

# Table des matières

## Remerciements

## Dédicaces

Introduction générale.....	1
----------------------------	---

## Chapitre I : Etude de l'Existant

I. Introduction.....	2
II. Présentation de L'Université Abou Baker Belkaid (UABT) .....	2
II.1 Organigramme de l'UABT.....	4
II.2 Présentation de pôle de Chetouane.....	5
II.3 L'Organigramme de La Faculté des Sciences.....	6
II.4 Le Service du Personnel .....	7
III. Mission des enseignants chercheurs.....	7
IV. Conclusion.....	7

## Chapitre II : Le langage UML

I. Introduction.....	8
II. Origines et Historique d'UML .....	8
III. Evolution d'UML.....	9
V. Les Caractéristiques d'UML .....	10
VI. Les éléments d'UML .....	10
VI.1 Les éléments structurels.....	10
VI.2 Les éléments de regroupement .....	11
VI.3 Les éléments de comportement .....	11

VI.4 Les relations .....	11
VII. Les vues .....	12
VII.1 Vue fonctionnelle .....	12
VII.2 Vue structurelle ou statique .....	12
VII.3 Vue dynamique .....	12
VIII. Les diagrammes .....	12
VIII.1 Diagramme structurelle ou statique .....	13
VIII.2 Diagramme de comportement .....	14
VIII.3 Diagramme d'interaction dynamique.....	15
IX .Processus UP .....	15
IX.1. Introduction.....	15
IX.2. Disciplines d'UP .....	15
IX.3. Phase du Processus Unifié .....	17
X. Conclusion .....	17

### **Chapitre III : Modélisation du système**

I. Introduction .....	19
II. Environnement de modélisation .....	19
III. Modélisation de système .....	19
III.1 Diagramme de cas d'utilisation.....	19
III.2 Les diagrammes de séquence .....	20
III.3 Diagramme de classe.....	28
IV. Modèle logique de données.....	30
V. Conclusion .....	31

### **Chapitre IV : L'implémentation du système**

I. Introduction .....	32
II. L'environnement de programmation Delphi .....	32
III. L'environnement de gestion de base de données .....	32

IV. Description de l'application.....	33
V. Conclusion .....	36
<b>Conclusion général .....</b>	<b>37</b>

# Table des illustrations

Figure I.1 : Organigramme de l'UABT .....	4
Figure I.2 : Organigramme de Faculté de Science .....	6
Figure III.1 : Diagramme de cas d'utilisation .....	20
Figure III.2 : Diagramme de Séquence « authentification » .....	21
Figure III.3 : Diagramme de Séquence « ajouter enseignant » .....	22
Figure III.4 : Diagramme de séquence « Supprimer enseignant » .....	23
Figure III.5 : Diagramme de Séquence « modifier enseignant » .....	24
Figure III.6 : Diagramme de séquence « promotion » .....	25
Figure III.7 : Diagramme de séquence « avancement » .....	26
Figure III.8 : Diagramme de séquence « titularisation » .....	26
Figure III.9 : Diagramme de séquence « gestion de congé » .....	27
Figure III.10 : Diagramme de séquence « impression attestation de travail » .....	27
Figure III.11 : Diagramme de séquence « attribution de fonction » .....	28
Figure III.12 : Diagramme de classe .....	29

# Introduction Générale

---

## **Introduction générale :**

Le monde informatique est un monde très vaste, son apparition correspond à un changement total de culture, jour après jour l'informatique occupe une place intéressante dans l'entreprise, puisqu'elle permet de mémoriser, de traiter et de diffuser l'information nécessaire dans un délais très court, qui permettra donc de prendre les bonnes décisions dans le temps voulu ;

La tendance actuelle des entreprises et des organisations est l'informatisation de leur système d'informations ou une partie de ce système, dans cette vision la faculté des sciences de l'université de Tlemcen essaie de mettre en œuvre des applications pour ces différents services ;

En effet, le nombre des enseignants de cette faculté a atteint 200 enseignants, ce nombre important rend le suivie et l'administration des enseignants une tache difficile et coûteuse en temps et en ressources humaines, ainsi le développement d'un logiciel est devenu une préoccupation prioritaire pour le doyen de la faculté ;

Notre projet de fin d'étude vient d'un coté pour répondre a ce besoin professionnel, et d'un autre coté pour appliquer les connaissances et la formation que nous avons acquis durant ces trois années ;

Dans cet objectif, notre travail consiste à réaliser une application de gestion du personnel enseignant. Pour la mise en œuvre de notre application nous avons utilisé l'environnement ModelMaker comme outils de modélisation en langage **UML**, ainsi que l'environnement Delphi pour la réalisation de l'application.

## **Structure du Mémoire :**

Ce mémoire est décomposé en quatre chapitres :

Le premier chapitre est consacré à l'Etude de l'Existant.

Le deuxième chapitre est consacré à la présentation du langage **UML** (Les définition, les descriptions, et l'utilité de différents diagrammes).

Le troisième chapitre est consacré à la modélisation du système étudié, on utilisant les différents diagrammes (diagramme de cas d'utilisation, diagramme de séquence et diagramme de classe) d'**UML**.

Le dernier chapitre consiste à la phase de réalisation et de mise en œuvre de l'application.

Finalement, nous terminons notre mémoire par une conclusion générale.

# **Chapitre I : Etude de l'Existant**

**I. Introduction**

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 service personnel concerné par notre étude. Par la suite, un intérêt particulier est porté à la gestion des personnels de la faculté de science. Dans un premier temps nous présenterons l'Université Abou Bekr Belkaid Tlemcen (**UABT**).

**II. Présentation de L'Université Abou Baker Belkaid (UABT) [1]**

L'Université Abou Baker Belkaid est une université située à Tlemcen en Algérie. Elle a été créée par le décret N°89-138 du 01-08-1989, modifié et complété par le décret exécutif N°95\_208 du 05/12/1998. L'UABT est le fruit d'une évolution progressive ascensionnelle.

L'enseignement supérieur était au début, une démarche à l'initiative d'un centre universitaire dont l'apothéose regroupait à l'origine (1974-1984) les seuls troncs communs des Sciences exactes & Biologie.

Cet enseignement s'est graduellement étendu à de nouvelles filières, couvrant ainsi d'exercice en exercice un ensemble de cycles de formations et exhibant à l'étudiant la possibilité de poursuivre l'intégrité de son cursus de graduation à Tlemcen.

Cette étape se caractérise aussi par la mise sur pied un enseignement de niveau « s » (Diplôme d'Etudes Universitaires Appliquées **DEUA**), par le développement de la première post-graduation dans le quasi totalité des filières évidentes à Tlemcen. Enfin par le lancement de la deuxième post graduation de puis 1991\_1992.

C'est à l'issue de ces quinze années de gestion qu'apparaît l'Université **UABT**, entité nouvelle, riche de cette longue période de maturation et ouverte à de nouveaux défis.

Ces derniers, qui s'inscrivent dans le cadre d'une stratégie plus globale de développement de l'université de Tlemcen, s'identifient à une volonté de développer de nouvelles filières. Cette politique allant de paire avec l'objectif d'une restructuration ensemble de la formation universitaire à Tlemcen, autour des pôles principaux, en particulier, notamment dans la ZHUN d'Imama et Chetouane.

L'Université de Tlemcen dispose actuellement d'un organe infrastructurel très important réparti entre sept pôles universitaires :

- Le pôle Chetouane.
- Le pôle Imama.
- Le pôle centre ville.
- Le pôle Kiffene.
- Le Bel-Horizon.
- Annexe Maghnia.

On s'intéresse dans cette étude par le pôle de chetouane, qui fera l'objet de la section suivante.



II.1 Organigramme de l'UABT :

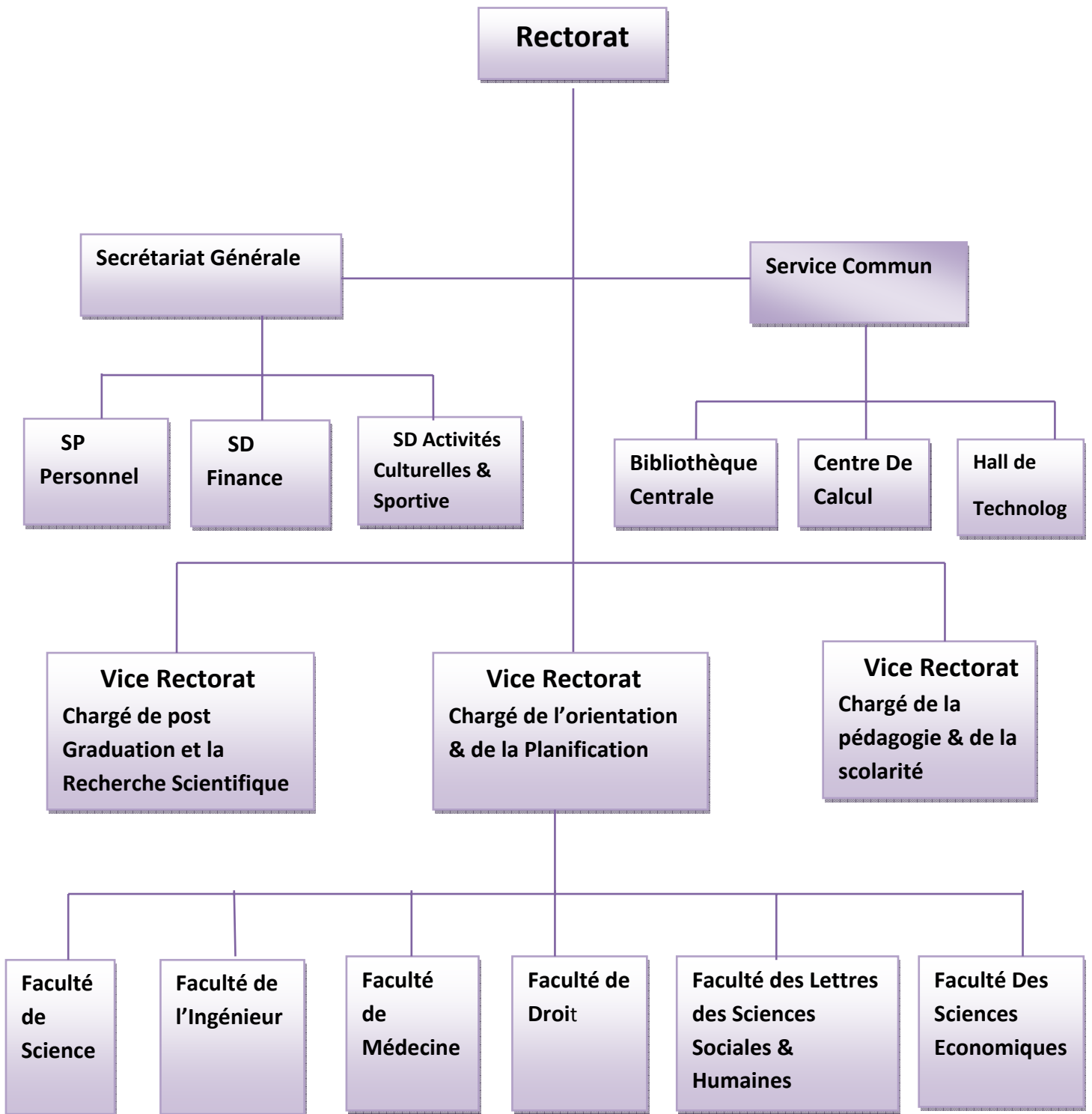


Figure I. 1: organigramme de l'UABT

**II.2 Présentation de pôle de Chetouane [1] :**

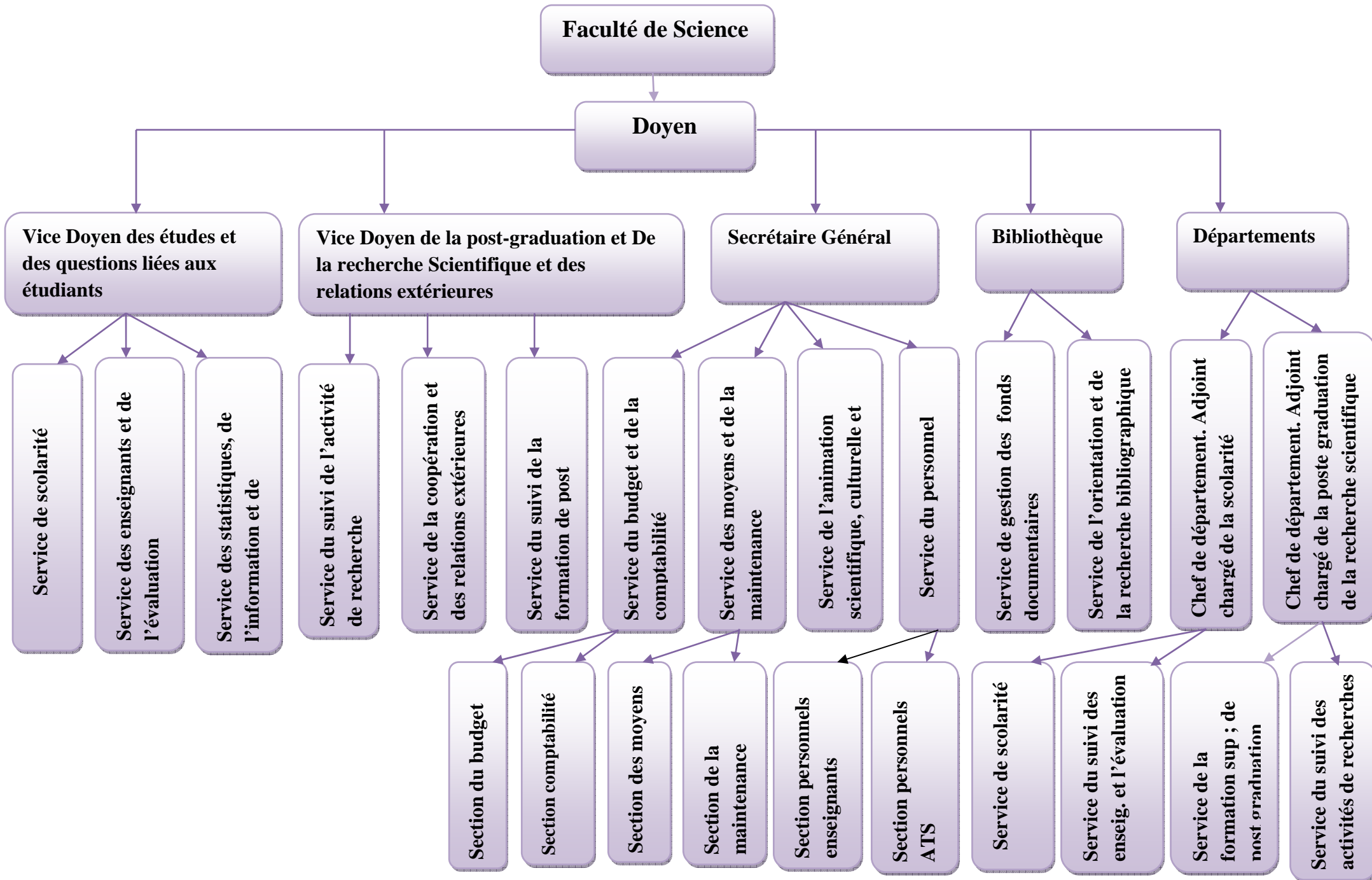
Le pôle de chetouane est situé au niveau de la commune de chetouane «<nord-est>> de la ville de Tlemcen dans un cadre agréable et à proximité de la zone industrielle, et il s'étend sur une superficie de 38 hectares. De plus, ce pôle abrite la faculté des sciences et la faculté de technologie pour une capacité théorique de 4400 places depuis 2004, composé de trois amphis de 300 places chacun et 44 salles plus 2 amphis de 150 places, 36 salles, 2 ateliers et 20 laboratoires.

Nous avons mené notre étude dans le Doyenné de faculté des sciences en particulier dans le service du personnel , cette faculté comporte les structures suivantes:

- 1) Les Départements :
  - Informatique.
  - Mathématique.
  - Chimie.
  - Physique.
- 2) Doyenné (administration).
- 3) Bibliothèque.
- 4) Laboratoires de recherches.

**II.3 L'Organigramme de La Faculté des Sciences :**

L'Organigramme de la figure I.2 présente la hiérarchie des différentes structures de la faculté des sciences.



## II.4 Le Service du Personnel :

La gestion du personnel recouvre l'ensemble des procédures et pratiques mises en œuvre pour administrer, contrôler et suivre le personnel implique dans l'activité d'une entreprise ou une organisation.

L'administration concerne essentiellement les activités suivantes :

- L'enregistrement, le suivie et le contrôle des données individuelles et collectives.
- L'application des dispositions légales et réglementaires dans l'organisation.
- La gestion de la promotion et de l'avancement dans le cycle de vie de la carrière du Personnel.
- La gestion des mouvements du Personnel (mutation, détachement,...).

## III. Mission des enseignants chercheurs : [2]

Les enseignants chercheurs, à travers l'enseignement et la recherche, accomplissent une mission de service public d'enseignement supérieur.

A ce titre, ils sont tenus de :

- ✓ Dispenser un enseignement de qualité et actualisé, lié aux évolutions de la science et des connaissances, de la technologie et des méthodes pédagogiques et didactiques, en confirmé avec les normes éthiques et professionnelles.
- ✓ Participer à l'élaboration du savoir et assurer la transmission des connaissances au titre de la formation initiale et continue.
- ✓ Mener des activités de recherches-formation pour développer leurs aptitudes et leurs capacités à exercer la fonction d'enseignant chercheur.

## III. Conclusion :

Dans ce chapitre nous avons présenté une étude détaillée sur le service personnel concerné par l'étude notre système « gestion personnel enseignant ». La modélisation de ce système se fait par le langage UML qui se présenté dans le chapitre suivant.

## **Chapitre II : Le langage UML**

### I. Introduction :

Le génie logiciel et la méthodologie s'efforcent de couvrir tous les aspects de la vie du logiciel. Issus de l'expérience des développeurs, concepteurs et chefs de projets, ils sont en constante évolution, parallèlement à l'évolution des techniques informatiques et du savoir-faire des équipes.

Comme toutes les tentatives de mise à plat d'une expérience et d'un savoir-faire, les méthodologies ont parfois souffert d'une formalisation excessive, imposant aux développeurs des contraintes parfois contre-productives sur leur façon de travailler.

Avec la mise en commun de l'expérience et la maturation des savoir-faire, on voit se développer à présent des méthodes de travail à la fois plus proches de la pratique réelle des experts et moins contraignantes.

**UML** qui se veut un instrument de capitalisation des savoir-faire puisqu'il propose un langage qui soit commun à tous les experts logiciel, va dans le sens de cet assouplissement des contraintes méthodologiques.

### II. Origines et Historique d'UML [3]:

La modélisation objet consiste à créer une représentation informatique des éléments du monde réel auxquels on s'intéresse, sans se préoccuper de l'implémentation. Il s'agit donc de déterminer les objets présents et d'isoler leurs données et les fonctions qui les utilisent. Pour cela des méthodes ont été mises au point. Entre 1970 et 1990, de nombreux analystes ont mis au point des approches orientées objets, si bien qu'en 1994 il existait plus de 50 méthodes objet. Toutefois seules 3 méthodes ont véritablement émergé :

- **OMT** de **James Rumbaugh** (General Electric) qui fournissait une représentation graphique des statique et fonctionnel d'un système ;
- **OOD** de **Gready Booch** (Department of Defense) qui introduisait le concept de paquetage (package) ;
- **OOSE** de **Jacobson** (Ericsson) qui fondait son analyse la description des besoins des utilisateurs (cas d'utilisation, ou use case).

Chaque méthode avait ses avantages et ses partisans. Bien que le nombre de méthodes s'était accru, la profession risquait encore de se diviser entre ces trois méthodes, créant autant de communautés qui auraient du mal à communiquer.

Dans un objectif de normalisation et de standardisation les trois constructeurs qui régnaient chacun sur l'une des trois méthodes se mirent d'accord pour définir une méthode commune qui fédérerait leurs apports respectifs (ils sont depuis surnommés « the Amigos »).

**UML** (Unified Modeling Language) pour langage de modélisation unifié est né de cet effort de convergence. Le terme langage est quant à lui employé pour souligner qu'**UML** a l'ambition d'être un langage et non celle d'être une méthode ;

### III. Evolution d'UML:[3]

**UML** a progressé par étapes. Booch et Rumbaugh et quelques autres se mirent tout d'abord d'accord en 1995 pour concevoir une méthode unifiée. unified méthode 0.8. Jacobson les rejoignit en 1996 pour aboutir à **UML 0.9** (il faut noter le remplacement du mot méthode par le mot langage plus modeste et surtout plus adéquat). Les autres les plus importants dans le monde du logiciel (IBM, Microsoft, Oracle, DEC, HP, Rational, Unisys ...) s'associèrent alors à l'effort et **UML 1.0** fut soumis à l'OMG (Object Management Group). L'OMG l'adopte comme langage de modélisation des systèmes d'information à objets en novembre 1997 sous sa version 1.1, date à partir de laquelle **UML** devient un standard international. La version qui a vu nos jours est la version **UML 2.2** en 2009.

### IV. Définition UML [3]

**UML** (sigle désignant l'unified modeling language ou «langage de modélisation unifié») est un langage de modélisation graphique à base de pictogrammes, il est apparu dans le monde du génie logiciel, dans le cadre de la «conception orientée objet» **UML** est couramment utilisé dans les projets logiciels.

**UML** est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet, **UML** offre un standard de modélisation, pour représenter l'architecture logicielle.

### V. Les Caractéristiques d'UML [4] :

- 1) **UML** est un langage formel et normalisé caractérisé par :
  - ✓ Gain de précision.
  - ✓ Gage de stabilité.
  - ✓ l'utilisation d'outils.
- 2) Encourage un support de communication performant
  - Il cadre l'analyse.
  - Il facilite la compréhension de représentations abstraites complexes.
  - Son caractère polyvalent et souple en font un langage universel.
- 3) La mise en pratique d'**UML** nécessite un apprentissage et passe par une période d'adaptation.
- 4) Le processus (nom couvert par **UML**) est une autre clé de la réussite d'un projet. Or, l'intégration d'**UML** dans un processus n'est pas triviale et améliorer un processus est une tâche complexe et longue.

### VI. Les éléments d'UML :

#### VI.1 Les éléments structurels :

- a) **Les classes** : Une classe est un type abstrait caractérisé par des propriétés (attributs et méthodes) communes à un ensemble d'objets et permettant de créer des objets ayant ces propriétés.

Classe =attributs+méthodes+instanciation.
- b) **Use case** : Use case (cas d'utilisation) : utilisation du système. Les cas d'utilisation permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système.
- c) **Composant** : C'est une partie physique d'un système.
- d) **Nœud** : C'est un élément physique. Lors d'une exécution, il représente une ressource ayant une capacité de calcul.



- e) **Interface:** C'est un ensemble d'opération définissant la fonction ou d'un composant.
- f) **La collaboration :** Elle définit une interaction entre plusieurs éléments, elle a un rôle structurel et comportemental.

### VI.2 Les éléments de regroupement :

Ils représentent les parties organisationnelles du modèle il existe différents types d'éléments organisationnels :

- a) **Package :** Un paquetage étant un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle **UML**
- b) **Les éléments d'annotation :** les notes permettent de donner des informations.

### VI.3 Les éléments de comportement:

- a) **Les Interactions :** elles sont constituées par l'envoi de messages ou d'événements provoquant des actions chez le récepteur.
- b) **Les états :** ils permettent' de constituer des automates d'écrivant le comportement d'une classe ou d'une méthode.
- c) **Les activités :** les activités sont des comportements exécutables séquentiellement ou parallèlement.

### VI.4 Les relations :

- a) **La dépendance :** c'est une relation sémantique indiquant que tout changement de l'élément indépendant peut affecter l'élément dépendant.
- b) **L'association :** c'est une relation structurelle d'écrivant un ensemble de liens entre différents objets.
- c) **La généralisation :** cette relation permet de définir des niveaux d'abstraction entre les classe, le but est de permettre de manipuler de façon homogène des ensembles d'objets qui partagent les mêmes propriétés. Cette factorisation des traitements s'appelle le polymorphisme.

### **VII. Les vues [5]:**

Les vues observables du système le décrivent d'un point de vue donné, qui peut être organisationnel, dynamique, temporel, architectural, géographique, ...etc. En combinant tous ces vues, il est possible de définir le système complet.

#### **VII.1 Vue fonctionnelle :**

Interactive, qui est représentée à l'aide de diagrammes de cas d'utilisation et diagrammes des séquences.

Elle cherche à appréhender les interactions entre les différents acteurs/utilisateurs et le système, sous forme d'objectif à atteindre d'un côté et sous forme chronologique de scénarios d'interaction typiques de l'autre côté.

#### **VII.2 Vue structurelle ou statique :**

Réunit les diagrammes des classes et diagrammes de packages. Les premiers favorisent la structuration des données et tentent d'identifier les objets/composants constituant le programme, leurs attributs, opération et méthodes, ainsi que les liens ou associations qui les unissent. Les seconds s'attachent à regrouper les classes fortement liées entre elles en des composants les plus autonomes possibles.

#### **VII.3 Vue dynamique :**

Qui est exprimée par les diagrammes d'états. Cette vue est plus algorithmique et orientée « traitement », elle vise à décrire l'évolution (la dynamique) des objets complexes du programme tout au long de leur cycle de vie. De leur naissance à leur mort, les objets voient leur changement d'états guidés par les interactions avec les autres objets.

### **VIII. Les diagrammes [5]:**

Les diagrammes sont des éléments graphiques, ceux-ci décrivent le contenu des vues, qui sont des notations abstraites. Les diagrammes peuvent faire partie de plusieurs vues.

### VIII.1 Diagramme structurelle ou statique :

#### a) Diagramme de classe :

Un diagramme des classes décrit le type des objets du système ainsi que les différentes formes de relation statique qui les relient entre eux. On distingue classiquement deux types principaux de relation entre objets :

Les associations, bien connues des vieux modèles entité/association utilisés dans la conception des bases de données depuis les années 70 ;

Les sous types, particulièrement en vogue en conception orientée objets, puisqu' ils s'expriment très bien à l'aide de l'héritage en programmation.

#### b) Diagramme d'objet :

Un objet est une instance d'une classe, un diagramme d'objets est un ensemble d'objets respectant les contraintes du diagramme de classe, respect des cardinalités. Chaque attribut d'une classe a une valeur affectée dans chaque instance de cette classe.

#### c) Diagramme de composant :

Il permet de montrer les composants du système d'un point de vue physique, tels qu'ils sont mise en œuvre.

#### d) Diagramme de déploiement :

Il sert à représenter les éléments matériels et la manière dont les composants du système sont répartis sur ces éléments matériels et interagissent entre eux.

#### e) Diagramme de paquetages :

Le diagramme de paquetage sert à représenter les dépendances entre paquetage. C'est- à dire les dépendances entre ensembles de définition.

#### f) Diagramme de structure composite :

Permet de décrire sous forme de boîte blanche les relations entre composants d'une classe.

### VIII.2 Diagramme de comportement :

#### a) Diagramme de cas d'utilisation :

Un cas d'utilisation modélise une interaction entre le système informatique à développer et un utilisateur ou un acteur interagissant avec le système. Plus précisément, un cas d'utilisation décrit une séquence d'actions réalisées par le système qui produit un résultat observable pour un acteur.

Il y a en général deux types de description des use cases :

- ❖ Une description textuelle de chaque cas ;
- ❖ Le diagramme des cas d'utilisation qui présente une synthèse de l'ensemble des cas ;
  - **Dépendances entre cas d'utilisation :**

Il est parfois intéressant d'utiliser des liens entre cas, **UML** en fournit deux types :

La relation **utilise** (include) : est employée quand deux cas d'utilisation ont en commun une même fonctionnalité et que l'on souhaite factoriser celle-ci en créant un sous-cas, ou cas intermédiaire, afin de marquer les différences d'utilisation.

La relation **étend** (ex-tend) : nous dirons qu'il y a extension d'un cas d'utilisation quand un cas est globalement similaire à un autre ou lorsque un cas doit être spécialisée ou adaptée.

#### a) Diagramme état- transitions :

Permet de décrire sous forme de machine à états finis le comportement du système ou de ses composants.

#### b) Diagramme d'activité :

Permet de décrire sous forme de flux ou d'enchaînement d'activités le comportement du système ou de ses composants.

### VIII.3 Diagramme d'interaction dynamique :

#### a) Diagramme de séquence :

Représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.

#### b) Diagramme de communication :

Représentation simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets.

#### c) Diagramme global d'interaction :

Permet de décrire les enchainements possibles entre les scénarii préalablement identifiés sous forme de diagramme de séquences.

## IX .Processus UP [5]

**UML** est un langage qui permet de représenter des modèles, mais il ne définit pas le processus d'élaboration des modèles. De ce fait, il faudra choisir un processus (démarche à suivre) pour pouvoir modéliser avec ce langage. Plusieurs processus existent parmi eux le Processus Unifier UP.

### IX.1. Introduction :

Complément idéal d'**UML**, le processus unifié (UP pour unified process) est le processus de développement logiciel né de la fusion des travaux d'Ivar Jacobson, Grady Booch et de James Rumbaugh, les trois concepteurs du langage **UML**. Fruit des meilleures pratiques de l'ingénierie logicielle, le processus unifié décrit les activités qui permettent de traduire les besoins d'un utilisateur en un système logiciel.

Le processus unifié est décrit par ses auteurs comme une méthode pilotée par les cas d'utilisation, centrée sur l'architecture, **itérative** et **incrémentale**.

### IX.2. Disciplines d'UP :

Les disciplines permettent de regrouper les différentes activités du PU en six disciplines d'ingénierie (workflows du processus), directement liées au processus de développement, et trois disciplines de support (workflow de soutien).

Les neufs disciplines du PU sont :

### **1. La modélisation du métier :**

Il s'agit d'identifier les acteurs, les processus métier et la vision métier que le projet doit implémenter.

### **2. La gestion des exigences :**

Basée sur les cas d'utilisation, est une discipline essentielle d'UP qui vise à capturer le plus fidèlement possible les exigences des utilisateurs du projet.

### **3. L'analyse et la conception :**

Ont pour objectif de traduire les cas d'utilisation en autant de vues de l'architecture du logiciel que nécessaire. Cette discipline peut exploiter la palette des diagrammes UML pour la production de ces différentes vues

### **4. L'implémentation :**

Vise le triple objectif de raffiner le modèle de conception, générer le code source et les tests unitaires associés, intégrer le travail d'implémentation des différentes équipes travaillant en parallèle

### **5. Les tests :**

Jouent un rôle central dans le PU. Ils sont continuellement mis en œuvre pendant le processus de développement. Ils doivent également comprendre des tests de non régression essentiels dans tout processus incrémental.

### **6. Le déploiement :**

Prend en charge les activités de configuration et de conditionnement du système à livre.

### **7. La gestion de la configuration et des changements :**

Cette discipline utilisera avantageusement des systèmes de gestion de versions comme CVS (Concurrent Versions system) par exemple.

### **8. La gestion de projet :**

Est une discipline critique du PU qui a la responsabilité de planifier et de piloter l'ensemble des activités du projet.

### **9. L'environnement :**

Est la discipline responsable de la logistique au sens large du projet : normes, standards, environnement de développement, infrastructure matérielle...

### **IX.3. Phase du Processus Unifié :**

Les itérations d'UP s'inscrivent dans quatre phases successives dont la validation constitue des jalons importants du processus de développement :

#### **a) Initialisation :**

La phase d'initialisation a pour objectif de trouver un compromis entre les exigences et contraintes.

#### **b) Elaboration :**

Cette phase devrait également conduire à une révision et une précision du planning du projet.

#### **c) Construction :**

C'est dans cette phase que la capture des exigences doit être finalisée, mais aussi et surtout, que les différents incréments de l'application doivent être conçus et implémentés.

#### **d) Transition :**

Cette phase consiste à finaliser le produit et à effectuer la livraison du système auprès des utilisateurs finaux.

## **X. Conclusion**

Comme **UML** n'impose pas de méthode de travail particulière, il peut être intégré à n'importe quel processus de développement logiciel de manière transparente, en effet

nous avons choisi le processus UP qu'on utilisera pour modéliser notre système. Cette modélisation sera détaillée dans le chapitre suivant.



## **Chapitre III : Modélisation du système**

### I. Introduction :

Dans ce chapitre nous introduisons l'analyse de notre système en utilisant le langage **UML**, en s'appuyant sur le processus UP, en conséquence, nous allons détailler les trois étapes : tous d'abord, nous commencerons par définir le diagramme de cas d'utilisation, ensuite les cas d'utilisation vont être détaillés en plusieurs diagrammes de séquences, nous terminerons par représenter le diagramme de classe qui décrit la structure statique de notre système.

### II. Environnement de modélisation :

Il existe plusieurs outils de modélisations **UML** qui sont disponibles comme logiciels payants ou gratuits, on peut citer : PowerAmc, Objecteering, Describe, RationalnRose Modeler, ArgoUml, ModelMaker .

Nous avons utilisé l'outil Modelmaker Edition Pascal pour la création des trois types de diagrammes.

- **Modelmaker [6]** : permet d'éditer les diagrammes d'**UML**, il inclut deux éditions : une édition pascal et une édition C # et de générer un code en pascal objet dédié pour un environnement Delphi.

### III. Modélisation de système :

#### III.1. Diagramme de cas d'utilisation :

Ce diagramme permet de décrire l'interaction entre les acteurs et le système.

##### a. Identification des acteurs du service personnel :

Un acteur est l'idéalisation d'un rôle joué par une personne ou un groupe de personnes. L'acteur qui interagit avec notre système est :

- **Chef de service** : utilisateur de système.

##### b. Identification des cas d'utilisation :

Un cas d'utilisation présente une fonctionnalité du système, notre système comporte les cas d'utilisation suivant :

- Gestion de l'enseignant.
- Titularisation de l'enseignant.
- Gestion des Promotion de l'enseignant.
- Gestion d'avancement.
- Gestion de congé.
- Impression d'attestation de travail.

- Attribution de fonction.

Les cas d'utilisation sont schématisés dans le diagramme représenté dans la Figure III.1

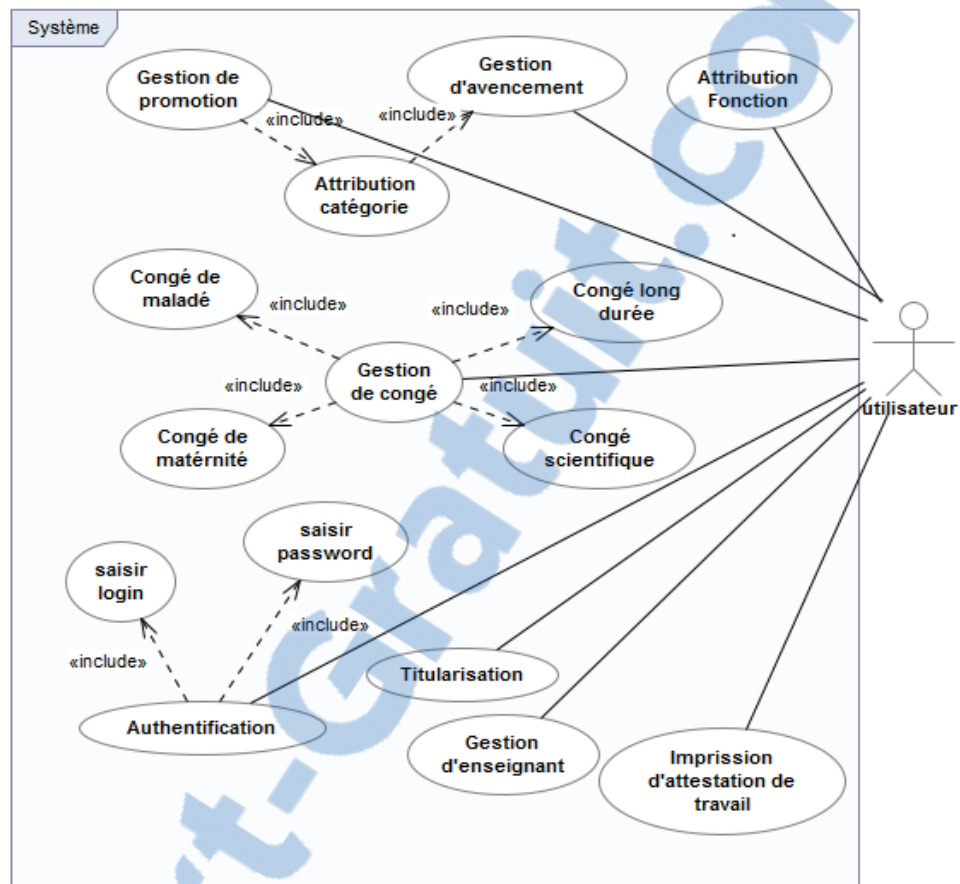


Figure III.1: Diagramme de cas d'utilisation

### III.2. Les diagrammes de séquence :

Dans cette partie nous allons présenter les interactions des objets du système par un diagramme de séquence pour chaque scénario de chaque cas d'utilisation.

#### a) Authentification :

Lorsque l'utilisateur demande l'accès à l'application, il doit tout d'abord saisir son login et son mot de passe, s'il possède un compte, le système va afficher le menu principal de l'application sinon il affiche un message d'erreur. Ce scénario est présenté par le diagramme de séquence de la figure III.2.

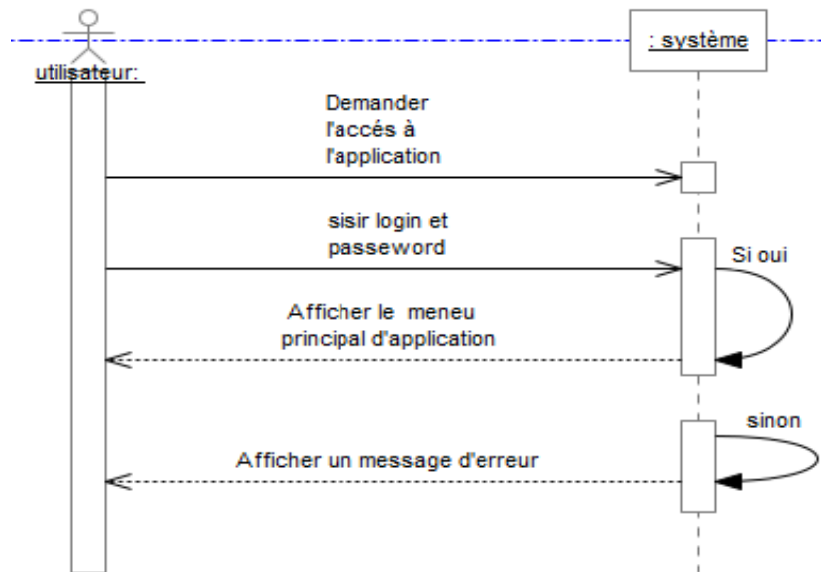


Figure III.2 diagramme de séquence « authentification »

#### b) Gestion enseignant :

Ce cas comporte trois scénarios.

##### ➤ Ajouter enseignant :

L'ajout d'un enseignant peut être lancé après le recrutement d'un enseignant ou après sa mutation.

Après la réception d'un dossier administratif de l'enseignant, l'utilisateur va saisir les données de nouveau enseignant (nom, prénom, date-naissance, diplôme, grade, fonction...) et enregistrer ces données. Ce scénario est présenté dans la figure III.3

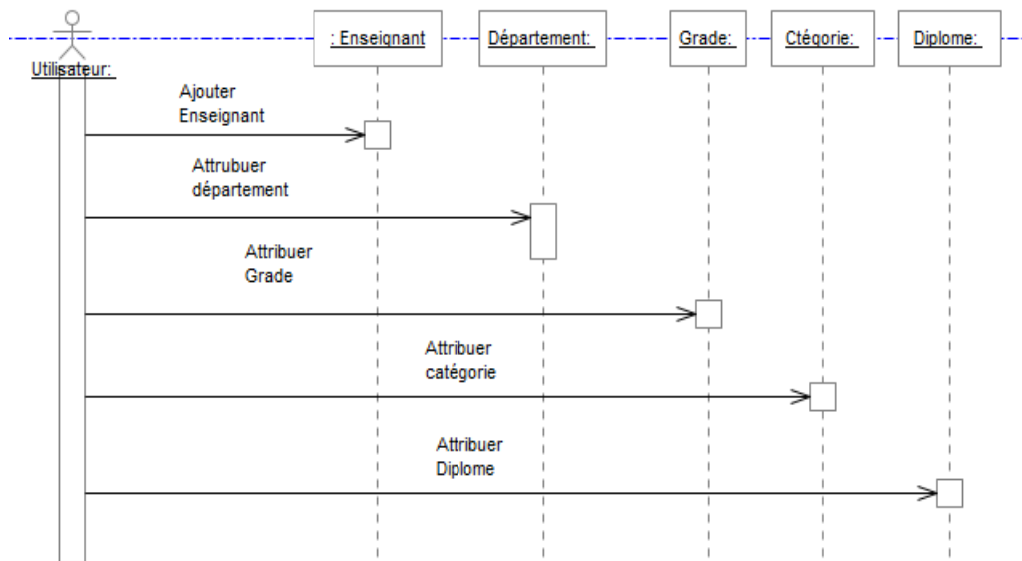


Figure III.3 diagramme de séquence « ajouté enseignant »

### ➤ Suppression d'enseignant :

Ce scénario est activé dans les cas suivants :

- Suite au décès de l'enseignant.
- Suite à la mutation de l'enseignant vers une autre université.
- Suite à sa démission.
- Suite à sa mise en retraite ;

L'utilisateur sélectionne l'enseignant et le supprime, le système demande la confirmation de la suppression et l'utilisateur valide ou annule la suppression. Le diagramme suivant (figure III.4) illustre ce scénario.

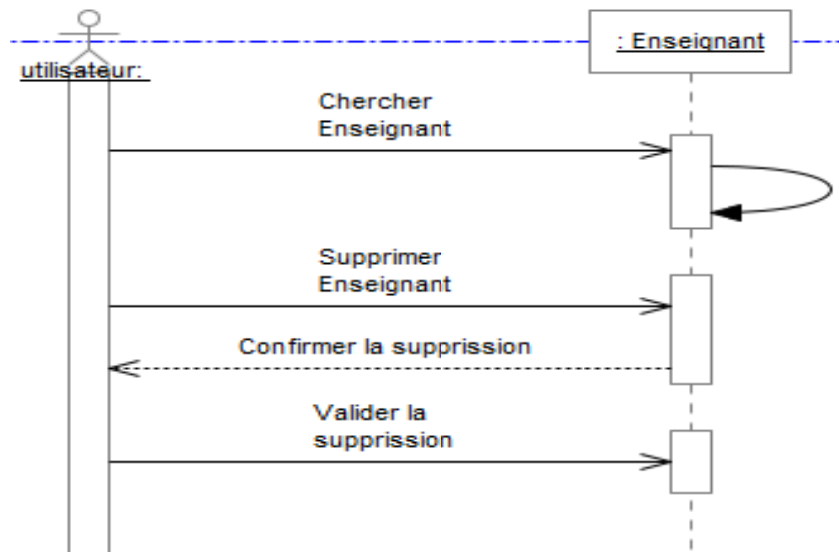


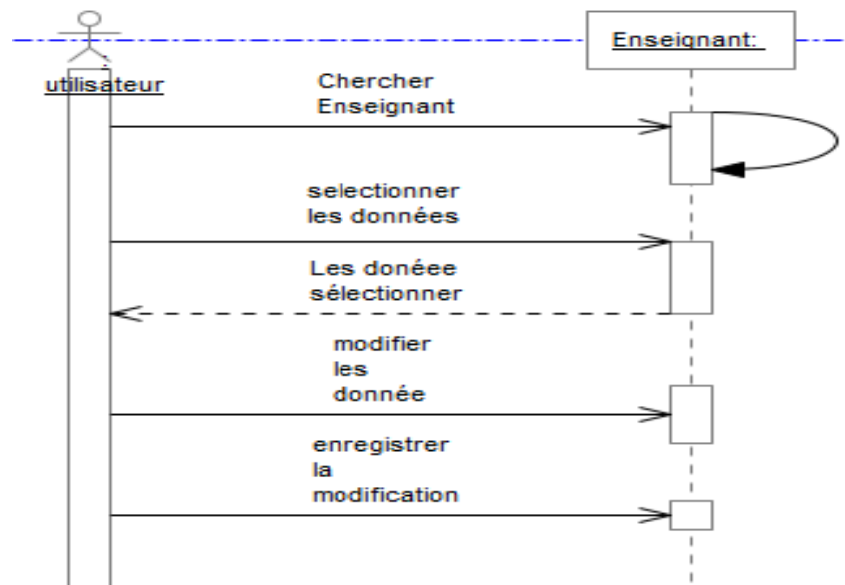
Figure III.4 : diagramme de séquence « supprimé enseignant »

➤ **Modifier enseignant :**

Ce scénario est activé dans les cas suivants :

- Lors de la modification des données de l'enseignant (changement d'adresse, de la situation familiale...)
- Lors de La saisie des donnés erronées (adresse, nom, date de naissance...) lors de l'ajout de l'enseignant.

L'utilisateur va sélectionner l'enseignant et sélectionne les données a modifier puis il valide les modifications. Le diagramme suivant (figure III.5) montre ce scénario :



*Figure III.5 : diagramme de séquence « modifié enseignant »*

**c) Promotion de l'enseignant :**

L'utilisateur cherche l'enseignant dans le fichier et vérifie les conditions de promotion, si c'est le cas, il va attribuer le nouveau grade, mis à jour la nouvelle catégorie et le nouveau échelon, puis il crée la décision de promotion et lance son impression. Ce scénario est présenté dans la figure ci-dessous (figure III.6).

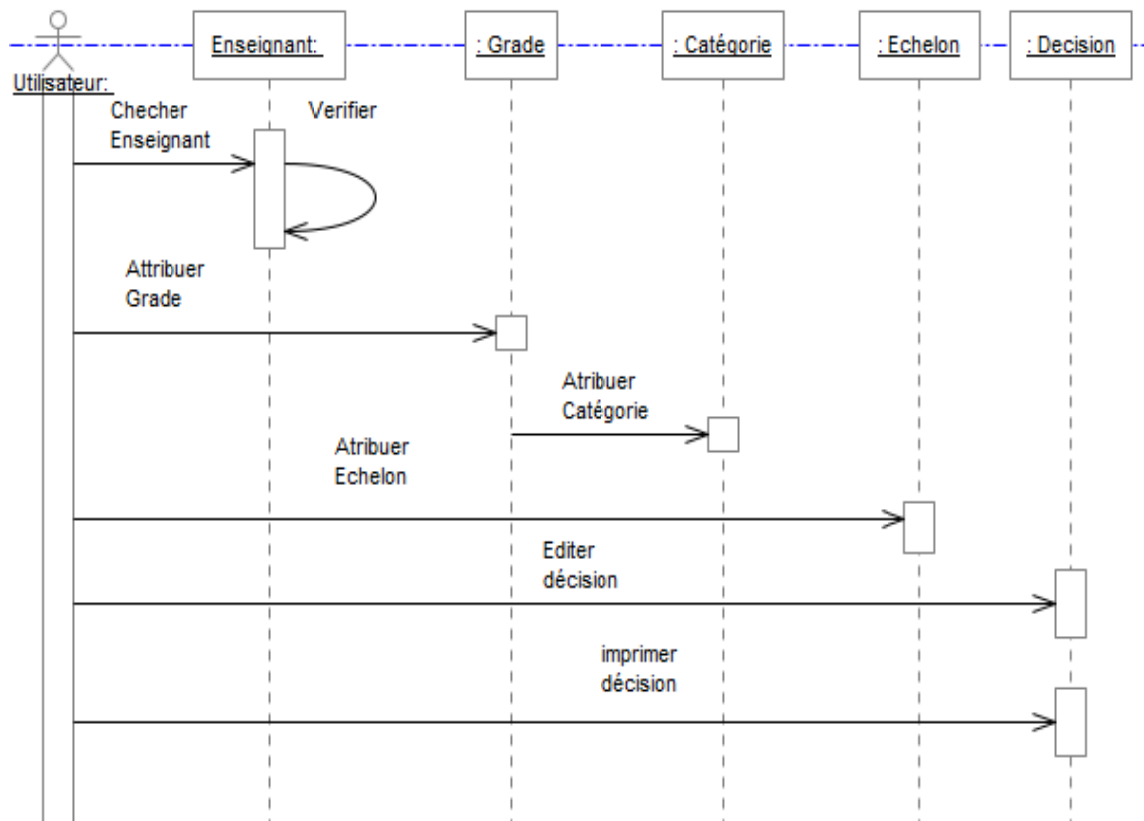


Figure III.6 : diagramme de séquence « promotion »

#### d) Gestion de l'avancement:

L'avancement dans les échelons est présenté comme suit :

L'utilisateur cherche l'enseignant et vérifie les conditions d'avancement, si c'est le cas, il sélectionne le nouvel échelon, puis il crée la décision d'avancement et l'imprime. Le diagramme suivant (figure III.7) montre ce scénario.



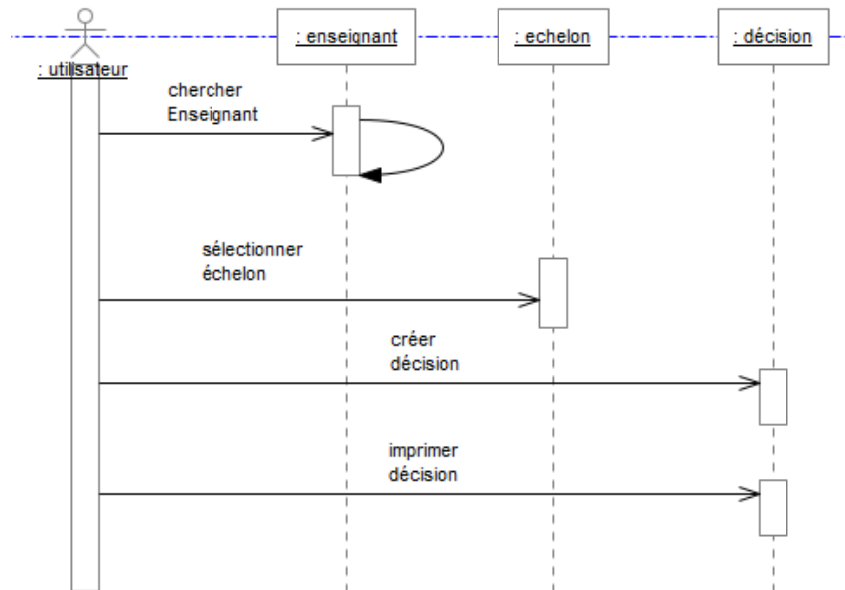


Figure III.7 diagramme de séquence « avancement »

**e) Titularisation de l’enseignant :**

La titularisation des enseignants chercheurs est présentée comme suit :

A la réception de l’avis du Comité Scientifique de Département pour la titularisation de l’enseignant, l’utilisateur doit d’abord sélectionner l’enseignant et vérifie les conditions de titularisation, si les conditions sont vérifiés, l’utilisateur mis a jour le statut de l’enseignant, puis il édite la décision et l’imprime. Le diagramme suivant (figure III.8) montre ce scénario

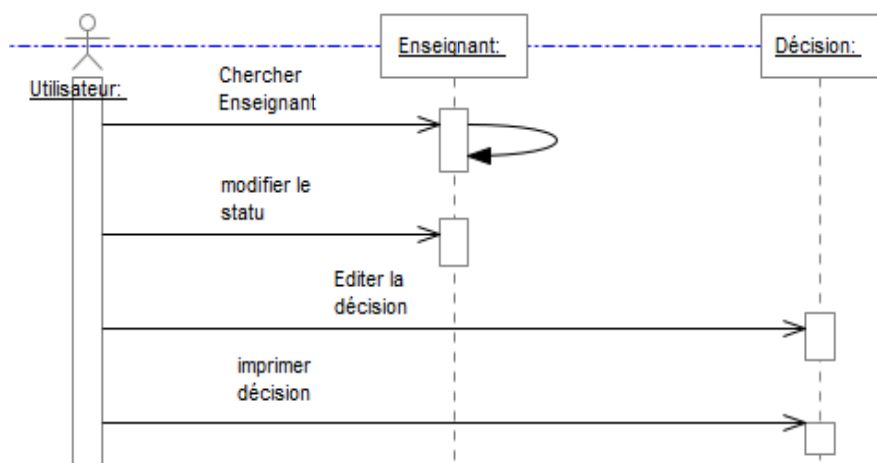


Figure III.8 : diagramme de séquence « titularisation »

## f) Gestion de congé :

Suite à la demande d'un congé par l'enseignant, l'utilisateur doit d'abord chercher l'enseignant et sélectionner le type de congé, puis il édite la décision et lance son impression.

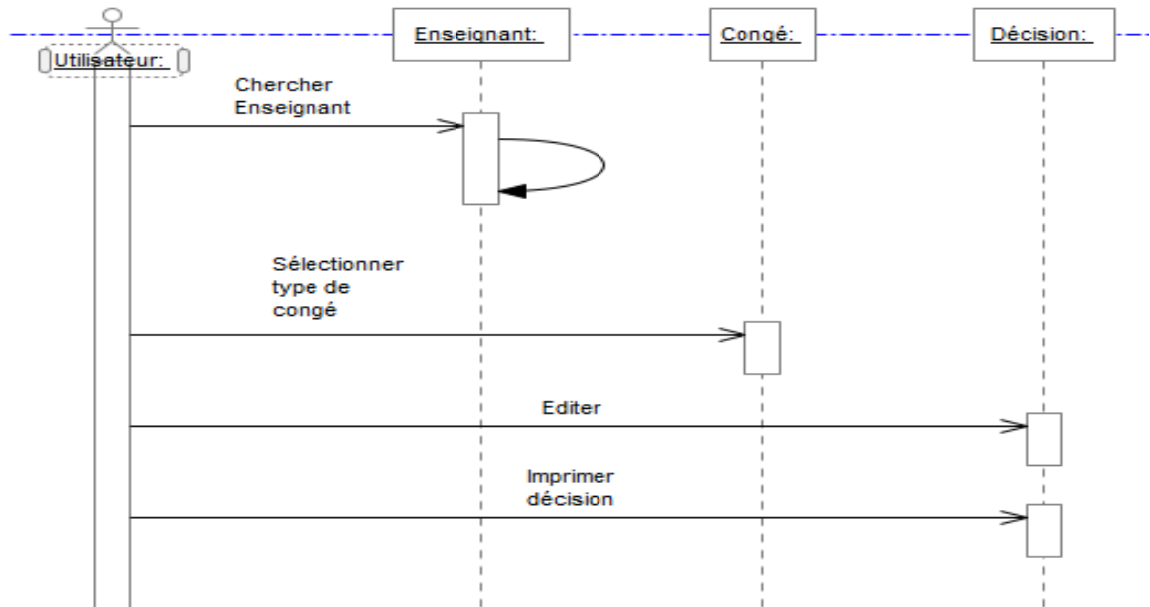


Figure III.9 : diagramme de séquence « gestion de congé »

## g) Impression de l'attestation de travail :

Suite à la demande de l'enseignant d'une attestation de travail, l'utilisateur cherche l'enseignant, puis il lance l'impression de l'attestation.

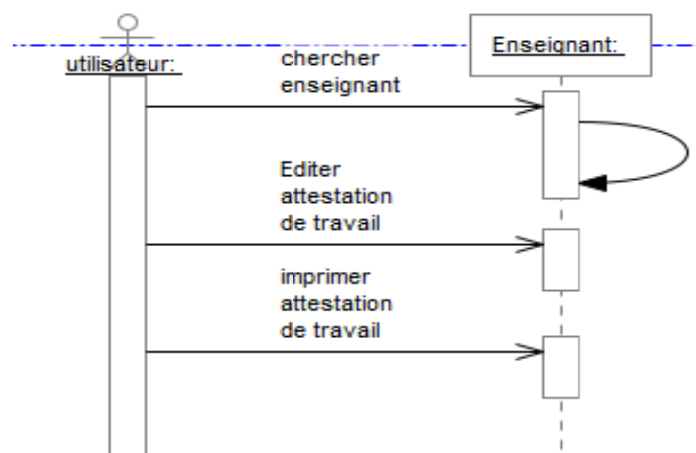


Figure III.10 : diagramme de séquence « impression attestation de travail »

## h) Attribution Fonction :

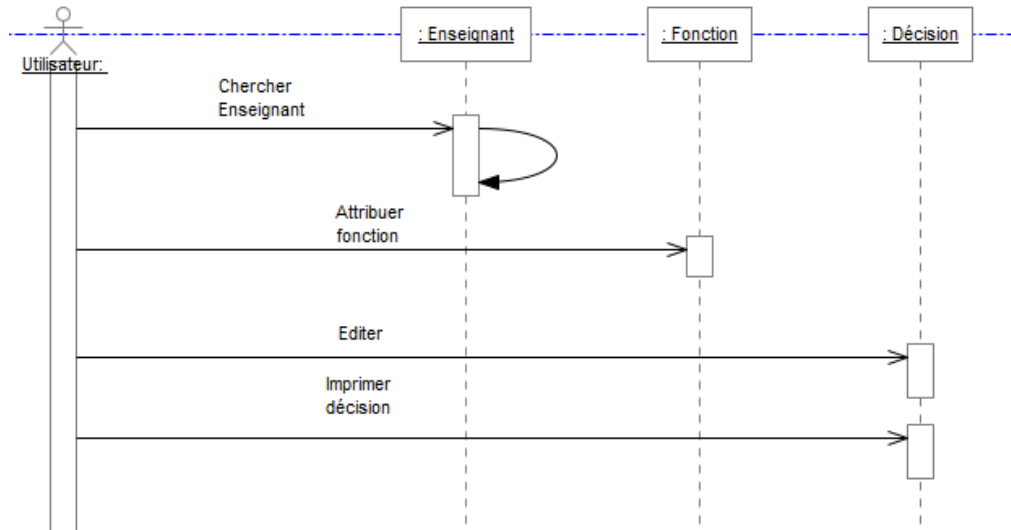


Figure III.11 : diagramme de séquence « attribution fonction »

### III.3. Diagramme de classe :

Le diagramme de classes identifie les classes de notre système et les associations entre elles.

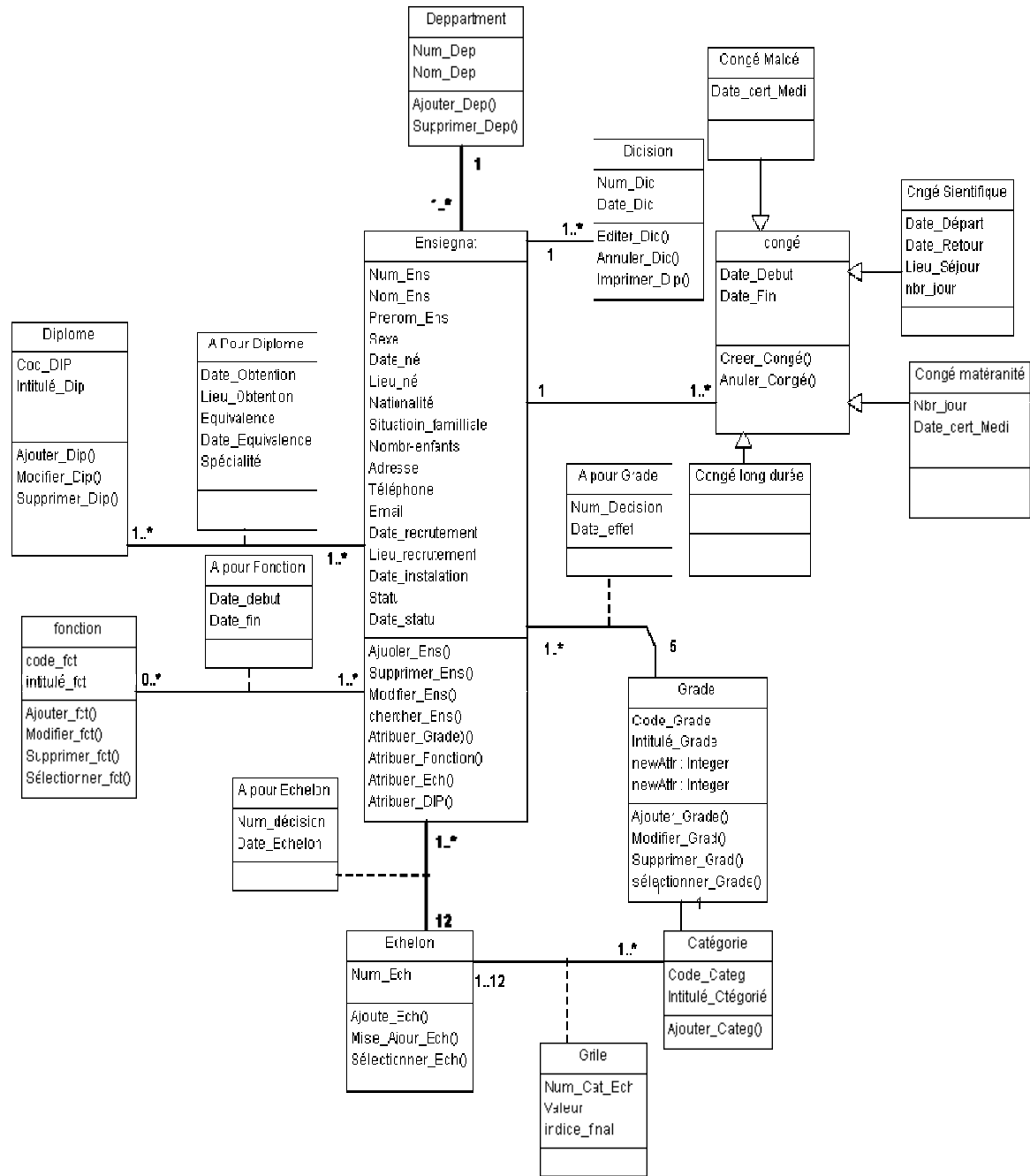


Figure III.12 : diagramme de classe

**IV. Modèle logique de données :**

Le modèle logique des données consiste à décrire la structure de données utilisée sans faire référence à un langage de programmation. Il s'agit donc de préciser le type de données utilisées lors des traitements.

Le modèle logique présenté ci-dessous a été obtenu en application des règles de passage [7] sur le diagramme de classe de la figure précédente.

**Enseignant** (Num\_ens, Nom\_ens, Prénom\_ens, Sexe, Date\_né, Lieu\_né, Nationalité, Situation\_Familiale, Nombre\_enfant, Adresse, Téléphone, Email, Date\_Recrutement, lieu\_Recrutement, Date\_Installation, statu, Date\_statu, code\_Dép\*);

**Département** (code\_Dép, intitulé\_Dép);

**Diplôme** (code\_Dip, intitulé\_Dip);

**A pour Diplôme** (Num\_ens, code\_Dip, Date\_Obtention, Lieu\_Obtention, Equivalence, Date\_Equivalence, Spécialité);

**Fonction** (code\_Fct, Intitulé\_Fct);

**A pour Fonction** (Num\_ens, code\_Fct, Date\_debut, Date\_fin);

**Grade** (Code\_Grade, Intitulé\_Grade);

**A pour Grade** (Num\_ens, code\_grade, Date\_effet, Num\_décision);

**A pour Echelon** (Num\_ens, Num\_Ech, Date\_effet, Num\_décision);

**Catégories** (code\_Categ, intitulé\_categ);

**Grille** (code\_catég, Num\_Ech, valeur, indice\_final);

**Congé** (Date\_Debut, Date\_Fin, Num\_décision, type, Num\_ens\*);

**Congé\_Scientifique** (Date\_Debut, Date\_Fin, Lieu\_Séjour, Num\_décision, Num\_ens\*);

**Congé\_Matérnité** (Date\_Debut, Date\_Fin, Date\_FM, Durée, Num\_décision, Num\_ens\*);

**Congé\_Maladé** (Date\_Début, Date\_Fin, Date\_FM, Durée, Num\_décision, Num\_ens\*);

**Congé\_Long\_Durée** ( Date\_Début, Date\_Fin, Num\_décision Num\_ens\* ) ;

**Décision** (Num\_Déc, Date\_Déc, type, Num\_Ens\* ) ;

### V. Conclusion :

Dans ce chapitre nous avons présenté la modélisation de la structure statique et dynamique de notre système en utilisant un sous ensemble des diagrammes d'**UML** et en appliquant une partie du processus unifié comme démarche de conception et de modélisation. Cette conception est une étape nécessaire et très importante pour pouvoir créer notre base de données et réaliser l'application, le prochain chapitre détaillera les étapes de l'implémentation de notre application.

## Chapitre IV : Implémentation du système

**I. Introduction :**

Ce chapitre est consacré à l'implémentation de notre application qui s'appuie sur la modélisation présentée dans le chapitre précédent, pour l'implémentation nous avons utilisé Delphi version 5.

**II. L'environnement de programmation Delphi [8]:**

Delphi est un environnement de programmation visuel orienté objet pour le développement rapide d'application (**RAD**). Il incarne la suite logique de la famille turbo pascal. C'est un outil moderne qui fait appel à une conception visuelle des applications. De plus, il prend en charge le maintien automatique d'une partie du code source.

Voici quelques unes des caractéristiques de Delphi :

- Il Supporte le langage Pascal orienté objet.
- Utilisation visuelle bidirectionnelle.
- Compilateur produisant du natif.
- Traitement complet des exceptions.
- Possibilité de créer des programmes exécutables et des DLL.
- Bibliothèque des composants extensibles.
- Déboguer graphique intégré.
- Support de tous les API de Windows : OLE2, DDE, VBX, OCX...

**III. L'environnement de gestion de base de données :**

Nous avons utilisé le module de base données version 5 (BDE) de Delphi pour la création de différentes tables de notre base de données.

Ce module permet de créer une base de données dans différents formats, tel que les base de données Accès, Oracle, Sybase, c'est un environnement graphique qui permet la création, la modification et la suppression des tables, ainsi que la définition des requêtes SQL permettant de retourner différent type de résultat à partir des tables.



#### IV. Description de l'application :

Nous allons présenter dans cette partie les principales fiches de l'application.

##### ➤ Fiche d'authentification :

Cette fiche permet à utilisateur de s'authentifier pour pouvoir accéder aux autres interfaces du système :



Figure IV.1: Fiche d'authentification

##### ➤ Fiche Principale

C'est la fiche qui s'affiche lorsque l'utilisateur s'est authentifié correctement. A partir de cette fenêtre l'utilisateur peut accéder aux autres fiches de l'application en utilisant le menu.



Figure IV.2: Fiche principale

➤ **Fiche d'enseignant :**

Cette fiche permet d'afficher toutes les informations d'un enseignant. A partir de cette fiche l'utilisateur peut réaliser les différents scénarios cités dans le chapitre précédent, il peut ajouter, supprimer ou modifier un enseignant comme il peut attribuer le nouveau grade, échelon, fonction, diplôme ou congé en utilisant le bouton correspondant dans cette fiche. Il peut aussi faire la recherche d'enseignant par son nom.

The screenshot shows a web application window titled 'Enseignant'. At the top, there are navigation buttons: 'Avancement', 'Promotion', 'Fonction', 'Diplome', 'Congé', 'Imprimer attestation de travail', and 'Fermer'. The main area contains a form with the following fields:

- Département: Inf (dropdown)
- Numéro d'Enseignant: 2 (text)
- Nom: Benmmamar (text)
- Nationalité: algérienne (text)
- Prénom: Badr (text)
- Adresse: Tlemcen (text)
- Date de naissance: 02/05/1980 (text)
- Téléphone: 778164589 (text)
- Lieu de naissance: Tlemcen (text)
- Email: Benamar@13.con (text)
- sexe: Masculin (dropdown)
- Date recrutement: 07/04/1998 (text)
- Situation familial: celibatair (dropdown)
- Date installation: 22/07/1998 (text)
- Nbr d'Enfant: (empty text)
- Statu: titulaire (dropdown)
- Date Titularisation: 12/02/2011 (text)

On the right side, there are action buttons: 'Ajouter', 'Modifier', 'Supprimer', 'Enregistrer', and 'Annuler'. Below these is a search box labeled 'Recherche par Nom'. On the left side, there are navigation buttons: 'Premier', 'Suivant', 'Précédent', and 'Dernier'. At the bottom, there is a table with the following data:

Num_ens	Nom_ens	Prenom_ens	Date_nai	Lieu_nai
2	Benmmamar	Badr	02/05/1980	Tlemcen
3	Chikhawi	Hadjar	01/01/1989	Tmouchent
45	mouffok	benattou	10/01/1976	sba

*Figure IV.3: Fiche d'enseignant*

On peut prendre comme exemples :

➤ **La fiche fonction :**

On peut accéder a cette fiche a partir de la fiche enseignant, cette fiche permet a l'utilisateur de géré des fonctions occuper par un enseignant.

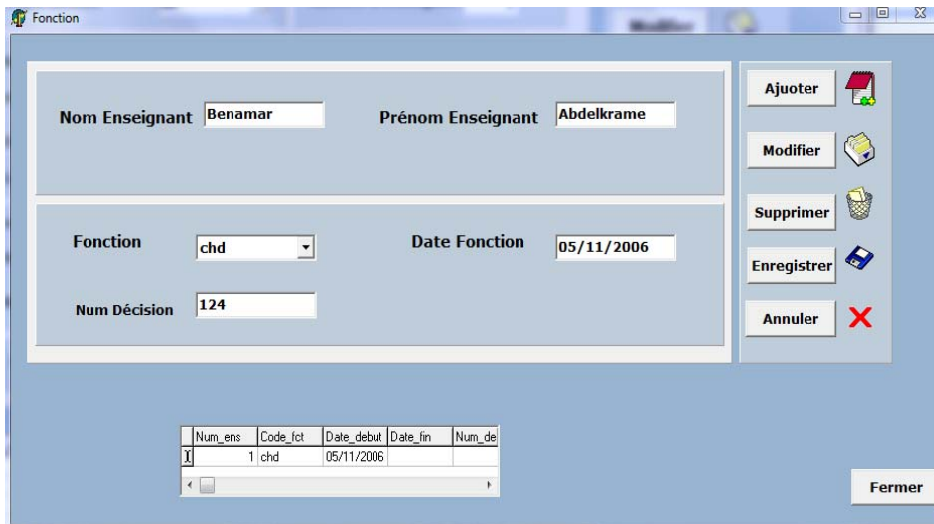


Figure IV.4 fiche fonction

➤ **Fiche congé :**

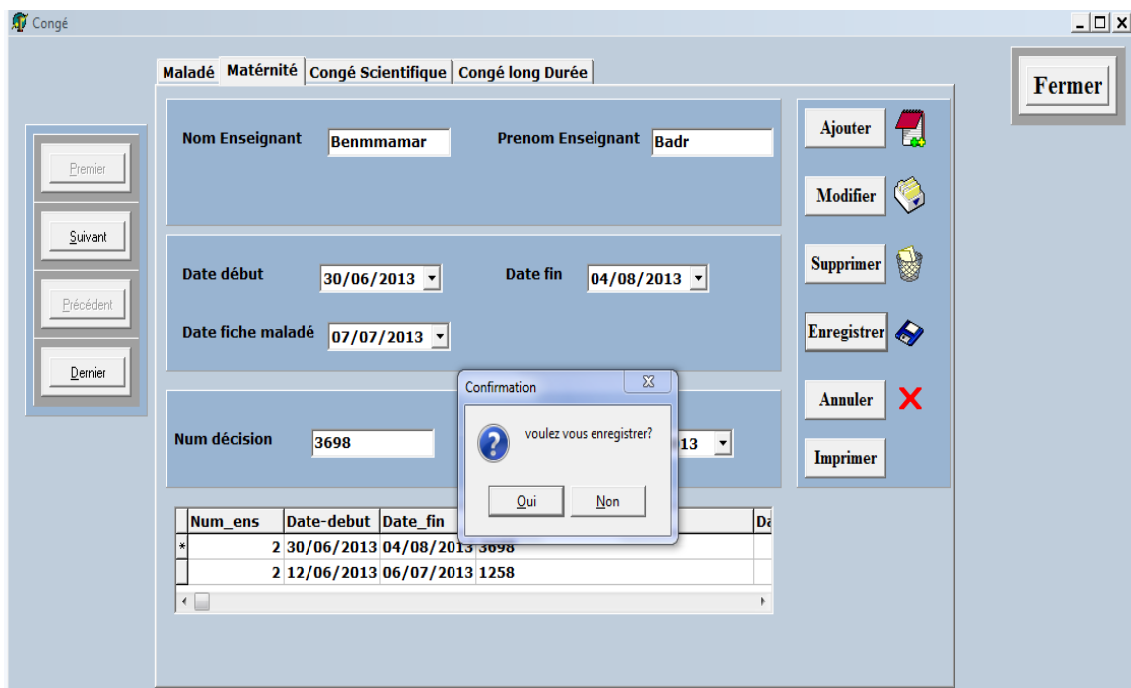


Figure IV.5 fiche de congé.

➤ **Fiche attestation de travail :**

The screenshot shows a window titled 'Attestation'. On the left, there is a vertical stack of four buttons: 'Premier', 'Suivant', 'Précédent', and 'Dernier'. The main area contains a form with the following fields:

- Noml Enseignant: Benmmamar
- Prénom Enseignant: Badr
- Num Attestation: 1452
- Date Attestation: 2013

On the right side of the form, there are three buttons: 'Fermer' (top right), 'Imprimer' (with a printer icon), and 'Annuler' (with a red 'X' icon).

Below the form is a table with the following data:

	Num_attes	Num_ens	Date_attest
*	2	1452	2013

*Figure IV.6 fiche attestation de travail.*

➤ **Fiche affichage des tables :**

A partir de cette fiche on peut mettre à jour les tables (département, diplôme, gade, fonction,...).

### ✚ Fiche diplôme :

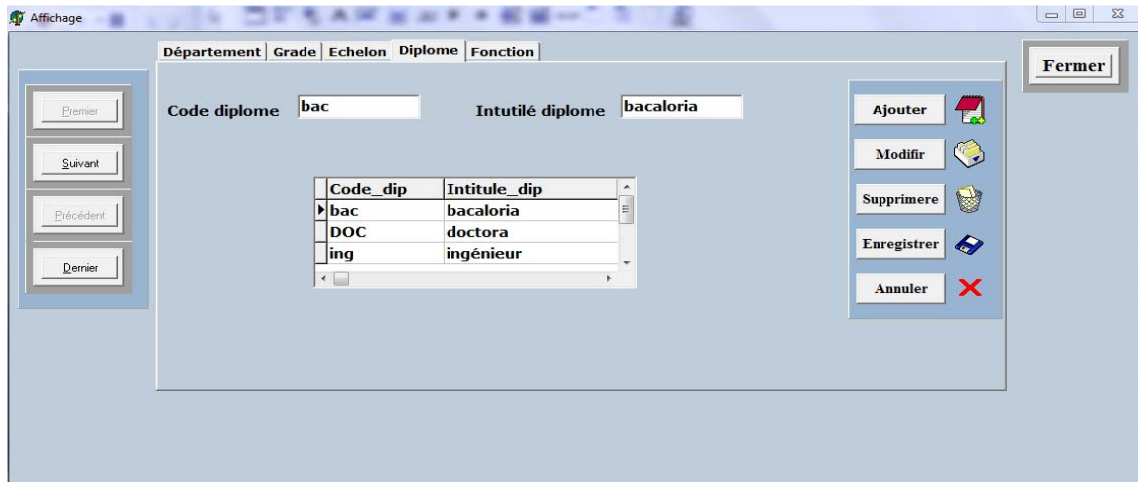


Figure IV.6 fiche diplôme.

### ✚ Fiche grade :

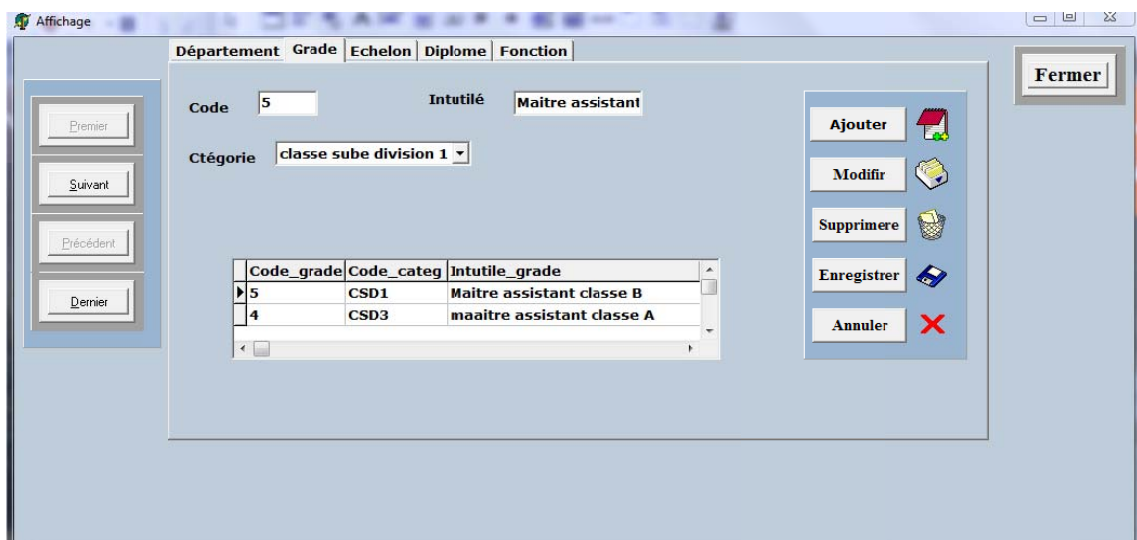


Figure IV.7 fiche Grade.

## V. Conclusion :

Dans ce dernier chapitre, nous avons présenté la partie réalisation de notre projet, et nous avons décrit les fiches les plus importantes de notre application.

Rapport-Gratuit.com

## Conclusion Générale

---

### Conclusion général :

Le travail qui nous a été confié consiste à réaliser le système de gestion personnel d'enseignant. L'analyse, la spécification, la conception sont faites orienté objet et décrites en UML en se basant sur le processus UP.

L'implémentation de l'application a été réalisé par l'environnement Delphi, notre application a pour bute principal de facilité les taches a chef service (utilisateur de système).

Pendant la réalisation de notre travail nous avons acquis beaucoup de connaissances concernant le langage de modélisation UML et l'environnement Delphi.

Nous avons souhaité d'avoir plus de temps pour mieux traiter le sujet proposé.

Mais nous espérons que notre travail sera évolué et amélioré par autres promotion et qu'il sera un aide pour eux.



## Références bibliographiques

[1] [www.université abou bakr belkaid](http://www.université.abou-bakr-belkaid.dz)

[2] [www.JORADP.dz](http://www.JORADP.dz)

[3] Pascal Roques et Frank vallée, « UML En Action », Eyrolles 2002

[4] Yohann Richarde, Modélisation avec UML, Février 2000

[5] Pièce-Alain Muller et Nathalie Gaertner, « Modélisation avec UML », Presses de l'Université du Québec 2009

[6] Modelmaker 11.0 user Manuel Pascal and C# edition

[7] Grille Poy, "conception de base de données avec UML", Presses de l'Université du Québec 2009

[8] Mc.BELIAD :Delphi 5 et 6 programmation des Base de données

## Résumé :

Notre projet de fin d'étude consiste à la conception et la réalisation d'un système permettant d'optimiser la gestion du personnel enseignant.

Pour la modélisation du système on a utilisé le langage **UML**, en s'appuyant sur le processus unifié (**UP**).

Pour l'implémentation de l'application, nous avons utilisé l'environnement de développement orienté objet **Delphi**.

## Summary

Our project of end of study consists with the design and the realization of a system making it possible to optimize the teaching staff management.

For the modeling of the system one used language **UML**, while being based on the unified process (**UP**).

For the implementation of the application, we used the environment of development directed **Delphi** object.

## الملخص

يتركز مشروعنا على تصميم نظام يسهل عمل هيئة التدريس.

برمجة هذا النظام تمت من خلال دراسة تفصيلية عن طريق استعمال UML و عملية UP.

تم تطبيق و تنفيذ هذا البرنامج باستعمال البرمجة Delphi .