

République Algérienne Démocratique et Populaire  
Ministère de L'enseignement Supérieur et de la Recherche Scientifique  
**Université A/Mira de Bejaïa**  
**Faculté des Sciences Exactes**  
**Département Informatique**



# Mémoire de Fin de Cycle

En vue de l'obtention du diplôme Master professionnel en  
Informatique

**Option** : Administration et Sécurité des Réseaux

**Thème**

**Un outil semi-automatique pour la  
gestion des emplois du temps, des  
examens et des soutenances**  
**Cas d'étude :**  
**département d'informatique**

**Réalisé par :**

M<sup>lle</sup> BRAHMI Saloua.

M<sup>lle</sup> KETFI Lamia.

**Devant le jury composé de :**

Président : M<sup>r</sup> AMROUN Kamel.

Examinatrice1 : M<sup>me</sup> TIGHIDET Soraya.

Examinatrice2 : M<sup>me</sup> BELKHIRI Louiza.

Promoteur : M<sup>r</sup> FARAH Zoubeyr.

PROMOTION 2015-2016

## Dédicaces

*Je dédie ce modeste travail  
À mes chers parents, qui sont la cause de mon existence dans cette vie,  
pour leur soutien, leur patience et leur amour qui m'ont donné la force pour continuer  
mes études  
Aucune dédicace, aucun mot ne pourrait exprimer à leur juste valeur la gratitude et  
l'amour que je vous porte,  
Aujourd'hui, Je mets entre vos mains, le fruit de longues années d'études, À mes soeurs  
Halima, Badoura et Sabrinna et à mon frère Salim ,  
Je leurs souhaite une bonne continuation dans leurs études  
À mes amis Faouzi, Nassim, Fouad et à tous mes amis du proche ou du loin sans  
exception, dont la liste est longue du groupe de choc.  
Merci.*

**Lamia**

*À mes très chers parents,  
À mes frère : Fayçel, Mounir, Khirddine et ma sœur Chahrazad, mes grands parents,  
À ma précieuse famille, mes cousins et cousines, Oncles et tantes,  
À mes amis et collègues,  
Et à toutes les personnes que j'ai connues et qui m'ont aidées, un grand MERCI à tous.*

**Saloua**

## Remerciements

**L**OUANGE À Dieu, le miséricordieux, sans lui rien de tout cela n'aurait pu être.

Au terme de la rédaction de ce mémoire, nous tenons à remercier notre encadreur Mr FARAH.Z pour ses précieux conseils et son aide durant toute la période du travail.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

On n'oublie pas de remercier nos parents, nos frères et sœurs pour leur soutien moral et physique.

Sans oublier aussi de remercier Mr OUZEGGANE Redouane et Mme BELKHIRI Louiza.

Tous nos remerciements à : Rafik, Fatah, Yassamina, Souad, Lydia, Assia pour les merveilleux moments qu'on a passé ensemble, et à tous nos amis sans exception.

*L.S*

## Résumé

Dans ce mémoire nous avons réalisé un outil semi-automatique pour la gestion des emplois du temps, des examens et des soutenances au sein du département d'informatique de l'université de Bejaïa.

Pour ce qui est de la conception nous avons choisi UML2 pour sa simplicité, richesse et performance en matière de conception.

Concernant l'implémentation de notre base de données, nous avons utilisé MySQL comme serveur de gestion de base de données.

Nous avons choisi les langages de programmation java sous NetBeans.

***Mots clés : Emplois du temps, Soutenances, Examens, Planification, UML2, MySQL, JAVA, NetBeans.***

## Abstract

In this paper we performed semi-automated tool support for managing timetables, exams and defenses in the Computer Science Department of the University of Béjaïa.

In terms of design we chose UML2 for its simplicity, richness and performance in the design. Concerning the implementation of our database we used MySQL as a database management server.

We chose the programming language Java in NetBeans for the realization of our application.

***Keywords : Timetables, Defenses, Testing, Planning, UML2, MySQL, Java, NetBeans.***

# Table des matières

<b>Dédicaces</b>	<b>i</b>
<b>Remerciements</b>	<b>ii</b>
<b>Résumé</b>	<b>iii</b>
<b>Table des matières</b>	<b>iv</b>
<b>Liste des figures</b>	<b>vi</b>
<b>Liste des tableaux</b>	<b>vii</b>
<b>Liste des Acronymes</b>	<b>viii</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Problématique</b>	<b>4</b>
1.1 Présentation de l'organisme d'accueil : "département d'Informatique" . . . . .	4
1.2 Missions du département . . . . .	5
1.3 Problématique . . . . .	5
1.3.1 Définition d'un planning . . . . .	6
1.3.2 A Quoi sert un planning . . . . .	6
1.4 La pédagogie . . . . .	6
1.4.1 Etude du système pédagogique . . . . .	6
1.5 Description du problème à résoudre . . . . .	8
1.5.1 Suivi pédagogique des enseignants . . . . .	9
1.5.2 Les contraintes . . . . .	10
1.6 Conclusion . . . . .	11
<b>2 Analyse et conception</b>	<b>12</b>
2.1 Présentation d'UML . . . . .	12
2.1.1 Points forts d'UML . . . . .	13
2.1.2 Présentation générale des diagrammes d'UML . . . . .	13
2.2 Processus unifié . . . . .	15
2.2.1 Définition . . . . .	15
2.2.2 Caractéristiques du processus unifié . . . . .	15
2.2.3 Différentes phases du processus unifié . . . . .	16
2.2.4 Activités du processus unifié . . . . .	17

---

2.3	Modélisation fonctionnelle de notre système . . . . .	17
2.3.1	Description des cas d'utilisations . . . . .	17
2.3.2	Description textuelle des cas d'utilisation . . . . .	18
2.3.3	Visual paradigm . . . . .	19
2.4	Diagramme de cas d'utilisation de l'application . . . . .	20
2.4.1	Réalisation des diagrammes de séquences . . . . .	23
2.4.2	Réalisation des diagrammes d'activité . . . . .	29
2.5	Modélisation structurelle de notre application . . . . .	32
2.5.1	Dictionnaire de données épuré . . . . .	32
2.5.2	Diagramme de classes . . . . .	34
2.5.3	Les concepts d'un diagramme de classe . . . . .	34
2.5.4	Relations entre les classes . . . . .	35
2.5.5	Diagramme de classes de l'application à réaliser . . . . .	36
2.6	Modèle Relationnel de données . . . . .	37
2.6.1	Règles de dérivation du modèle relationnel à partir d'un diagramme de classes . . . . .	37
2.7	Conclusion . . . . .	38
<b>3</b>	<b>Implémentation et réalisation</b>	<b>39</b>
3.1	Outils et langages de développements . . . . .	39
3.1.1	Wamp Server . . . . .	39
3.1.2	Java . . . . .	42
3.1.3	NetBeans . . . . .	42
3.1.4	Jdk . . . . .	43
3.1.5	EDI . . . . .	44
3.2	Présentation de quelques interfaces de notre application . . . . .	44
3.2.1	Page d'authentification . . . . .	44
3.2.2	Ajouter enseignant . . . . .	46
3.2.3	Modifier enseignant . . . . .	47
3.2.4	Génération d'un emplois du temps . . . . .	48
3.2.5	Planning des examens . . . . .	49
3.2.6	Ajouter un thème . . . . .	49
3.2.7	Ajouter une soutenance . . . . .	50
3.3	Conclusion . . . . .	51
	<b>Bibliographie</b>	<b>52</b>

# LISTE DES FIGURES

2.1	<i>Les types des diagrammes UML2[8]. . . . .</i>	15
2.2	<i>Les phases du processus unifié[8]. . . . .</i>	17
2.3	<i>Diagramme générale de cas d'utilisation. . . . .</i>	20
2.4	<i>Diagramme de cas d'utilisation pour la gestion des emplois du temps. . . . .</i>	21
2.5	<i>Diagramme de cas d'utilisation pour la gestion des examens. . . . .</i>	22
2.6	<i>Diagramme de cas d'utilisation pour la gestion des soutenances. . . . .</i>	23
2.7	<i>Diagramme de séquence du cas d'utilisation : Authentification. . . . .</i>	25
2.8	<i>Diagramme de séquence du cas d'utilisation : gestion des enseignants. . . . .</i>	27
2.9	<i>Diagramme de séquence du cas d'utilisation : gestion des locaux. . . . .</i>	28
2.10	<i>Diagramme de séquence du cas d'utilisation : afficher emplois du temps par section. . . . .</i>	29
2.11	<i>Diagramme d'activité : Authentification. . . . .</i>	30
2.12	<i>Diagramme d'activité : Ajouter un enseignant. . . . .</i>	30
2.13	<i>Diagramme d'activité : Modifier un enseignant. . . . .</i>	31
2.14	<i>Diagramme d'activité : Supprimer un enseignant. . . . .</i>	31
2.15	<i>Représentation graphique d'une classe. . . . .</i>	35
2.16	<i>Diagramme de classes de l'application. . . . .</i>	36
3.1	<i>les composants de wamp . . . . .</i>	40
3.2	<i>MySQL . . . . .</i>	40
3.3	<i>Exemple de tables de bases de données . . . . .</i>	41
3.4	<i>NetBeans . . . . .</i>	42
3.5	<i>Jdk . . . . .</i>	43
3.6	<i>Fenêtre d'authentification . . . . .</i>	45
3.7	<i>Message d'erreur . . . . .</i>	45
3.8	<i>Ajouter enseignant . . . . .</i>	46
3.9	<i>Modifier enseignant . . . . .</i>	47
3.10	<i>Génération d'un emplois du temps . . . . .</i>	48
3.11	<i>Planning des examens . . . . .</i>	49
3.12	<i>Ajouter thème . . . . .</i>	50
3.13	<i>Ajouter une soutenance . . . . .</i>	50
3.14	<i>Membre de jury . . . . .</i>	51

# LISTE DES TABLEAUX

2.1	Description du cas d'utilisation : Authentification . . . . .	18
2.2	Description du cas d'utilisation : Gestion des enseignants. . . . .	19
2.3	Dictionnaire de données épuré. . . . .	33

# Liste des Acronymes

<b>API</b>	<b>A</b> pplication <b>P</b> rogramming <b>I</b> nterface .
<b>DEUA</b>	<b>D</b> iplôme <b>E</b> tude <b>U</b> niversitaire <b>A</b> ppliquée.
<b>DMax</b>	<b>D</b> urée <b>M</b> aximal.
<b>DMin</b>	<b>D</b> urée <b>M</b> inimal .
<b>EDI</b>	<b>E</b> nvironnement <b>D</b> éveloppement <b>I</b> ntégré .
<b>HMax</b>	<b>H</b> nergy <b>M</b> aximale
<b>HMin</b>	<b>H</b> eure <b>M</b> inimal
<b>IDE</b>	<b>I</b> ntegrate <b>D</b> éveloppement <b>E</b> nvironnement.
<b>JAR</b>	<b>J</b> ava <b>A</b> Rchive.
<b>JDB</b>	<b>J</b> ava <b>D</b> ata <b>B</b> ase.
<b>JDBC</b>	<b>J</b> ava <b>D</b> ata <b>B</b> ase <b>C</b> onnectivity.
<b>JDK</b>	<b>J</b> ava <b>D</b> éveloppement <b>K</b> it.
<b>JPA</b>	<b>J</b> ava <b>P</b> ersistence <b>A</b> PI.
<b>JVM</b>	<b>J</b> ava <b>V</b> irtual <b>M</b> achine.
<b>LMD</b>	<b>L</b> icence <b>M</b> aster <b>D</b> octorat.
<b>TD</b>	<b>T</b> ravaux <b>D</b> irigées.
<b>TP</b>	<b>T</b> ravaux <b>P</b> ratiques.
<b>UML</b>	<b>U</b> nified <b>M</b> odeling <b>L</b> anguage .
<b>UP</b>	<b>U</b> nified <b>P</b> rocess.

# Introduction générale

Les situations où un planning est utile sont nombreuses. Elles justifient l'existence de différentes formes de plannings dans un même système : plannings à court, moyen et à long terme. L'intérêt d'élaborer des plannings s'est vu accroître de plus en plus.

Il existe différentes versions de planification au niveau des établissements universitaire mais la plupart peuvent être classifié en trois grandes catégories qui sont : emploi du temps universitaire, planning des examens et planning des soutenances.

Le département d'informatique de l'université A. Mira-Bejaïa comme tous autres départements gère les emplois du temps au début de chaque semestre de chaque année, comme il gère aussi la planification des examens et soutenances, le problème de planification consiste à définir un certain nombre d'affectations qui permet d'assigner plusieurs ressources sur une période de temps en respectant des contraintes imposées.

Dans ce contexte, nous allons nous intéresser au problème qui consiste à développer une application (outil semi-automatique) d'assistance pour la gestion des emplois du temps, gestion des examens et gestion des soutenances pour le département d'informatique de l'université de Bejaïa. En accompagnant tous les acteurs impliqués dans ce processus.

Le premier point que nous avons traité est la gestion des emplois du temps .la génération d'un emploi du temps est un travail long et difficile à assuré en chaque début de semestre. C'est pour cette raison que la semi-automatisation de la gestion d'emplois du temps, est une tâche d'une grande importance car, en effet elle permet de gagner beaucoup d'heures de travail pour le département, pour les enseignants et pour les étudiants, et de fournir des solutions optimales avec satisfaction d'un maximum de contraintes.

La difficulté de cette tâche fastidieuse et répétitive, qui fait généralement intervenir de nombreux éléments d'information, est liée à la nature des contraintes à satisfaire.

Le deuxième problème traité est celui de la planification des soutenances. Où nous avons été contraints d'énumérer :

En premier lieu, l'échange de l'information et de la communication entre les étudiants et les encadreurs, le contrôle, le suivi et l'évaluation des mémoires.

Et en deuxième lieu, assurer la gestion des soutenances, préciser la liste des thèmes, la liste des étudiants, leurs membres de jury (promoteur, Co promoteur, président, examinateur1 et examinateur2), ainsi que la date, l'heure et le lieu de chaque soutenance.

Le troisième problème traité est le processus de planification des examens, l'application doit assurer une partie de paramétrage qui intégrera l'examen, sa durée et la liste des enseignants surveillants.

L'existence d'un horaire respectant toutes les contraintes n'est pas garantie dans le cas général. En réalité, il s'agit de trouver un horaire aussi bon que possible dans un temps de calcul raisonnable tout en évitant les conflits, le nombre de ces conflits doit être minimisé pour que l'horaire soit jugé satisfaisant par les différentes parties concernées. (Direction de l'université, enseignants, étudiants,..etc.)

**Le présent rapport est composé de trois chapitres. Son organisation est comme suit :**

Dans le premier chapitre, nous présentons dans un premier temps un aperçu général sur notre organisme d'accueil qui est le département d'informatique, ses missions. Dans un second temps, nous donnons une vision globale sur la pédagogie ainsi qu'on a définis toutes les contraintes physiques d'un part et les contraintes pédagogiques d'autre part pour les trois catégories de planification : emplois du temps, examens et soutenances.

La conception et la modélisation du processus de gestion des emplois du temps, des examens et des soutenances au niveau de départements d'informatique, en utilisant UML comme langage de modélisation et le processus unifié qu'on a suivi durant tout le processus de développement, fera l'objet du deuxième chapitre.

Le troisième chapitre consacré à la réalisation de notre application, dans un premier lieu nous donnons les outils de développement utilisés, tel que le langage java, le SGBD

MySQL et NetBeans. Dans un deuxième lieu, nous présentons quelques interfaces de notre application avec des fonctionnalités.

Enfin, notre travail s'achève par une conclusion générale résumant les grands points qui ont été abordés ainsi que les perspectives que nous souhaitons accomplir dans la future.

# 1

## Problématique

### Introduction

Dans ce chapitre, nous présentons une étude générale du système pédagogique universitaire.

Décrivons notre projet qui consiste à développer un outil semi-automatique pour la gestion des emplois du temps, gestion des examens et gestion des soutenances au sein du département d'informatique. afin d'affecter à chaque enseignant une charge hebdomadaire répartie sur un ensemble de créneaux en l'affectant à des groupes ou bien sections et à des locaux afin d'éviter toute sorte de chevauchement.

### 1.1 Présentation de l'organisme d'accueil : "département d'Informatique"

Le département d'informatique de l'UAMB avait assuré, depuis sa création en fin de l'année 2002, trois types de formation, à savoir la formation du système classique des deux cycles de la graduation, le cycle court (2003-2006) en vue de l'obtention du diplôme de DEUA en informatique et le cycle long (2003-2012) en vue de l'obtention du diplôme d'ingénieur d'état en informatique. A partir de l'année 2003 le département d'informatique a

commencé à prendre en charge la formation du système L.M.D (Licence, Master et Doctorat) jusqu'à l'heure actuelle et qui est le seul système fonctionnel actuellement.

## 1.2 Missions du département

Le département assure un suivi pédagogique des cycles de graduation et de post-graduation ; gérer la scolarité des étudiants (inscription, évaluation, présence aux enseignements, absences et sanctions) et des enseignants (matières enseignées, affectation des modules, volume horaire, emplois du temps, planning des examens, gestion des soutenances, absences, saisie des notes, délibérations ...).

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

- Licence.
- Master 1 Socle commun.
- Master 1 Professionnel.
- Master 1 Recherche.
- Master 2 Administration et sécurité des réseaux.
- Master 2 Génie logiciel.
- Master 2 Réseaux et systèmes distribués.

## 1.3 Problématique

La problématique de notre département est la planification des emplois du temps, des examens et des soutenances.

Pour chaque niveau d'étude, les étudiants sont rassemblés en sections et chaque section en groupes. Ces derniers ont un planning hebdomadaire, ou chaque module est assuré sous forme de séances de cours, TP ou TD dans des locaux répartis au début de chaque semestre.

Pour chaque examen les étudiants sont rassemblés en sections ou en groupes, dans des amphithéâtres ou des salles de TD, où chaque local est surveillée par un ou plusieurs enseignants.

Généralement les soutenances des étudiants de L3 et de M2 s'effectuent dans les Halls A et B ou les salles de TD des blocs d'enseignement.

### 1.3.1 Définition d'un planning

Les plannings sont des calendriers de travail, où figurent à la fois le temps, l'affectation du personnel, les jours et les horaires de travail ainsi que les congés et repos [1].

### 1.3.2 A Quoi sert un planning

Depuis le début des années 80, la gestion des ressources humaines a été reconnue comme une activité stratégique pour l'entreprise. Avec cette reconnaissance, l'intérêt d'élaborer des plannings s'est vu accroître de plus en plus car ils permettent :

- aux entreprises exerçant une activité continue ou quasi-continue de répartir convenablement leur personnel (compagnies aériennes, entreprises de transports, hôpitaux, etc...).
- à toutes les entreprises de surmonter leur exigences de productivité et de mieux gérer les présences et absences de leur personnel.

Les situations où un planning est utile sont nombreuses. Elles justifient l'existence de différentes formes de plannings dans un même système : plannings à court, moyen et à long terme [2].

## 1.4 La pédagogie

Pédagogie, théorie de l'enseignement, qui s'est imposée à partir du XIXe siècle comme science de l'éducation, ou didactique expérimentale, et s'interroge aujourd'hui sur les conditions de réception du savoir, sur le contenu et l'évaluation de celui-ci, sur le rôle de l'éducateur et de l'étudiant dans le processus éducatif et, plus globalement, sur les finalités de cet apprentissage, indissociable d'une norme sociale et culturelle[2].

### 1.4.1 Etude du système pédagogique

Il s'avère que ce système ne possède pas de particularité pédagogique ciblée et possède de nombreuses similarités avec le fonctionnement d'autres universités. Ainsi, l'UAMB est une université qui regroupe différentes formations du LMD. Les étudiants s'inscrivent en début de chaque année universitaire, c'est -à-dire généralement au début de septembre. Mais cela peut varier d'une formation à l'autre. Le programme pédagogique de chaque formation est connu. Ce programme précise les matières à suivre, leurs volumes horaires et quelques informations pédagogiques (répartition en cours, TD et TP). Selon les besoins pédagogiques et les conditions physiques des ressources, chaque formation est structurée en sections qui peuvent eux-mêmes être scindés en groupes [3].

#### 1.4.1.1 L'activité pédagogique

L'activité pédagogique est modélisée à l'aide de trois entités : les locaux, les enseignements et les modules. D'une manière générale, les enseignants, les groupes, les matériels constituent les ressources de l'enseignement. Les matériels sont les ressources qui seront attribués aux séances de l'enseignement. Un enseignement peut être assuré simultanément par plusieurs enseignants. C'est le cas, par exemple de certains TP d'informatique. Les modules sont des ensembles d'enseignements[3].

#### 1.4.1.2 Les ressources

Les ressources considérées sont les entités physiques nécessaires à l'élaboration des emplois du temps, des examens et des soutenances. Il s'agit des locaux, des enseignants, des étudiants (groupes/sections) et des matériels. Les ressources sont caractérisées par des données abstraites et des données spécifiques. Les données abstraites caractérisent chaque ressource et sont constituées d'un code, qui permet de la différencier des autres ressources, son calendrier qui précise quels sont les jours de disponibilité et d'indisponibilité et sa description. En plus de ces données abstraites, chaque ressource possède des caractéristiques spécifiques qui dépendent du type de la ressource. L'intérêt de distinguer ces deux types de caractéristiques est que l'outil peut très facilement évoluer pour prendre en compte de nouveaux types de ressources.

Dans la suite nous décrivons les caractéristiques spécifiques des ressources considérées dans l'étude.

##### - Les ressources de type " salle "

Une salle est un lieu dans lequel sont assurés des enseignements. Le type d'une salle est une indication sur le type d'enseignement qu'on peut y faire.

##### - Les ressources de type " enseignant "

L'enseignant désigne une personne pouvant assurer des enseignements. Chaque enseignant est caractérisé par : son identifiant, son nom et son prénom, son grade, sa spécialité, affiliation, son statut, son num\_tel et son e-mail.

##### - Les ressources de type " groupe "

En général les étudiant d'une filière sont décomposer en section et chaque section est décomposé en plusieurs groupes.

##### - Les ressources de type " matériel "

Pour qu'un enseignement puisse se dérouler, l'enseignant utilise un matériel pédagogique. Généralement ce matériel est limité à un tableau, des craies, éventuellement un rétroprojecteur et parfois un vidéo projecteur. Pour des enseignements particuliers, l'enseignant peut avoir besoin de matériel spécifique.

#### - Les entités temporelles

Pour modéliser le temps, les entités : date, heure, durée, créneau et calendrier sont définies. Une date désigne un instant défini par un triplet (jour, mois, année). A partir de ce triplet, on détermine la valeur qui lui est associée sur l'axe des jours. Pour avoir un grain plus fin, la notion d'heure est utilisée. Il s'agit d'un nombre entier compris entre la valeur minimale HMin et la valeur maximale HMax. Ces deux nombres correspondent à des heures par rapport à une date. Par exemple, dans notre cas, nous avons choisi HMin=8h00 et HMax=17h50 par défaut. Une durée est un nombre compris entre DMin et Dmax = HMax - HMin. DMin représente la plus petite unité temporelle disponible.

## 1.5 Description du problème à résoudre

Dans un établissement éducatif, un ensemble d'étudiants groupés sous une structure hiérarchique (filiales, promotions, sections, groupes, ...) sont censés avoir un ensemble d'enseignements qui se répètent périodiquement, chacun de ces enseignements s'étend sur une durée de temps, dont l'unité élémentaire est la période.

Résoudre le problème de l'emploi du temps, examens et soutenances revient à affecter à chacun de ces enseignements un nombre de périodes consécutives égal à la durée qu'il exige, un local dont le type et la capacité sont convenables, et un enseignant apte à assurer le module concerné par l'enseignement de façon à prévenir les conflits sur les enseignants, sur les étudiants et sur les locaux.

Dans notre cas, le problème de l'emploi du temps, d'examens et de soutenances étudié est celui de département d'informatique où les responsables pédagogiques ont besoin chaque année d'établir une nouvelle planification des différentes promotions en essayant au mieux de satisfaire les contraintes " humaines " des enseignants et des étudiants, les contraintes pédagogiques imposées par la progression des enseignements et en tenant compte des contraintes " physiques " liées aux ressources matérielles (les locaux).

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 de chaque formation est connu à priori.

Ce programme précise les modules à suivre, leurs volumes horaires et quelques informations pédagogiques (répartition en cours, travaux dirigés, travaux pratiques etc. . .).

Le programme pédagogique des soutenances de chaque fin de cycle est connu à priori. Ce programme précise la liste des thèmes, la liste des étudiants, leurs membres de jury (promoteur, Co promoteur, président, encadreur1 et encadreur2), la date et l'heure de chaque soutenance.

Le programme pédagogique des examens de chaque formation est connu à priori. Ce programme précise les modules à faire passer examen, leurs volumes horaires ou maximum de deux heures et la liste des enseignants surveillants.

Selon les besoins pédagogiques et les conditions physiques des ressources, chaque formation est structurée en promotions, en sections, et en groupes. En résumé, les données du problème à résoudre sont constituées par :

- Un ensemble de créneaux horaires étalés sur une semaine de six jours, du samedi au jeudi avec un nombre six périodes. La durée d'une période est d'une heure trente minutes pour les emplois du temps et de maximum deux heures pour les soutenances et les examens.
- Un ensemble de promotions ou groupes d'étudiants.
- Un ensemble de cours, TD ou TP à programmer dans la semaine.
- Un ensemble de locaux (salles, amphis).

### **1.5.1 Suivi pédagogique des enseignants**

Un enseignant est chargé d'assurer des séances de cours magistral, de TD ou de TP repartis au cours du jour. Une journée pédagogique est composé de six séances d'une heure et demie chacune.

On rencontre des enseignants permanents, associés ou vacataires. Chaque enseignant possède son propre grade de professeur, maitre de conférences, maitre-assistant, ou enfin professeur-ingénieur.

Le volume horaire effectif d'un enseignant doit être au-delà de neuf heures par semaine. La charge pédagogique d'un enseignant s'étale sur trois jours au maximum dans la semaine. Pour créer un emploi du temps, on répartit les séances dans la semaine selon la situation de l'enseignant et l'occupation des salles ou d'autres contraintes [4].

## 1.5.2 Les contraintes

### 1.5.2.1 Les contraintes physiques

Ces contraintes ne doivent pas être violées sinon cela conduirait à des situations conflictuelles.

- Une ressource ne peut pas être occupée en même temps dans deux séances différentes.
- Dès qu'une ressource est occupée par une séance, toutes ses ressources filles sont également occupées par la même séance et cela récursivement (il s'agit d'une contrainte liée à la hiérarchisation).
- On ne peut pas mettre plus d'étudiants qu'il n'y a de places dans un local.
- Le volume horaire total des séances d'un enseignement ne peut pas dépasser le volume prévu.

### 1.5.2.2 Les contraintes pédagogiques

Des contraintes pédagogiques ont pu être identifiées. Elles se différencient des contraintes physiques par le fait qu'elles peuvent éventuellement être violées. Typiquement ces contraintes sont utilisées pour exprimer ce que doit être un " bon " planning. Voici quelques exemples de ces contraintes :

#### **Les contraintes pédagogiques cas d'emplois du temps**

- Un enseignant ne peut pas assurer deux enseignements en même temps.
- Pendant un créneau donné, un local ne peut pas être utilisé que pour un seul enseignement.
- 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 vice versa.
- Un TP n'aura pas lieu dans un amphithéâtre ou dans une salle TD.
- Un TD n'aura pas lieu dans un amphithéâtre ou dans une salle TP.
- Un cours n'aura pas lieu dans une salle TP.

#### **Les contraintes pédagogiques cas de planning des examens**

- Un enseignant ne peut pas effectuer deux surveillances en même temps.
- Un niveau ou une spécialité (option) ne peut pas avoir deux examens en même temps.
- Un examen ne peut s'effectuer dans salle TP.
- Pendant un créneau donné, un local sera occupé que pour un seul examen.

- Un niveau ou une spécialité (option) ne peut pas avoir plus d'un examen pendant la journée.
- Un module ne peut pas avoir plus d'un examen.

### **Les contraintes pédagogiques cas des soutenances**

- Un enseignant ne peut pas participer à deux soutenances en même temps.
- Pendant un créneau donné, un local sera occupé que pour une seule soutenance.
- Un thème ne peut pas avoir deux soutenances.
- Un étudiant ne peut pas avoir deux thèmes.

## **1.6 Conclusion**

Durant l'analyse de l'existant nous avons pu recenser toutes les informations nécessaires et indispensables pour l'accomplissement de notre projet. Ces informations tirées entre autre à partir de l'étude du système pédagogique nous aident énormément à entamer notre travail concernant le chapitre suivant sur l'analyse et la conception de notre système en utilisant le langage UML (Unified Modeling Language).

# 2

## Analyse et conception

### Introduction

L'étude du système existant est une étape primordiale vers la conception d'une application performante répondant bien aux besoins des utilisateurs. En effet, l'analyse nous permet de prendre connaissance du domaine dans lequel l'organisme souhaite améliorer son fonctionnement, de soulever les anomalies et insuffisances rencontrées et de suggérer dans la limite du possible, des solutions bénéfiques.

La conception nous permet de décrire complètement le futur système à l'aide des différents modèles de données et de traitements.

La nouvelle application doit permettre d'atteindre les nouveaux objectifs fixés d'après les souhaits exprimés.

### 2.1 Présentation d'UML

L'UML est un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue [5][6].

### 2.1.1 Points forts d'UML

- UML est un langage formel et normalisé (gains de pression, stabilité, encourage l'utilisation d'outils).
- UML est un support de communication performant :
  - Il cadre l'analyse.
  - Il facilite la compréhension abstraite.
- Son caractère polyvalent et sa souplesse en font un langage universel.[6]

### 2.1.2 Présentation générale des diagrammes d'UML

UML [6] s'articule autour de treize diagrammes, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel. Ces diagrammes sont repartis en deux grands groupes :

#### 1. Diagrammes structurels

Ils représentent l'aspect statique d'un système (classes, objets, composants, . . .)[7][8]. Dans ce qui suit, nous décrivons les six diagrammes structurels d'UML.

- **Diagramme de classes**

Il représente la description statique du système où chaque classe a une partie données et une partie traitements. C'est le diagramme pivot de l'ensemble de la modélisation d'un système.

- **Diagramme d'objets**

Permet la représentation d'instances des classes et les liens qui les relient.

- **Diagramme de composants**

Il représente les différents constituants du logiciel au niveau de l'implémentation d'un système.

- **Diagramme de déploiement**

Décrit l'architecture technique d'un système en se basant sur la répartition des composants dans la configuration d'exploitation.

- **Diagramme de paquetage**

décrit l'ensemble du système structuré en paquetage. Chaque paquetage est un ensemble homogène d'éléments du système.

- **Diagramme de structure composite**

Permet de décrire la structure interne d'un ensemble complexe et composé, et met

aussi l'accent sur les liens entre les sous-ensembles qui collaborent.

## 2. Diagrammes de comportement

Ces diagrammes représentent la partie dynamique d'un système réagissant aux événements et permettant de produire les résultats attendus par les utilisateurs.

- **Diagramme des cas d'utilisation**

Destiné à représenter les besoins des utilisateurs par rapport au système. C'est l'un des diagrammes les plus structurants dans l'analyse d'un système.

- **Diagramme d'état transition**

Montre les différents états des objets en réaction aux événements.

- **Diagramme d'activités**

Donne une vision des enchainements des activités propres à une opération ou à un cas d'utilisation.

- **Diagramme de séquences**

Permet de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations.

- **Diagramme de communication**

C'est une autre représentation des scénarios de cas d'utilisation qui met plus l'accent sur les objets et les messages échangés.

- **Diagramme globale d'interaction**

Fournit une vue générale des interactions décrites dans le diagramme de séquence et des flots de contrôle décrits dans le diagramme d'activités.

- **Diagramme de temps**

Permet de représenter les états et les interactions d'objets dans un contexte ou le temps à une forte influence sur le comportement du système à gérer.

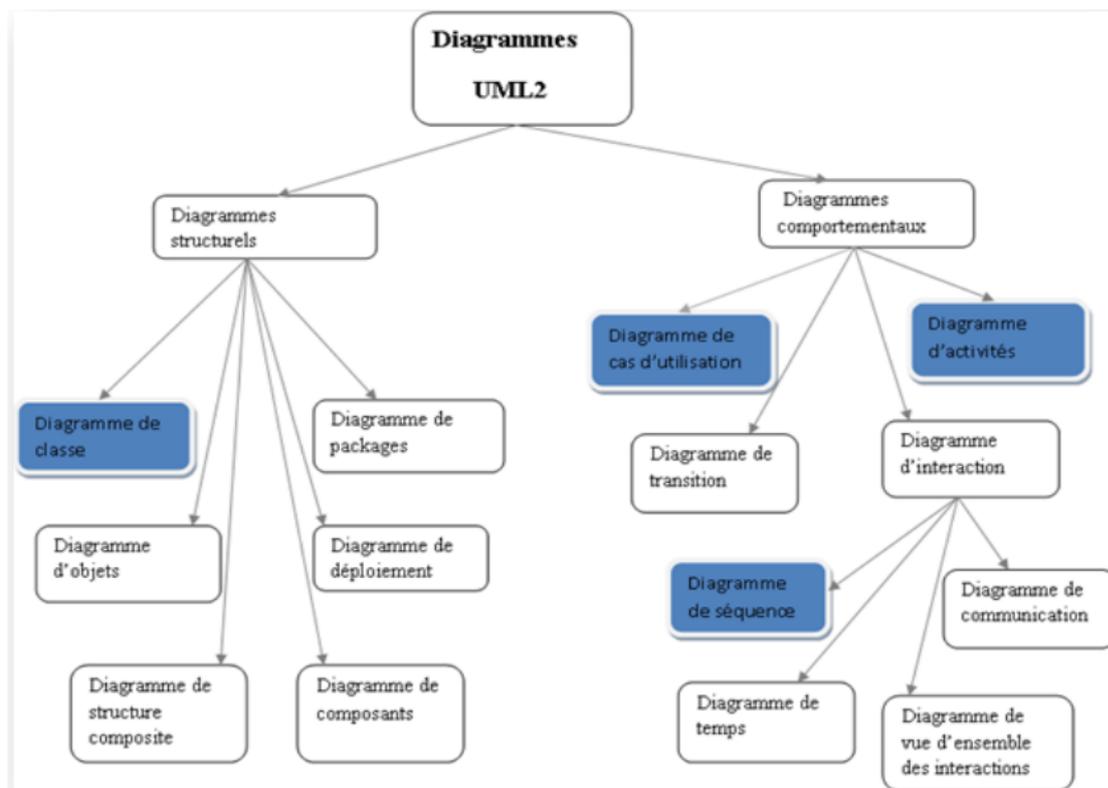


FIG. 2.1 – Les types des diagrammes UML2[8].

## 2.2 Processus unifié

### 2.2.1 Définition

Le Processus Unifié (UP pour Unified Process) est un processus de développement de logiciels. Ce dernier est une trame commune des meilleures pratiques de développement. La définition d'un processus UP est constituée de plusieurs disciplines d'activité de production et de contrôle de cette production [9].

### 2.2.2 Caractéristiques du processus unifié

Tout processus UP répond aux caractéristiques ci-après[9][10].

- **Itératif et incrémental**

C'est la meilleure pratique de gestion des risques d'ordre à la fois technique et fonctionnelle. On peut estimer qu'un projet qui ne produit rien d'exécutible dans les 9 mois court un risque majeur d'échec. Chaque itération garantit que les équipes sont capables d'intégrer l'environnement technique pour développer un produit final

et fournit aux utilisateurs un résultat tangible de leurs spécifications. Le suivi des itérations constitue par ailleurs un excellent contrôle des coûts et des délais.

- **Centré sur l'architecture**

Tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitée. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML et pas seulement documentée en texte.

- **Piloté par les risques**

Les risques majeurs du projet doivent être identifiés au plus tôt, mais surtout levés le plus rapidement possible. Les mesures à prendre dans ce cadre déterminent l'ordre des itérations.

- **Conduit par les cas d'utilisation**

Le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d'utilisation du futur système sont identifiés, décrits avec précision et priorité.

### 2.2.3 Différentes phases du processus unifié

La gestion d'un tel processus est organisée suivant les quatre phases [10] :

#### 1. Phase d'initialisation

Elle conduit à définir la vision du projet, sa portée et sa faisabilité afin de pouvoir décider au mieux de sa poursuite ou de son arrêt.

#### 2. Phase d'élaboration

Elle poursuit trois objectifs principaux en parallèle :

- Identifier et décrire la majeure partie des besoins des utilisateurs.
- Construire l'architecture de base du système.
- Lever les risques majeurs du projet.

#### 3. Phase de construction

Elle consiste surtout à concevoir et implémenter l'ensemble des éléments opérationnels (autres que ceux de l'architecture de base). C'est la phase la plus consommatrice en ressources et en efforts.

#### 4. Phase de transition

Permet de faire passer le système informatique des mains de développeurs à celles des utilisateurs finaux. Les mots-clés sont : conversion des données, formation des utilisateurs, déploiement et béta-tests.

La figure suivante montre les phases du processus unifié :

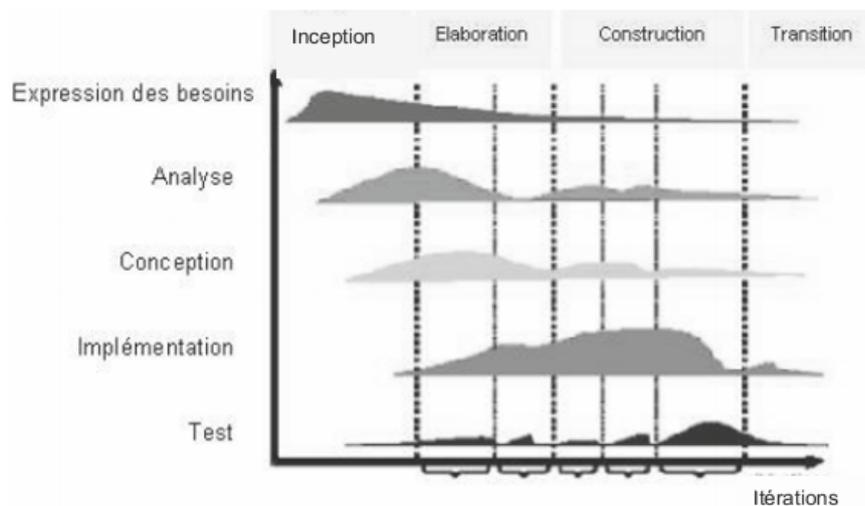


FIG. 2.2 – Les phases du processus unifié[8].

### 2.2.4 Activités du processus unifié

Ses activités de développement sont définies par cinq disciplines fondamentales qui décrivent la capture des exigences, l'analyse et la conception, l'implémentation, le test et le déploiement. Enfin, trois disciplines appelées de support complètent le tableau : gestion de projet, gestion du changement et de la configuration, ainsi que la mise à disposition d'un environnement complet de développement incluant aussi bien des outils informatiques que des documents et des guides méthodologiques [11].

## 2.3 Modélisation fonctionnelle de notre système

### 2.3.1 Description des cas d'utilisations

Un cas d'utilisation est une narration qui décrit un scénario appliqué à une utilisation particulière, dans lequel les acteurs fournissent des entrées pour lesquelles le système produit une sortie observable. Le cas d'utilisation doit juste exprimer ce que doit faire un acteur, il décrit le comportement du système vu de l'extérieur [11].

Dans notre projet, le diagramme de cas d'utilisation est composé d'un seul acteur ; il s'agit de l'administrateur. Ce dernier alimente la base de données du système en introduisant les ressources nécessaires à l'élaboration des emplois du temps, planning des examens et planning des soutenances. Les ressources considérées sont les entités physiques suivantes : Enseignants, Locaux, Modules, Sections et groupes.

### 2.3.2 Description textuelle des cas d'utilisation

La fiche de description textuelle d'un cas d'utilisation n'est pas définie par UML. Le formalisme qu'on utilisera est juste une proposition.

#### 1- Cas d'utilisation : authentification

L'authentification est le point d'entrée de l'application puisque, elle présente la première fenêtre qu'il affiche pour authentifier l'administrateur.

Cas d'utilisation	Authentification
Résumé	Vérifier les possibilités d'accès au système
Scénario nominal	[Debut] <ol style="list-style-type: none"> <li>1. Accès à l'application.</li> <li>2. Saisie du login et du mot de passe.</li> <li>3. Le système vérifie les données saisies.</li> <li>4. Le système affiche l'interface correspondante.</li> </ol> [Fin]
Alternative	A1 (alternative) est une étiquette qui indique que le système réaffichera le formulaire d'authentification dans le cas où les données introduites sont erronées.

TAB. 2.1 – Description du cas d'utilisation : Authentification

#### 2- Cas d'utilisation : gestion des enseignants

Le cas d'utilisation " gestion des enseignants " est caractérisé par les trois scénarios suivants :

- Ajout.
- Modification.
- Suppression.

Cas d'utilisation	Gestion des enseignants
Résumé	L'administrateur gère les enseignants impliqués dans la planification, ceci en ajoutant, en modifiant ou en supprimant un enseignant.
Scénario nominal	<p>[Debut]</p> <ol style="list-style-type: none"> <li>1. Accès à l'application.</li> <li>2. Authentification.</li> <li>3. Le système affiche l'interface correspondante.</li> <li>4. L'administrateur demande le formulaire de gestion des enseignants.</li> <li>5. Le système affiche la liste des enseignants.</li> <li>6. Trois alternatives se présentent : dans le cas où l'administrateur veut ajouter un enseignant, le système affiche le formulaire d'ajout. Si l'administrateur choisit de modifier (ou de supprimer) un enseignant, il doit alors sélectionner l'enseignant en question, puis effectue l'opération.</li> <li>7. Le système affiche un message de confirmation.</li> </ol> <p>[Fin]</p>

TAB. 2.2 – Description du cas d'utilisation : Gestion des enseignants.

### 2.3.3 Visual paradigm

Visual Paradigm dispose de tous les diagrammes UML, des outils essentiellement dans le système et la conception de base de données. Outils de modélisation innovants comme le catalogue de ressources, transitaire et Nicknamer rend la modélisation du système facile et rentable. Doc. Composer vous permet de produire spécification détaillée prêt à l'emploi en discussion avec seulement quelques clics.

## 2.4 Diagramme de cas d'utilisation de l'application

Le diagramme de cas d'utilisation est un formalisme permettant de modéliser le fonctionnement d'un système par un découpage en fonctionnalités. Il illustre de plus la nature des interactions avec ces fonctionnalités offertes à titre de services à des acteurs externes au système. Chaque fonctionnalité est appelée un cas d'utilisation[12].

Le diagramme ci-après représente le diagramme général de cas d'utilisation de notre application.

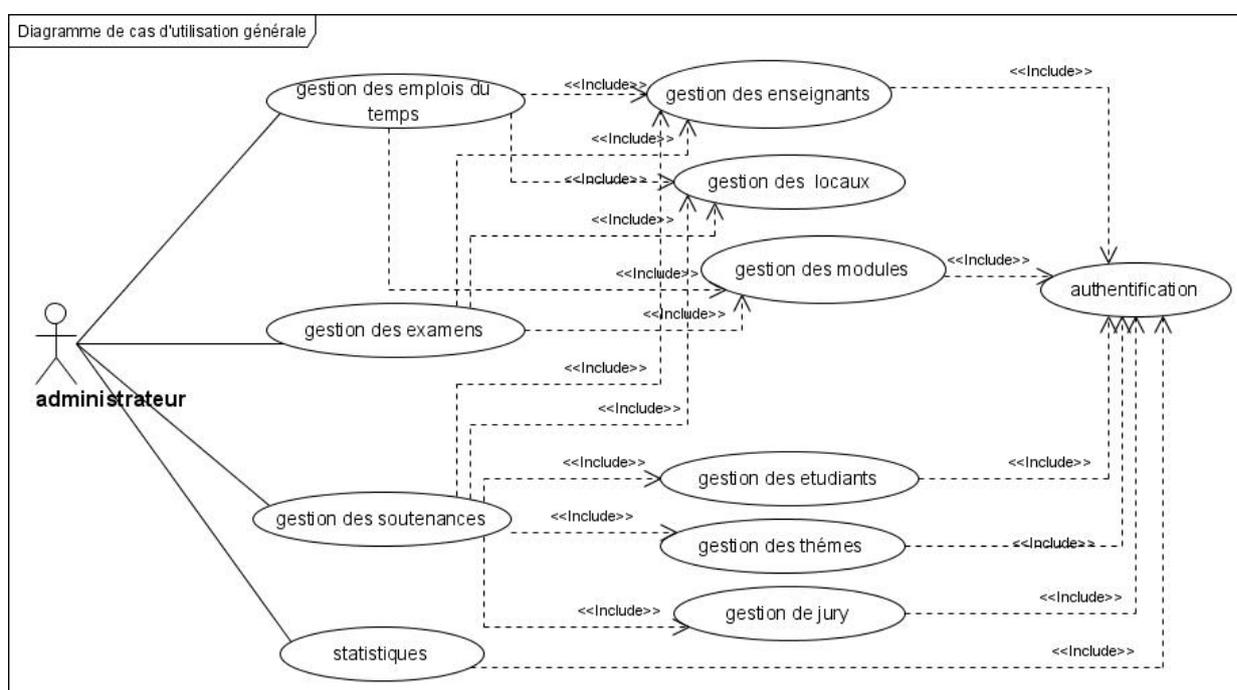


FIG. 2.3 – Diagramme générale de cas d'utilisation.

Dans le diagramme de cas d'utilisation ci-dessous, nous décrivons les cas d'utilisation de la gestion des emplois du temps.

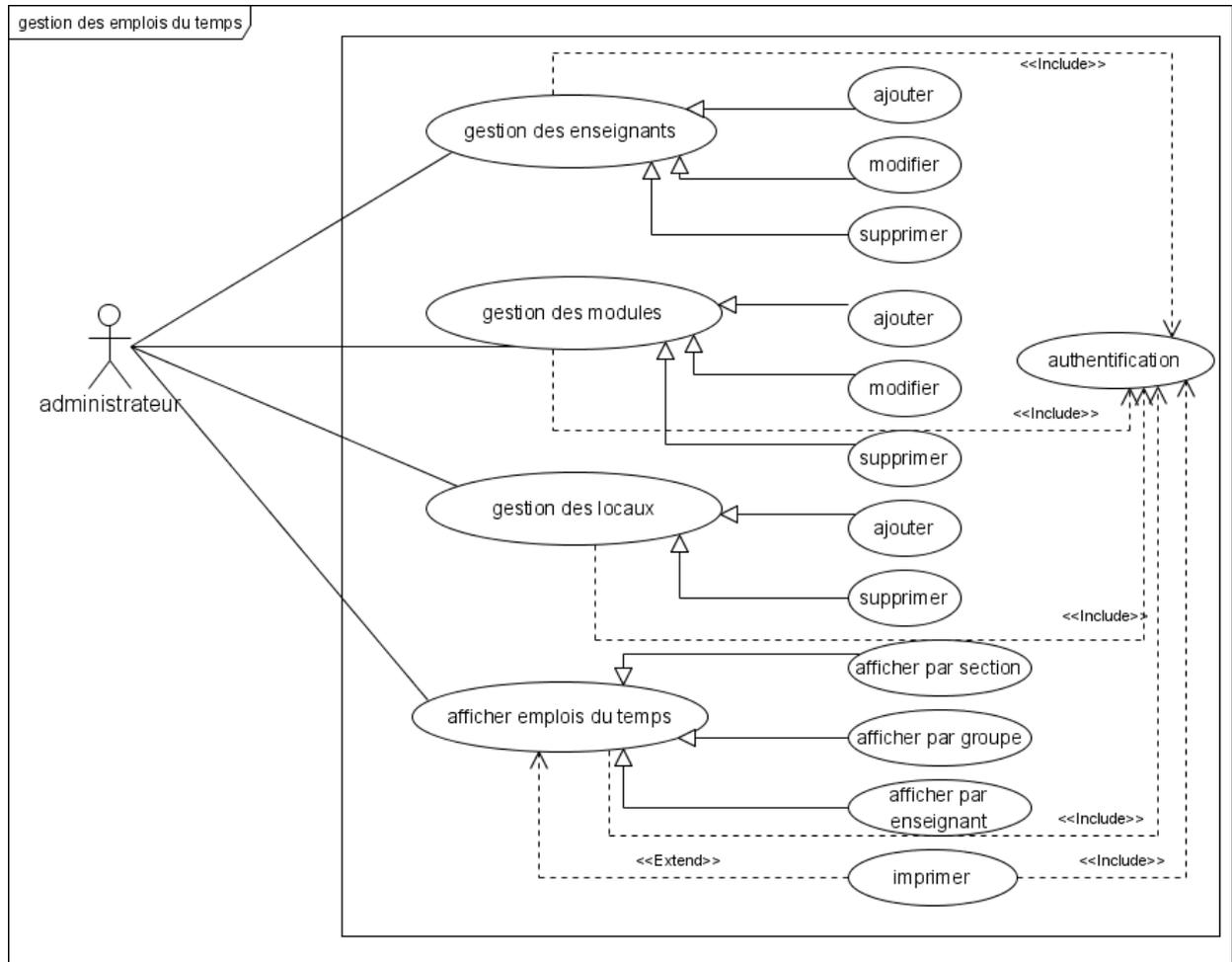


FIG. 2.4 – Diagramme de cas d'utilisation pour la gestion des emplois du temps.

Dans le diagramme de cas d'utilisation ci-dessous, nous décrivons les cas d'utilisation de la gestion des examens.

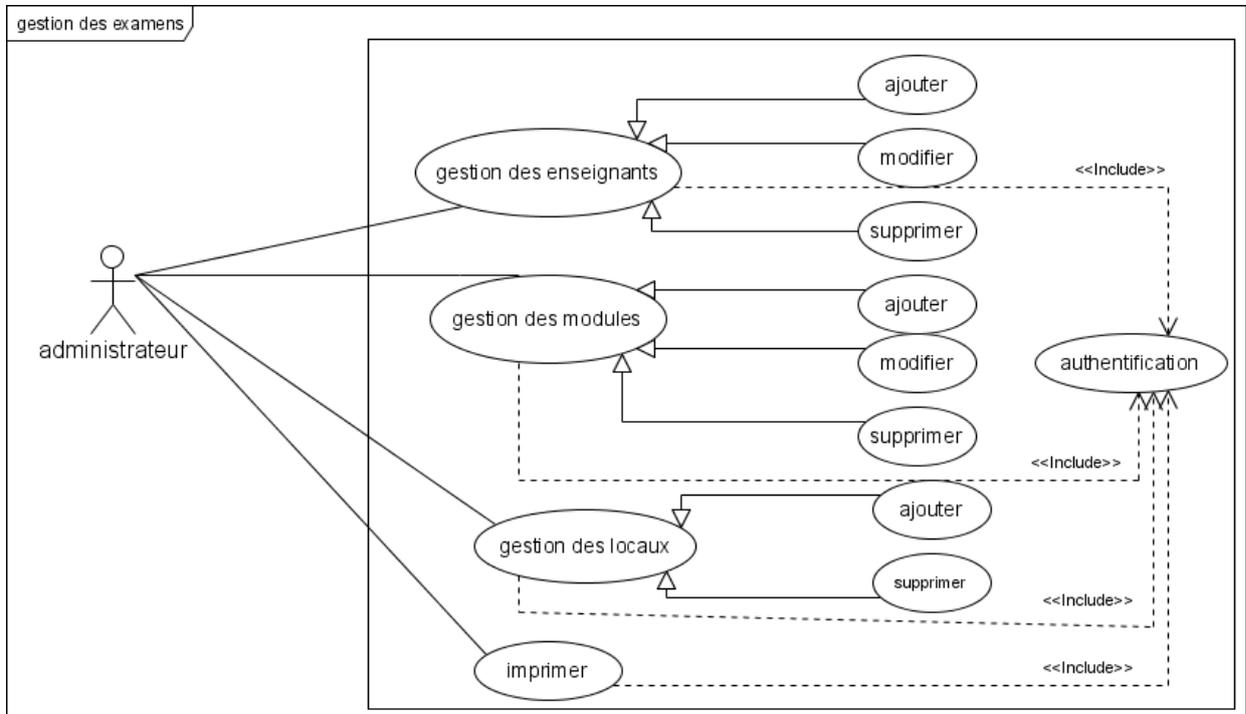


FIG. 2.5 – Diagramme de cas d'utilisation pour la gestion des examens.

Dans le diagramme de cas d'utilisation ci-dessous, nous décrivons les cas d'utilisation de la gestion des soutenances.

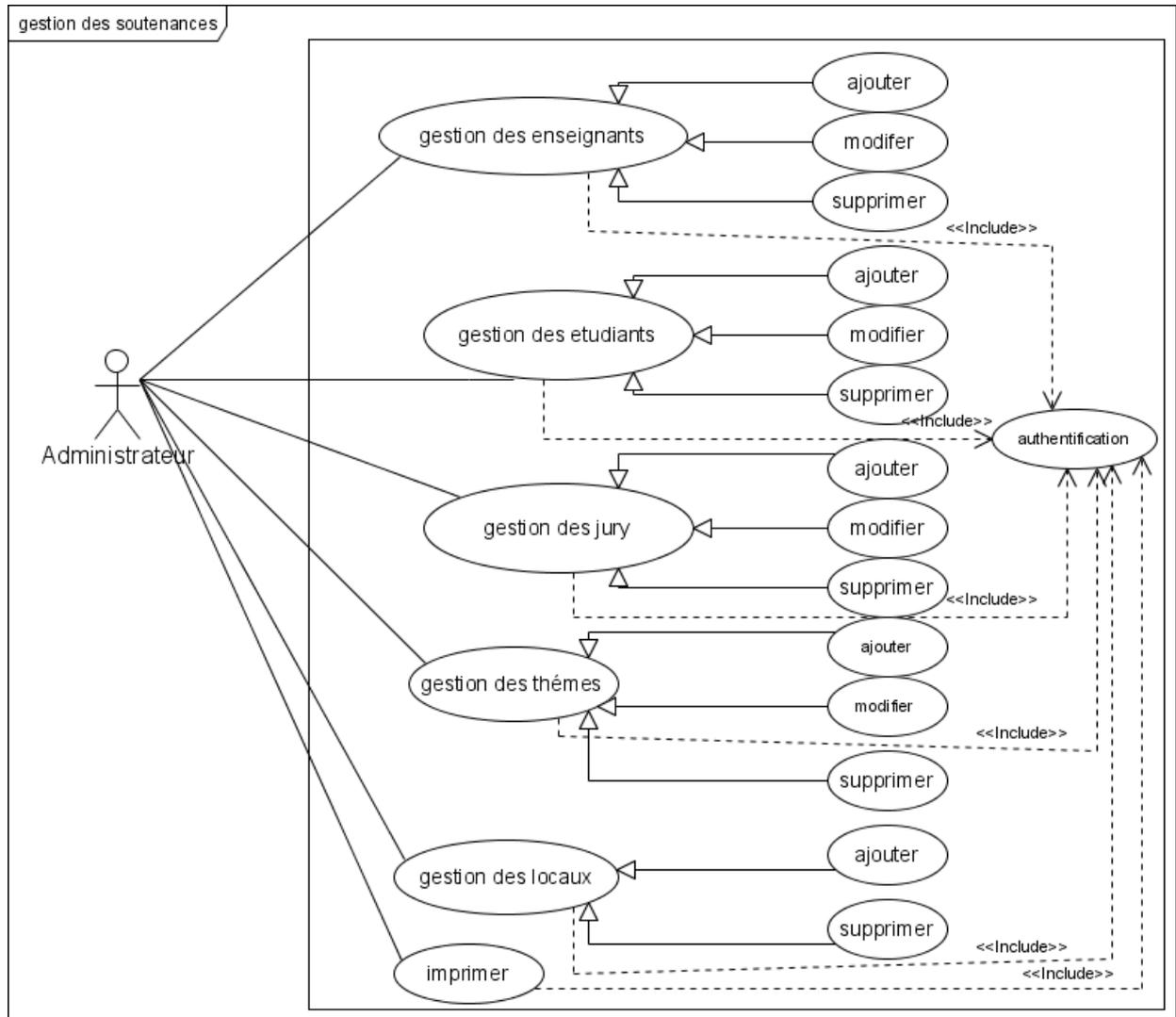


FIG. 2.6 – Diagramme de cas d'utilisation pour la gestion des soutenances.

## 2.4.1 Réalisation des diagrammes de séquences

L'objectif du diagramme de séquence est de représenter les interactions entre objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés[13][14].

### 1. Diagramme de séquence du cas d'utilisation : Authentification

L'authentification consiste à assurer la confidentialité des données, elle se base sur la

vérification des informations associées à un utilisateur (généralement un login et un mot de passe). Lors d'une authentification deux cas se présentent : les informations introduites par l'utilisateur sont incomplètes, dans ce cas un message d'erreur s'affiche, ou les informations saisies sont complètes et le système procède à leur vérification. Ceci explique l'utilisation de l'opérateur "alt".

Le même opérateur illustre les deux réactions du système, après la vérification des informations saisies par l'utilisateur, soit par l'affichage d'un message d'erreur, ou de l'interface correspondante.

L'opérateur "alt" : correspond à une instruction de test avec une ou plusieurs alternatives possibles. Il est aussi permis d'utiliser les clauses de type sinon.

L'opérateur "loop" : correspond à une instruction de boucle permettant d'exécuter une séquence d'interactions tant qu'une condition est satisfaite.

L'opérateur "ref" : permet d'appeler une séquence d'interactions décrite par ailleurs constituant ainsi une sorte de sous diagramme de séquence.

L'opérateur "opt" : (optional) correspond à une instruction de test sans alternative (sinon).

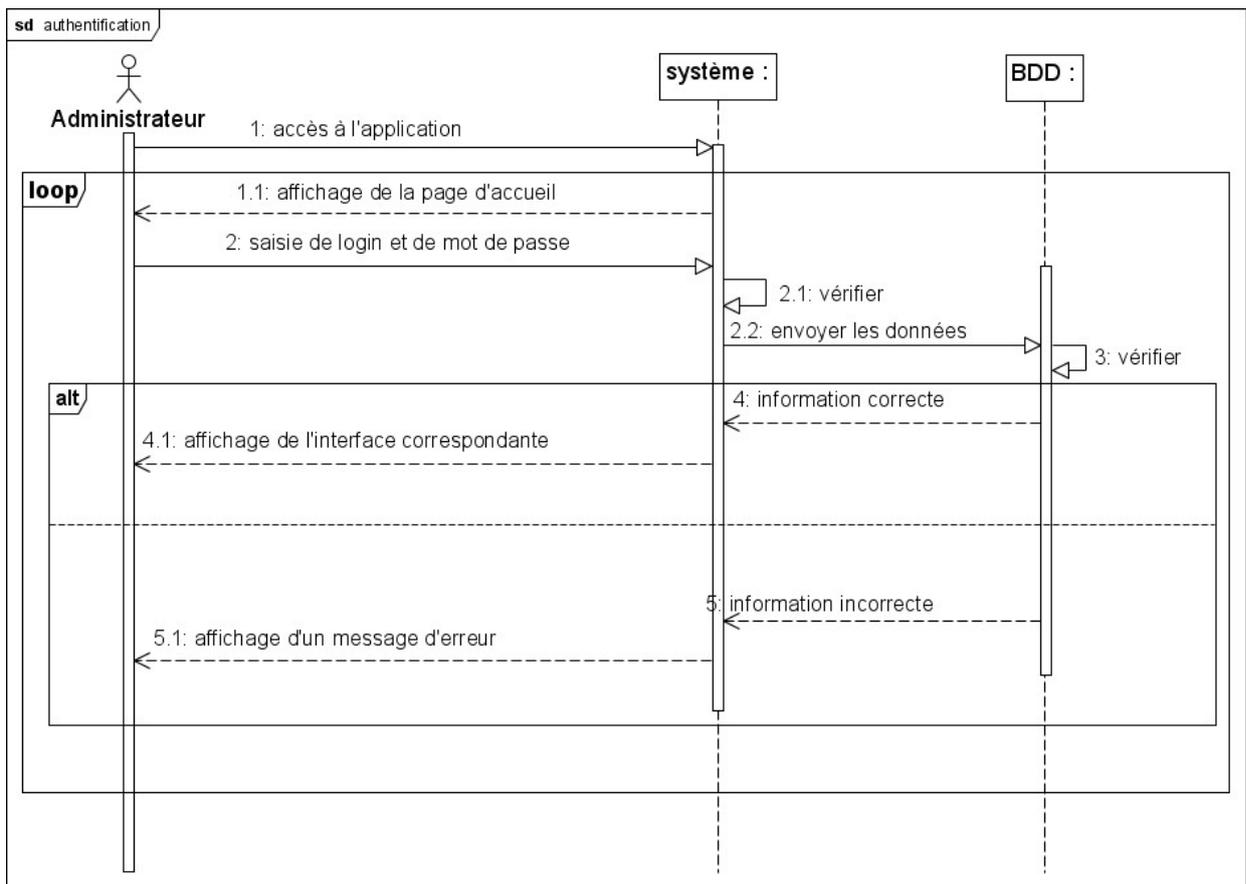


FIG. 2.7 – Diagramme de séquence du cas d'utilisation : Authentification.

## **2. Diagramme de séquence du cas d'utilisation : Gestion des enseignants**

Ce cas d'utilisation comporte trois scénarios " ajouter ", " modifier " ou " supprimer " un enseignant. Dans le cas d'ajout : Le système répond à la demande de l'administrateur concernant l'ajout d'un enseignant par l'affichage d'un formulaire qui sera validé après remplissage.

Dans le cas de modification : La réponse du système, pour la requête de modification, est la liste des enseignants. Une fois l'utilisateur à modifier est choisi, un formulaire s'affiche afin d'apporter les modifications souhaitées.

Dans le cas de suppression : La réponse du système, pour la requête de suppression, est la liste des enseignants. Une fois l'enseignant à supprimer est sélectionné, il sera supprimé définitivement de la base de données.

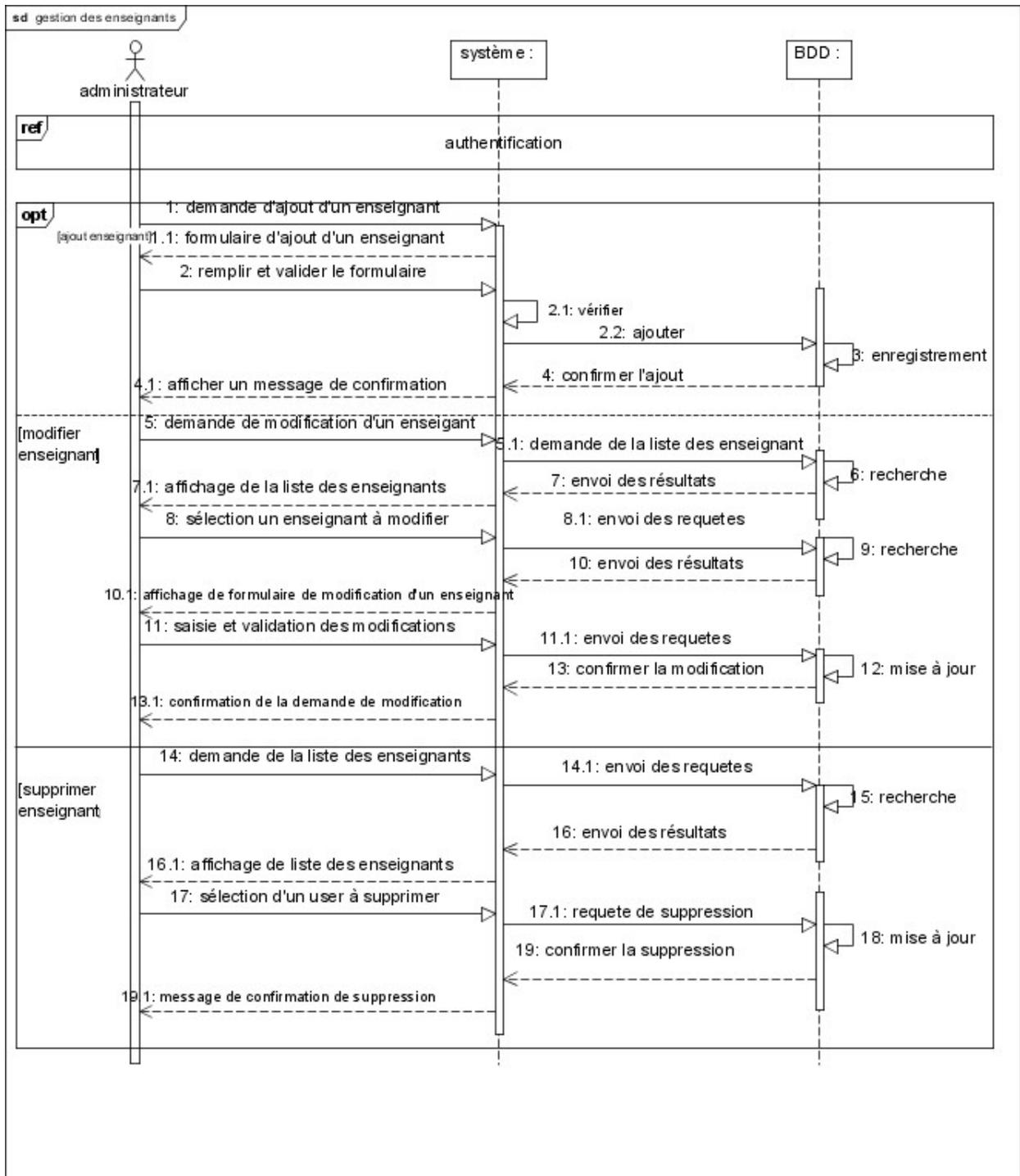


FIG. 2.8 – Diagramme de séquence du cas d'utilisation : gestion des enseignants.

### 3. Diagramme de séquence du cas d'utilisation : Gestion des locaux

Ce cas d'utilisation comporte deux scénarios " ajouter " ou " supprimer " un local. Dans le cas d'ajout : Le système répond à la demande de l'administrateur concernant l'ajout d'un local par l'affichage d'un formulaire qui sera validé après remplissage. Dans le cas de suppression : La réponse du système, pour la requête de suppression, est la liste des locaux. Une fois un local à supprimer est sélectionné, il sera supprimé définitivement de la base de données.

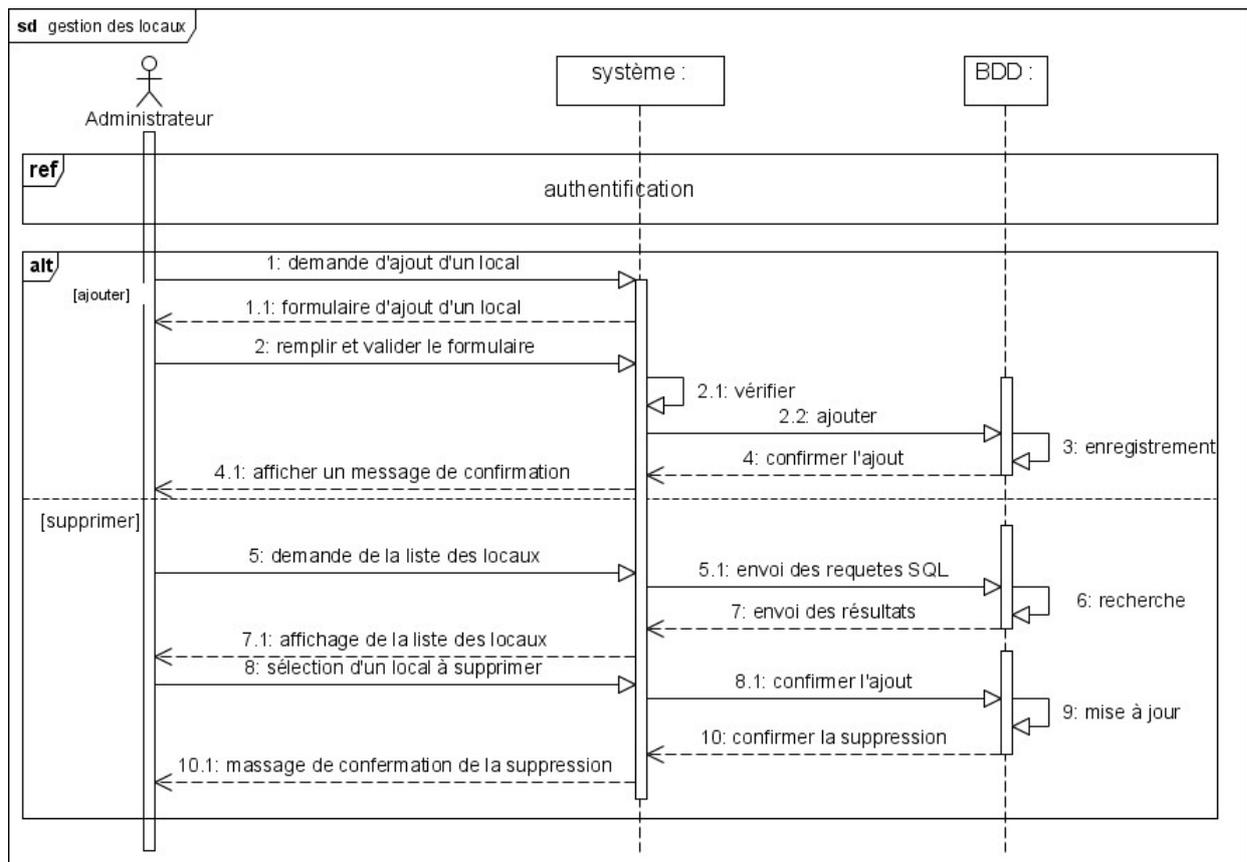


FIG. 2.9 – Diagramme de séquence du cas d'utilisation : gestion des locaux.

#### 4. Diagramme de séquence du cas d'utilisation : afficher emplois du temps par section

Ce diagramme présente l'affichage de l'emploi du temps pour un niveau et une section donnée.

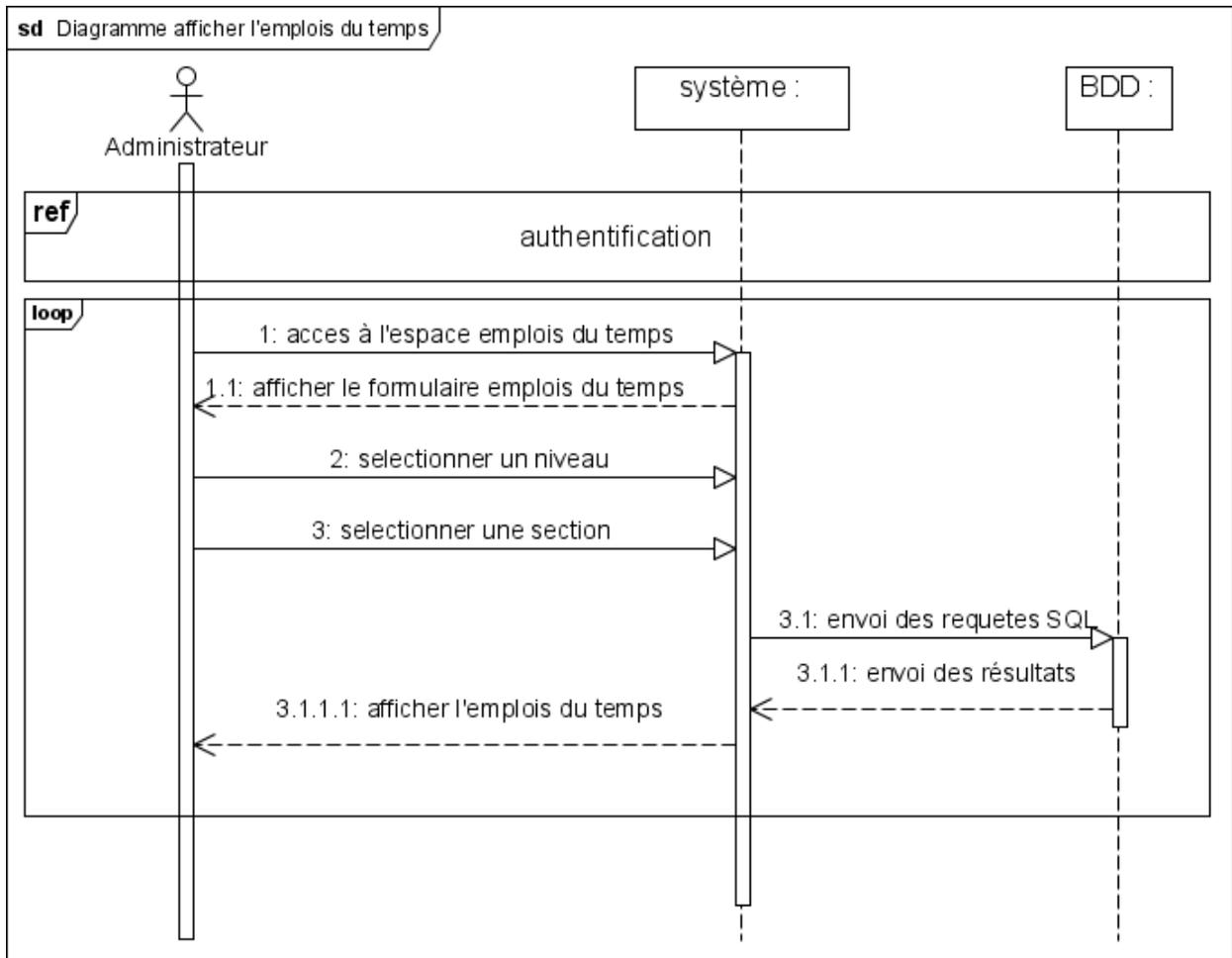


FIG. 2.10 – Diagramme de séquence du cas d'utilisation : afficher emplois du temps par section.

### 2.4.2 Réalisation des diagrammes d'activité

Le diagramme d'activité est une variante du diagramme d'état transition .Il permet de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

Les diagramme d'activité de nos cas d'utilisation sont présentés sur les figures suivantes :

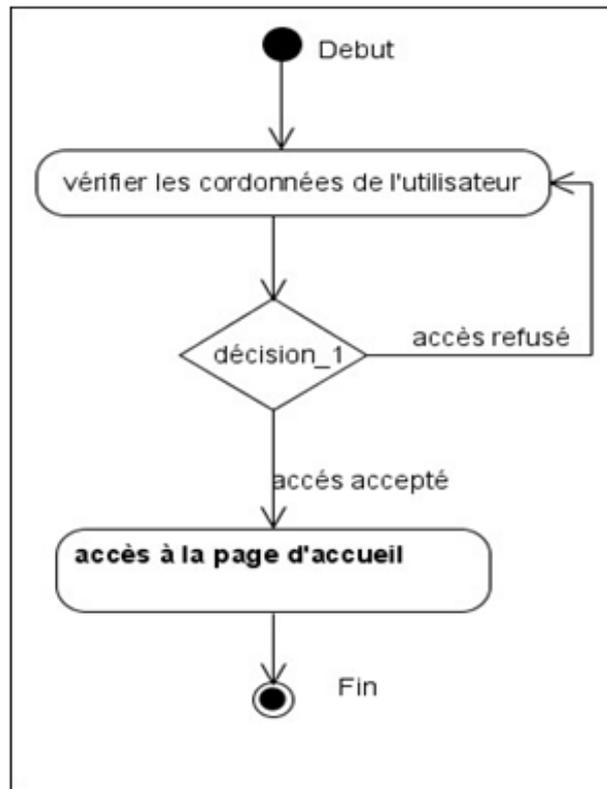


FIG. 2.11 – Diagramme d'activité : Authentification.

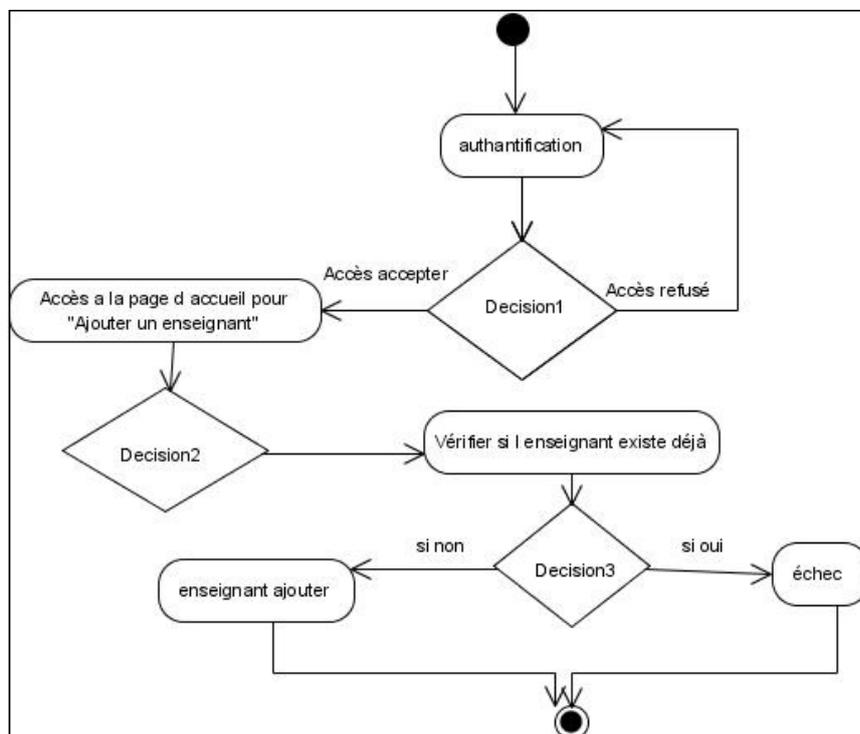


FIG. 2.12 – Diagramme d'activité : Ajouter un enseignant.

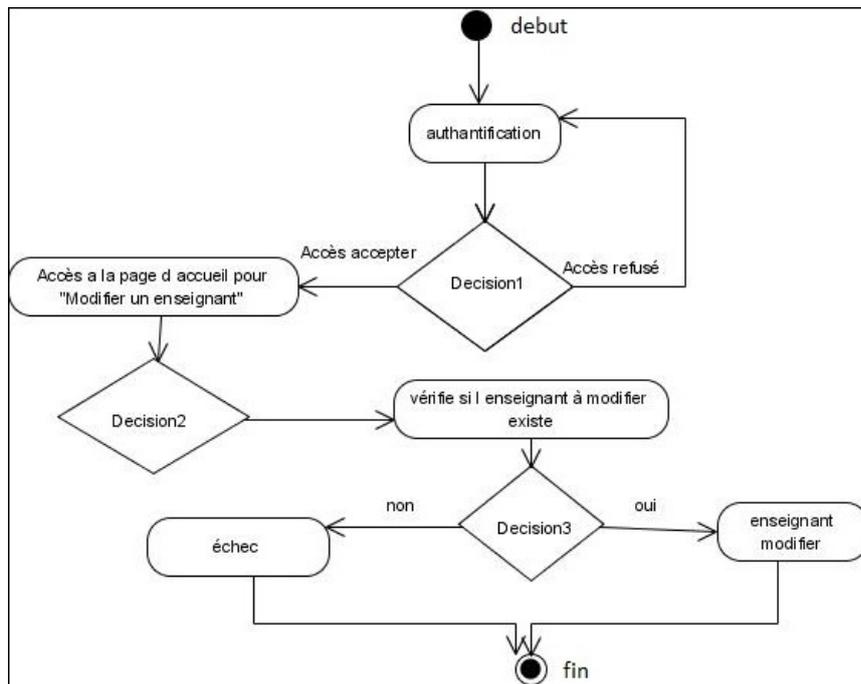


FIG. 2.13 – Diagramme d'activité : Modifier un enseignant.

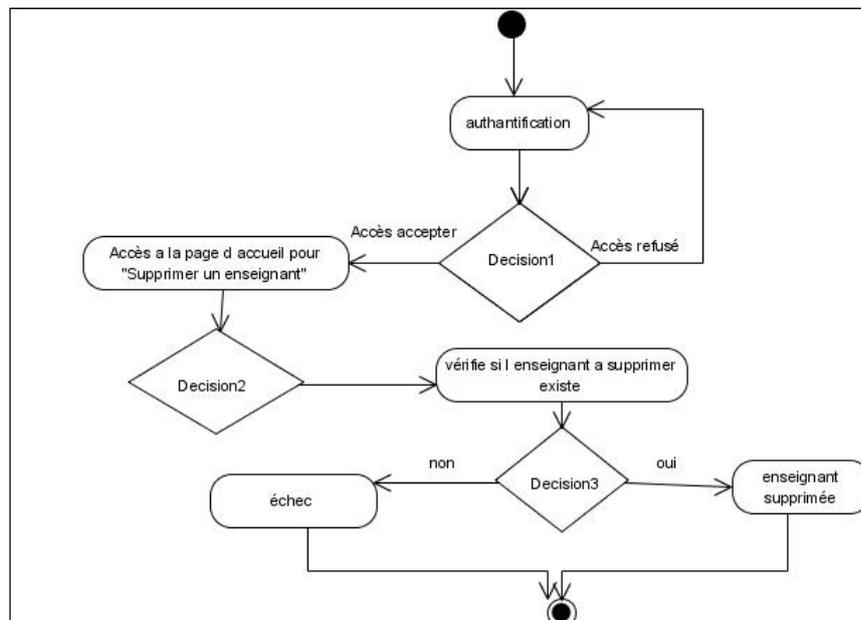


FIG. 2.14 – Diagramme d'activité : Supprimer un enseignant.

## **2.5 Modélisation structurelle de notre application**

### **2.5.1 Dictionnaire de données épuré**

Le tableau suivant décrit les attributs de chacune des classes et classe d'association du diagramme de notre application.

Classe	Attributs	Désignation attributs	Types attributs
Enseignant	identifiant	Code de l'enseignant	VARCHAR
	nom_ens	Nom de l'enseignant	VARCHAR
	prenom_ens	Prénom de l'enseignant	VARCHAR
	grade	Grade de l'enseignant	VARCHAR
	specialite	Spécialité de l'enseignant	VARCHAR
	num_tel	Numéro de téléphone de l'enseignant	VARCHAR
	email	Email de l'enseignant	VARCHAR
	statut	Statut de l'enseignant	VARCHAR
module	affiliation	Affiliation de l'enseignant	VARCHAR
	id_module	Code du module	VARCHAR
	libelle	Nom du module	VARCHAR
	nbr_cours	Nombre de cours d'un module	INT
	nbr_td	Nombre de td d'un module	INT
assurer	nbr_tp	Nombre de tp d'un module	INT
	type_enseignement	Type enseignement	VARCHAR
	heur	Heur du créneau	VARCHAR
local	jour	Jour du créneau	VARCHAR
	id_local	Identificateur du local	VARCHAR
	num_local	Nom du local	VARCHAR
	type_local	Type du local	VARCHAR
niveau	capacite	Nombre de place de local	INT
	num_niveau	Niveau d'étude	VARCHAR
section	type	Spécialité	VARCHAR
	id_section	Identificateur de la section	VARCHAR
groupe	nom_section	Nom de la section	VARCHAR
	id_groupe	Identificateur du groupe	VARCHAR
sujet	nom_groupe	Le numéro du groupe	VARCHAR
	code_sujet	Code du sujet	VARCHAR
	theme	Thème	TEXT
etudiant	niveau_etud	Niveau d'étude	VARCHAR
	Matricule	Identificateur de l'étudiant	VARCHAR
	Nom	Nom de l'étudiant	VARCHAR
jury	Prenom	Prénom de l'étudiant	VARCHAR
	qualité	qaualité d'un Membre de jury	VARCHAR
	Date	Date de soutenance	DATE
surveiller	heure	Heure de soutenance	VARCHAR
	date_exam	Date de l'examen	DATE
	heure_exam	L'heure de l'examen	VARCHAR

TAB. 2.3 – Dictionnaire de données épuré.

## 2.5.2 Diagramme de classes

Le diagramme de classe constitue l'un des pivots essentiels de la modélisation avec UML. En effet, ce diagramme permet de donner la représentation statique du système à développer. Cette représentation est basée sur les concepts de classe et d'association. Chaque classe se décrit par les données et les traitements dont elle est responsable. Les traitements sont matérialisés par des opérations. Le détail des traitements n'est pas représenté directement dans le diagramme de classe, mais dans l'algorithme général et le pseudo-code correspondant [14].

## 2.5.3 Les concepts d'un diagramme de classe

- **Un objet**

Un Objet est un concept, une abstraction ou une chose qui a un sens dans le contexte du système à modéliser. Chaque objet a une identité et peut être distingué des autres, sans considérer a priori les valeurs de ses propriétés [14].

- **Une classe**

Une classe représente la description abstraite d'un ensemble d'objets possédant les mêmes caractéristiques. On peut parler également de type [13].

Un objet est une instance d'une classe. La classe représente l'abstraction de ses objets.

Au niveau de l'implémentation, c'est-à-dire au cours de l'exécution d'un programme, l'identificateur d'un objet correspond à une adresse mémoire [14].

Ce qui suit est la représentation graphique d'une classe :

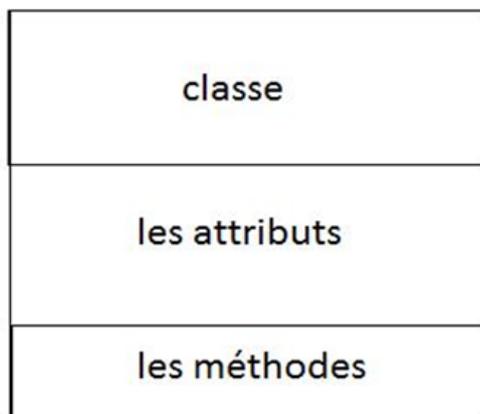


FIG. 2.15 – Représentation graphique d'une classe.

- **Attribut**

Un attribut est une donnée élémentaire servant à caractériser les classes et les relations [12].

- **Opération**

Une opération représente un élément de comportement (un service) contenu dans une classe. Nous ajouterons plutôt les opérations en conception objet, car cela fait partie des choix d'attribution des responsabilités aux objets [11].

- **Méthode**

Une méthode est une opération programmée sur les objets d'une classe [8].

- **Classe d'association**

Il s'agit d'une association promue au rang de classe. Elle possède de tout à la fois les caractéristiques d'une association et celle d'une classe et peut donc porter des attributs qui se valorisent pour chaque lien [13].

## 2.5.4 Relations entre les classes

En UML, il existe plusieurs types de relations entre les classes :



## 2.6 Modèle Relationnel de données

Le concepteur d'une base de données relationnelle doit élaborer ce qu'il est convenu d'appeler le schéma relationnel de la base de données. Cette activité consiste à définir toutes les relations normalisées de la base de données et les domaines de leurs attributs. Théoriquement cela consisterait à décrire par intention chaque relation et définir les domaines de chaque attribut de la relation [14].

### 2.6.1 Règles de dérivation du 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 [15] :

#### 1. 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 l'identifiant. Dans le cas où aucun attribut ne convient en tant qu'identifiant, il faut en ajouter un de telle sorte que la relation dispose d'une clé primaire.

#### Classe -association et n-aire

#### 2. Règle 2 : Association un-à-plusieurs

Il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.

#### 3. Règle 3 : Associations plusieurs-à-plusieurs

L'association devient une relation dont la clé primaire est formée par la concaténation des identifiants des classes connectées à l'association. Les attributs de l'association doivent être ajoutés à la nouvelle relation.

L'application des règles de passage énumérées précédemment, nous permet d'avoir le modèle relationnel de la base de données de l'application à mettre en œuvre.

- Enseignant (identifiant, nom\_ens, prenom\_ens, grade, specialite, num\_tel, email, statut, affiliation).
- Module (id\_module, libelle, nbr\_cours, nbr\_td, nbr\_tp, num\_niveau#) où num\_niveau est une clé étrangère qui fait référence à la relation Niveau.
- Assurer (identifiant#, id\_module#, id\_local#, heur, jour, type\_enseignement).
- Local (id\_local, num\_local, type\_local, capacite).

- Niveau (num\_niveau, type).
- Section (id\_section, num\_niveau#, nom\_section).
- Groupe (id\_groupe, id\_section#, num\_niveau#, nom\_groupe).
- Etudiant (matricule, nom, prenom, code\_sujet#) où code\_sujet est une clé étrangère qui fait référence à la relation Sujet.
- Sujet (code\_sujet, theme, niveau\_etud).
- Jury (identifiant#, code\_sujet#, qualite, date, heure).
- surveiller (identifiant#, id\_groupe#, date\_exam, heure\_exam, id\_local#, id\_module#).

## 2.7 Conclusion

Dans ce chapitre, nous avons commencé par une présentation de l'UML, ses différents diagrammes ainsi que le processus unifié suivi durant le développement. Ensuite, nous avons présenté les cas d'utilisation, les diagrammes de séquence qui leur correspondent ainsi que les diagrammes d'activité. Enfin, nous avons terminé par le diagramme de classes et le modèle relationnel de données qui nous permet d'avoir le schéma de la base de données de l'application à réaliser dans le chapitre suivant.

# 3

## Implémentation et réalisation

### Introduction

L'étape de l'implémentation et de la réalisation est la dernière de notre projet. On la considère comme étant l'étape la plus cruciale puisqu'elle traite l'onglet pratique du projet.

Commençons d'abord, par une brève illustration de l'environnement de travail ainsi que l'ensemble des logiciels que nous avons utilisé dans la réalisation de l'application et l'implémentation de la base de données, puis passons à un aperçu des interfaces les plus importantes de notre application.

### 3.1 Outils et langages de développements

#### 3.1.1 Wamp Server

Wamp Server est une plateforme de développement web de type WAMP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. Wamp Server n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (apache et MySQL), un interpréteur de script(PHP), ainsi que PhpMyAdmin pour l'administration web des bases MySQL. Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs à travers un tray-icon (icône près de l'horloge de Windows)[16].

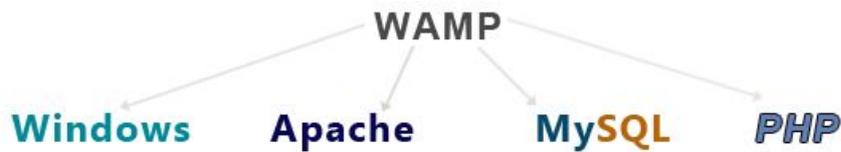


FIG. 3.1 – les composants de wamp

### 3.1.1.1 Présentation du système de gestion de bases de données MySQL

MySQL est un système de gestion de bases de données relationnelles performant et puissant, doté d'une grande facilité d'utilisation et d'une architecture client/ serveur qui comprend un serveur de bases de données multitâches et multi-utilisateurs, ainsi que divers programmes clients. Il ne s'agit pas donc d'un simple langage de base de données, bien que MySQL utilise SQL, un standard parmi les langages de base de données. Ce langage permet de définir, de manipuler et de sécuriser les données.



FIG. 3.2 – MySQL

### 3.1.1.2 Principe de fonctionnement

Le serveur MySQL [17] est le gestionnaire du système de base de données. C'est lui qui manipule toutes les instructions adressées à la base de données. Par exemple, si on veut créer une nouvelle base de données, il faut envoyer un message au serveur MySQL disant : " crée une nouvelle base de données que tu appelleras nouvelle base ". Le serveur MySQL crée alors un sous répertoire dans son dossier de données, lui donne le nom " nouvelle base" et crée les fichiers nécessaires au format requis dans ce nouveau sous répertoire. De la même façon, pour ajouter des données à cette base de données, vous envoyez un message au serveur MYSQL en lui fournissant les données et en lui disant dans quel endroit vous voulez qu'elles soient rangées.

Le dialogue avec la base de données s'effectue en passant des messages au serveur MySQL.

Ces messages peuvent être envoyés de différentes façons, par exemple, via PHP. Il existe des instructions dans ce langage pour adresser des messages au serveur MySQL.

### 3.1.1.3 Avantages de MySQL

- La rapidité constitue son principal atout. MySQL permet de traiter et de maintenir de grosses bases de données avec une grande fiabilité.
- MySQL est conçu comme un système multithread et peut donc utiliser une machine dotée de plusieurs processeurs.
- Il possède aussi une grande portabilité lui permettant d’être installé sur divers systèmes d’exploitation.
- MySQL a été développé dans le cadre des logiciels open source.
- De nombreux langages de programmation disposant d’une API (Application Programming Interface) permettant de travailler directement avec MySQL, c’est notamment le cas pour le PHP [18].

### 3.1.1.4 Exemple de tables de bases de données

Table	Action	Lignes	Type	Interclassement	Taille	Perte
assurer	Afficher Structure Rechercher Insérer Vider Supprimer	0	InnoDB	latin1_swedish_ci	64,0 Kio	-
enseignant	Afficher Structure Rechercher Insérer Vider Supprimer	77	InnoDB	latin1_swedish_ci	16,0 Kio	-
etudiant	Afficher Structure Rechercher Insérer Vider Supprimer	0	InnoDB	latin1_swedish_ci	32,0 Kio	-
groupe	Afficher Structure Rechercher Insérer Vider Supprimer	40	InnoDB	latin1_swedish_ci	48,0 Kio	-
jury	Afficher Structure Rechercher Insérer Vider Supprimer	0	InnoDB	latin1_swedish_ci	128,0 Kio	-
local	Afficher Structure Rechercher Insérer Vider Supprimer	17	InnoDB	latin1_swedish_ci	16,0 Kio	-
module	Afficher Structure Rechercher Insérer Vider Supprimer	10	InnoDB	latin1_swedish_ci	32,0 Kio	-
niveau	Afficher Structure Rechercher Insérer Vider Supprimer	8	InnoDB	latin1_swedish_ci	16,0 Kio	-
section	Afficher Structure Rechercher Insérer Vider Supprimer	14	InnoDB	latin1_swedish_ci	32,0 Kio	-
sujet	Afficher Structure Rechercher Insérer Vider Supprimer	0	InnoDB	latin1_swedish_ci	16,0 Kio	-
surveiller	Afficher Structure Rechercher Insérer Vider Supprimer	3	InnoDB	latin1_swedish_ci	64,0 Kio	-
<b>11 tables</b>	<b>Somme</b>	<b>169</b>	<b>InnoDB</b>	<b>latin1_swedish_ci</b>	<b>464,0 Kio</b>	<b>0</b>

FIG. 3.3 – Exemple de tables de bases de données

### 3.1.1.5 Connecteur JDBC

Pour accéder à la base de données, JPA utilise un connecteur. Pour les bases de données relationnelles, le connecteur utilisé est JDBC. Il faut configurer ce connecteur de la manière suivante :

- Lui donner un nom unique permettant à l’application de le référencer comme une source de données.

- Définir un pool de connexion, c'est-à-dire un ensemble de liens prédéfinis entre le serveur d'applications et le serveur de bases de données qui pourront être utilisés par l'application. La configuration du pool de connexion consiste à donner l'adresse du serveur de bases de données, le nom de la base à laquelle accéder ainsi qu'un login/mot de passe d'accès [19].

### 3.1.2 Java

Nous avons utilisé le langage de programmation java [20] qui est un langage a usage général, évolué et orienté objet et qui reprend en grande partie la syntaxe du langage C++.

Java possède les avantages suivants :

- Robustesse : le langage fournit des structures facilitant l'élimination des bugs.
- Portabilité : un processus java s'exécute dans un environnement virtuel le rendant indépendant de spécificités effectives.
- Dynamicité : un programme java peut facilement s'enrichir sans avoir besoin d'être arrêté.

### 3.1.3 NetBeans



FIG. 3.4 – *NetBeans*

Nous allons à programmer en java on utilisant NetBeans, qui est un environnement de développement intégré c.-à-d. une application qui propose dans un même système de fenêtrage des outils qui facilite la tache de développeur par exemple il propose :

- Un éditeur de texte avec coloration syntaxique : il colorie automatiquement les mots clé utilisé et les éléments syntaxiques important de langage.
- Auto complétion : il propose sous certain condition les choix d'écriture disponible de l'instruction en cours d'écriture.
- Fenêtre de compilation.
- Fenêtre d'exécution : qui permet de visualiser les résultats de l'application.

### 3.1.3.1 Fonctions de NetBeans

- Configuration et gestion de l'interface graphique des utilisateurs.
- Support de différents langages de programmation.
- Traitement de code source (édition, navigation, formatage, inspection, etc.).
- Fonctions d'import/export depuis et vers d'autres IDE, tels qu'Eclipse ou JBuilder.
- Accès et gestion de bases de données, serveur Web, ressources partagées.
- Gestion de tâches.
- Documentation intégrée.

### 3.1.3.2 Avantages de NetBeans

- Distribuer en code open source.
- Entièrement gratuit.
- Sa facilité d'installation et d'usage.

## 3.1.4 Jdk



FIG. 3.5 – *Jdk*

Java Développeur Kit [21] est l'environnement dans lequel le code Java est compilé pour être transformé en byte code afin que la machine virtuelle JAVA (JVM) puisse l'interpréter.

Les composants primaires du JDK sont une sélection d'outils de programmation, incluant :

- Javac : le compilateur, qui convertit le code source en fichier .class (contenant le byte code Java).
- Jar : l'archiveur, qui met sous forme d'un paquetage unique l'ensemble des fichiers class en un fichier JAR.
- Javadoc : le générateur de documentation, qui génère automatiquement de la documentation à partir des commentaires du code source.
- Jdb : le débogueur.

### **3.1.5 EDI**

On appelle EDI (ou IDE), acronyme de " Environnement de développement intégré " l'interface qu'offre Delphi pour aider l'utilisateur à construire son application. Cette interface ressemble plus à un atelier où l'on dispose d'une boîte à outils et d'un ensemble d'objets qui servent à fabriquer une application. On n'écrit pas une application mais on la fabrique.

- C'est un langage très évolue de niveau haut.
- Intégration d'un BDE pour les applications qui utilise les bases de données.
- Contient une grande bibliothèque des composantes visuelles et non visuelle.

## **3.2 Présentation de quelques interfaces de notre application**

Dans cette section, nous présentons quelques interfaces de notre application que nous avons réalisée.

### **3.2.1 Page d'authentification**

C'est l'interface principale de l'application, elle permet à l'administrateur d'accéder à l'application en saisissant un login et un mot de passe pour s'authentifier. Depuis cette page, il pourra accéder aux autres interfaces .



FIG. 3.6 – Fenêtre d'authentification

Si une erreur d'identification surgit sur le login et / ou le mot de passe, le système retourne un message d'erreur.

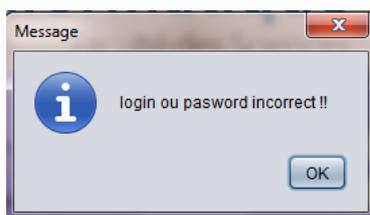
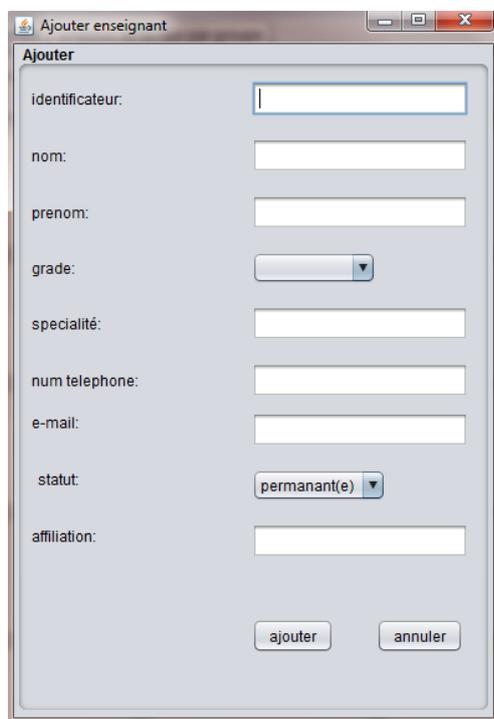


FIG. 3.7 – Message d'erreur

### 3.2.2 Ajouter enseignant

L'interface suivante représente le formulaire d'ajout d'un enseignant. L'administrateur remplit tous les champs puis enregistre pour que l'enseignant soit ajouté directement à la base de données.



The image shows a software window titled "Ajouter enseignant". Inside the window, there is a form with the following fields and controls:

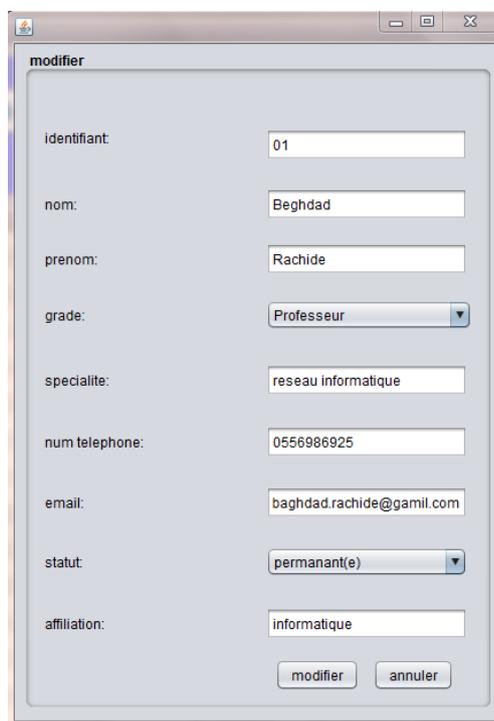
- identificateur: text input field
- nom: text input field
- prenom: text input field
- grade: dropdown menu
- specialité: text input field
- num telephone: text input field
- e-mail: text input field
- statut: dropdown menu with "permanant(e)" selected
- affiliation: text input field

At the bottom of the form, there are two buttons: "ajouter" and "annuler".

FIG. 3.8 – *Ajouter enseignant*

### 3.2.3 Modifier enseignant

L'interface suivante représente le formulaire de modification d'un enseignant. L'administrateur modifie les champs qu'il désire puis enregistre pour que l'enseignant soit modifié directement à la base de données.



The image shows a graphical user interface window titled "modifier". It contains a form with the following fields and values:

Label	Value
identifiant:	01
nom:	Beghdad
prenom:	Rachide
grade:	Professeur
specialite:	reseau informatique
num telephone:	0556986925
email:	baghdad.rachide@gamil.com
statut:	permanant(e)
affiliation:	informatique

At the bottom of the form, there are two buttons: "modifier" and "annuler".

FIG. 3.9 – *Modifier enseignant*

### 3.2.4 Génération d'un emploi du temps

Pour imprimer l'emploi du temps il suffit de cliquer sur le bouton "imprime" qui apparait sur l'interface emplois du temps. La génération du fichier est sur la figure suivante :

		niveau: I2 section: A					
		8h00-9h30	9h40-11h20	11h20-12h50	13h00-14h30	14h40-16h10	16h20-17h50
sam							
dim	tp SE Sabri BNS4 A1 tp BDD Aabache B1S23 A5	tp BDD Aabache B1S24 A2 tp SE Sabri BNS4 A3 td TG Slimani B8S10 A4	td BDD Demouche B8S2 A1 tp BDD Khanouche B1S24 A3 tp SE Sabri BNS4 A4 td TG Slimani B8S13 A5			cours AJEL Aissani A22	tp BDD Farah B1S23 A1 tp SE Baadache BNS4 A2
lun	tp DAWEB Tasoult B1S23 A1 tp RX Houha HYDS6 A3 tp BDD Farah B1S4 A4	tp DAWEB Tasoult B1S23 A3 tp RX Houha HYDS6 A4 tp SE Sabri BNS4 A5			cours BDD Tahakourt A22	cours TG Slimani A22	td TG Slimani B8S13 A3 td BDD Boutrid B8S14 A4
mar	tp RX Houha HYDS6 A2 td ANG Rahmani B8S11 A4 tp DAWEB Tasoult B1S3 A5	cours DAWEB Sider A22	cours SE Atmani A22		td GL Kouicem B8S2 A5	td GL Kouicem B8S10 A2 td BDD Khanouche B8S13 A3 td SE Atmani B8S17 A4 tp RX Betaita B1S4 A5	td SE Atmani B8S15 A2 td GL Kouicem B8S13 A3
mer	cours RX Touazi A22	td TG Slimani B8S13 A1 td RX Larbi B8S10 A2 tp DAWEB Kesri B1S24 A4 td BDD Mir B8S11 A5	td GL Kouicem B8S13 A1 td TG Slimani B8S15 A2 td RX Larbi B8S14 A3		td GL Kouicem B8S10 A4 td RX Larbi B8S13 A5	cours GL Kouicem A22	
jeu	td RX Larbi B8S14 A1 td ANG Rahmani B8S10 A2 td SE Atmani B8S13 A3	td ANG Rahmani B8S14 A1 tp DAWEB Kesri B1S24 A2 td RX Larbi B8S17 A4 td SE Atmani B8S16 A5	td BDD Mir B8S17 A2 td ANG Rahmani B8S16 A5		td ANG Rahmani B8S14 A3		

FIG. 3.10 – Génération d'un emploi du temps

### 3.2.5 Planning des examens

Pour imprimer un planning des examens il suffit de cliquer sur le bouton "imprimer" qui apparait sur l'interface imprimer examen.



FIG. 3.11 – Planning des examens

### 3.2.6 Ajouter un thème

Pour ajouter un thème on accède à l'interface liste des thèmes puis on cliquant sur le bouton ajouter. L'interface suivante s'affiche, on remplissant tous les champs du formulaire le thème sera bien ajouter à la liste des thèmes après un clic sur le bouton ajouter.

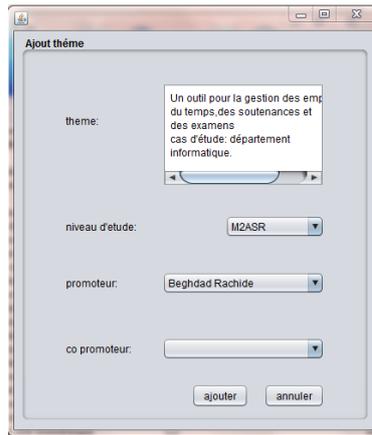


FIG. 3.12 – Ajouter thème

### 3.2.7 Ajouter une soutenance

Pour ajouter une soutenance on clique sur le bouton ajouter de la liste des soutenances. L'interface suivante s'affiche.

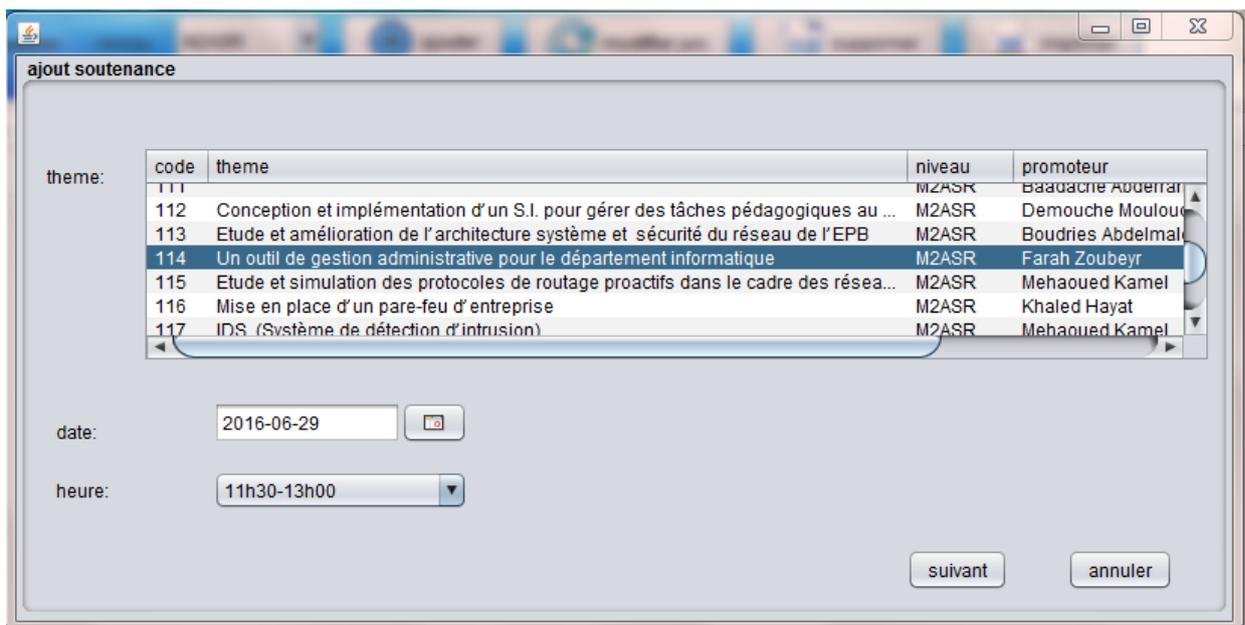


FIG. 3.13 – Ajouter une soutenance

En cliquant sur suivant dans l'interface ajouter soutenance, le formulaire suivant s'affiche en permettant l'ajout d'un membre de jury, quel que soit president, examinateur1 ou examinateur2 ainsi que l'ajout d'un local .

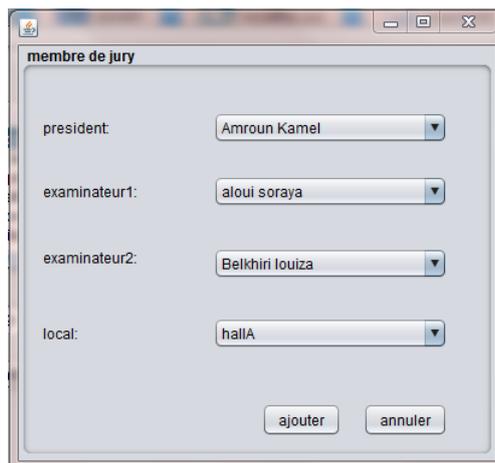


FIG. 3.14 – *Membre de jury*

### 3.3 Conclusion

Dans ce chapitre, nous avons présenté les outils de développement que nous avons utilisé dans la réalisation de notre application , tout en justifiant nos choix technologiques. En effet, l'utilisation du MySQL et du java sous NetBeans comme environnement de développement.

A la fin de ce chapitre, nous avons présenté quelques interfaces, constituant notre application, et que nous avons jugé les plus importantes.

# Conclusion générale

Dans ce travail nous avons réalisé une application pour la gestion des emplois du temps, des examens et des soutenances pour le département d'informatique de l'université A. Mira-Béjaïa.

La première partie de notre travail a visé la présentation de département d'informatique, ses missions, quelques concepts de la planification et de la pédagogie, à la fin de ce chapitre nous avons définis les contraintes physiques et pédagogiques.

En second lieu, nous avons introduit le langage de modélisation UML2 et le processus unifié qu'on a suivi durant tout le processus de développement.

Après ceci, nous avons fait une spécification et analyse des fonctionnalités de l'application à travers les diagrammes de cas d'utilisation, de séquence et d'activité. Dans la dernière partie, nous avons entamé la conception statique dans laquelle nous avons décrit le diagramme de classes associé au projet, suivi du modèle relationnel de données obtenu par l'application des règles de passage.

En fin nous avons réalisé notre application en utilisant plusieurs outils de développement dédiés à la programmation.

Nous avons réussi à mettre en œuvre notre système qui nous a permis de répondre aux exigences des usagers et de simplifier la tâche de gestion à l'administrateur.

Cette expérience nous a permis d'acquérir des compétences et d'enrichir nos connaissances dans le domaine du développement. Ceci nous a permis de mettre en œuvre une application assurant les fonctionnalités attendues, donc nous pouvons affirmer l'atteinte des objectifs visés par ce travail.

Comme perspectives, nous souhaitons l'automatisation de notre application.

# Bibliographie

- [1] Troudi. F, Résolution du problème de l'emploi du temps : Proposition d'un algorithme évolutionnaire multi objectif, mémoire Magister en Informatique, Université Mentouri-Constantine, 2006.
- [2] [Http : //futurcpe.free.fr/donnees/histoire/pedagogie](http://futurcpe.free.fr/donnees/histoire/pedagogie).
- [3] BOUZIANE. H, Un système automatique pour la génération des emplois du temps, Mémoire master en Informatique Université Abou Bakr Belkaid-Tlemcen, 2012.
- [4] Fadjra. D, Noussiba. G, Conception et réalisation d'un portail web (e-université) pour le suivi pédagogique des enseignants et l'évaluation des étudiants, Université Kasdi Merbah-Ouargla, 2013.
- [5] [Http : //uml.free.fr](http://uml.free.fr).
- [6] [Http : // UML.developpez.com/Lp/cours/uml-free](http://UML.developpez.com/Lp/cours/uml-free).
- [7] Pascal Roques, "UML2 Modéliser une application Web", 3e édition EYROLLES, 2000.
- [8] Joseph. Gabay et David. Gabay, "UML2 Analyse et conception", 1ère édition. DUNOD", 2008.
- [9] Pascal. Roque et Franck. Vallée, " ULM 2 en action, de l'analyse des besoins à la conception ", 4ème édition. EYROLLES, 2007.
- [10] Pascal. Roques, "Les cahiers du programmeur UM2, modélisé une application web", 4ème édition, EYROLLES, 2007.
- [11] Pascal. Roque, " ULM 2 en action, de l'analyse des besoins à la conception ", 4ème édition. EYROLLES, 2009.
- [12] Rémy. Fanader et Hervé. Leroux, "UML principes de modélisation", DUNOD, paris", 1999.
- [13] Pascal. Roques, "UML2 par la pratique", Edition EYROLLES. Paris 5, 2006.
- [14] Joseph. Gabay et David. Gabay, "UML2 Analyse et conception", 1ère édition. DUNOD", 2008

- [15] Gilles et Roy, "UML2 modéliser une application web", EYROLLES.Paris, 4ème édition, 2008.
- [16] [www.maxiscience.com/wampserveur/tout-savoir.html](http://www.maxiscience.com/wampserveur/tout-savoir.html).
- [17] Aarry Lellman, "PHP6 et MySQL5 ", Avril 2008.
- [18] Jean Michel Aquilina, "Aide -mémoire MYSQL", Novembre 2002.
- [19] BENSAPHLA. TANI Hidayet, DIB. Nesrine, Mémoire de Licence en Informatique Gestion d'emploi du temps des soutenances le 09 Juin 2014.
- [20] Doudoux. M , Développons en java .livre, 2013.
- [21] Kambouche. F, Bengoudifa. A, Gestion d'emploi du temps des soutenances, Université Abou Bakr Belkaid- Tlemcen 2014.