

**GESTON DES PERSONNELS**  
**GOUVERNORAT DU KAIROUAN**

**Par : ZIED TOUNSI**

**PROJET DE FIN D'ETUDE**

**Licence Appliquée**

**En**

**SCIENCES ET TECHNIQUES DE L'INFORMATIONS ET DE COMMUNICATIONS**

**UNIVERSITE VIRTUELLE DE TUNIS**

**Encadreur : M. Riadh Bouhouchi**

**Année Universitaire : 2010/2011**



# *DÉDICACES*

*A ma plus grande source de fierté : mon père Hassen*

*A la lueur qui éclaire ma vie : ma mère Fatma*

*Pour toute la tendresse avec laquelle ils m'ont entouré et tous les sacrifices qu'ils ont fait pour moi*

*A mes chères et bien aimées frères et sœur : Wissem, Walid, Marwen, Chokri, Yamen, Riadh, Belgacem, Marwa, Sahar*

*A ma chère et fidèle amie Rahma*

*A tous mes collègues de la Division des affaires sociales*

*A tous les enseignants auxquels je dois ce que je suis maintenant*

*A tous mes amis à l'UVT et qui me sait*

*Je dédie ce travail.*

*Tounsi Zied*

# ***REMERCIEMENTS***

*Je souhaite d'abord exprimer ma vive reconnaissance pour mon encadreur Monsieur Riadh Bouhouchi qui m'a toujours guidé sans jamais me contraindre, Je le remercie très sincèrement pour m'avoir donné l'opportunité de mener à bien ce travail par sa disponibilité et son soutien, pour le temps qu'il m'a consacré et ses nombreux et précieux conseils, et je le remercie surtout pour m'avoir soutenu dans tous le sens du terme, depuis la première séance du cours que j'ai eu la chance d'assister avec lui.*

*Je tiens à exprimer de même ma gratitude à mon encadreur du Gouvernorat de Kairouan Monsieur Chokri Mahfoudhi Qui a cru en moi et qui n'a cessé de me faire profiter ses précieux conseils et ses remarques. Et à tout le personnel de la Division des affaires sociales en particulier Madame Mariem.*

# TABLE DES MATIÈRES

Introduction Générale .....	10
I.Préface .....	11
II.Présentation de L'organisme d'accueil .....	11
II.1. Création : .....	11
II.2. Les divisions relevant au gouvernorat : .....	12
II.2.1. Les missions du gouverneur comprennent : .....	12
II.2.2. Les délégués de gouvernorat : .....	12
II.2.3. Présentation du Division des affaires administratives générales : .....	12
II.3. L'organigramme : .....	13
III.Plan du rapport.....	15
Chapitre1: Spécification et analyse des besoins .....	16
I.Introduction .....	17
II.Etude de l'existant.....	17
II.1.Problématique: .....	17
II.2.Solution Proposée : .....	17
III.Spécification de besoins .....	17
III.1.Besoins fonctionnels : .....	17
III.2.Besoins non fonctionnels : .....	18
IV.Cahier des charges.....	18
Chapitre 2 : La Conception.....	19
I.Introduction .....	20
I.1. Introduction d'UML : .....	20
II.Conception générale.....	21
II.1. Le choix du cycle de vie : .....	21
II.2. Choix de la méthode de conception : .....	21

III.Conception détaillée .....	22
III.1. Méthodologie de travail : .....	22
III.2. Diagramme de cas d'utilisation : .....	22
III .2.1. Identification des acteurs : .....	22
III.2.2. Identification des cas d'utilisation : .....	23
III.2.3. Description textuelle des cas d'utilisation : .....	23
III.3. Diagramme d'activité : .....	29
III.3.1. Processus d'authentification : .....	29
III.3.2. Processus d'ajout d'un personnel : .....	31
III.3.3. Processus de modification d'un personnel : .....	32
III.3.4. Processus de suppression d'un personnel : .....	33
III.4. Diagramme de séquence : .....	34
III.4.1. Digramme de séquence de modèle de conception : « s'identifier » .....	34
III.4.2. Digramme de séquence de modèle de conception « Consulter» .....	35
III.4.3. Digramme de séquence de modèle de conception : « Ajouter personnel » .....	36
III.4.4. Diagramme de séquence de modèle de conception «Modifier Personnel » .....	37
III.4.5. Diagramme de séquence de modèle de conception «Supprimer Personnel » .....	38
III.4.6. Diagramme de séquence de modèle de conception «Ajouter Administrateur » .....	39
III.4.7. Diagramme de séquence de modèle de conception «Modifier Administrateur » .....	40
III.4.8. Diagramme de séquence de modèle de conception «Supprimer Administrateur » .....	41
III.4.9. Diagramme de séquence de modèle de conception « Ajouter Prime » .....	42
III.4.10. Diagramme de séquence de modèle de conception « Associer Prime » .....	43
III.5. Digramme de classe: .....	43
III.6. Digramme de package : .....	45
Chapitre 3 : Réalisation .....	46
I.Introduction : .....	47
II.Environnement matériel : .....	47
III.Environnement Logiciel : .....	47

IV.Réalisation :	50
IV.Chronogramme de déroulement de stage	54
V.Conclusion	55
Conclusion Générale	56
Conclusion Générale et perspective	57
Neto-graphie	59
Annexes	60

# TABLE DES FIGURES

Figure 1 : Organigramme du Gouvernorat de Kairouan.....	14
Figure 2 : cycle de vie en cascade .....	21
Figure 3 : Digramme de cas d'utilisation .....	28
Figure 4: Digramme d'activités authentications .....	30
Figure 5: Digramme d'activités d'ajouter un personnel .....	31
Figure 6: Digramme de modification d'un personnel .....	32
Figure 7: Digramme d'activités de suppression d'un personnel .....	33
Figure 8: Digramme de séquence de cas d'utilisation « s'identifier » .....	34
Figure 9: Diagramme de séquence de cas d'utilisation « Consulter ».....	35
Figure 10 : Digramme de séquence de cas d'utilisation « Ajouter Personnel ».....	36
Figure 11 : Digramme de séquence de cas d'utilisation « Modifier Personnel ».....	37
Figure 12 : Digramme de séquence de cas d'utilisation « Supprimer Personnel » .....	38
Figure 13 : Digramme de séquence de cas d'utilisation « Ajouter Administrateur » .....	39
Figure 14 : Diagramme de séquence de cas d'utilisation « Modifier Administrateur ».....	40
Figure 15 : Diagramme de séquence de modèle de conception .....	41
Figure 16 : Diagramme de séquence de cas d'utilisation « Ajouter Prime » .....	42
Figure 17 : Diagramme de séquence de cas d'utilisation « Associer prime ».....	43
Figure 18 : Diagramme de classe .....	44
Figure 19 : Diagramme de package .....	45
Figure 20: Interface d'authentification. ....	50
Figure 21 : Interface d'Accueil.....	51
Figure22: Fenêtre d'ajout d'un nouveau personnel .....	52
Figure 23 : Interface d'ajout d'un nouvel administrateur .....	53
Figure 24 : Interface des gestions des primes .....	54
Figure 25 : Planning prévisionnel du projet .....	55



# TABLE DES TABLEAUX

Tableau 1 : Identification des acteurs .....	22
Tableau 2 : Identification des acteurs par cas d'utilisation .....	23
Tableau 3 : Description de scénario « Accéder au système ».....	23
Tableau 4 : Description de scénario « Consulter Personnel » .....	24
Tableau 5 : Description de scénario « Ajouter Personnel ».....	24
Tableau 6 : Description de scénario « Modifier Personnel ».....	25
Tableau 7 : Description de scénario « Supprimer personnel » .....	25
Tableau 8: Description de scénario « Ajouter Administrateur » .....	26
Tableau 9 : Description de scénario « Modifier Administrateur » .....	26
Tableau 10 : Description de scénario « Supprimer Administrateur » .....	27
Tableau 11 : Description de scénario « Ajouter prime » .....	27
Tableau 12 : Description de scénario « Associer prime » .....	28

---

# *Introduction Générale*

## I.Préface

Face à l'individualisation de plus en plus poussée de la gestion du personnel, les administrations doivent trouver les outils nécessaires pour la gestion des personnels qui s'automatise pour optimiser les ressources humaines. Dans ce but, les administrations s'équipent du logiciel permettant une gestion fine des personnels.

Le contrôle des employés est une problématique classique du monde de l'emploi, il est progressivement adopté par toutes les administrations, et particulièrement par l'organisme du Gouvernorat qui comporte plus d'une centaine de salariés. Mieux encore, la mise en œuvre des applications de gestion des employés.

Dans ce cadre la Division des affaires administratives générales dans le Gouvernorat de Kairouan m'a proposé de développer une application de gestion de personnel qui permet de gérer les agents de ce gouvernorat qui contient plus que 180 salariés.

Le système de gestion du personnel est une solution plus au moins adoptée à ce problème car cette application va nous permettre de gérer les différentes catégories d'employés (ouvriers, cadre,...) ainsi que leurs salaires.

## II.Présentation de L'organisme d'accueil

### II.1. Création :

Le Gouvernorat de Kairouan a été créé selon le décret du 21 juin 1956 portant sur l'organisation administrative de la république tunisienne.

Le gouvernorat est :

- Situé au centre ouest de la Tunisie
- Leur surface est de 6712 km<sup>2</sup>
- Leur population est de 561,890
- comporte 11 délégations et 12 municipalités

## **II.2. Les divisions relevant au gouvernorat :**

Le gouvernorat dans l'organisation administrative du territoire de la République Tunisienne occupe une double fonction, il est considéré d'une part comme une autorité politique puisqu'il présente le président de la république dans le gouvernorat, et d'autre part, il occupe un emploi professionnel car il est directeur du conseil régional.

### ***II.2.1. Les missions du gouverneur comprennent :***

- L'exécution des lois et des règlements.
- L'exécution de la stratégie de l'Etat dans le gouvernorat.
- L'exécution des mesures de sureté générale.
- Informer les pouvoirs de tout le changement dans le gouvernorat.
- Assurer la bonne marche des services administratifs et technique de l'arrondissement et de leur contrôle.
- Veiller à l'exécution du programme annuel de la propreté et de protection de l'environnement.
- Organiser des manifestations culturelles de la jeunesse.
- Suivre les collectivités et les conseils nationaux

### ***II.2.2. Les délégués de gouvernorat :***

Le décret n°90-1234 du 1 aout fixant le nombre de délégués au siège de gouvernorat en trois :

- Un délégué qui s'occupe des affaires économiques.
- Un délégué qui s'occupe des affaires sociales.
- Un délégué qui s'occupe des affaires religieuses.

### ***II.2.3. Présentation du Division des affaires administratives générales :***

La division des affaires administratives et du personnel a pour mission d'assurer une gestion rationnelle des ressources humaines, financières et matérielles en rapport avec l'autre division intérieure et délégations extérieures.

A ce titre elle est chargée :

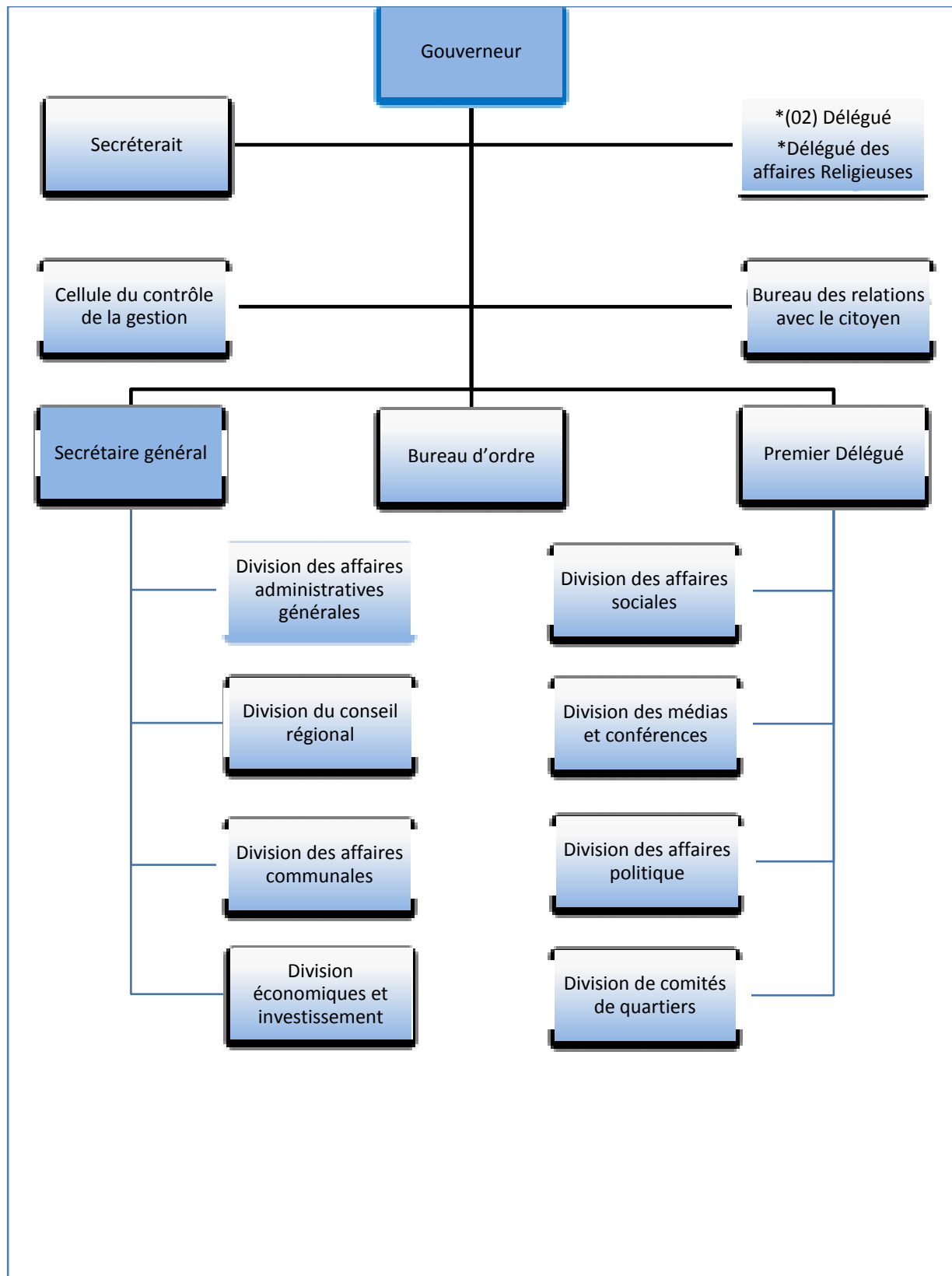
- de la planification des besoins en personnel cadres et agents,
- de la préparation de tous les actes liés à la gestion administrative.
- De la tenue et de la mise à jour des dossiers et des fiches individuelles du personnel,
- De la gestion du matériel et des fournitures des bureaux de la division.

- De la gestion des crédits du déplacement et du transport, des imprimés généraux, l'eau, l'électricité, le téléphone, le télex, le fax etc...

### **II.3. L'organigramme :**

L'organigramme du gouvernorat a été assujetti au décret n° 93-1476 du 9 juillet 1993 portant l'organisation des services du gouvernorat et de délégation.

Ce décret a divisé le gouvernorat selon des Divisions relevant au gouverneur, des Divisions relevant au premier délégué et des Divisions relevant au secrétaire générale.



**Figure 1 : Organigramme du Gouvernorat de Kairouan**

### III. Plan du rapport

Ce rapport est structuré en quatre chapitres :

- Le premier chapitre comporte une description de l'état de l'art des différents thèmes abordés par le sujet à savoir la gestion des personnels du Gouvernorat de Kairouan et à l'étude de l'existant.
- Le deuxième chapitre vise à présenter une spécification détaillée du projet en relevant les différentes fonctionnalités et besoins de chacun de ses composants.
- Le troisième chapitre est consacré à la conception ou est décrit l'ensemble des diagrammes UML relatifs à l'application de notre projet.
- Enfin, le chapitre réalisation est consacré à l'exposé de la solution implémentée ainsi que l'environnement logiciel et matériel utilisé.

Je termine par une conclusion générale par laquelle je dresse un bilan du travail effectué ainsi que les problèmes restant ouverts.

# *Chapitre **1** : Spécification et analyse des besoins*

---



## I.Introduction

Avant d'entamer le projet, il est nécessaire de bien étudier les besoins de l'application, de déterminer les problèmes et les potentiels de la solution projetée tout en prenant compte le point de vue conceptuel, organisationnel, technique, etc. et par la suite trouver les solutions adaptables pour le travail demandé.

Pour ce chapitre, nous commençons par une étude de l'existant, nous spécifions les besoins fonctionnels et non fonctionnels.

## II.Etude de l'existant

### II.1.Problématique:

Dans la vie professionnelle la gestion du personnel pose un grand problème que ce soit pour les organismes ou pour les employés car l'opération de gérer les personnels du calcul du salaire était auparavant une opération manuelle qui s'effectue sur papier ou bien sur simple fichier Excel. Ce qui cause des problèmes de différents types.

### II.2.Solution Proposée :

Notre application a pour objectif de :

- Gérer les ressources humaines existant au sein du Gouvernorat :
  - Ajouter un nouveau personnel
  - Modifier les données concernant un employé existant
  - Supprimer un employé de notre application
  - Consulter les champs du personnel demandé
- Calculer le salaire de chaque employé. En comptant les primes associées à chaque personnel

## III.Spécification de besoins

### III.1.Besoins fonctionnels :

Ces besoins fonctionnels sont directement liés aux tâches à réaliser, et ils doivent être transparents le plus que possible face aux utilisateurs.

Notre application devrait supporter les fonctionnalités techniques suivantes :

- Possibilité de sauvegarde, de restauration, et d'archivage de la base des données
- Contrôler en temps réel les différents mouvements
- Traitement des anomalies du mouvement au moment de calcul du nombre d'heures de travail
- les entrées/sorties du personnel doivent être éditées sur imprimante (intitulé du mouvement ainsi que les dates et les heures)

### **III.2.Besoins non fonctionnels :**

Le logiciel du système de gestion personnel doit être compatible avec plusieurs types de machines. En outre, le système doit être bien modulé et surtout facile à utiliser.

#### **Efficacité**

Pour marchander un tel produit logiciel, il faut qu'il supporte plusieurs milieux tels que les réseaux d'entreprise et les machines sur lesquelles il s'exécute, il doit aussi aux exigences suivantes:

- logiciel Multi plateforme centralisé utilisant les systèmes (LINUX, Windows...)

#### **Sécurité**

Le système doit être sécurisé pour éviter toute tentative d'utilisation non appropriée (consultations et modifications non permises), et d'attaques pouvant engendrer des anomalies dans le fonctionnement.

- offre une sécurité totale au niveau de la gestion des accès des données.

#### **Stabilité**

L'application doit être stable dans son environnement d'exécution et ne doit pas contenir des bogues bloquants).

#### **Durée**

Ce projet a débuté le 1 mars 2011 et s'est étale sur une période de 4 mois. La réalisation des différentes parties de ce projet ont été réparties selon le degré de complexité chacune.

#### **La rapidité de traitement**

En effet, vu le nombre important des transactions quotidiennes, il est impérativement nécessaire que la durée d'exécution des traitements s'approche le plus possible du temps réel.

## **IV.Cahier des charges**

Dans ce projet, nous nous intéressons à l'élaboration d'une application permettant aux cadres du Gouvernorat de Kairouan la gestion de personnel associer avec la prime :

Cette application doit garantir :

- La gestion et le suivi des personnels du Gouvernorat du Kairouan
- Ajouter et associer la prime de chaque personnel
- Impression du document nécessaire pour le personnel par exemple Attestation de travail, Bulletin de paie.

## *Chapitre 2 : La Conception*

---

## I.Introduction

L'étape de conception constitue un pont entre la définition et l'implémentation. Elle permet de consolider les ébauches que nous avons recensé dans le chapitre précédent, et prend en vue la présentation des divers aspects de fonctionnalités de notre application. La conception décrit le fonctionnement de notre système par une étude générale basée sur la spécification des besoins en premier lieu, puis une étude détaillée du projet sur plusieurs approches.

Dans ce chapitre, nous présentons la conception des différents éléments du système. Nous utilisons pour cela :

- UML comme langage de modélisation, qui nous permettra d'analyser et de concevoir notre projet à la fois d'une manière statique (dans notre cas on utilisera le diagramme de classes et diagramme de déploiement) et dynamique (diagramme de cas d'utilisation et diagramme de séquence).
- Pascestar UML Diagramme comme outil de conception.

### I.1. Introduction d'UML :

Il existe des méthodes de conception orientées objet autre qu'UML : comme Booch et OMT (Object Modeling Technique). Etant donné que ces méthodes se développaient déjà indépendamment l'une de l'autre et étaient mondialement reconnues pour une période assez importante comme les principales méthodes orientées objet, Booch (le concepteur de la méthode qui a porté son nom) et Rumbaugh (l'inventeur de l'OMT) ont joint leurs forces pour réaliser une unification complète de leurs méthodes.

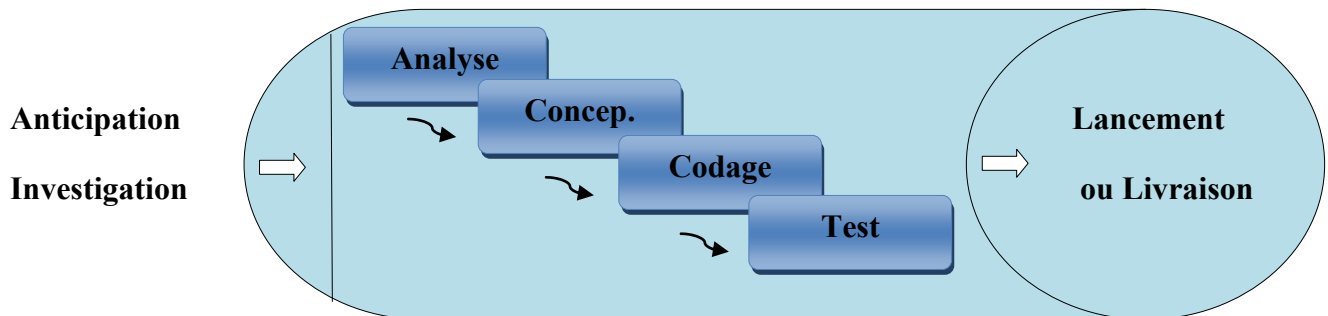
Unified Modeling Language (UML) n'est pas un éloignement radical des méthodes Booch, OMT, ou OOSE, mais plutôt le successeur légitime de ceux-ci. UML est plus expressif, plus propre et plus uniforme que toutes les autres méthodes. Cela signifie qu'il y a un bénéfice à modéliser avec UML, parce que cette méthode permettra aux projets de modéliser des choses qui n'auraient pas pu l'être d'auparavant. Les utilisations de la plupart des autres méthodes et langages de modélisation de notions et de terminologie qui obscurcissent les similarités de bases ces différentes approches, donc nous aurons une conception universelle et efficace.

## II. Conception générale

### II.1. Le choix du cycle de vie :

Pour chaque produit (ou application) conçu et développé on choisit une démarche pour le suivre tout au long du projet que l'on appelle cycle de vie du produit. Il existe plusieurs types de cycles de vie (cycle de vie en V, en cascade,.....).

Pour notre projet on a choisi de suivre le cycle de vie en cascade :



**Figure 2 : cycle de vie en cascade**

Le modèle de cycle de vie en cascade a été mis au point dès 1966, puis formalisé aux alentours de 1970. Dans ce modèle le principe est très simple : chaque phase se termine à une date précise par la production de certains documents ou logiciels. Les résultats sont définis sur la base des interactions entre étapes, ils sont soumis à une revue approfondie et on ne passe à la phase suivante que s'ils sont jugés satisfaisants.

### II.2. Choix de la méthode de conception :

Pour concevoir une application informatique il y a plusieurs méthodes de conception par exemple l'approche objet. Pour la conception de notre application, nous avons choisi l'approche objet. En effet, cette approche présente plusieurs avantages, à savoir :

- Le système développé est plus facile à maintenir du fait que les objets sont indépendants ils peuvent donc être modifiés. Mais, le fait de modifier l'implémentation d'un objet ou de lui ajouter des services ne doit pas affecter les autres objets du système.
- Les objets sont considérés comme des composants réutilisables appropriés vu leur indépendance. On peut alors développer des conceptions à l'aide des objets créés dans une autre conception.
- L'analyse orientée objet permet d'examiner un problème en mettant en évidence les classes et les objets correspondants sous forme de composants indépendants qui interagissent selon des modalités bien définies.

## III. Conception détaillée

### III.1. Méthodologie de travail :

Les avantages du langage UML :

- décomposer le processus de développement
- mettre en relation les experts métiers et les analystes
- séparer l'analyse de la réalisation
- migrer facilement vers une architecture objet d'un point de vue statique et dynamique.

### III.2. Diagramme de cas d'utilisation :

Pour administrer la base, l'administrateur doit tout d'abord s'identifier (saisir son login et mot de passe), puis le système vérifie ces paramètres, ensuite le système lui attribut un ensemble de privilèges, enfin l'administrateur peut agir avec la base selon ses droits donnés.

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés (actor), ils interagissent avec les cas d'utilisation (use case).

#### III.2.1. Identification des acteurs :

Modélisation métier	Administrer système
Acteur	Type
Administrateur	Principale
Personnel	Principale

**Tableau 1 : Identification des acteurs**

Pour administrer la base, l'administrateur doit tout d'abord s'identifier (saisir son login et mot de passe), puis le système vérifie ces paramètres, ensuite le système lui attribut un ensemble de privilèges, enfin l'administrateur peut agir avec la base selon ses droits donnés.

### III.2.2. Identification des cas d'utilisation :

Le tableau ci-dessous nous présente les acteurs ainsi que les cas d'utilisation que les appartiennent.

Acteur	Cas d'utilisation
Personnel	-Consulter
Administrateur	-Ajouter Personnel - Modifier Personnel -Supprimer Personnel -Ajouter Administrateur -Modifier Administrateur -Supprimer Administrateur - Ajouter Prime -Associer Prime

**Tableau 2 : Identification des acteurs par cas d'utilisation**

### III.2.3. Description textuelle des cas d'utilisation :

#### Description de cas d'utilisation « s'authentifier »

Ce cas d'utilisation est celui dans le lequel l'administrateur (ou le responsable) peut effectuer des opérations sur le système tel que la consultation des différents champs de la base de données, la modification aussi l'association d'un prime à un ensemble d'employé....

#### Description des scénarios :

- Pré condition : l'administrateur possède un login et un mot de passe
- Scénario nominal :

Administrateur	Système
1- Saisir le Login et le mot de passe	2- Vérifier les coordonnées saisis 3- Accès au système

- Scénario alternatif : Saisi erroné

#### Enchaînement au point 3

Administrateur	Système
	3-Afficher un message d'erreur
Reprise au point 1	
<ul style="list-style-type: none"> <li>• Post Condition : Connexion au système établit</li> </ul>	

**Tableau 3 : Description de scénario « Accéder au système »**

#### Description de cas d'utilisation « Consulter »

Ce cas d'utilisation permet de consulter les différents champs de la base de données. En déclenchant cette fonction le système déclenche automatiquement une recherche pour pouvoir consulter les champs du personnel demandé.

**Description des scénarios :**

<ul style="list-style-type: none"> <li>• Pré condition : l'administrateur a bien saisi le Login et le mot de passe</li> <li>• Scénario nominal :</li> </ul>	
Administrateur/Personnel	Système
1- Cliquer sur consulter	2- Déclencher une recherche
3- Remplir le formulaire	4- Afficher les données
<ul style="list-style-type: none"> <li>• Scénario alternatif : Personnel inexistant</li> </ul> Enchaînement au point 4	
Administrateur	Système
	4-Afficher un message d'erreur
Reprise au point 3 Post condition : Données affichées.	

**Tableau 4 : Description de scénario « Consulter Personnel »****Description de cas d'utilisation «Ajouter Personnel»**

Ce cas d'utilisation permet à l'administrateur d'ajouter un nouvel personnel, simplement en remplissant les champs du formulaire.

**Description des scénarios :**

<ul style="list-style-type: none"> <li>• Pré condition : l'administrateur possède le privilège d'ajout du personnel</li> <li>• Scénario nominal :</li> </ul>	
Administrateur	Système
1- Cliquer sur Nouveau	
2- Remplir le formulaire	3- Vérifier les données saisis 4-Ajouter Personnel
<ul style="list-style-type: none"> <li>• Scénario alternatif : Données invalides</li> </ul> Enchaînement au point 4	
Administrateur	Système
	4-Afficher un message d'erreur
<ul style="list-style-type: none"> <li>• Poste Condition : Personnel ajouté</li> </ul>	

**Tableau 5 : Description de scénario « Ajouter Personnel »****Description de cas d'utilisation «Modifier Personnel»**

Ce cas d'utilisation permet à l'administrateur de modifier les données concernant un personnel déjà ajouté au système.

**Description des scénarios :**



<ul style="list-style-type: none"> <li>• Pré condition : l'administrateur possède le privilège de modifier un personnel existant</li> <li>• Scénario nominal :</li> </ul>	
Administrateur	Système
1- Cliquer sur « Modifier » 2- Remplir formulaire	3- Déclencher une recherche 4- Afficher donnée
5- Modifier donnée 6- Cliquer sur « Valider »	7- Vérifier données 8- Modifier les données
<ul style="list-style-type: none"> <li>• Scénario alternatif 1 : Données invalides</li> </ul> <p>Enchaînement au point 7</p>	
Administrateur	Système
	8-Afficher un message d'erreur
<ul style="list-style-type: none"> <li>• Scénario alternatif 2 : Personnel inexistant</li> </ul> <p>Enchaînement au point 3</p>	
Administrateur	Système
	4-Afficher un message d'erreur
Poste Condition : Personnel modifiée	

**Tableau 6 : Description de scénario « Modifier Personnel »**

*Description de cas d'utilisation «Supprimer Personnel»*

Ce cas d'utilisation permet à l'administrateur de supprimer un personnel de notre système.

**Description des scénarios :**

<ul style="list-style-type: none"> <li>• Pré condition : l'administrateur possède le privilège de supprimer un personnel existant</li> <li>• Scénario nominal :</li> </ul>	
Administrateur	Système
1- Cliquer sur « Supprimer» 2- Remplir formulaire	3- Déclencher une recherche 4- Afficher donnée
5- Cliquer sur « Supprimer »	6- Supprimer les données du personnel
<ul style="list-style-type: none"> <li>• Scénario alternatif 1 : Personnel inexistant</li> </ul> <p>Enchaînement au point 3</p>	
Administrateur	Système
	4-Afficher un message d'erreur
• Poste Condition : Personnel supprimée	

**Tableau 7 : Description de scénario « Supprimer personnel »**

*Description de cas d'utilisation «Ajouter Administrateur»*

Ce cas d'utilisation permet à l'administrateur attribuer des privilèges à un personnel donné

**Description des scénarios :**

<ul style="list-style-type: none"> <li>• Pré condition : l'administrateur possède le privilège d'attribuer des privilèges</li> <li>• Scénario nominal :</li> </ul>	
Administrateur	Système
1- Cliquer sur « Nouveau »	
2- Remplir le formulaire	3- Vérifier les données saisis 4- Ajouter Administrateur
<ul style="list-style-type: none"> <li>• Scénario alternatif : Données invalides</li> </ul> <p>Enchaînement au point 4</p>	
Administrateur	Système
	4-Afficher un message d'erreur
<ul style="list-style-type: none"> <li>• Poste Condition : Privilège attribué</li> </ul>	

**Tableau 8: Description de scénario « Ajouter Administrateur »**

*Description de cas d'utilisation «modifier Administrateur»*

Ce cas d'utilisation permet à l'administrateur de modifier l'ensemble de privilège d'un administrateur donné.

**Description des scénarios :**

<ul style="list-style-type: none"> <li>• Pré condition : l'administrateur possède le privilège de modifier les privilèges</li> <li>• Scénario nominal :</li> </ul>	
Administrateur	Système
1- Cliquer sur « Modifier »	
2- Remplir le formulaire	3-Rechercher donné
4-Cliquer sur « Valider »	5- Vérifier les données saisis 6- Modifier Privilège
<ul style="list-style-type: none"> <li>• Scénario alternatif : Données invalides</li> </ul> <p>Enchaînement au point 4</p>	
Administrateur	Système
	4-Afficher un message d'erreur
<ul style="list-style-type: none"> <li>• Poste Condition : Privilège modifié</li> </ul>	

**Tableau 9 : Description de scénario « Modifier Administrateur »**

**Description de cas d'utilisation «Supprimer Administrateur»**

Ce cas d'utilisation permet à l'administrateur de priver un autre administrateur d'un ensemble de privilège.

**Description des scénarios :**

<ul style="list-style-type: none"> <li>• <b>Pré condition :</b> l'administrateur possède le privilège de supprimer un administrateur</li> <li>• <b>Scénario nominal :</b></li> </ul>	
Administrateur	Système
1- Cliquer sur « Supprimer»	
2- Remplir le formulaire	
5-Cliquer sur « Valider »	3- Rechercher donné 4- Afficher donné 6-Supprimer privilège
<ul style="list-style-type: none"> <li>• <b>Scénario alternatif : Administrateur inexistant</b> Enchainement au point 4</li> </ul>	
Administrateur	Système
	4-Afficher un message d'erreur
<ul style="list-style-type: none"> <li>• <b>Poste Condition : Administrateur supprimer</b></li> </ul>	

**Tableau 10 : Description de scénario « Supprimer Administrateur »**

**Description de cas d'utilisation «Ajouter prime»**

Ce cas d'utilisation permet à l'administrateur d'ajouter une prime

**Description des scénarios :**

<ul style="list-style-type: none"> <li>• <b>Pré condition :</b> l'administrateur possède le privilège d'ajouter une nouvelle prime</li> <li>• <b>Scénario nominal :</b></li> </ul>	
Administrateur	Système
1- Cliquer sur « Nouveau »	
2- Remplir formulaire	
	3- Vérifier donnée 4- Ajouter Prime
<ul style="list-style-type: none"> <li>• <b>Scénario alternatif : Donnée invalide</b> Enchainement au point 3</li> </ul>	
Administrateur	Système
	4-Afficher un message d'erreur
<b>Poste Condition : Prime ajoutée</b>	

**Tableau 11 : Description de scénario « Ajouter prime »**

**Description de cas d'utilisation « Associer prime »**

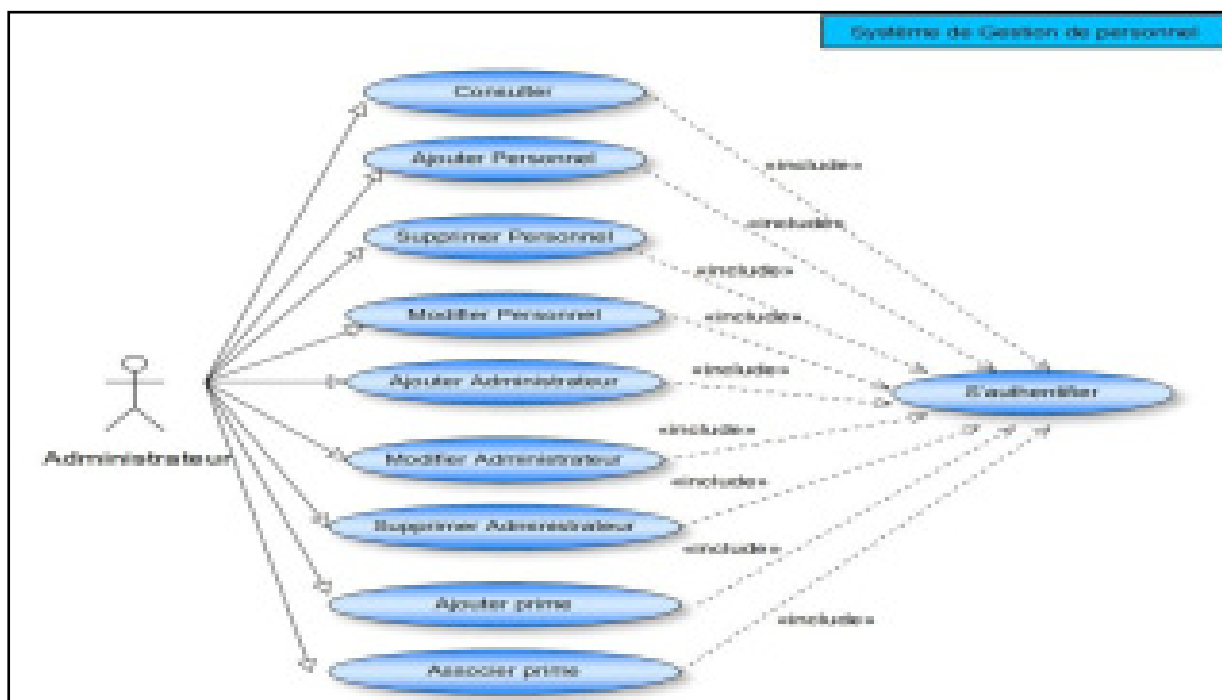
Ce cas d'utilisation permet à l'administrateur d'associer prime à un ensemble de personnel

**Description des scénarios :**

<ul style="list-style-type: none"> <li><b>Pré condition :</b> l'administrateur possède le privilège d'associer une prime à un ensemble de personnel</li> </ul>	
<b>Scénario nominal :</b>	
<b>Administrateur</b>	<b>Système</b>
1- Cliquer sur « Associer » 2- Remplir formulaire	
	3- Vérifier saisi 4- Associer prime
<ul style="list-style-type: none"> <li><b>Scénario alternatif : Donnée invalide</b></li> </ul>	
<b>Enchaînement au point 3</b>	
<b>Administrateur</b>	<b>Système</b>
	4-Afficher un message d'erreur
<b>Poste Condition : Prime associée</b>	

**Tableau 12 : Description de scénario « Associer prime »**

**Diagramme de cas d'utilisation** Dans ce diagramme, nous allons présenter les différents acteurs ainsi que les cas d'utilisation qui sont liés à ces acteurs, aussi on va présenter les relations entre les cas d'utilisation eux même.



**Figure 3 : Diagramme de cas d'utilisation**

La figure 3 nous affiche le diagramme de cas d'utilisation qui explique les relations entre les acteurs ainsi que ces cas d'utilisation, aussi les entre cas d'utilisation elle-même.

### **III.3. Diagramme d'activité :**

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

Le diagramme d'activités est une représentation proche de l'organigramme ; la description d'un cas d'utilisation par un diagramme d'activités correspond à sa traduction algorithmique. Une activité est l'exécution d'une partie du cas d'utilisation, elle est représentée par un rectangle aux bords arrondis.

Le diagramme d'activité est sémantiquement proche des diagrammes de communication (appelés diagramme de collaboration en UML 1), ou d'état-transitions, ces derniers offrant une vision microscopique des objets du système.

#### ***III.3.1. Processus d'authentification :***

Pour accéder à notre application, l'utilisateur doit s'authentifier en entrant son login et son mot de passe, après vérification des champs saisis l'application interroge la base de données pour charger le groupe et les privilèges accordés à l'utilisateur. L'utilisateur accédera automatiquement à son interface d'accueil personnel.

Le processus d'authentification peut être résumé dans le diagramme d'activité suivant :

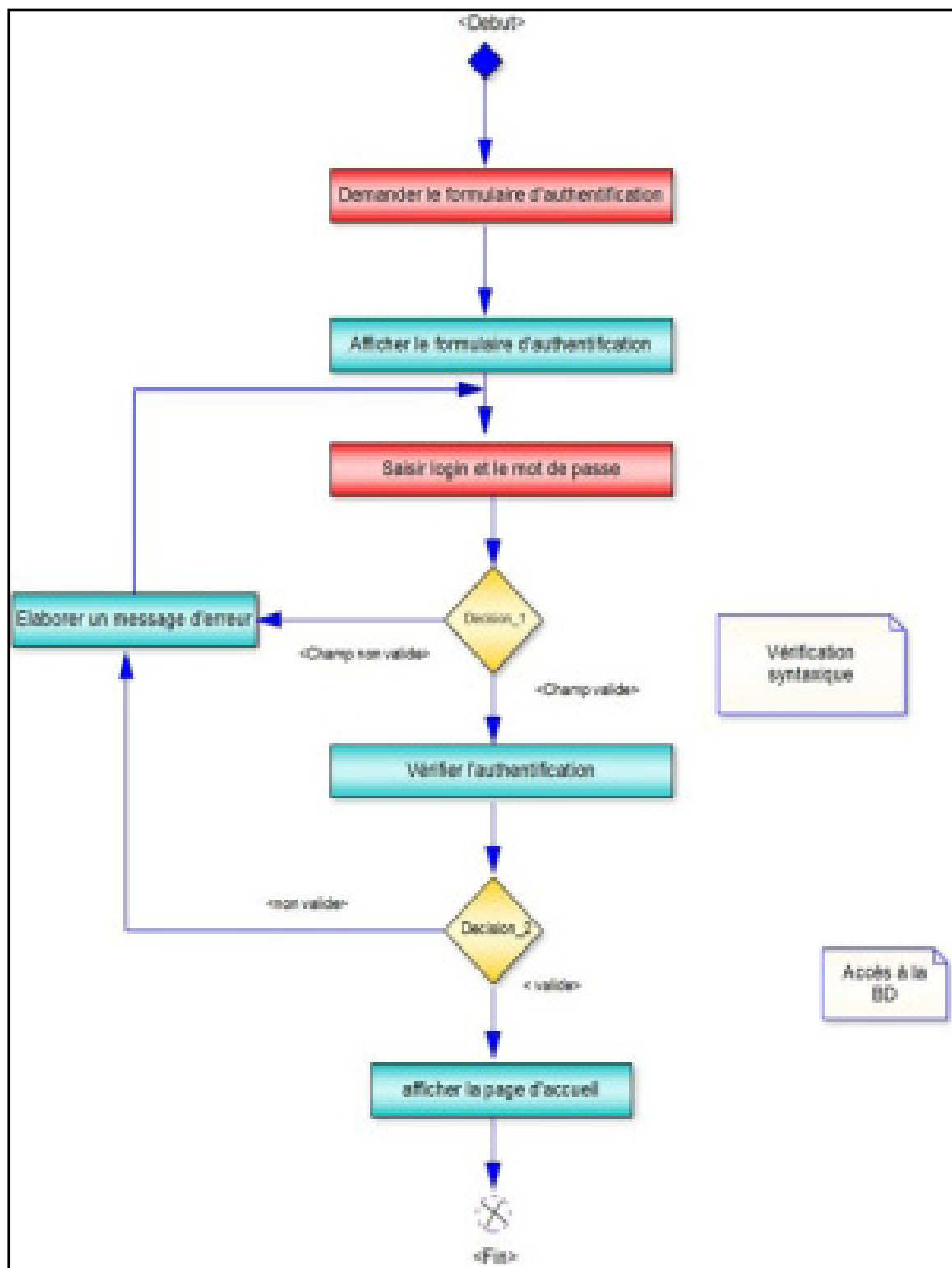


Figure 4: Digramme d'activités authentifications

### III.3.2. Processus d'ajout d'un personnel :

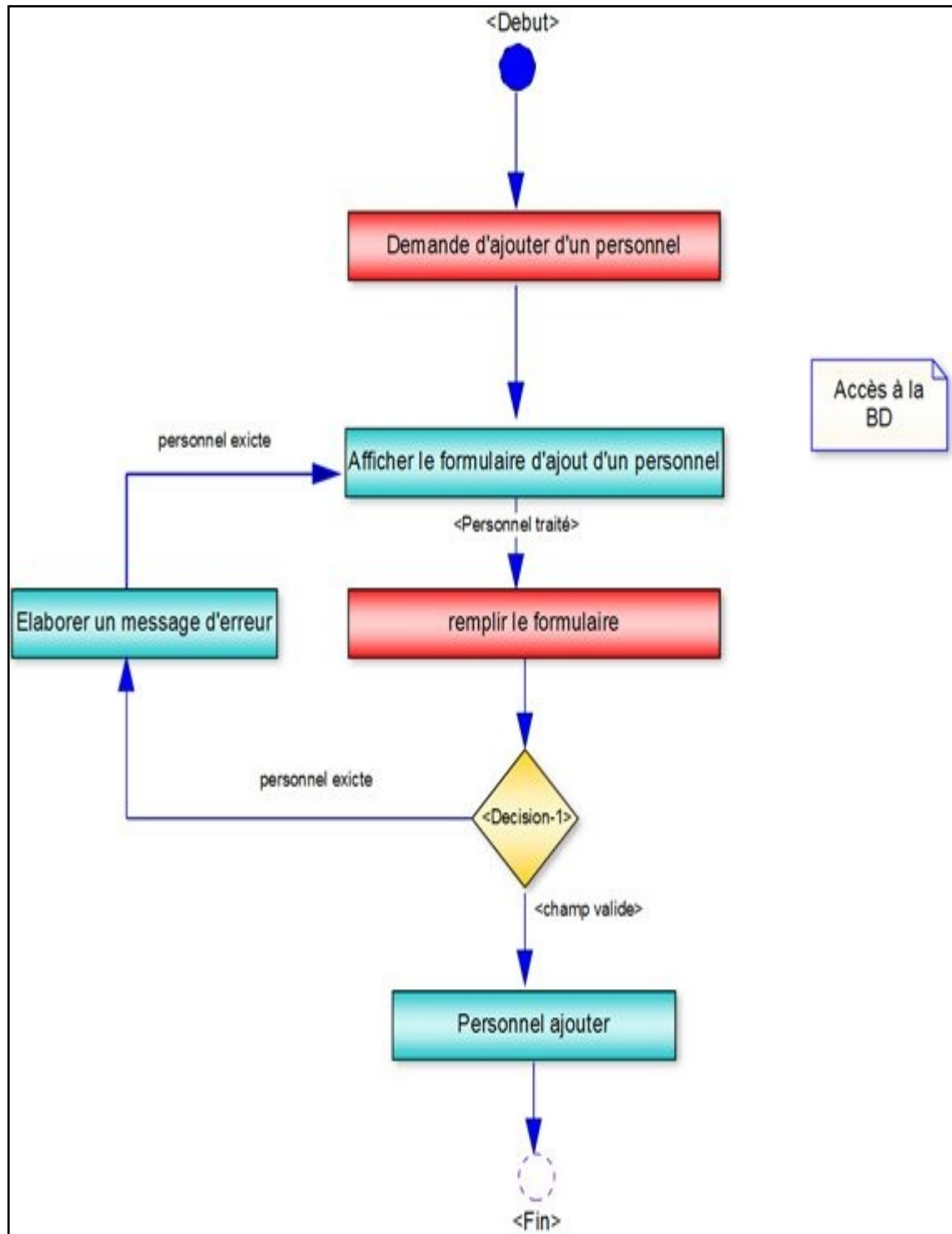


Figure 5: Digramme d'activités d'ajouter un personnel

### III.3.3. Processus de modification d'un personnel :

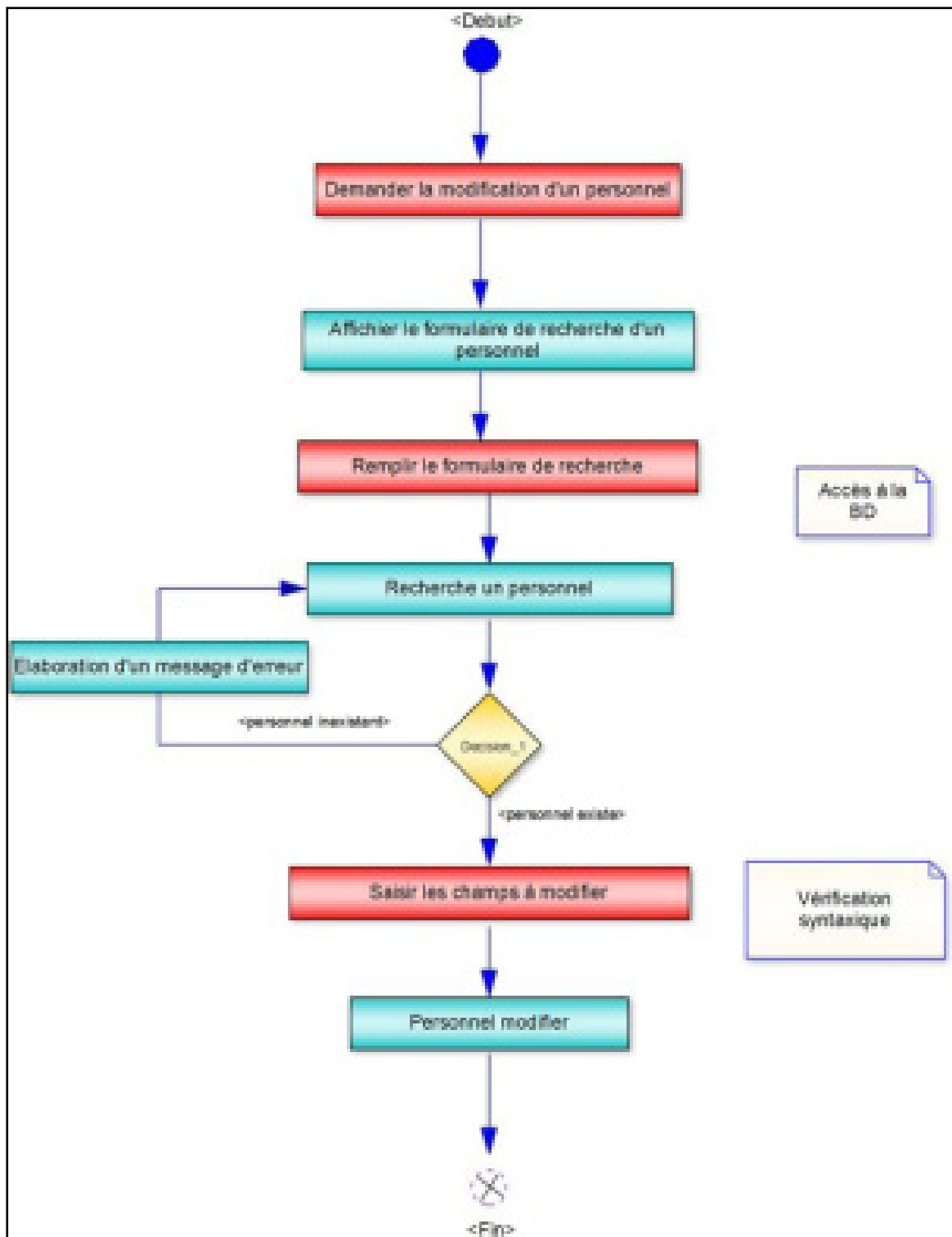


Figure 6: Digramme de modification d'un personnel



### III.3.4. Processus de suppression d'un personnel :

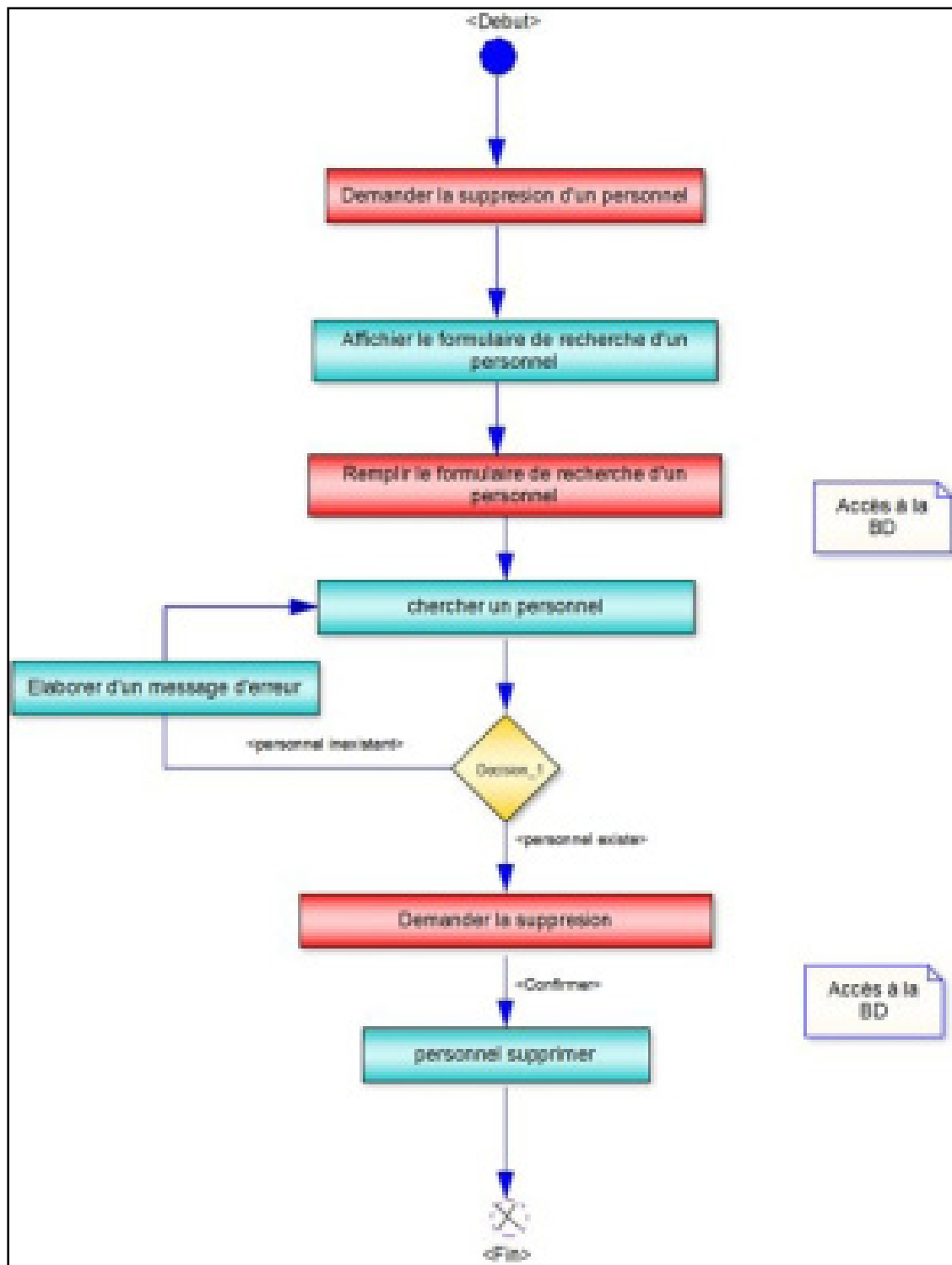
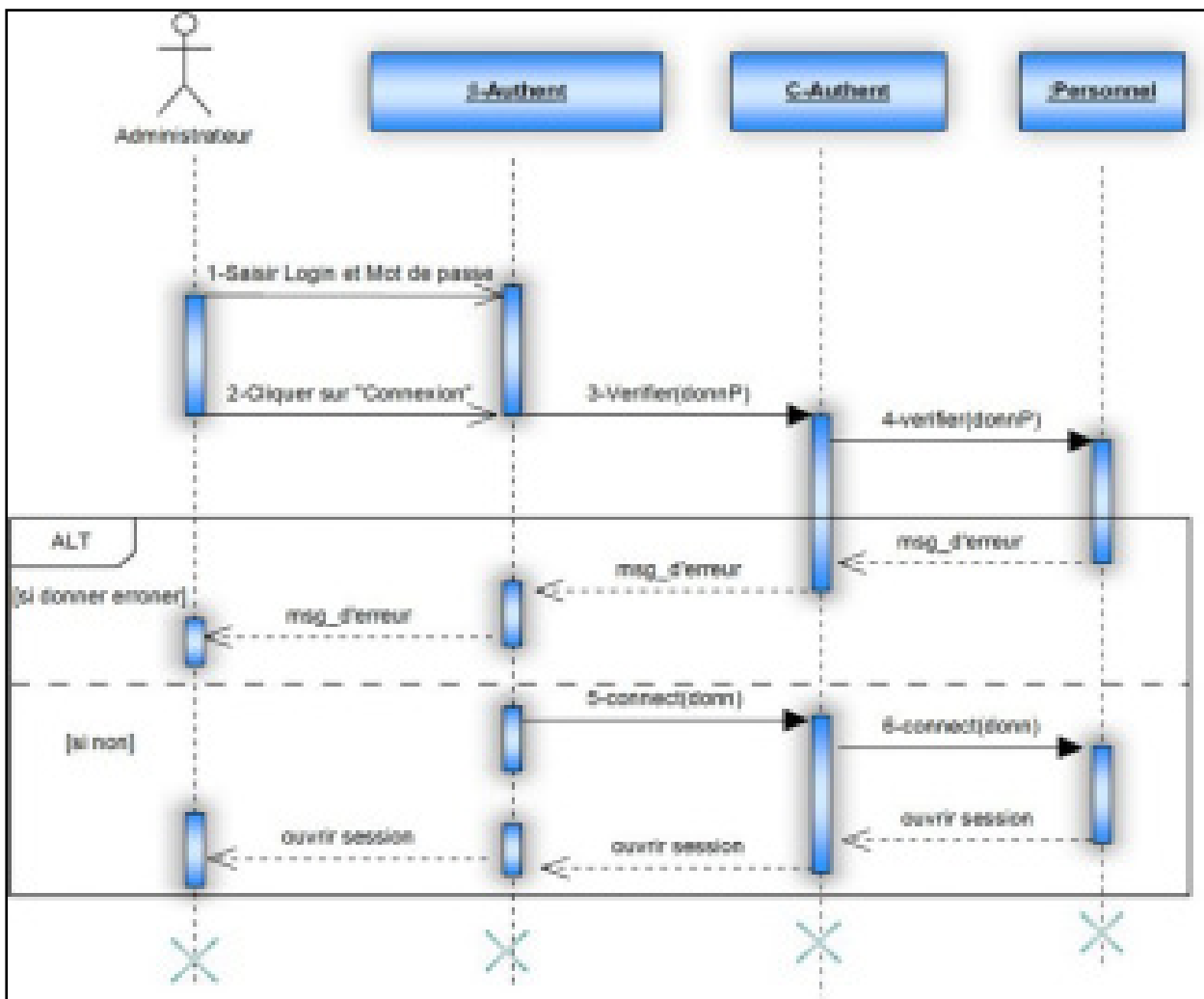


Figure 7: Diagramme d'activités de suppression d'un personnel

### III.4. Diagramme de séquence :

Les diagrammes de séquence sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML. On montre ces interactions dans le cadre d'un scénario d'un Diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets. Les périodes d'activité des classes sont symbolisées par des rectangles.

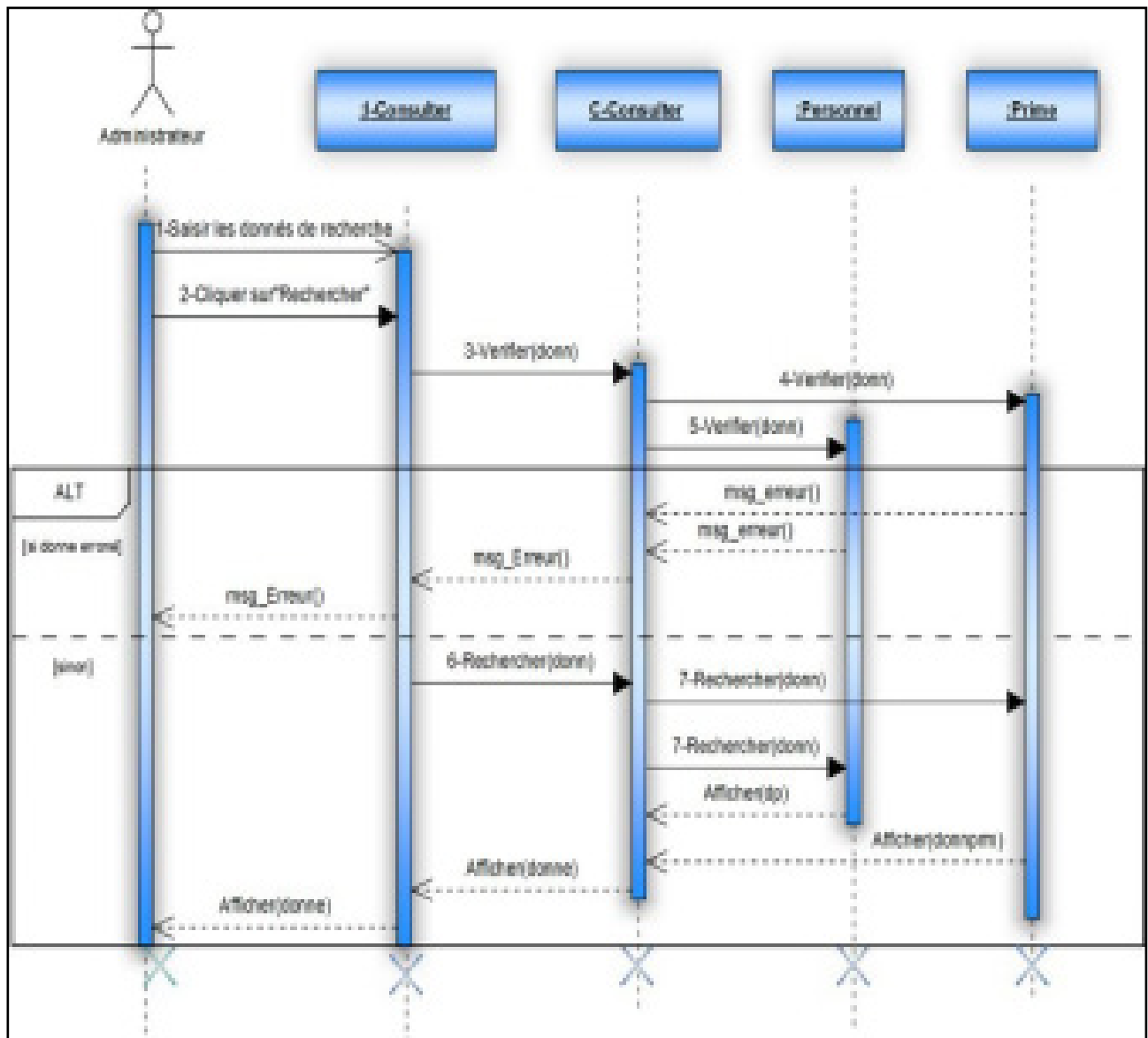
#### III.4.1. Diagramme de séquence de modèle de conception : « s'identifier »



**Figure 8: Diagramme de séquence de cas d'utilisation « s'identifier »**

Ce diagramme de séquence nous donne la séquence ment du cas d'utilisation « Accéder au système ». Car l'administrateur doit répéter la saisi de son Login et mot de passe jusqu'à ce qu'il les saisi correctement.

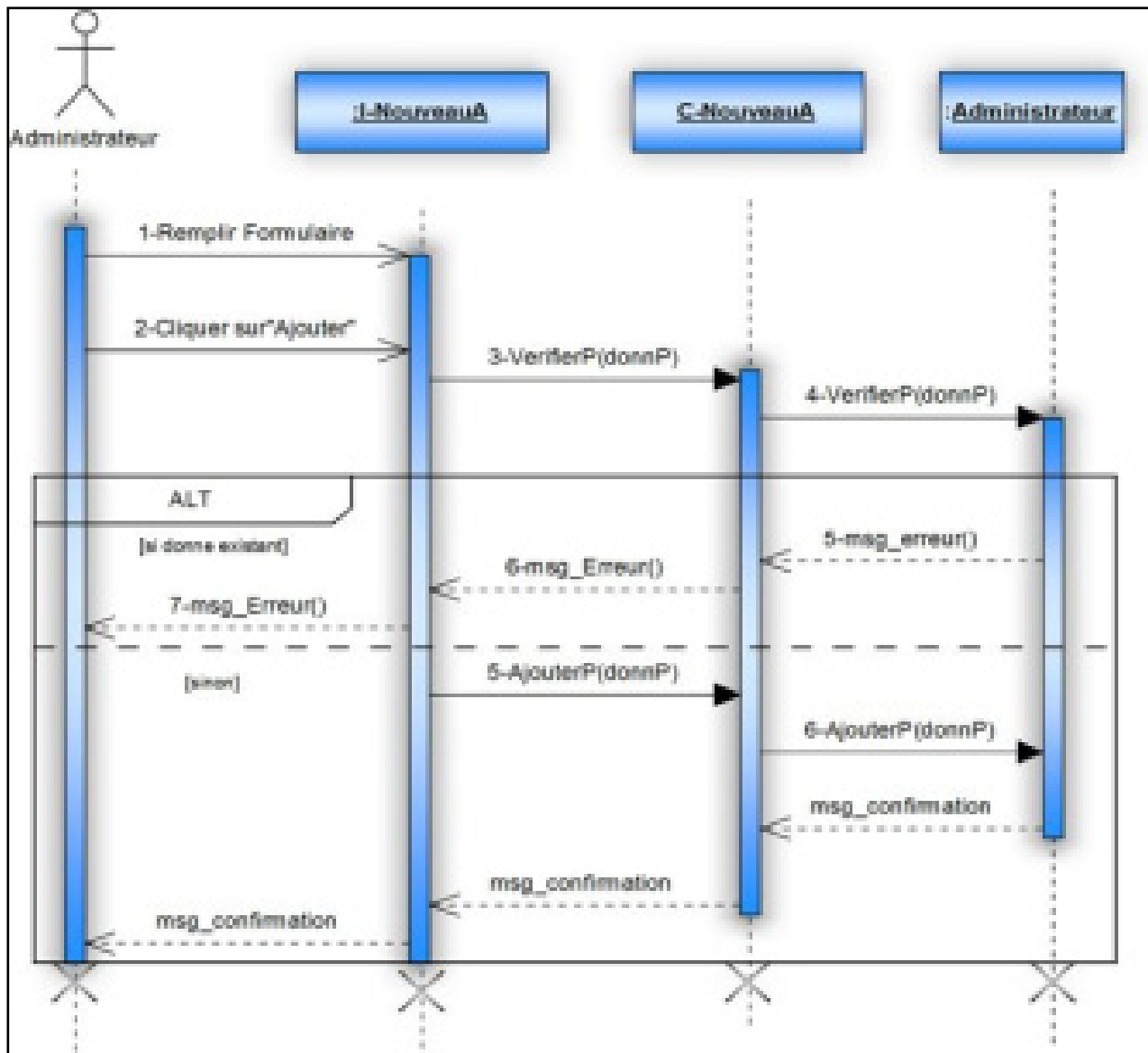
### III.4.2. Diagramme de séquence de modèle de conception « Consulter »



**Figure 9: Diagramme de séquence de cas d'utilisation « Consulter »**

La figure 9 affiche le diagramme de séquence de cas d'utilisation « Consulter », ce diagramme nous donne l'ordre chronologique dont lequel s'effectue l'opération de recherche d'un employé ou d'un prime. Si les données de recherche introduites sont invalides un message d'erreur sera afficher sinon les données voulus seront afficher.

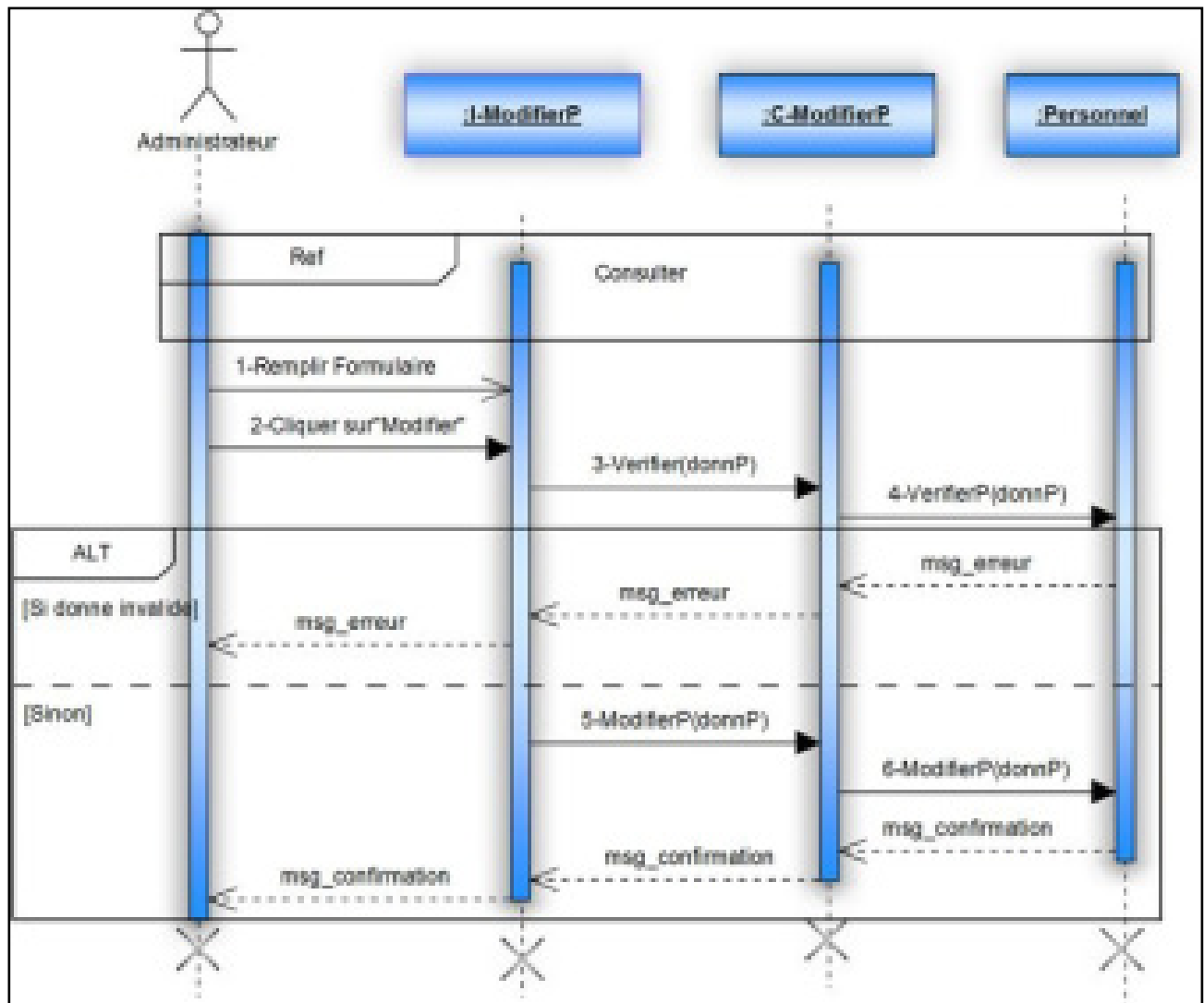
### III.4.3. Diagramme de séquence de modèle de conception : « Ajouter personnel »



**Figure 10 : Diagramme de séquence de cas d'utilisation « Ajouter Personnel »**

Le diagramme de séquence ci-dessus nous décrit comment le système agit avec le cas d'utilisation « Ajouter Personnel ». Tout d'abord l'administrateur remplit le formulaire d'ajout d'un nouveau personnel, le système les vérifie ensuite il affiche un message de confirmation d'ajout ou un message d'erreur s'il existe.

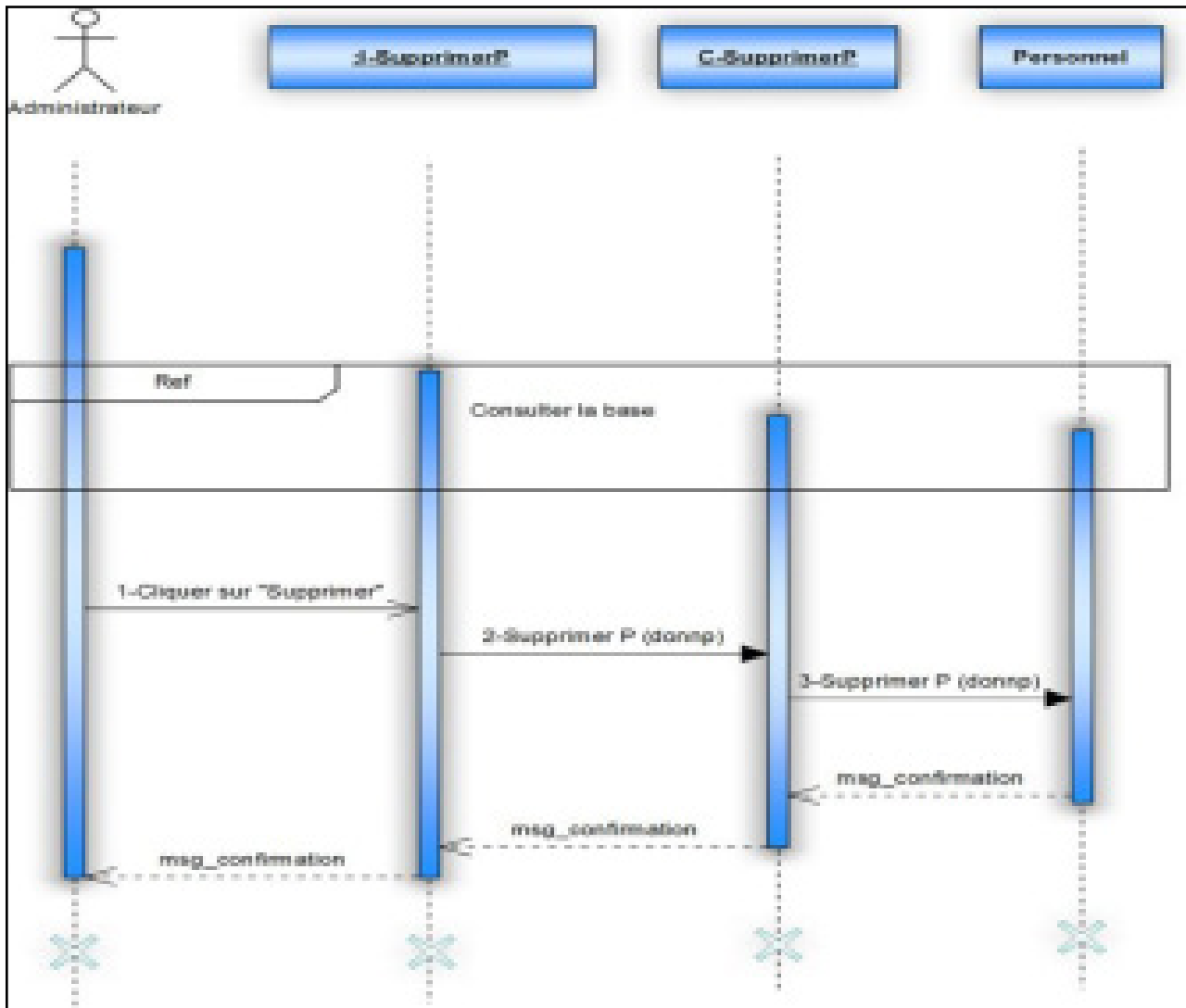
#### III.4.4. Diagramme de séquence de modèle de conception «Modifier Personnel »



**Figure 11 : Diagramme de séquence de cas d'utilisation « Modifier Personnel »**

Ce diagramme nous indique la manière dont laquelle se fait l'opération de mise à jour des données concernant un personnel : le système fait une référence au cas d'utilisation « Consulter » pour chercher le personnel désiré puis l'administrateur remplit le formulaire, enfin le système mis à jour les données et affiche un message de confirmation ou il affiche un message d'erreur s'il existe.

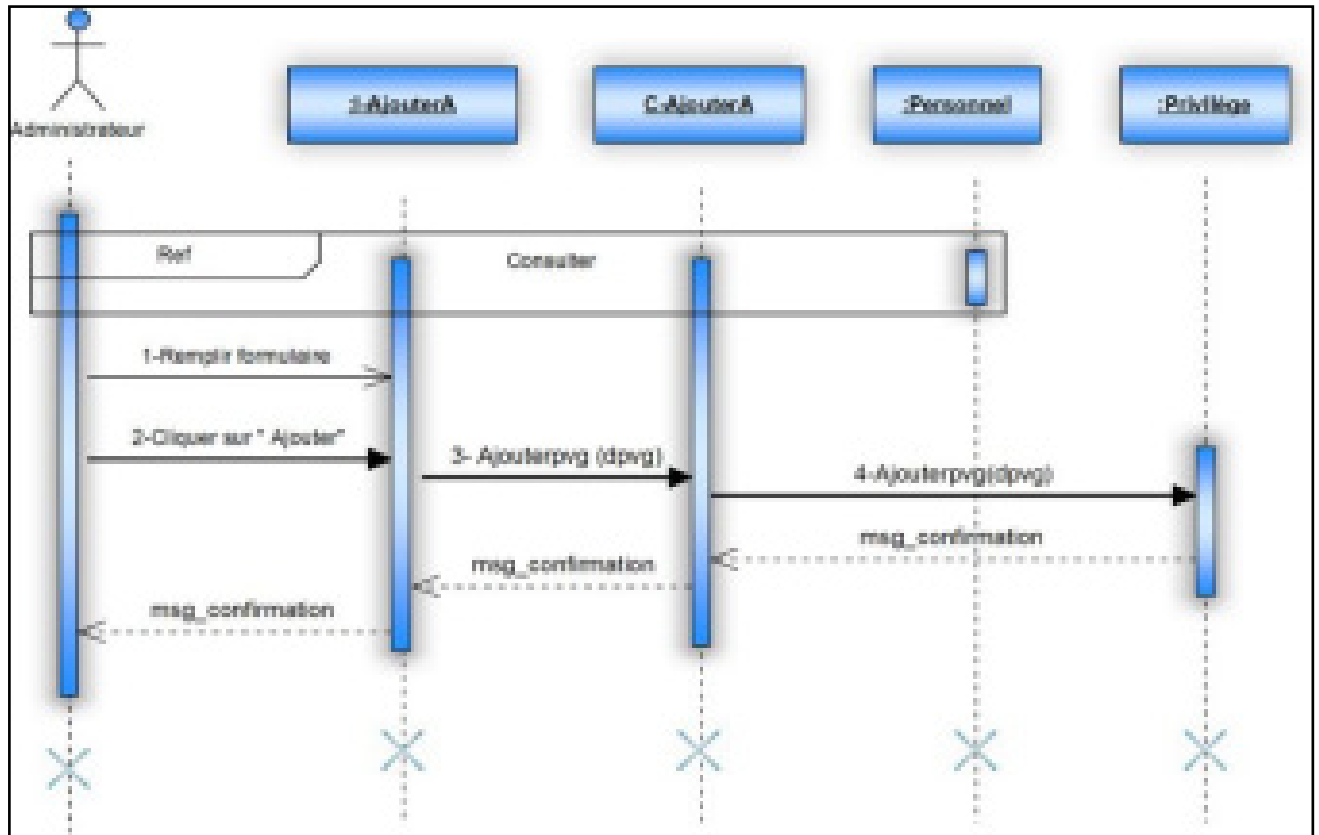
### III.4.5. Diagramme de séquence de modèle de conception «Supprimer Personnel »



**Figure 12 : Diagramme de séquence de cas d'utilisation « Supprimer Personnel »**

Le diagramme de séquence de cas d'utilisation « Supprimer Personnel » nous donne des informations sur l'opération de suppression d'un personnel. En premier lieu, l'administrateur cherche le personnel à supprimer, puis il clique sur le bouton « Supprimer » pour détruire toutes ces données.

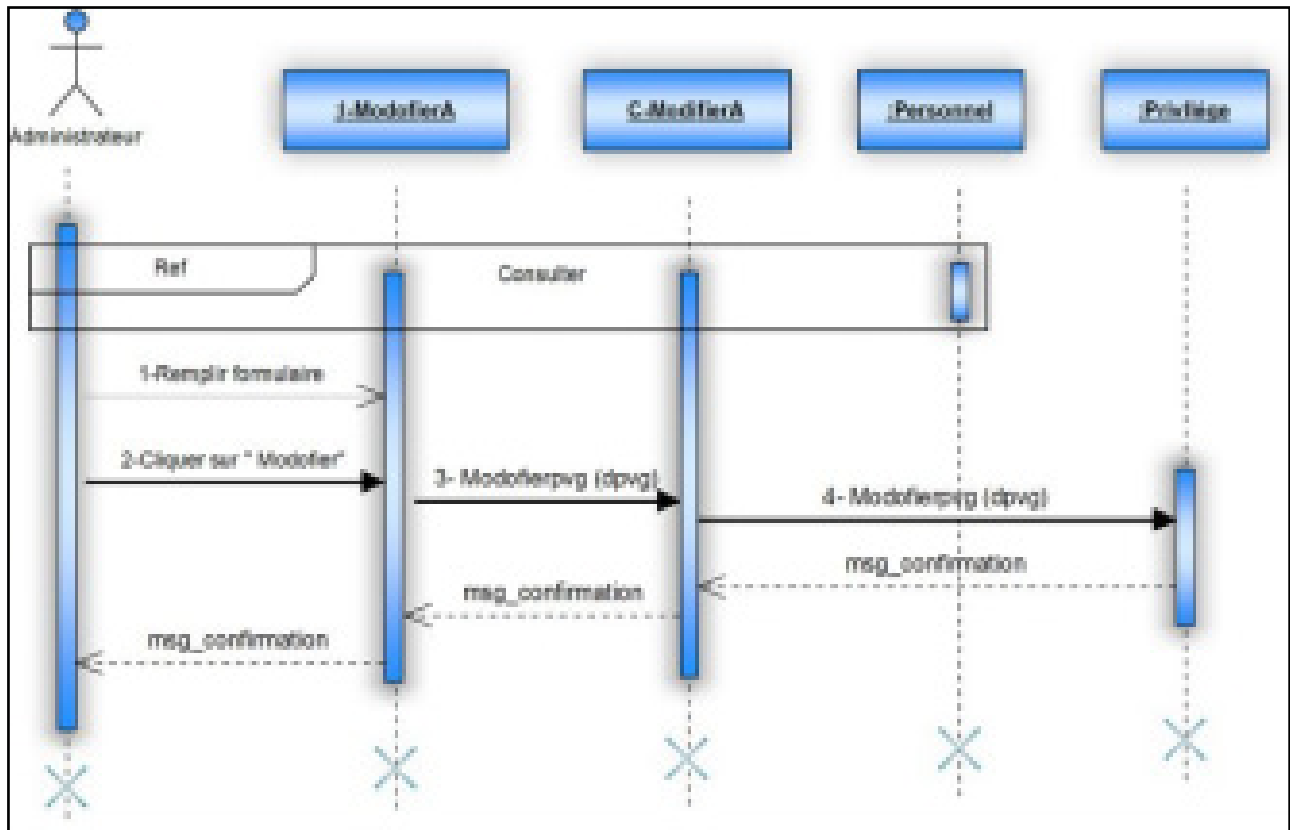
### III.4.6. Diagramme de séquence de modèle de conception «Ajouter Administrateur »



**Figure 13 : Diagramme de séquence de cas d'utilisation « Ajouter Administrateur »**

La figure 13 donne une description détaillée associée au cas d'utilisation « Ajouter Administrateur », ce cas d'utilisation permet d'accorder des privilèges à un personnel déjà existant.

### III.4.7. Diagramme de séquence de modèle de conception « Modifier Administrateur »

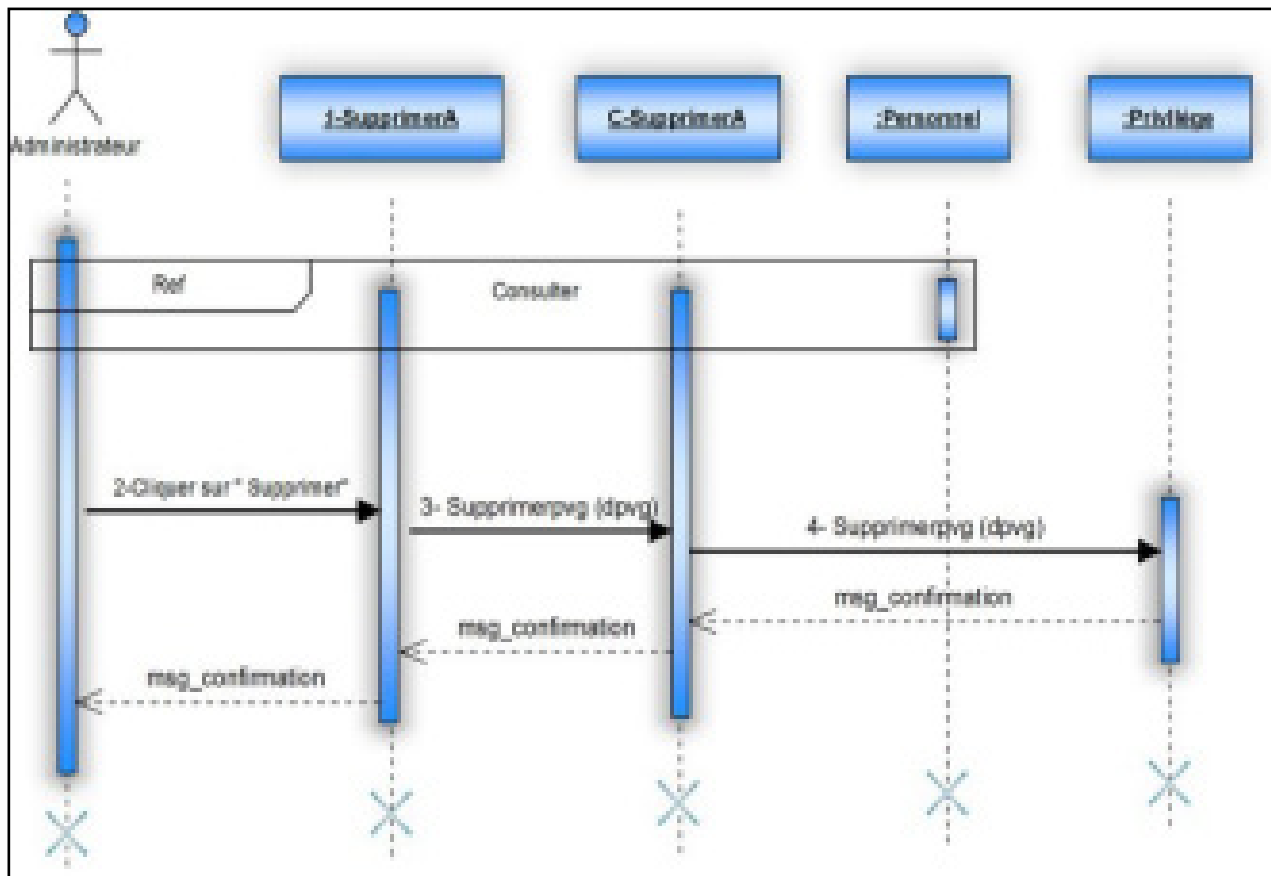


**Figure 14 : Diagramme de séquence de cas d'utilisation « Modifier Administrateur »**

Le diagramme de séquence de cas d'utilisation « Modifier Administrateur » nous donne démarche dont lequel l'administrateur va modifier les privilèges accordés un autre administrateur.



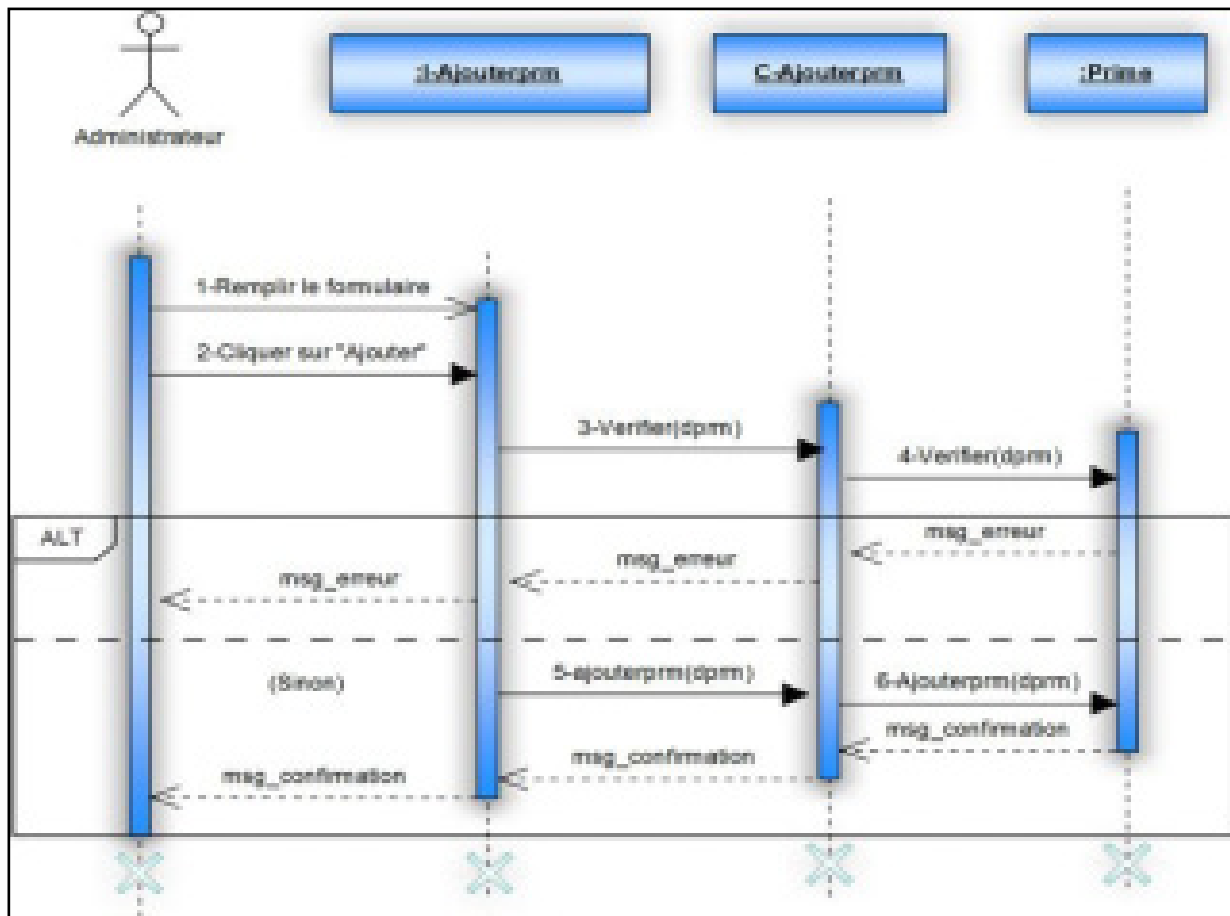
### III.4.8. Diagramme de séquence de modèle de conception «Supprimer Administrateur »



**Figure 15 : Diagramme de séquence de modèle de conception**

Le diagramme de séquence de cas d'utilisation « Supprimer Administrateur » explique la procédure de suppression d'un administrateur de système. L'administrateur qui possède ce privilège cherche l'administrateur voulu, puis il choisit les privilèges à priver. Enfin il valide son travail en cliquant sur le bouton « Envoyer ». Le système va afficher un message d'erreur s'il existe ou un message de confirmation.

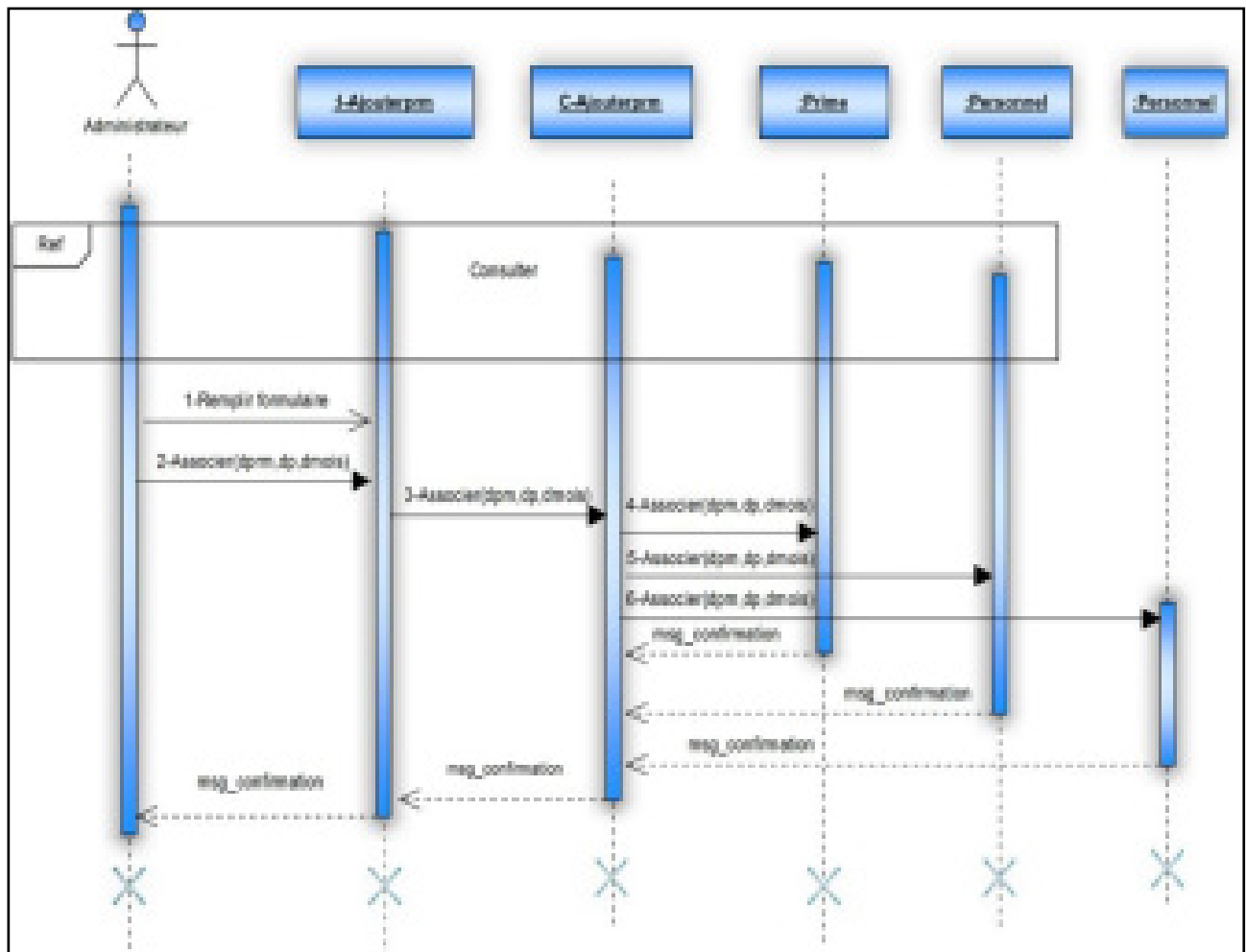
### III.4.9. Diagramme de séquence de modèle de conception « Ajouter Prime »



**Figure 16 : Diagramme de séquence de cas d'utilisation « Ajouter Prime »**

Dans ce diagramme nous avons expliqué l'opération d'ajout d'une nouvelle prime : l'administrateur du système doit en premier lieu remplir le formulaire. Ensuite il clique sur le bouton « Enregistrer » pour sauvegarder les données, s'il existe une erreur le système affiche un message d'erreur, s'il n'existe pas d'erreur le système affiche de confirmation.

### III.4.10. Diagramme de séquence de modèle de conception « Associer Prime »



**Figure 17 : Diagramme de séquence de cas d'utilisation « Associer prime »**

Le diagramme ci-dessus nous explique la démarche qu'il suit le cas d'utilisation « Associer prime » afin d'exécuter cette tâche : L'administrateur cherche tout d'abord le personnel à supprimer. Puis, il clique sur le bouton « Supprimer » donc toute donnée relatives à ce personnel sera supprimer du système.

### III.5. Diagramme de classe:

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

Il s'agit d'une vue statique car on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classe d'entité représente les différentes entités de notre système, leurs attributs ainsi que les relations entre elles.

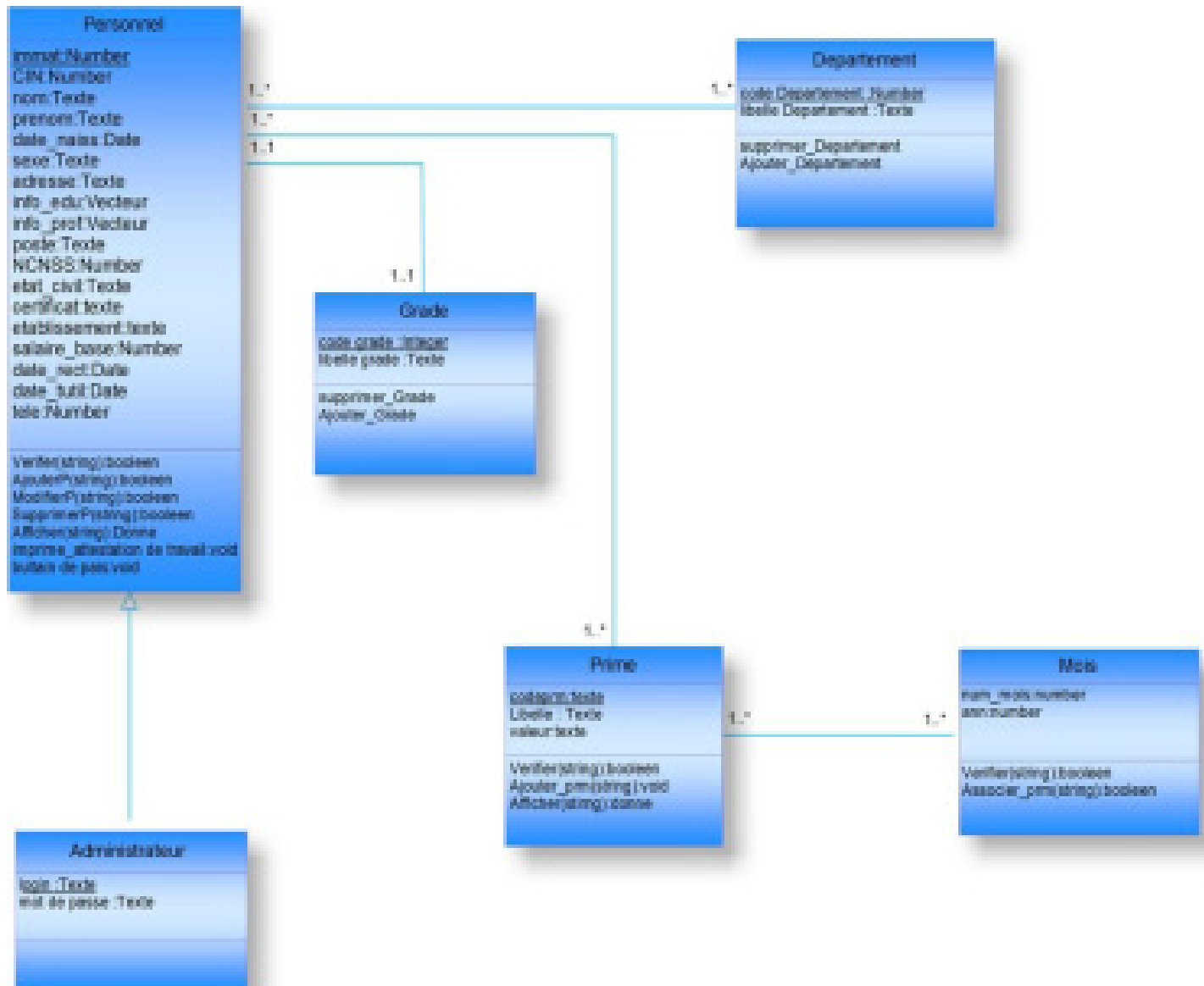
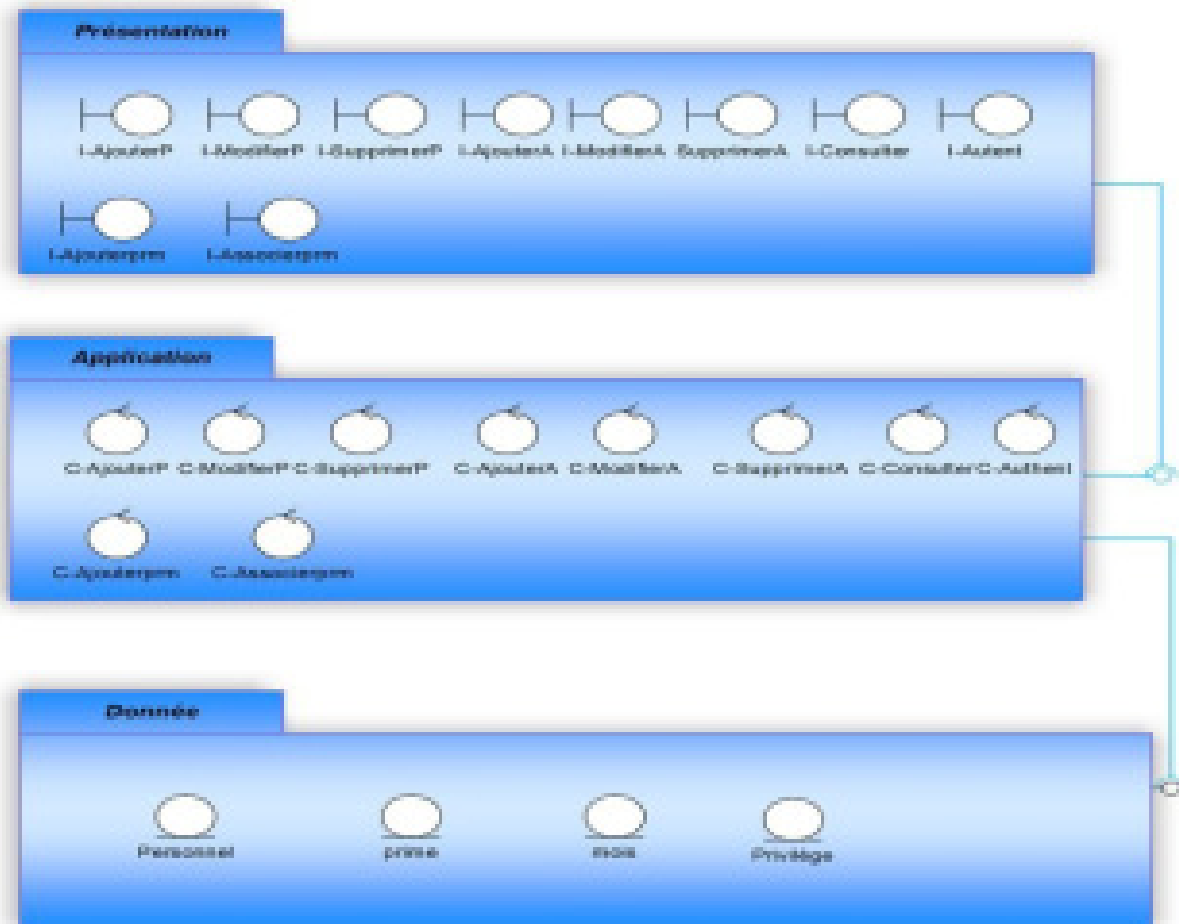


Figure 18 : Diagramme de classe

### III.6. Digramme de package :



**Figure 19 : Diagramme de package**

La figure 19 qui représente le digramme de package, nous informe sur les packages de notre système. Nous avons découpé le système en trois grand package, le premier représente le package de présentation qui contient toute les interfaces du système, le deuxième comporte les classe de contrôle qui le package d'Application et le dernier celui des données qui contient tout les entités du système.

## IV.Conclusion

Dans ce chapitre une combinaison de différents types de diagrammes UML (diagramme de cas d'utilisation, diagramme d'activité diagramme de séquence, de classe et de package) a offert une vue complète des aspects statiques et dynamiques d'un système.

## *Chapitre 3 : Réalisation*

---

## I.Introduction :

Pour commencer l'implantation de la base de données pour s'interagir avec l'application réalisée il faut tout d'abord choisir un environnement matériel et logiciel adéquat pour le bon déroulement de travail.

## II.Environnement matériel :

Notre application va être réalisée sur une machine qui comporte les caractéristiques suivant :

- Marque : Dell
- Modèle : INSPIRON 1545
- Processeur : Intel Core 2 Duo T6600 @2.20 GHz
- Disque dur : 300GO
- RAM : 2GO
- Système d'exploitation : Windows 7 édition intégrale

## III.Environnement Logiciel :



**NetBeans** est un environnement de développement intégré (IDE) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution Licence). En plus de Java, NetBeans permet également de supporter 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).



**Conteneur de Servlet Tomcat 6.0 :** apache Tomcat est un conteneur libre de servlet Java 2 Eenterprise Edition (J2EE). Issu du projet Jakarta, Tomcat est désormais un projet principal de la fondation Apache. Tomcat implémente les spécifications des Servlets et des JSP de sun Microsystems. Il inclut des outils pour la configuration et la gestion, mais peut également être configuré en éditant des fichiers de configuration XML. Comme Tomcat inclut un serveur http interne, il est aussi considéré comme un serveur http.



**MySQL server 5.5** : MySQL est le SGBDR Open Source le plus populaire au monde. Il permet de mieux répondre aux problématiques d'entreprise.

Au menu des nouveautés fonctionnelles : les vues, les procédures stockées, les déclencheurs, de nouveaux moteurs de stockage, la base de données et diverses petites améliorations.



**Toad for MySQL** : Toad est un outil de développement et de gestion de base de données MySQL. Il offre un environnement complet et intégré de développement PL/SQL qui permet de gérer tous les objets d'une base de données Oracle via une interface graphique intuitive. Des modules de débogage, de tuning, de formatage et d'administration intégrés permettent de détecter et corriger les erreurs de programmation, de simplifier l'administration quotidienne et d'optimiser les performances des bases de données MySQL.



**Pascestar UML Diagrammer** est un logiciel de modélisation UML. Ce logiciel permet de créer des diagrammes UML d'une façon simple et fiable. Il est un logiciel facile à utiliser et permet de mieux élaborer la conception des projets.



**JAVA** est langage orienté objet multiplateforme, Conçu par Sun Microsystems en 1982, présenté officiellement le 23 mai 1995 au SunWorld. JAVA est langage fortement typé, multitâche, et aussi un langage portable. Aussi JAVA gère la mémoire par un Garbage Collector qui permet la libération d'une manière automatique. Mais, elle occupe un espace mémoire important par rapport aux autres langages, ainsi que le temps d'exécution est assez important à cause de l'interprétation du code par la machine virtuelle.



Nous allons présenter dans ce qui suit quelques unes :

## **JSP « Java Server Pages »**

La technologie JSP fournit un moyen simple et extensible pour générer du contenu dynamique pour le client web.

Une page JSP est un document texte qui décrit comment traiter la requête d'un client et comment créer une réponse.

Une page JSP contient :

- Des informations de formatage (modèle) du document web, habituellement en
- HTML ou XML. Les concepteurs web peuvent modifier cette partie de la page sans
- affecter les parties dynamiques. Cette approche permet de séparer la présentation du contenu dynamique.
- Des éléments JSP et de script pour générer le contenu dynamique du document Web. La plupart des pages JSP utilisent aussi des JavaBeans et/ou des Enterprise JavaBeans pour réaliser les opérations complexes de l'application. Les JSP permettent en standard d'instancier des beans, de modifier ou lire leurs attributs et de télécharger des applets. La technologie JSP est extensible en utilisant des balises personnalisées qui peuvent être encapsulées dans des bibliothèques de balises personnalisées (taglibs)

## **Hibernate.jar :**

- Bibliothèque du Framework Hibernante

## **Servlet :**

- Une servlet est un composant qui étend les fonctionnalités d'un serveur web de manière portable et efficace.
- Un serveur web héberge des classes Java servlets qui sont exécutées à l'intérieur du container web. Le serveur web associe une ou plusieurs URLs à chaque servlet.
- La servlet est invoquée lorsque des requêtes HTTP utilisateur sont soumises au serveur.
- Quand la servlet reçoit une requête du client, elle génère une réponse, éventuellement en utilisant le logique métier contenu dans des EJBs ou en interrogeant directement une base de données. Elle retourne alors une réponse HTML ou XML au demandeur.
- Un développeur de servlet utilise l'API servlet pour :
- Initialiser et finaliser la servlet

- Accéder à l'environnement de la servlet
- Recevoir ou rediriger les requêtes et envoyer les réponses
- Interagir avec d'autres servlets ou composants
- Maintenir les informations de sessions du client
- Filtrer avant ou après traitement les requêtes et les réponses
- Implémenter la sécurité sur le tiers web

## IV.Réalisation :

Dans cette phase du rapport, on va afficher un ensemble d'interface Homme Machine qui est nécessaire pour notre application.

### ➤ Authentification

Tout d'abord, l'interface de démarrage est celle de l'authentification:

The screenshot shows a web application interface for 'Gestion des Personnels' (Personnel Management) under the 'Ministère de l'Intérieur' (Ministry of the Interior) and 'Gouvernement du Kazakhstan' (Government of Kazakhstan). The interface features a login form with the following elements:

- Navigation Links:** 'Accueil' (Home), 'Présentation' (Presentation), and 'Contact'.
- Form Fields:** 'Identifiant' (Username) and 'Mot de passe' (Password).
- Buttons:** 'Se connecter' (Login) and a green checkmark button.
- Logos:** The Turkish flag and the coat of arms of the Republic of Kazakhstan.
- Footer:** 'KAZAKHSTAN'.

**Figure 20: Interface d'authentification.**

L'authentification est une étape primordiale que chaque utilisateur de notre système doit y passer pour accéder à l'application. Cette phase assure, en effet, la sécurité de l'application. En demandant l'accès à l'application, l'utilisateur se voit dans l'obligation de s'authentifier à travers son compte. L'application vérifie l'existence de ce compte dans sa base des données de

contenu. Si l'utilisateur est identifié dans la base, il accède à l'application selon son mode d'accès fixé par l'administrateur. Une fois les données sont valides, la page d'accueil de l'utilisateur est chargée.

➤ **Interface Accueil :**



**Figure 21 : Interface d'Accueil**

C'est la première fenêtre qui s'affiche si après l'authentification. Cette fenêtre gère presque toute l'application la plupart des fenêtres qui vont s'ouvrir y prennent source.

➤ **Interface ajout nouveau personnel:**

Ministère de l'Intérieur  
Gouvernement du Koweït

Gestion des Personnels  
Gouvernement du Koweït

Espace Administrateur: Mr DOUMI Ziad

Accueil | Gestion des personnels | Gestion des postes | Gestion des documents | Déconnexion

LISTE DES PERSONNELS / AJOUTER ADMINISTRATEUR

Remplir Les paramètres ...

Matricule	1004208	NOM	BT779848
Nom	Hammadi	Prénom	Ali
Date de naissance	14/11/1981	Sexe	Homme
Adresse	Kai	Diplôme	Technicien
Etablissement		CNS	
Poste		Salaire	
Date de recrutement		Nouveau Administrateur?	
Service		Date de démissionation	
Téléphone		Etat civil	Célibataire

Envoyer

KABOUCASU

**Figure22: Fenêtre d'ajout d'un nouveau personnel**

Bouton pour  
enregistrer  
l'action faite

Cette interface permet de gérer les informations nécessaires pour les personnels et encore pour sauvegarder leurs formulaires en cliquant sur le bouton « envoyer ».

➤ **Interface ajout nouvel administrateur:**

The screenshot shows a web application interface for 'Gestion des Personnels' (Personnel Management) under the 'Ministère de l'Intérieur - Gouvernement de Madagascar'. The header includes the title 'Gestion des Personnels' and the user's role 'Espace Administrateur: Mr TOMY Riel'. A navigation bar contains links: 'Accueil', 'Gestion des personnels', 'Gestion des primes', 'Gestion des documents', and 'Déconnexion'. The main heading is 'LISTE DES PERSONNELS - AJOUTER ADMINISTRATEUR'. Below this, a red text prompt says 'Saisir les paramètres ...'. There are three input fields labeled 'Matricule', 'Login', and 'Mot de passe'. An 'Envoyer' button is positioned below the 'Mot de passe' field. At the bottom, a green line separates the form from the footer, which contains the text 'ADMINISTRATEUR'.

**Figure 23 : Interface d'ajout d'un nouvel administrateur**

L'administrateur peut accorder, ou priver un autre administrateur d'un ensemble de privilèges. A la fin de son travail, l'administrateur clique sur « Envoyer » pour enregistrer et gérer les tâches.

➤ **Interface Gestion des primes :**

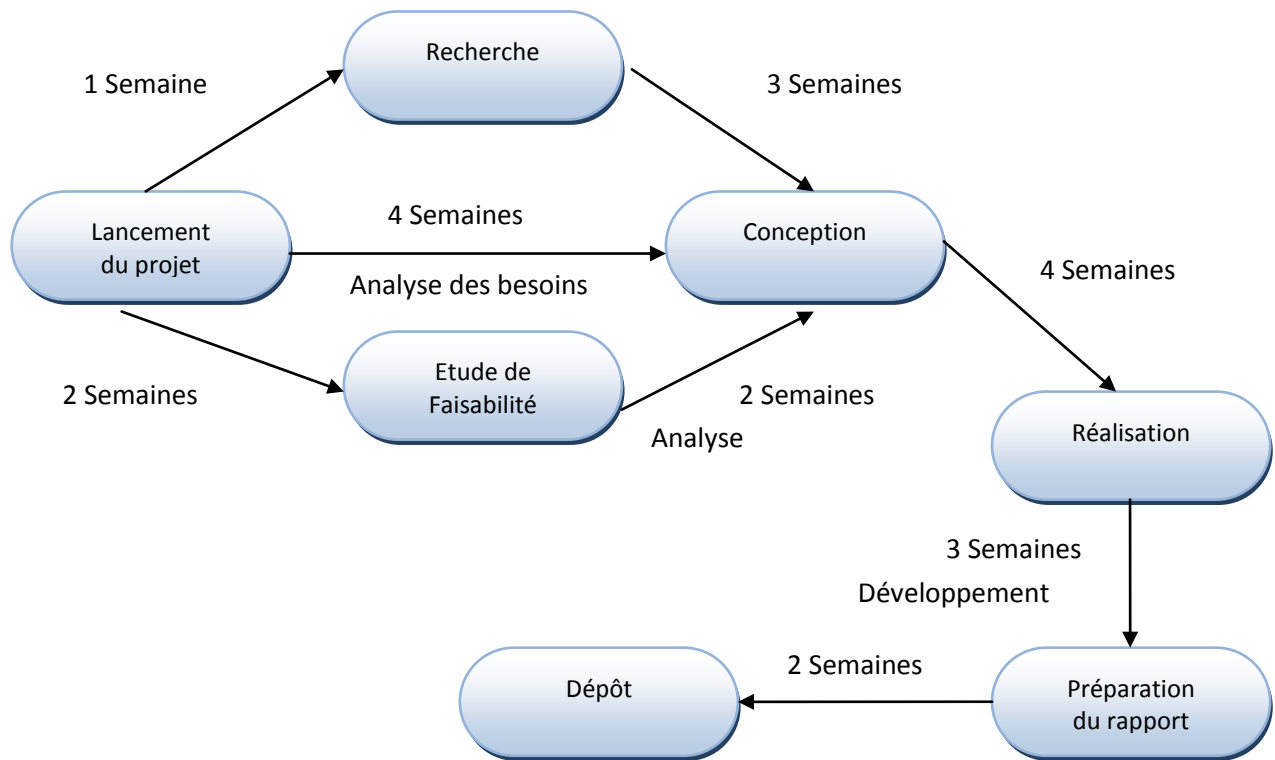


**Figure 24 : Interface des gestions des primes**

L'administrateur peut ajouter une nouvelle prime en remplissant les champs puis cliquer sur « Liste des primes » ou « Affecter prime » pour ajouter et visualiser la prime de chaque personnel et sauvegarder les données saisis.

## **IV. Chronogramme de déroulement de stage**

Le stage s'est déroulé en respectant le cahier des charges. Grâce au modèle CASCADE et sa séparation entre les logiques métiers, la génération des vues et le contrôle des actions que nous avons pu travailler en parallèle sur la conception de la partie model et la réalisation de la partie contrôle. Or d'après ce modèle, nous pouvons décomposer le travail en trois parties que nous pouvons les concevoir et implémenter indépendamment. Enfin, il est à remarquer que la partie documentation et apprentissage technique s'est étendu tout au long du stage. Ceci est due à la diversité des outils et des nouvelles technologies que nous avons utilisés.



**Figure 25 : Planning prévisionnel du projet**

## V. Conclusion

Ce chapitre nous a permis de mettre en œuvre notre conception. Nous avons commencé par présenter l'environnement matériel et logiciel de travail. Pour la suite, nous avons présenté quelques interfaces de l'application afin de donner une meilleure idée sur le travail réalisé.

---

## *Conclusion Générale*



## Conclusion Générale et perspective

Dans ce rapport de projet de fin d'étude, nous avons conçu et développé une application de gestion de personnel qui permet au responsable de ressource humaine du Gouvernorat de Kairouan de bien contrôler et gérer les employés inscrits dans cet organisme. Cette application, permet au gestionnaire d'ajouter des nouveaux personnels, mise à jour des données concernant un employé existant ainsi que l'augmentation des salaires en ajoutant des primes et les associant à un ensemble de personnels.

Sur le plan technique, cette expérience nous a permis de découvrir et d'acquérir des connaissances nouvelles, de renforcer nos connaissances programmation orientée objet et le développement d'une base de données MySQL, et d'autre part, particulièrement avec son logiciel NetBeans.

Ce stage nous a permis aussi de maîtriser le langage java appliqué au développement d'une application selon la norme J2EE. Un autre aspect assez important de notre expérience était l'opportunité de découvrir le milieu professionnel avec tout ce qu'il exige de responsabilité, de discipline et travail en équipe.

Finalement, l'application de gestion de personnel est une application modeste qu'on peut mieux la développer car c'est une application nécessaire pour chaque administration surtout dans le monde d'aujourd'hui qui est régné par les outils informatique.



# *Annexes*

# Annexe A : J2EE

## 1. Pourquoi utiliser une plateforme ?

Une plateforme est une base générique qui fournit un ensemble de fonctionnalité utile pour une majorité d'application. Une plateforme se construit sur la base d'un ensemble de besoins générique partagé entre plusieurs applications. Il peut exister plusieurs types de plateforme. De la plus générique à la plus spécifique (optimisée pour un type de métier précis par exemple). Bon nombre de grandes entreprises ont déjà développé des plateformes tels que : IBM (WebSphere), SAP...

L'avantage principal de partir d'une plateforme est que l'équipe de développement n'a pas à s'acquitter de développer certaines tâches (connexion à la base de données par exemple, gestion d'objets...). Ce sont des tâches que l'on retrouve très souvent dans un grand nombre de projet et qui n'ont pas d'intérêt à être recoder à chaque fois (perte de temps et d'argent). De plus, mieux vaut travailler sur une plateforme qui présente une stabilité (ça évite des débogages inutiles sur une plateforme qui présente une forte stabilité (ça évite des débogages inutiles !).

Un autre avantage est facilité de prise en main des API de cette plateforme. En effet, celle-ci cache très souvent la complexité d'accès à telle ou telle ressource et permet donc un gain de temps énorme pour le développeur qui a donc plus de temps pour se préoccuper du fonctionnement réel de son application (pas de tâche ardue ou générique à développer).

## 2. Qu'est ce que J2EE ?

J2EE (Java 2 Enterprise Edition) est une norme proposée par la société Sun, portée par un consortium de sociétés internationales, visant à définir un standard de développement d'applications d'entreprises multi-niveaux, basées sur des composants. On parle généralement de « plate-forme J2EE » pour désigner l'ensemble constitué des services (API) offerts et de l'infrastructure d'exécution.

J2EE comprend notamment :

- Les spécifications du serveur d'application, c'est-à-dire de l'environnement d'exécution
- J2EE définit finement les rôles et les interfaces pour les applications ainsi que l'environnement dans lequel elles seront exécutées. Ces recommandations permettent ainsi à des entreprises tierces de développer des serveurs d'application conformes aux spécifications ainsi définies, sans avoir à redévelopper les principaux services.

Des services, au travers d'API, c'est-à-dire des extensions java indépendantes permettant d'offrir en standard un certain nombre de fonctionnalités. Sun fournit une implémentation minimale de ces API appelée J2EE SDK (J2EE Software Development Kit).

### **3. Les acteurs d'une application J2EE**

La réalisation d'une application basée sur l'architecture J2EE fait appel à différents types de compétences jusqu'à la supervision de l'application en passant par le développement.

Pour ce faire, Java 2EE définit les éléments suivants :

- Une plate-forme (Java EE Platform), pour héberger et exécuter les applications.
- Une suite de tests (Java EE Compatibility Test Suite) pour vérifier la compatibilité.
- Une réalisation de référence (Java EE Reference Implementation), qui est GlassFish.
- Un catalogue de bonnes pratiques (Java EE BluePrints).

## Annexe B : l'approche orientée objet

L'approche orientée objet considère le logiciel comme une collection d'objets dissociées, identifiés et possédant des caractéristiques. Une caractéristique est soit un attribut (i.e une donnée caractérisant l'état de l'objet), soit une entité comportementale de l'objet (i.e une fonction). La fonctionnalité du logiciel émerge alors de l'interaction entre les différents objets. Qui le constituent. L'une des particularités de cette approche est qu'elle rapproche les données qui le constituent. L'une des particularités de cette approche est qu'elle rapproche les données et leurs traitements associés au sein d'un unique objet.

Comme nous venons de le dire, un objet est caractérisé par plusieurs notions :

### L'identité

L'objet possède une identité, qui permet de le distinguer des autres objets, indépendamment de son état. On construit généralement cette identité grâce à un identifiant découlant naturellement du problème (par exemple un personnel peut avoir une ou plusieurs prime par mois)

### Les attributs

Il s'agit des données caractérisant l'objet. Ce sont des variables stockant des informations sur l'état de l'objet.

### Les méthodes

Les méthodes d'un objet caractérisent son comportement, c'est-à-dire l'ensemble des actions (appelées opérations) que l'objet est à même de réaliser. Ces opérations permettent de faire réagir l'objet aux sollicitations (ou d'agir sur les autres objets) de plus, les opérations sont étroitement liées aux attributs, car leurs actions peuvent dépendre des valeurs des attributs, ou bien les modifier.

## Annexe C : UML

UML est un langage graphique qui permet de représenter et de communiquer les divers aspects d'un système d'information. Aux graphiques sont bien sur associés des textes qui expliquent leur contenu. UML est donc un métalangage car il fournit les éléments permettant de construire le modèle qui, lui, sera le langage du projet.

Il est impossible de donner une représentation graphique complète d'un logiciel, ou de tout autre système complexe, de même qu'il est impossible de représenter entièrement une statue (à trois dimensions) par des photographies (à deux dimensions). Mais il est possible de donner sur un tel système des vues partielles, analogues chacune à une photographie d'une statue, et dont la conjonction la conjonction donnera une idée utilisable en pratique sans risque d'erreur grave.

UML 2.0 comporte ainsi treize de diagramme représentant autant de vues distinctes pour représenter des concepts particuliers du système d'information. Ils répartissent en deux grands groupes :

### **Diagrammes structurels ou diagramme statiques (UML Structure)**

- Diagramme de classes (Class diagram)
- Diagramme d'objets (Object diagram)
- Diagramme de composants (Component diagram)
- Diagramme de déploiement (Deployment diagram)
- Diagramme de paquetages (Package diagram)
- Diagramme de structures composites (Composites structure diagram)

### **Diagramme comportementaux ou diagrammes dynamiques (UML Behavior)**

- Diagramme de cas d'utilisation (Use case diagram)
- Diagramme d'activités (Activity diagram)
- Diagramme d'états-transitions (State machine diagram)

### **Diagrammes d'interaction (Interaction diagram)**

- Diagramme de séquence (Sequence diagram)
- Diagramme de communication (Communication diagram)
- Diagramme global d'interaction (Interaction overview diagram)
- Diagramme de temps (Timing diagram)

---

# *Neto-graphie*

Cours JAVA de Mr Biadh Bouhouchi Université Virtuelle de Tunis

Adresses Internet:

[www.code-source.com](http://www.code-source.com)

[www.phpsources.org](http://www.phpsources.org)

[www.siteduzero.com](http://www.siteduzero.com)

[www.phpscripts-fr.net](http://www.phpscripts-fr.net)

[www.joomla.fr](http://www.joomla.fr)

[www.commentcamarche.net](http://www.commentcamarche.net)

[www.wikipedia.com](http://www.wikipedia.com)

- CSS:

<http://www.commentcamarche.net/css>

- Menus :

<http://iubito.free.fr>

<http://www.codes-sources.com/>

<http://www.javafr.com/>

<http://www.developpez.com/>

<http://www.youtube.com/>