



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE

DEPARTEMENT INFORMATIQUE

MEMOIRE DE FIN D'ETUDES

En vue de l'obtention du diplôme de Master2 en informatique

Option : Conduite de Projets Informatiques

Thème

CONCEPTION ET REALISATION D'UNE
APPLICATION SEMI-AUTOMATIQUE POUR
LA GESTION DES EMPLOIS DE TEMPS

CAS : DEPARTEMENT INFORMATIQUE
(UMMTO)

Dirigé par :

Mr Y.Yacine

Réalisé par :

Proposé par :

Mr M.RAMDHANE

M^{elle} : NAMANE LYNDA

M^{elle} : NEHAND AMEL

Promotion : 2014/2015



Dédicaces

Je dédie ce travail à

Mes très chers parents.

Mes très chers grands parents.

Mes très chères sœurs.

Mes très chers frères.

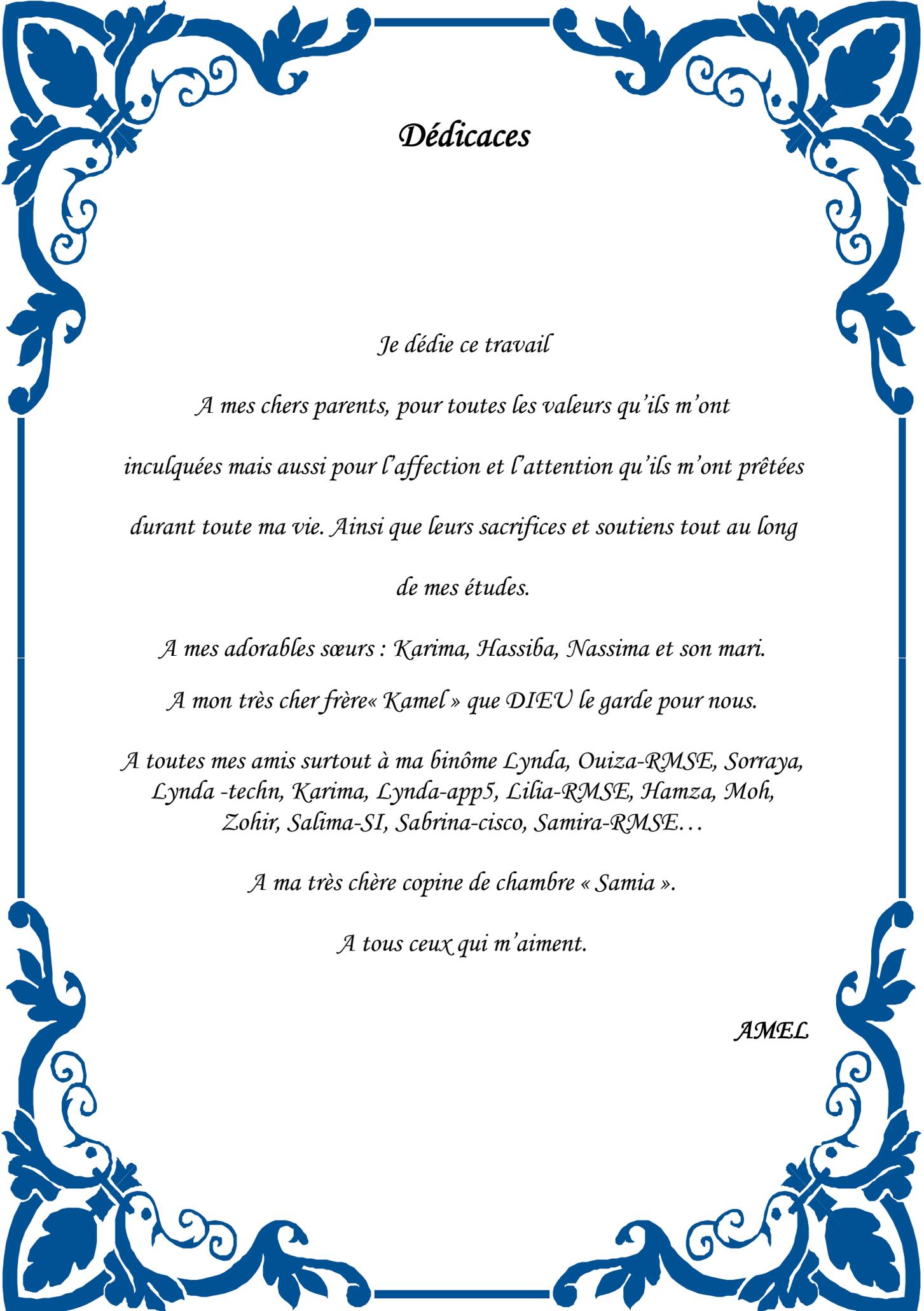
A toute la famille

A mon binôme Amel

Tous mes amis, ainsi que toute la promotion du Master 2 en

Informatique CPI 2014/2015

Lynda



Dédicaces

Je dédie ce travail

*A mes chers parents, pour toutes les valeurs qu'ils m'ont
inculquées mais aussi pour l'affection et l'attention qu'ils m'ont prêtées
durant toute ma vie. Ainsi que leurs sacrifices et soutiens tout au long
de mes études.*

A mes adorables sœurs : Karima, Hassiba, Nassima et son mari.

A mon très cher frère « Kamel » que DIEU le garde pour nous.

*A toutes mes amis surtout à ma binôme Lynda, Ouiza-RMSE, Sorraya,
Lynda -techn, Karima, Lynda-app5, Lilia-RMSE, Hamza, Moh,
Zohir, Salima-SI, Sabrina-cisco, Samira-RMSE...*

A ma très chère copine de chambre « Samia ».

A tous ceux qui m'aiment.

AMEL

Remerciement

*Nous tenons à remercier notre promoteur
Mr Y. Yacine pour la qualité de son encadrement, et son suivi durant
toute la durée du projet. Nous tenons, également, à lui exprimer notre
reconnaissance pour le temps qu'il nous a consacré, ainsi que pour ses
encouragements.*

*Que les membres du jury trouvent ici nos plus vifs
remerciements pour avoir accepté d'honorer par leur jugement notre
travail.*

*Notre profonde gratitude et sincères remerciements vont à tous les
professeurs qui nous ont suivis durant notre parcours d'étude.*

*Nous réservons ici une place particulière pour remercier vivement toute
notre section M2 CPI « 2014/2015 », et tous ceux qui nous ont aidés et
encouragés à la réalisation de
ce modeste travail en particulier Mr NESSAH .A.*

Sommaire

Introduction générale	
Chapitre I Généralités sur les emplois de temps	
Introduction	1
I. La problématique de la planification d'horaires de travail	1
I.1 La planification	1
I.2 La définition d'un planning	1
I.3 L'utilité d'un planning	2
I.4 L'évaluation un planning	3
I.5 Le planificateur	3
II. Différents types de plannings :	3
II.1 Types de plannings dans le domaine de la santé :	4
II.2 Types de plannings dans le domaine de transport :	4
II.3 Types de plannings dans le domaine de la pédagogie :	5
Conclusion	6
Chapitre II Analyse et conception	
I. Analyse	
I.1. Organisme d'accueil et étude de l'existant	7
I.1.1. Organisme d'accueil	7
I.1.1.1. Présentation de l'organisme d'accueil	7
I.1.1.2. Champs d'étude	8
I.1.1.2.1. Mission	8
I.1.1.2.2 Infrastructures et moyens humains	8
I.1.2. Etude de l'existant	8
I.1.2.1 Etude du système pédagogique	9
I.1.2.2. L'activité pédagogique	10
I.1.2.3. Les ressources	10
I.1.2.3.1. Les ressources de type « salle »	11
I.1.2.3.2. Les ressources de type « enseignant »	11
I.1.2.3.3. Les ressources de type « groupe »	11
I.1.2.3.4. Les ressources de type « matériel »	11
I.1.2.4. La notion du temps	12
I.1.2.4.1. Les entités temporelles	12
I.1.2.4.2. Les séances et les réservations	12
I.1.2.5. Les filières	13
I.1.2.6. Les contraintes	13
I.1.2.6.1. Les contraintes physiques	13
I.1.2.6.2. Les contraintes pédagogiques	13
I.2 Objectif du travail :	14

Sommaire

I.3 Identification des acteurs et des tâches :	14
I.3.1 Les acteurs :	14
I.3.2 Spécification des tâches :	15
I.4 Diagramme de contexte :	16
I.5 Etablissement des cas d'utilisation :	17
I.5.1 Le Diagramme de cas d'utilisation pour l'acteur « Enseignant »	17
I.5.2 Le Diagramme de cas d'utilisation pour l'acteur « Planificateur »	18
I.5.3 Le Diagramme de cas d'utilisation pour l'acteur « Administrateur »	18
I.6. Description textuelle des cas d'utilisation :	19
I.6.1 Description de cas d'utilisation « Remplir la fiche de disponibilité »	19
I.6 .2 Description de cas d'utilisation« ajouter un module »	20
I.6 .3 Description de cas d'utilisation« Créer un emploi de temps »	20
II. Conception	
II.1. Les diagrammes de séquence :	21
II.1.1 Diagramme de séquence pour le cas d'utilisation « authentification».	21
II.1.2 Diagramme de séquence pour le cas d'utilisation «Créer une formation ».	22
II.1.3 Diagramme de séquence pour le cas d'utilisation «Remplir une fiche de vœux ».	23
II.2. Diagrammes d'activité	24
II.2.1. Diagramme d'activité « ajouter une salle »	24
II.2.2. Diagramme d'activité « créer un emploi de temps ».	25
II.2.3. Diagramme d'activité « Authentification ».	25
II.3. Diagrammes de classe :	26
II.3.1. Diagramme de classe : Créer un emploi de temps	26
II.3.2 Diagramme de classe : Ajouter une formation	26
II.3.3. Diagramme de classe : Remplir la fiche de vœux	27
III. Conception de la BDD	
III .1. Les règles de gestion	27
III.1.1. Les contraintes dures	28
III.1.2. Les contraintes de préférence	28
III.2. Diagramme de classe général	29
III. 3. Les règles de passage de MLD vers le MPD.	30
III.4. Le modèle relationnel de données.	30
Conclusion	31
Chapitre III Réalisation	
Introduction	32
I. Présentation « java »	32
I.1. Java : Une technologie Java	32
I.2. Java : un langage de programmation	32
I.3. Java : Une plateforme	33

Sommaire

II. Le JDK	34
III. Les interfaces graphiques du Java	34
IV. Outils de développement	35
IV.1 Présentation de NetBeans IDE :	35
IV.2.Présentation de WAMPserver	36
IV.2.1. PhpMyAdmin	36
IV.3. Le JDBC	37
III. Le modèle physique de données	38
III.1. Les objectifs du MPD	38
III.2. Structure des tables utilisées	38
IV. Présentation de l'application	43
IV.1. Connexion	43
IV.2. Espace administrateur	43
IV.3. Espace enseignant	45
IV.4. Espace planificateur	47
Conclusion	49
Conclusion générale	
Annexe	

Liste des illustrations

Figure II.1 : Diagramme de contexte.	15
Figure II.2 : Le Diagramme de cas d'utilisation pour l'acteur « Enseignant »	18
Figure II.3 : Le Diagramme de cas d'utilisation pour l'acteur « Planificateur »	18
Figure II.4 : Le Diagramme de cas d'utilisation pour l'acteur « Administrateur »	17
Figure II.5: Diagramme de séquence pour la réalisation de cas d'utilisation « authentification ».	21
Figure II.6 : Diagramme de séquence pour la réalisation du cas d'utilisation « Créer une formation ».	22
Figure II.7 : Diagramme de séquence pour la réalisation du cas d'utilisation « Remplir une fiche de vœux ».	23
Figure II.8 : diagramme d'activité « ajouter une salle »	24
Figure II.9 : Diagramme d'activité pour le cas d'utilisation « créer un emploi de temps ».	25
Figure II.10 : Diagramme d'activité pour le cas d'utilisation « Authentification ».	25
Figure II.11 : Diagramme de classe « Créer un emploi de temps »	26
Figure II.12 : Diagramme de classe « Ajouter une formation »	26
Figure II.13 : Diagramme de classe « Remplir la fiche de vœux »	27
Figure II.14 : Diagramme de classe générale	29
Figure III.1. Interface de l'IDE Netbeans	36
Figure III.2. Accès à la base de données ‘’ planning ‘’ à partir de PhpMyAdmin	37
Figure III.3. Interface d'authentification	43
Figure III.4.Gestion des utilisateurs	44
Figure III.5.Ajouter un utilisateur	44
Figure III.6.Fiche de vœux d'un enseignant	45

Liste des illustrations

Figure III.7. La disponibilité	46
Figure III.8.Gestion des formations	47
Figure III.9.Gestion des modules	47
Figure III.10 La spécification d'un emploi de temps.	48
Figure III.11.Programmer une séance	48
Figure III.12.Affichage d'un planning	49

Liste des tableaux

Tableau II.1 : Spécification des taches.	15, 16
Tableau II.2 : Description textuelle de cas d'utilisation « Remplir la fiche de disponibilité »	19
Tableau II.3 : Description textuelle de cas d'utilisation « ajouter un module »	20
Tableau II.4 : Description textuelle de cas d'utilisation « Créer un emploi de temps »	20

Introduction générale

Introduction générale

La construction d'un emploi du temps est un travail long et difficile à mettre en place et qui doit être répétée à intervalle régulier. C'est pour cette raison que l'automatisation d'emplois du temps, est une tâche d'une grande importance car elle permet de gagner beaucoup d'heures de travail pour les employés, les institutions et les entreprises, et de fournir des solutions optimales avec satisfaction de contraintes en quelques minutes, ce qui peut accroître la productivité, la qualité de l'éducation, qualité des services et enfin, la qualité de vie.

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 recherche.

Dans le domaine de l'éducation, 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 classifiées en deux grandes catégories qui sont : emploi du temps pour examen et emploi du temps universitaire.

La problématique de génération d'emploi du temps du département d'informatique de l'UMMTO représente un vrai challenge semestriel.

L'existence d'un horaire respectant à la lettre 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 tolérant la présence de 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, Intervenant, Etudiants,...etc.)

Dans ce mémoire nous allons nous intéresser au problème de génération semi-automatique d'emploi du temps universitaire et plus précisément à celui du département d'informatique de l'UMMTO.

Introduction générale

Ce dernier est organisé en trois chapitres. Dans le premier, nous introduisons les notions liées aux problèmes de planification d'horaires de travail et les différents types de plannings dans différents domaines de travail avec un aperçu des méthodes utilisées pour la réalisation de ces plannings.

Le deuxième chapitre porte L'analyse et la conception, ou nous avons opté pour le langage *UML* pour la modélisation de notre travail à l'aide des différents formalismes, étant le mieux adapté pour les applications web.

Et le dernier chapitre est consacré à la réalisation et l'implémentation de l'application, elle consiste à présenter les outils de développement, les langages et les techniques utilisées pour l'implémentation du projet.

Chapitre I

Généralités sur les emplois de temps

Introduction :

Ce chapitre met en scène la problématique de la planification des horaires dans un contexte général et sa complexité au quotidien dans les organismes. En effet, la question de l'aménagement du temps de travail et de ses enjeux préoccupe toute société ou établissement actif ce qui a incité les chercheurs à proposer des méthodes et des techniques pour aider à gérer au mieux les horaires de travail. Pour cela nous définissons les différents types de plannings dans différents domaines de travail et plus particulièrement dans le domaine pédagogique.

I. La problématique de la planification d'horaires de travail [1]:

La planification d'horaires de travail est un processus très complexe, qui vise à organiser des activités humaines (principalement de travail) dans le temps et à optimiser l'utilisation des ressources, de façon à couvrir un besoin exprimé par une charge de travail prévisionnelle sous diverses contraintes. Elle aboutit à des programmes définissant les horaires de travail et de repos.

Pour mieux cerner ce qui est la planification et la complexité à sa réalisation, on s'intéresse à un ensemble de définitions :

I.1 La planification [1] :

La planification est un instrument de gestion dont l'objectif est d'aboutir à des programmes permettant d'organiser et planifier le travail. Ceci passe par la détermination des capacités de tout un chacun et par le recensement des activités futures et des besoins en personnel.

La planification vise à affecter les ressources humaines pour chaque intervalle de temps, de telle manière que les besoins par intervalle et les différentes contraintes soient satisfaites.

I.2 Définition d'un planning [2] :

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

Les plannings peuvent être utilisés pour planifier les horaires de présences du personnel ou les tâches effectuées par le personnel :

➤ **Planning des horaires de présence :**

Ce type de planning est utilisé pour prévoir les horaires de présence du personnel sans préciser les tâches journalières à effectuer soit pour des raisons de sécurité, soit pour une meilleure souplesse.

➤ **Planning des tâches :**

Ce type de planning est utilisé dans des organismes et entreprises à haute technicité, comportant plusieurs métiers et compétences distincts, où il est souhaitable d'affecter le personnel en fonction des tâches. Ce qui exige une décomposition fine des opérations et le repérage des tâches que chaque personne est capable d'accomplir.

Les plannings peuvent être journaliers, hebdomadaires, mensuels ou annuels.

Certains plannings sont cycliques, s'ils reflètent une certaine périodicité des horaires individuels c'est à dire si au bout d'une durée D (mesurée généralement en semaine). Autrement, ils sont dits acycliques c'est à dire ils sont différents chaque semaine.

I.3 L'utilité d'un plannings [1] :

Depuis le début des années 80, la gestion des ressources humaines a été reconnue comme une activité stratégique pour les organismes et les entreprises. 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, établissements éducatifs, etc...),
- Aux entreprises cherchant à se rendre plus accessibles à la clientèle d'étaler les horaires d'ouverture (grands magasins, banques, 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.

I.4 Evaluation d'un planning [3] :

Pour que les plannings élaborés soient satisfaisants, ils doivent vérifier un ensemble de contraintes et établir un meilleur compromis entre les différents acteurs (exemple : le chef d'entreprise, le planificateur, le commercial, le syndicaliste et le salarié).

Lorsque les différentes solutions alternatives sont connues, une négociation se déroule de la manière suivante : chaque acteur donne son opinion. Les points d'accord sont très vite expédiés et les points litigieux sont débattus. Et des solutions de compromis sont dégagées.

Les difficultés de négociation augmentent avec le nombre d'acteurs et le nombre de solutions alternatives. Les moyens informatiques apportent une certaine aide notamment dans l'acquisition et la confrontation des données individuelles.

I.5 Le planificateur :

Dans la plupart des entreprises et organismes, cette tâche peut être déléguée à des cadres de l'entreprise appelés planificateurs.

Le planificateur doit prendre la décision qui correspond le mieux aux préférences des différents acteurs, justifier son choix, car son expérience de la tâche fait de lui la personne la mieux placée 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 pertinent.

II. Différents types de plannings :

Si créer un planning optimisé d'une journée est aisé, mais pour créer un bon planning pour un mois ou une année est beaucoup plus complexe. En plus de la complexité, il faut tenir compte de la diversité des contraintes applicables.

Pour ce qui suit, on évoquera les différents types de plannings.

II.1 Types de plannings dans le domaine de la santé :

Les plannings dans le domaine de la santé sont des calendriers de travail où figurent à la fois le temps, et l'affectation des personnels (jours et horaires de travail, repos). Ils sont établis au niveau de chaque équipe, ils sont à la fois une tâche, un document d'organisation du travail, et un élément contribuant à la gestion administrative du personnel. Cette tâche est parmi les plus difficiles et les plus délicates.

Difficile parce qu'elle repose sur la recherche de solutions combinatoire, répond à des contraintes multiples, remise en cause de manière fréquente par l'absentéisme et délicate car elle impose toujours une négociation avec les acteurs (médecins, infirmiers...) et la direction du service de soins et l'administration.

Les documents établis sont des calendriers sur lesquels on inscrit les affectations des médecins et des infirmiers ; ils sont généralement des tableaux à double entrée tel que en ligne le personnel et en colonne le temps.

II.2 Types de plannings dans le domaine de transport :

Le transport est une activité complexe qui fait intervenir des investissements lourds, du personnel qualifié et une informatique très coûteuse.

En effet, dans le transport routier, il est toujours nécessaire de gérer aux mieux les ressources existantes en optimisant les investissements. Comme les clients exigent toujours plus de flexibilité, il faut offrir des services sur mesure, replanifier en permanence et en temps réel et gérer le personnel qualifié qui est une opération très complexe car il faut tenir compte de plusieurs contraintes (contrats, temps de travail, manque du personnel qualifié,...).

Dans le transport maritime, la gestion des escales et la gestion du personnel docker est aussi une activité complexe qui nécessite un effort considérable de la part des planificateurs. Les navires doivent rester à quai un temps minimum et les équipes docker doivent être disponibles. Cette activité représente un enjeu économique majeur.

En effet, la qualité de la planification des travaux influe directement sur la rentabilité de l'activité de l'entreprise d'où la nécessité de la gestion des escales (planifier le placement des navires sur les quais, planifier la disponibilité des ressources matérielles nécessaires, positionner des équipes

sur des navires). Afin d'optimiser les coûts liés aux chargements et déchargements des navires et la gestion du personnel dockers (les besoins en équipe et en qualification pour chaque tâche issue de la gestion des escales et les contraintes liées à la gestion de personnel) afin d'optimiser l'affectation des ressources tout en tenant compte des contraintes liées à l'organisation du travail.

Dans le transport aérien, la gestion des flux de trafic aérien correspond aussi à des problèmes d'optimisation dont la résolution est très complexe. En effet, le contrôle de la circulation aérienne organise les flux aériens afin d'assurer la sécurité des vols (en terme de risque de collision), d'améliorer la capacité du réseau de routes sur lequel les avions se déplacent et de construire des programmes de vols optimisés.

II.3 Types de plannings dans le domaine de la pédagogie :

La confection d'horaires (ou confection d'emploi du temps) dans les établissements universitaires est un travail très important. Pour fournir une solution, nécessite d'être capable de s'adapter aux changements dynamiques de l'environnement en tenant compte de la diversité des contraintes telles que l'interdépendance des programmes d'enseignement, la multitude des matières étudiées et les contraintes sur ces matières (cours, TD, TP...), la durée des cours, les contraintes de disponibilité des enseignants, la disponibilité limitée des salles. C'est un problème qui peut être défini comme un problème qui fait assigner quelques événements dans un nombre limité de périodes. Il peut être divisé en deux catégories principales : la confection d'horaires des cours et la confection d'horaires des examens. [4]

La confection de plannings d'horaires est donc une tâche très difficile et sa solution manuelle peut exiger beaucoup d'effort ce qui a attiré énormément l'attention de la communauté scientifique.

Les problèmes des emplois de temps s'étendent de la construction des emplois du temps semestriels ou annuels dans les universités, écoles ou collèges aux emplois du temps d'examen à la fin de ces périodes. Les premières activités d'emploi du temps ont été effectuées manuellement et un emploi du temps typique, une fois construit est resté statique avec seulement quelques changements nécessaires. [5]

Cependant la nature des enseignements a changé considérablement au cours des années et ainsi les exigences en matière de confection d'emploi du temps sont devenues beaucoup plus compliquées qu'ils ont eu l'habitude de l'être.

Par conséquent le besoin de la génération automatisée d'emploi du temps augmente et ainsi le développement d'un système de génération d'emploi du temps qui produit des solutions valables est essentiel.

En conséquence, pendant les 30 dernières années, beaucoup d'approches liées à l'automatisation des emplois du temps ont été publiées aux conférences et journaux. De plus, plusieurs applications ont été développés et mises en œuvres avec divers succès. [6]

Conclusion

On peut conclure que la planification des horaires présente des enjeux à la fois sur un plan économique et un plan social. Toutefois, sa complexité impose de s'appuyer sur une démarche scientifique pour apporter des réponses pragmatiques à une catégorie générale de problèmes.

Il s'agit donc de développer des outils de planification d'horaires, basés sur des techniques efficaces d'optimisation de ressources qui permettent de construire des programmes de travail, respectant la réglementation du travail et garantissant une bonne couverture de charge tout en limitant les coûts.

Parmi tous les types de plannings cités, c'est sur les plannings pédagogiques que nous allons porter notre intérêt, et plus particulièrement sur les plannings ou emploi du temps des cours des écoles de formation.

Au cours de ce chapitre, nous avons présenté certains principes concernant de la planification d'horaire de travail. Le chapitre suivant, sera consacré à l'analyse et la conception de notre application.

Chapitre II

Analyse et conception

Introduction

Dans les projets de gestion d'emploi du temps, l'optimisation de solution est un réel problème du fait que la recherche d'une solution est complexe car il s'agit d'un problème de résolution de contraintes.

L'objectif principal de notre étude est de mettre en œuvre une application qui rendra le processus de génération des emplois du temps plus efficace en termes d'implication d'utilisateurs, et qui est capable d'agir sur les changements des contraintes.

Dans ce chapitre, nous avons deux phases Analyse et conception ; la première permet de mettre en évidence les différents acteurs intervenants dans le système projeté, la deuxième s'appuyant sur les résultats de l'analyse permet la description du futur système en utilisant les modèles de données et de traitements. Pour ce faire, notre démarche va s'appuyer sur le langage UML.

I. Analyse

I.1. Organisme d'accueil et étude de l'existant

L'analyse de l'existant est une étape importante dans le cycle de vie d'un système, il s'agit de connaître la situation actuelle de l'organisation pour pouvoir porter un jugement juste. Ainsi, l'analyse du système existant doit nous fournir toute l'information nécessaire, afin d'établir une bonne conception et de proposer de bonnes solutions.

Dans ce chapitre, nous allons présenter une étude générale du système pédagogique universitaire (ressources logicielles et matérielles) concerné par notre étude.

Par la suite, un intérêt particulier est porté à l'application existante de gestion des emplois du temps du département d'informatique.

Dans un premier temps nous présenterons le département informatique de l'Université Mouloud Mammeri de Tizi-Ouzou (UMMTO).

I.1.1. Organisme d'accueil

I.1.1.1. Présentation de l'organisme d'accueil

Le département d'informatique de l'UMMTO avait assuré depuis sa création, la formation du système classique des deux cycles de la graduation à savoir le cycle court en vue de l'obtention du diplôme de DEUA en informatique et le cycle long en vue de l'obtention du diplôme d'ingénieur d'état en informatique (options : système d'information avancé et informatique industrielle).

A partir de l'année universitaire (2005/2006) le département d'informatique a commencé à prendre en charge la formation du système L.M.D (Licence, Master et Doctorat).

I.1.1.2. Champs d'étude

I.1.1.2.1 Mission

De par les missions qui lui ont été conférées, l'UMMTO s'est fixé comme objectif de former des éléments compétents capables de relever les défis du futur et de contribuer au développement national.

Ses principales missions se résument comme suit :

- **Les formations du 1er cycle (Licence :** Informatique général et Multimédia).
- **Les formations du 2eme cycle (Master académique :** Système d'Information (SI), conduite de projet informatique (CPI), Réseau mobile et système embarqué(RMSE) **et Master professionnelle :** ingénieria en système d'information (ISI)).
- **La formation du 3eme cycle (Doctorat)**
 - Le développement et la promotion de la recherche scientifique dans les domaines de technologie de pointe notamment les technologies de l'information et de la communication, en coopération avec des centres de recherches et des universités nationaux et internationaux.
 - L'assistance du secteur industriel et socioprofessionnel en matière d'informatisation.

I.1.1.2.1 Infrastructures et moyens humains

Pour son bon fonctionnement, la direction des études dispose d'infrastructures et de moyens conséquents, à savoir : Un seul amphithéâtre, Douze salles de travaux dirigés et Onze laboratoires de travaux pratiques dotés de matériels performants.

I.1.2. Etude de l'existant

Chaque année, les responsables pédagogiques de la direction des études ont pour mission de concevoir les emplois du temps des différentes filières 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 salles, les équipements, etc.).

Ainsi, l'idée de mettre en place un système de gestion des emplois du temps est née, dont l'objectif est :

- La génération semi-automatique des emplois du temps.
- Consultation et suivi des emplois du temps.

Par conséquent, notre étude concerne la direction des études et en particulier le service du suivi des enseignements et de l'évaluation.

I.1.2.1. 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'UMMTO est une université qui regroupe différentes formations du LMD. Elles ont une durée qui varie entre trois à huit ans. 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...

Administrativement, les enseignants doivent assurer un nombre minimal d'heures qui est défini dans leur statut. Selon le type d'enseignement, on pondère le nombre d'heures par un coefficient. Ainsi, par exemple, une heure de cours est comptabilisée comme 1,5 heure de travaux dirigés. Lorsqu'un enseignant accepte la responsabilité d'un enseignement, il est tenu d'en respecter le volume horaire prévu par le responsable pédagogique.

En cas d'absence, un enseignant doit prévoir des séances de rattrapage. Il doit donc connaître précisément la disponibilité des ressources de sa séance.

Cette organisation garantit que tous les étudiants qui suivent une même formation auront eu le même volume horaire d'enseignement.

Chaque formation est gérée par son responsable pédagogique qui crée les emplois du temps en concertation avec les enseignants mais aussi, moins souvent, avec les responsables des autres formations. Les enseignants peuvent assurer leurs enseignements dans différentes formations. Lors de la création des emplois du temps, les responsables pédagogiques ne tiennent généralement pas compte explicitement des salles disponibles.

Dans la réalité, les responsables pédagogiques s'appuient sur l'expérience des années précédentes pour planifier les enseignements.

Une fois que les emplois du temps sont créés, l'attribution des salles est assurée par différents responsables qui peuvent être des personnels administratifs ou techniques ou même des enseignants selon le type des salles de cours, TD ou TP.

L'emploi du temps sert également aux personnels administratifs pour suivre le travail des intervenants en vue de leur rémunération, pour prévoir l'achat de nouveaux équipements ou de leur mise en maintenance, pour argumenter la demande d'acquisition de nouveaux locaux, etc. Sur le plan pédagogique, l'étude des emplois du temps sert à mettre en évidence certains problèmes tels que les surcharges de travail des intervenants, les mauvaises répartitions des enseignements, et elle permet ainsi de corriger et améliorer le planning les années suivantes. Le nombre des ressources disponibles croît, malheureusement, bien moins rapidement que le nombre des formations ou que leur diversification.

Cela oblige donc les responsables à « optimiser » la planification des ressources et rend plus délicate leur gestion. Le besoin d'un outil d'aide devient donc essentiel pour cette gestion efficace des ressources disponibles. L'objectif n'étant pas de remplacer les acteurs actuels mais de les aider, il est donc important de considérer les différents profils des futurs utilisateurs, en particulier par rapport à leur maîtrise des outils informatiques. Pour satisfaire un large éventail de profils d'utilisateurs et faciliter l'acceptation de l'outil visé, il convient de soigner en particulier la robustesse de celui-ci vis-à-vis des erreurs de manipulation, la rapidité d'exécution des commandes, la puissance des fonctions assurées.

I.1.2.2. L'activité pédagogique

L'activité pédagogique est modélisée à l'aide de trois entités : les matières, 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.

I.1.2.3. Les ressources

Les ressources considérées sont les entités physiques nécessaires à l'élaboration des emplois du temps. Il s'agit des salles, des enseignants, des groupes, des étudiants 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.

I.1.2.3.1. 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.

I.1.2.3.2. Les ressources de type « enseignant »

L'enseignant désigne une personne pouvant assurer des enseignements. Chaque enseignant est caractérisé par : son nom et son prénom, son grade, sa résidence administrative, sa spécialité. La spécialité renseigne sur les matières que peut enseigner un enseignant.

I.1.2.3.3. Les ressources de type « groupe »

Un groupe est un ensemble d'étudiants. Il se compose d'autres groupes qui peuvent être réduits à un seul étudiant (un binôme ou monôme). Cette modélisation ne fait pas de distinction entre groupe et étudiant. En général les étudiants d'une filière sont décomposés en section et chaque section est décomposée en plusieurs groupes.

I.1.2.3.4. 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.

I.1.2.4. La notion du temps

Les analyses menées avec les utilisateurs mettent en avant différentes perceptions du temps. Pour représenter le temps, deux notions de base sont nécessaires : l'*instant* et la *durée*. De plus, pour le problème des emplois du temps on rajoute la notion de *fréquence*.

Cette notion de fréquence n'est pas indispensable puisqu'elle peut s'exprimer à partir de celles d'instant et de durée. Dans les emplois du temps, on dira qu'un enseignement a lieu tous les 15 jours (*fréquence*) si la durée qui sépare deux séances consécutives de l'enseignement est d'au moins 15 jours (*durée*). Par exemple, dire « qu'il y a cours de mathématiques tous les sept jours » c'est-à-dire que la durée qui sépare deux séances consécutives de ce cours est de 7 jours. Ainsi on distingue la durée de chaque séance et de la durée qui sépare les séances.

I.1.2.4.1. 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 $HMax=17h00$ par défaut.

Une *durée* est un nombre compris entre $DMin$ et $Dmax = Hmax - HMin$. $DMin$ représente la plus petite unité temporelle disponible.

I.1.2.4.2. Les séances et les réservations

Une *séance* correspond à une instance temporelle d'un enseignement à une date donnée, pendant un créneau précis. L'ensemble des séances d'une ressource est appelé *planning* de cette ressource. Il s'agit de l'ensemble des séances dans lesquelles apparaît la ressource. Ainsi, il est possible de considérer le planning d'une salle au même titre que celui d'un enseignant. La notion de planning apparaît également au niveau des enseignements : cela permet ainsi de manipuler l'évolution dans le temps d'un enseignement particulier.

Une *réservation* correspond à un créneau ajouté au planning d'une ressource. Elle correspond à une option posée sur l'occupation de cette ressource. Par rapport à une séance, une réservation n'est pas associée à un enseignement.

I.1.2.5. Les filières

La notion de filière permet de limiter le nombre des ressources à considérer. En effet, le responsable d'une filière n'a besoin de connaître que le planning des ressources qu'ils utilisent. Cependant, certains conflits ne peuvent être détectés que lorsque les différentes filières sont rassemblées dans une base de données commune. Il s'agit surtout des conflits d'enseignants et de salles puisqu'il est fréquent qu'un enseignant intervienne dans plusieurs filières ou qu'une salle soit partagée par plusieurs filières.

I.1.2.6. Les contraintes

L'analyse sur le terrain montre que les données gérées doivent vérifier certaines contraintes pour garantir leur cohérence. De manière abstraite, les contraintes à respecter peuvent être classées en deux groupes : les *contraintes physiques* et les *contraintes pédagogiques*.

I.1.2.6.1. Les contraintes physiques

Ces contraintes ne doivent pas être violées sinon cela conduirait à des situations conflictuelles. On dira qu'il y a un « conflit physique de ressource » entre deux séances $s1$ et $s2$ si ces deux séances ont une ressource en commun pendant une durée non nulle. Voici les contraintes physiques :

- 1- Une ressource ne peut pas être occupée en même temps dans deux séances différentes ;
- 2- Dès qu'une ressource r est occupée par une séance s , toutes ses ressources filles (RF) sont également occupées par la même séance et cela récursivement (il s'agit d'une contrainte liée à la hiérarchisation) ;
- 3- On ne peut pas mettre plus d'étudiants qu'il n'y a de places dans une salle ;
- 4- Le volume horaire total des séances d'un enseignement ne peut pas dépasser le volume prévu ;
- 5- Les calendriers sont respectés pour toutes les ressources ;

I.1.2.6.2. Les contraintes pédagogiques

Lors des interviews, des contraintes pédagogiques ont pu être identifiées. Elles se différencient des contraintes physiques par le fait qu'elles peuvent éventuellement être violées, dans ce cas on obtient des emplois du temps de moins bonne qualité d'un point de

vue pédagogique. Typiquement ces contraintes sont utilisées pour exprimer ce que doit être un «bon» emploi du temps. Voici quelques exemples de ces contraintes :

- 1- Homogénéiser la durée des séances d'un enseignement,
- 2- Éviter les séances creuses dans l'emploi du temps,
- 3- Éviter de placer plusieurs séances d'un même enseignement dans une journée,
- 4- Éviter de programmer des séances à partir de 17h,
- 5- Éviter de placer des cours l'après-midi,
- 6- Favoriser la régularité dans les emplois du temps.

On peut remarquer que dans l'expression de ces contraintes pédagogiques les termes utilisés ne sont pas précis (éviter, devraient, si possible ...). Ces contraintes sont plus difficiles à formaliser que les contraintes physiques et leur traitement est plus délicat. Il peut y avoir des degrés d'importance entre les contraintes et une même contrainte peut avoir différents «degrés » d'importance. Par exemple, la contrainte « éviter les séances creuses dans l'emploi du temps » représente l'ensemble de toutes les contraintes « éviter les séances creuses d'une durée D dans l'emploi du temps », où $D = [30\text{min}, 1\text{h}, 1\text{h}30, \dots]$. Mais il est moins important d'éviter les séances creuses de 60 minutes que d'éviter les trous de 10 h.

I.2 Objectif du travail

Notre objectif est de développer une application semi-automatique pour la gestion des emplois de temps de façon à satisfaire au mieux un ensemble d'objectifs tels que :

- Améliorer de la qualité de service.
- Améliorer des conditions de travail.
- Eviter la mobilisation de plusieurs personnes, plusieurs jours de travail.
- Faciliter la gestion des emplois de temps.

I.3 Identification des acteurs et leurs tâches

I.3.1. Les acteurs

Dans notre application on distingue trois principaux acteurs, pour chaque acteur on a identifié les fonctionnalités lui permettant d'accomplir son rôle dans le système :

L'administrateur : C'est celui qui est chargé d'administrer l'application, après avoir confirmé son inscription on lui donne les droits d'accès (login, mot de passe) et il devient membre.

- ❖ **L'enseignant** : Toute personne chargée de donner des cours. Après avoir confirmé son inscription on lui donne les codes d'accès (login, mot de passe) par l'administrateur et elle devient membre.
- ❖ **Le planificateur**: C'est celui qui est chargé de gérer les emplois de temps selon les fiches de vœux, le grade de l'enseignant et d'autres contraintes.

I.3.2 Spécification des taches

Le tableau suivant résume les différentes taches assuré par chaque acteur du système.

Acteurs	Tâches	Sous tâches
Planificateur	S'identifier	
	Gestion des emplois de temps	Créer un emploi de temps Supprimer un emploi de temps Consulter un emploi de temps
	Gestion des formations	Ajouter Modifier Supprimer Consulter
	Gestion des modules	Ajouter Modifier Supprimer Consulter
	Gestion des semestres	Ajouter Modifier Supprimer Consulter
	Gestion des salles	Ajouter Modifier Supprimer Consulter
	Gestion des sections	Ajouter Modifier Supprimer Consulter
	Gestion des groupes	Ajouter Modifier

		Supprimer Consulter
	Se déconnecter	
Enseignant	S'identifier	
	Consulter son emploi de temps Remplir la fiche de vœux Remplir la disponibilité Changer son mot de passe	
	Se déconnecter	
Administrateur	S'identifier	
	Gestion des utilisateurs	Ajouter Modifier Supprimer Consulter
	Se déconnecter	

Tableau II.1 : Spécification des tâches.

I.4 Diagramme de contexte

Le diagramme de contexte est un modèle conceptuel de flux qui permet d'avoir une version globale des interactions entre le système et les liens avec l'environnement extérieur. Il permet aussi de bien délimiter le champ de l'étude.

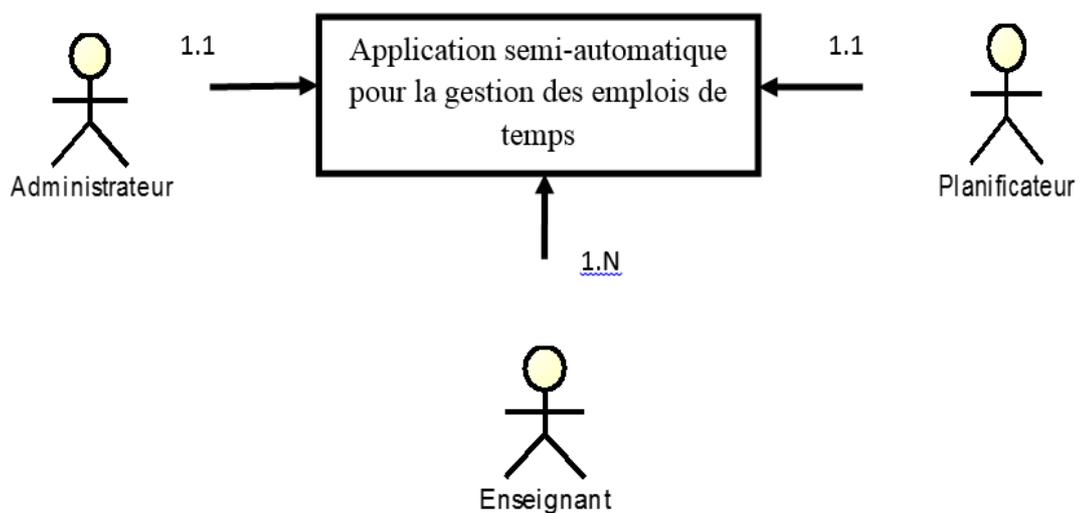


Figure II.1 : Diagramme de contexte.

I.5 Etablissement des cas d'utilisation

I.5.1 Le Diagramme de cas d'utilisation pour l'acteur « Enseignant »

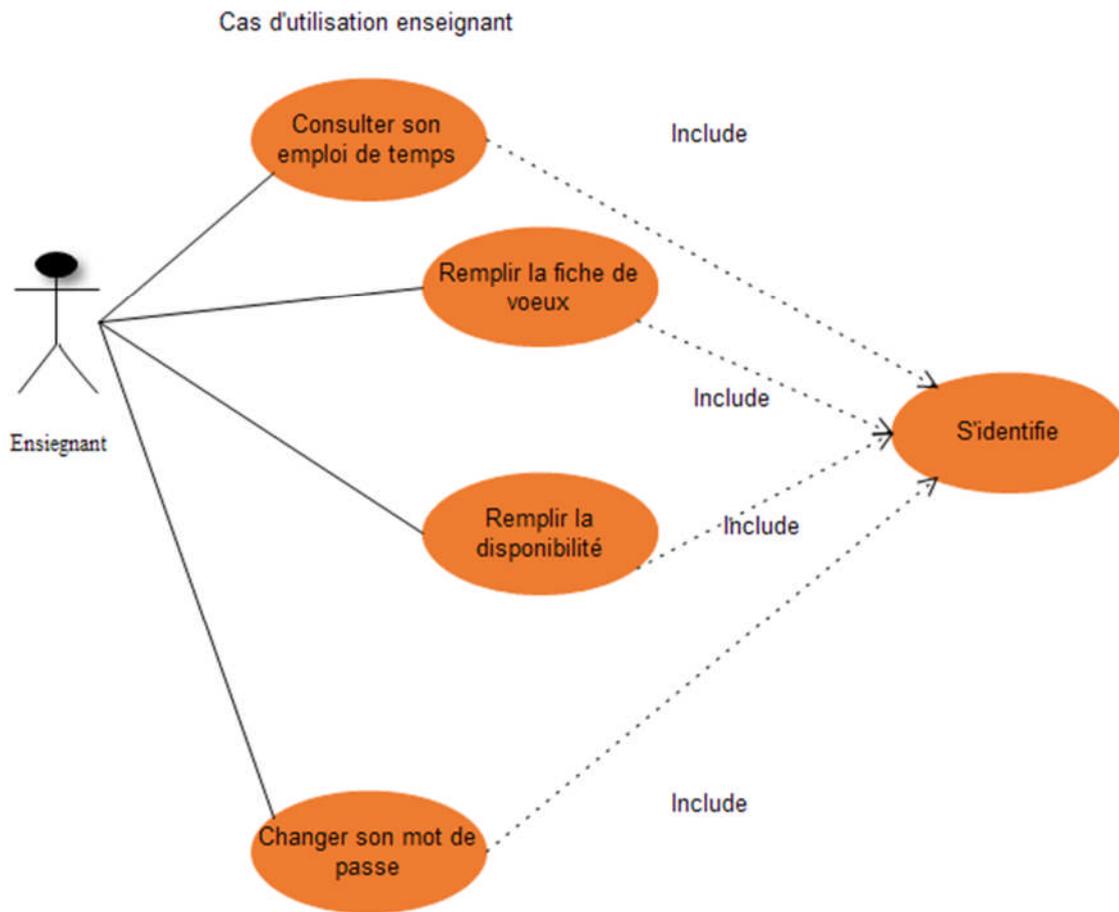


Figure II.2 : Le Diagramme de cas d'utilisation pour l'acteur « Enseignant ».

I.5.2 Le Diagramme de cas d'utilisation pour l'acteur « Planificateur »

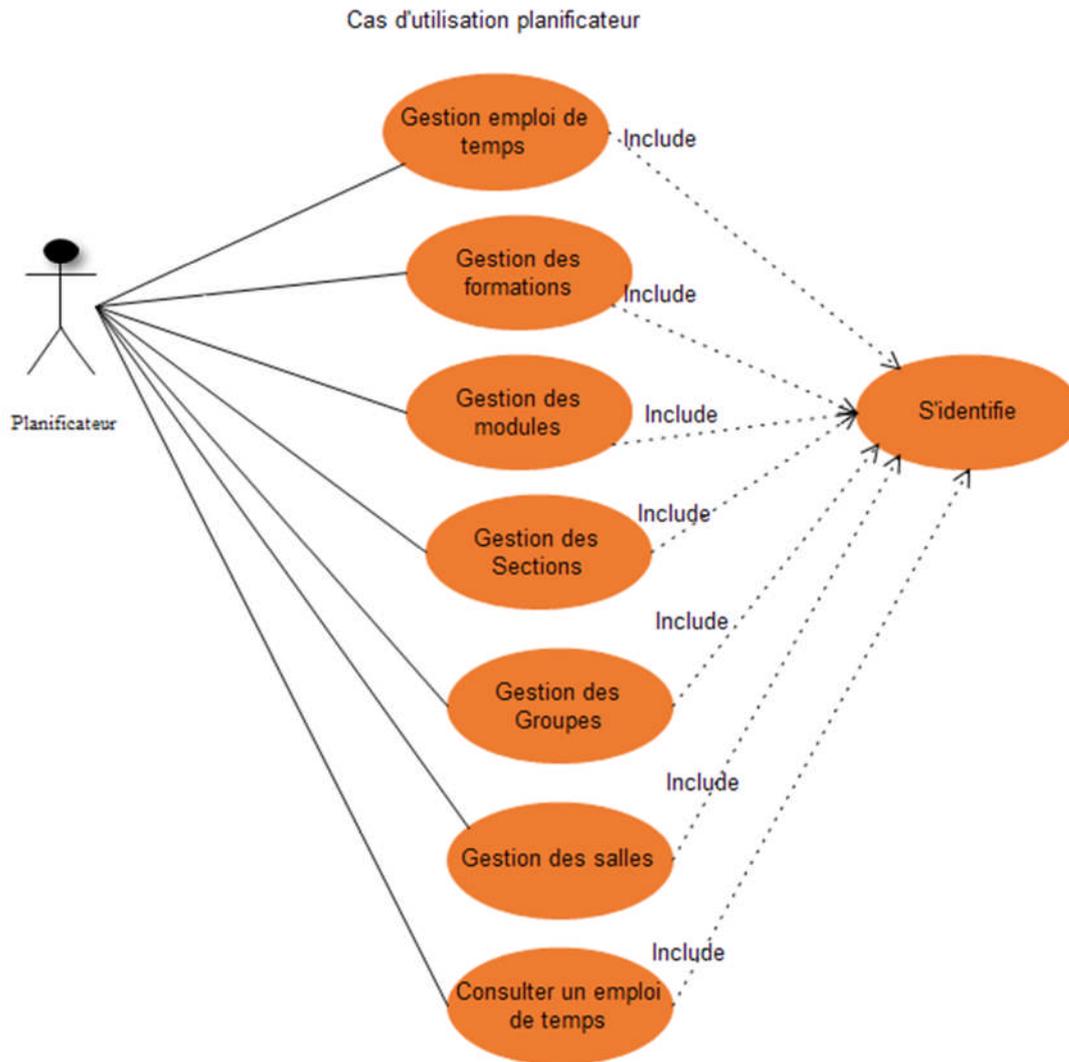


Figure II.3 : Le Diagramme de cas d'utilisation pour l'acteur « Planificateur ».

I.5.3 Le Diagramme de cas d'utilisation pour l'acteur « Administrateur »

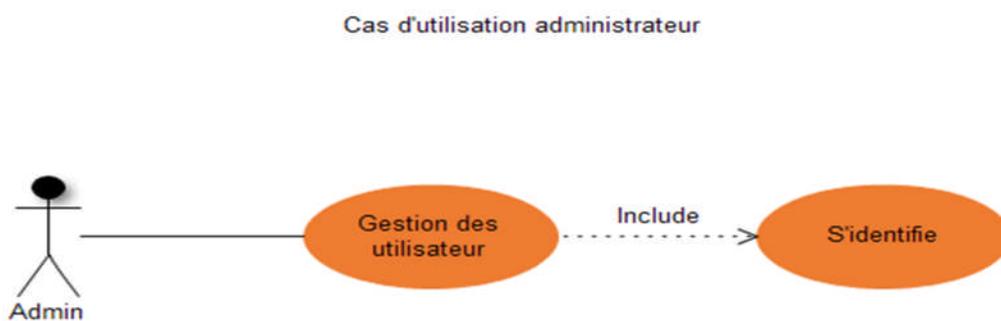


Figure II.4 : Le Diagramme de cas d'utilisation pour l'acteur « Administrateur »

I.6. Description textuelle des cas d'utilisation**I.6.1 Description de cas d'utilisation « Remplir la fiche de disponibilité »**

Description de cas « Remplir la fiche de disponibilité »
<p><u>Identification :</u></p> <p>Nom de cas : Remplir la fiche de disponibilité.</p> <p>But : Décrire les étapes permettant à l'enseignant de remplir la fiche de disponibilité.</p> <p>Acteur principal : Enseignant.</p>
<p><u>Pré condition :</u></p> <p>L'enseignant possède un compte (s'identifier et saisie son mot de passe).</p>
<p><u>Enchaînement nominal :</u></p> <ol style="list-style-type: none">1. L'enseignant attient son espace ;2. Le système affiche « Espace enseignant » ;3. L'enseignant sélection « Disponibilité » ;4. Le système retourne le formulaire « Disponibilité » ;5. L'enseignant remplit le formulaire et clique sur le bouton « Valider » ;6. Le système affiche la page de confirmation après avoir valider les données saisies.
<p><u>Post condition :</u></p> <p>L'enseignant remplit sa fiche disponibilité.</p>

Tableau II.2 : Description textuelle de cas d'utilisation « Remplir la fiche de disponibilité »

I.6.2. Description de cas d'utilisation « Ajouter un module »

<p>Description de cas « Ajouter un module »</p> <hr/> <p><u>Identification :</u></p> <p>Nom de cas : Ajouter un module.</p> <p>But : Décrire les étapes permettant au Planificateur d'ajouter un nouveau module.</p> <p>Acteur principal : Planificateur.</p> <hr/> <p><u>Pré condition :</u></p> <p>Le Planificateur possède un compte (s'identifier et saisie son mot de passe).</p> <p><u>Enchaînement nominal :</u></p> <ol style="list-style-type: none"> 1. Le Planificateur attend son espace ; 2. Le système affiche « Espace Planificateur » ; 3. Le Planificateur sélection « Gestion des module » ; 4. Le système affiche la page « Gestion des module » ; 5. Le planificateur sélection «Ajouter un module » ; 6. Le système affiche le formulaire d'ajout ; 7. Le Planificateur remplit le formulaire en suit clique sur le bouton de validation ; 8. Le système vérifie les données saisies ; 9. Le système affiche la page de confirmation après la validation des données saisies ; <p><u>Enchaînement alternative :</u></p> <p>A1 : Le module existe déjà ;</p> <p>9. Le système indique que le module existe déjà ;</p> <p>10. Le système affiche de nouveau le formulaire d'ajout ;</p> <p>L'enchaînement continue à partir de point N°7 de la séquence nominale.</p> <p><u>Post condition :</u></p> <p>Un nouveau module a été créé</p>

Tableau II.3 : Description textuelle de cas d'utilisation « Ajouter un module »

<p>Description de cas « Créer un emploi de temps»</p> <hr/> <p><u>Identification :</u></p> <p>Nom de cas : Créer un emploi de temps</p> <p>But : Décrire les étapes permettant au planificateur de Créer un emploi de temps</p> <p>Acteur principal : Planificateur</p>
--

Pré condition :

Le planificateur possède un compte (s'identifier et saisie le mot de passe).

Enchaînement nominal :

1. Le planificateur attend son espace ;
2. Le système affiche « Espace planificateur » ;
3. Le planificateur clique sur le lien « Gestion des emplois de temps » ;
4. Le système affiche la page « Gestion des emplois de temps » ;
5. Le planificateur remplit les champs et clique sur le bouton « valider » ;
6. Le système lui affiche un autre formulaire pour créer les séances
7. Le planificateur remplit les champs et les valide
8. Le système vérifie les données saisies et affiche le message de confirmation
9. Le planificateur ferme le formulaire et le planning créer s'affiche

Post condition :

Un nouveau planning a été créé.

Tableau II.4 : Description textuelle de cas d'utilisation « Créer un emploi de temps »

II. Conception

II.1. Les diagrammes de séquence

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

Vu le nombre d'importance de cas d'utilisation qu'on a recensé, nous allons décrire que cinq exemple de cas d'utilisation

II.1.1 Diagramme de séquence pour le cas d'utilisation « Authentification »

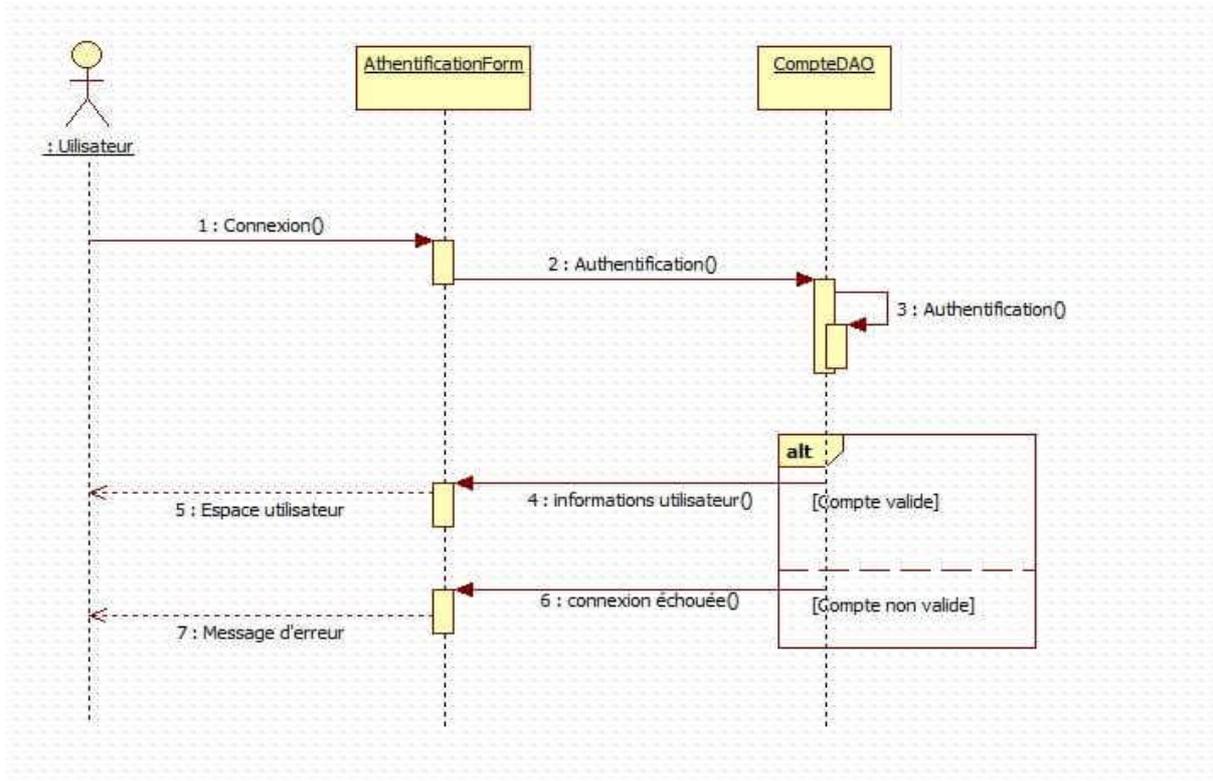


Figure II.5 : Diagramme de séquence pour la réalisation de cas d'utilisation « Authentification »

II.1.2 Diagramme de séquence pour le cas d'utilisation «créer une formation »

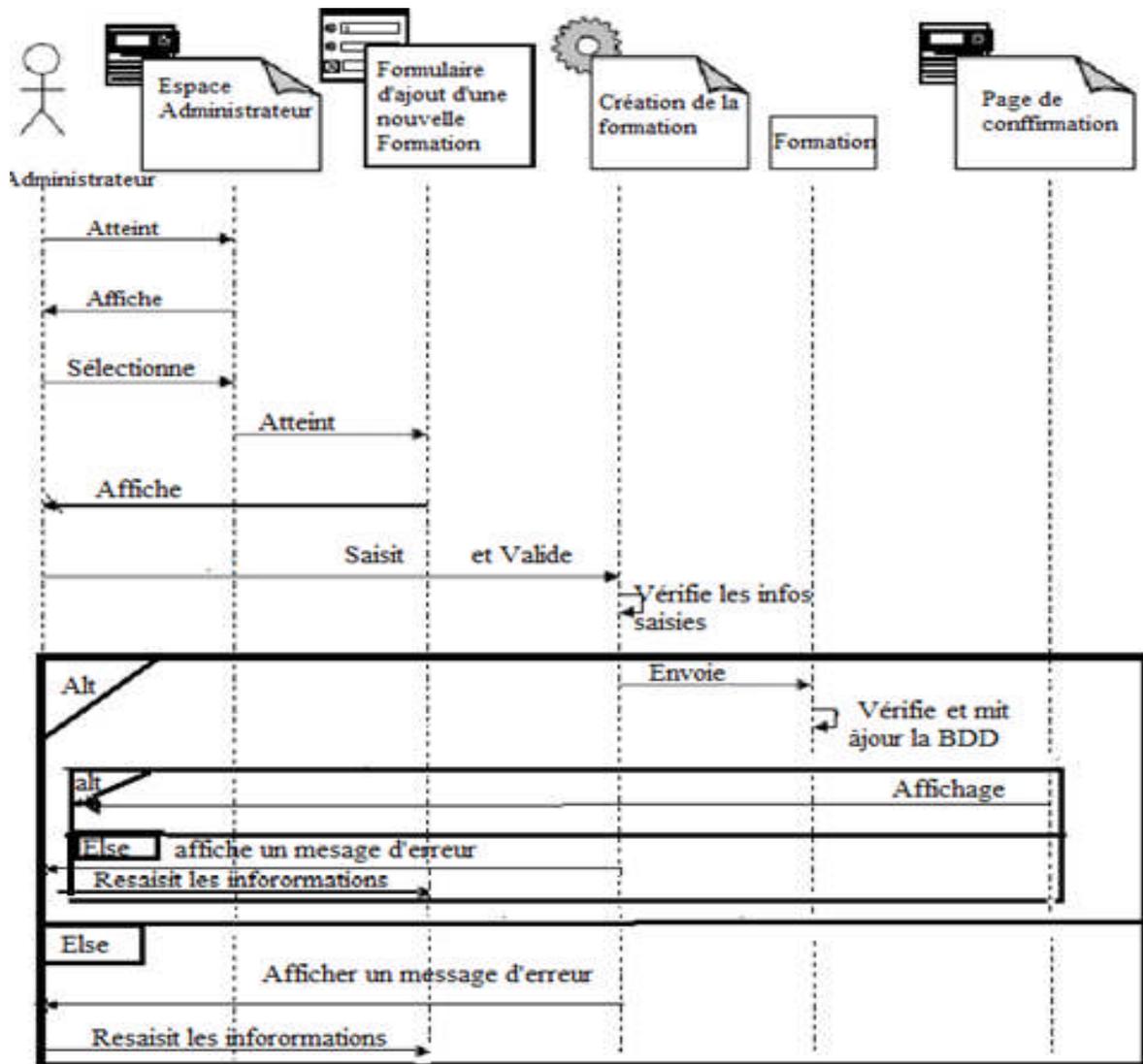


Figure II.6 : Diagramme de séquence pour la réalisation du cas d'utilisation «créer une formation »

Description

1. L'administrateur accède à son espace après avoir authentifié (saisit son identificateur et son mot de passe) sur la page d'accueil du système ;
2. L'administrateur clique sur le bouton « Gestion des formations » ;
3. Sur la page «Gestion des formations », l'administrateur clique sur le bouton « Ajouter une formation », le système lui affiche la page contenant le formulaire d'ajout ;
4. L'administrateur remplit le formulaire, en suit envoie ce dernier ;
5. Le système vérifie les informations saisies par rapport au type et au nombre des caractères spécifiés pour chaque information saisie ;
6. Si le résultat de la vérification est positif, le system envoie les informations saisies à la base de données sinon le système réaffiche le formulaire contenant les messages d'erreurs ;

7. Le système vérifie les informations envoyées, si ces informations n'existent pas dans la base de données ;
8. Si le résultat de la vérification est positif, le system mit à jour la table « Formation » et affiche à l'administrateur un message de confirmation de la création d'une nouvelle formation. Sinon le système affiche un message d'erreur.

II.1.3 Diagramme de séquence pour le cas d'utilisation « Remplir une fiche de vœux »

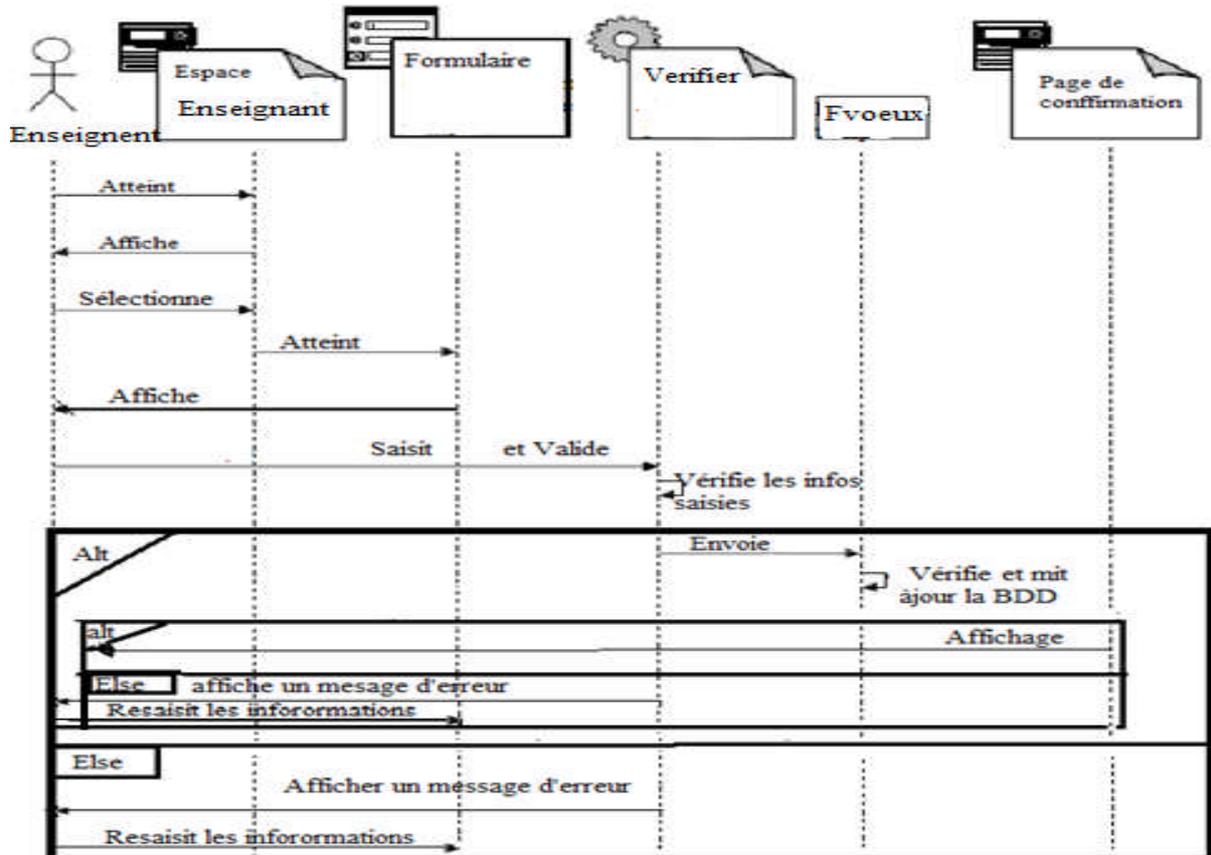


Figure II.7 : Diagramme de séquence pour la réalisation du cas d'utilisation « Remplir une fiche de vœux ».

Description :

1. L'enseignant accède à son espace après avoir authentifié (saisit son identificateur et son mot de passe) sur la page d'accueil du système ;
2. L'enseignant clique sur le bouton « Fiche de vœux » ;
3. Le système lui affiche la page contenant le formulaire d'ajout ;
4. L'enseignant remplit le formulaire, en suit envoie ce dernier ;
5. Le système vérifie les informations saisies par rapport au type et au nombre des caractères spécifiés pour chaque information saisie ;
6. Si le résultat de la vérification est positif, le system envoie les informations saisies à la base de données. Sinon le système réaffiche le formulaire contenant les messages d'erreurs ;

7. Le système vérifie les informations envoyées, si ces informations n'existent pas dans la base de données ;
8. Si le résultat de la vérification est positif, le system mit à jour la table « vœux » et affiche à l'administrateur un message de confirmation de la création d'une nouvelle fiche de vœux.

II.2. Diagramme d'activité

II.2.1 Diagramme d'activité « ajouter une salle »

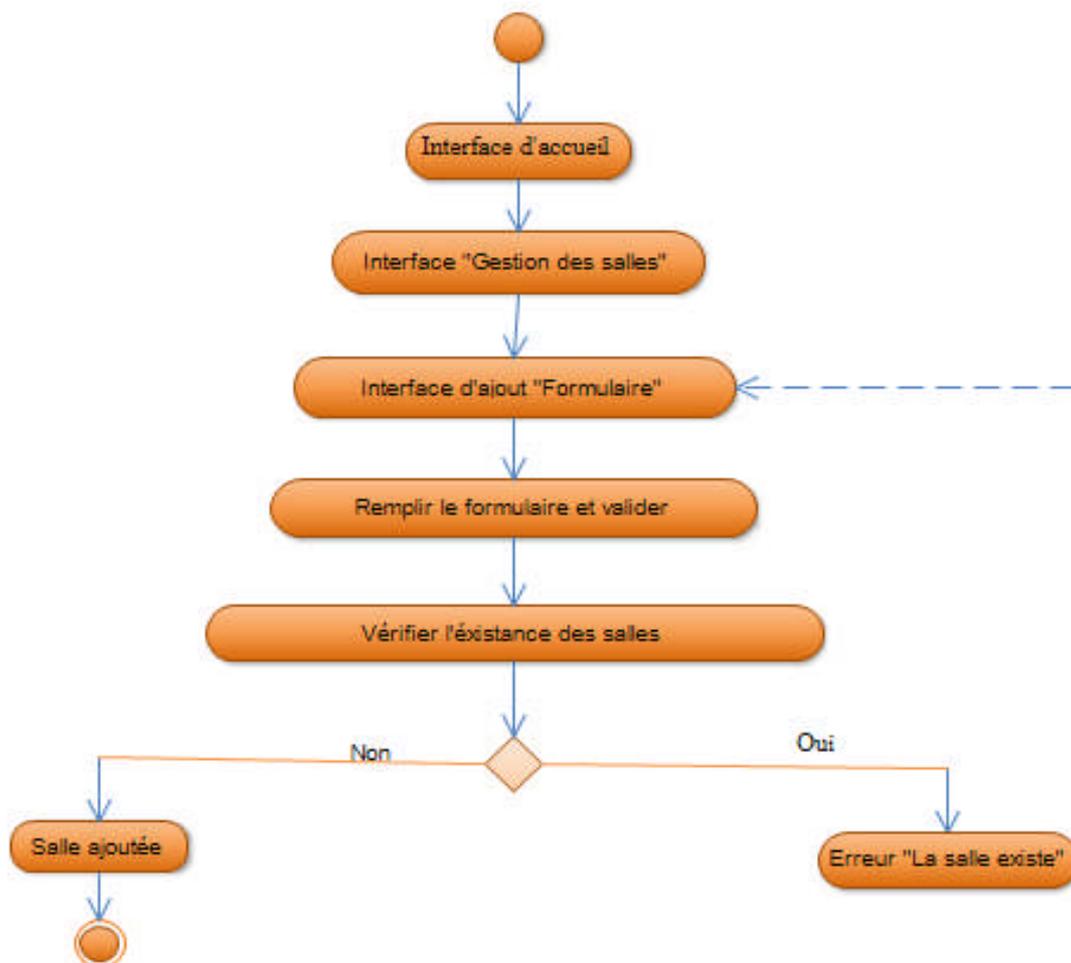


Figure II.8 : Diagramme d'activité « ajouter une salle ».

II.2.2 Diagramme d'activité « créer un emploi de temps »

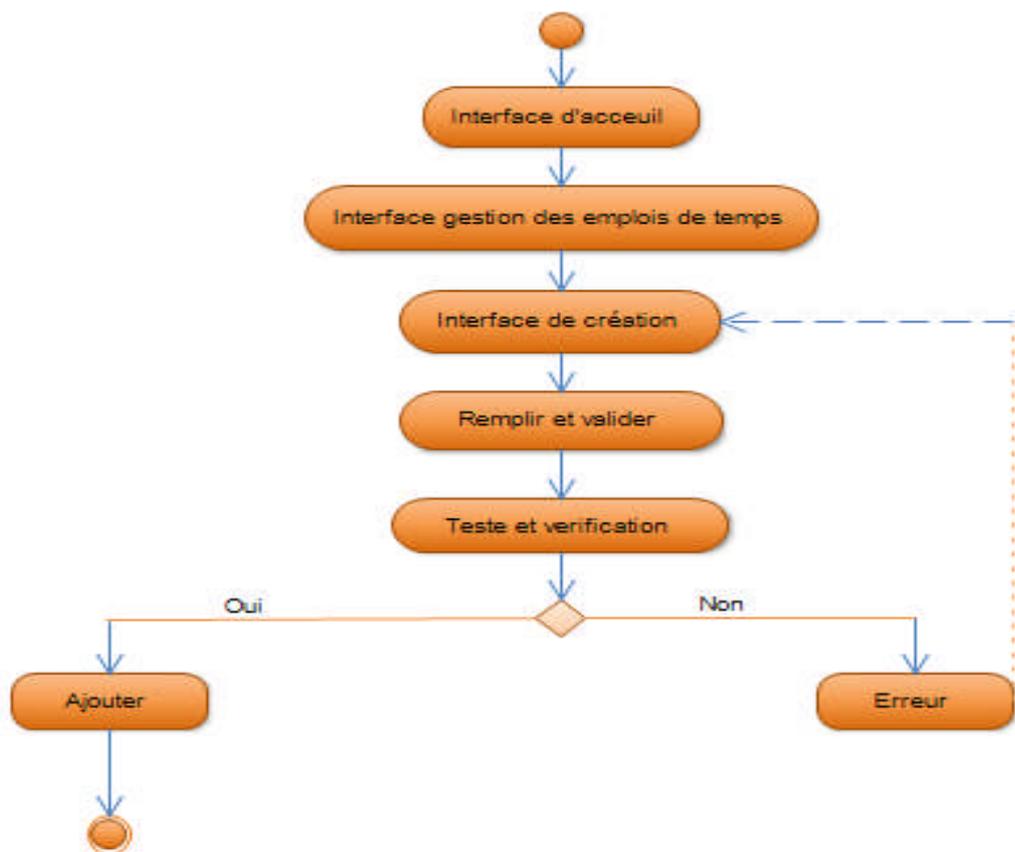


Figure II.9 : Diagramme d'activité pour le cas d'utilisation « créer un emploi de temps ».

II.2.3 Diagramme d'activité « Authentification »

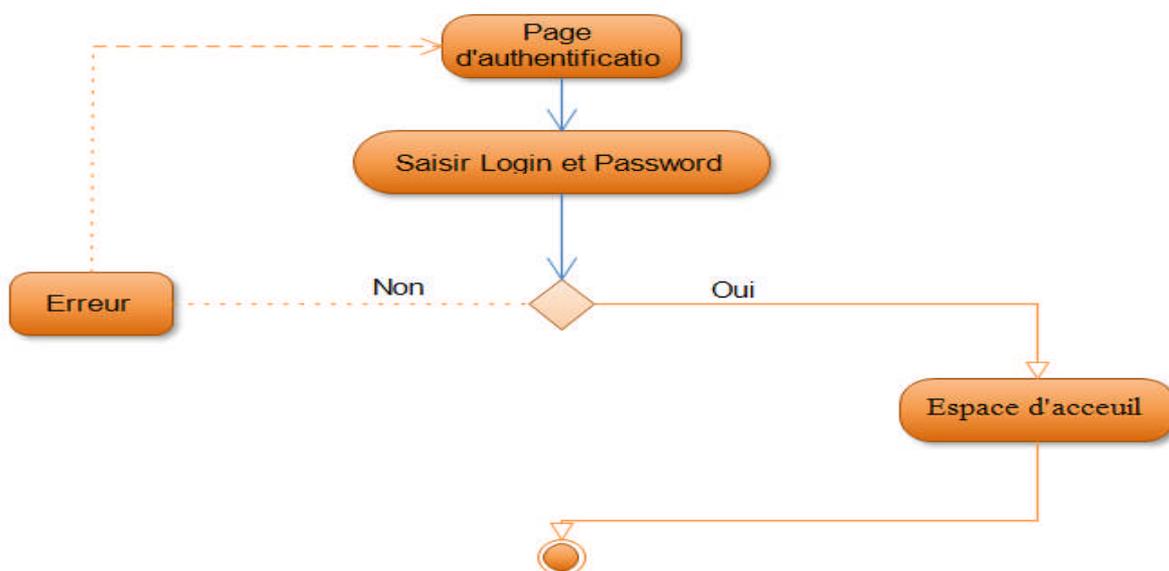


Figure II.10 : Diagramme d'activité pour le cas d'utilisation « Authentification ».

II.2. Diagramme de classe

Le diagramme de classe est le plus important dans la modélisation orientée objet. Il représente un ensemble de classe, d'interface et de collaboration ainsi que leur relation, il a pour objet de décrire la structure des entités manipulées par des utilisateurs.

Dans notre cas on va représenter quelque cas d'utilisation :

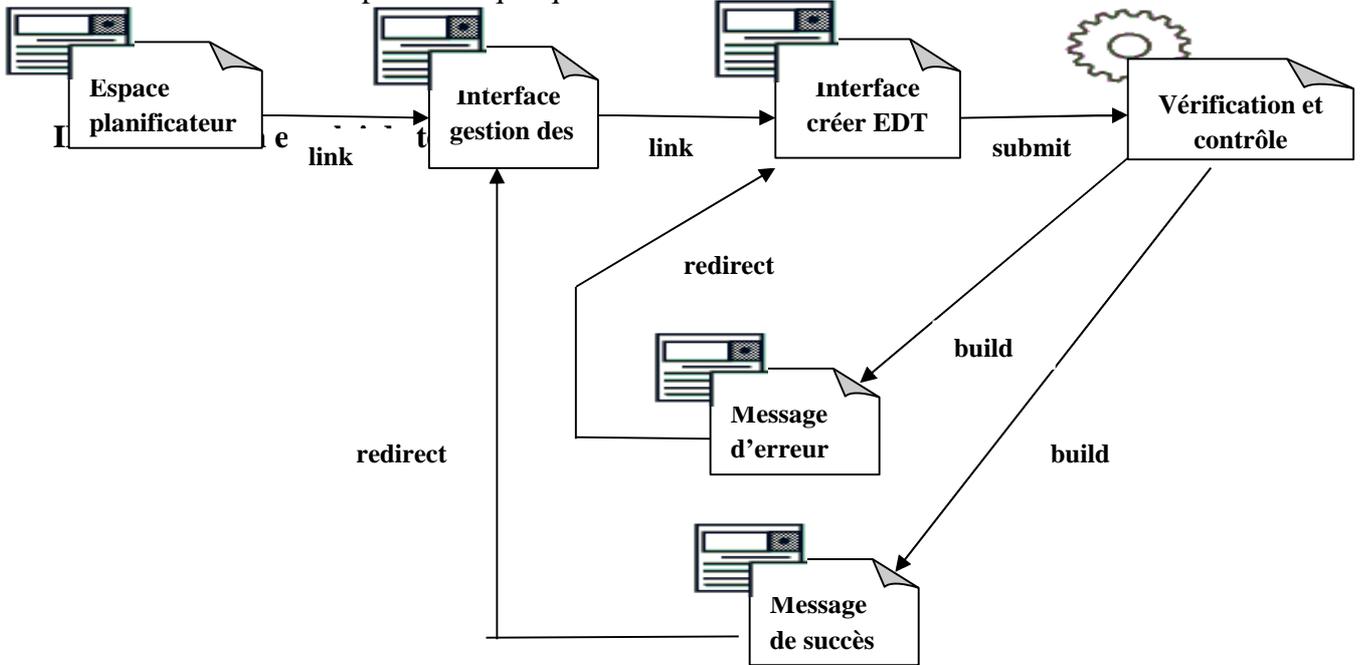


Figure II.11 : Créer un emploi de temps

II.2.2 Ajouter une formation

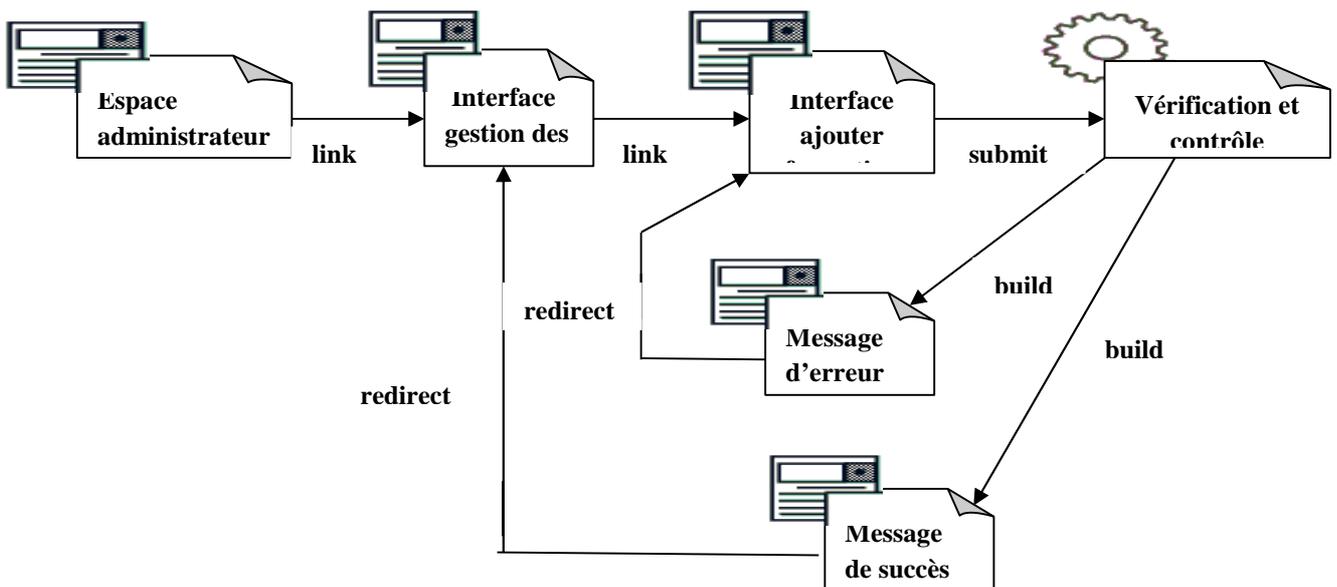


Figure II.12 : Ajouter une formation

II.2.3. Remplir la fiche de vœux

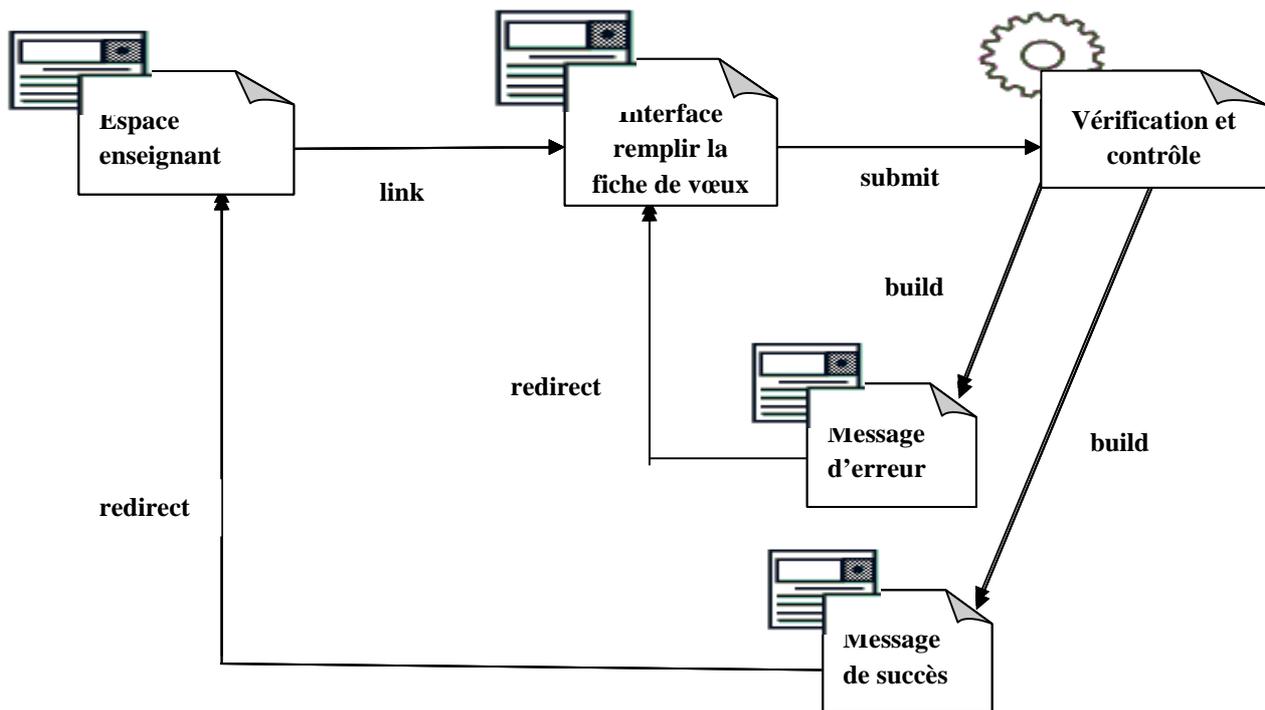


Figure II.13 : Remplir la fiche de vœux

III. Conception de la BDD

Dans ce niveau, le travail consiste en premier lieu à définir les règles de gestion, puis le modèle conceptuel de manière à concevoir la structure de la base de données, en se basant sur modèle conceptuel de données.

III.1. Les règles de gestion

Pour réaliser notre application nous avons un ensemble de problème à résoudre, les données de ce problème sont constituées par :

- un ensemble de créneaux horaires étalés sur une semaine de cinq jours, du dimanche au jeudi avec un nombre de périodes, et chaque période a une durée de d'une heure trente minutes.
- 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 et labos).

L'affectation des modules, enseignants, locaux à des périodes est soumise à des contraintes qui diffèrent selon leurs priorités (l'intérêt que recouvre la satisfaction d'une contrainte). Une contrainte ne revêt pas nécessairement un aspect absolu (soit elle est vérifiée ou violée) mais peut être formulée sous forme d'un objectif qui doit être approché autant que possible, selon ce critère, les contraintes peuvent être réparties en deux grandes classes : les contraintes dures (absolues) et les contraintes de préférences.

III.1.1. Les contraintes dures

Ce type de contraintes doit être obligatoirement satisfait dans toutes les situations car la violation de l'une de ces contraintes rend l'emploi du temps inefficace dans la réalité. On distingue dans notre cas cinq contraintes dures :

- Un enseignant ne peut pas être affecté à deux séances différentes à la même période.
- Une salle ne peut pas accueillir deux séances différentes à la même période.
- Chaque enseignant doit enseigner un module qui entre dans ses compétences.
- Un module doit respecter le nombre de séances hebdomadaires de ce dernier c'est à dire si un module est enseigné trois fois par semaine, alors il doit apparaître 3 fois dans le créneau horaire de la promotion.
- Un emploi du temps doit comporter tous les modules d'une promotion.
- La charge journalière d'un enseignant ne doit pas être dépassée.

III.1.2. Les contraintes de préférence

Contrairement au type de contraintes précédent, les contraintes de préférences n'exigent pas la vérification stricte, mais d'approcher au maximum de l'objectif voulu. Dans notre cas, on distingue :

- Essayer d'éviter aux (enseignant ou étudiants) des pertes de temps par de trop longs espacements entre deux séances d'une même journée (pas de trous).
- Eviter que certains jours se trouvent surcharger alors que d'autres le sont moins.
- Eviter d'affecter une période jugée non convenable à un enseignant, sauf si cela est inévitable
- Les modules de coefficient minimal ne doivent pas occuper les séances de la matinée d'une journée donnée, au détriment des modules de coefficients élevés.

- Minimiser les déplacements des étudiants dans l'établissement.
- Libérer quelques après-midi pour les enseignants

III.2. Diagramme de classe général

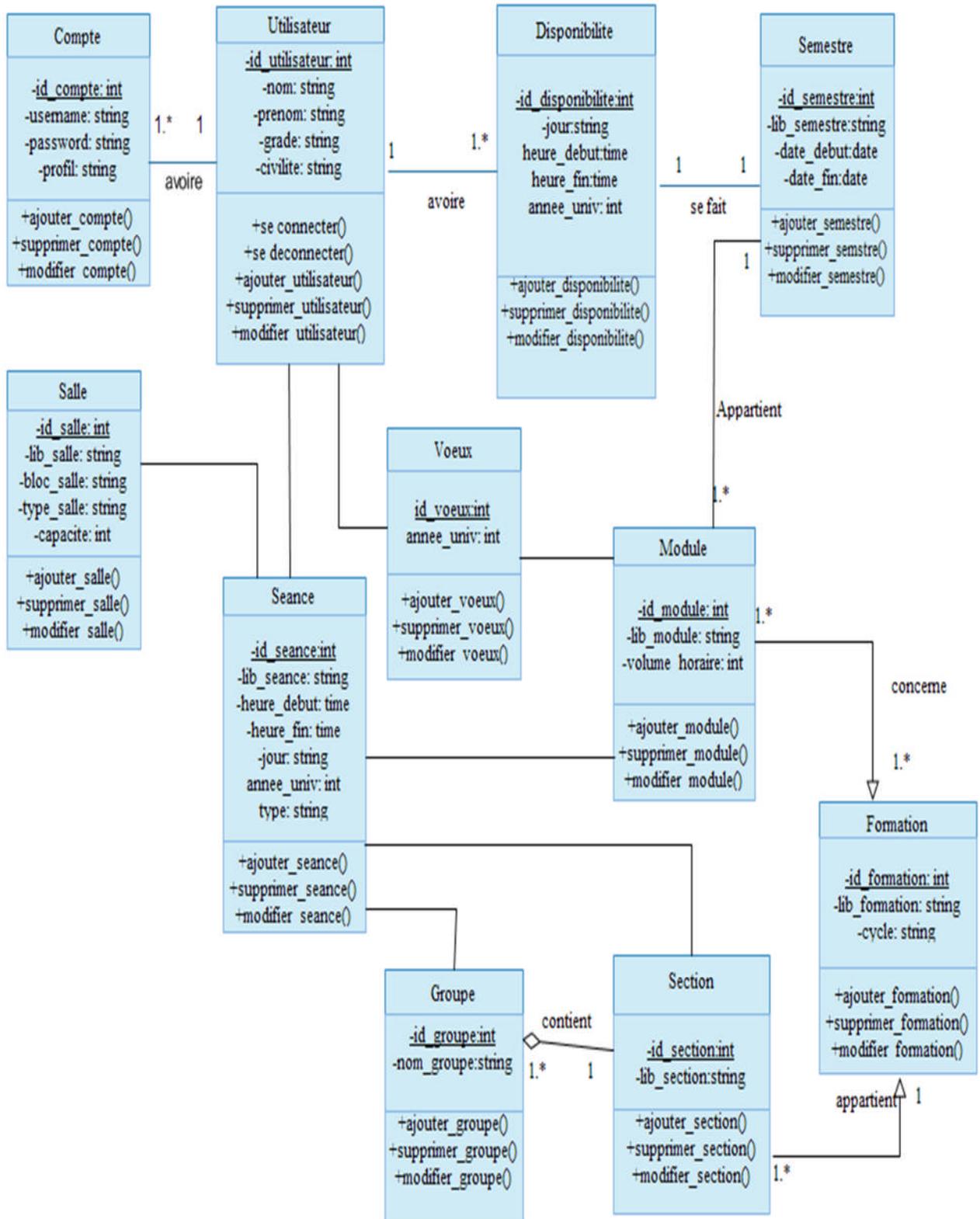


Figure II.14 : Diagramme de classe générale.

III.3. Les règles de passage du MLD au MPD

Le passage du modèle logique de données (MLD) vers le modèle physique de données (MPD) exige que les tables qui jusque-là sont externe à la base de données se traduisent en fichiers faisant partie intégrante de la base de données. Ainsi

- Les tables décrites au niveau du schéma logique deviennent des fichiers de données appelées « tables » ;
- Les attributs deviennent des champs de tables ;
- Les identifiants deviennent des clés primaires ;
- Les clés héritées deviennent des clés secondaires.

III.4. Le modèle relationnel de données

Après l'élaboration des diagrammes des cas d'utilisation, des diagrammes de séquences puis le diagramme de classes, ce dernier nous a servi de référence pour l'implémentation de la partie base de données, et cela parce qu'il met en évidence toutes les classes entités et leurs attributs. A cet effet, chaque classe sera implémentée comme une table dont les colonnes ne seront rien d'autre que les attributs de la classe.

utilisateur (id_utilisateur, nom, prenom, grade) ;

compte (id_compte, id_utilisateur*, username, password, profil) ;

formation (id_formation, lib_formation, cycle) ;

groupe (id_groupe, id_section*, nom_groupe) ;

section (id_section, id_groupe*, lib_section) ;

module (id_module, id_formation*, id_semestre*, lib_module, volum_horaire) ;

salle (id_salle, lib_salle, bloc_salle, type_salle, capacité) ;

seance (id_seance, id_utilisateur*, id_salle*, id_module*, id_formation*, id_semestre*, type, id_section*, id_groupe*, lib_seance, jour, plage_horaire, annee_universitaire) ;

vœux (id_veoux, id_utilisateur*, id_module*, annee_universitaire) ;

Disponibilite(id_disponibilite, id_utilisataur*, id_semestre*, jour, plage_horaire, annee_universitaire, disponible) ;

semestre (id_semestre, lib_semestre, date_debut_semestre, date_fin_semestre).

Conclusion

Dans ce chapitre nous avons présenté les différents processus nécessaires à la réalisation de notre application et cela, en passant par deux phases ; La première a été analytique, du point de vue utilisateur et les opérations qu'il est apte à réaliser ainsi que les besoins d'une telle application. La seconde nous a permis détaillé le processus de conception objet, de manière progressive, à travers des diagrammes UML des cas d'utilisation, de séquences et de classes. Ceci nous a permis donc, d'aboutir de façon efficace à la dernière étape qui est l'implémentation. La mise en œuvre de celle-ci est explicitée dans le chapitre suivant.

Chapitre III

Réalisation

Introduction

Nous nous intéressons dans ce chapitre à l'implémentation de notre application, et cela en décrivant d'abord l'environnement et les outils de développement que nous avons utilisé, ensuite en présentant un exemple des différentes interfaces de cette dernière ainsi que les fonctionnalités qu'elles regroupent.

I. La présentation de « Java » :

I.1. Java : Une technologie Java : [site, 00]

Java est une technologie développée par Sun Microsystems dans les années **1990**. Elle correspond à plusieurs produits et spécifications de logiciels qui, ensemble, constituent un système pour développer et déployer des applications autonomes et portables qui s'exécutent indépendamment du système d'exploitation utilisé.

Depuis des années, Sun Microsystems appelle Java la « technologie Java » dans son ensemble. En pratique et par abus de langage, beaucoup de programmeurs utilisent le mot « Java » pour désigner le langage de programmation, tandis que la plate-forme d'exécution est appelée « JRE » (Java Runtime Environment, environnement d'exécution Java), le système de compilation : « JDK » (Java Development Kit, kit de développement Java) plutôt que « compilateur Java » et un EDI ou IDE (Integrated Development Environment, environnement de développement intégré) qui utilise le JDK.

I.2. Java : Un langage de programmation : [site, 01]

Le langage Java est un langage de programmation informatique orienté objet présenté officiellement le **23 mai 1995**. C'est un langage puissant et performant, tirant expérience des autres langages apparus avant lui en particuliers le C++. En effet, java est basé sur sa syntaxe mais avec des fonctionnalités qui en rendent la programmation plus simple et plus sûre comme la gestion automatique de la mémoire grâce au mécanisme du ramasse-miettes (Garbage Collector). Lors de la création du langage Java, il avait été décidé que ce langage devait répondre à 5 objectifs :

- Simple, orienté objet et familier ;

- Robuste et sûr ;
- Indépendant de la machine employée pour l'exécution ;
- Très performant ;
- Multi tâches et dynamique.

L'indépendance vis-à-vis de la plate-forme, une de ses grandes forces, signifie que les programmes écrits en Java fonctionnent de manière parfaitement similaire sur différentes architectures matérielles. On peut effectuer le développement sur une architecture donnée et faire tourner l'application sur toutes les autres.

I.3. Java : Une plateforme : [site, 01]

La plate-forme Java (the Java Platform en anglais) est une plate-forme produite permettant de développer et d'exécuter des programmes écrits en langage Java indépendante de tout processeur et de tout système d'exploitation. Elle est constituée de plusieurs programmes, chacun d'entre eux apportant une fonctionnalité de l'ensemble des capacités de la plate-forme.

Par exemple, l'un d'entre eux qui appelé le compilateur Java et qui permet de convertir les codes source Java en bytecode Java (un langage intermédiaire destinée à la JVM (Java Virtual Machine ou machine virtuelle Java)) est fourni par le JDK. Un autre élément, appelé le JRE (Java Runtime Environment) implémente habituellement la JVM grâce à un compilateur JIT (Just In Time) qui convertit à la volée le bytecode intermédiaire en un code natif pour la machine.

Pour éviter les problèmes qui se poseraient avec une configuration universelle, qui obligerait à tout installer sur tous les appareils.

La plate-forme Java 2 a été divisée en trois grandes éditions : L'édition J2EE (Java 2 Enterprise Edition) est conçue pour déployer des solutions serveur de haut niveau ; l'édition J2SE (Java 2 Standard Edition) est utilisée pour les ordinateurs de bureau ; l'édition J2ME (Java 2 Micro Edition) est spécialement prévue pour les petits appareils électroniques d'usage courant tels que les téléphones cellulaires.

Grâce à cette approche, la fonctionnalité appropriée est disponible pour les différents types d'appareil.

II. Le JDK (Java Development Kit)

Le Kit de développement Java, communément appelé JDK, est l'environnement dans lequel le code Java est compilé pour être transformé en bytecode afin que la machine virtuelle Java (JVM) puisse l'interpréter. En effet, comme dans tous les langages, on commence par créer un fichier source. Le fichier source d'un programme écrit en Java est un simple fichier texte dont l'extension est par convention «.java ». Lorsque le programme est prêt à être "essayé", il s'agit de le compiler (le traduire en langage machine) à l'aide d'un compilateur.

Toutefois, contrairement aux langages compilés traditionnels, pour lesquels le compilateur crée un fichier binaire directement exécutable par un processeur donné (c'est-à-dire un fichier binaire contenant des instructions spécifiques à un processeur), le code source Java est compilé en un langage intermédiaire (appelé pseudocode ou bytecode) dans un fichier portant le même nom que le fichier source à l'exception de son extension (.class). Cette caractéristique est majeure, car c'est elle qui fait qu'un programme écrit en Java est portable, c'est-à-dire qu'il ne dépend pas d'une plate-forme donnée.

En réalité le code intermédiaire n'est exécutable sur aucune plate-forme sans la présence d'une machine virtuelle, un interpréteur (la machine virtuelle est d'ailleurs parfois appelée interpréteur Java) tournant sur une plate-forme donnée, et capable d'interpréter le code intermédiaire.

III. Les interfaces graphiques du Java : [site, 00]

Une interface graphique ou GUI (Graphical User Interface) est un ensemble de commandes affichées à l'écran de l'ordinateur et permettant de piloter un logiciel sans saisie de lignes de commandes.

Les interfaces graphiques assurent le dialogue entre les utilisateurs et une application. Dans un premier temps, Java proposait l'API AWT (Abstract Window Toolkit) le tout premier proposé par Sun (développé en 30 jours). Implémentation via des composants natifs. Peu de composants disponibles pour créer des interfaces graphiques.

Depuis, Java propose une nouvelle API nommée Swing, librairie pure Java, construite au-dessus d'un sous-ensemble d'AWT. De nombreux composants disponibles (arbre, menus, sélecteur). Entièrement portable mais plutôt lent.

Ces deux API peuvent être utilisées pour développer des applications ou des applets. AWT et Swing sont étroitement liés. La majeure partie des classes AWT existe dans Swing (e.g. Button et JButton). La programmation Swing nécessite de temps en temps l'usage de classes AWT (en particulier pour les événements et les polices) privilégier, quand c'est possible, l'utilisation des classes Swing. Les classes s'obtiennent en important les paquetages suivants : (java.awt.* ;) et (javax.swing.* ;) .

Face aux problèmes de performance de Swing, IBM a créé sa propre bibliothèque nommée SWT (Standard Widget Toolkit) librairie nouvelle génération soutenue par le projet Eclipse. Implémentation native et nombreux composants. Très rapide. La vélocité de cette application favorise une utilisation grandissante de cette bibliothèque.

II. Outils de développement

Cette partie a pour objectif de décrire les différents outils dont on avait besoin pour la réalisation de notre application et que sans eux, le développement serait un peu ardu. Cependant, nous avons opté pour l'environnement de développement intégré (IDE) **NetBeans** version 8.0.3 qui est doté d'un JDBC (Java Database Connectivity) qui permet au langage Java d'accéder à des bases de données par l'intermédiaire du langage SQL. Comme cet outil est codé en Java, donc pour les utiliser il nous fallait l'installation d'un Java Card Development Kit (JDK-8u25-windows-i586) contenant un environnement d'exécution **JRE**

IV.1. Présentation de NetBeans IDE

NetBeans est un IDE Java. Il fut développé à l'origine par une équipe d'étudiants à Prague, racheté ensuite par Sun Microsystems. En 2002, Sun a décidé de rendre NetBeans opensource.

L'IDE NetBeans est donc un environnement de développement intégré permettant d'écrire, compiler, déboguer et déployer des programmes Java. En plus de Java, il supporte différents autres langages, comme Python, C, C++, XML, Ruby, PHP et HTML.

Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi langage, refactoring, éditeur graphique d'interfaces et de pages web). Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X et Open VMS.

Sa conception est complètement modulaire : Tout est module. Ce qui fait de NetBeans une boîte à outils facilement améliorable ou modifiable essentiellement grâce aux plug-ins

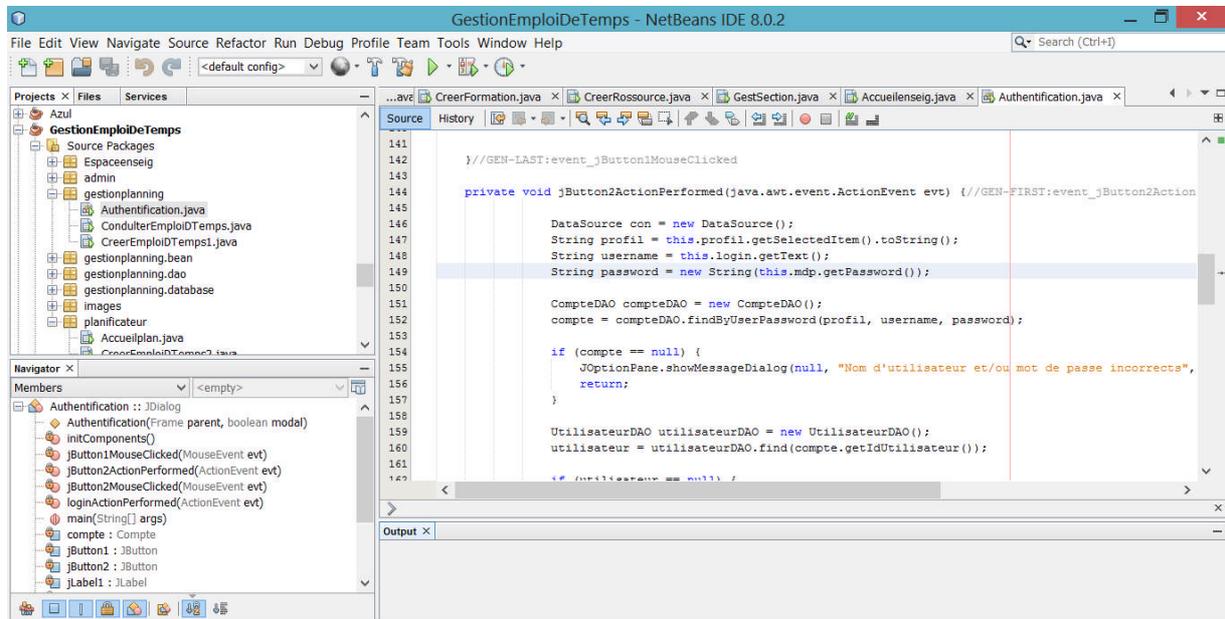


Figure III.1 Interface de l'IDE Netbeans

IV.2 Présentation de Wampserver : [site, 00]

Wampserver est un paquetage contenant à la fois deux serveurs (Apache et MySQL), un interpréteur de script (PHP), les deux bases SQL PhpMyAdmin et SQLiteManager pour gérer plus facilement les bases de données. Il permet d'installer automatiquement et facilement une plateforme permettant l'exploitation d'un site web en PHP qui éventuellement aurait besoin d'un accès à une base de données.

IV.2.1 PhpMyAdmin

PhpMyAdmin est un utilitaire rendant plus conviviale l'administration de base de données MySQL. Il consiste en un ensemble de scripts permettant d'administrer des bases de données MySQL en passant par un navigateur web.

❖ Les fonctions de PhpMyAdmin

- Création et suppression de la base de données ;

- Importer et exporter la base de données ;
- Création, modification, copie et suppression de table ;
- Edition, ajout et suppression de champ ;
- Exécution des requêtes SQL ;
- Création d'index ;
- Chargement de fichier dans des tables.

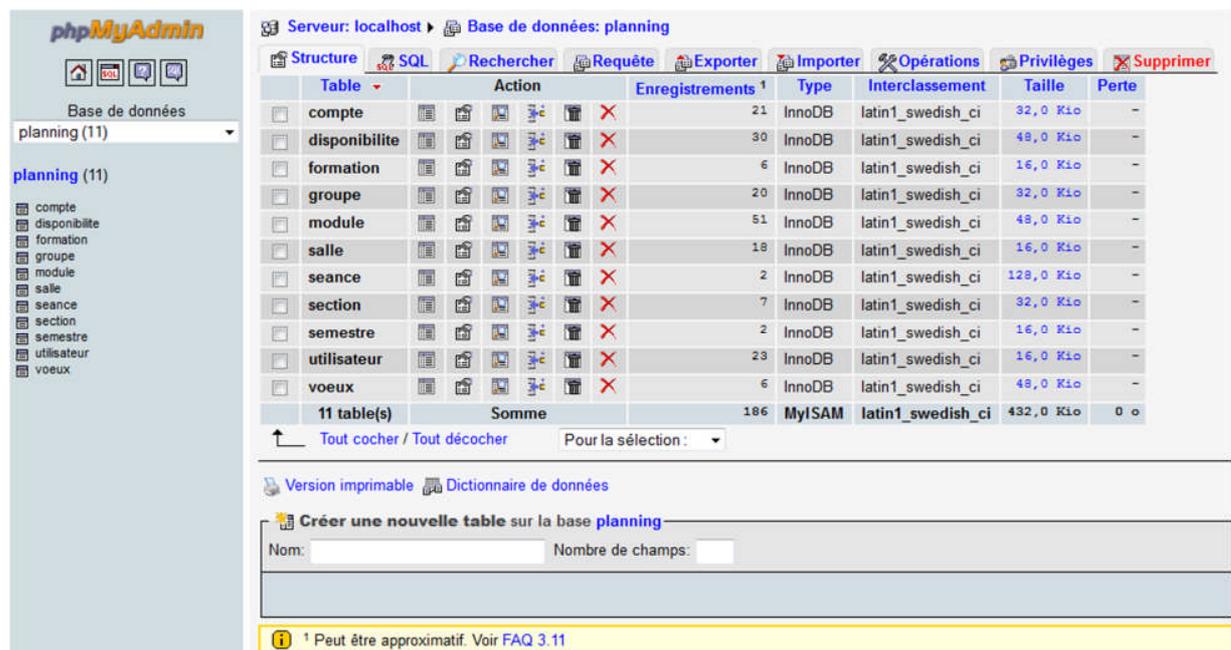


Figure III.2 Accès à la base de données « planning » à partir de PhpMyAdmin.

IV.3 Le JDBC

Pour assurer la compatibilité de JAVA avec diverses bases de données, les applications JAVA utilisent les mêmes instructions pour s'adresser au pilote JDBC qui est un ensemble de classes et d'interfaces qui prennent en charge les spécificités du serveur de base de données, ainsi, il permet à un programme java d'accéder via des requêtes SQL à un moteur de base de données et faciliter le changement d'éditeur de base de données.

Quand un programme java souhaite accéder à une base de données, il commence par demander le changement du pilote en mémoire ensuite l'utiliser dans une servlet ou JSP pour établir la connexion et effectuer les requêtes souhaitées.

V. Le modèle physique de données

Le Modèle Physique de Données (MPD) est une étape de définition des données à l'intérieur de la structure physique de l'ordinateur c'est-à-dire le résultat de la décision technique qui a été prise en fonction des objets et des contraintes techniques. Cette étape consiste à implémenter le modèle dans le SGBD (système gestion de base de données), c'est-à-dire le traduire dans un langage de définition de données. Le langage généralement utilisé pour ce type d'opération est le SQL.

V.1. Les objectifs du MPD

Le modèle physique de données (MPD) s'intéresse aux objectifs suivants :

- La gestion interne des données en fonction des traitements ;
- Définir la place nécessaire à chaque table ;
- Définir l'implémentation physique de la base de données ;
- Optimiser l'accès à l'information.

V.2. Structure des tables utilisées

- **Table « utilisateur »**

Nom de champ	Type	Taille
id-utilisateur	INT	11
nom	VARCHAR	30
prenom	VARCHAR	30
grade	VARCHAR	30
civilite	VARCHAR	3

- **Table « compte »**

Nom de champ	Type	Taille
id_compte	INT	20
username	VARCHAR	30
password	VARCHAR	30
profil	VARCHAR	30
id_utilisateur	INT	20

- **Table « formation »**

Nom de champ	Type	Taille
id_formation	INT	20
lib_formation	VARCHAR	30
cycle	VARCHAR	30

- **Table « section »**

Nom de champ	Type	Taille
id_section	INT	20
lib_section	VARCHAR	30
id_formation	INT	20

- **Table « groupe »**

Nom de champ	Type	Taille
id_groupe	INT	20
nom_groupe	VARCHAR	30
id_section	INT	20

- **Table « semestre »**

Nom de champ	Type	Taille
id_semestre	INT	20
lib_semestre	VARCHAR	30
date_debut_semestre	VARCHAR	10
date_fin_semestre	VARCHAR	10

- **Table « module »**

Nom de champ	Type	Taille
id_module	INT	20
lib_module	VARCHAR	30
id_formation	INT	20
Id_semestre	INT	20
volume_horaire	INT	20

- **Table « salle »**

Nom de champ	Type	Taille
id_salle	INT	20
lib_salle	VARCHAR	30
type_salle	VARCHAR	30
bloc_salle	VARCHAR	30
capacite	INT	20

- **Table « seance »**

Nom de champ	Type	Taille
id_seance	INT	20
id_module	INT	20
id_utilisateur	INT	20
id_salle	INT	20
id_section	INT	20
id_groupe	INT	20
lib_seance	VARCHAR	20
type	VARCHAR	20
plage_horaire	VARCHAR	20
id_semestre	INT	20

id_formation	INT	20
jour	VARCHAR	10
Annee_universitaire	VARCHAR	9

- **Table « Vœux »**

Nom de champ	Type	Taille
id_voeux	INT	20
id_utilisateur	INT	20
id_module	INT	20
Annee_universitaire	VARCHAR	9

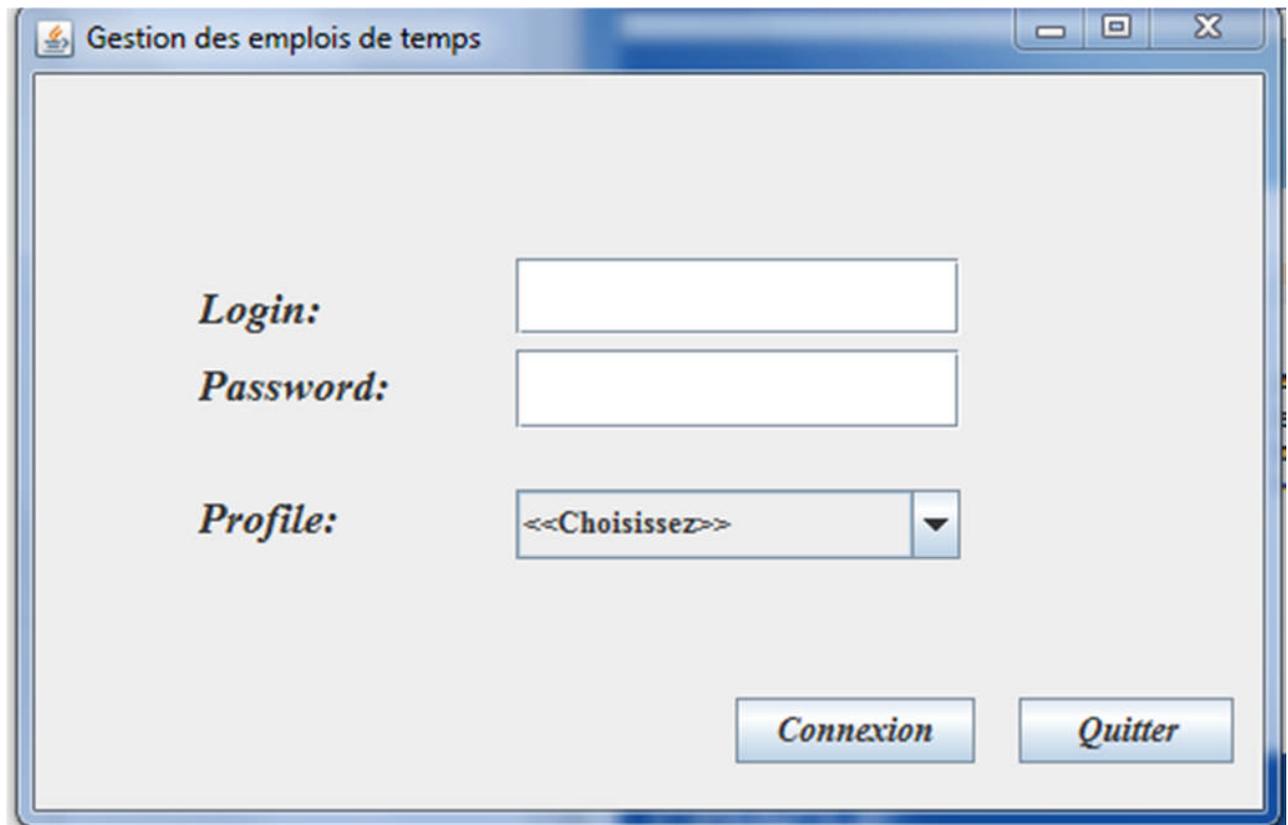
- **Table « disponibilite »**

Nom de champ	Type	Taille
id_disponibilite	INT	20
id_utilisateur	INT	20
id_semestre	INT	20
jour	VARCHAR	10
plage_horaire	VARCHAR	20
annee_universitaire	VARCHAR	9
disponible	VARCHAR	1

VI. Présentation de quelques interfaces de notre application

Les interfaces de notre système sont conçues de manière à être simples, naturelles, compréhensible et d'utilisation faciles.

VI.1. La fenêtre principale (Connexion)



The image shows a screenshot of a Windows-style application window titled "Gestion des emplois de temps". The window contains a login form with three input fields: "Login:", "Password:", and "Profile:". The "Profile:" field is a dropdown menu with the text "<<Choisissez>>" and a downward arrow. Below the input fields are two buttons: "Connexion" and "Quitter".

Figure III.3. Interface d'authentification.

VI.2.Espace administrateur

Espace administrateur est conçu pour gérer les utilisateurs (les enseignants, l'administrateur, le planificateur).

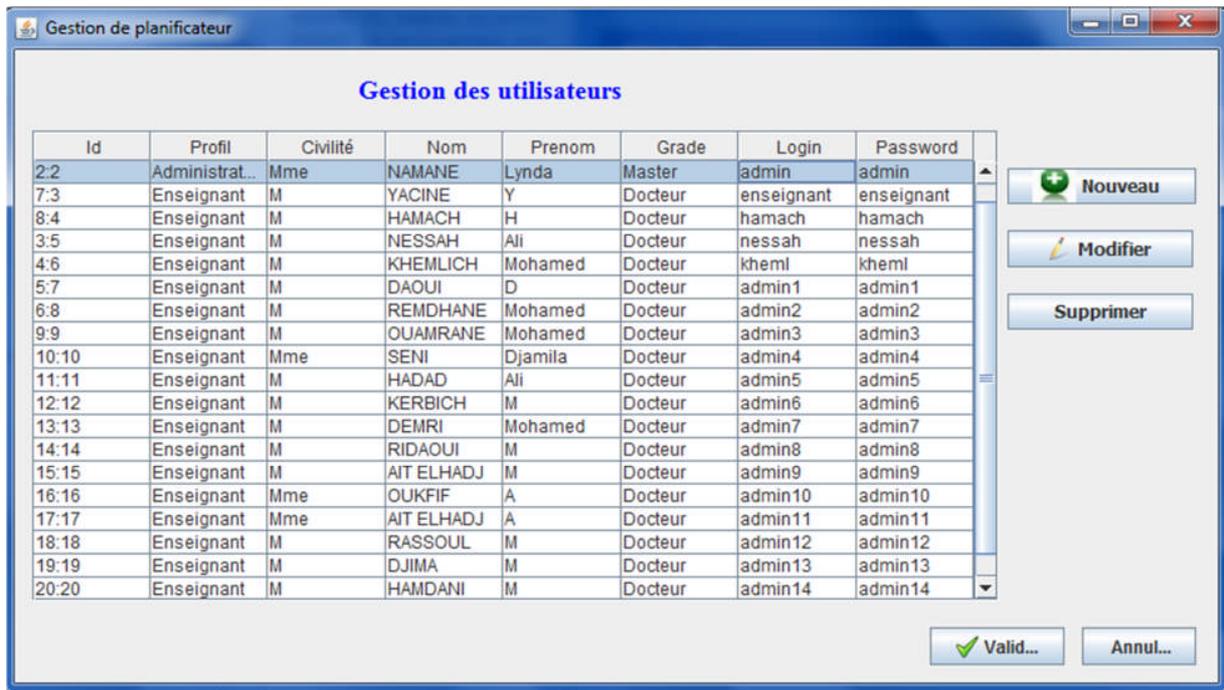


Figure III.4. Gestion des utilisateurs.

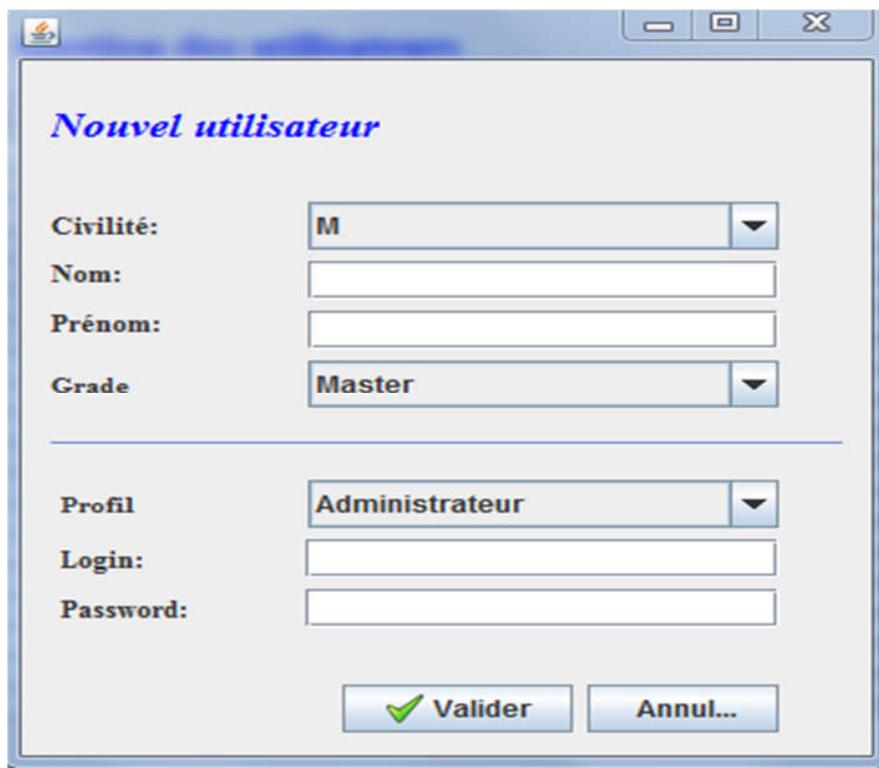


Figure III.5. Ajouter un utilisateur.

VI.3.Espace enseignant

Chaque enseignant peut gérer sa fiche de vœux, sa disponibilité et consulter son emploi de temps

Gestion des emplois de temps

Fiche de vœux de M YACINE Y

Formation:

Semestre:

Sélection	Id	Module	Volume horaire
<input type="checkbox"/>		1 Architecture des ordin...	30
<input checked="" type="checkbox"/>		2 Algorithmique et struc...	30
<input type="checkbox"/>		3 Système d information	30
<input checked="" type="checkbox"/>		4 Analyse numérique	30
<input type="checkbox"/>		5 Probabilités et statisti...	30
<input type="checkbox"/>		6 Logique mathématique	30

Figure III.6. Fiche de vœux d'un enseignant.

Gestion des emplois de temps

Disponibilité de M YACINE Y

Année universitaire 2015/2016

Semestre S1

Jour/Horaire	08:00-09:30	09:30-11:00	11:00-12:30	12:30-14:00	14:00-15:30	15:30-17:00
Dimanche	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Lundi	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mardi	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mercredi	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Jeudi	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Tout sélectionner... Tout désélectionner

Appliquer Fermer

Figure III.7. La disponibilité.

VI.4. Espace planificateur

Espace pour planificateur qui génère un emploi de temps

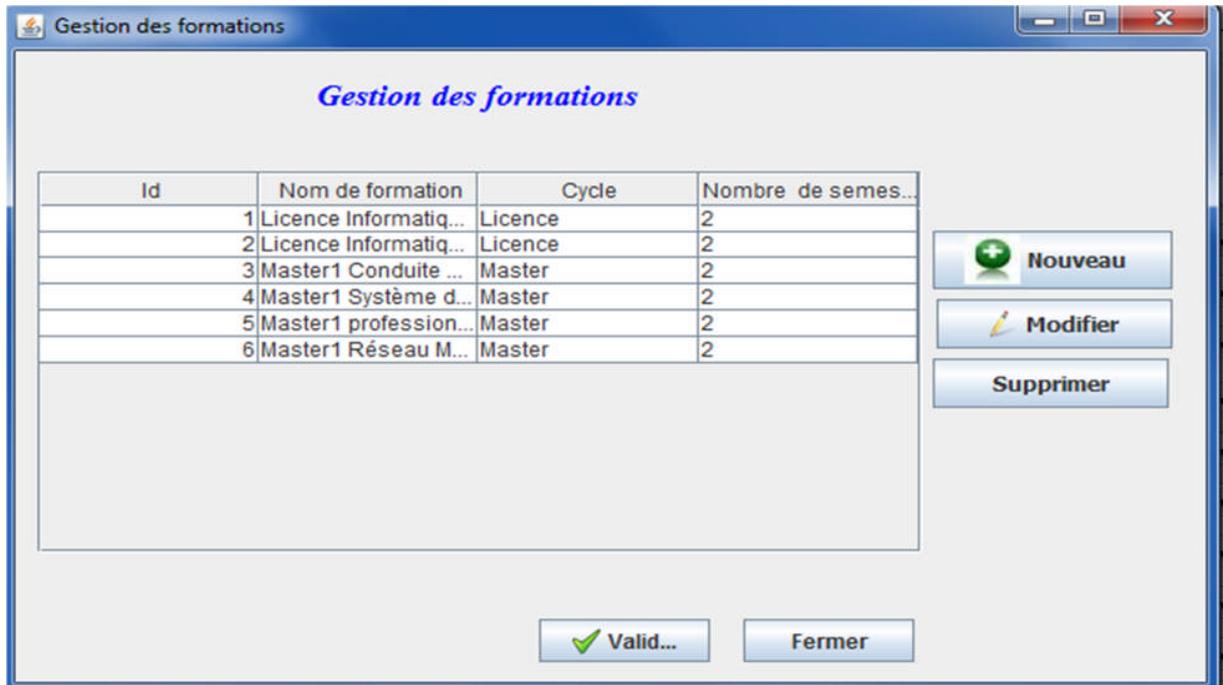


Figure III.8. Gestion des formations.

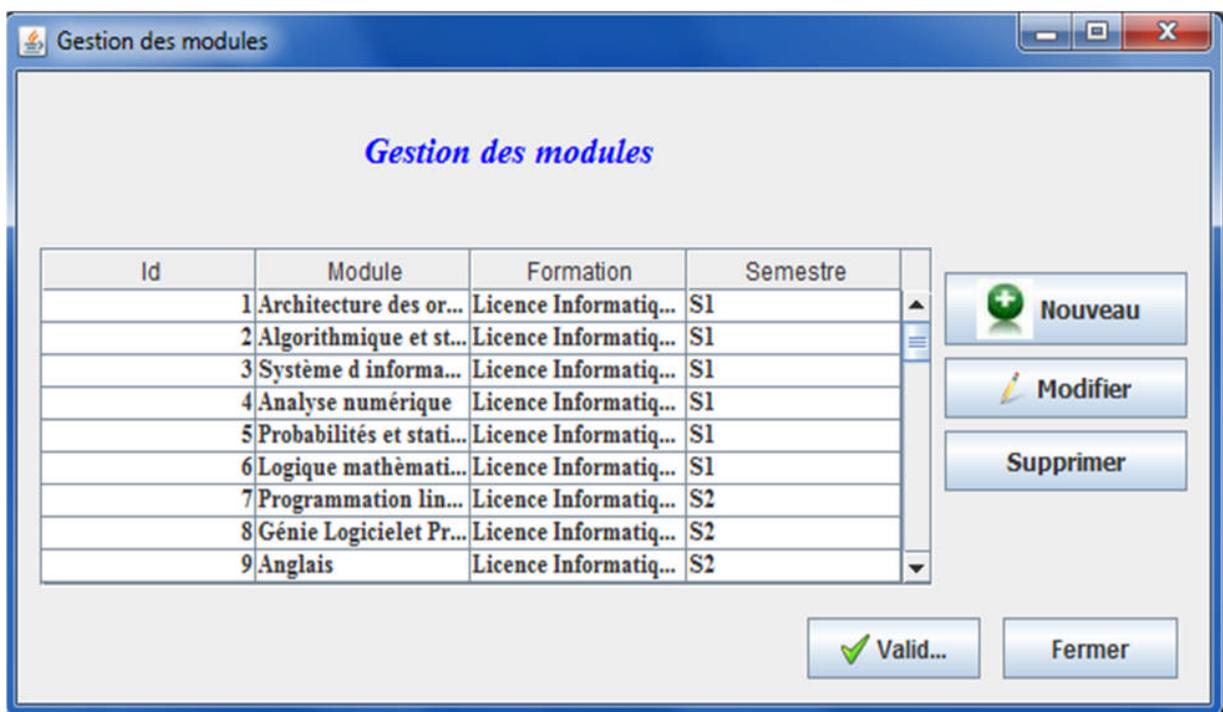


Figure III.9. Gestion des modules.

Année: 2015/2016

Formation: Master1 Conduite de projets informatiques

Semestre: S1

Section: A

Ajouter

Fermer

Figure III.10. La spécification d'un emploi de temps.

Programmer une nouvelle séance

Jour: Mercredi

Horaire: 21:30-14:00

Module: Réseaux et NTC

Enseignant: M KHEMLICH Mohamed Afficher tout

Type séance: Cours

Groupe: G1

Salle: Amphis A (capacité: 150)

Enregistrer

Fermer

Figure III.11. Programmer une séance.

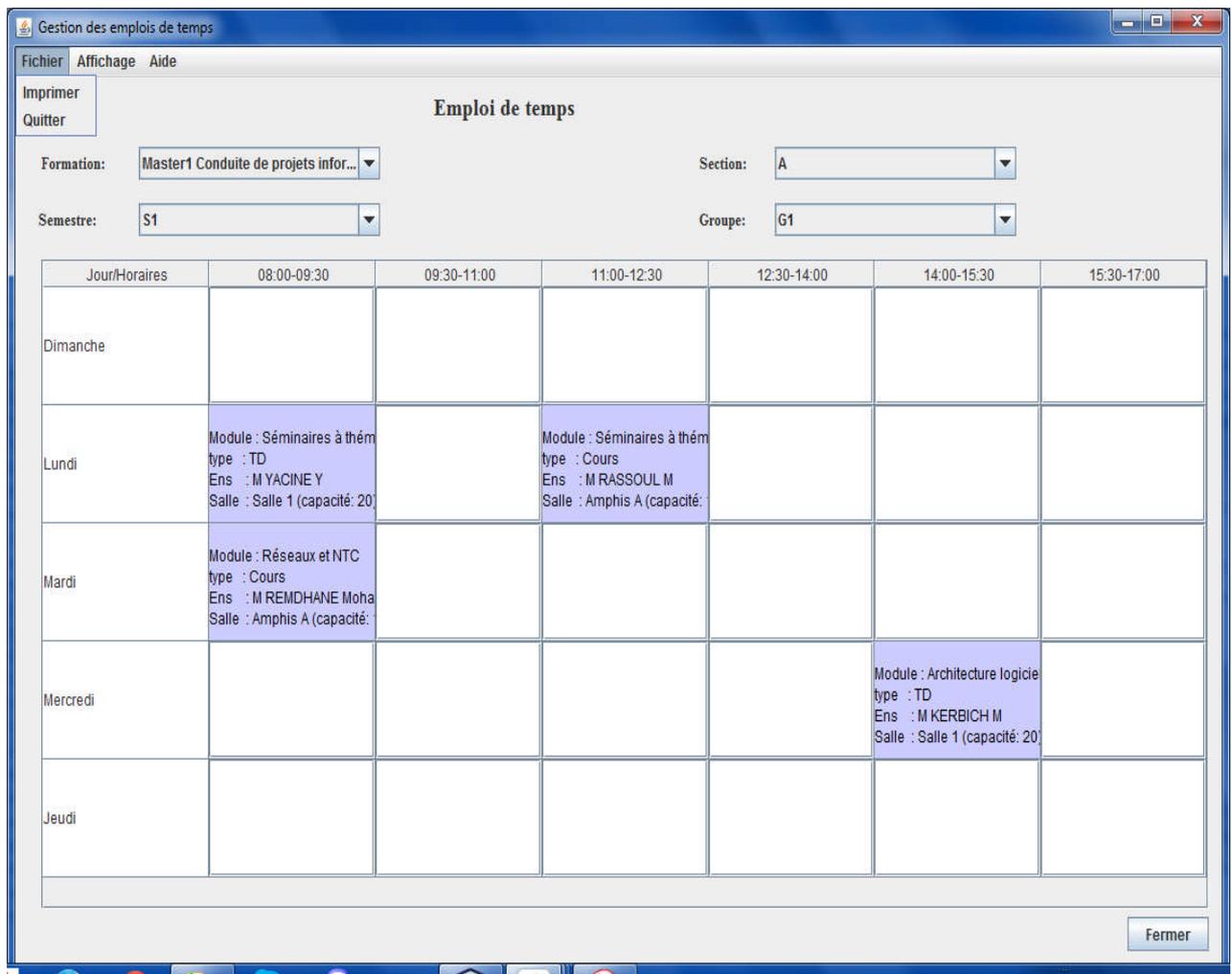


Figure III.12. Affichage d'un planning.

Conclusion

A la fin de la réalisation, le résultat sera une application qui fait la génération des emplois du temps des différentes promotions et enseignants sans aucun conflit au niveau des séances. Cette application a permis de répondre aux besoins des administratifs par la résolution des problèmes de la gestion des emplois du temps ainsi que les conflits qui peuvent exister.

Conclusion générale

Conclusion générale

L'objectif de ce travail a été l'analyse et la réalisation d'un projet de fin d'étude relatif à l'information de la génération semi-automatique d'emplois du temps au sein du département informatique de l'université Mouloud Mammeri de Tizi-Ouzou(UMMTO).

Dans ce projet, nous avons exploité nos efforts et nos connaissances pour pouvoir bien analyser le sujet de la génération d'emplois du temps. Nous espérons que ce travail peut ramener un plus au département informatique.

Ainsi, nous avons appris les différentes étapes de conception d'un système d'information en utilisant la méthode UML. D'une autre coté, la programmation en utilisant le langage JAVA, ainsi que la programmation des base des données avec SQL.

Perspectives

Nous souhaitons qu'on ait 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 :

- L'extension du système aux autres départements et facultés;
 - Injecter un mécanisme qui permet le traitement automatique et intelligent pour la génération de l'emploi du temps;
 - Améliorer l'interface du système.
-

Annexe

Introduction

La complexité croissante des systèmes d'information étaient derrière l'apparition des différents modes de modélisation, certains de ces modes ont montrés des limites, d'où la nécessité d'adapter ces modes à ces systèmes.

Dans l'objectif d'accomplir ces insuffisances, plusieurs méthodes sont apparues. Le groupe OMG a développé une notation standard utilisable dans les développements informatique basés sur l'objet, ainsi c'était l'apparition de l'UML (Unified Modelling Language) « Langage de Modélisation objet Unifié ».

1. La démarche de modélisation

Et pour réaliser ces objets nous avons suivi la démarche de modélisation suivante :

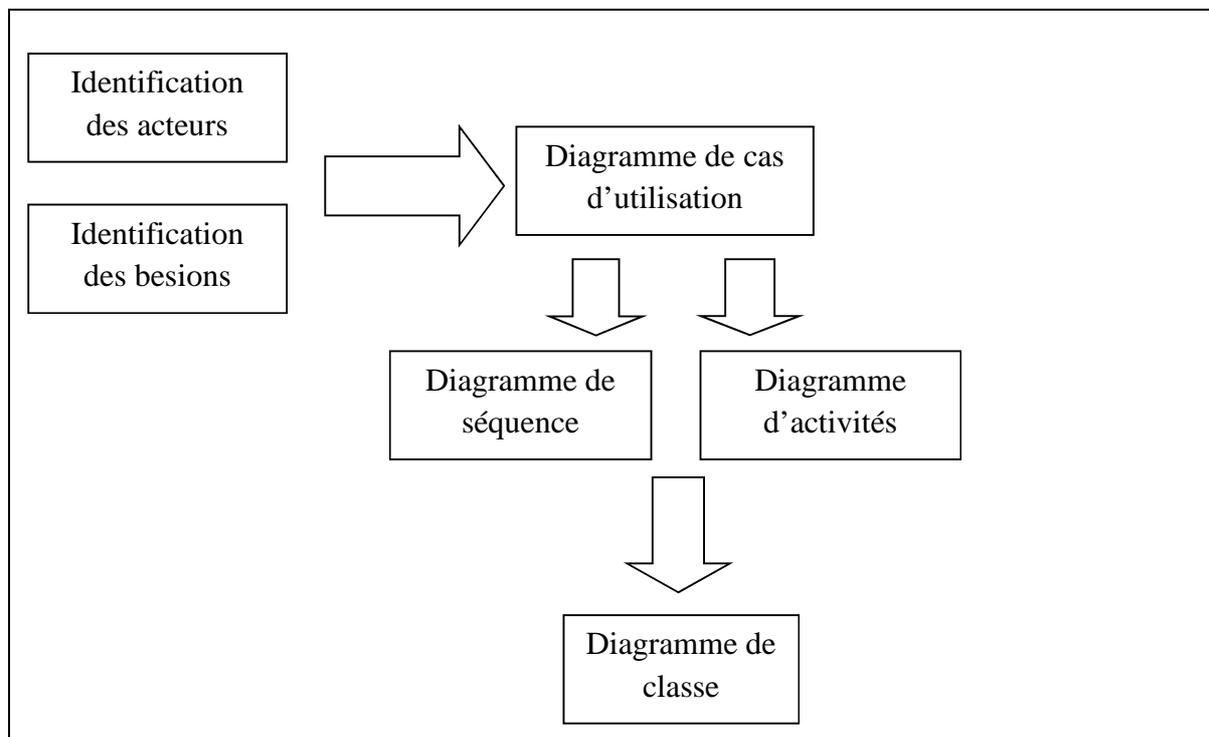


Figure 1. Méthodologie de modélisation de l'application.

. 2. Présentation d'UML [7]

UML est une norme OMG (Object Management Group), il permet d'exprimer et d'élaborer des modèles objet, il a été conçue pour servir de support à une analyse basée sur les concepts objets. C'est un langage formel servant d'un support de communication qui facilite la compréhension de solution objet grâce à l'utilisation des notations graphiques qui permettent de visualiser les solutions objet.

3. Notations UML [7]

UML est un langage de spécification et de représentation graphique des objets, il propose plusieurs modèles qui sont des descriptions du système étudié, et parmi ces modèles :

- Le modèle de classe qui capture la structure classique ;
- Le modèle des cas d'utilisation qui décrit les besoins de l'utilisateur ;
- Le modèle d'interaction qui décrit les scénarios et les flots de messages ;
- Le modèle des états qui exprime le comportement dynamique des objets;
- Le modèle de réalisation qui montre les unités de travail ;
- Le modèle de déploiement qui précise la répartition des processus ;

Ces modèles sont élaborés au moyen de diagramme qui spécifie un aspect précis du système ; UML offre une vue complète des aspects statiques et dynamiques en distinguant les diagrammes suivants :

- **Vue statique**

- ✓ Diagramme de cas d'utilisation.
- ✓ Diagramme d'objet.
- ✓ Diagramme de classe.
- ✓ Diagramme de composant.
- ✓ Diagramme de déploiement.

- **Vue dynamique**

- ✓ Diagramme de collaboration.
 - ✓ Diagramme de séquence.
 - ✓ Diagramme d'états-transitions.
 - ✓ Diagramme d'activités.
-

4. Le formalisme d'UML [8]

L'UML se décompose en plusieurs sous-ensembles :

4.1. Les modèles d'élément

Les modèles d'élément sont les briques des diagrammes UML, ces modèles sont utilisés dans plusieurs types de diagrammes.

Exemple d'élément : cas d'utilisation, classe, association, etc.

4.1.1. Élément structurels

4.1.1. 1. Cas d'utilisation : Un cas d'utilisation (en anglais *use case*) permet de mettre en évidence les relations fonctionnelles entre les acteurs et le système étudié. Le format de représentation d'un cas d'utilisation est complètement libre mais UML propose un formalisme et des concepts issus de bonnes pratiques.

Un acteur, au sens UML, représente le rôle d'une entité externe (utilisateur humain ou non) interagissant avec le système. Il est représenté par un bonhomme en fil de fer (en anglais *stick man*).

L'objectif poursuivi par les cas d'utilisation est de permettre de décrire la finalité des interactions du système et de ses utilisateurs.

- **Représentation**

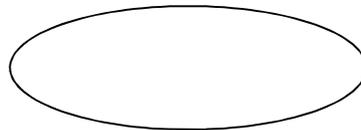


Figure 2 : Cas d'utilisation.

- **Relations entre les cas d'utilisation**

Deux types de relations standards entre cas d'utilisation sont proposés par

UML :

- **La relation d'inclusion** sert à enrichir un cas d'utilisation par un autre cas d'utilisation. Cette relation est représentée en UML par une flèche pointillée munie du stéréotype : «include».

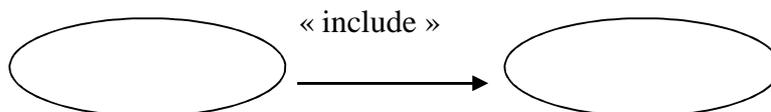


Figure 3 : La relation « include ».

- **La relation d'extension**

Cette relation enrichit un cas d'utilisation par un autre cas d'utilisation. Elle est représentée en UML par une flèche pointillée munie de stéréotype

« extend ».

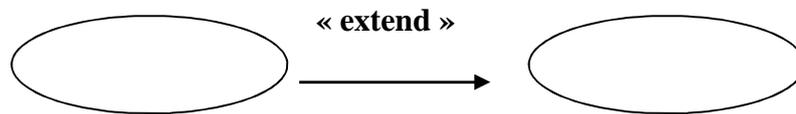


Figure 4 : La relation « extend ».

4.1.1.2. La classe

Une classe est un ensemble d'objets similaire, c'est-à-dire possédant la même structure et le même comportement et constituée des mêmes attributs et méthodes. Chaque classe est représentée sous forme d'un rectangle divisé en trois sections. Le premier contient le nom de la classe (non souligné), le second les attributs et le derniers les opérations.

A. Sa représentation UML

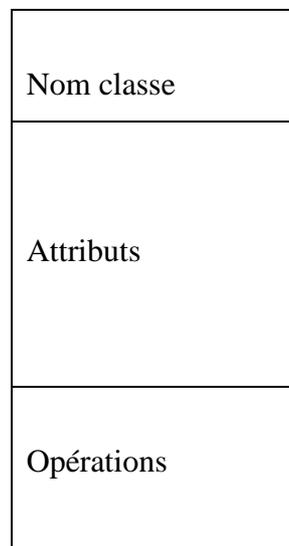


Figure 5 : La représentation UML d'une classe.

B. Les relations entre classes :

Les relations sont les observables du système. Elles décrivent le système d'un point de vue donné, qui peut être organisationnel, dynamique, temporel, architectural, géographique, logique, etc. En combinant toutes ces relations, il est possible de définir ou retrouver le système complet.

B.1 La relation d'héritage :

L'héritage constitue une relation de spécialisation. Elle est notée, en UML, par une flèche allant de la classe spécialisée vers la classe originale (de la classe vers la superclasse).

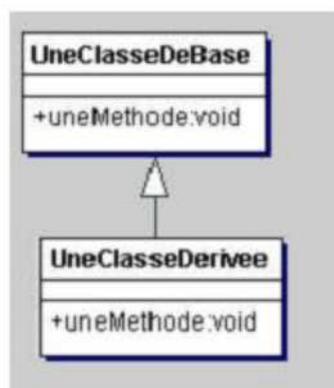


Figure 6 : Relation d'héritage.

On notera que la relation inverse n'est en principe pas documentée ; c'est que cette relation n'est en principe pas connue au moment de la conception d'un produit, et qu'elle est susceptible de changer au cours du temps. De plus, une superclasse n'est pas censée dépendre de ses dérivées.

B.2 La relation d'implémentation

Une classe peut implémenter une interface ; elle peut aussi en implémenter plusieurs. En notation UML, cette relation est dénotée par une flèche en traitillés.

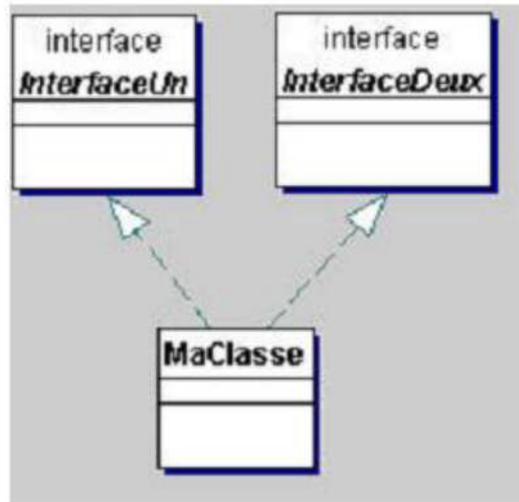


Figure 7 : Implémentation d'une interface.

B.3 La relation d'agrégation

Lorsqu'un objet en contient d'autres, on parle d'agrégation. Le diagramme de classes d'UML décrit cette relation par une flèche pleine.

Il faut noter que l'agrégation est parfois appelée "relation de contenance".

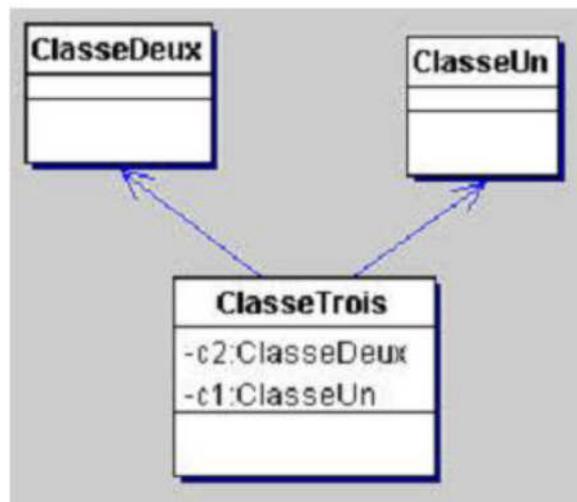


Figure 8 : Relation d'agrégation.

B.4 Relation de dépendance

La notion de dépendance est plus floue que les précédentes. Il est difficile de faire une liste complète des possibles relations de dépendance. Certains outils de modélisation offrent la possibilité de tracer automatiquement les relations directes : hélas, les relations indirectes sont souvent négligées, parce que difficiles à détecter ; d'autre part, le schéma résultant de ces outils

est souvent encombré de relations évidentes qu'il eût peut-être mieux valu, pour des raisons de lisibilité, passer sous silence.

C'est pourquoi beaucoup d'outils laissent à l'utilisateur le soin de définir ce type de relation.

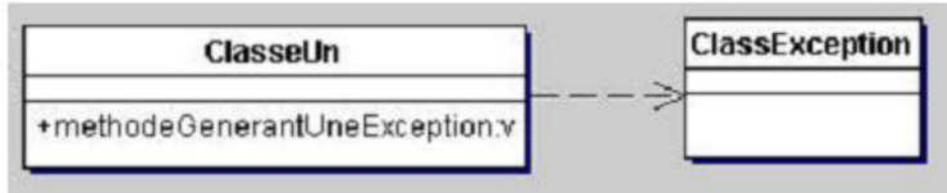


Figure 9 : Relation de dépendance.

Bibliographie

[1] : Chan Yew Chéong, Peter, « La planification du personnel : acteurs, actions et termes multiples pour une planification opérationnelle des personnes », Thèse de doctorat, Institut IMAG, Université Joseph Fourier-Grenoble, 1 octobre 2002.

[2] : Georges Weil, Kamel Heus, Patrice François, « Gymnaste : Aide à l'élaboration des roulements infirmiers. Du traitement des absences au management participatif », Laboratoire TIMC, SILM, CHU de Grenoble, Université Joseph Fourier- Grenoble, 1994.

[3] : Remy-Robert, Alexandre Joseph, « Systèmes interactifs d'aide à l'élaboration de plannings de travail de personnel », Thèse de doctorat, Laboratoire TIMC, Institut IMAG, Université Joseph Fourier-Grenoble, 07 novembre 2003.

[4] : Burke E. ,kingston j., Jackson k., Weare R., “Automated university Timetabling : the state of the art”, the Computer Journal 40 (9) 565-571, 1997.

[5] : De Werra D., “An Introduction to Timetabling”, European Journal of Operational research 19, 151-162.1985.

[6] : Schaerf A., et Schaerf M. “Local search techniques for large high school timetabling” , in proceeding of the 1 st international conferenceon the practice and theory of automated timetabling, pp. 313-323, 1995.

[7] : T. Chaari, Cours « Atelier UML », Maître assistant à l'institut supérieur d'électronique et de communication, tarak.chaari@gmail.com.

[8] : BUSH, V. (1945). As we may think. The Atlantic Monthly.

Liste des sites

[site, 00] : <http://www.google.fr/>

[site, 01] : <http://www.wikipedia.org/wiki/>
