaque formation est connu à l'avance, avec des informations sur les matières à suivre, les horaires et la répartition des cours, TD et TP. En

fonction des besoins pédagogiques et des ressources disponibles, chaque formation est divisée en sections, elles-mêmes pouvant être subdivisées en groupes [3].

I.4.1. L'activité pédagogique

L'activité pédagogique est un ensemble d'actions et de pratiques qui visent à faciliter l'apprentissage des étudiants. La modélisation de l'activité pédagogique utilise trois entités : les locaux, les enseignements et les modules. Les ressources de l'enseignement comprennent généralement les enseignants, les groupes d'étudiants et les matériels nécessaires. Les matériels sont les ressources qui seront utilisées lors des séances d'enseignement. Il est possible qu'un enseignement soit assuré simultanément par plusieurs enseignants, comme c'est le cas pour certains travaux pratiques d'informatique. Les modules sont des ensembles d'enseignements [3].

I.4.2. Les ressources

Les ressources sont une composante clé de la planification de l'emploi du temps universitaire. Les enseignants et les administrateurs universitaires doivent prendre en compte les ressources disponibles lors de la création de l'emploi du temps pour garantir une organisation efficace des enseignements et un accès adéquat des étudiants aux ressources nécessaires. Les ressources incluent les salles de cours, les enseignants et les groupes d'étudiants.

Chaque ressource est identifiée par un code unique, un calendrier précisant les disponibilités et les indisponibilités, et une description. En outre, chaque type de ressource possède des caractéristiques spécifiques, qui permettent d'ajouter facilement de nouveaux types de ressources à l'outil de planification. Dans la suite de l'étude, nous décrirons en détail les caractéristiques spécifiques des ressources considérées.

- Les ressources de type « local »

Il s'agit des espaces physiques dans lesquels les activités pédagogiques sont menées, telles que les cours magistraux, les travaux dirigés, les travaux pratiques, les examens, etc. Les salles peuvent être de différentes tailles et équipées de différents matériels en fonction des besoins pédagogiques (tableau, vidéoprojecteur, ordinateurs, etc.).

- Les ressources de type « enseignant »

Correspondent à des personnes qui ont la capacité de dispenser des enseignements. Chaque individu possédant cette ressource est identifié par son nom et son prénom, son grade, son lieu d'affectation administrative ainsi que sa spécialité. Cette dernière indique les matières qu'il peut enseigner.

- Les ressources de type « groupe »

En général, les étudiants d'une filière sont subdivisés en sections, qui sont ensuite divisées en plusieurs groupes.

I.5. Description des problèmes à résoudre

Dans un établissement éducatif, l'ensemble des étudiants appartenant à une filière et étant au même niveau d'étude est appelé « Promotion », cette dernière est divisée en sections qui peuvent être à leur tour divisées en groupes. On fréquente des cas où une promotion représente la section et le groupe au même temps ; comme celle de fin de cycle.

Les étudiants sont censés avoir un ensemble d'enseignements, déterminés en cours, TD et TP. Chacun de ces enseignements s'étend sur une durée de temps limitée généralement en une heure et demie. Un cours est dispensé à une section ou à un groupe quand ils sont confondus, alors qu'un TD et un TP sont destinés seulement à un groupe.

Résoudre le problème de l'emploi du temps, revient à affecter aux séances la durée et le local dont le type et la capacité qui leurs convient, pour éviter les conflits sur les enseignants, les étudiants et les locaux afin d'assurer le module concerné par l'enseignement.

Le département d'informatique regroupe différentes formations qui ont une durée qui varie entre trois ans (licence) et cinq ans (master).

Le programme pédagogique d'emplois du temps précise les modules à suivre, leurs volumes horaires et quelques informations pédagogiques (répartition des cours, travaux dirigés, travaux pratiques etc...). Les enseignants n'ont pas le même volume horaire, cela dépend le niveau de chacun (professeur, maître-assistant, maître de conférences, vacataire...), en prenant un tel exemple : Le plafond d'heures pour un professeur ne dépasse pas six heures par semaine

tandis qu'un vacataire possède plus de temps, pour qu'il arrive à douze heures par semaine. Le respect de volume horaire et La prise en compte de la disponibilité des enseignants sont obligatoires. En cas d'absence, le déroulement des enseignements doit être effectué plus normal, et les séances doivent être rattrapé. Un enseignant doit donc connaître précisément la disponibilité des ressources de sa séance (salle, enseignant, groupe).

Selon les besoins pédagogiques et les conditions physiques des ressources, chaque formation est structurée en niveau, en sections, et groupes.

Les données du problème à résoudre se résument en ce qui suit :

- Un ensemble de créneaux horaires étalés sur une semaine de cinq jours, du samedi au mercredi ou du dimanche au jeudi. La durée d'une période est 1h 30 min.
- Un ensemble de niveaux ou groupes d'étudiants.
- Un ensemble de cours, TD ou TP à programmer dans la semaine.
- Un ensemble de locaux (salles, amphis...).

I.5.1. Objectifs

L'objectif de notre projet est de mettre en place une application web destinée à la gestion des emplois du temps du département informatique. Cette application sera conçue pour réduire l'utilisation du papier, améliorer l'accessibilité rapide à l'information et sécuriser l'accès aux ces informations par une authentification.

I.5.2. Les contraintes

Pour la bonne réalisation de notre système, nous nous sommes basés sur les contraintes fondamentales pour la création d'un emploi du temps.

I.5.2.1. Les contraintes physiques

Ces contraintes ne doivent pas être violées sinon cela conduirait à des situations conflictuelles. Voici-quelques exemples :

- Un enseignant ne peut pas être affecté à deux séances différentes au même créneau.
- Une salle /amphi ne peut pas accueillir deux séances différentes au même temps.
- Un module doit respecter le nombre des séances hebdomadaires.

- Respecter la charge journalière d'un enseignant.
- Si un TD ou un TP est affecté à un groupe donné, on ne peut pas affecter un cours à la section à laquelle appartient ce groupe et inversement.
- Un TP n'aura pas lieu dans un amphithéâtre ou dans une salle de TD.
- Un TD ou un cours n'aura pas lieu dans une salle de TP.
- Capacité des locaux doit être supérieure ou égale à l'effectif du groupe/section.

I.5.2.2. Les contraintes pédagogiques

Les contraintes pédagogiques peuvent éventuellement être violées, contrairement aux contraintes physiques. Ces contraintes sont souvent utilisées pour exprimer ce que doit être un " bon " planning. Quelques exemples de ces contraintes incluent :

- Prévenir une surcharge de certains jours tandis que d'autres jours sont moins chargés.
- Libérer quelques après-midis pour les enseignants.
- Minimiser les déplacements des étudiants dans l'établissement.
- Pas plus de trois séances consécutives pour le même enseignant.
- S'assurer que tous les enseignements prévus soient affectés.
- Eviter de programmer des activités qui se terminent après 18h00.

Conclusion

Notre projet consiste en la conception et la mise en place d'un système de gestion automatisée des emplois du temps pour le département d'informatique. Nous avons commencé par réaliser une analyse de l'existant, au cours de laquelle nous avons collecté toutes les informations nécessaires à la réussite de notre projet. Le prochain chapitre portera sur la gestion et la planification des horaires de travail et les solutions proposées pour résoudre ces problématiques et avoir une gestion optimale et un bon planning.

Chapitre II : Planification d'horaire de travail et processus de modélisation

Introduction

Dans ce chapitre, nous plongerons au cœur de la planification d'horaires de travail, en explorant la problématique qui l'entoure et en définissant les concepts de planification et de planning. Nous examinerons également l'importance cruciale d'un planning bien conçu dans divers domaines professionnels et nous nous pencherons sur le processus de création et d'évaluation de ces plannings. Enfin, nous mettrons en avant le rôle clé de l'Unified Modeling Language (UML) et du Processus Unifié (UP) dans l'optimisation de la gestion des horaires de travail.

II.1. Problématique de la planification d'horaires de travail

La planification d'horaires de travail est une problématique importante dans de nombreux Champs professionnels, car elle peut avoir un impact significatif sur la productivité, la satisfaction des employés et la qualité du service offert. Offert Cependant, la planification d'horaires de travail peut être complexe, en particulier dans les secteurs où la demande fluctue considérablement, tels que les soins de santé, le commerce de détail, les transports, l'hôtellerie et la restauration.

II.2. Définition de planification et planning

Nous allons définir les terminologies suivantes

II.2.1. Planification

La planification peut être définie comme le processus d'élaboration d'un plan d'action pour atteindre un objectif spécifique. Cela implique d'analyser les objectifs à long terme, d'identifier les tâches nécessaires pour les atteindre, de définir les priorités, de prévoir les ressources nécessaires et de créer un calendrier pour l'ensemble du processus.

II.2.2. Planning

Un planning est un document qui permet de planifier et d'organiser les activités d'une entreprise, une école, un hôpital, une administration, etc. Le planning peut inclure les horaires de travail des employés, les horaires de cours pour les étudiants, les plages de disponibilité pour les

consultations médicales ou les rendez-vous administratifs, les dates de réunions, les périodes de vacances ou de congés, les échéances de projets, etc. Le planning de l'établissement peut être élaboré par différents responsables selon l'organisation. Par exemple, un responsable des ressources humaines peut élaborer le planning des employés, un responsable pédagogique peut élaborer le planning des cours, un responsable médical peut élaborer le planning des consultations [1].

II.3. Utilité d'un planning

Un planning peut être très utile pour plusieurs raisons. Voici quelque avantage spécifique de l'utilisation d'un planning pour un établissement.

- ✓ Planification des activités : Un planning peut aider à organiser les activités quotidiennes hebdomadaires ou mensuelles de l'établissement. Il peut également aider à coordonner les tâches entre les différents départements, en évitant les conflits d'horaires et en permettant une meilleure collaboration.
- ✓ Gestion de ressources : Un planning peut aider à gérer les ressources de l'établissement, notamment les personnels, les équipements, les salles et les budgets.

II.4. Elaboration d'un planning

Généralement cette tâche peut être centralisée à des cadres de l'entreprise appelés planificateurs, ces derniers doit prendre la décision qui convient aux préférences des différents acteurs car son expérience de la tâche fait de lui un interlocuteur privilégié pour évaluer rapidement et effectuer des jugements de l'orientation à donner à la recherche de solutions de meilleure qualité afin d'aboutir à un choix approprié [4].

II.5. Comment est évalué un planning ?

L'évaluation d'un planning peut varier en fonction de l'objectif du planning.

Cependant, voici quelques critères clés qui peuvent être utilisés pour évaluer un planning :

- ✓ La cohérence avec le programme d'études : Le planning doit être cohérent avec le programme d'étude et les objectifs d'apprentissage.

- ✓ L'efficacité de l'utilisation du temps : Le planning doit être structuré de manière à maximiser l'utilisation efficace du temps disponible pour les activités d'enseignement et d'apprentissage.
- ✓ La faisabilité : Le planning doit être réalisable dans les limites des ressources disponibles, tels que le temps, le personnel, et le budget.
- ✓ La qualité : le planning doit être détaillé, précis, et structuré de manière à permettre une exécution efficace des activités.
- ✓ La flexibilité : Le planning doit être suffisamment flexible pour permettre l'adaptation aux imprévus, aux changements dans les priorités de l'établissement, et besoins des parties prenantes.

Enfin, une fois que le planning a été évalué et validé, il peut être communiqué aux enseignants et aux étudiants afin de permettre une organisation efficace de leur emploi du temps.

II.6. Processus utilisés

Selon la nature de notre projet, nous avons choisi UP et UML pour sa réalisation.

II.6.1 Unified Modeling Language (UML)

UML se définit comme un langage de modélisation graphique et textuel destiné à faciliter la conception des documents nécessaires au développement d'un logiciel, décrire des besoins, spécifier des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.



Pour notre projet, nous présentons trois diagrammes d'UML [5].

• Diagramme de cas d'utilisation :

Le diagramme de cas d'utilisation est un outil de modélisation visuelle utilisé dans l'ingénierie logicielle pour représenter les interactions entre les utilisateurs (acteurs) et le système, afin de décrire les fonctionnalités et les exigences du système logiciel [6].

• Diagramme de séquence :

Représente des échanges de messages entre objets selon un point de vue temporel, il permet aussi de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets [7].

- **Diagramme de classe :**

Le diagramme de classe est un outil de modélisation visuelle utilisé dans le domaine de l'ingénierie logicielle pour représenter les classes, les interfaces, les attributs, les méthodes et les relations entre eux. Il est généralement utilisé pour concevoir et documenter la structure d'un système logiciel orienté objet [8].

II.6.2 Processus Unifié

UP est un guide méthodologique pour réaliser des logiciels en conseillant et pilotant l'équipe dans ses différentes activités pour réduire la complexité des projets (on sait ou on en est et ce qu'il nous reste à faire). Ce processus permet d'éclaircir qui fait quoi, et permet à chacun de savoir quelle est sa place dans le processus de production du logiciel. Le processus unifié se caractérise par une démarche itérative et incrémentale, pilotée par les cas d'utilisation, et centrée sur l'architecture et les modèles UML [9].

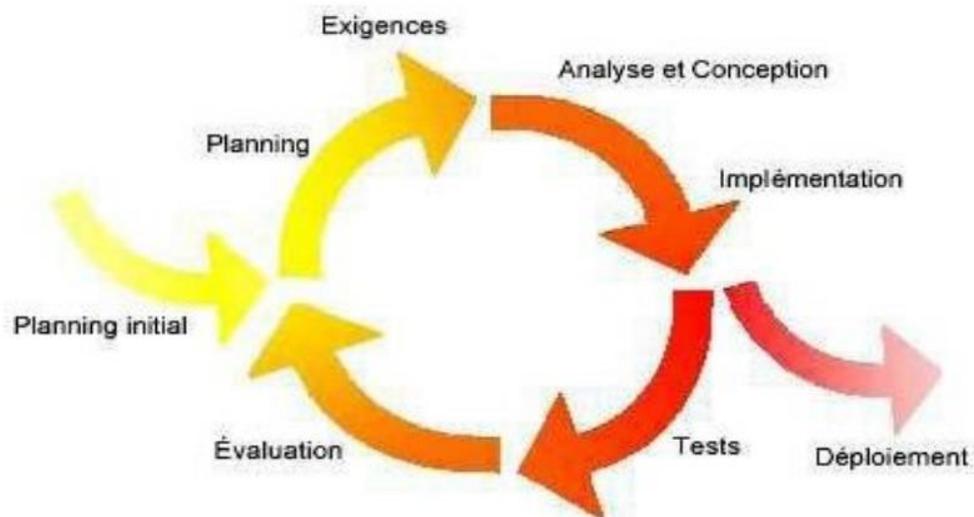


Figure 1: Déroulement du processus unifié [10].

II.6.2.1 Phases de Processus Unifié

La méthode UP se base sur quatre phases [10]:

- **Analyse des besoins** : C'est la première phase du processus unifié. Il s'agit de tracer ce qui doit figurer à l'intérieur du système et ce qui doit rester à l'extérieur (délimiter la portée du système et le bien comprendre), identifier les acteurs, dégager les fonctionnalités initiales, établir une architecture candidate (on doit essayer de construire une architecture capable de fonctionner), il faut également identifier les risques critiques susceptibles de faire obstacles au bon déroulement du projet.
- **Elaboration** : Elle consiste à stabiliser l'architecture du système. Il s'agit alors de raffiner le modèle initial de cas d'utilisation, capturer de nouveaux besoins, analyser et concevoir la majorité des cas d'utilisation formulés, et si possible implémenter et tester les cas d'utilisation initiaux.
- **Construction** : Dans cette phase, il faut capturer tous les besoins restants parce qu'il n'est pratiquement plus possible de le faire dans la prochaine phase, continuer l'analyse, la conception et surtout l'implémentation de tous les cas d'utilisation. Enfin, fournir une version exécutable du système.
- **Transition** : C'est la dernière phase du processus UP. Elle consiste à vérifier si le système offre les services exigés par les utilisateurs, détecter les défaillances, combler les manques dans la documentation du logiciel et adapter le produit à l'environnement (mise en place et installation).

II.6.2.2 Activités du processus unifié

Chaque phase est constituée d'une succession d'activités. Les activités du processus UP sont les suivantes [10]:

1. Expression des besoins

C'est la compréhension et l'expression des besoins fonctionnels et non fonctionnels, ainsi que la livraison d'une liste comprenant les exigences du client.

2. Analyse

L'analyse est l'activité de préparation à la conception. Elle permet d'accéder à une compréhension des besoins et des exigences du client, et aux outils de réalisation en prenant en compte le choix d'architecture technique retenu pour le développement et l'exploitation système.

3. Conception

La conception permet d'acquérir une compréhension approfondie des contraintes liées au langage de programmation, à l'utilisation des composants et au système d'exploitation. Elle détermine les principales interfaces.

4. Implémentation

C'est le résultat de la conception, on implémente le système sous forme de composants (de code source, de scripts, d'exécutable et d'autres éléments du même type.). Elle a pour objectif de planifier l'intégration et de produire les classes et les sous-systèmes sous formes de codes sources.

5. Tests

Les tests permettent de vérifier les résultats de l'implémentation de toutes les exigences en testant la construction, et de s'assurer de la bonne intégration de tous les composants dans le logiciel.

La figure suivante décrit les différentes activités et les phases d'UP :

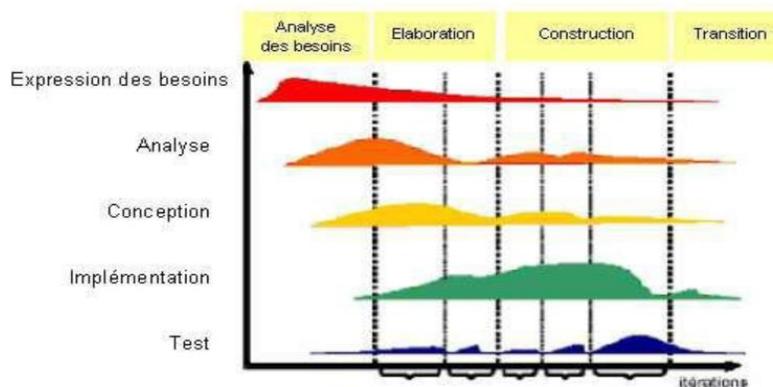


Figure 2:Description du processus unifié [10]

II.7. Avantage UML & UP

UML est un langage de modélisation qui permet de définir clairement les besoins des clients, de généraliser les aspects de conception et de l'architecture, de structurer de manière cohérente les fonctionnalités et les données et également de faciliter la maintenance.

UP est un processus itératif et incrémental qui se lit selon deux axes: vertical (enchaînement de disciplines et d'activités au sein d'une itération) et horizontal (enchaînement dynamique sur l'axe temporel de phases et d'itérations), ce qui nous permet de tester à chaque itération sans attendre la fin du projet.

Conclusion

En conclusion, ce chapitre a exploré la planification d'horaires de travail, défini son importance, examiné le processus de création de plannings, et mis en avant le rôle crucial d'UML et d'UP. Cette compréhension renforcera la qualité de l'enseignement et des activités du département informatique.

Chapitre III : Analyse et conception

Introduction

Afin de garantir une planification efficace et une exécution maîtrisée des projets de développement d'applications ou de systèmes informatiques, il est essentiel de suivre une méthode ou une approche clairement définie. Cette démarche permet d'organiser le travail de manière optimale et de s'assurer que les résultats répondent aux attentes des parties prenantes impliquées dans le projet. Il est donc crucial de mettre en place une méthodologie de développement structurée pour assurer la réussite du projet et éviter les erreurs et les retards et c'est l'objectif de ce chapitre présenté dans les points qui suivent.

III.1. Spécification des besoins

Après avoir étudié le projet, nous avons déterminé les besoins auxquels notre application doit répondre. Nous avons identifié deux types de besoins : les besoins fonctionnels, qui décrivent les fonctionnalités attendues de l'application, et les besoins non fonctionnels, qui se rapportent aux critères de performance, de sécurité, d'utilisabilité...

III.1.1. Besoins fonctionnels

Nous allons décrire ci-dessous les exigences que le système doit remplir à travers les différentes tâches suivantes :

➤ **Le gestionnaire :**

Création des emplois du temps : Le gestionnaire doit pouvoir créer les emplois du temps en fonction des contraintes de l'établissement.

L'affectation des enseignants aux modules, le gestionnaire doit attribuer à chaque module un enseignant.

Gestion des ressources : le gestionnaire doit pouvoir ajouter, modifier et supprimer toutes les ressources telles que les locaux, sections, groupes, et enseignants. Il doit pouvoir affecter les salles, enseignants, modules appropriés à chaque séance.

➤ **Les enseignants et les étudiants :**

Consultation de l'emploi du temps : L'enseignant et l'étudiant doivent pouvoir accéder aux emplois du temps pour connaître les horaires de cours et les salles de classe.

III.1.2. Besoins non fonctionnels

Une fois les besoins fonctionnels identifiés, il est important de prendre en compte les besoins non fonctionnels tout au long du processus de développement de l'application. Parmi ces besoins non fonctionnels, nous pouvons citer :

- **L'authentification** : chaque utilisateur doit s'identifier avec un login unique et un mot de passe pour accéder à l'application.
- **La rapidité de traitement** : l'application doit optimiser les traitements pour minimiser le temps de réponse et s'approcher autant que possible du temps réel.
- **La sécurité et la confidentialité** : la sécurité des données doit être garantie avec une attribution appropriée des droits d'accès.
- **La performance** : l'application doit répondre de manière optimale aux exigences des utilisateurs.
- **Ergonomie** : L'application doit offrir une interface simple et facile à utiliser.

III.2. Analyse des besoins

III.2.1. Identification des acteurs

Un acteur est la représentation concrète d'un rôle joué par des entités externes, telles que des utilisateurs, des dispositifs matériels ou d'autres systèmes, qui interagissent directement avec le système étudié [5].

Les acteurs de notre système sont :

- **L'Étudiant** : Toute personne consulte et imprime les emplois du temps.
- **L'Enseignant**: personne habilitée à donner des enseignements au département d'informatique. Ses tâches seront celle de consulter leurs emplois du temps personnel.

- **Le gestionnaire :** Le rôle du gestionnaire est le plus important dans la gestion d'emploi du temps, c'est la personne chargée d'administrer l'application web (gestion de la base de données).

III.2.2. Diagramme de contexte

Le diagramme de contexte est un outil conceptuel qui permet de visualiser les interactions entre un système et son environnement, ainsi que les liens entre eux. Il est utile pour avoir une vue d'ensemble de ces interactions et pour délimiter clairement le champ d'étude.

Pour notre cas le contexte est donné par la figure suivante :

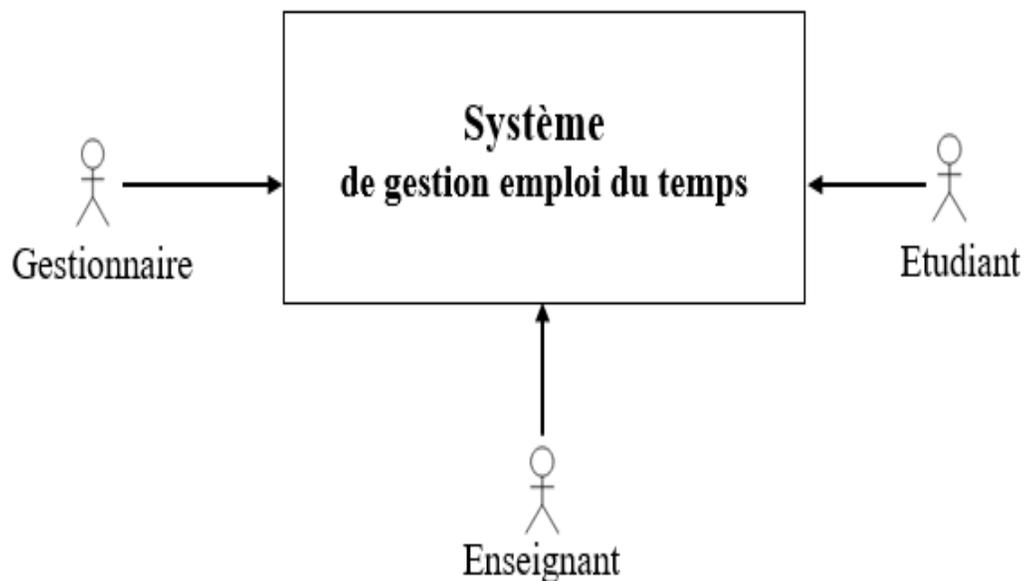


Figure 3: Diagramme de contexte

III.3. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation montre les interactions fonctionnelles entre les acteurs et le système à l'étude.

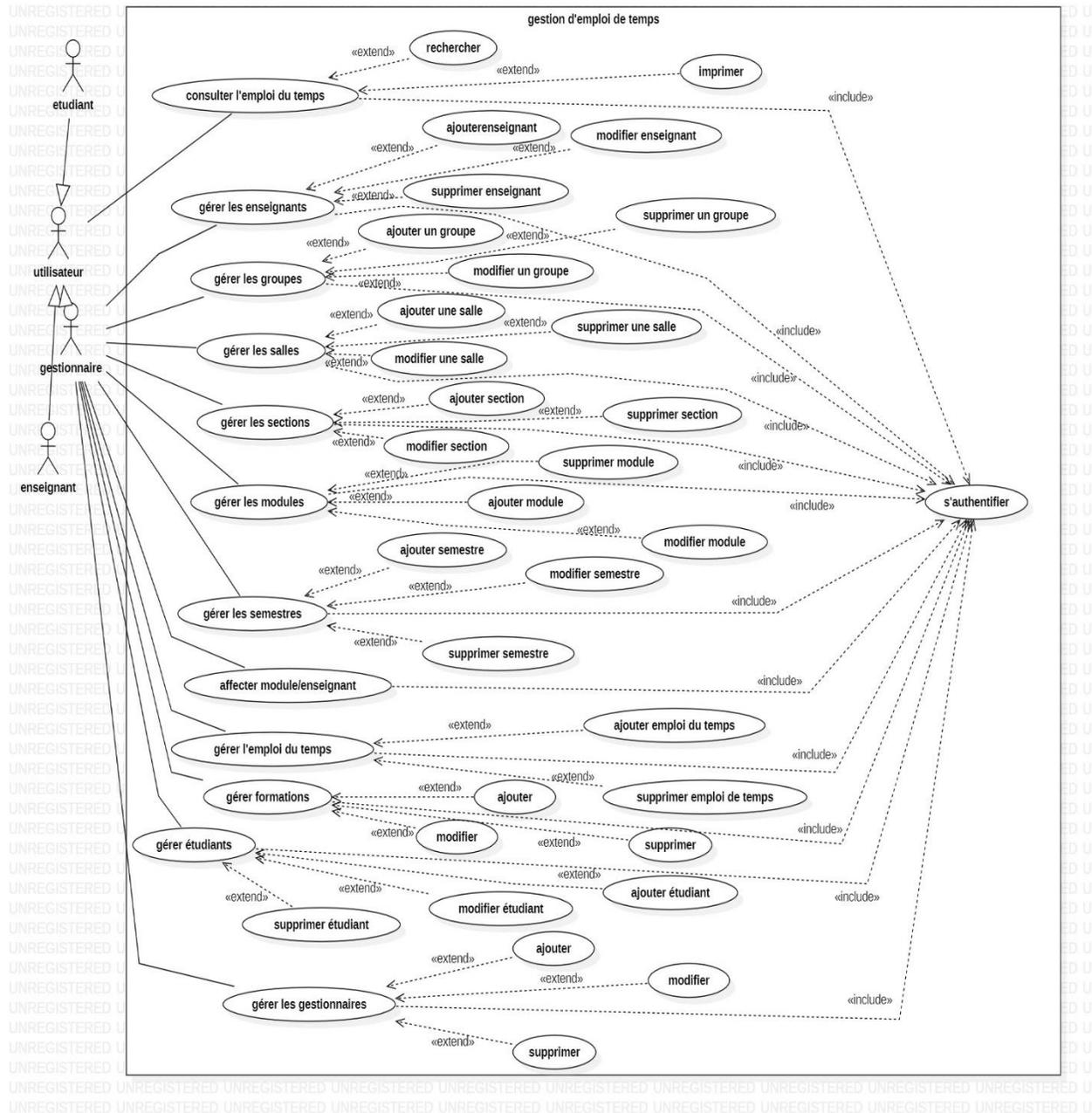


Figure 4: Diagramme de cas d'utilisation « globale »

III.4. Description des cas d'utilisations

➤ Cas d'utilisation « authentification » :

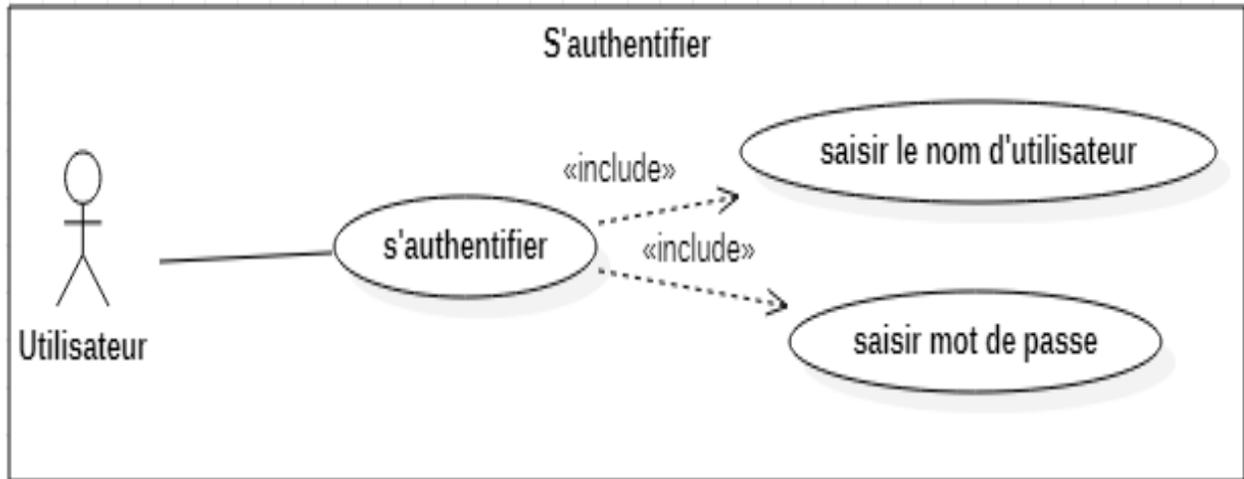


Figure 5: Diagramme de cas d'utilisation « authentification »

Acteurs : Utilisateur

Objectif : Permet à l'utilisateur de se connecter au système, et assurer que l'utilisateur est bien celui qui prétant être.

Scénario nominal :

1. Le système affiche la fenêtre d'authentification.
2. L'utilisateur introduit son nom et son mot de passe.
3. Le système vérifie si les données saisies sont valides.
4. Le système affiche l'espace approprié pour chaque utilisateur.

Scénario alternatif :

Si les informations saisir sont incorrectes ou incomplets le système affiche un message d'erreur et réaffiche la fenêtre d'authentification.

➤ Cas d'utilisation « Gérer les étudiants » :

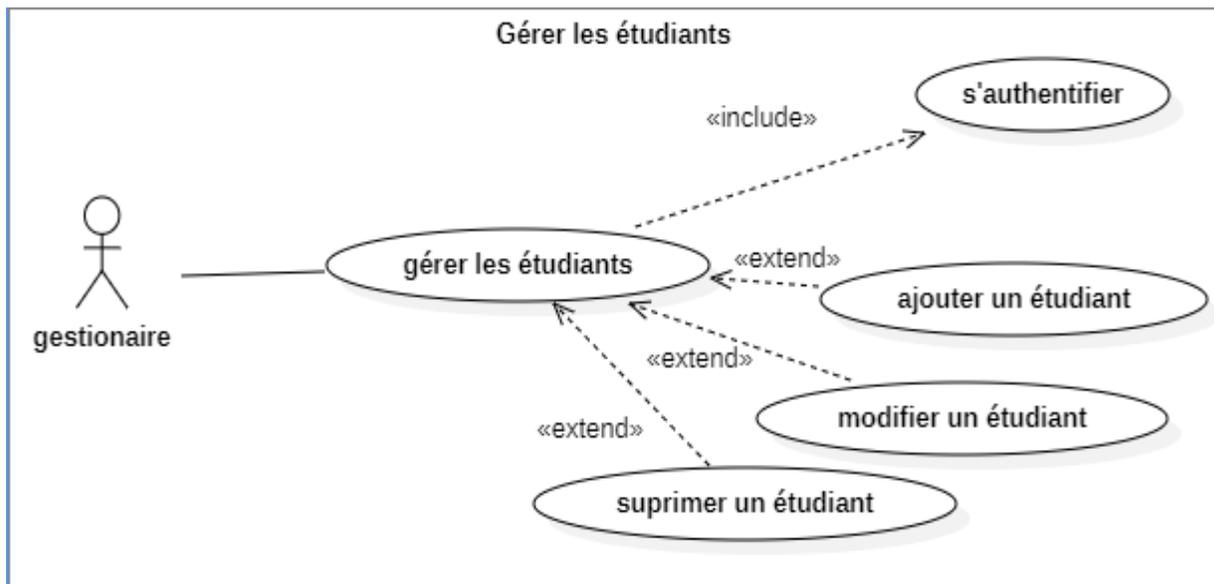


Figure 6: Diagramme de cas d'utilisation « Gérer les étudiants »

Acteur principale : Gestionnaire

Objectif : ajouter, modifier, supprimer un étudiant

Scenario nominal :

Cas 1 : Ajouter un étudiant

1. Le gestionnaire choisit ajouter un étudiant.
2. Le système affiche le formulaire à remplir.
3. Le gestionnaire remplit et valide le formulaire.
4. Le système ajoute les informations dans la base de données.

Cas 2 : Modifier un étudiant

1. Le gestionnaire choisit modifier un étudiant.
2. Le système affiche le formulaire de modification.
3. Le gestionnaire modifie les champs voulus.
4. Le système met à jour les informations.

Cas 3 : Supprimer un étudiant

1. Le gestionnaire choisit supprimer un étudiant.
2. Le système demande une confirmation.
3. Le gestionnaire confirme la suppression.
4. Le système supprimé l'étudiant de la base de données.

Scenario alternatif :

Cas 1 :

- ✓ Étudiant existe déjà ou les champs incomplets : le système affiche un message d'erreur et réaffiche le formulaire d'ajout.

Cas 2 :

- ✓ Les champs incomplets : Le système affiche un message d'erreur et réaffiche le formulaire de modification.

Cas 3 :

- ✓ Le gestionnaire annule la suppression : le système revient à la liste.

Cas d'utilisation « gérer les enseignants » :

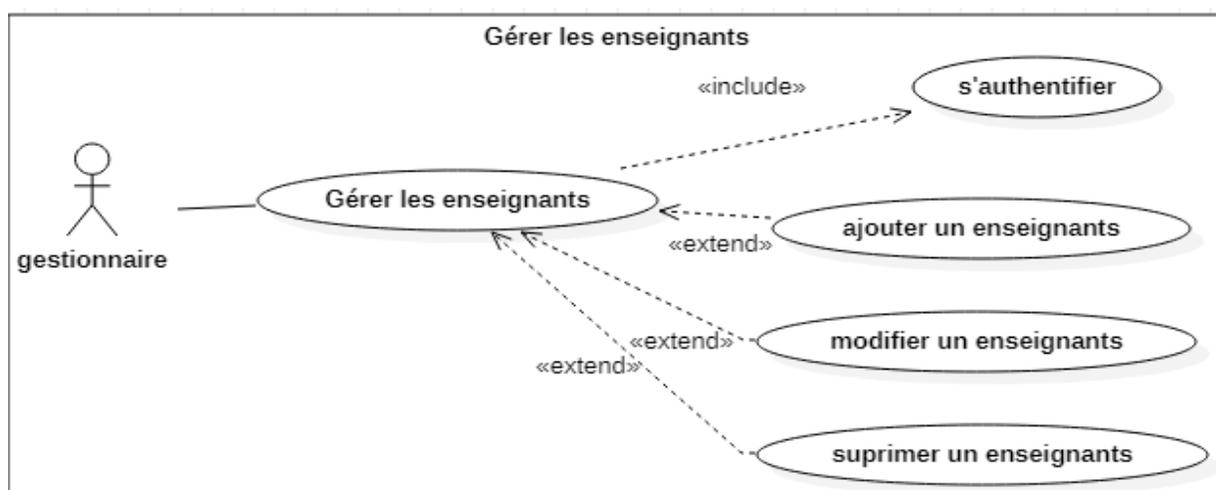


Figure 7: Diagramme de cas d'utilisation « gérer les enseignants »

Acteur principale : Gestionnaire

Objectif : ajouter, modifier, supprimer un enseignant.

Scenario nominal :

Cas 1 : Ajouter un enseignant

1. Le gestionnaire choisit ajouter un enseignant.
2. Le système affiche le formulaire à remplir.
3. Le gestionnaire remplit et valide le formulaire.
4. Le système ajoute les informations dans la base de données.

Cas 2 : Modifier un enseignant

1. Le gestionnaire choisit modifier un enseignant.
2. Le système affiche le formulaire de modification.
3. Le gestionnaire modifie les champs voulus.
4. Le système met à jour les informations.

Cas 3 : Supprimer un enseignant

1. Le gestionnaire choisit supprimer un enseignant.
2. Le système demande une confirmation.
3. Le gestionnaire confirme la suppression.
4. Le système supprime l'enseignant de la base de données.

Scenario alternatif :

Cas 1 :

- ✓ Enseignant existe déjà ou les champs incomplets : le système affiche un message d'erreur et réaffiche le formulaire d'ajout.

Cas 2 :

- ✓ Les champs incomplets : Le système affiche un message d'erreur et réaffiche le formulaire de modification.

Cas 3 :

- ✓ Le gestionnaire annule la suppression : le système revient à la liste.

Cas d'utilisation « consulter un EDT » :

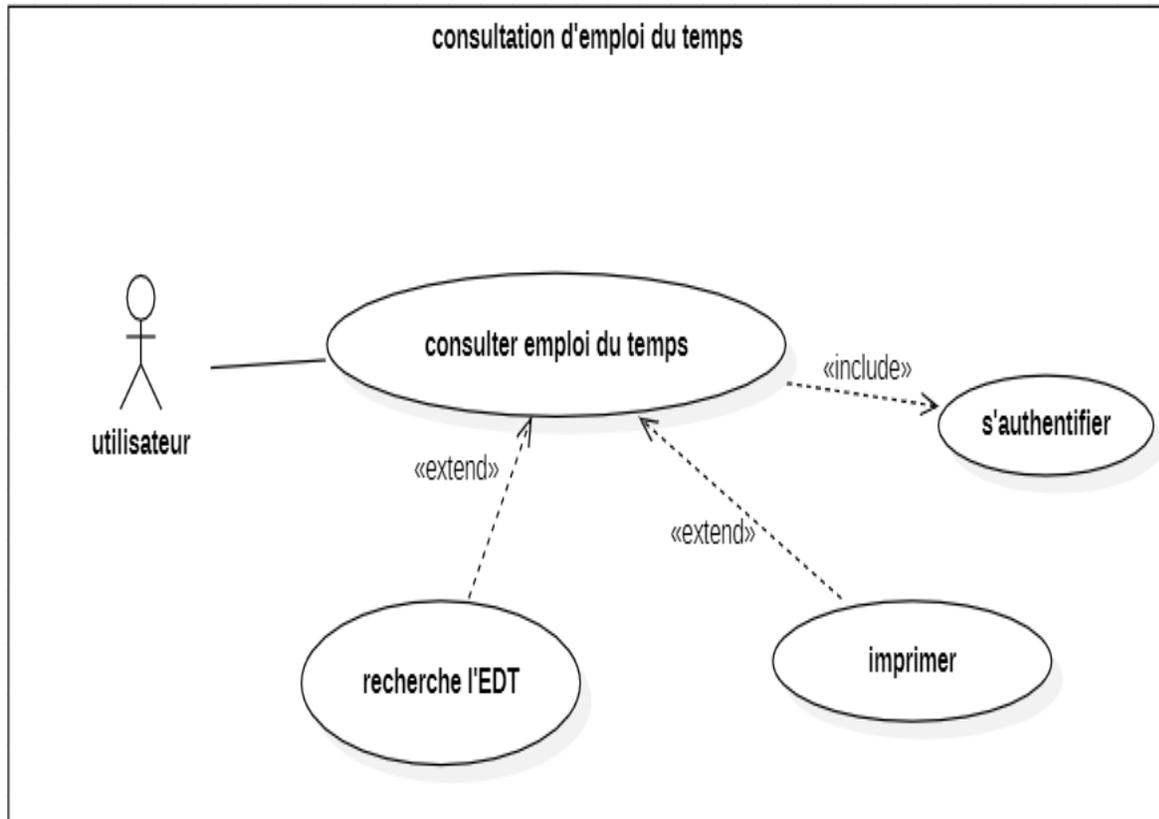


Figure 8: Diagramme de cas d'utilisation « consulter un EDT »

Acteur principale : Enseignant, Etudiant.

Objectif : consulter un emploi de temps.

Scénario nominale :

1. Après que l'utilisateur sera authentifié.
2. Le système lui affiche un moteur de recherche.
3. L'acteur introduit ce qu'il souhaite rechercher.
4. Le système lui affiche les EDT qu'il a recherché.

➤ Cas d'utilisation « Gérer les sections » :

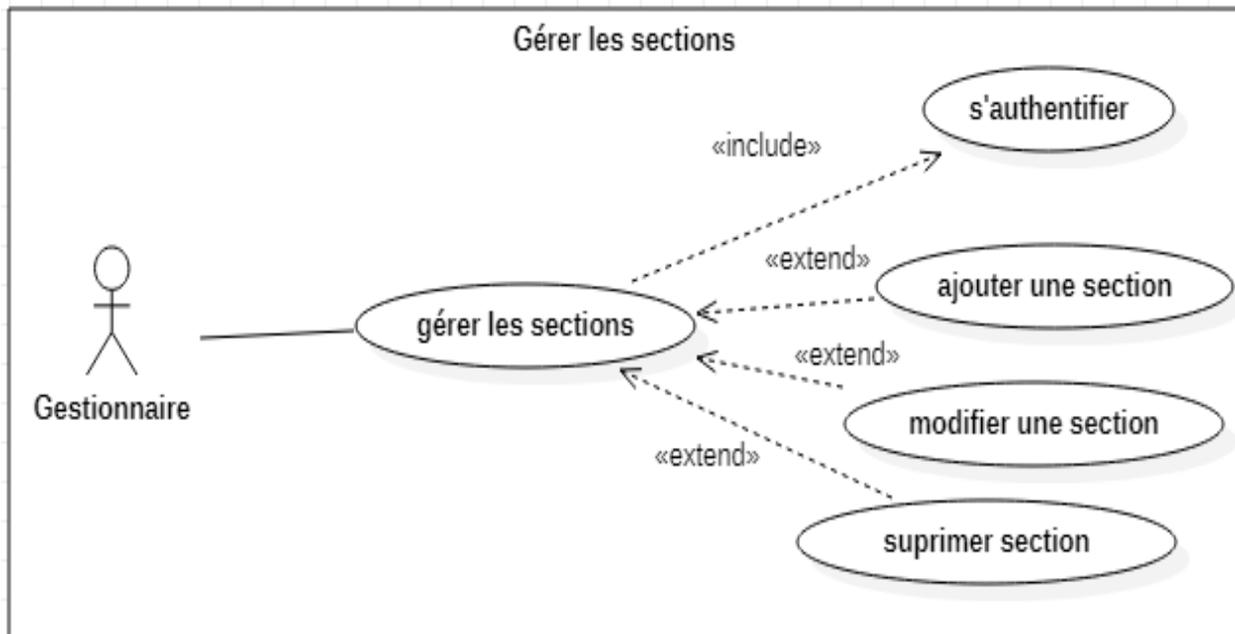


Figure 9: Diagramme de cas d'utilisation « Gérer les sections »

Acteur principale : Gestionnaire

Objectif : ajouter, modifier, supprimer une section.

Scenario nominal :

Cas 1 : Ajouter une section

1. Le gestionnaire choisit ajouter une section.
2. Le système affiche le formulaire à remplir.
3. Le gestionnaire remplit et valide le formulaire.
4. Le système ajoute les informations dans la base de données.

Cas 2 : Modifier une section

1. Le gestionnaire choisit modifier une section.
2. Le système affiche le formulaire de modification.
3. Le gestionnaire modifie les champs voulus.
4. Le système met à jour les informations.

Cas 3 : Supprimer une section

1. Le gestionnaire choisit supprimer une section.
2. Le système demande une confirmation.
3. Le gestionnaire confirme la suppression.
4. Le système supprimer la section de la base de données.

Scenario alternatif :

Cas 1 :

- ✓ Section existe déjà ou les champs incomplets : le système affiche un message d'erreur et réafficher le formulaire d'ajout.

Cas 2 :

- ✓ Les champs incomplets : Le système affiche un message d'erreur et réafficher le formulaire de modification.

Cas 3 :

- ✓ Le gestionnaire annule la suppression : le système revient à la liste.

➤ **Cas d'utilisation « Gérer les formations » :**

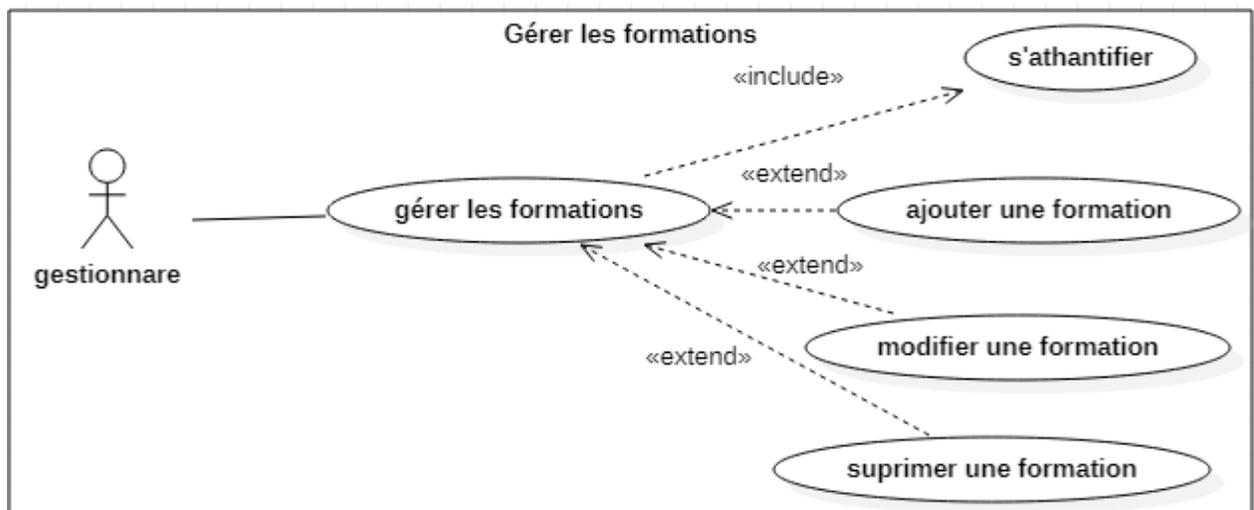


Figure 10 : Diagramme de cas d'utilisation « Gérer les formations »

Acteur principale : Gestionnaire

Objectif : ajouter, modifier, supprimer une formation.

Scenario nominal :

Cas 1 : Ajouter une formation

1. Le gestionnaire choisit ajouter une formation.
2. Le système affiche le formulaire à remplir.
3. Le gestionnaire remplit et valide le formulaire.
4. Le système ajoute les informations dans la base de données

Cas 2 : Modifier une formation

1. Le gestionnaire choisit modifier une formation.
2. Le système affiche le formulaire de modification.
3. Le gestionnaire modifie les champs voulus.
4. Le système met à jour les informations.

Cas 3 : Supprimer une formation

1. Le gestionnaire choisit supprimer une formation.
2. Le système demande une confirmation.
3. Le gestionnaire confirme la suppression.
4. Le système supprime l'information de la base de données.

Scenario alternatif :

Cas 1 :

- ✓ Enseignant existe déjà ou les champs incomplets : le système affiche un message d'erreur et réaffiche le formulaire d'ajout.

Cas 2 :

- ✓ Les champs incomplets : Le système affiche un message d'erreur et réaffiche le formulaire de modification.

Cas 3 :

- ✓ Le gestionnaire annule la suppression : le système revient à la liste.

III.5. Conception

III.5.1. Diagramme de séquence

Le diagramme de séquence représente des échanges de messages entre éléments, dans le cadre d'un fonctionnement particulier du système.

Vu le nombre important de cas d'utilisation qu'on a recensé, nous allons décrire que six exemples de cas d'utilisation.

III.5.1.1. Diagramme de séquence « Authentification »

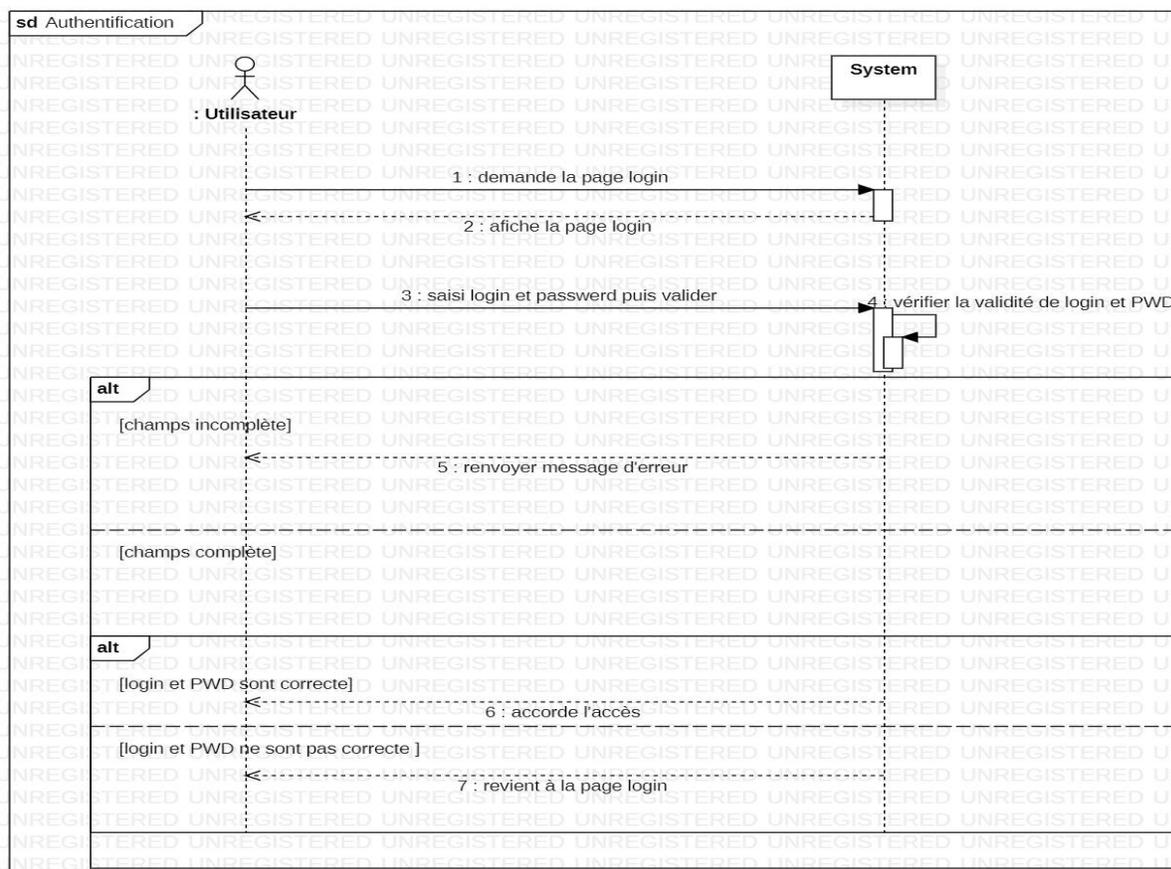


Figure 11: Diagramme de séquence « Authentification »

III.5.1.2. Diagramme de séquence « Ajouter formation »

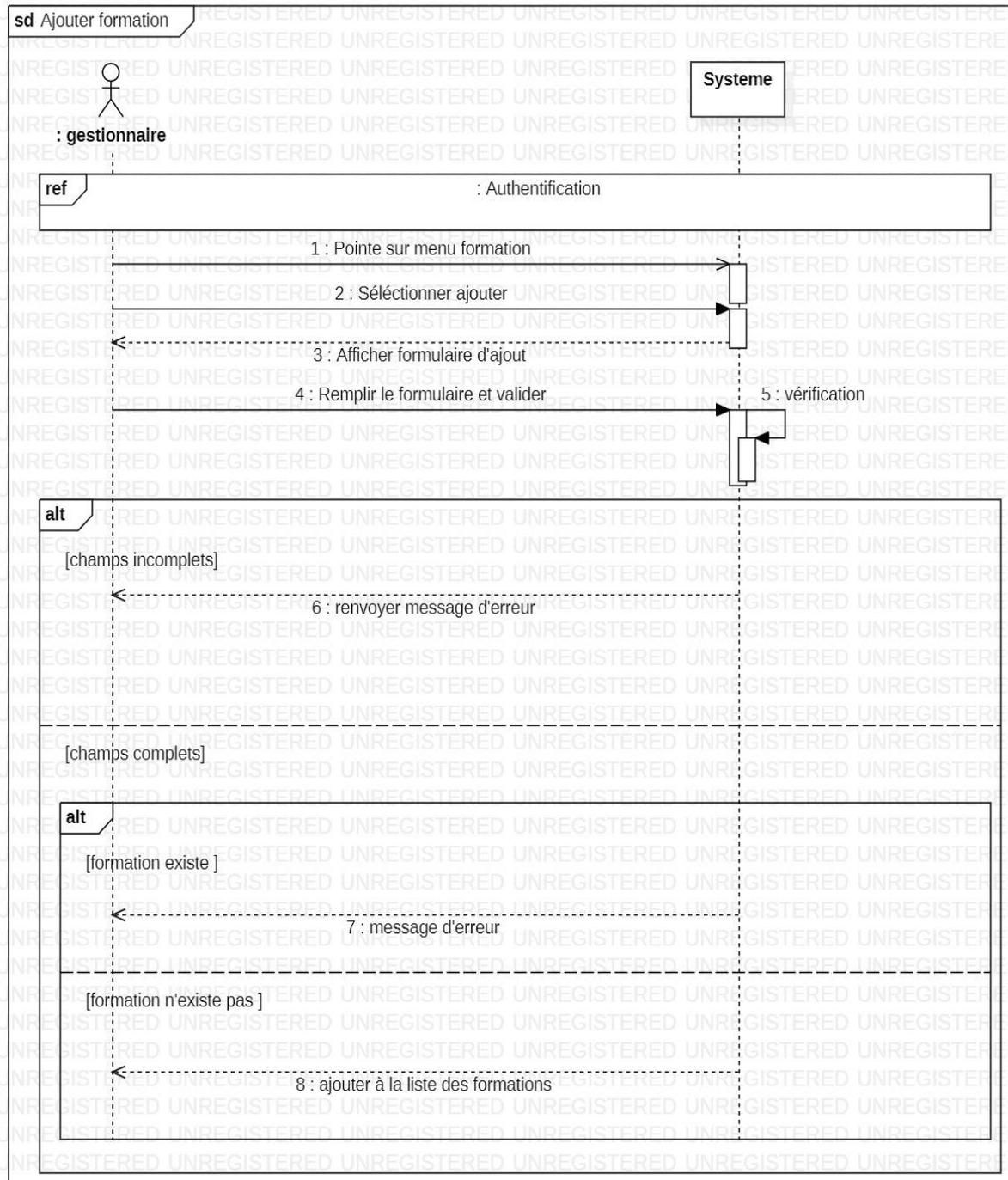


Figure 12: Diagramme de séquence « Ajouter formation »

III.5.1.3. Diagramme de séquence « Modifier un étudiant »

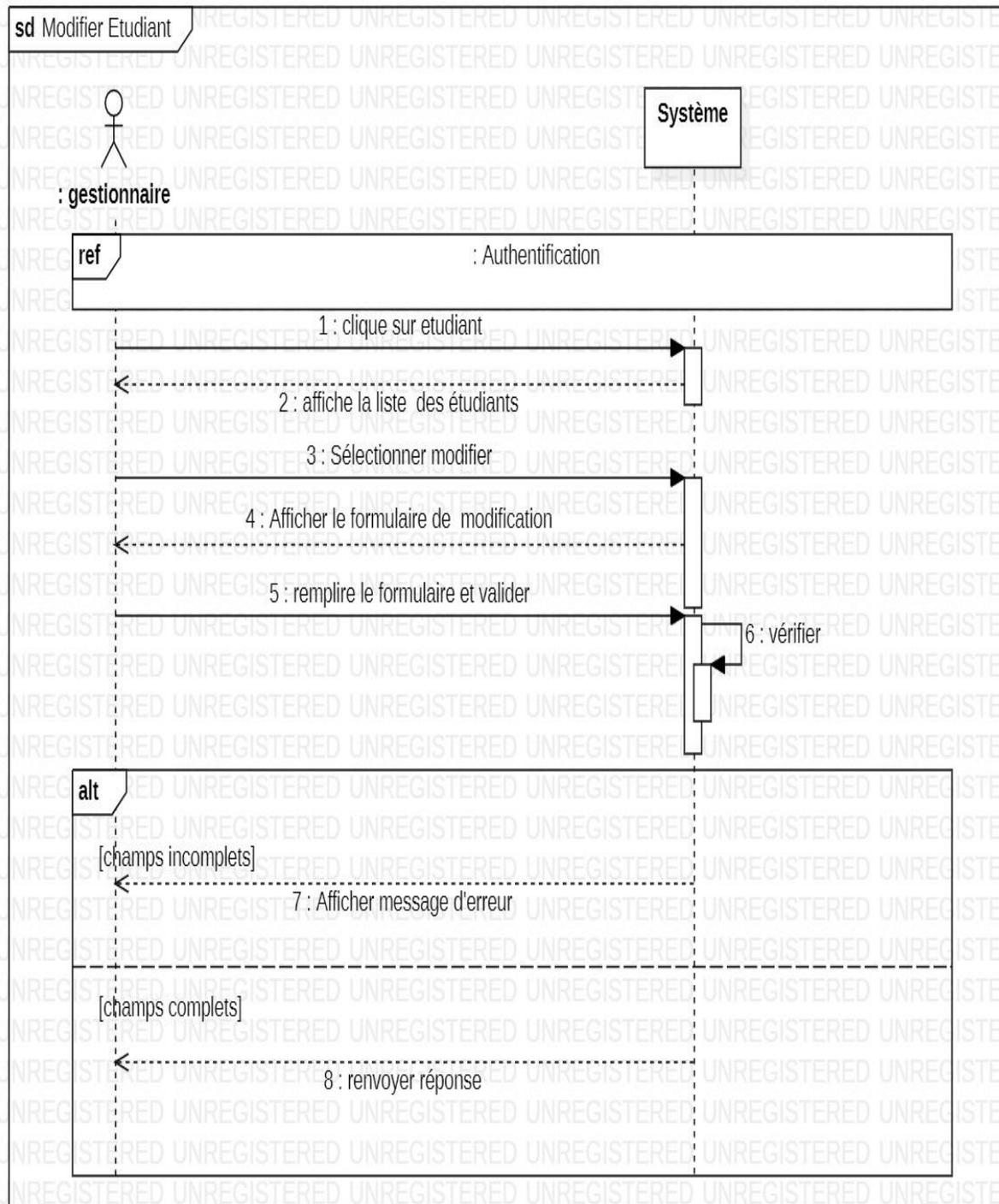


Figure 13 : Diagramme de séquence « Modifier étudiant »

III.5.1.4. Diagramme de séquence « Supprimer un enseignant »

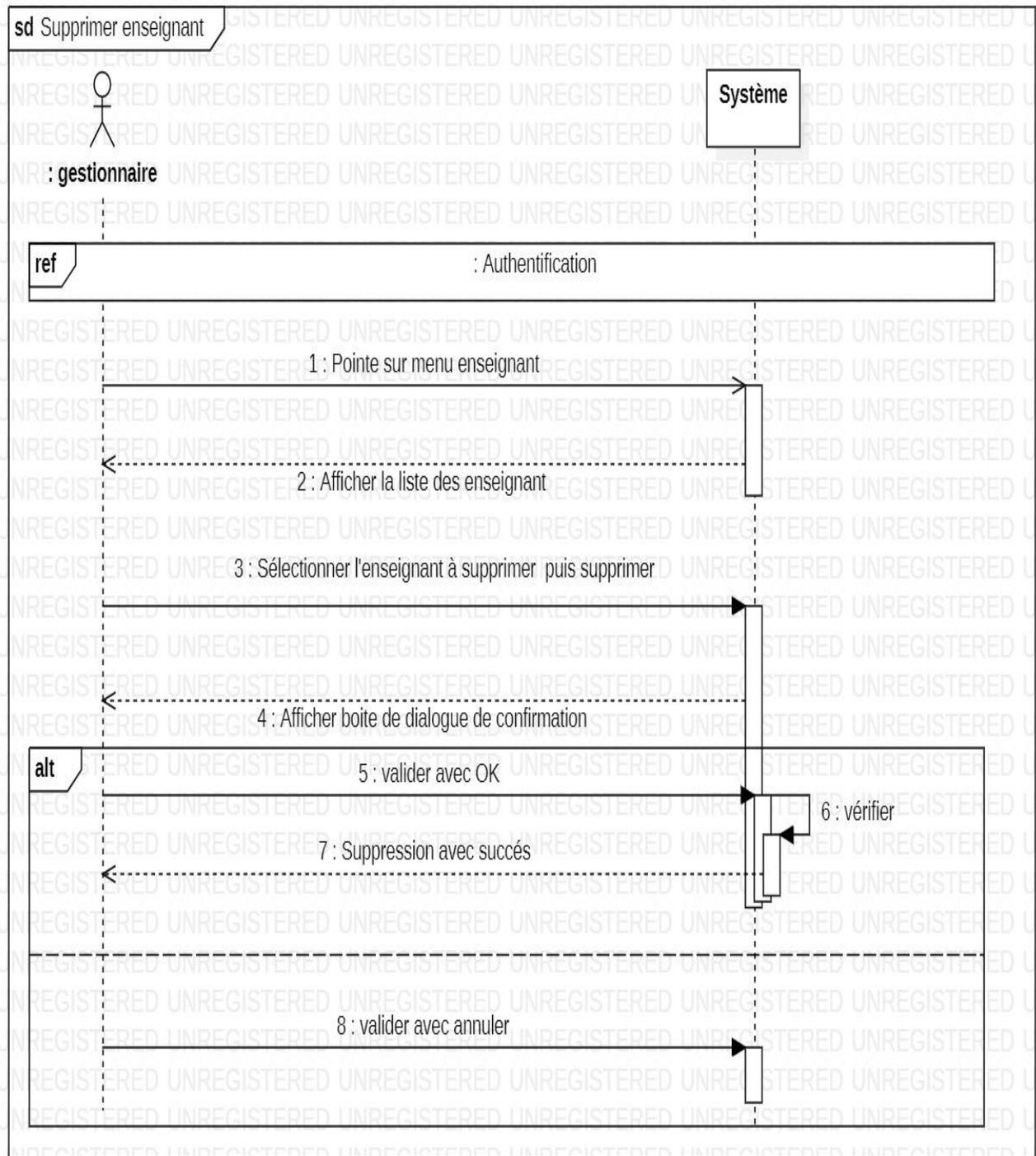


Figure 14: Diagramme de séquence « Supprimer enseignant »

III.5.1.5. Diagramme de séquence « Créer un EDT »

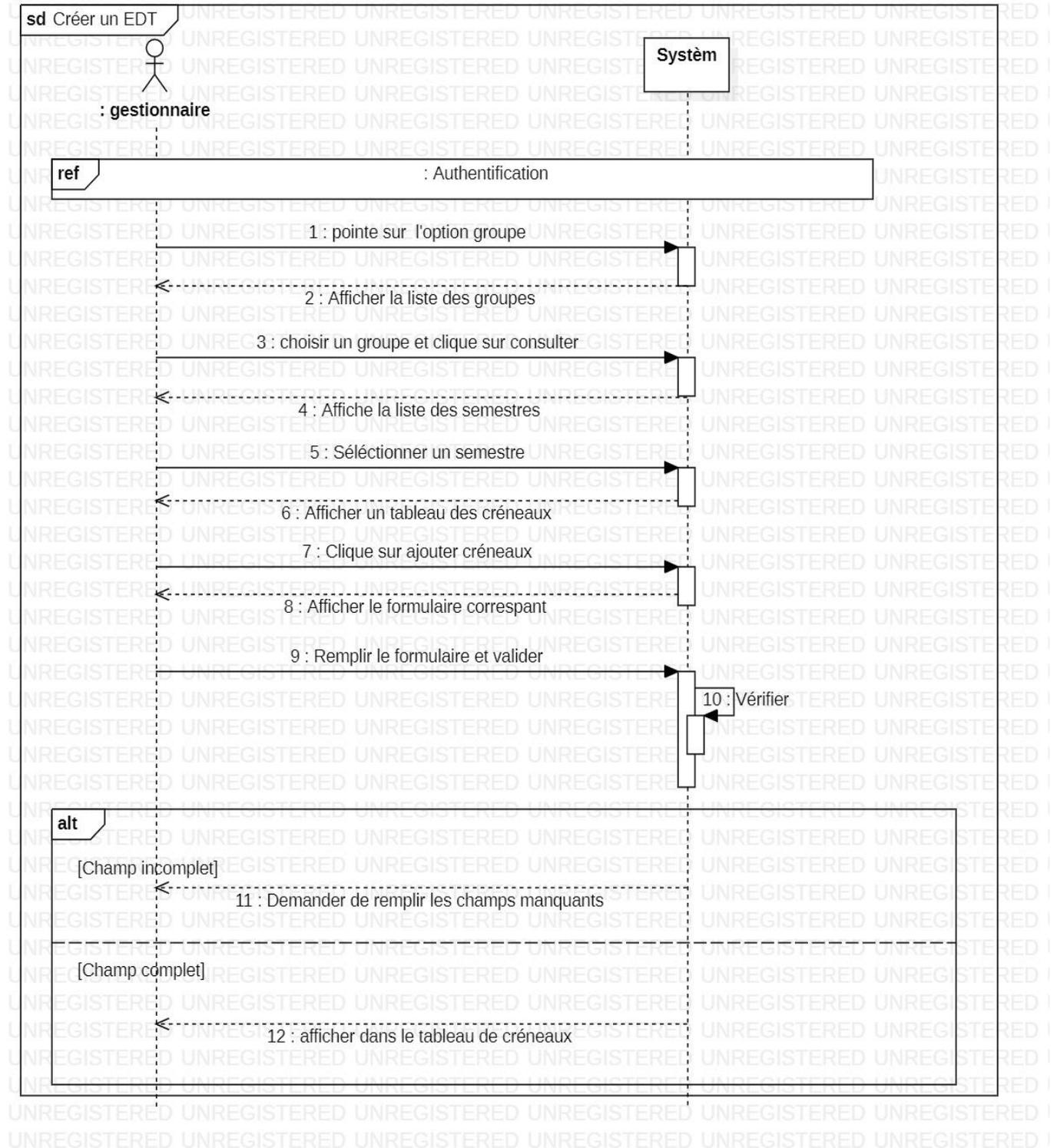


Figure 15: Diagramme de séquence « Créer un EDT »

III.5.1.6. Diagramme de séquence « Consulter et imprimer l'EDT »

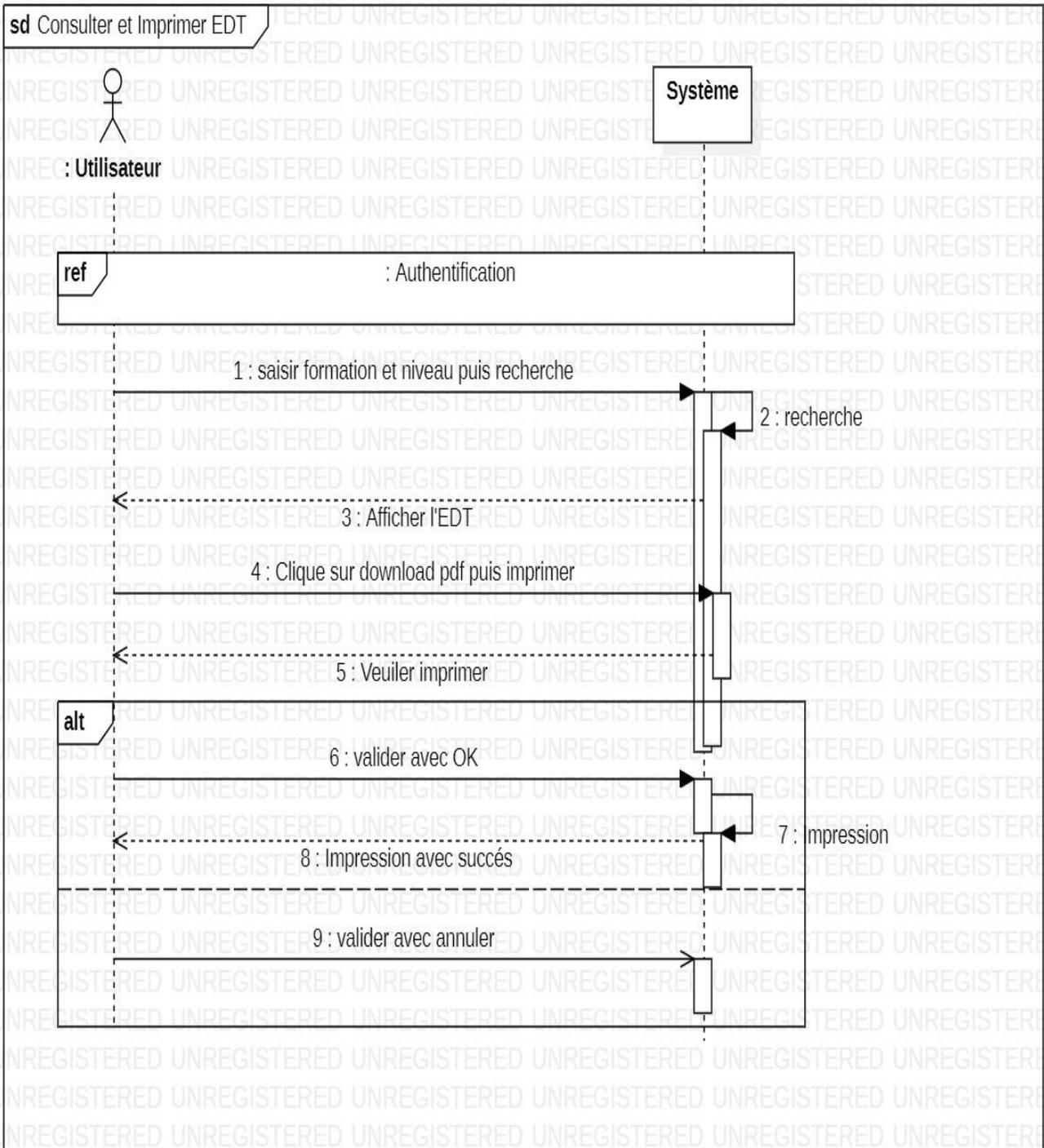


Figure 16: Diagramme de séquence « Consulter et imprimer l'EDT »

III.5.2. Diagramme de classe

Avant d'élaborer le diagramme de classe nous allons définir le dictionnaire de données.

III.5.2.1. Dictionnaire de données

Le dictionnaire de données permet de définir les informations nécessaires. Il précise le nom, le type et la désignation de chaque attribut.

Nom de la classe	L'attribut	Désignation	Type
Utilisateur	Id-user	Identifiant de l'utilisateur	Integerfield
	Nom	Nom de l'utilisateur	Charfield
	Prénom	Prénom de l'utilisateur	Charfield
	Login	Login de l'utilisateur	Charfield
	Password	Mot de passe de l'utilisateur	Password
	Email	L'adresse email de l'utilisateur	Charfield

Tableau 1 : Structure de la table utilisateur

Nom de la classe	L'attribut	Désignation	Type
Enseignant	Grade	Le grade de l'enseignant	Charfield
	Charge_journalière	Le nombre d'heure par jour	Integerfield

Tableau 2: Structure de la table enseignant

Nom de la classe	L'attribut	Désignation	Type
Formation	Id_formation	Identifiant de la formation	Integerfield
	Nom_formation	Nom de la formation	Charfield
	Niveau	Niveau scolaire de l'étudiant	Charfield
	Nb_semestre	Nombre de semestre de chaque formation	Integerfield

Tableau 3: Structure de la table formation

Nom de la classe	L'attribut	Désignation	Type
Groupe	Id_groupe Nom_groupe	Identifiant Nom de groupe	Integerfield Charfield

Tableau 4 : Structure de la table groupe

Nom de la classe	L'attribut	Désignation	Type
Semestre	Id_semestre Nom_semestre Session Date_début Date_fin	Identifiant de semestre Nom de semestre Les différents périodes d'un semestre Date de début de semestre Date de fin de semestre	Integerfield Charfield Charfield Datefield Datefield

Tableau 5 : Structure de la table semestre

Nom de la classe	L'attribut	Désignation	Type
Section	Id_section Nom_section	Identifiant de la section Nom de la section	Integerfield Charfield

Tableau 6 : Structure de la table section

Nom de la classe	L'attribut	Désignation	Type
Module	Id_module Nom_module Volume_hébdomadaire	Identifiant de module Nom de module Nombre de séances par semaine	Integerfield Charfield Integerfield

Tableau 7 : Structure de la table modules

Nom de la classe	L'attribut	Désignation	Type
Salle	Id_salle nom-salle capacité type	Identifiant de salle Nom de la salle Nombre de place dans la salle Type de la salle (salle de TD, TP, cour)	Integerfield Charfield Integerfield Charfield

Tableau 8 : Structure de la table salle

Nom de la classe	L'attribut	Désignation	Type
Séance	Jour Créneau Type	Jour de la séance Heure de début et fin de la séance Type de la séance (séance de TD, TP, cour)	Charfield Integerfield Charfield

Tableau 9 : Structure de la table séance

La figure ci-dessus illustre diagramme de classe de système

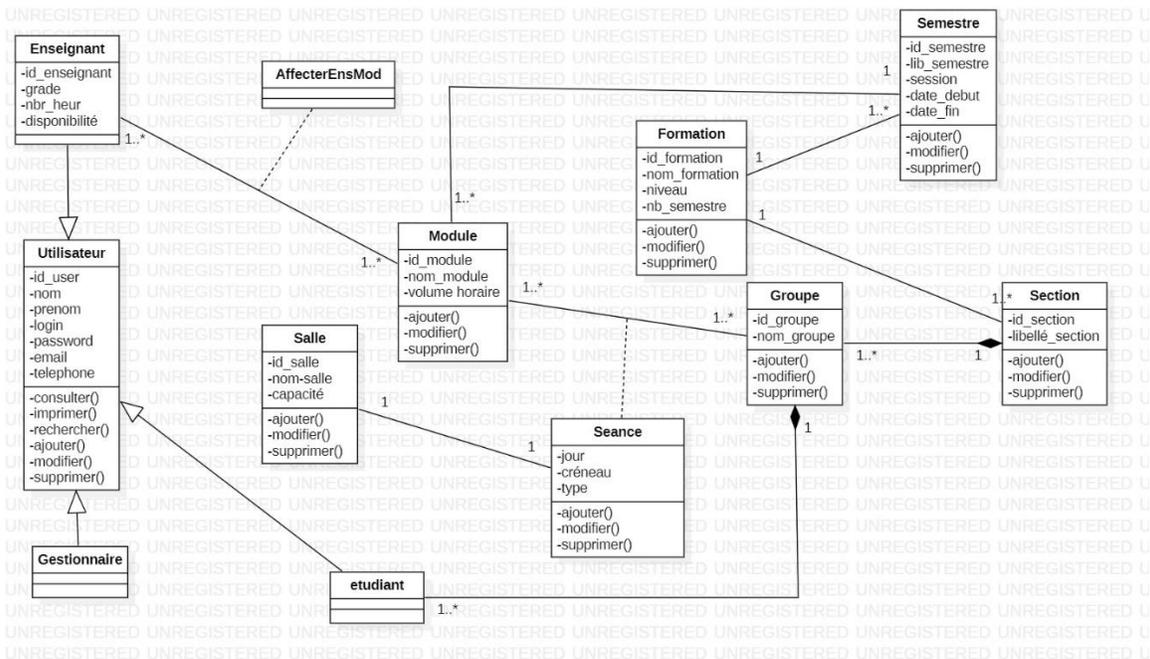


Figure 17 : Diagramme de classe

III.5.3. Modèle relationnel

A partir du diagramme de classes, nous allons réaliser le modèle relationnel qui est le modèle logique de données, ce modèle décrit de façon abstraite comment sont représentées les données dans une base de données.

III.5.3.1. Les règles de passage au modèle relationnel à partir d'un diagramme de classes

Pour avoir le schéma relationnel de la base de données de l'application à réaliser à partir du modèle de classes, nous utilisons les règles de passage suivantes [11].

Règle 1 : Transformation des classes : chaque classe du diagramme UML devient une relation, il faut choisir un attribut de la classe pouvant jouer le rôle de clé (le rôle de l'identifiant).

Règle 2 : Transformation des associations : Nous distinguons trois familles d'associations :

- Association 1..* : 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.
- Association *.* et n-aire et classe-association : La classe-association devient une relation. La clé primaire de cette relation est la concaténation des identifiants des classes connectées à l'association.
- Association 1..1 : Il faut ajouter un attribut de type 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 3 : présence d'une généralisation (méthode 1 : push-up) : Créer une relation avec tous les attributs des classes. Ajouter un attribut pour distinguer les types d'objets.

III.5.3.2. Le passage au modèle relationnel

Gestionnaire (id_user, nom, prénom, login, password, email).

Etudiant (id_user, nom, prénom, login, password, email, #id_groupe).

Enseignant(id_user, nom, prénom, login, password, email, téléphone, grade, charge_journalière).

Module (id_module, nom_module, volume_hébdomadaire, #id_semestre).

AffecterEns_Mod (#id_user, #id_module).

Salle (id_salle, nom_salle, capacité, type).

Séance (id_groupe, #id_module, jour, créneau, type).

Formation (id_formation, nom, niveau, nb_semestre).

Groupe (id_groupe, nom, #id_section).

Section (id_section, lib_section, #id_formation).

Semestre (id_semestre, lib_semestre, session, date_debut, date_fin, #id_formation).

Conclusion

Au cours de ce chapitre, nous avons utilisé un langage de modélisation appelée UML pour planifier notre application. Nous avons choisi quelques diagrammes pour expliquer comment notre application interagit avec son environnement, tels que le diagramme de cas d'utilisation et les diagrammes de séquences et de classes. Enfin, nous avons élaboré le modèle relationnel.

Dans le chapitre suivant, nous allons présenter l'implémentation à l'aide des différents outils logiciels décrit dans le même chapitre.

Chapitre IV : Implémentation et résultats

Introduction

Dans les précédents chapitres, nous avons établi un cahier des charges détaillé qui nous a permis de fixer un objectif concret. Par la suite, en procédant à une analyse et une conception, nous avons pu acquérir une vision claire de notre application web.

Le présent chapitre marque la réalisation de notre projet, nous allons présenter l'environnement et les outils de développement de notre application, expliquer son fonctionnement et présenter quelques interfaces illustratives. Et nous avons consacré une attention particulière à définir les perspectives d'application tout au long de notre travail.

IV.1. Application web

Une application web est un type de logiciel applicatif qui est hébergé sur un serveur et qui peut être accessible via un navigateur web. Contrairement à un logiciel traditionnel, l'utilisateur n'a pas besoin d'installer l'application sur son ordinateur. Pour accéder à l'application, il lui suffit simplement de se connecter à celle-ci en utilisant son navigateur préféré. De nos jours, la tendance est d'offrir une expérience utilisateur similaire à celle des logiciels installés directement sur les ordinateurs, avec des fonctionnalités équivalentes [12].

IV.2. Environnements de développement

IV.2.1. Visual studio code

VSCoDe est un éditeur de code source et un environnement de développement intégré (IDE) de Microsoft. Il est open-source et cross-Platform, c'est-à-dire qu'il fonctionne sur Windows, Linux et Mac. Il a été conçu pour les développeurs web, mais il prend en charge de nombreux autres langages de programmation tels que C++, C#, Python, Java, etc. Il offre de nombreuses fonctionnalités comme la coloration syntaxique, l'auto-complétion, la mise en évidence des erreurs, la navigation de code, le débogage, la gestion des versions, l'intégration avec Git, et beaucoup



d'autre. Il est également extensible à l'aide d'une grande variété d'extensions développées par la communauté, permettant aux développeurs de personnaliser l'éditeur selon leurs besoins.

IV.2.2. StarUML

Est un logiciel de modélisation UML (Unified Modeling Language) open source, dédiés aux plateformes Windows. Il est capable de prendre en charge l'ingénierie de logiciels dans plusieurs langages de programmation, dont Java et Python.



Le logiciel a été conçu en prévoyant l'ajout de plugins supplémentaires pour pouvoir être facilement adapté aux besoins évolutifs de ses utilisateurs. De plus, il permet la gestion de l'exportation des données au format XML, ainsi que l'exportation au format JPG afin d'intégrer des diagrammes dans des documents [13].

IV.3. Langages et outils du développement

IV.3.1. HTML

HTML (HyperText Markup Language) est un langage de balisage utilisé pour créer des pages web. Il permet de structurer et de présenter le contenu d'une page web, ainsi que de décrire la relation entre les différents éléments. En utilisant des balises HTML, on peut définir des titres, des paragraphes, des liens hypertexte, des images, des tableaux, des formulaires, etc. Le balisage HTML est interprété par les navigateurs web pour afficher le contenu d'une page web de manière claire et organisée.



IV.3.2. CSS

CSS (Cascading Style Sheets) est un langage de feuilles de style qui est utilisé pour décrire la présentation d'un document HTML ou XML. CSS permet de contrôler le style et mise en forme des éléments d'une page web tels que les couleurs, les polices de caractère, les marges, les bordures, les alignements, les tailles, l'espacement...etc.



En utilisant CSS, les concepteurs web peuvent séparer le contenu et la présentation d'une page, ce qui facilite la maintenance et l'adaptation du design aux différents appareils et résolutions d'écran [14].

IV.3.3. Python

Python est un langage de programmation dynamique, gratuit, extensible, simple, permet une approche modulaire orienté-objet, il supporte l'héritage multiple et la surcharge des opérateurs, il est portable non seulement sur les différentes variantes de d'Unix, mais aussi sur les OS propriétaires : Mac OS, BeOSNeXStep, MS-DOS et les différentes variantes de Windows. Développé depuis 1989 par Guido van Rossum et des nombreux contributeurs bénévoles [15].



➤ Pourquoi Python ?

Python s'est imposé comme un langage de programmation incontournable pour les développeurs du monde entier. Sa polyvalence intégrée en fait une solution idéale pour les diverses applications logicielles telles que les applications mobiles, les applications de bureau, le développement web et la programmation hardware. L'un des principaux avantages de python réside dans la facilité d'intégration avec d'autres langages de programmation tels que Ruby, java, PHP. De plus, les développeurs bénéficient d'un vaste choix de bibliothèques et des Framework gratuits, notamment Django, qui simplifient et accélèrent le développement de leurs projets web. En résumé, Python offre aux développeurs la possibilité d'évaluer rapidement et aisément leurs projets, tout en profitant d'une multitude de ressources disponibles pour faciliter leur travail [16].

IV.3.4. Django

Django est un Framework gratuit, a une communauté active, une bonne documentation, open-source de développement web, écrit en python.



Il permet aux développeurs de créer rapidement et efficacement des applications web dynamiques et interactives en fournissant des outils pour gérer l'authentification utilisateur, la création de formulaires, la manipulation de base de données et la gestion de routage. Django est largement utilisé pour la création de site web [17].

IV.3.5. JavaScript

JavaScript est un langage de programmation utilisé principalement dans le développement web. Il permet d'ajouter de l'interactivité aux sites internet en manipulant le contenu HTML et CSS, et en communiquant avec les serveurs pour afficher des données dynamiques. JavaScript est également utilisé pour créer des fonctionnalités interactives telles que des animations, des formulaires et des menus



déroulants. Il est polyvalent et peut être utilisé pour développer des applications web, des applications mobiles hybrides, des extensions de navigateur et même des serveurs. En somme, JavaScript est essentiel pour créer des expériences utilisateur dynamiques et interactives sur le web [18].

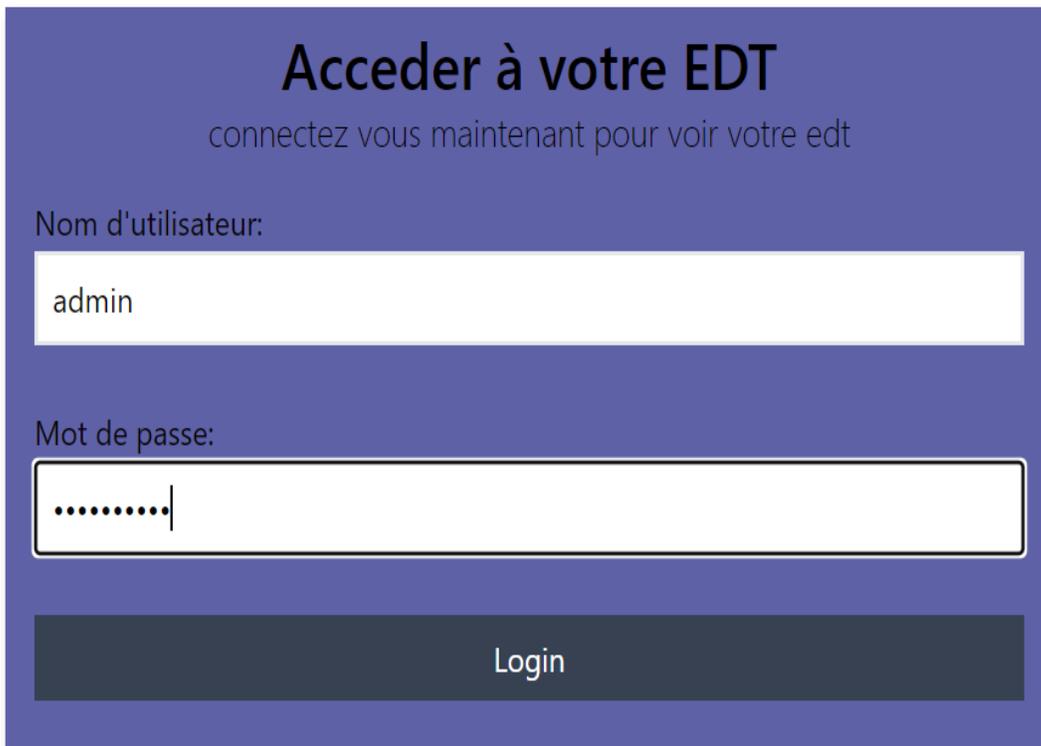
IV.4. Les fonctionnalités de l'application

L'interface graphique fait référence à la manière dont les éléments d'un logiciel sont organisés et affichés à l'écran pour l'utilisateur. Cela implique de placer les menus, les boutons et les fonctionnalités de manière adéquate dans la fenêtre.

Les figures suivantes décrivent les différentes interfaces disponibles dans notre application et leurs utilités.

IV.4.1. Connexion

Cette interface est la porte d'accès à notre application, où l'utilisateur peut saisir son nom d'utilisateur et son mot de passe afin d'accéder au tableau de bord.



The image shows a login interface with a dark blue background. At the top, the text 'Accéder à votre EDT' is displayed in a large, bold, white font. Below it, the instruction 'connectez vous maintenant pour voir votre edt' is written in a smaller, lighter blue font. There are two input fields: the first is labeled 'Nom d'utilisateur:' and contains the text 'admin'; the second is labeled 'Mot de passe:' and contains a series of dots. At the bottom, there is a dark blue button with the text 'Login' in white.

Figure 18 : Interface de connexion.

IV.4.2. Espace gestionnaire

IV.4.2.1. Tableau de bord

Après avoir saisi le nom d'utilisateur et le mot de passe puis connecté, le Dashboard sera affiché (Figure 2). Cette page est constituée par les éléments suivants :

- Un menu principal qui contient des différents liens d'autres pages (gestionnaires, étudiants, enseignants, semestres, formations, sections, groupes, modules, salles).
- Une fonctionnalité se déconnecter permet de retourner au page d'authentification et une marque de l'utilisateur qui accède à l'application (gestionnaire, enseignant, étudiant).
- Des statistiques qui montrent le nombre d'étudiants dans chaque formation



Figure 19 : Tableau de bord.

IV.4.2.2. Ajouter formation

Voici l'interface correspondant au formulaire d'ajout d'une formation. Le gestionnaire complète tous les champs requis avant de procéder à l'enregistrement, ce qui permet d'ajouter directement la formation à la base de données.

Ajouter une nouvelle formation

Nom:

Niveau:

Nombre de semestres:

Ajouter

Nom	Actions
RN	supprimer modifier
GL	supprimer modifier
ASR	supprimer modifier
IA	supprimer modifier
SIA	supprimer modifier
RS	supprimer modifier

Figure 20 : Ajouter formation.

IV.4.2.3. Modifier un enseignant

Cette interface comprend un formulaire de modification. Le gestionnaire modifier le champ voulu et enregistrer les modifications.

Modifier l'enseignant narimensaad

Nom d'utilisateur:

Prénom:

Nom:

Email:

Mot de passe:

Charge journalière:

Grade:

Ajouter

Nom d'utilisateur	Charge journalière	Actions
achroufane6	20	supprimer modifier modules
bouchebah6	17	supprimer modifier modules
djebbar6	10	supprimer modifier modules
elboussi6	26	supprimer modifier modules
narimensaad	19	supprimer modifier modules
sider6	30	supprimer modifier modules
zerarga6	19	supprimer modifier modules

Figure 21 : Modifier un enseignant.

IV.4.2.4. Affecter module a l'enseignant

Pour affecter un module à un enseignant, il est nécessaire d'ajouter à la fois des modules et des enseignants. Ensuite, chaque enseignant se voit attribuer les modules qui lui correspondent. Voici l'interface d'affectation des modules aux enseignants.

Liste des enseignants

[Ajouter](#)

Nom d'utilisateur	Prénom	Nom	Email	Grade	Charge journalière	Actions
adel	Z/	Adel	zadel@gmail.com	professeur	13	supprimer modifier modules
ciliam	C/	mebarki	mebarkicia@gmail.com	vacataire	16	supprimer modifier modules
dada	F/	mebarki	fidaameb@gmail.com	vacataire	18	supprimer modifier modules
djebar6	N/	djebar	naceradjeb@gmail.com	professeur	16	supprimer modifier modules
elbouhissi6	H/	elbouhissi	houda@gmail.com	professeur	17	supprimer modifier modules
mounir	saber	DJAHID	mamine19@gmail.com	vacataire	11	supprimer modifier modules
narimensaad	narimen	SAAD	saad_narimen@live.com	professeur	12	supprimer modifier modules

22 sur 24 - Presse-Papi
Élément ajouté.

Figure 22 : La liste des enseignants.

Liste des modules

[Ajouter](#)

Nom	Volume hebdomadaire	Semestre	Formation	Niveau	Actions
LM	2	S1	RN	L2	supprimer modifier
POO	2	S1	RN	L2	supprimer modifier
Proba	2	S1	RN	L2	supprimer modifier
Analyse	3	S1	RN	L2	supprimer modifier
OOE	2	S1	RN	L2	supprimer modifier
AGL	2	S1	GL	M2	supprimer modifier

Figure 23 : La liste des modules

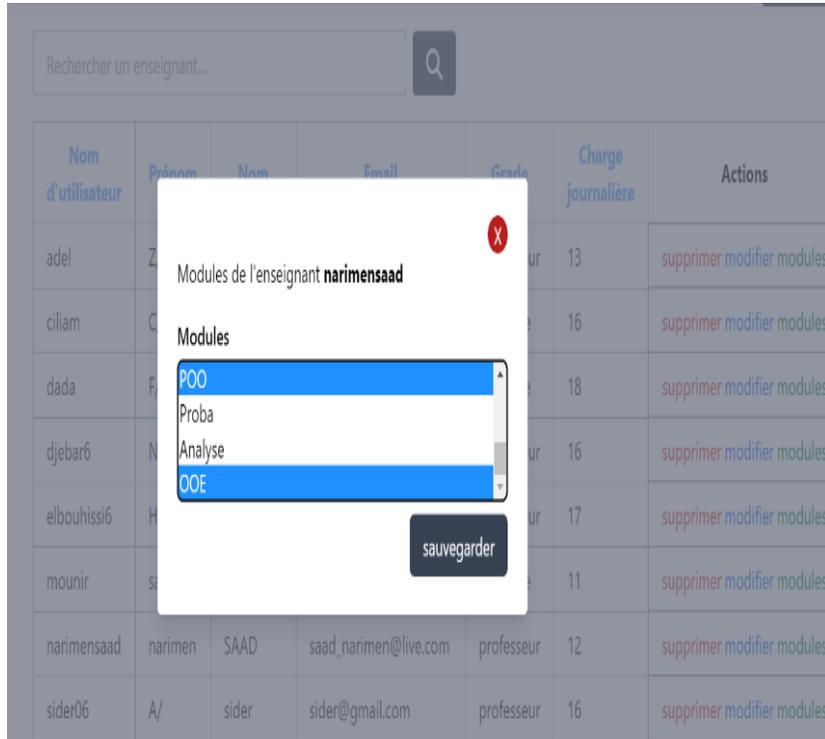


Figure 24 : Affecter module a l'enseignant

IV.4.2.5. Recherche d'un enseignant

Pour rechercher un enseignant spécifique, il est nécessaire d'entrer son nom dans la barre de recherche, puis de valider. Le système affichera ensuite l'enseignant qui porte ce nom.

Liste des enseignants

[Ajouter](#)

SAAD

Nom d'utilisateur	Prénom	Nom	Email	Grade	Charge journalière	Actions
narimensaad	narimen	SAAD	saad_narimen@live.com	professeur	12	supprimer modifier modules

Figure 25 : La recherche d'un enseignant

IV.4.2.6. Supprimer un enseignant

Dans cette interface lorsque le gestionnaire clique pour supprimer un enseignant le système affiche un message de confirmation avant de procéder à la suppression.

Liste des enseignants

Ajouter

Rechercher un enseignant...

Nom d'utilisateur	Prénom	Nom	Email	Grade	Charge journalière	Actions
dada	F/	mebarki	fdaameb@gmail.com	vacataire	18	supprimer modifier modules
elbouhissif	H/	elbouhissi	houda@gmail.com	professeur	17	supprimer modifier modules
zemergaif	L/	Zererga	lofizer@gmail.com	professeur	17	supprimer modifier modules
ssour	ssber	DIAHID	mamine19@gmail.com	vacataire	11	supprimer modifier modules
cilam	C/	mebarki	mebarkicila@gmail.com	vacataire	16	supprimer modifier modules
djebarif	N/	djebar	naceradjeib@gmail.com	professeur	16	supprimer modifier modules
narimensaad	narimen	SAAD	saad_narimen@live.com	professeur	12	supprimer modifier modules

Liste des enseignants

Ajouter

Rechercher un enseignant...

Nom d'utilisateur	Prénom	Nom	Email	Grade	Charge journalière	Actions
adel	Z/	Adel	zadel@gmail.com	professeur	13	supprimer modifier modules
cilam	C/	mebarki	mebarkicila@gmail.com	vacataire	16	supprimer modifier modules
dada	F/	mebarki	fdaameb@gmail.com	vacataire	18	supprimer modifier modules
djebarif	N/	djebar	naceradjeib@gmail.com	professeur	16	supprimer modifier modules
elbouhissif	H/	elbouhissi	houda@gmail.com	professeur	17	supprimer modifier modules
narimensaad	narimen	SAAD	saad_narimen@live.com	professeur	12	supprimer modifier modules

127.0.0.1:8000 indique

Etes vous sur de vouloir supprimer cette ressource?

Figure 26 : Suppression d'un enseignant

IV.4.2.7. Génération d'un emploi du temps

Cette interface permet aux gestionnaires d'ajouter ou supprimer une séance.

Universit  de b jaia
admin

Admin

Universit  de b jaia

- Gestionnaires
- Etudiants
- Enseignants
- Semestres
- Formations
- Sections
- Groupes
- Modules
- Salles

Choisir un semestre:

S1 : 2022-09-24 - 2023-02-28

	08:00 - 09:30	09:40 - 11:10	11:20 - 12:50	13:00 - 14:30	14:40 - 16:10	16:20 - 17:50
Dimanche	TI tp S01 A/ Sider	BDD cours A16 L/ Zererga	SW cours A16 H/ Elbouhissi		TI tp S13 A/ Sider	
Lundi			AGL td S9B8 N/ djebbar	TW cours A16 A/ Sider		
Mardi		SW cours A16				

Figure 27 : Cr er un emploi du temps

IV.4.3. Espace enseignant

IV.4.3.1. Consultation d'un emploi du temps

Cette interface permet aux enseignants, une fois authentifiés, de consulter leurs emplois du temps personnel, de plus, elle offre la possibilité de choisir et consulter l'emploi du temps d'autre enseignant.

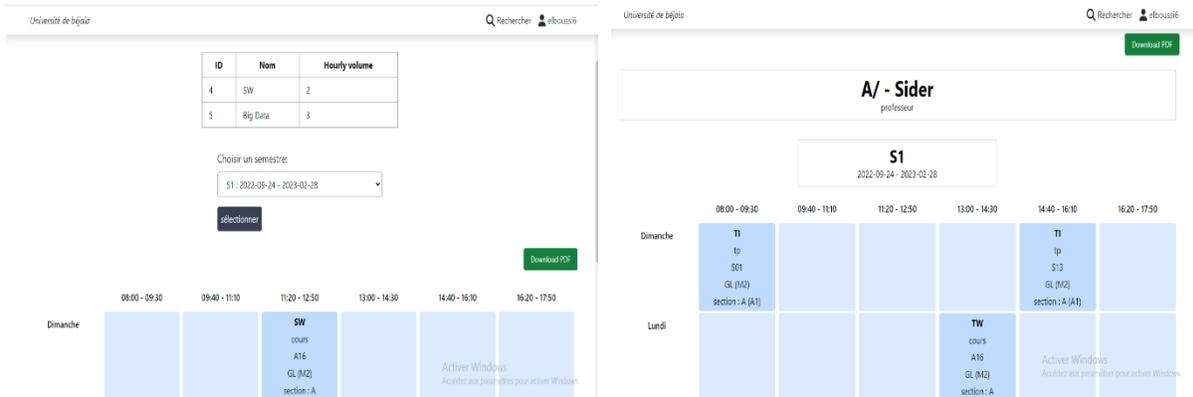


Figure 28 : Interface de Consultation d'un EDT

IV.4.4. Espace étudiant

IV.4.4.1. Consultation d'un emploi du temps

Cette interface permet aux étudiants, une fois authentifiés, de consulter leurs emplois du temps de leurs groupes, de plus, elle offre la possibilité de choisir et consulter l'emploi du temps d'autres groupes.

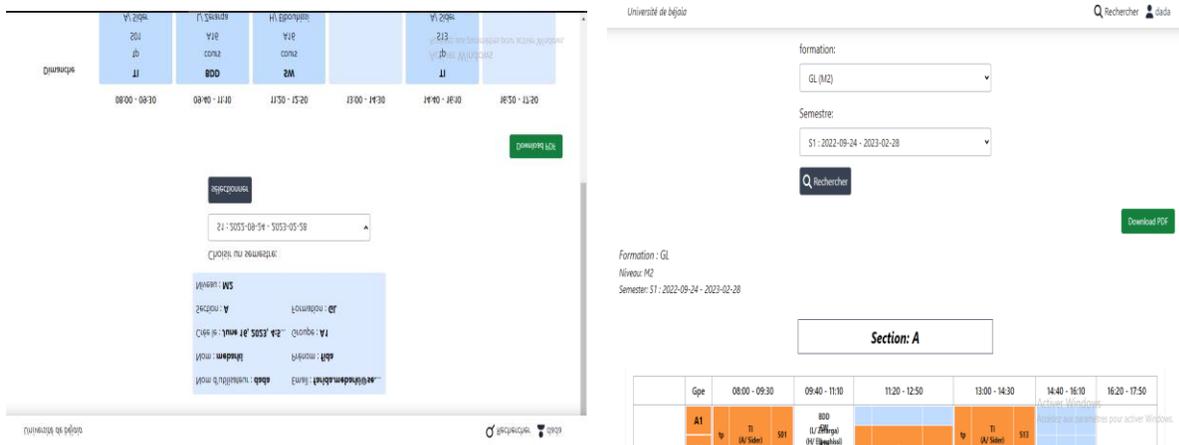


Figure 29 : Consulter les EDT

IV.4.4.2. Imprimer un emploi du temps

La dernière interface possède à un emploi du temps après que l'utilisateur lui enregistrée sous forme d'un PDF, et la possibilité de l'imprimer.

The screenshot shows a PDF viewer interface with the following details:

- File name: `semestre_S1-formation_GL-niveau_M2.pdf`
- Page: 1 / 2
- Zoom: 100%
- Metadata:
 - Formation : GL
 - Niveau: M2
 - Semester: S1 : 2022-09-24 - 2023-02-28
- Section: A
- Table with columns for time slots and rows for days (Dimanche, Lundi) and groups (A1, A2, A3, A4).

	Gpe	08:00 - 09:30	09:40 - 11:10	11:20 - 12:50	13:00 - 14:30	14:40 - 16:10	16:20 - 17:50				
Dimanche	A1	tp TI (A/ Sider)	S01	BDD (L/ Zargha) (H/ Elghisli) A16	td	AGL (N/ djebbar)	S1188	tp TI (A/ Sider)	S13		
	A2										
	A3										
	A4										
Lundi	A1			td	AGL (N/ djebbar)	S988	TW (A/ Sider)	A16			
	A2										
	A3										
	A4										

Figure 30 : Imprimer un emploi du temps

Conclusion

En conclusion, nous avons réussi à concevoir et développer une application performante qui permet la génération des emplois du temps pour les différentes promotions et enseignants, en garantissant l'absence de conflits au niveau des séances. Cette application a répondu de manière efficace aux besoins administratifs en résolvant les problèmes inhérents à la gestion des emplois du temps et en éliminant les conflits potentiels.

Conclusion générale

Notre projet a été réalisé dans le cadre d'un projet de fin d'études, pour objectif d'implémenter une solution informatisée pour la gestion d'emploi du temps au sein du département Informatique de l'université Bejaïa, afin de faciliter la vie de toutes personnes participant à la mise en place de dernière, et ainsi exploiter au mieux les ressources humaines et matérielles.

Notre travail se résume à la conception et la réalisation d'une application web permettant la gestion d'emploi du temps. Dans ce contexte, nous avons cherché à développer une application flexible, efficace et facile à exploiter.

La réalisation de ce travail nous a permis de bénéficier de plusieurs avantages. Dans un premier temps, nous avons pu approfondir nos connaissances théoriques et pratiques en rapport avec le Web et les techniques de programmation et dans un second temps, nous nous sommes familiarisés avec un certain nombre d'outils et de logiciels de développement Web.

✓ Perspectives

Nous souhaitons et espérons avoir résolu une grande partie du problème de la génération d'emploi du temps, en désirants que nous aurons l'occasion de le développer et l'enrichir pour plus d'efficacité, selon les perspectives suivantes :

- Envoyer des notifications pour informer les enseignants et les étudiants des modifications EDT et des mises à jour importantes ;
- Communication fluide entre les différents acteurs impliqués ;
- Améliorer l'interface du système.

Nous espérons que ce modeste mémoire soit un modèle pour les autres étudiants notamment dans la réalisation des applications web.

Bibliographie

- [1] T. O. Tigrine Youva, Conception et réalisation d'une application web pour la gestion des emplois du temps Cas : Mémoire de Fin d'Etudes, Tizi-Ouzou: Université Mouloud Mammeri, 2016/2017.
- [2] K. L. BRAHMI Saloua, Un outil semi-automatique pour la gestion des emplois du temps, des examens et des soutenances Cas d'étude : Mémoire de Fin de Cycle, Béjaia: Département d'informatique, Université A:mira, 2015/2016.
- [3] H. Bouziane, Un système automatique pour la génération des emplois du temps, Mémoire master en Informatique, Tlemcen: Université Abou Bakr Belkaid, 2012.
- [4] M. A. N. B. Melle Bendada Meriem, Gestion des Emplois du Temps Approche Graphique, Mémoire de fin d'études, Tlemcen: Université Abou Bakr Belkaid, 2015/2016.
- [5] P. M. Salama, Conception et réalisation d'un portail web à l'intention des différents acteurs burundais , Mémoire : Informatique et télécommunications, Université Lumière de Bujumbura, 2012.
- [6] I. Corporation, «Rational Software Architect 9.5.0,» 15 août 2015. [En ligne]. Available: [https://www.ibm.com/docs/fr/rational-soft-arch/9.5?topic=diagrams-use-case. .](https://www.ibm.com/docs/fr/rational-soft-arch/9.5?topic=diagrams-use-case.)
- [7] J. G. D. Gabay, UML2 analyse et conception, Edition Dunod, 2008.
- [8] L. AUDIBERT, «UML 2 De l'apprentissage à la pratique,» 12 janvier 2009. [En ligne]. Available: <https://laurent-audibert.developpez.com/Cours-UML/>.
- [9] «Qu'est-ce qu'un processus unifié (up) ?,» 2023. [En ligne]. Available: <https://fr.theastrologypage.com/unified-process>.
- [10] «UP : Unified Process,» 2004. [En ligne]. Available: <https://sabricole.developpez.com/uml/tutoriel/unifiedProcess>.
- [11] S. Crozat, «Passage UML-Relationnel,» [En ligne]. Available: <https://librecours.net/module/bdd0/uml-r?V=print>.
- [12] «Application Web,» Idéematic, 2023. [En ligne]. Available: <https://www.ideematic.com/dictionnaire-digital/application-web>.
- [13] «StarUML,» Public Domain, 22 février 2011. [En ligne]. Available: <https://air.imag.fr/index.php/StarUML>.
- [14] O. A. SAICHE Cylia, , Conception et réalisation d'une application web pour la gestion des étudiants d'une école privée, Mémoire de fin de cycle, Bejaia: Université A/Mira , 2014 /2015.

- [15] G. Swinnen, Apprendre à programmer avec Python3, 2000-2012.
- [16] R. Juillet, «Développement web : les avantages de Python,» 29 août 2022. [En ligne]. Available: <https://www.bocasay.com/fr/developpement-web-avantages-python/#:~:text=Python%20est%20devenu%20un%20langage,web%20et%20la%20programmation%20hardware>.
- [17] «Resources for Developers, by Developers,» [En ligne]. Available: <https://developer.mozilla.org/fr/docs/Learn/Serverside/Django/Introduction>. [Accès le Mai 2023].
- [18] «Qu'est-ce que Javascript ?,» Amazon Web Services, [En ligne]. Available: <https://aws.amazon.com/fr/what-is/javascript/>. [Accès le Mai 2023].

Résumé

L'objectif de ce travail est d'informatiser la gestion des emplois du temps au sein du département d'Informatique de l'Université de Bejaïa. Cette informatisation se matérialise par la mise en place d'un système qui gère le fonctionnement du département informatique. Ce système vise à faciliter la gestion des emplois du temps pour l'administration tout en maintenant un contact permanent avec les utilisateurs. Pour ce faire, nous avons utilisé le langage de modélisation UML pour schématiser la solution, ainsi que le langage de programmation Python avec le Framework Django.

Abstract

The objective of this work is to computerize the schedules planning for the Department of Computer Science at the University of Bejaia. This computerization is achieved by implementing a system that handles the functions of the Department. The purpose of this system is to facilitate the schedules planning for the administration and maintain constant communication with the users. To accomplish this, we utilized the UML modeling language to diagram the solution, along with the Python programming language and the Django framework.