

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMARI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE  
DEPARTEMENT D'INFORMATIQUE

## **Mémoire de Fin d'Etudes de MASTER ACADEMIQUE**

Domaine : **Mathématiques et Informatique**

Filière : **Informatique**

Spécialité : **Conduite de projet informatique**

*Présenté par*

**Naima ABBOU**

**Ghania MANE**

Thème

# **Conception et Réalisation d'une application web pour la planification des activités pédagogiques Cas : Département d'Informatique**

*Mémoire soutenu publiquement le 14 / 07 / 2016 devant le jury composé de :*

**Président : Mr KERBICHE Mhend**

**Encadreur : Mr HAMEG Samir**

**Examineur : Mr DIB Ahmed**

**Examineur : Mr CHEBOUBA Loukmane**

# Remerciement

Au terme de ce travail,

Nous tenons en premier lieu à remercier le bon Dieu pour le courage et la patience qu'il nous a donné afin de mener ce projet à terme.

Nous tenons à exprimer notre profondes gratitude à notre promoteur, Mr Hameg de nous avoir encadré et guidé tout au long de notre projet, pour ses conseils judicieux minutieusement prodigués. Aussi nous tenons à lui reconnaître le temps précieux qu'il nous a consacré.

On remercie les membres de jury pour l'honneur qu'ils nous font en jugeant ce modeste travail

Nos remerciements vont aussi à tous le personnel de l'UMMTO et particulièrement à Mr Ramdane qui nous a généreusement aidé pour réaliser ce travail

Nos sincères sentiments vont à tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce projet

## Dédicaces

**A celle et A celui qui m'abreuvent d'amour et d'affection intarissable, sources de mon bonheur et ma raison d'être, ma très chère mère et mon très cher père. Que Dieu me les garde encore aussi longtemps que je vivrai.**

**Ames adorables frères qui ont su guider mes pas vers ce que je suis devenu aujourd'hui.**

**A mes oncles cousins et cousines, voisins et voisines.**

**A mes grands parents maternelles et paternelle que le grand Dieu les garde en paix dans son vaste paradis.**

**A mon binôme qui par leur participation effective a permis à ce projet de voir le jour**

**A tous mes ami(e)s avec qui j'ai partagée des beaux et inoubliables moments de ma vie.**

**Et a la mémoire de mon cher ami Samir qui est toujours présent dans mon cœur malgré son absence.**

**Enfin, à tous ce beau monde, que j'aime, je dédie ce modeste travail.**

**Ghania MANE**

# Dédicaces

*Je dédie ce modeste travail :*

*A La mémoire de Mon père que dieu l'accueil dans son vaste paradis ;*

*A mon adorable et chère mère que j'aime énormément, pour sa présence et son soutien, et surtout pour la confiance qu'elle a toujours eu en moi, j'espère qu'elle trouvera dans ce modeste travail toute ma reconnaissance et tout mon amour ;*

*A Mes très chers frères, surtout mon grand frère « Larbi » que je remercie pour son soutien Psychologique et financier ;*

*A mes très chères sœurs ;*

*A mon binôme qui par leur participation effective a permis à ce projet de voir le jour ;*

*A toutes mes amies ainsi qu'à tous ceux qui me sont chers.*

*Naima ABBOU*

# Sommaire

<b>Introduction générale.....</b>	<b>1</b>
-----------------------------------	----------

## Chapitre I : JavaEE

<b>I. Introduction .....</b>	<b>2</b>
<b>II. JEE.....</b>	<b>2</b>
<b>II.1. Définition javaEE .....</b>	<b>2</b>
<b>II.2. Fonctionnement interne .....</b>	<b>2</b>
<b>II.3. Les éléments de la JavaEE .....</b>	<b>2</b>
<b>II.4. Architecture JavaEE.....</b>	<b>3</b>
<b>II.5. Les services JEE.. .....</b>	<b>4</b>
<b>II.6. L'environnement d'exécution .....</b>	<b>4</b>
<b>II.7. Les Conteneurs JEE .....</b>	<b>5</b>
<b>II.8. JSF .....</b>	<b>6</b>
<b>II.8.1.Présentation .....</b>	<b>6</b>
<b>II.8.2. Modèle de composants JSF UI .....</b>	<b>6</b>
<b>II.8.3. Architecture JSF .....</b>	<b>6</b>
<b>II.8.4. Avantage JSF .....</b>	<b>8</b>
<b>II.9. Les API JEE .....</b>	<b>8</b>
<b>II.10. Java EE et la sécurité.....</b>	<b>9</b>
<b>II.11. Objectifs Java EE.....</b>	<b>9</b>
<b>III. Le Framework .....</b>	<b>9</b>
<b>III.1. Présentation.....</b>	<b>9</b>
<b>III.2. Hibernate .....</b>	<b>10</b>
<b>III.2.1. Définition .....</b>	<b>10</b>
<b>III.2.2. Architecture .....</b>	<b>11</b>
<b>III.2.3. Configuration objet .....</b>	<b>12</b>

III.2.4. Object sessionFactory .....	12
III.2.5. Objet de la session .....	12
III.2.6. Transaction Object.....	12
III.2.7. Query Object .....	13
III.2.8. Critères Object .....	13
<b>IV. Conclusion.....</b>	<b>13</b>

## Chapitre II : Etude de l'existant

<b>I. Introduction.....</b>	<b>14</b>
<b>II. Présentation de l'université Mouloud Mammeri de Tizi-Ouzou.....</b>	<b>14</b>
II.1.Historique .....	14
<b>III. Présentation de la faculté G.E.I .....</b>	<b>15</b>
III .1. L'organigramme de la faculté de GEI .....	16
<b>IV. Présentation du département d'informatique .....</b>	<b>16</b>
IV.1. Formations .....	16
IV.2. Post de Graduation .....	17
IV.3. L'organigramme du département d'informatique.....	17
IV.4 Infrastructure et personnel.....	18
IV.4.1.Infrastructure .....	18
IV.4.2.Personnel.....	18
<b>V. Etude des postes de travail .....</b>	<b>19</b>
V.1 Définition d'un poste de travail .....	19
V.2 Les postes du travail étudiés.....	19
V.3 Fiche d'analyse des postes.....	20
<b>VI. Etude des documents.....</b>	<b>22</b>
VI.1. Liste des documents .....	22
<b>VI. Fiches d'analyse des documents.....</b>	<b>23</b>
<b>VIII. Diagramme des flux .....</b>	<b>26</b>

VIII.1. Définition.....	26
VIII.2. Le formalisme graphique .....	26
VIII.3. Acteurs du domaine d'étude.....	27
VIII.4. Représentation du diagramme des flux existant .....	27
<b>IX. Etude de la codification.....</b>	<b>28</b>
IX.1. Définitions .....	28
IX.2. Caractéristiques d'une codification .....	28
IX .3. Type de codification .....	28
<b>X. Problématique et objectif de cette étude.....</b>	<b>31</b>
X.1. Problématique .....	31
X.2. Objectifs de cette étude .....	31
<b>XI. Conclusion.....</b>	<b>32</b>

Chapitre III : Etude de l'existant

<b>I. Introduction .....</b>	<b>33</b>
<b>II. Présentation UML .....</b>	<b>33</b>
II.1. Définition .....	33
II.2. La démarche suivie .....	33
<b>III. Phase d'analyse .....</b>	<b>35</b>
III. 1. Introduction .....	35
III.2. Spécification des besoins .....	35
III.3. Identification des acteurs de l'application .....	36
III.4. Représentation des cas d'utilisations .....	36
III.4.2. Spécification des tâches pour chaque acteur .....	36
III.4.3. Spécification des scénarios .....	38
III.4 .4 Description des cas d'utilisation avec des scénarios .....	41
III.4.5 Diagrammes des cas d'utilisations .....	44

<b>III.5. Les Diagrammes de séquence.....</b>	<b>44</b>
<b>III.5.1. Diagramme de séquence cas d'utilisation « Authentification utilisateur ....</b>	<b>44</b>
<b>III.5.2. Diagramme de séquence cas d'utilisation « Ajouter formation » .....</b>	<b>45</b>
<b>III.5.3. Diagramme de séquence du cas d'utilisation « Modifier Formation » .....</b>	<b>46</b>
<b>III.5.4. Diagramme de séquence du cas d'utilisation « Supprimer Formation » .....</b>	<b>48</b>
<b>III.5.5. Diagramme de séquence cas d'utilisation « Ajouter enseignant » .....</b>	<b>49</b>
<b>III.5.6. Diagramme de séquence du cas d'utilisation « Afficher enseignant .....</b>	<b>50</b>
<b>III.5.7. Diagramme de séquence du cas d'utilisation « Saisir disponibilité ».....</b>	<b>51</b>
<b>III.5.8. Diagramme de séquence du cas d'utilisation « Changer le mot de passe » .....</b>	<b>52</b>
<b>III.6. Diagrammes d'activité .....</b>	<b>53</b>
<b>III.6.1. Diagramme d'activité du cas d'utilisation « S'authentifier » .....</b>	<b>53</b>
<b>III.6.2 : Diagramme d'activité du cas d'utilisation « Ajouter Formation » .....</b>	<b>54</b>
<b>III.6.3: Diagramme d'activité du cas d'utilisation «Modification d'une formation » .....</b>	<b>55</b>
<b>III.6.4: Diagramme d'activité du cas d'utilisation «Suppression d'une formation » .....</b>	<b>56</b>
<b>III.6.5: Diagramme d'activité du cas d'utilisation « Ajouter Enseignant » .....</b>	<b>57</b>
<b>III.6.6: Diagramme d'activité du cas d'utilisation « Afficher Enseignant » .....</b>	<b>58</b>
<b>III.6.7: Diagramme d'activité du cas d'utilisation « Saisir disponibilité ».....</b>	<b>59</b>
<b>III.6.8. Diagramme d'activité du cas d'utilisation « Gestion du compte » .....</b>	<b>60</b>

<b>IV.</b>	<b>Phase de Conception .....</b>	<b>61</b>
<b>IV.1.</b>	<b>Diagrammes des classes .....</b>	<b>61</b>
<b>IV. 2.</b>	<b>Règles de gestion .....</b>	<b>61</b>
<b>IV.3.</b>	<b>Le modèle logique .....</b>	<b>61</b>
<b>IV.3.1</b>	<b>Définition d'un schéma relationnel .....</b>	<b>61</b>
<b>IV.3.2</b>	<b>Règles de passage du diagramme de classe vers le modèle relationnel.....</b>	<b>62</b>
<b>IV.4.</b>	<b>Modèle physique de données.....</b>	<b>64</b>
<b>V.</b>	<b>Conclusion.....</b>	<b>66</b>

**Chapitre IV : Réalisation**

<b>I.</b>	<b>Introduction .....</b>	<b>67</b>
<b>II.</b>	<b>Les outils de développement.....</b>	<b>67</b>
<b>II.1.</b>	<b>Le serveur Web Apache Tomcat .....</b>	<b>67</b>
<b>II.2.</b>	<b>Hibernate Framework.....</b>	<b>68</b>
<b>II.3.</b>	<b>EJB .....</b>	<b>68</b>
<b>II.4.</b>	<b>JDBC .....</b>	<b>69</b>
<b>II.5.</b>	<b>JSF .....</b>	<b>69</b>
<b>II.6.</b>	<b>NetBeans .....</b>	<b>70</b>
<b>II.7.</b>	<b>SGBDR (Oracle .....</b>	<b>70</b>
<b>III.</b>	<b>Les langages de programmation.....</b>	<b>71</b>
<b>III.1.</b>	<b>Java.....</b>	<b>71</b>
<b>III.2.</b>	<b>Java EE .....</b>	<b>72</b>
<b>III.3.</b>	<b>SQL.....</b>	<b>72</b>
<b>III.4.</b>	<b>XHTML.....</b>	<b>73</b>
<b>IV.</b>	<b>Configuration des différentes parties de notre application .....</b>	<b>73</b>
<b>IV.1.</b>	<b>Configuration de NetBeans .....</b>	<b>73</b>
<b>IV.1.1.</b>	<b>Création d'un projet.....</b>	<b>73</b>

IV.1.2. Etablissement de la connexion .....	74
IV.2. Configuration de Hibernate .....	74
<b>V. Présentation de quelques interfaces de notre application.....</b>	<b>75</b>
V.1. La page d'accueil .....	75
V.2. La Page d'authentification .....	75
V.3. L'espace administrateur .....	76
V.4. L'espace enseignant.....	76
V.5. Page Gestion Formation.....	77
V.6. : page ajouter Formation .....	77
V.7. page modifier Formation .....	78
V.8. page supprimer Formation.....	79
V.9. Page Gestion Enseignant.....	80
V.10. Page ajouter Enseignant.....	81
V.11. Page afficher Enseignant.....	82
V.12. Page Gestion Disponibilite Enseignant .....	82
V.13. Page saisie disponibilite Enseignant .....	83
V.14. Page Consulter Disponibilite Enseignant .....	84
<b>VI. Conclusion .....</b>	<b>84</b>
<b>Conclusion générale .....</b>	<b>85</b>
<b>Annexe .....</b>	<b>86</b>

# Liste des Figure

## Chapitre I : « JavaEE »

<b>Figure II.4:</b> Architecture JavaEE.....	<b>3</b>
<b>Figure II.8.3:</b> Architecture JSF.....	<b>7</b>
<b>Figure III.2.2. :</b> Architecture de Hibernate.....	<b>11</b>

## Chapitre II : « Etude de L'existant »

<b>Figure III.1:</b> l'organigramme de la faculté de Génie Electrique et d'Informatique.....	<b>16</b>
<b>Figure IV.3:</b> l'organigramme du département d'Informatique	<b>17</b>
<b>Figure VIII:</b> le schéma du diagramme des flux.....	<b>26</b>
<b>Figure VIII.4 :</b> le diagramme des flux existant.....	<b>27</b>

## Chapitre III : « Analyse et Conception

<b>Figure II.2 :</b> Démarche de modélisation de l'application.....	<b>34</b>
<b>Figure III.5.1 :</b> Diagramme de séquence du cas d'utilisation « Authentification utilisateur» .....	<b>45</b>
<b>Figure III.5.2.:</b> Diagramme de Séquence du cas d'utilisation « Ajouter formation » .....	<b>46</b>
<b>Figure III.5.3.</b> Diagramme de séquence du cas d'utilisation « Modifier formation» .....	<b>47</b>
<b>Figure III.5.4 :</b> Diagramme de séquence du cas d'utilisation « Supprimer formation ».....	<b>48</b>
<b>Figure III.5.5</b> Diagramme de séquence du cas d'utilisation « Ajouter enseignant ».....	<b>49</b>
<b>Figure III.5.6 :</b> Diagramme de séquence du cas d'utilisation « Afficher enseignant».....	<b>50</b>
<b>Figure III.5.7 :</b> Diagramme de séquence du cas d'utilisation « Saisir disponibilité».....	<b>51</b>
<b>Figure III.5.8</b> Diagramme de séquence du cas d'utilisation « Changer le mot de passe».....	<b>52</b>
<b>Figure III.6.1:</b> Diagramme d'activité du cas d'utilisation « S'authentifier ».....	<b>53</b>

<b>Figure III.6.2 :</b> Diagramme d'activité du cas d'utilisation « Ajouter Formation ».....	<b>54</b>
<b>Figure III.6.3:</b> Diagramme d'activité du cas d'utilisation «Modification d'une formation ».....	<b>55</b>
<b>Figure III.6.4:</b> Diagramme d'activité du cas d'utilisation «Suppression d'une formation » .....	<b>56</b>
<b>Figure III.6.5:</b> Diagramme d'activité du cas d'utilisation « Ajouter Enseignant» .....	<b>57</b>
<b>Figure III.6.6:</b> Diagramme d'activité du cas d'utilisation « Afficher Enseignant» .....	<b>58</b>
<b>Figure III.6.7:</b> Diagramme d'activité du cas d'utilisation « Saisir disponibilité» .....	<b>59</b>
<b>Figure III.6.8.</b> Diagramme d'activité du cas d'utilisation « Gestion du compte » .....	<b>60</b>

**Chapitre IV : « Réalisation »**

<b>Figure II.1 :</b> l'interface Apache Tomcat.....	<b>67</b>
<b>Figure II.2:</b> L'interface de Hibernate.....	<b>68</b>
<b>Figure II.4 :</b> L'interface de NetBeans.....	<b>70</b>
<b>Figure II.5 :</b> Interface d'Oracle.....	<b>71</b>
<b>Figure III.2:</b> L'interface de JEE.....	<b>72</b>
<b>Figure V.1 :</b> page d'accueil.....	<b>75</b>
<b>Figure V.2 :</b> page d'authentification.....	<b>75</b>
<b>Figure IV.3 :</b> L'espace administrateur.....	<b>76</b>
<b>Figure V.4 :</b> L'espace enseignant.....	<b>76</b>
<b>Figure V.5 :</b> Page Gestion Formation.....	<b>77</b>
<b>Figure V.6 :</b> page ajouter Formation.....	<b>78</b>
<b>Figure V.7:</b> page modifier Formation.....	<b>79</b>
<b>Figure V.8 :</b> page supprimer Formation.....	<b>80</b>
<b>Figure V.9 :</b> Page Gestion Enseignant.....	<b>80</b>
<b>Figure V.10 :</b> page ajouter Enseignant.....	<b>81</b>

<b>Figure V.11</b> : page afficher Enseignant.....	<b>82</b>
<b>Figure V.12</b> : page Gestion Disponibilite Enseignant.....	<b>82</b>
<b>Figure V.13</b> : page saisie disponibilite Enseignant.....	<b>83</b>
<b>Figure V.14</b> : page Consulter disponibilite Enseignant.....	<b>84</b>

# Liste des tableaux

## Chapitre I : JavaEE

<b>Tableau II.9 : API JEE.....</b>	<b>8</b>
------------------------------------	----------

## Chapitre III : Analyse et Conception

<b>Tableau III.4.2 : Identification des cas d'utilisation.....</b>	<b>37</b>
<b>Tableau III.4.3 : Spécification des scénarios.....</b>	<b>40</b>

# Introduction générale

L'informatique, science de traitement automatique de l'information par des machines; elle est devenue un outil de travail indispensable demandée par des différents organismes, à fin d'aboutir à une gestion meilleur de leur application. Elle est devenue un instrument indispensable de toute bonne administration dans différents domaines à l'instar de l'enseignement, le commerce, la santé...

Vu la quantité importante d'information manipulées continuellement dans ces secteurs qui est encore manuel, d'où elle est confrontée à un bon nombre de difficultés telles que :

- La difficulté de saisir au moment voulu ce qui engendre un retard dans le traitement de l'information.
- La perte d'information et difficulté de recherche et d'accès aux documents.
- Travaux manuels élevés, lourds et pénibles, ce qui engendre la perte d'information.

Les difficultés rencontrées au quotidien dans la gestion, du nombre important de document et dans la communication au sein de domaine d'enseignement pourrait être contournée par l'utilisation de l'informatique.

C'est dans cette optique que s'inscrit notre projet, gestion des activités pédagogiques au niveau du département d'informatique de l'ummo, visant à assurer des meilleures conditions du travail pour le corps pédagogiques, soit un gain de temps, un meilleur rendement, saisie et impression des documents très facile et rapide ,la récupération de celles-ci en cas de perte, organisation et bon fonctionnement de département.

Pour une bonne organisation de notre travail, nous avons adopté la structure chapitrée suivante :

- Le premier chapitre est consacré pour la présentation de langage de programmation javaEE.
- Le deuxième chapitre est consacré pour la présentation de l'organisme d'accueil « Département d'informatique » situé à Tizi Ouzou.
- Le troisième chapitre : retracera les différentes étapes pour l'implémentation de notre application « Analyse et conception ».
- Le quatrième chapitre intitulé « Réalisation »: porte sur la réalisation de notre application dont on définira et présentera ses différentes interfaces.

# Chapitre I : Java EE

## I. Introduction :

Dans le monde des technologies de l'information, les applications d'entreprise doivent être conçues, construits et fabriqués pour moins d'argent, avec une plus grande vitesse et avec moins de ressources. Sun (le concepteur de Java) a mis en place un ensemble de technologies pour réaliser des applications Web. Ces technologies sont regroupées sous le nom J2EE (Java 2 Entreprise Edition). Depuis la version 5, le chiffre 2 a disparu (ambiguïté avec la version)

Il est apparu à la fin des années 90 destiné aux gros (très gros) systèmes d'entreprises. Il a pour objectif de crédibiliser java pour les applications d'entreprise. Java EE ne remplace en aucun cas J2SE mais elle constitue une extension de J2SE .Elle permet un développement d'applications qui vont s'exécuter sur un serveur d'applications.

## II. JEE :

### II.1. Définition javaEE : [1]

Java Entreprise Edition, ou javaEE (anciennement J2EE), est une spécification pour la technique Java d'Oracle plus particulièrement destinée aux applications d'entreprise. Ces applications sont considérées dans une approche multi-niveaux. En simplifiant, il est possible de représenter Java EE comme un ensemble de spécifications d'API, une architecture, une méthode de packaging et de déploiement d'applications et la gestion d'applications déployées sur un serveur compatible Java.

### II.2. Fonctionnement interne : [2]

Le langage Java, sur lequel les bibliothèques Java EE sont utilisées, met à disposition un Compilateur et une machine virtuelle (JVM – Java Virtuel Machine) qui se charge de créer un environnement standard pour le lancement de l'application sur tout type de système opérationnel. Le compilateur compile le code source est produit le byte code, soit un code intermédiaire qui sera ensuite lu par la machine virtuelle Java. Chaque système opérationnel majeur possède une JVM expressément codée.

### II.3. Les éléments de la JavaEE : [1]

- Une plate-forme (Java EE Platform), pour héberger et exécuter les applications, incluant outre Java SE des bibliothèques logicielles additionnelles du Java Development Kit (JDK) .Elle est basée sur des spécifications donc les projets sont portables sur n'importe quel serveur d'applications conforme (Tomcat, JBoss, WebSphere...).
- Une suite de tests (Java EE Compatibility Test Suite) pour vérifier la compatibilité.
- Une réalisation de référence (Java EE Reference Implementation) dénommée GlassFish .

- Un catalogue de bonnes pratiques (Java EE BluePrints).

#### II.4. Architecture JavaEE: [3]

L'architecture d'une application se découpe idéalement en au moins trois tiers :

- la partie cliente : c'est la partie qui permet le dialogue avec l'utilisateur.
- la partie serveur : c'est la partie qui encapsule les traitements.
- la partie donnée : c'est la partie qui stocke les données

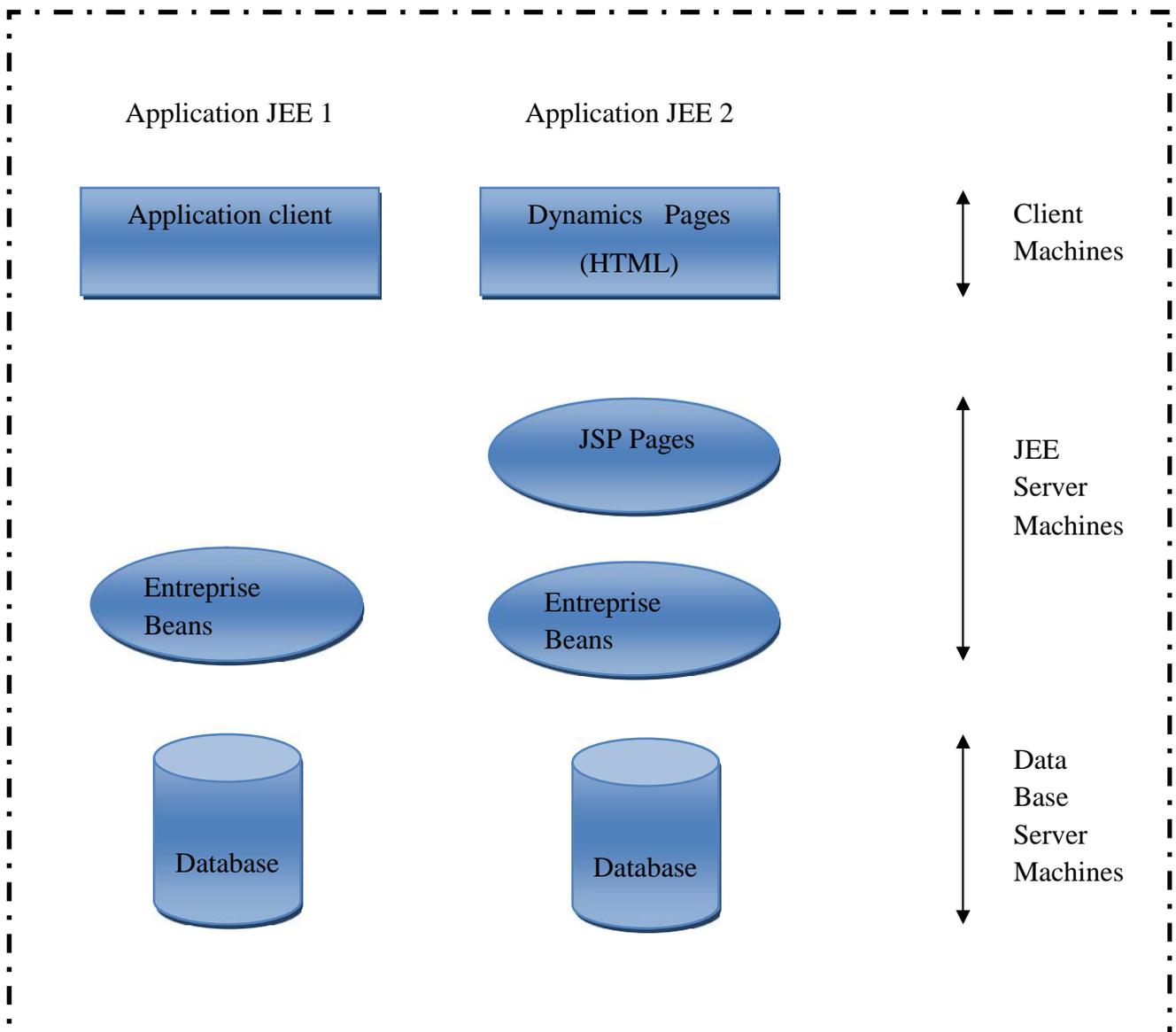


Figure II.4 : Architecture JavaEE

## II.5. Les services JEE: [4]

### ➤ Services d'infrastructures :

- JDBC (Java Database Connectivity) : API d'accès aux SGBDs
- JNDI (Java Naming and Directory Interface) : service de noms (annuaire) de référencement des objets
- JTA/JTS (Java Transaction API/Service) : service de gestion des transactions distribuées
- JCA (J2EE Connector Architecture) est une API de connexion au système d'information de l'entreprise.
- JMX (Java Management Extension) fournit des extensions permettant de développer des applications web de supervision d'applications.

### ➤ Services de communication :

- JAAS (Java Authentication and Authorization Service) est une API de gestion de l'authentification et des droits d'accès.
- Java Mail est une API permettant l'envoi de courrier électronique.
- JMS (Java Message Service) service de gestion des messages asynchrones (appelées MOM pour Middleware Object Message) entre applications.
- RMI-IIOP est une API permettant la communication synchrone entre objets.

## II.6. L'environnement d'exécution : [5]

J2EE propose des spécifications pour une infrastructure dans laquelle s'exécutent les composants. Ces spécifications décrivent les rôles de chaque élément et précisent un ensemble d'interfaces pour permettre à chacun de ces éléments de communiquer.

Ceci permet de séparer les applications et l'environnement dans lequel elles s'exécutent. Les spécifications précisent à l'aide des API un certain nombre de fonctionnalités que doit implémenter l'environnement d'exécution. Ces fonctionnalités permettent aux développeurs de se concentrer sur la logique métier.

Pour exécuter ces composants de natures différentes, J2EE définit des conteneurs pour chacun d'eux. Il définit pour chaque composant des interfaces qui leur permettront de

dialoguer avec les composants lors de leur exécution. Les conteneurs permettent aux applications d'accéder aux ressources et aux services en utilisant les API.

Les appels aux composants se font par des clients en passant par les conteneurs. Les clients n'accèdent pas directement aux composants mais sollicitent le conteneur pour les utiliser.

## II.7. Les Conteneurs JEE : [5]

Les conteneurs assurent la gestion du cycle de vie des composants qui s'exécutent en eux. Les conteneurs fournissent des services qui peuvent être utilisés par les applications lors de leur exécution.

Il existe plusieurs conteneurs définis par J2EE:

- conteneur web : pour exécuter les servlets et les JSP
- conteneur d'EJB : pour exécuter les EJB
- conteneur client : pour exécuter des applications sur les postes qui utilisent des composants J2EE

Les serveurs d'applications peuvent fournir un conteneur web uniquement (exemple : Tomcat) ou un conteneur d'EJB uniquement (exemple : JBoss, Jonas, ...) ou les deux (exemple : Websphere, Weblogic, ...).

Pour déployer une application dans un conteneur, il faut lui fournir deux éléments :

- l'application avec tous les composants (classes compilées, ressources ...) regroupés dans une archive ou module. Chaque conteneur possède son propre format d'archive.
  - un fichier descripteur de déploiement contenu dans le module qui précise au conteneur des options pour exécuter l'application
- JSP(JavaServer Pages) : est une technique basée sur Java qui permet aux développeurs de créer dynamiquement du code HTML, XML ou tout autre type de page web. Cette technique permet au code Java et à certaines actions prédéfinies d'être ajoutés dans un contenu statique
- Servlets : Technologie Java utilisé pour effectuées des traitements coté serveur en réponse aux requêtes provenant en général de poste clients distants. Bien que les Servlets puissent répondre à n'importe quel type de requête, elles sont généralement employées pour répondre à des requêtes de type HTTP et qui permettent de retourner dynamiquement des pages HTML.

- EJB : Est une architecture de composants logiciels côté serveur pour la plateforme de développement Java EE. Cette architecture propose un cadre pour créer des composants distribués (c'est-à-dire déployés sur des serveurs distants) écrit en langage de programmation Java hébergés au sein d'un serveur applicatif.
  
- JDBC (Java DataBase Connectivity) : est une interface de programmation pour les programmes utilisant la plateforme Java.  
Elle permet aux applications Java d'accéder par le biais d'une interface commune à des sources de données pour lesquelles il existe des pilotes JDBC.

## **II.8. JSF :**

### **II.8.1.Présentation: [6]**

Java Server Faces (JSF) est un framework web MVC qui simplifie la construction d'interfaces utilisateur (UI) pour les applications basées sur le serveur en utilisant des composants de l'interface utilisateur réutilisables dans une page. JSF fournit le service pour se connecter des widgets de l'interface utilisateur avec des sources de données et aux gestionnaires d'événements côté serveur. La spécification JSF définit un ensemble de composants d'interface utilisateur standard et fournit une interface de programmation d'application (API) pour le développement de composants. JSF permet la réutilisation et l'extension des composants d'interface utilisateur standard existants. Le but de JSF est d'accroître la productivité des développeurs dans le développement des interfaces utilisateur tout en facilitant leur maintenance).

### **II.8.2. Modèle de composants JSF UI : [6]**

JSF fournit aux développeurs la capacité de créer des applications Web à partir de collections de composants d'interface utilisateur qui peuvent se rendre de différentes façons pour les types de clients multiples (par exemple navigateur HTML , concevoir sans fil ou WAP ) . Comme il fournit des bibliothèques de base (ensemble de composants d'interface utilisateur de base - des éléments d'entrée HTML standard).

### **II.8.3. Architecture JSF: [6]**

La technologie JSF est un cadre pour le développement, la construction de composants d'interface côté serveur User et de les utiliser dans une application web. La technologie JSF est basée sur l'architecture Model View Controller (MVC) pour séparer la logique de la présentation.

### ➤ Le Design Pattern MVC

Le modèle de conception MVC conçoit une application à l'aide de trois modules distincts :

- Le **modèle** représente le cœur (algorithmique) de l'application : traitements des données, interactions avec la base de données, etc. Il décrit les données manipulées par l'application. Il regroupe la gestion de ces données et est responsable de leur intégrité.
  - La **vue** fait l'interface avec l'utilisateur. Sa première tâche est d'afficher les données qu'elle a récupérées auprès du modèle. Sa seconde tâche est de recevoir tous les actions de l'utilisateur (clic de souris, sélection d'une entrées, boutons, ...). Ses différents événements sont envoyés au contrôleur.
  - Le **contrôleur** est chargé de la synchronisation du modèle et de la vue. Il reçoit tous les événements de l'utilisateur et enclenche les actions à effectuer. Si une action nécessite un changement des données, le contrôleur demande la modification des données au modèle et ensuite avertit la vue que les données ont changé pour que celle-ci se mette à jour.
- Flux de traitement : lorsqu'un client envoie une requête à l'application :
    - la requête envoyée depuis la vue est analysée par le contrôleur (par exemple un clic de souris pour lancer un traitement de données) ;
    - le contrôleur demande au modèle approprié d'effectuer les traitements et notifie à la vue que la requête est traitée ;
    - la vue notifiée fait une requête au modèle pour se mettre à jour (par exemple affiche le résultat du traitement via le modèle).



Figure II.8.3: Architecture JSF

**II.8.4. Avantage JSF: [7]**

- Gérer l'état de l'interface utilisateur dans les demandes de plusieurs serveurs
- Faire le transfert des données entre les composants de l'interface utilisateur
- Réduit l'effort dans la création et la maintenance des applications qui fonctionnent sur un serveur d'applications Java et rendront l'application UI à un client cible.
- une séparation de la couche présentation des autres couches (MVC)
- un mapping entre l'objet et l'HTML
- un ensemble de composants riches et réutilisables
- une liaison simple entre les actions côté client de l'utilisateur (event listener) et le code Java côté serveur
- Création de nouveaux composants graphiques
- JSF peut être utilisé pour générer autre chose que du HTML (XUL, XML, WML, ...)

**II.9. Les API JEE : [5]**

API (Application Programming Interface) (Interface Applicative de Programmation). Les API sont un moyen efficace de faire communiquer entre elles deux applications. J2EE regroupe un ensemble d'API pour le développement d'applications d'entreprise. Le tableau suivant représente quelque API ainsi que leur rôles:

API	Rôle
Entreprise Java Bean (EJB)	Composants serveurs contenant la logique métier
Servlets	Composants basés sur le concept C/S pour ajouter des fonctionnalités à un serveur. Pour le moment, principalement utilisé pour étendre un serveur web
Java Server Pages (JSP)	
Java Naming and Directory Interface (JNDI)	Accès aux services de nommage et aux annuaires d'entreprises
Java Database Connectivity (JDBC)	Accès aux bases de données. J2EE intègre une extension de cette API
Java Transaction API (JTA) Java Transaction Service (JTS)	Support des transactions
Java Server Faces (JSF)	

**Tableau II.9 : API JEE**

**II.10. Java EE et la sécurité : [2]**

Vu la complexité des ces serveurs d'application il est très difficile d'assurer la protection contre les menaces et les intrusions dans ce type d'architecture, donc il faut faire attention aux risques que peut produire un serveur mal configuré et essayer de fermer les portes d'entrées que peuvent laisser de telles serveurs pour diminuer les risques d'attaque et protéger ce type d'application.

**II.11. Objectifs Java EE : [8]**

- Faciliter le développement de nouvelles applications à base de composants
- Intégration avec les systèmes d'information existants
- Support pour les applications « critiques » de l'entreprise
- Disponibilité, tolérance aux pannes, montée en charge, sécurité ...
- une architecture d'applications basée sur les composants qui permet un découpage de l'application et donc une séparation des rôles lors du développement
- la possibilité de s'interfacer avec le système d'information existant grâce à de nombreuses API : JDBC, JNDI, JMS, JCA ...
- la possibilité de choisir les outils de développement et le ou les serveurs d'applications utilisés qu'ils soient commerciaux ou libres
- grande flexibilité dans le choix de l'architecture de l'application en combinant les différents composants.

**III. Le Framework :****III.1. Présentation : [9]**

Le terme framework est fréquemment utilisé dans des contextes différents mais il peut être traduit par cadre de développement.

Les framework se présentent sous diverses formes, qui peuvent inclure tout ou partie des éléments suivants :

- un ensemble de classes généralement regroupées sous la forme de bibliothèques pour proposer des services plus ou moins sophistiqués
- un cadre de conception reposant sur les design patterns pour proposer tout ou partie d'un squelette d'application
- des recommandations sur la mise en œuvre et des exemples d'utilisation
- des normes de développement
- des outils facilitant la mise en œuvre

L'objectif d'un framework est de faciliter la mise en œuvre des fonctionnalités de son domaine d'activité. Il doit permettre au développeur de se concentrer sur les tâches spécifiques de l'application à développer plutôt que sur les tâches techniques récurrentes telles que :

- l'architecture de base de l'application
- l'accès aux données
- l'internationalisation
- la journalisation des événements
- la sécurité (authentification et gestion des rôles)
- le paramétrage de l'application

La mise en œuvre d'un framework permet notamment :

- de capitaliser le savoir-faire
- d'accroître la productivité des développeurs une fois le framework pris en main
- d'homogénéiser les développements des applications en assurant la réutilisation de composants fiables
- donc de faciliter la maintenance notamment évolutive des applications
- Cependant, cette mise en œuvre peut se heurter à certaines difficultés :
- le temps de prise en main du framework par les développeurs peut être plus ou moins long en fonction de différents facteurs (complexité du framework, richesse de sa documentation, expérience des développeurs, ...)
- les évolutions du framework qu'il faut répercuter dans les applications existantes

## **III.2. Hibernate :**

### **III.2.1. Définition : [10]**

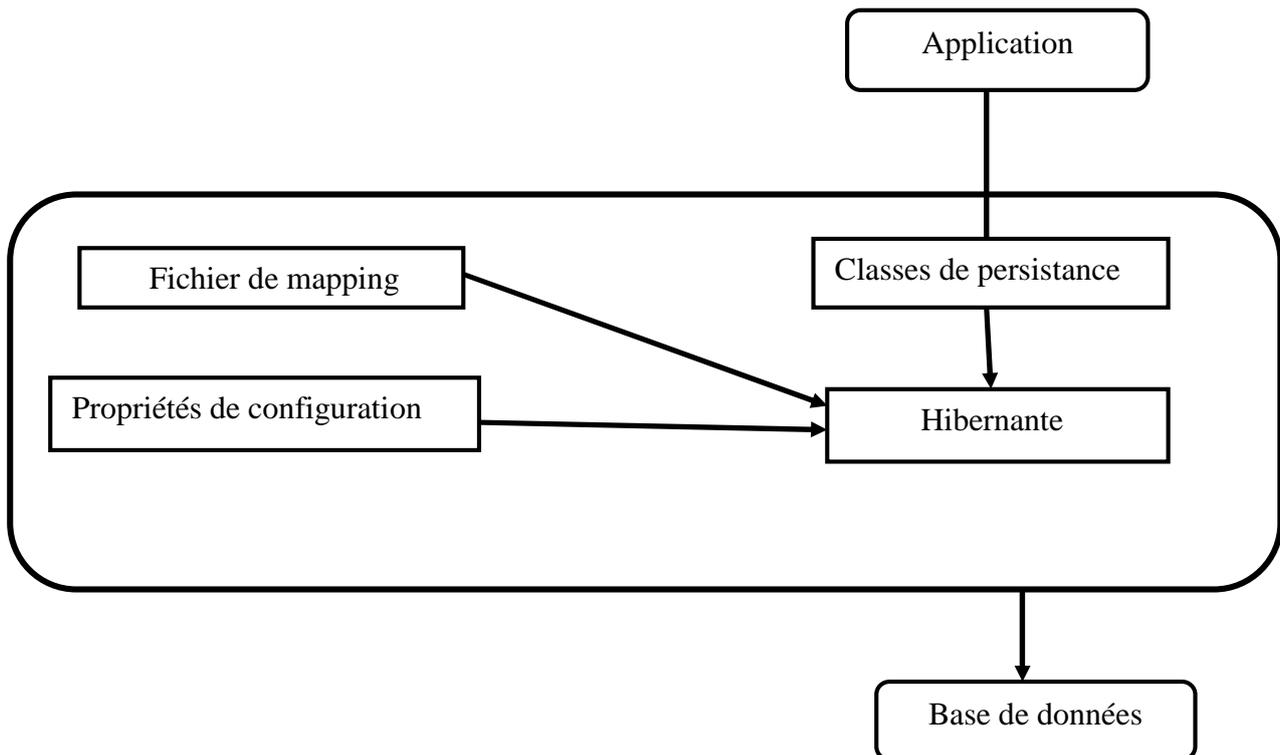
Hibernate est un framework open source gérant la persistance des objets en base de données relationnelle. Il est adaptable en termes d'architecture, il peut donc être utilisé aussi bien dans un développement client lourd, que dans un environnement web léger de type Apache Tomcat ou dans un environnement J2EE complet : Websphere, JBoss Application Server et Oracle Weblogic Server.

Hibernate apporte une solution aux problèmes d'adaptation entre le paradigme objet et les SGBD en remplaçant les accès à la base de données par des appels à des méthodes objet de haut niveau.

**III.2.2. Architecture : [11]**

Les éléments nécessaires :

- Classe de persistance : une classe de JavaBean pour l'encapsulation des données d'une table
- Correspondance entre la classe et la table : fichier de mapping
- Propriétés de configuration : Informations permettant la connexion a la BDD.



**Figure III.2.2.** : Architecture de Hibernate.

### III.2.3. Configuration objet:[12]

L'objet de configuration est le premier objet de Hibernate vous créez dans toute application Hibernate ; Il est généralement créé une seule fois lors de l'application de l'initialisation. Elle représente un fichier de configuration ou des propriétés requises par la mise en veille prolongée.

L'objet de configuration fournit deux composants clés :

- Base de données de connexion : Ceci est géré par un ou plusieurs fichiers de configuration soutenu par Hibernate. Ces fichiers sont hibernate.properties et hibernate.cfg.xml.
- Configuration de mappage de classe : Ce composant crée la connexion entre le Java classes et tables de base de données.

### III.2.4. Object SessionFactory : [12]

Objet de configuration est utilisé pour créer un objet SessionFactory qui à son tour configure Hibernate pour l'application en utilisant le fichier de configuration fourni et permet une session objet à instancier. Le SessionFactory est un objet thread-safe et utilisé par tous les fils d'une application.

Le SessionFactory est un objet lourd ; il est généralement créé lors du démarrage de l'application et conservée pour un usage ultérieur. Vous auriez besoin d'un objet SessionFactory par base de données en utilisant un fichier de configuration séparé. Donc, si vous utilisez plusieurs bases de données, alors que vous auriez pour créer plusieurs objets SessionFactory.

### III.2.5. Objet de la session : [12]

Une session est utilisée pour obtenir une connexion physique avec une base de données. L'objet de session est léger et conçu pour être instancié à chaque fois qu'une interaction est nécessaire à la base de données. Les objets persistants sont enregistrés et récupérés à travers un objet Session.

Les objets de session ne doivent pas être maintenues ouverts pendant une longue période, ils devraient être créés et les détruire au besoin.

### III.2.6. Transaction Object : [12]

Une transaction représente une unité de travail avec la base de données et la plupart des supports de SGBDR.

**III.2.7. Query Object : [12]**

Objets de requête utilisent SQL ou Hibernate Query Language (HQL) chaîne pour récupérer des données à partir des objets de base de données créés. Une instance de requête est utilisée pour lier les paramètres de la requête, limiter le nombre de résultats retournés par la requête, et enfin pour exécuter la requête.

**III.2.8. Critères Object : [12]**

Objets de critères utilisés pour créer et exécuter des requêtes par critère orientés objet pour récupérer les objets.

**IV. Conclusion :**

Au cours de ce chapitre, nous avons présenté le fonctionnement, les éléments, l'architecteur javaEE, ainsi que les services de la Java EE et l'environnement d'exécution, comme on a présenté le framework Hibernate ainsi que son architecture. Ces notions seront utilisées pour le développement de notre projet.

Dans le chapitre qui suit nous allons présenter l'organisme d'accueil.

## **Chapitre II : Etude de l'existant**

**I. Introduction :**

Au tout début, de la conception d'une application pour un organisme donné, il est important de connaître à la fois le terrain sur lequel on travaille et fixer les buts à atteindre. Cela dépend de l'étude de l'existant que nous allons traiter au cours de ce chapitre. Elle permet de communiquer avec les utilisateurs à fin de mieux comprendre leurs besoins et de réaliser un logiciel adéquat qui répond à toutes les attentes, avec une bonne étude des postes de travail, des documents et registres.

**II. Présentation de l'université Mouloud Mammeri de Tizi-Ouzou :****II.1.Historique : [13]**

L'université de Tizi-Ouzou a ouvert ses portes en 1977, elle avait alors le statut de centre universitaire et était implantée à Oued Aissi à 6km de sa résidence actuelle (l'axe de la nouvelle ville).

Les modes de fonctionnement de l'université, de 1977 à ce jour sont les suivants :

- **De 1977 à 1984 :**

L'université avait le statut de centre universitaire dirigé par un directeur au plan administratif assisté par trois directeurs adjoints chargés respectivement de la scolarité, du post graduation et de la planification ainsi que d'un secrétaire général.

A leur tour, les directeurs adjoints étaient assistés des chefs de départements administratifs qui étaient assistés à leur tour par des chefs de divisions.

- **En 1985 :** Le centre universitaire fut dissous et les instituts pédagogiques qui le constituaient, ont été érigés en instituts nationaux d'enseignements supérieurs (I.N.E.S).

Les I.N.E.S étaient dirigés par des directeurs assistés chacun de trois directeurs adjoints chargés respectivement de la pédagogie, la recherche et de l'administration générale.

- **En1990 :**

Les I.N.E.S furent à leur tour dissous et érigés en instituts dépendants de l'université de Tizi-Ouzou.

L'université de Tizi-Ouzou, de 1977 à 1990, fonctionnait avec des instituts et départements pédagogiques.

- **En 1999 :**

Les instituts ont été dissous également et érigés en facultés .après le décret exécutif No98/32 du 02/12/1998, l'article stipule la création au sein de l'université de Tizi-Ouzou des facultés suivantes:

- Faculté de médecine
- Faculté des sciences.
- Faculté de droit.
- Faculté des lettres et sciences humains.
- Faculté de génie de la construction.
- Faculté de génie électrique et informatique.
- Faculté des sciences biologiques et agronomiques.
- Faculté des sciences économiques et sciences de gestion.

### **III. Présentation de la faculté G.E.I : [13]**

La faculté est une unité autonome d'enseignement et de recherche de l'université dans le domaine de la science et de la connaissance.

Elle est pluridisciplinaire, mais elle peut être le cas échéant créée autour d'une discipline dominante. Elle assure notamment :

- des enseignements de graduation et de post graduation.
- Des activités de recherche scientifique.

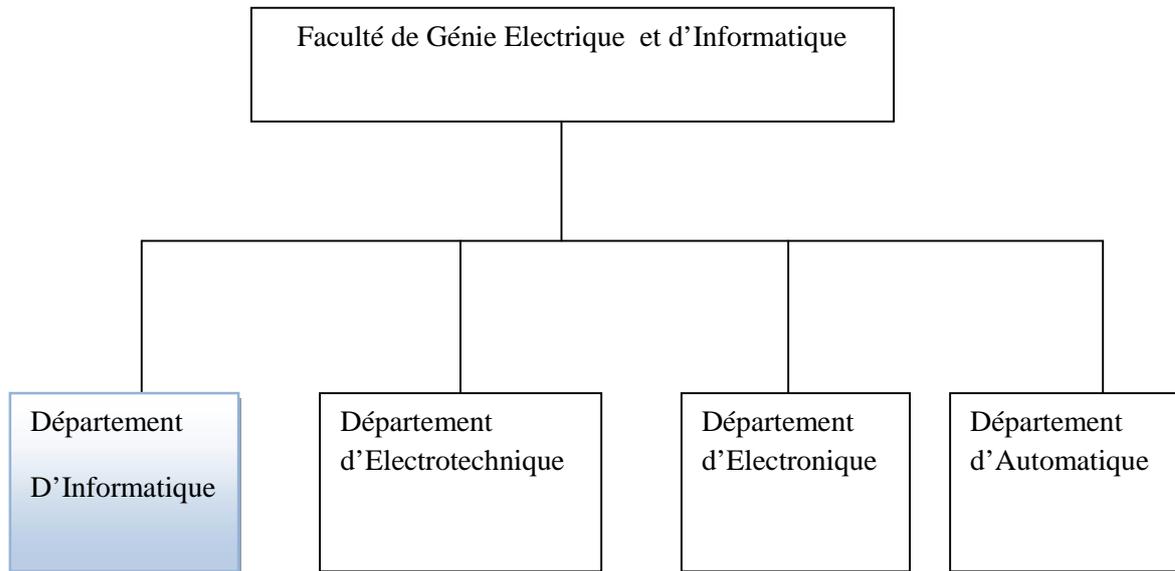
La faculté de G.E.I est une organisation à caractère de formation et de recherche, elle est composée des départements d'automatique, d'électronique, d'électrotechnique et d'informatique, dont elle assure la coordination des activités, et comporte une bibliothèque organisée en services et en sections.

La faculté à pour missions:

- Les enseignements de graduation et de post graduation.
- Les activités de recherche scientifique.
- Les actions de formation continue de perfectionnement et de recyclage.

### III .1 L'organigramme de la faculté de G.E.I :

La faculté de génie électrique et d'informatique est répartie comme suit :



**Figure III.1:** l'organigramme de la faculté de Génie Electrique et d'Informatique.

## IV. Présentation du département d'informatique : [14]

### IV.1.Formations :

Le département reçoit des étudiants ayant passés avec succès les deux semestres du domaine sciences et techniques et après études des fiches de vœux et sur la base des places pédagogiques disponibles, on trouve deux cycles qui sont les suivants :

1. Cycle Licence (3ans) :
  - Licence Informatique.
  - Licence Informatique et Multimédia.
  
2. Cycle Master (2ans):
  - Master Conduite de Projet Informatique.
  - Master Réseaux Mobilités et Systèmes Embarqués.
  - Master Systèmes Informatiques.
  - Master Ingénierie des Systèmes d'Information.

**IV.2.Post de Graduation :**

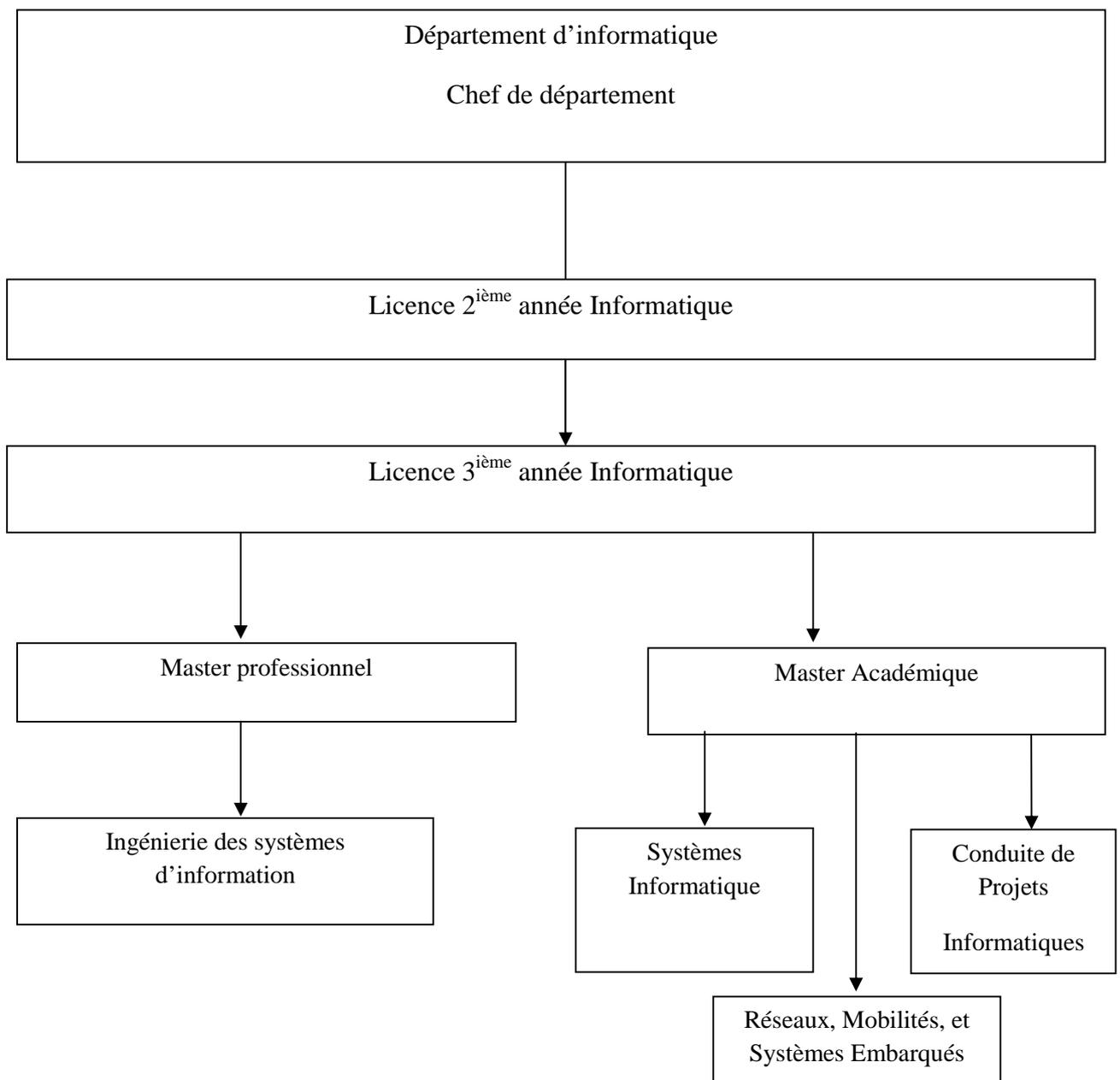
Le département assure plusieurs formations en doctorat LMD, doctorat en science .

1.Intelligence A artificielle et Systèmes d'Information.

2.Réseaux, Mobilité et Systèmes Embarqués.

**IV.3. L'organigramme du département d'informatique:**

Le Département informatique est organisé comme suit :



**Figure IV.3 :** L'organigramme du département d'Informatique.

**IV.4 Infrastructure et personnel :****IV.4.1. Infrastructure :**

le département d'informatique se compose de :

- 02 amphis de cours de place
- 12 salles de TD de place
- 11 Labo (salles de TP).

**IV.4.2. Personnel :**

## 1. L'équipe administrative :

- Le chef du département d'Informatique.
- Le chef du département d'Informatique adjoint à la pédagogie.
- Le chef du département d'Informatique adjoint à la PG.
- Le président du comité Scientifique du département d'Informatique.

## 2. le personnel Enseignant :

- 03 Professeurs (pr).
- 04 Maitres de Conférences Classe A (MCA).
- 06 Maitre de Conférences Classe B (MCB).
- 27 Maitres Assistants Classe A (MACA).
- 19 Maitres Assistants Classe B (MACB).

➤ Effectif : environ 60 enseignant.

## 3. Effectifs Etudiant (2013/2014) :

- Licence Informatique générale (L2) :187 étudiants.
- Licence Informatique générale (L3) :337 étudiants.
- Licence Informatique et Multimédias (L2) :18 étudiants.
- Licence Informatiques Multimédias (L3) :24 étudiants.
- Master Informatique Conduite de Projet Informatique (M1) :74 étudiants.
- Master Informatique conduite de Projet Informatique (M2) :73 étudiants.
- Master Informatique Système Informatique(M1) :72 étudiants.
- Master Informatique Système Informatique(M2) :58 étudiants.
- Master Informatique Ingénieries des Systèmes d'information(M1) :31 étudiants.
- Master Informatique Ingénieries des Systèmes d'information(M2) :58 étudiants.

➤ L'effectif d'étudiants en Licence Informatique : 566.

➤ L'effectif d'étudiants en Master Informatique : 430.

**V. Etude des postes de travail :**

Cette étude permet de savoir comment l'information circule entre les différents postes ainsi pour comprendre l'aspect organisationnel de ce Département.

**V.1 Définition d'un poste de travail : [15]**

Un poste de travail est la position d'une ou plusieurs personnes ayant un certain nombre d'opérations à exécuter en utilisant un ensemble d'informations.

**V.2 Les poste de travail étudiés :**

Cette étude permet de savoir comment est-ce que l'information circule entre les différents services ainsi comprendre l'aspect organisationnel de cette structure. On a :

- Le poste de chef de département adjoint: occupé par le chef de département adjoint qui est responsable de la gestion du département ;
- Le poste de l'agent de suivi des enseignements : occupé par le secrétaire qui est chargée d'assurer un bon fonctionnement pour le département.

On va créer une fiche d'analyse pour chaque poste afin d'étudier :

- Ces responsabilités .
- Ces tâches à accomplir .
- Les documents provenant à ce poste .
- Les documents diffusés par ce poste.

V.3 Fiche d'analyse des postes :

1. Poste de travail N°01 : Chef de département adjoint

<b>Fiche d'étude du poste N°01</b>	
<p><b>Désignation :</b> Chef de département adjoint</p> <p><b>Effectif :</b> 01</p> <p><b>Lieu d'implantation :</b> Faculté de Génie Electrique et d'Informatique</p>	
<p><b>Responsabilités :</b> -Diriger le département d'Informatique</p> <p>-Superviser et veiller au bon fonctionnement du département d'Informatique</p>	
<p><b>Tâches à accomplir :</b></p> <p>-Vérification des emplois du temps, les plannings d'examens de différentes épreuves.</p> <p>-Signature des emplois du temps, des plannings d'examens, et la convocation de surveillance.</p>	
<b>Document provenant à ce poste</b>	
<b>Désignation</b>	<b>Provenance</b>
<p>-Emplois du temps</p> <p>-Plannings des examens</p> <p>-Convocation de surveillance des examens.</p>	<p>-Agent de suivie des enseignements.</p> <p>-Agent de suivie des enseignements</p> <p>-Agent de suivie des enseignements</p>
<b>Documents diffusés par ce poste</b>	
<b>Désignation</b>	<b>Destination</b>
<p>-Emplois du temps</p> <p>-Plannings des examens</p> <p>-Convocation de surveillance des examens.</p>	<p>-Agent de suivie des enseignements.</p> <p>-Agent de suivie des enseignements</p> <p>-Agent de suivie des enseignements</p>

2. Poste de travail N°02 : Agent suivi des enseignements

<b>Fiche d'étude du poste N°02</b>	
<p><b>Désignation :</b> Agent suivi des enseignements  <b>Effectif :</b> 01  <b>Lieu d'implantation :</b> Département d'informatique</p>	
<p><b>Responsabilités :</b> suivi des enseignements et de l'évaluation</p>	
<p><b>Tâches à accomplir :</b>                      -Planification des emplois du temps, plannings d'examens puis les transmet au Chef de département adjoint pour signature.                      -Saisie et impression des emplois du temps et des plannings des examens.                      -affichage des emplois du temps et des plannings des examens.                      -Impression et envoi de la convocation de surveillance à l'enseignant.</p>	
<b>Document provenant à ce poste</b>	
<b>Désignation</b>	<b>Provenance</b>
-Emplois du temps (signé). -Plannings des examens (signé). -Convocation de surveillance des examens (signée).	-chef du département adjoint. -chef du département adjoint. -chef du département adjoint.
<b>Documents diffusés par ce poste</b>	
<b>Désignation</b>	<b>Destination</b>
-Emplois du temps (non signé). -Plannings des examens (non signé). -Convocation de surveillance des examens (non signée). -Convocation de surveillance des examens (signée).	-chef du département adjoint. -chef du département adjoint. -chef du département adjoint. -enseignant.

**VI. Etude des documents :**

L'étude des documents nous permet de savoir comment est-ce que les informations circulent entre les différents composants du champ d'étude et l'extérieur. On trouve les documents internes circulant entre les acteurs internes au champ d'étude et des documents externes permettant d'avoir des informations de l'extérieur ou de les donner aux acteurs externes au champ d'étude.

**VI.1. Liste des documents :**

<b>N°</b>	<b>Désignation</b>
<b>01</b>	Emploi du temps
<b>02</b>	Planning des examens
<b>03</b>	Convocation de surveillance des examens

## VII. Fiches d'analyse des documents :

## 1. Fiche d'analyse du document n° : 01

<b>Identification</b>			
<b>Code :</b> /			
<b>Désignation :</b> emplois du temps			
<b>Nature du document :</b> interne			
<b>Emetteur :</b> Chef de département adjoint.			
<b>Destinataire :</b> Agent de suivi des enseignements			
<b>Nombre d'exemplaire :</b> 01			
<b>Rôle :</b> communiquer les horaires des cours, td et TP aux étudiants et enseignants			
<b>Description</b>			
<b>Rubrique</b>	<b>Taille</b>	<b>Type</b>	<b>Observation</b>
-Nom du département	25	A	
-Niveau d'étude	15	AN	
-Cycle d'étude	10	A	
-Année d'étude	10	AN	AAAA /AAAA
-Jour	08	A	
-Horaires	11	AN	Ex: 08h00-09h30
-Module	20	AN	
-Nom de l'enseignant	20	A	
-nature de l'activité	06	A	Cours/TD/TP
-Lieux	04	AN	

## 2. Fiche d'analyse du document n° : 02

<b>Identification</b>			
<b>Code :</b> /			
<b>Désignation :</b> planning des examens			
<b>Nature du document :</b> interne			
<b>Emetteur :</b> Chef de département adjoint.			
<b>Destinataire :</b> Agent de suivi des enseignements			
<b>Nombre d'exemplaire :</b> 01			
<b>Rôle :</b> communiquer les dates et les modules et les salles d'examens.			
<b>Description</b>			
<b>Rubrique</b>	<b>Taille</b>	<b>Type</b>	<b>Observation</b>
-Nom du département	25	A	
-Date de l'examen	10	AN	
-Module	20	AN	Ex: CPI
-Horaire de l'examen	11	AN	Ex:8h30-10h
-Lieux	04	AN	
-Nom d'enseignant surveillant	20	A	
-Niveau d'étude	15	AN	
-Cycle d'étude	10	A	
-Spécialité	20	A	
-Jour	08	A	
-Année d'étude	10	AN	

## 3. Fiche d'analyse du document n° : 03

<b>Identification</b>			
<b>Code :</b> /			
<b>Désignation :</b> Convocation de surveillance			
<b>Nature du document :</b> interne			
<b>Emetteur :</b> Chef de département adjoint.			
<b>Destinataire :</b> Agent de suivi des enseignements			
<b>Nombre d'exemplaire :</b> 01			
<b>Rôle :</b> communiquer les dates et les modules et les salles d'examens aux enseignants			
<b>Description</b>			
<b>Rubrique</b>	<b>Taille</b>	<b>Type</b>	<b>Observation</b>
- Nom du département	25	A	Ex:8h30-10h
-Nom d'enseignant surveillant	20	A	
-Date de l'examen	10	AN	
-Horaire de l'examen	11	AN	
-Module	20	AN	
-Lieux	04	AN	
-Jour	08	A	

## VIII. Diagramme des flux :

### VIII.1. Définition : [15]

Le diagramme des flux est une représentation graphique des différents acteurs (Interne, externe) au domaine d'étude et les flux d'information échangés entre eux, cette représentation a pour but de délimiter le champ du projet.

- **L'acteur** : c'est une unité active intervenant dans le fonctionnement du système opérant, pour l'échange de l'information avec les autres acteurs .L'acteur peut être interne ou externe
- **Le flux** : Un flux d'information ou de données est un échange d'informations entre trois acteurs dans le cadre de système d'information concerné.

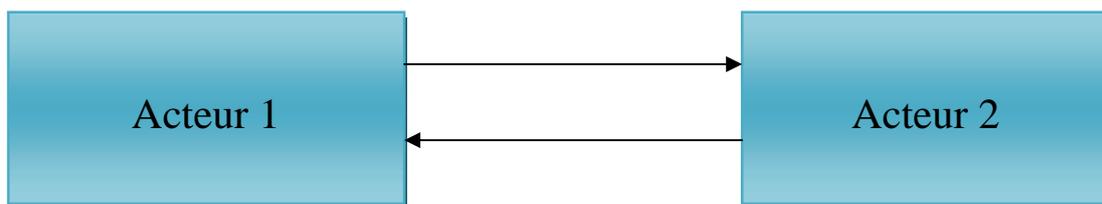
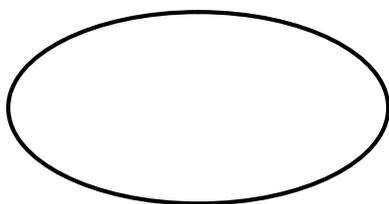


Figure VIII: le schéma du diagramme des flux.

### VIII.2. Le formalisme graphique :



Acteur interne au champ d'étude.



Acteur externe au champ d'étude.



Sens de circulation de l'information.

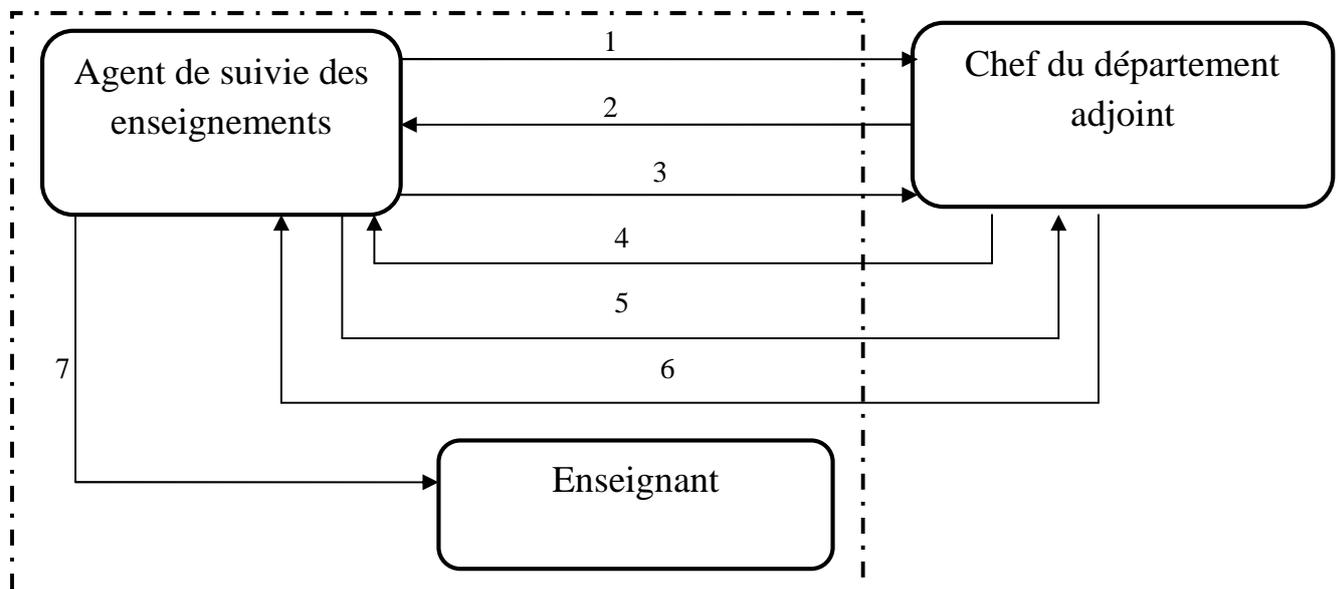
**VIII.3. Acteurs du domaine d'étude :**

Dans notre cas d'étude nous avons recensé les acteurs suivants :

- **Acteurs internes :**
  - le chef du département adjoint.
  - L'agent de suivie des enseignements.
  - L'enseignant.

**VIII.4. Représentation du diagramme des flux existant :**

Ce diagramme représente la circulation de l'information entre les acteurs.



**Figure VIII.4 :** le diagramme des flux existant.

 Domaine d'Etude

- **Description des flux :**

- 1 : Emplois du temps non signé
- 2 : Emplois du temps signé
- 3 : Planning des examens non signé
- 4 : Planning des examens signé
- 5 : Convocation de surveillance non signée
- 6 : Convocation de surveillance signée
- 7 : Convocation de surveillance signée

**IX. Etude de la codification :****IX.1. Définitions :[15]**

- **La codification :**

C'est l'opération qui consiste à remplacer une information sous sa forme naturelle par un code clair qui sera mieux adapté aux besoins de l'utilisateur de l'information.

- **Un code :**

C'est un nom abrégé ou une représentation de l'information permettant de désigner un objet ou un concept de manière claire et unique.

**IX.2. Caractéristiques d'une codification :[15]**

- Une codification ne doit pas être ambiguë. C'est-à-dire qu'elle doit associer un code et un seul à chaque information et de même à chaque information un et un seul code.
- Les codes doivent être concis (comportés très peu de positions ou caractères).
- Une codification doit être simple et facile à comprendre.
- Une codification doit permettre des extensions car l'ensemble des informations à codifier est susceptible de s'accroître dans le temps.
- Une codification doit permettre d'insérer de nouveaux objets entre des objets déjà existants sans remise en cause de la codification.

**IX .3. Type de codification :[15]**

- **Séquentielle:** elle consiste à attribuer à chaque information à codifier, un numéro de sorte que le numéro associés soient consécutifs.
- **Par tranche :** elle consiste à attribuer une tranche de code à chaque catégorie d'objet à codifier.

- **Articulée** : elle consiste à attribuer des codes décomposé en zone, chaque zone est appelée descripteur, chaque descripteur a une signification particulière relative à l'objet codifié.
  - **Par niveau** : c'est un cas particulier de la codification articulée, les descripteurs sont des niveaux.
  - **Mnémorique** : elle consiste à associer au nom de l'objet, un nom abrégé qui rappelle l'objet codifié.
- **Codification existantes :**
- N : Numérique.
  - A : Alphabétique.
  - AN : Alphanumérique.

Intitulé code	Format du code									
<p>Numéro de la carte enseignant</p>	<div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td style="padding: 2px 5px;">N</td> <td style="padding: 2px 5px;">P</td> <td style="padding: 2px 5px;">S</td> <td style="padding: 2px 5px;">S</td> <td style="padding: 2px 5px;">S</td> <td style="padding: 2px 5px;">S</td> <td style="padding: 2px 5px;">G</td> <td style="padding: 2px 5px;">G</td> <td style="padding: 2px 5px;">G</td> </tr> </table>   <div style="display: flex; justify-content: center; gap: 20px;"> <span>↔</span> <span>↔</span> <span>↔</span> <span>↔</span> </div> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 5px;"> <span>A</span> <span>B</span> <span>C</span> <span>D</span> </div> </div> <p>A : première lettre du nom d'enseignant.                      B : première lettre du prénom d'enseignant.                      C : numéro séquentiel d'enseignant.                      D : grade de l'enseignant :( PR: professeur, MCA : mettre conférence A, MCB : mettre conférence B, MAA : mettre assistant A, MAB : mettre assistant B                      MAG : magister, MAS : master,                      ING : ingénieur, DOC : doctorant).</p>	N	P	S	S	S	S	G	G	G
N	P	S	S	S	S	G	G	G		
<p>Identificateur de la séance</p>	<div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td style="padding: 2px 5px;">J</td> <td style="padding: 2px 5px;">N</td> </tr> </table>   <div style="display: flex; justify-content: center; gap: 20px;"> <span>↔</span> <span>↔</span> </div> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 5px;"> <span>A</span> <span>B</span> </div> </div> <p>A: La première lettre de jour                      B : Numéro de séance</p>	J	N							
J	N									
<p>Identificateur du semestre</p>	<div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td style="padding: 2px 5px;">S</td> <td style="padding: 2px 5px;">N</td> <td style="padding: 2px 5px;">C</td> </tr> </table>   <div style="display: flex; justify-content: center; gap: 20px;"> <span>↔</span> <span>↔</span> </div> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 5px;"> <span>A</span> <span>B</span> </div> </div> <p>A: Semestre.                      B : cycle.</p>	S	N	C						
S	N	C								
<p>Identificateur du lieu</p>	<div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td style="padding: 2px 5px;">L</td> <td style="padding: 2px 5px;">N</td> <td style="padding: 2px 5px;">N</td> </tr> </table>   <div style="display: flex; justify-content: center; gap: 20px;"> <span>↔</span> <span>↔</span> </div> <div style="display: flex; justify-content: center; gap: 20px; margin-top: 5px;"> <span>A</span> <span>B</span> </div> </div> <p>A : type du lieu                      B : numéro du lieu.</p>	L	N	N						
L	N	N								

**X. Problématique et objectif de cette étude :****X.1. Problématique :**

L'étude de l'existant est une étape très importante pour la réalisation d'un système informatique. Lors de cette dernière, nous avons relevé les différentes anomalies constatées.

Dans le cas étudié du département d'informatique, de nombreux dysfonctionnements ont poussé à vouloir mettre en place un système de planification.

Actuellement le département souffre du débordement de fichiers qui implique une anarchie du travail ; car la planification se fait manuellement ce qui cause la non sécurisation des données.

En effet, la problématique s'articule au tour des points suivants :

- La difficulté de saisir au moment voulu ce qui engendre un retard dans le traitement de l'information: nous avons constaté que le temps de saisie d'un emploi du temps ou d'un planning d'examen est très important.
- Travaux manuels élevés, lourds et pénibles ce qui provoque un risque d'erreur lors de remplissage des différents documents et d'oubli ou chevauchement des emplois de temps et des plannings d'examens.
- La perte d'information et difficulté de recherche et d'accès aux documents.

**X.2. Objectifs de cette étude :**

Comment assurer un accès rapide aux informations tout en assurant la sécurité et l'intégrité ainsi que la préservation des dossiers à long terme ?

L'informatique propose une solution d'archivage et de suivi très claire et offre plusieurs moyens pour préserver l'information en toute sécurité par l'authentification.

La solution proposée doit assurer les fonctions suivantes :

- saisie et impression des documents très facile et rapide.
- Faciliter la recherche et l'accès aux informations, ce qui permet un gain de temps.
- Automatiser certaines tâches qui se traitent manuellement.
- Stockage des informations sur des supports magnétiques ce qui assurera leur sécurité.

**XI. Conclusion :**

Dans ce chapitre on a présenté une description générale de département d'informatique, nous avons pu analyser l'organisme et localiser les anomalies informationnelles et organisationnelles auxquelles les taches de chaque acteur de l'organisation, les procédures de travail ainsi que les flux d'information qui circule entre les différents postes du travail. Elle a permis, entre autres, de mettre en évidence les insuffisances qui entravent le bon fonctionnement du service ce qui nous a faciliter de dégager une vision de la solution à réaliser. Cette vision décrit les utilisateurs concernés par le futur système ainsi que l'ensemble des fonctionnalités offertes par celui-ci. Un aperçu sur la consultation spécialisée ainsi que la problématique.

Dans le chapitre suivant, on va présenter quelque notion et concept qui vont nous permettre de concevoir notre projet.

## **Chapitre III : Analyse et Conception**

## **I. PIntroduction :**

La complexité et la taille des logiciels nécessitent des outils, des techniques et des méthodes pour mener à terme la réalisation d'applications dans les meilleures conditions. Tout processus de développement d'applications ou systèmes informatiques doit suivre une démarche bien définie, Pour cela, la normalisation de leur conception et de leur développement s'impose.

Les méthodes de conception viennent à point nommé pour guider le concepteur dans sa tâche de conception. Elles imposent, une certaine discipline de travail qui, en retour, garantit la modularité et la justesse des applications développées.

Au niveau de ce chapitre, nous allons présenter l'architecture conceptuelle de notre application en s'aidant des principes et des différents diagrammes d'UML (un standard en matière de modélisation Orientée objet).

## **II. Présentation UML:**

### **II.1. Définition : [16]**

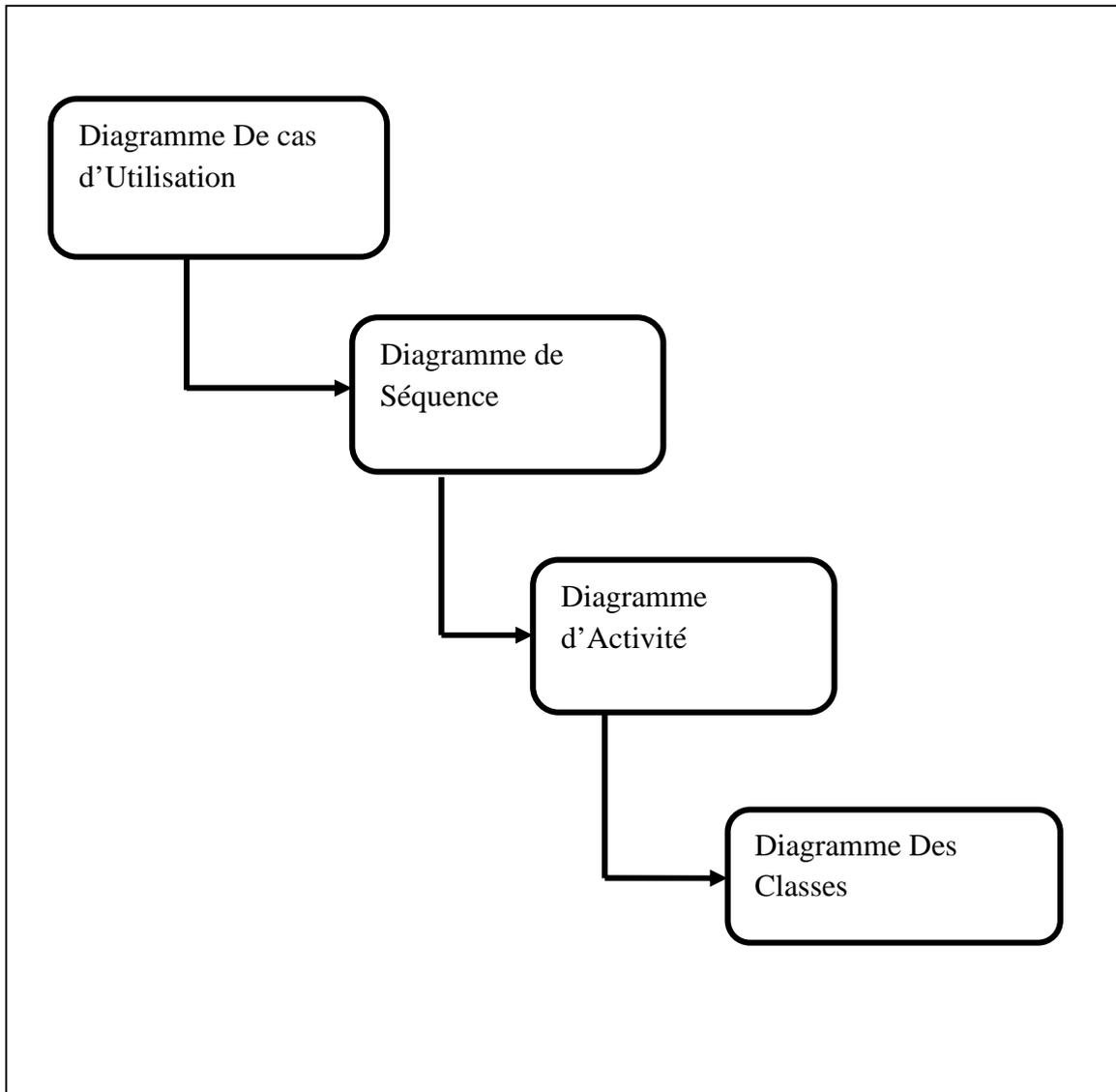
UML (Unified Modeling Language), c'est-à-dire langage unifier pour la modélisation. C'est une notation graphique destinée à la création de modèles orientés objet en vue de l'analyse et de la modélisation de logiciel orienté objet. Ce n'est pas une méthode c'est un ensemble d'outils permettant la modélisation de la future application informatique.

### **II.2. La démarche suivie :**

Pour une bonne modélisation on a adopté la démarche suivante :

- Identification et représentation des cas d'utilisation.
- Elaboration de diagrammes de séquence.
- Elaboration des diagrammes d'activités.
- Elaboration du diagramme de classes.

La figure suivante donne la présentation graphique de la démarche de Modélisation choisie pour concevoir l'application.



**Figure II.2 :** Démarche de modélisation de l'application

### III. Phase d'analyse :

#### III. 1. Introduction :

L'objectif de cette partie est la spécification de manière claire et concise de l'application d'une planification des activités pédagogiques, c'est pendant cette activité que sont définis les besoins (le but du système), qui vont diriger le développement et les tests du système formalisés ainsi que l'ensemble des contraintes, puis l'estimation de la faisabilité de ces besoins.

Pour se faire, il est nécessaire de déterminer globalement ce qui se trouve dans le champ de l'application.

#### III.2. Spécification des besoins :

C'est une étape primordiale au début de chaque démarche de développement. Son but est de veiller à développer un logiciel adéquat, sa finalité est la description générale des fonctionnalités du système, en répondant à la question suivante: "Quelles sont les fonctions du système?".

Notre projet porte sur l'objectif de développer une application pour l'automatisation des procédures de la planification des activités pédagogique.

Nous pouvons recenser quelques objectifs de notre projet sous forme de points :

##### ➤ A l'administrateur de :

- Gérer le compte.
- Gérer les emplois de temps (Programmer les séances cours, de TD et de TP pour chaque semestre).
- Gérer les plannings des examens (Programmer les séances d'examens pour chaque semestre).
- Gérer les lieux (salles, amphis, et labos).
- Gérer les modules.
- Consulter la disponibilité des enseignants.
- consulter la base de données.

##### ➤ A l'enseignant de :

- Introduire sa disponibilité par semestre.
- Gérer le compte

**III.3. Identification des acteurs de l'application :**

Un acteur représenté dans UML est un ensemble de rôles joué par des entités externes (Utilisateur humain, dispositif matériel ou autre système) qui interagit (saisie de donnée, réception d'information) directement avec le système étudié, autrement dit c'est un objet actif qui utilise les fonctions du système.

Dans le cas de notre application, nous avons identifié deux (02) acteurs qui interagissent avec le système :

- **L'Enseignant :**

C'est la personne qui a les droits d'accès à l'espace enseignant pour saisir sa disponibilité.

- **L'Administrateur :**

Acteur principal, c'est une Personne possédant les droits d'accès à l'espace administrateur, ayant des privilèges de gestion de la base de données (création, modification et suppression des données.).Elle joue aussi le rôle du secrétaire de département.

**III.4. Représentation des cas d'utilisations:**

Un cas d'utilisation est une structure des fonctionnalités nécessaires aux utilisateurs du système.

Le diagramme des cas d'utilisation = acteur+fonctionnalité

**III.4.2. Spécification des tâches pour chaque acteur:**

Une tâche est l'ensemble des différentes fonctions affectées à un acteur bien spécifiée.

Les acteurs définis précédemment effectuent un certain nombre de tâches, ces tâches sont résumées dans le tableau suivant :



**III.4.3. Spécification des scénarios :**

Un scénario est une instance de cas d'utilisation, il consiste en une description textuelle de chaque cas d'utilisation. On décrira de façon narrative toutes les interactions entre les acteurs et le système. Un scénario est un chemin logique traversant le cas d'utilisation.

Chaque acteur effectue des tâches et chaque tâche est décrite par des scénarios qu'on résume dans le tableau suivant :

Acteurs	Tâches	Scénarios
<b>Enseignant</b>	<b>T1:</b> Accéder à l'espace Enseignant	S1 : Saisir le login et le mot de passe puis cliquer sur valider.
	<b>T2 :</b> Saisir sa disponibilité	S1 S2 : sélectionner le lien « Enseignant » S3: sélectionner le lien « Saisir disponibilité » S4: remplir le formulaire de saisie de disponibilité puis cliquer sur valider
	<b>T3:</b> Changer le mot de passe	S1 S5: Sélectionner le lien « Compte» S 6: Remplir le formulaire du changement de mot de passe puis cliquer sur valider
<b>Administrateur</b>	<b>T4 :</b> Accéder à l'espace Administrateur	S1
	<b>T5 :</b> Gérer les formations	S1 S7: Sélectionner le lien « Formation » puis « Gestion Formation ». S8 : Ajouter, Consulter modifier, supprimer une formation.
	<b>T6 :</b> Gérer les modules	S1 S9: Sélectionner le lien « Module » puis « Gestion des modules ». S10 : Ajouter, Consulter modifier, supprimer un module.

	<b>T7 : Gérer les lieux</b>	S1 S11 : Sélectionner le lien « Lieu » puis « Gestion des lieux » S12 :Ajouter, consulter, modifier, supprimer.
	<b>T8 : Gérer les enseignants</b>	S1 S13: Sélectionner le lien « Enseignant » puis «Gestion des enseignants » S14:consulter la liste des enseignants, ajouter, supprimer, modifier
	<b>T9 : Gérer les Séances</b>	S1 S15 : Sélectionner le lien « Séance » puis « Gestion des Séances» S16: Sélectionner le lien « Gestion des emplois du temps» S17 : Ajouter, Consulter modifier, supprimer, imprimer S18: Gérer les plannings des examens S19: Ajouter, Consulter modifier,supprimer, imprimer
	<b>T10 : Consulter la base de données</b>	S1 S20 : Sélectionner le lien « Base de données » puis « Gestion base de données» S21 : Consulter
	<b>T11 : Changer le mot de passe</b>	S1 Idem que l'enseignant

Tableau III.4.3 : Spécification des scénarios.

**III.4 .4 Description des cas d'utilisation avec des scénarios:**

Pour détailler le déroulement d'un cas d'utilisation, la procédure la plus évidente consiste à recenser de façon textuelle toutes les interactions entre les acteurs et le système.

Dans ce qui suit nous décrivons donc quelques cas d'utilisation de notre système.

**Rôle :** Utilisateur (Administrateur ou Enseignant).

**Description :**

L'utilisateur accède à la page d'accueil

Après l'authentification l'utilisateur choisit la tâche désirée

**Cas d'utilisation :** « authentification ».

**Scénario:** S1.

**Rôle :** L'utilisateur.

**Description :**

L'utilisateur accède à la page d'accueil.

Le système affiche la page d'accueil.

L'utilisateur clique sur le bouton connexion.

Le système affiche la page d'authentification

L'utilisateur remplit le formulaire (saisir son login et mot de passe), puis il clique sur le bouton se connecter.

Le système affiche l'espace utilisateur si les données remplies correspondent à celles de la base de données, sinon il affiche un message d'erreur et renvoie le formulaire d'authentification.

**Description du cas d'utilisation «authentification »**

**Cas d'utilisation** « Ajouter formation»

**Scenario:** S1, S6, S7.

**Rôle :** Administrateur.

**Description :**

L'administrateur accède à son espace.

L'administrateur clique le lien « Gestion Formations », puis il clique sur « Nouvelle Formation »

Le système affiche le formulaire d'ajout.

L'administrateur remplit le formulaire d'ajout, puis il clique sur « Enregistrer Formation »

**Description du cas d'utilisation «Ajouter Formation »**

**Cas d'utilisation** «Modifier Formation»**Scenario:** S1, S6, S7.**Rôle :** Administrateur.**Description :**

L'administrateur accède à son espace.

L'administrateur sélectionne le lien « Gestion Formations.

Le système affiche la liste des formations.

L'administrateur sélectionne une formation puis il clique sur « MOD »

Le système affiche le formulaire de modification.

L'administrateur remplit le formulaire de modification, puis il clique sur « Modifier Formation »

**Description du cas d'utilisation** «Modifier Formation»**Cas d'utilisation** «Supprimer Formation»**Scenario:** S1, S6, S7.**Rôle :** Administrateur.**Description :**

L'administrateur accède à son espace.

L'administrateur sélectionne le lien « Gestion Formations.

Le système affiche la liste des formations.

L'administrateur sélectionne une formation puis il clique sur « SUP»

**Description du cas d'utilisation** «Supprimer Formation»**Cas d'utilisation** «Ajouter enseignant»**Scenario:** S1, S13, S14.**Rôle :** Administrateur.**Description :**

L'administrateur accède à son espace.

L'administrateur clique le lien « Gestion Enseignant », puis il clique sur « Nouveau Enseignant »

Le système affiche le formulaire d'ajout.

L'administrateur remplit le formulaire d'ajout, puis il clique sur « Enregistrer Enseignant »

**Description du cas d'utilisation** «Ajouter Enseignant»

**Cas d'utilisation « Afficher enseignant »****Scenario:** S1, S13, S14.**Rôle :** Administrateur.**Description :**

L'administrateur accède à son espace.

L'administrateur clique le lien « Gestion Enseignant », puis il clique sur le bouton d'affichage.

**Description du cas d'utilisation «Afficher Enseignant»****Cas d'utilisation « Saisir disponibilité »****Scenario:** S1, S2, S3, S4**Rôle :** Enseignant.**Description :**

L'enseignant sélectionne le lien « Enseignant ».

L'enseignant sélectionne le lien « saisir disponibilité ».

Le système affiche le formulaire disponibilité.

L'enseignant remplit le formulaire puis il clique sur valider.

**Description du cas d'utilisation «Saisir disponibilité»****Cas d'utilisation «Changer le mot de passe»****Scenario:** S1, S5, S6.**Rôle:** L'utilisateur.**Description:**

L'utilisateur accède à son espace.

L'utilisateur clique sur le lien « Compte».

le système affiche le formulaire de changement de mot de passe.

saisir l'ancien et le nouveau mot de passe, le confirmer puis valider.

**Description du cas d'utilisation «Changer le mot de passe»**

### III.4.5 Diagrammes des cas d'utilisations

Les diagrammes des cas d'utilisations représentent un ensemble de cas d'utilisation, les acteurs et leurs relations.

### III.5. Les Diagrammes de séquence

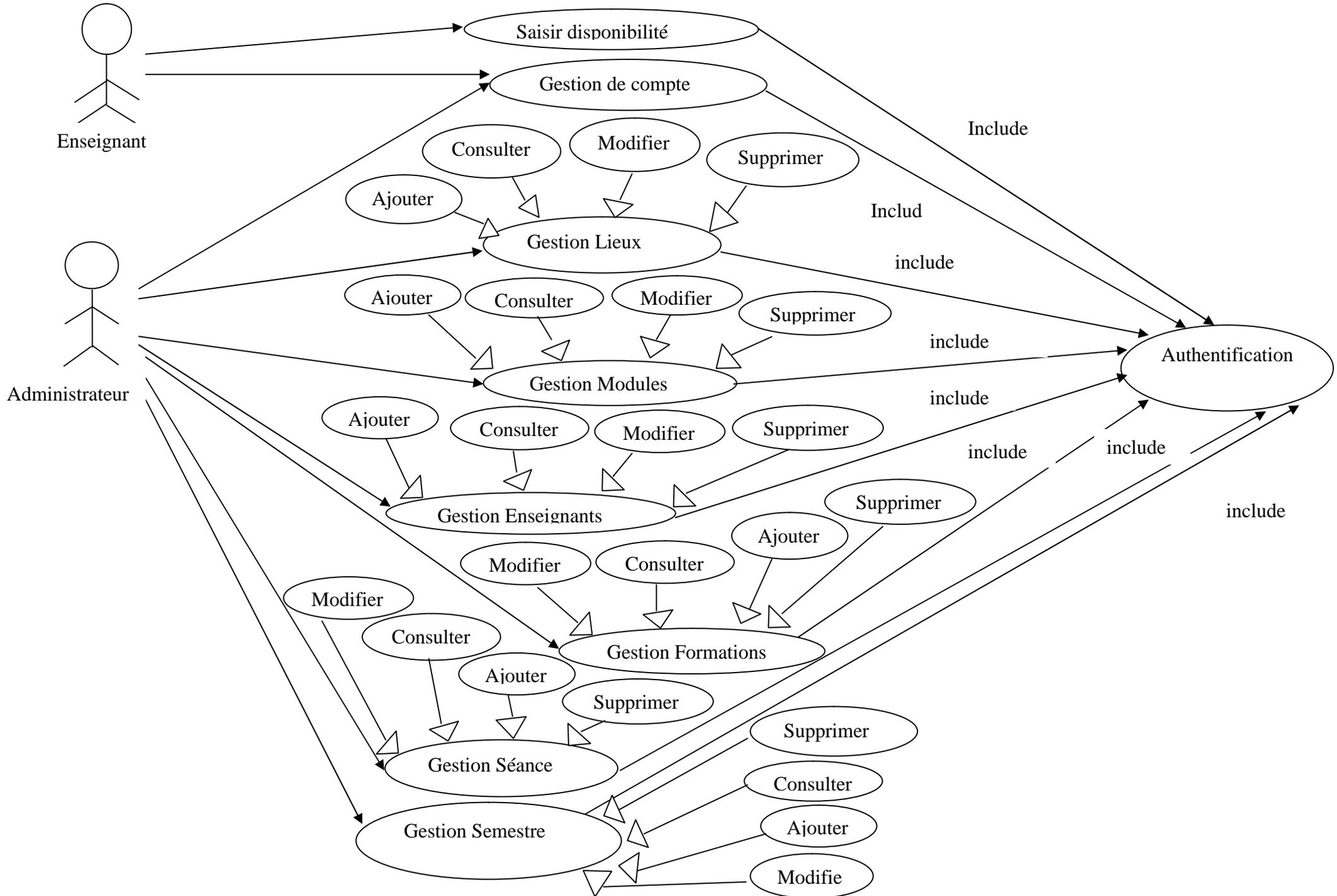
Après avoir décrit les cas d'utilisation, nous allons élaborer les diagrammes de séquences qui permettent de représenter les interactions entre les objets et mieux visualiser les séquences des messages entre les objets.

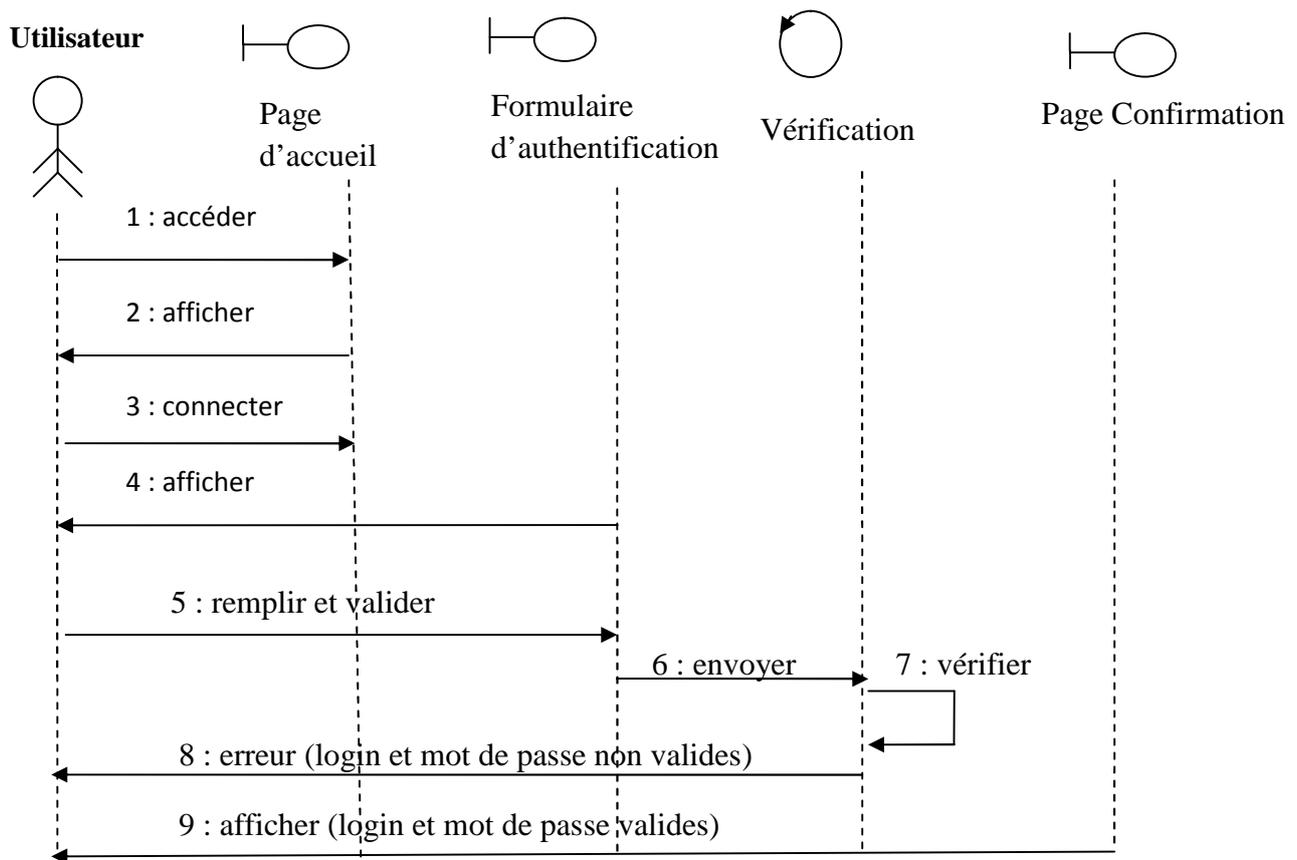
#### III.5.1. Diagramme de séquence cas d'utilisation « Authentification utilisateur »

**Scénario :**

- 1 : L'utilisateur accède à la page d'accueil.
- 2 : Le système affiche la page d'accueil.
- 3 : L'utilisateur clique sur le bouton connexion.
- 4 : Le système affiche le formulaire d'authentification.
- 5 : L'utilisateur remplit le formulaire d'authentification puis il clique sur le bouton se connecter.
- 6 : Le système envoie les données à la BDD
- 7 : le système vérifie les données.
- 8 : Si les informations ne sont pas valides le système refuse et affiche un message d'erreur.
- 9 : Si les informations sont valides le système affiche la page espace administrateur.

# Diagramme des Cas d'Utilisation



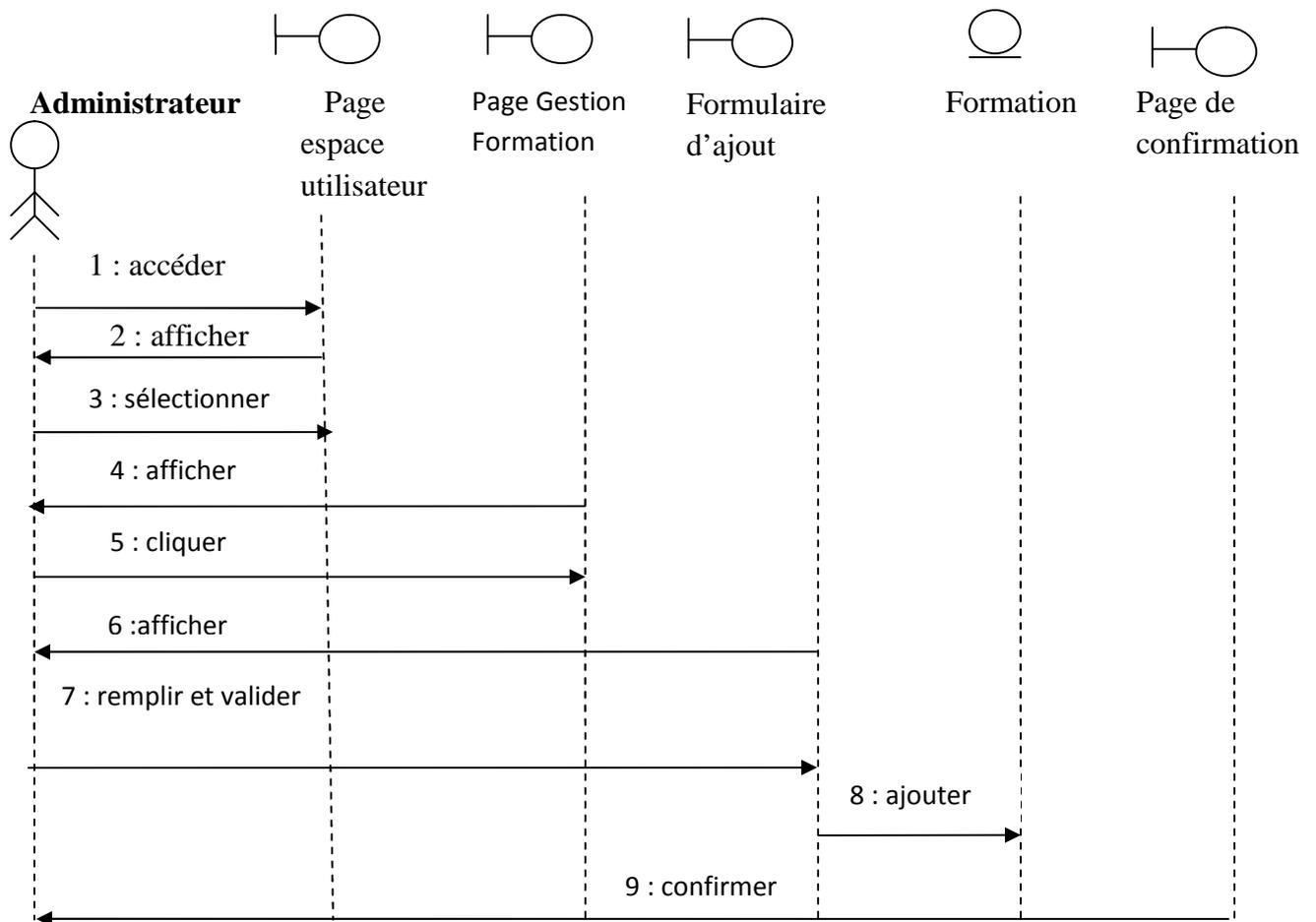


**Figure III.5.1 : Diagramme de séquence du cas d'utilisation  
« Authentification utilisateur »**

### III.5.2. Diagramme de séquence cas d'utilisation « Ajouter formation »

#### Scénario :

- 1 : L'administrateur accède à son espace.
- 2 : Le système affiche son espace.
- 3 : L'administrateur sélectionne le lien «Formation » puis il sélectionne le lien « Gestion Formation ».
- 4 : Le système affiche la page de gestion des formations.
- 5 : L'administrateur clique sur « Nouvelle Formation »
- 6 : Le système affiche le formulaire d'ajout.
- 7 : L'administrateur remplit le formulaire d'ajout puis clique sur « Enregistrer Formation ».
- 8 : Le système ajoute la formation.
- 9 : Le système affiche le message de confirmation.



**Figure III.5.2.: Diagramme de Séquence du cas d'utilisation « Ajouter formation »**

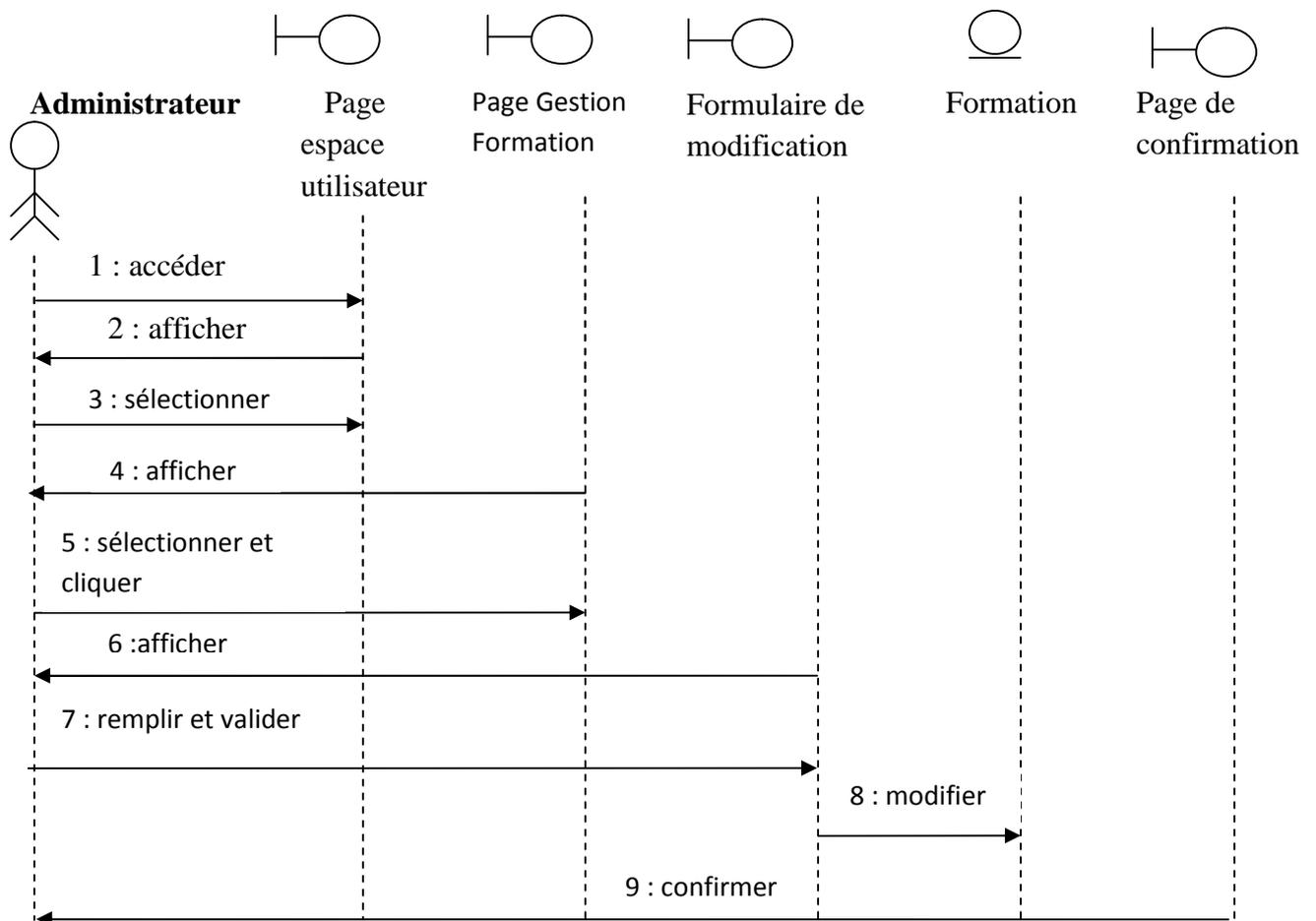
### III.5.3. Diagramme de séquence du cas d'utilisation « Modifier formation » :

#### Scenario :

- 1 : L'administrateur accède à son espace.
- 2 : Le système affiche son espace.
- 3 : L'administrateur sélectionne le lien «Formation »puis il sélectionne le lien « Gestion Formation ».
- 4 : Le système affiche la page de gestion des formations.
- 5 : L'administrateur sélectionne la formation puis clique sur « MOD »
- 6 : Le système affiche le formulaire de modification.
- 7 : L'administrateur remplit le formulaire de modification puis clique sur « Modifier Formation ».

8:Le système modifie la formation.

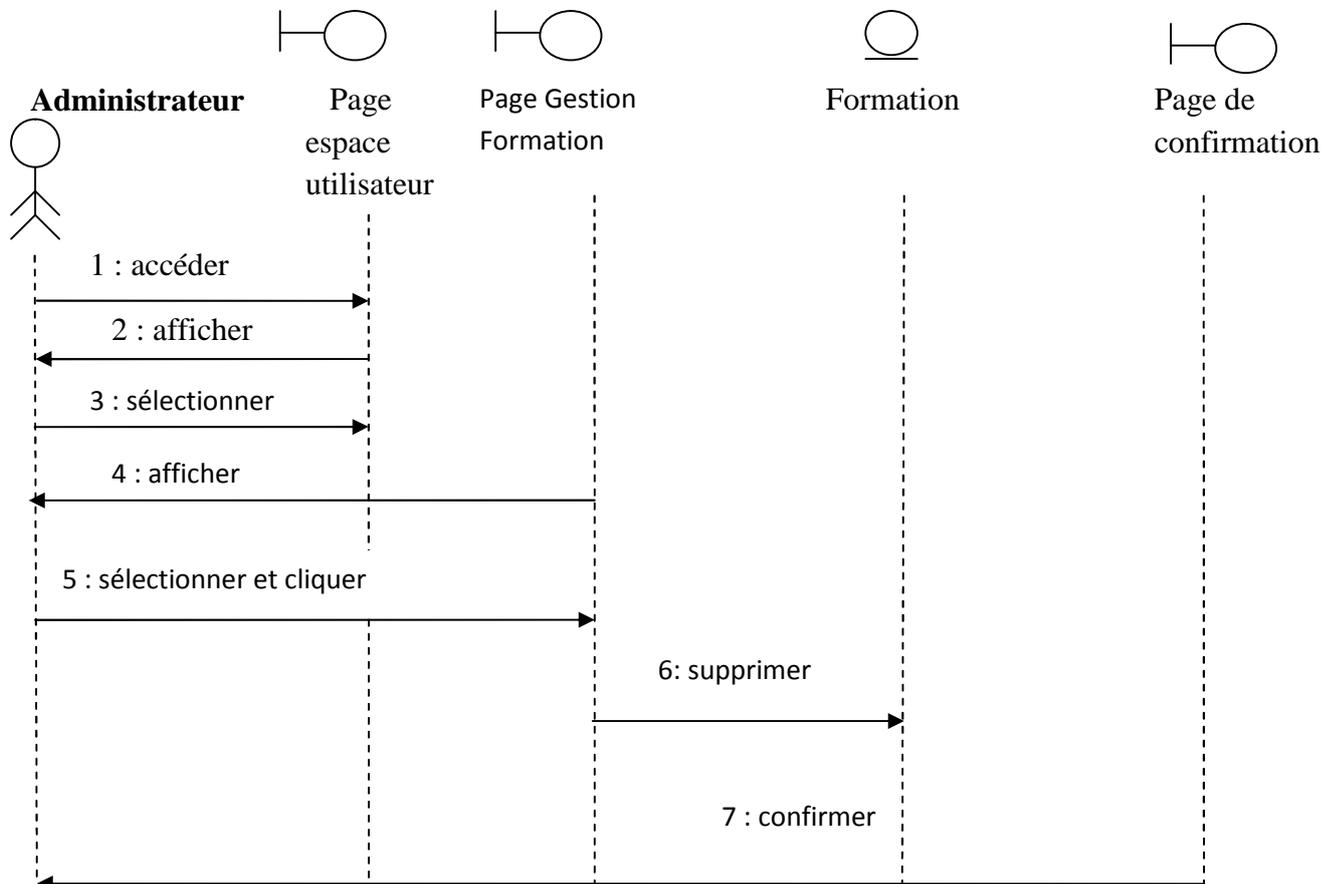
9: Le système affiche le message de confirmation.



**Figure III.5.3. Diagramme de séquence du cas d'utilisation « Modifier formation »**

**III.5.4. Diagramme de séquence du cas d'utilisation « Supprimer formation » :****Scenario :**

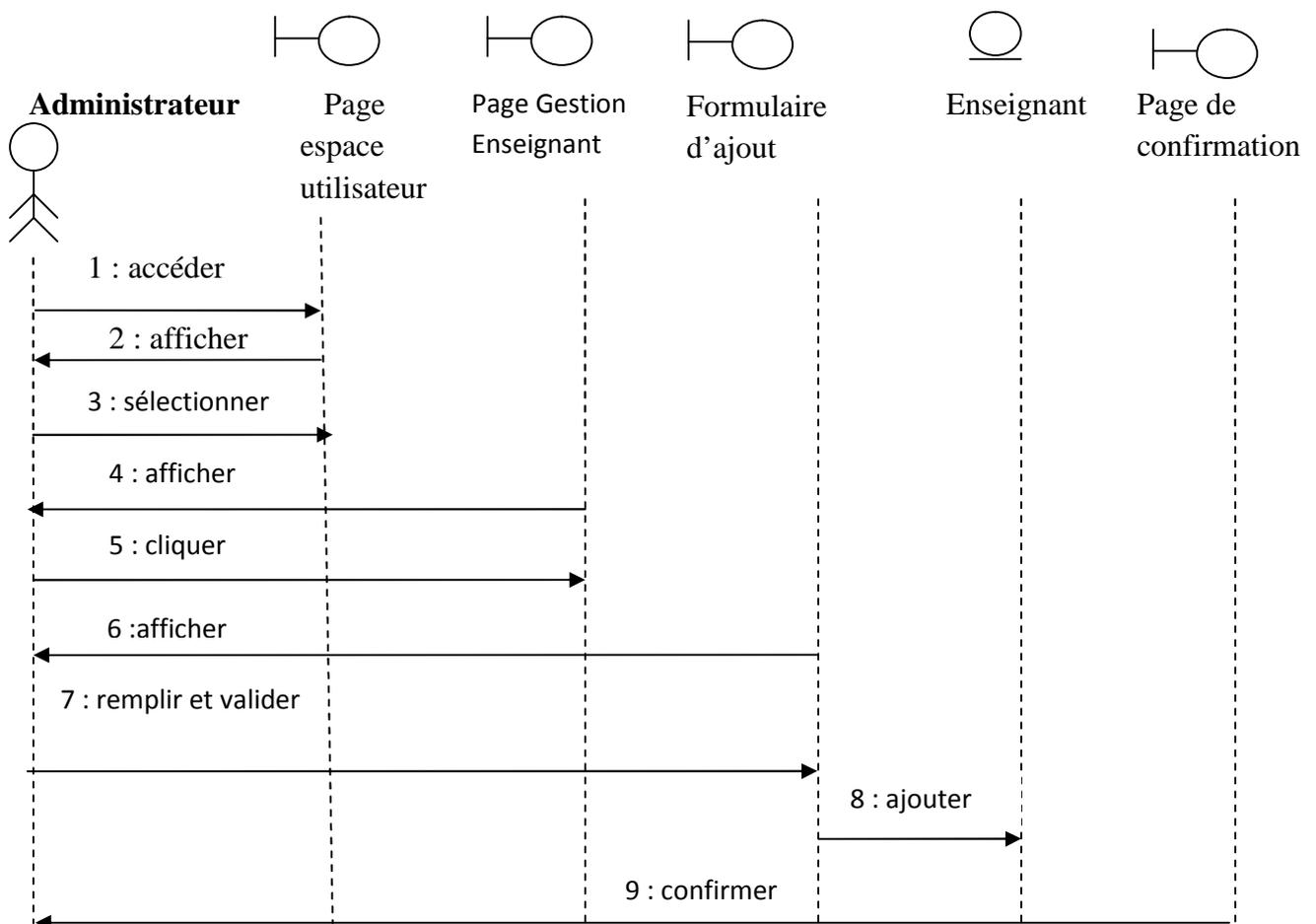
- 1 : L'administrateur accède à son espace.
- 2 : Le système affiche son espace.
- 3 : L'administrateur sélectionne le lien « Formation » puis il sélectionne le lien « Gestion Formation ».
- 4 : Le système affiche la page de gestion des formations.
- 5 : L'administrateur sélectionne la formation puis clique sur « SUP »
- 6 : Le système supprime la formation.
- 7 : Le système affiche le message de confirmation.

**Figure III.5.4 : Diagramme de séquence du cas d'utilisation « Supprimer formation »**

**III.5.5. Diagramme de séquence cas d'utilisation « Ajouter enseignant »**

**Scénario :**

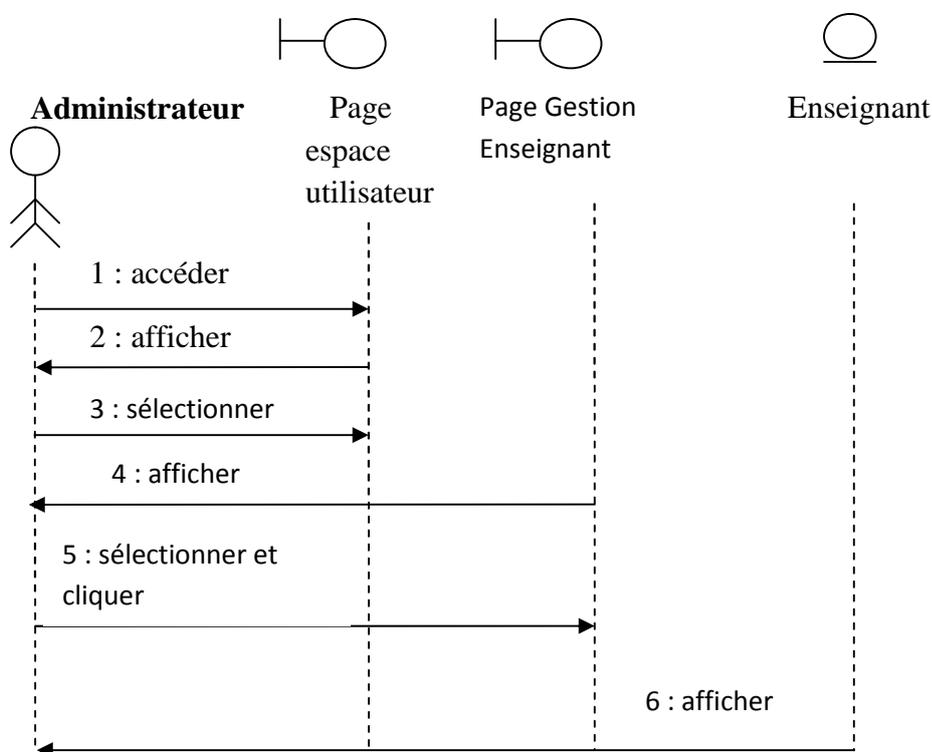
- 1 : L'administrateur accède à son espace.
- 2 : Le système affiche son espace.
- 3 : L'administrateur sélectionne le lien «Enseignant »puis il sélectionne le lien « Gestion Enseignant ».
- 4 : Le système affiche la page de gestion des enseignants.
- 5 : L'administrateur clique sur « Nouveau enseignant »
- 6 : Le système affiche le formulaire d'ajout.
- 7 : L'administrateur remplit le formulaire d'ajout puis clique sur « Enregistrer Enseignant».
- 8:Le système ajoute l'enseignant.
- 9: Le système affiche le message de confirmation.



**Figure III.5.5 Diagramme de séquence du cas d'utilisation « Ajouter enseignant »**

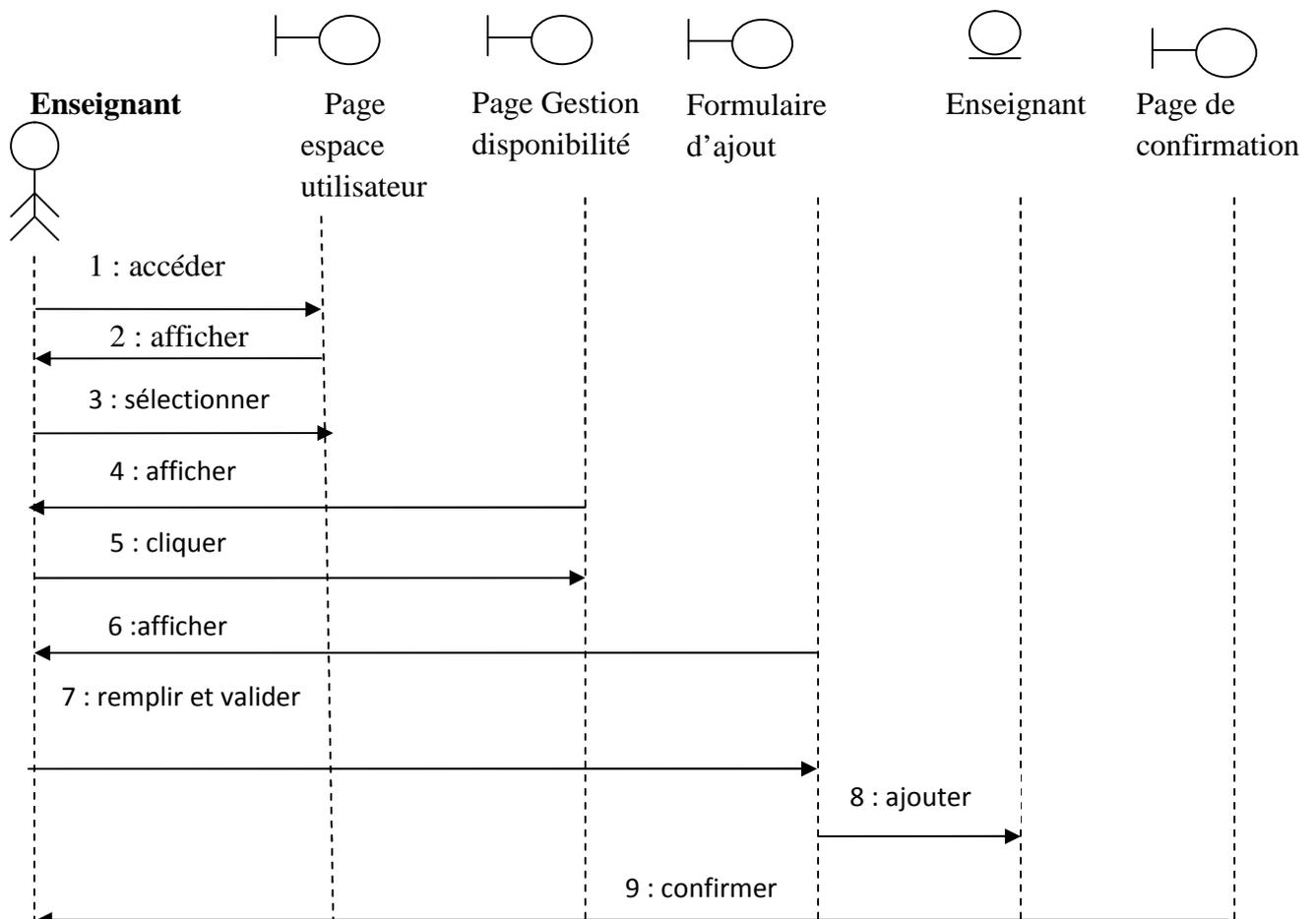
**III.5.6. Diagramme de séquence du cas d'utilisation « Afficher enseignant »****Scenario :**

- 1 : L'administrateur accède à son espace.
- 2 : Le système affiche son espace.
- 3 : L'administrateur sélectionne le lien «Enseignant »puis il sélectionne le lien « Gestion Enseignant ».
- 4 : Le système affiche la page de gestion des enseignants.
- 5 : L'administrateur sélectionne la formation puis clique sur le bouton d'affichage.
- 6:Le système affiche l'enseignant.

**Figure III.5.6 : Diagramme de séquence du cas d'utilisation « Afficher enseignant»**

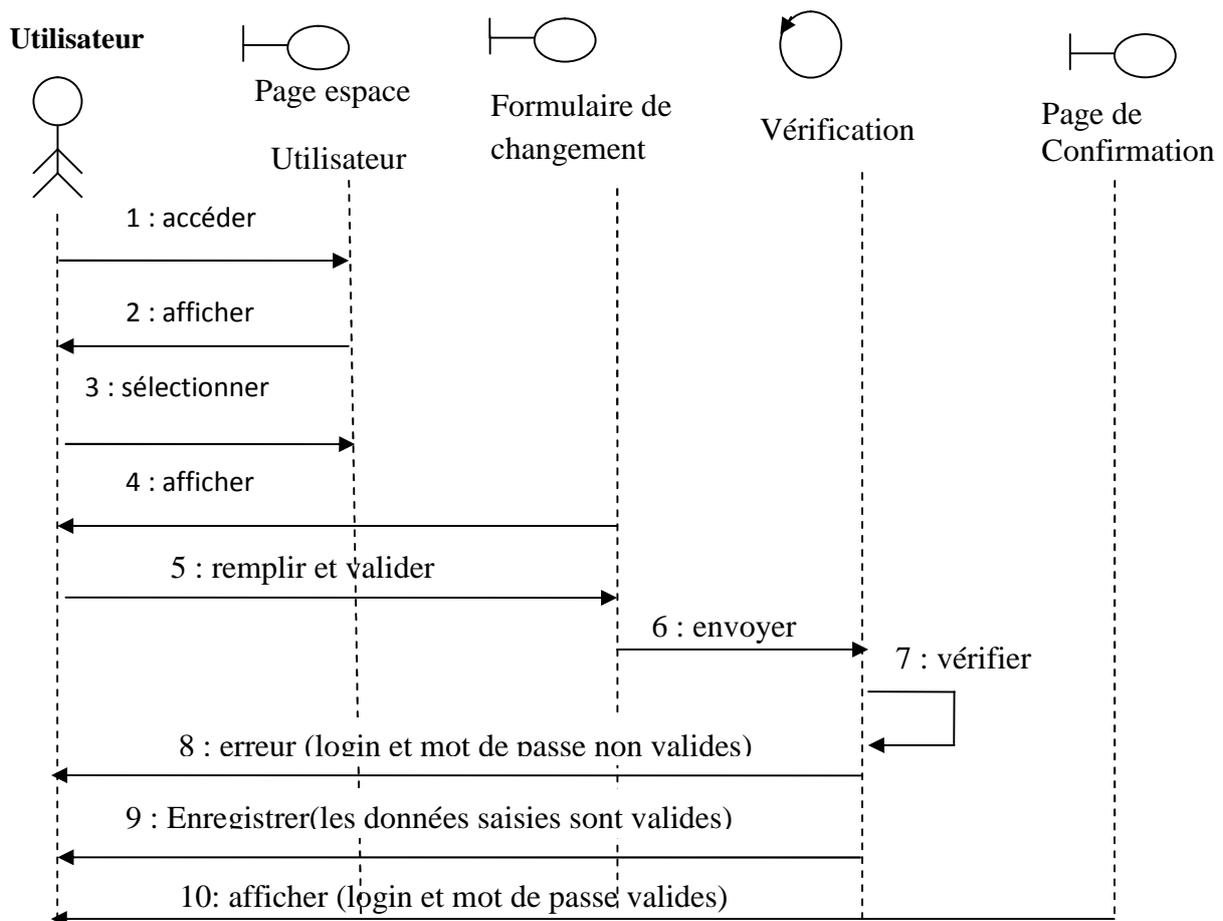
**III.5.7. Diagramme de séquence du cas d'utilisation « Saisir disponibilité » :****Scenario :**

- 1 : L'enseignant accède à son espace.
- 2 : Le système affiche son espace.
- 3 : L'enseignant sélectionne le lien «Enseignant »puis il sélectionne le lien «Saisir disponibilité».
- 4 : Le système affiche la page de gestion des enseignants.
- 5 : L'administrateur clique sur « Nouvelle Disponibilite »
- 6 : Le système affiche le formulaire d'ajout.
- 7 : L'administrateur remplit le formulaire d'ajout puis cliquer sur « Enregistrer Disponibilite».
- 8:Le système ajoute la disponibilité.
- 9: Le système affiche le message de confirmation.

**Figure III.5.7 Diagramme de séquence du cas d'utilisation « Saisir disponibilité »**

**III.5.8. Diagramme de séquence du cas d'utilisation « Changer le mot de passe» :****Scenario :**

- 1 : L'utilisateur accède à son espace.
- 2 : Le système affiche son espace.
- 3 : L'utilisateur sélectionne le lien «Compte».
- 4 : Le système affiche le formulaire de changement.
- 5 : L'administrateur remplit le formulaire puis cliquer sur « valider».
- 6 : Le système envoi les données à la BDD
- 7 : Le système vérifie les modifications et modifie.
- 8 : Si les modifications ne sont pas valides le système affiche un message d'erreur.
- 9 : Si les modifications sont valides le système enregistre les modifications.
- 10: Le système affiche la page de confirmation.

**Figure III.5.8 Diagramme de séquence du cas d'utilisation « Changer le mot de passe»**

### III.6. Diagrammes d'activité

Un diagramme d'activité permet de modéliser le comportement du système, dont la séquence des actions et leurs conditions d'exécution. Les actions sont les unités de base du comportement du système. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

#### III.6.1. Diagramme d'activité du cas d'utilisation « S'authentifier »

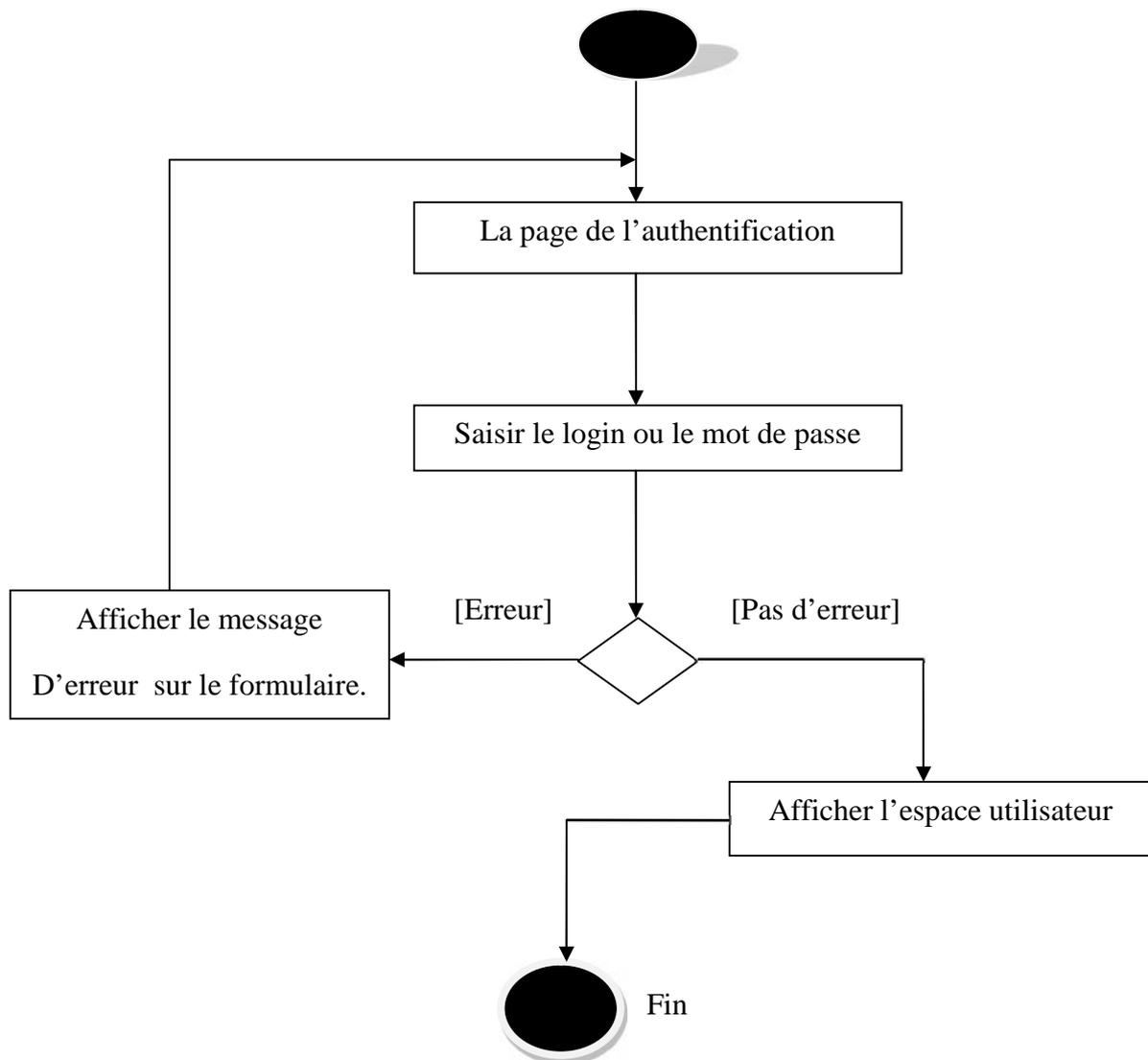


Figure III.6.1: Diagramme d'activité du cas d'utilisation « S'authentifier »

## III.6.2 : Diagramme d'activité du cas d'utilisation « Ajouter Formation »

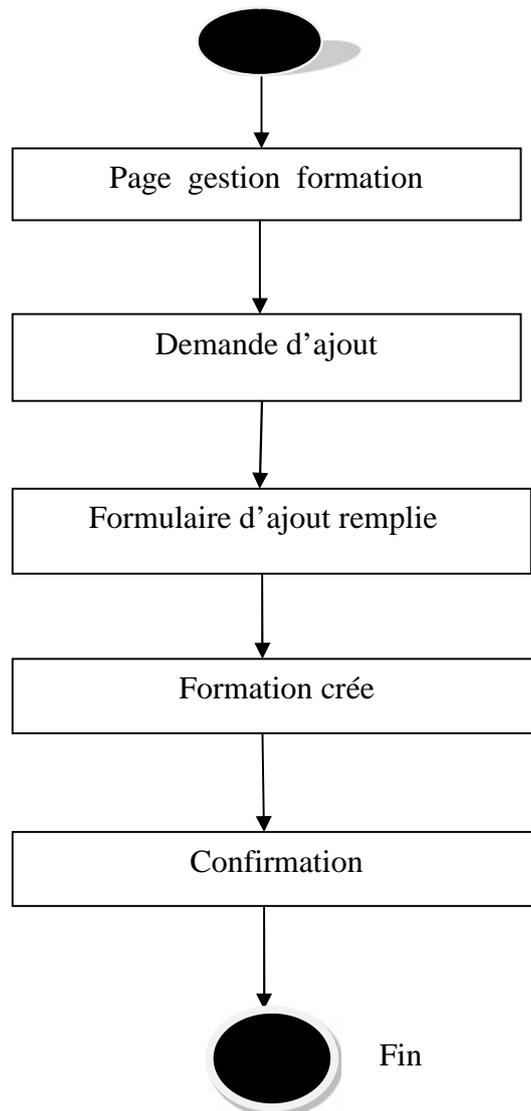


Figure III.6.2 : Diagramme d'activité du cas d'utilisation « Ajouter Formation ».

## III.6.3. Diagramme d'activité du cas d'utilisation «Modification d'une formation »

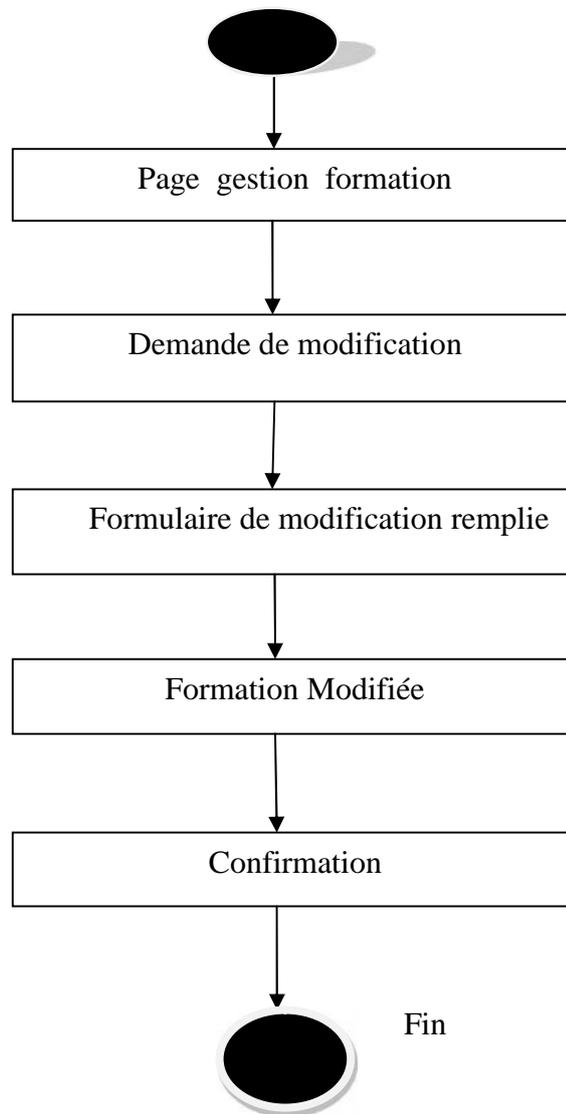
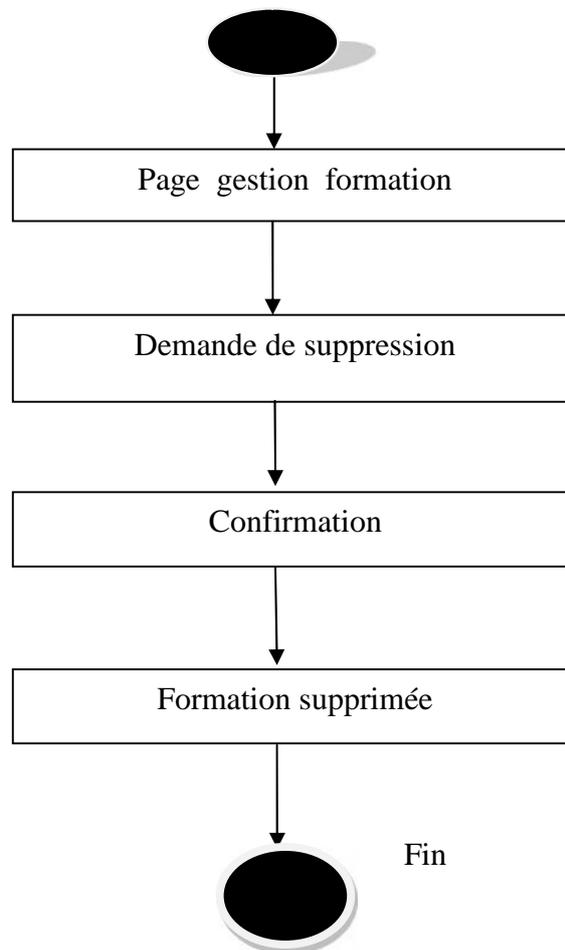
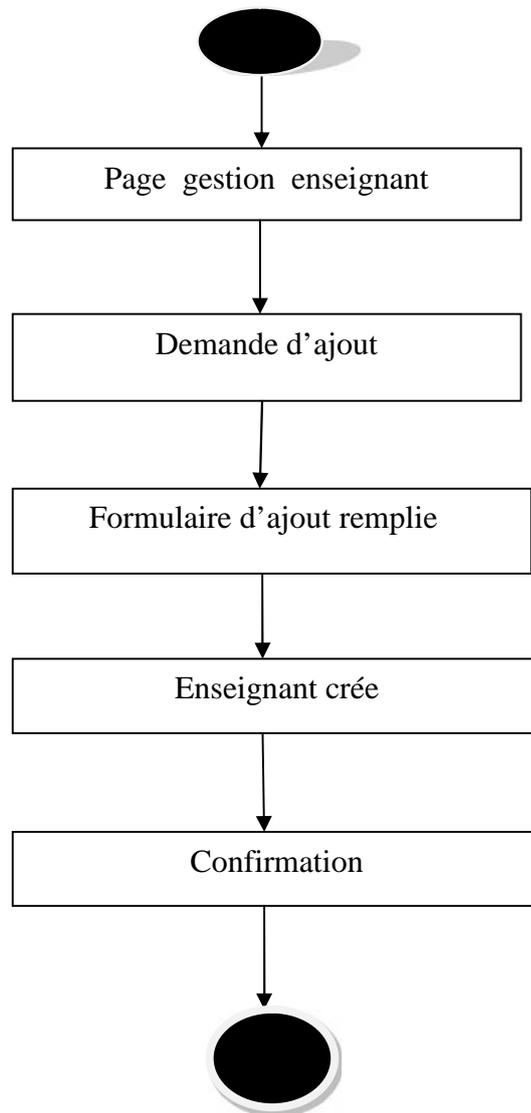
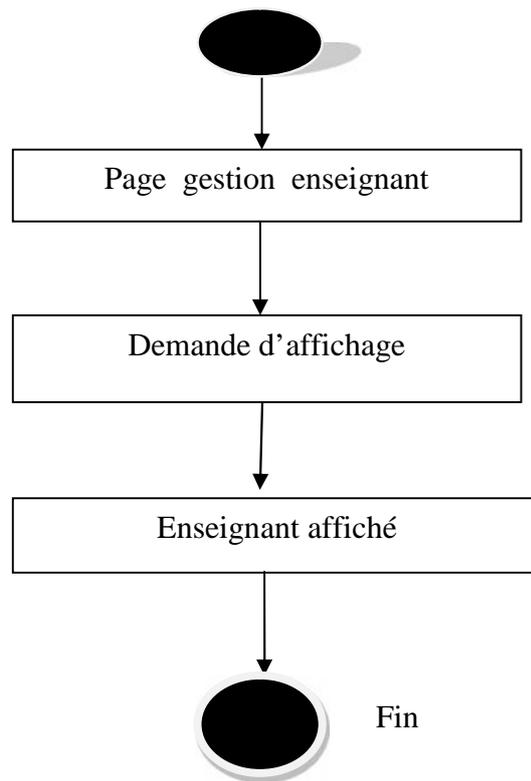
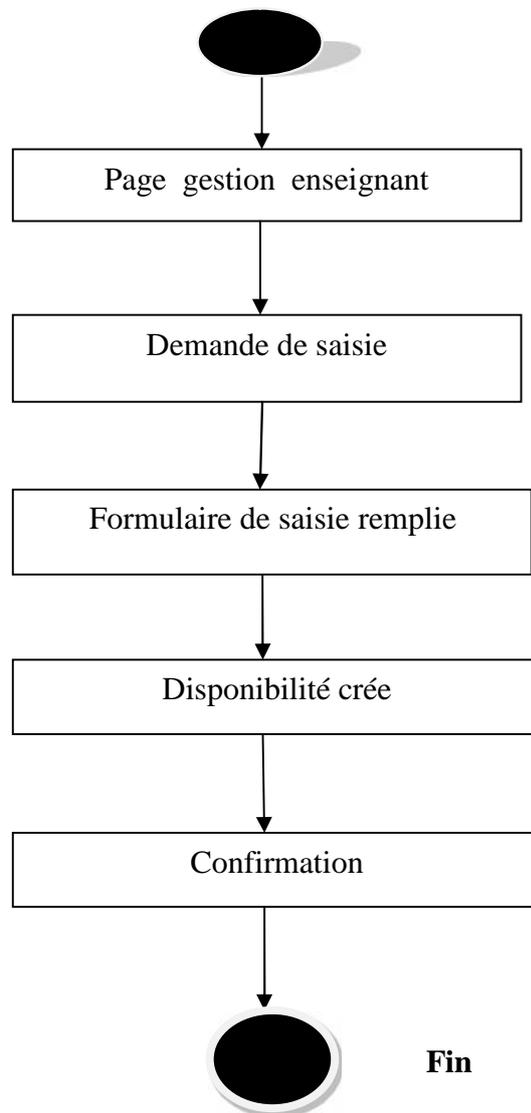


Figure III.6.3: Diagramme d'activité du cas d'utilisation «Modification d'une formation ».

**III.6.4. Diagramme d'activité du cas d'utilisation «Suppression d'une formation ».****Figure III.6.4: Diagramme d'activité du cas d'utilisation «Suppression d'une formation ».**

**III.6.5: Diagramme d'activité du cas d'utilisation « Ajouter Enseignant».****Figure III.6.5: Diagramme d'activité du cas d'utilisation « Ajouter Enseignant».**

**III.6.6: Diagramme d'activité du cas d'utilisation « Afficher Enseignant».****Figure III.6.6: Diagramme d'activité du cas d'utilisation « Afficher Enseignant».**

**III.6.7: Diagramme d'activité du cas d'utilisation « Saisir disponibilité».****Figure III.6.7: Diagramme d'activité du cas d'utilisation « Saisir disponibilité».**

## III.6.8. Diagramme d'activité du cas d'utilisation « Gestion du compte »

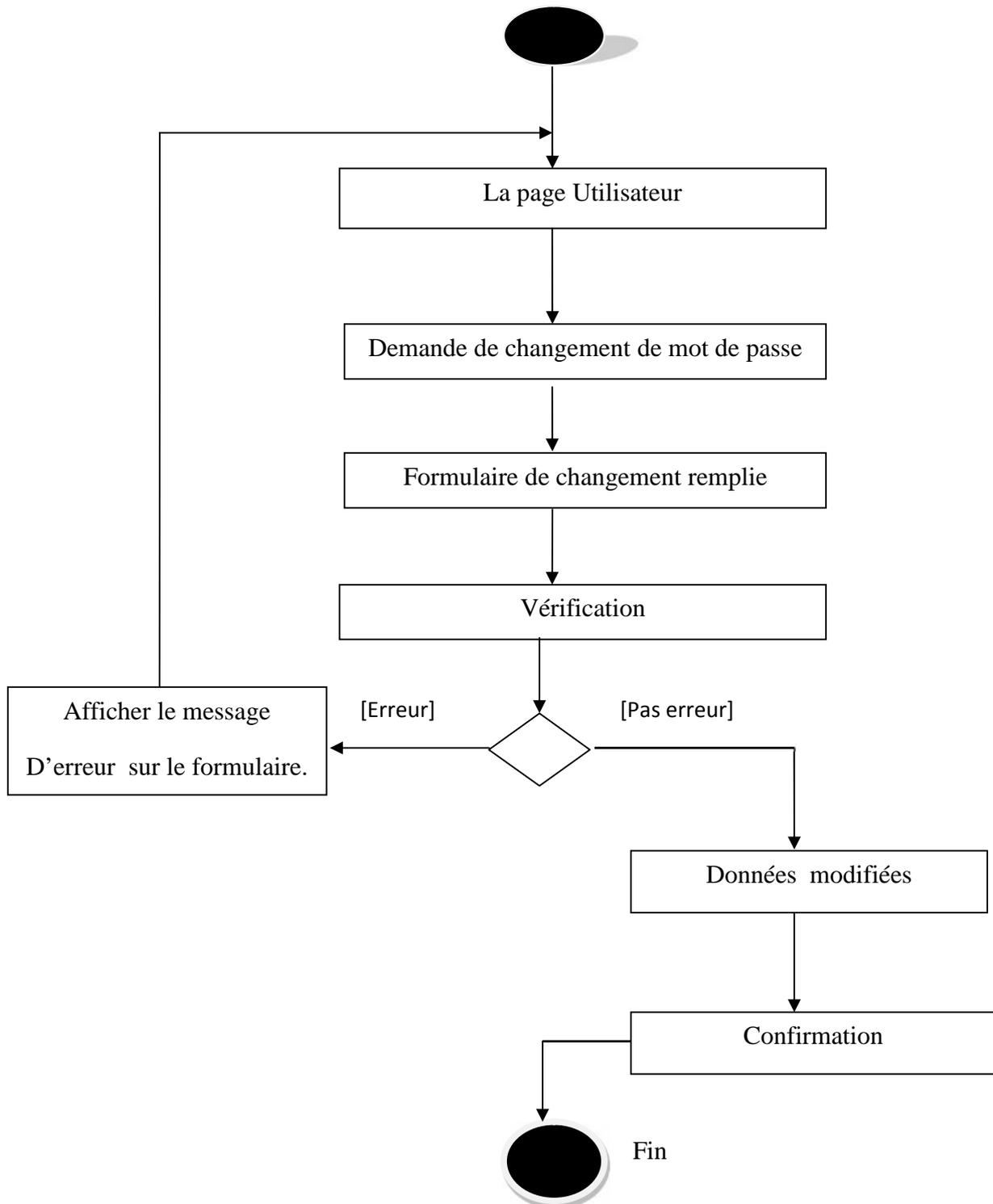


Figure III.6.8. Diagramme d'activité du cas d'utilisation « Gestion du compte »

#### IV. Phase de Conception :

La conception est la phase la plus complexe, son but principal est de rendre le modèle d'analyse réalisable sous forme logicielle.

Nous allons entamer une partie cruciale du développement logiciel qui constitue un pont entre la spécification et la réalisation. Elle comporte la conception de l'application ainsi que la conception de la base de données.

##### IV.1. Diagrammes des classes :

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

##### IV. 2. Règles de gestion :

Les relations entre les tables doivent respecter les relations suivantes :

- Un semestre concerne **un ou plusieurs** modules.
- Un module est concerné par **un et un seul** semestre.
- Un semestre concerne **un ou plusieurs** formations.
- Une formation est concernée par **un ou plusieurs** semestres.
- Un module est affecté à **un ou plusieurs** séances.
- Une séance est affectée à **un ou plusieurs** modules.
- Un lieu est affecté à **un ou plusieurs** séances.
- Une séance est affectée à **un ou plusieurs** lieux.
- Un enseignant est disponible pour **un ou plusieurs** séances.
- Une séance est disponible pour **un et un seul** enseignant.

##### IV.3.Le modèle logique :

###### IV.3.1 Définition d'un schéma relationnel :

Un schéma relationnel (logique) est un ensemble de relation en troisième forme normale, ce schéma peut être obtenu d'un schéma conceptuel tel qu'un diagramme de classe.

Afin d'assurer la cohérence de la base de données lors du passage du modèle conceptuel (le diagramme de classes) au modèle relationnel (logique), un certain nombre de règles doivent être respectées.

### IV.3.2 Règles de passage du diagramme de classe vers le modèle relationnel

Les règles de passage sont les suivantes :

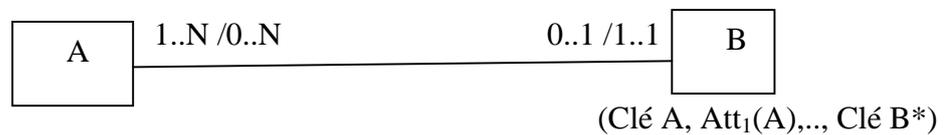
- **Transformation des classes :**

- ✚ La classe se transforme en une table (relation).
- ✚ Les attributs de la classe deviennent des attributs de la table.
- ✚ Choisir la clé primaire (ou les clés primaires).

- **Transformation des associations :**

- ✚ **Cas de liaison binaire à plusieurs (X..N) à un (X..1)**

Ajouter un attribut de type clé étrangère dans la relation à multiplicité plusieurs de l'association.



- ✚ **Cas de liaison plusieurs à plusieurs (X..N) à un (X..N)**

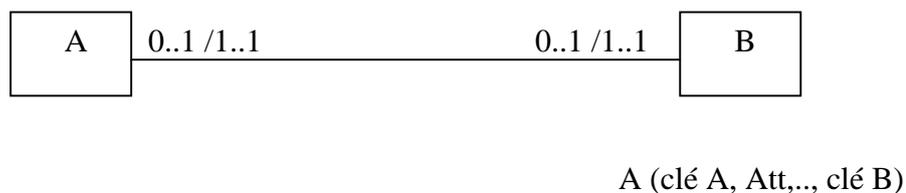
L'association devient une table (Relation) et elle aura comme clé primaire la clé de A et la clé de B.

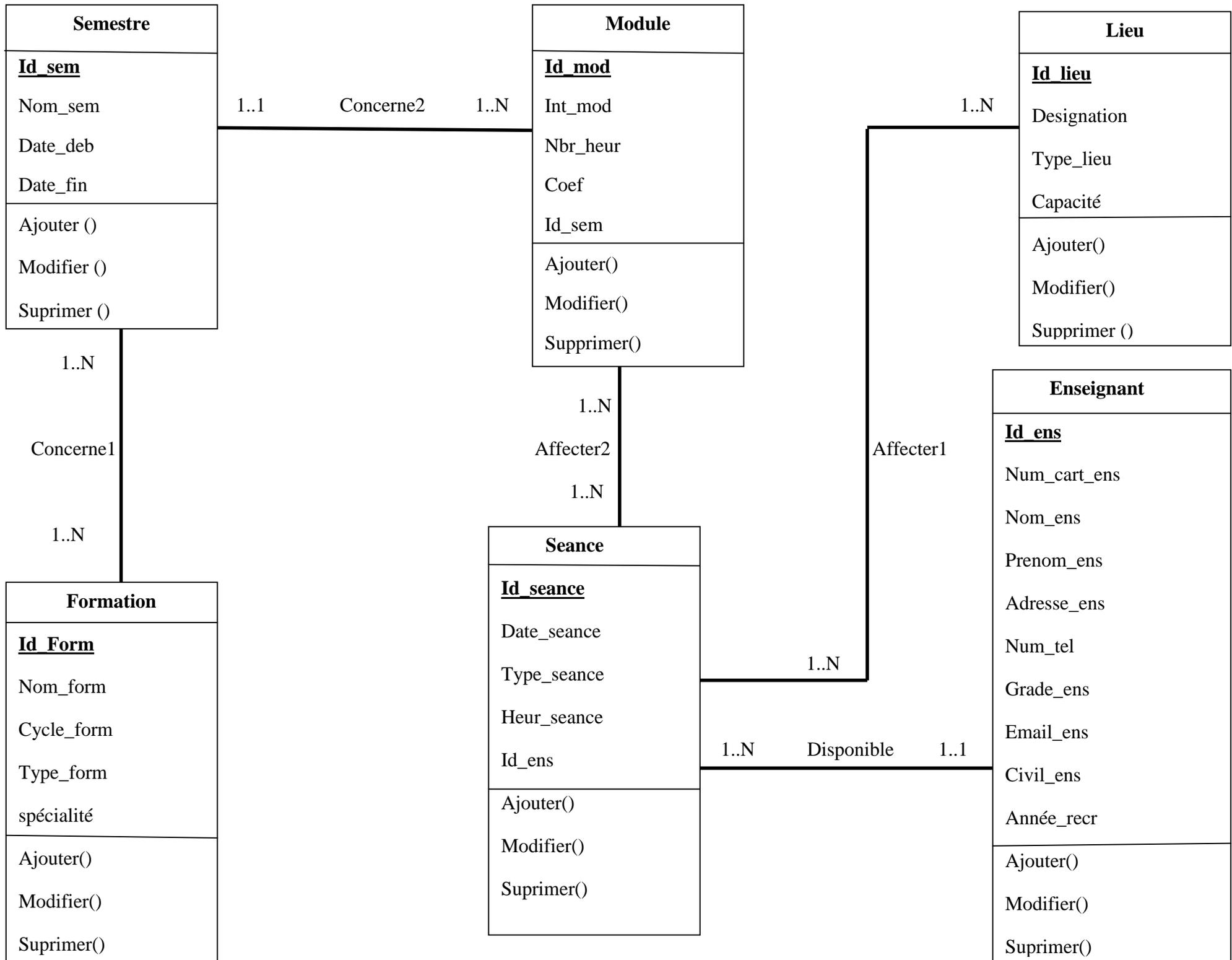


- ✚ **Cas Association binaire de type un (X..1) à un (X..1)**

IL faut ajouter un attribut de type clé étrangère dans la relation dérivée de la classe ayant la multiplicité minimale égale à un.

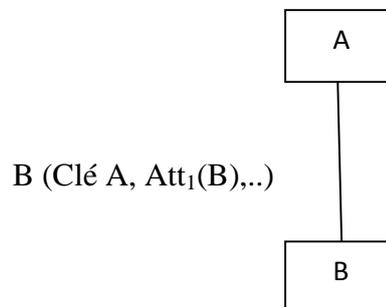
Dans le cas où les multiplicités minimales sont à 1, il est préférable de fusionner les deux entités.





- **Cas de l'héritage**

Transformer chaque sous classe en une table (relation). Les classe filles vont hériter la clé primaire de la classe mère.



- En se basant sur les règles citées précédemment pour le passage du modèle conceptuel au modèle logique, nous obtenons le modèle relationnel suivant :

**Semestre** (id\_sem, nom\_sem, date\_debut, date\_fin)

**Module** (id\_mod, int\_mod, nbr\_heur, coef, id\_sem\*)

**Formation** (id\_form, nom\_form, cycle\_form, type\_form, specialite)

**Concerne1** (id\_sem, id\_form)

**Séance** (Id\_seance, date\_séance, type\_seance, heur\_seance, id\_ens\*)

**Affecter2** (id\_mod, Id\_seance)

**Enseignant**(id\_ens, num\_cart\_ens, nom\_ens, prenom\_ens, adresse\_ens, num\_tel, grade, email, civil, année\_recr)

**Lieu** (Id\_lieu, designation, type\_lieu, capacite)

**Affecter1** (id\_seance, id\_lieu)

- **Remarque :**

\_ : Un attribut souligné signifie qu'il s'agit d'une clé primaire.

\_ : Un attribut suivi d'une étoile signifie qu'il s'agit d'une clé étrangère.

**IV.4. Modèle physique de données :**

Ce modèle nous donne la représentation physique de l'ensemble des tables de la base de données du système étudié.

- Table Formation :

Column Name	Data Type	Nullable	Default	Primary Key
ID_FORM	NUMBER(10,0)	No	-	1
NOM_FORM	VARCHAR2(50)	Yes	-	-
CYCLE_FORM	VARCHAR2(20)	Yes	-	-
TYPE_FORM	VARCHAR2(15)	Yes	-	-
SPECIALITE	VARCHAR2(40)	Yes	-	-

- Table Semestre :

Column Name	Data Type	Nullable	Default	Primary Key
ID_SEM	NUMBER(10,0)	No	-	1
DATE_DEB	DATE	Yes	-	-
DATE_FIN	DATE	Yes	-	-
NOM_SEM	VARCHAR2(15)	Yes	-	-

- Table Module :

Column Name	Data Type	Nullable	Default	Primary Key
ID_MOD	NUMBER(10,0)	No	-	1
INT_MOD	VARCHAR2(40)	Yes	-	-
NBR_HEUR	VARCHAR2(10)	Yes	-	-
ID_SEM	NUMBER(10,0)	Yes	-	-
COEF	NUMBER(10,0)	Yes	-	-

- Table Seance :

Column Name	Data Type	Nullable	Default	Primary Key
ID_SEANCE	NUMBER(19,0)	No	-	1
ID_ENS	NUMBER(10,0)	No	-	-
DATE_SEANCE	DATE	Yes	-	-
TYPE_SEANCE	VARCHAR2(10)	Yes	-	-
HEUR_SEANCE	VARCHAR2(40)	Yes	-	-

- Table Enseignant :

Column Name	Data Type	Nullable	Default	Primary Key
ID_ENS	NUMBER(10,0)	No	-	1
NOM_ENS	VARCHAR2(30)	Yes	-	-
PRENOM_ENS	VARCHAR2(30)	Yes	-	-
ADRES_ENS	VARCHAR2(50)	Yes	-	-
NUM_TEL	NUMBER(10,0)	Yes	-	-
GRADE	VARCHAR2(20)	Yes	-	-
EMAIL	VARCHAR2(30)	Yes	-	-
CIV_ENS	VARCHAR2(10)	Yes	-	-
ANNEE_RECR	VARCHAR2(10)	Yes	-	-
NUM_CART_ENS	NUMBER(10,0)	Yes	-	-

- Table Lieu :

Column Name	Data Type	Nullable	Default	Primary Key
ID_LIEU	NUMBER(10,0)	No	-	1
TYPE_LIEU	VARCHAR2(20)	Yes	-	-
CAPACITE	VARCHAR2(20)	Yes	-	-
DESIGNATION	VARCHAR2(70)	Yes	-	-

- Table Utilisateur :

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER(10,0)	No	-	1
LOGIN	VARCHAR2(50)	Yes	-	-
PWD	VARCHAR2(50)	Yes	-	-
PROFIL	VARCHAR2(20)	Yes	-	-

**V. Conclusion :**

Dans ce chapitre nous avons présenté une démarche de modélisation basée sur le langage UML. Nous avons commencé par l'analyse des besoins qui ont été traduits par la suite en formalismes graphiques à l'aide des diagrammes offerts par le langage UML.

Ensuite on a entamé la partie conception. Ce qui nous a permis de définir les classes de notre application que nous allons utiliser pour la réalisation de cette dernière.

Le chapitre suivant sera consacré à l'environnement technique de développement et à la réalisation (fonctionnement) de notre application.

## **Chapitre IV : Réalisation**

## I. Introduction :

Après avoir présenté dans le chapitre précédent les différentes étapes d'analyse et de conception, qui consiste en réalisation d'une application de gestion des activités pédagogiques de département ; l'étape réalisation s'impose. Cette dernière consiste à traduire et programmer tout ce qui a été conçu dans l'étape antérieure.

Pour cela nous avons utilisé Oracle Database Express Edition comme système de gestion de base de données, NetBeans comme environnement de développement.

Nous allons présenter les outils utilisés ainsi que les interfaces de notre application.

## II. Les outils de développement :

### II.1. Le serveur Web Apache Tomcat: [17]

Apache Tomcat est un conteneur Web libre de Servlets et JSP java EE. Il implémente la spécification des servlets et des JSP du Java Community Process, est paramétrable par des fichiers XML et de propriétés, et inclut des outils pour la configuration et la gestion ; il comporte également un serveur HTTP.

Tomcat ne constitue qu'un conteneur Web, et non un serveur Web à part entière, il gère spécifiquement les servlets et les JSP ; il peut être également parfois désigné comme moteur de servlets.

Tomcat est en réalité souvent employé en combinaison avec un serveur Web Apache ou d'autres serveurs Web. Tomcat a été écrit en langage java, il peut donc s'exécuter via la machine virtuelle java sur n'importe quel système d'exploitation la supportant.

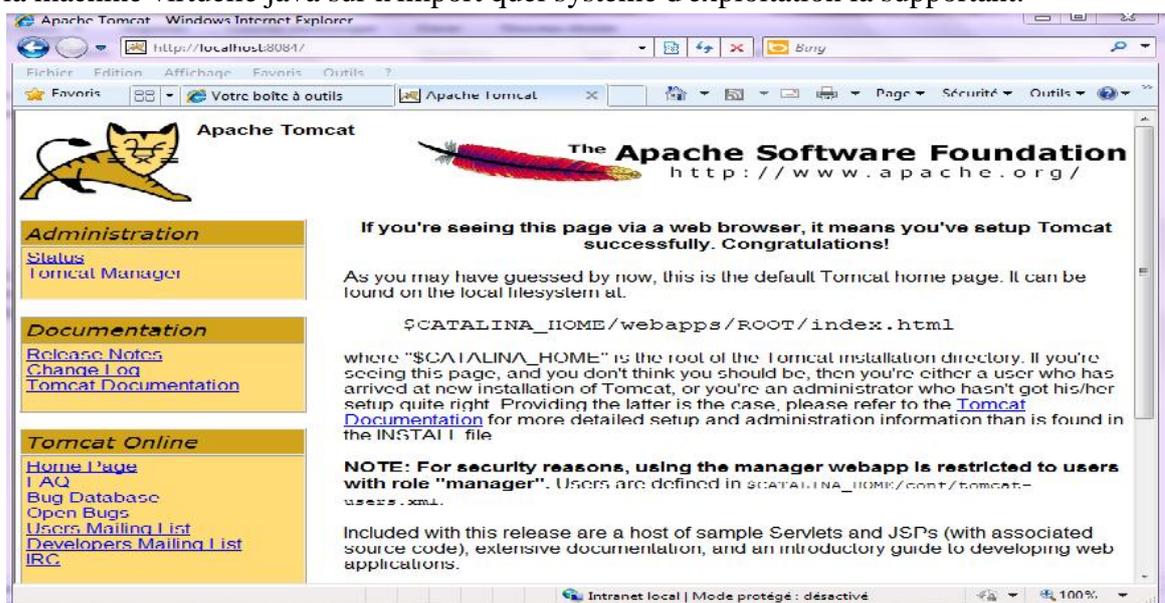


Figure II.1 : l'interface Apache Tomcat

## II.2. Hibernate Framework :[18]

C'est un logiciel, écrit en java, qui permet de faire le mapping entre Objets Java et objets stockés en base relationnelle en assurant la persistance, et aussi s'occupe du transfert des classes Java dans les tables de la BDD et des types de données dans les types de données SQL.

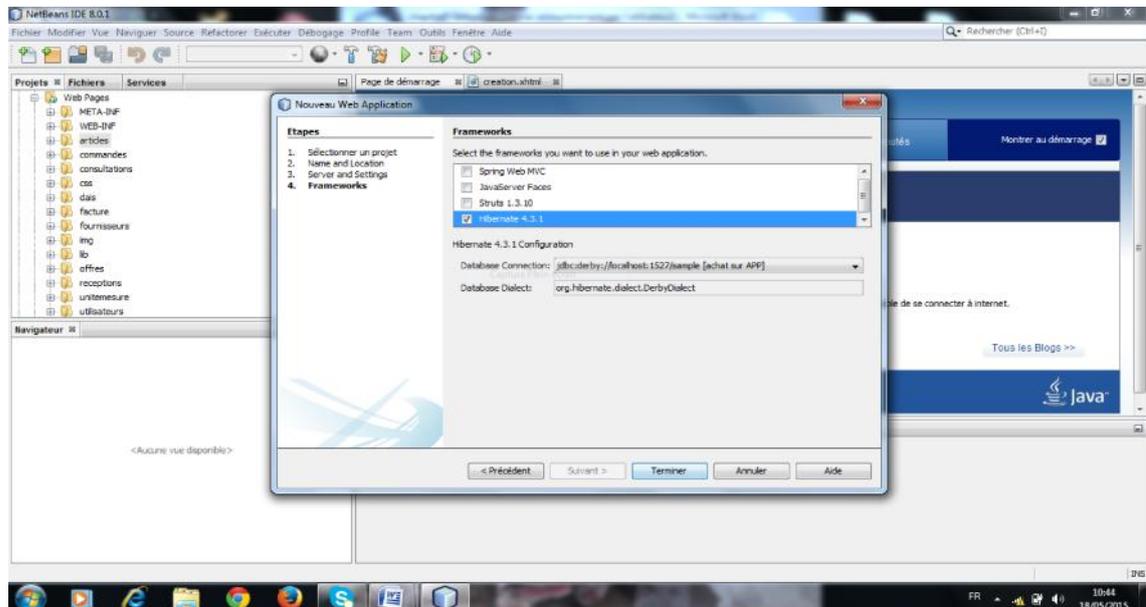


Figure II.2: L'interface de Hibernate.

## II.3. EJB : [19]

La technique Enterprise JavaBeans (EJB) est une architecture de composants logiciels côté serveur pour la plate-forme de développement JEE.

Cette architecture propose un cadre pour créer des composants distribués (c'est-à-dire déployés sur des serveurs distants) écrit en langage de programmation Java hébergés au sein d'un serveur applicatif permettant de représenter des données (EJB dit entité), de proposer des services avec ou sans conservation d'état entre les appels (EJB dit session), ou encore d'accomplir des tâches de manière asynchrone (EJB dit message). Tous les EJB peuvent évoluer dans un contexte transactionnel.

**II.4. JDBC : [20]**

L'API JDBC (Java DataBase Connectivity) permet aux applications Java d'accéder par le biais d'une interface commune à des sources de données pour lesquelles il existe des pilotes JDBC (ex: BDD Oracle, MySQL, ...). Il est constitué d'un ensemble de classes permettant de développer des applications capables de se connecter à des serveurs de bases de données(SGBD).

La combinaison Java/JDBC possède de nombreux avantages nous citerons entre autres les points suivants :

- la possibilité d'écrire du code applicatif pour le traitement des bases de données indépendamment de tout outil ou de tout langage propriétaire.
- la possibilité d'exécuter le code applicatif en question sur n'importe quel type de plate forme disposant d'une machine virtuelle : Windows, Linux, Mac, Paml, Téléphones mobiles ...
- Les dernières versions de JDBC apportent encore de nouvelles possibilités(le traitement de requêtes en opérant des déplacements autres que séquentiels et des mises à jour directes sur les bases de données, des traitements par lots, d'autre types de données, l'utilisation des transactions).

**II.5. JSF: [21]**

JSF (JavaServer Faces) : est un framework de développement d'applications Web en Java permettant de respecter le modèle d'architecture MVC et basé sur des composants côté présentation .il permet de :

- Une séparation de la couche présentation des autres couches (MVC)
- Un mapping entre l'HTML et l'objet
- Un ensemble de composants riches et réutilisables
- une liaison simple entre les actions côté client de l'utilisateur (event listener) et le code Java côté serveur
- Création de nouveaux composants graphiques
- JSF peut être utilisé pour générer autre chose que du HTML (XUL, XML, WML, ...)...etc.

## II.6. NetBeans : [22]

C'est un environnement de développement intégré (IDE) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License). En plus Java, NetBeans permet également de supporter différents autres langages, comme Python, C,C++, XML et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et des pages web). NetBeans est disponible sous Windows, Linux ...

NetBeans comprend un explorateur de bases de données qui supporte toutes les bases relationnelles pour lesquelles un connecteur JDBC existe (selon les versions des gestionnaires de bases de données): JavaDB (Derby) MySQL, PostgreSQL, Oracle, Microsoft SQL, PointBase, IBM Redistributable DB2.

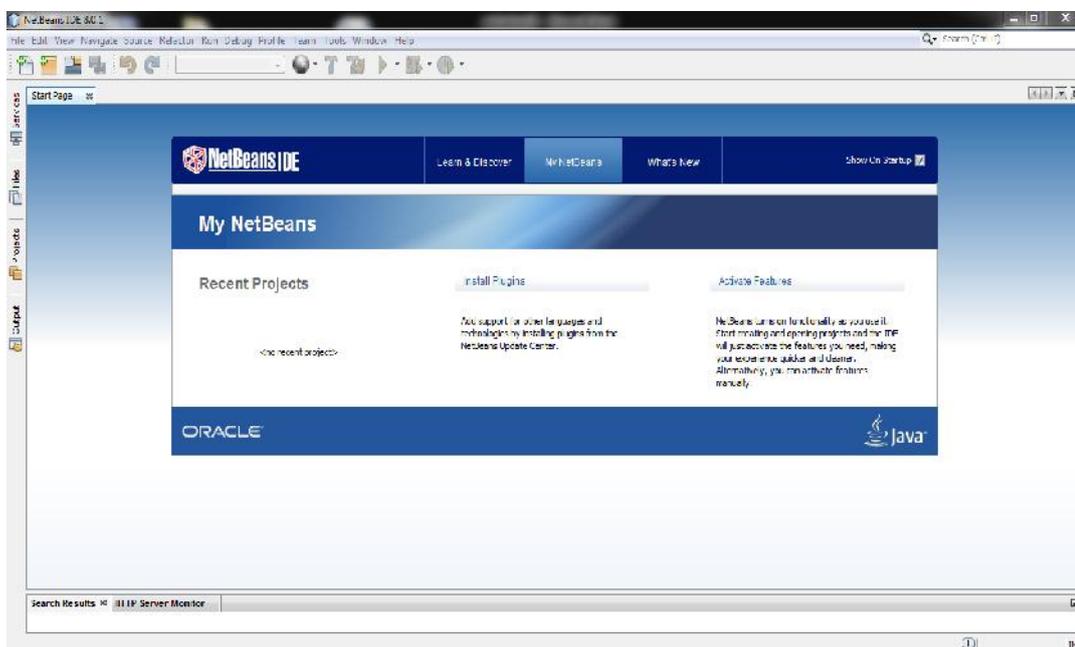


Figure II.4 : L'interface de NetBeans.

## II.7. SGBDR(Oracle) :[23]

Oracle est un système de gestion de base de données relationnelle (SGBDR) qui depuis l'introduction du support du modèle objet dans sa version 8 peut être aussi qualifié de système de gestion de base de données relationnel-objet (SGBDRO). Fourni par Oracle Corporation, il a été développé par Larry Ellison, accompagné d'autres personnes telles que Bob Miner et Ed Oates.

Il est libre de développer, déployer et distribuer, rapide à télécharger ; et simple à administrer. Oracle Database XE peut être installé sur une machine de taille hôte avec un certain nombre de processeurs.

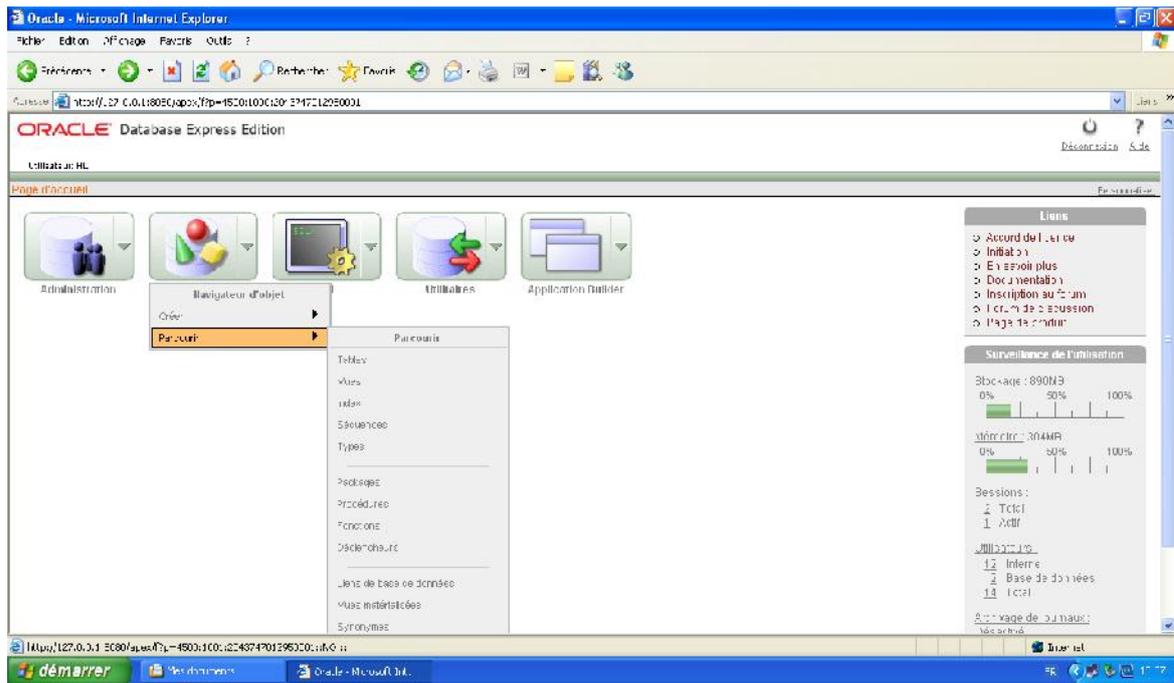


Figure II.5 : Interface d'Oracle.

### III. Les langages de programmation :

#### III.1. Java : [24]

Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au *SunWorld*. La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java.

La particularité et l'objectif central de Java est que les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. Pour cela, divers plateformes et frameworks associés visent à guider, sinon garantir, cette portabilité des applications développées en Java.

### III.2. Java EE : [24]

Java Enterprise Edition, ou **Java EE** (anciennement **J2EE**), est une spécification pour la technique Java de Sun plus particulièrement destinée aux applications d'entreprise.

Dans ce but, toute implémentation de cette spécification contient un ensemble d'extensions au *Framework* Java standard (JSE, Java Standard Edition) afin de faciliter la création d'applications réparties.

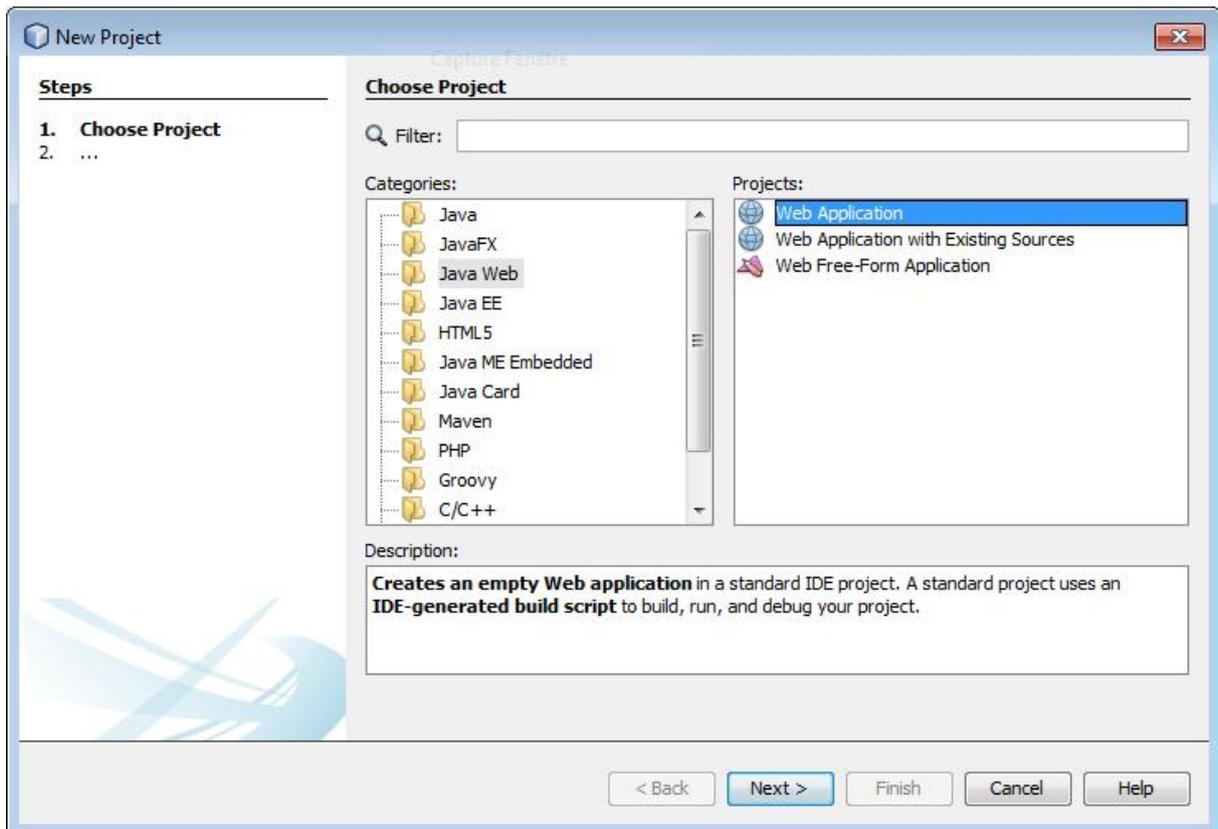


Figure III.2: L'interface de JEE

### III.3. SQL :[25]

SQL ( Structured Query Language) est un langage informatique de base de données conçue pour la gestion données dans les systèmes de gestion de bases de données relationnelles (SGBDR) .

SQL est un langage informatique standardisé qui a été initialement développé par IBM pour l'interrogation, la modification et la définition de bases de données relationnelles, en utilisant les états déclaratifs.

Il peut effectuer les tâches suivantes :

- exécuter des requêtes sur une base de données
- récupérer des données à partir d'une base de données
- insérer des enregistrements dans une base de données
- mettre à jour des enregistrements dans une base de données
- peut supprimer des enregistrements d'une base de données
- créer de nouvelles bases de données
- créer de nouvelles tables dans une base de données
- créer des procédures stockées dans une base de données
- créer des vues dans une base de données
- définir des autorisations sur des tables, des procédures et des vues

### III.4. XHTML : [26]

XHTML(Extensible Hyper text Markup Language) est un langage balisé servant à l'écriture de pages du www (World Wide Web). C'est une évolution de l'HTML (HyperText Markup Language), XHTML respectant la syntaxe définie par XML, plus récente et plus simple que la syntaxe définie par SGML respectée par HTML.

Les documents XHTML sont XML conformes car ils sont facilement visualisés, modifiés et validés avec des outils XML standard. Les documents XHTML peuvent être écrits pour fonctionner mieux qu'auparavant dans les navigateurs existants, ainsi que dans les nouveaux navigateurs. Les documents XHTML peuvent utiliser des applications telles que les scripts et les applets qui dépendent soit du modèle objet de document HTML ou le modèle objet de document XML. XHTML vous donne un format plus cohérent, bien structuré afin que vos pages Web peuvent être facilement analysées et traitées par présents et futurs navigateurs Web.

## IV. Configuration des différentes parties de notre application :

### IV.1. Configuration de NetBeans

#### IV.1.1. Création d'un projet :

- Démarrer NetBeans.
- Créer un projet (Créer une application web : menu File → New Project.) → Java Web → Web Application → donnez un nom pour votre projet → Next.

**IV.1.2. Etablissement de la connexion :**

- Juste après l'étape précédente, Choisir comme serveur Apache Tomcat
- —> Next
- Sélectionner java Server Faces comme web application —> sélectionner le menu components —> choisir PrimFaces —> More —> choisir PrimFaces 5.0 —> OK.
- Sélectionner Hibernate 4.3.1 —> dans Database Connection, choisir puis cliquer sur « New Database Connection »
- Choisir « OracleThin » —> Sélectionner le Driver File « ojdbc14-10g.jar » s'il est disponible sinon il faut l'ajouter en cliquant sur Add —> Next
- Taper le nom de la base de données et son mot de passe dans User Name et Password respectivement
- Cliquer sur Test Connection —> Next —> Next —> Finish.

**IV.2. Configuration de Hibernate :**

- Après l'établissement de connexion, on aura un fichier de configuration « hibernate.cfg.xml » contenant le nom de la base de données et son mot de passe.
- Le fichier de mapping (nom de la classe, extension .hbm.xml) : Hibernate a besoin de savoir comment charger et stocker des objets d'une class persistante. C'est là qu'intervient le fichier de mapping Hibernate qui indique à Hibernate la définition de la structure des tables et les clés primaires, exemple : Module.hbm.xml.
- Le fichier hibernat.reveng.xml : à sa création, il ramène toutes les tables de la base de données.

## V. Présentation de quelques interfaces de notre application :

### V.1. La page d'accueil :

C'est la première page vue par l'utilisateur, elle le conduit vers la page d'authentification



Figure V.1 : page d'accueil

### V.2. La Page d'authentification :

Cette étape permet à l'utilisateur d'accéder à son propre espace en s'identifiant par son «Login» et son «Mot de passe».

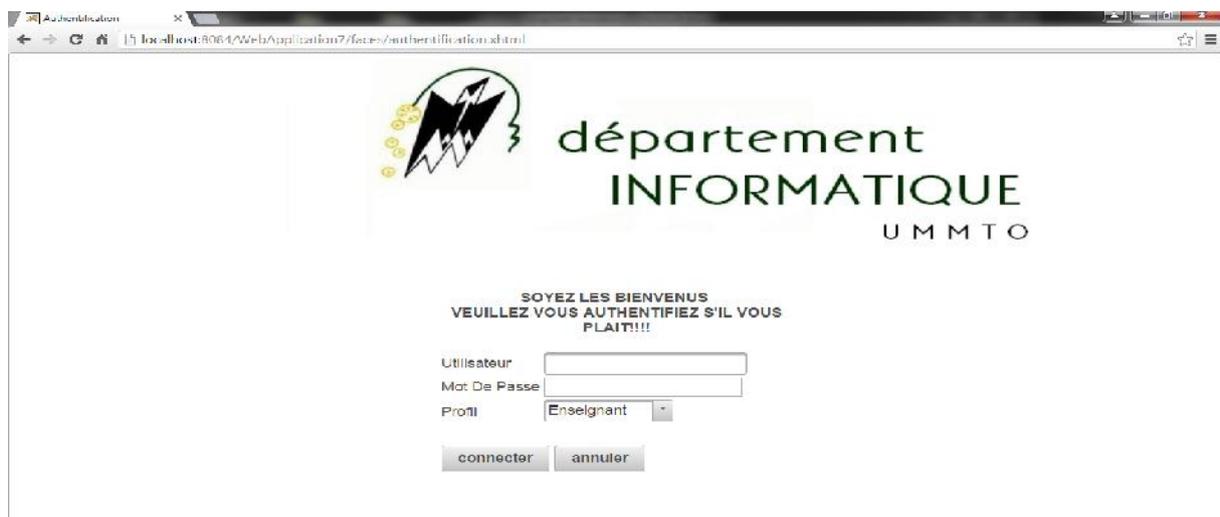


Figure V.2 : page d'authentification

### V.3. L'espace administrateur :

Cette page peut être atteinte par son acteur (administrateur) après avoir effectué une authentification, elle lui permet d'effectuer ses propres tâches.



Figure IV.3 : L'espace administrateur

### V.4. L'espace enseignant :

Cette page peut être atteinte par son acteur (enseignant) après avoir effectué une authentification, elle lui permet d'effectuer ses propres tâches.



Figure V.4 : L'espace enseignant

### V.5. Page Gestion Formation :

Cette fenêtre s'affiche on cliquant premièrement sur le lien »Formation » puis sur le lien « Gestion Formation » dans l'espace administrateur .à partir de là, l'administrateur peut ajouter, modifier, afficher et supprimer une formation.



Figure V.5 : Page Gestion Formation

### V.6. : page ajouter Formation :

Après avoir cliquer sur le bouton « Nouvelle Formation » dans la page Gestion Formation, le formulaire « Creation d'une formation » s'affiche, l'administrateur doit remplir les champs du formulaire puis il clique sur le bouton « Enregistrer Formation ».

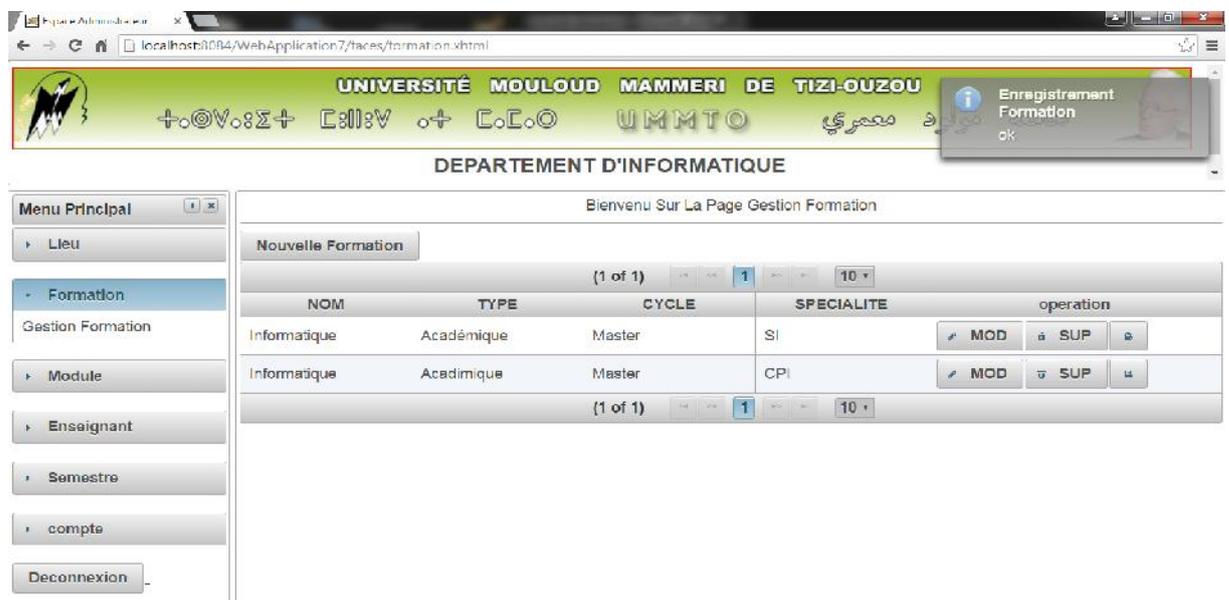
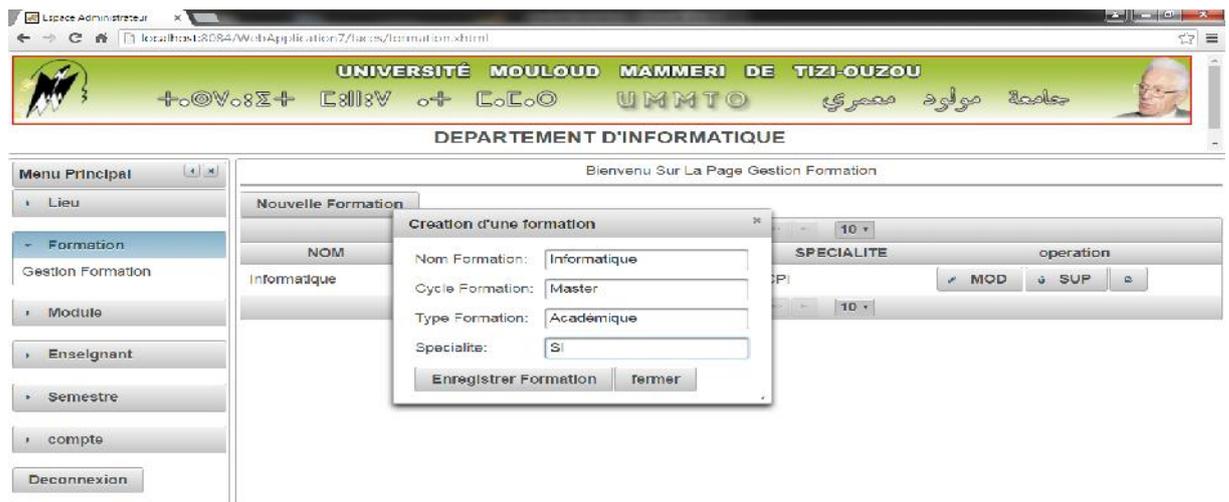
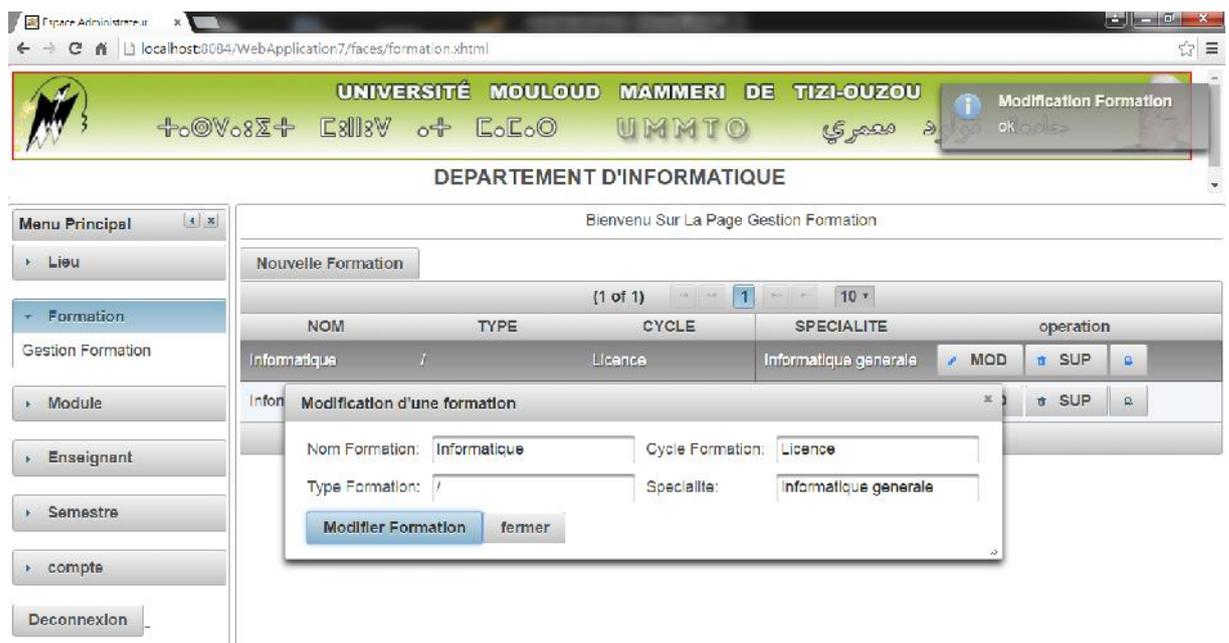
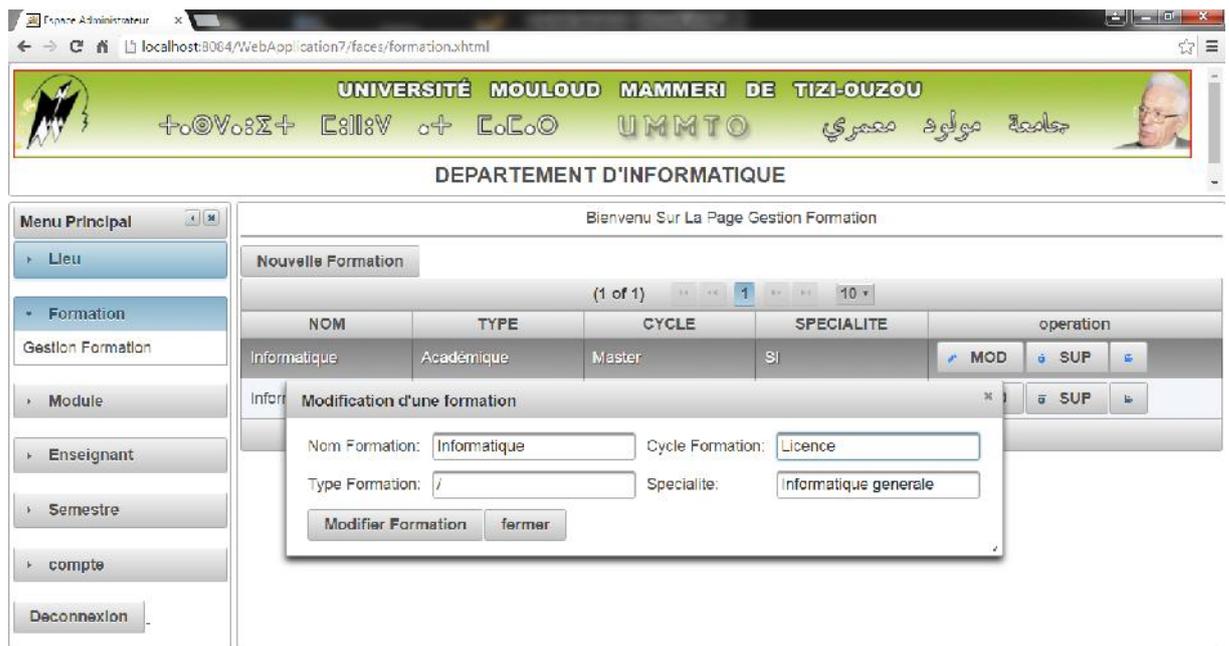


Figure V.6 : page ajouter Formation

### V.7. page modifier Formation :

Après avoir sélectionner une formation dans la page Gestion Formation, l'administrateur clique dur le bouton « MOD », le formulaire « Modification d'une formation » s'affiche, l'administrateur doit modifier les champs du formulaire puis il clique sur le bouton « Modifier Formation ».



**Figure V.7: page modifier Formation**

### V.8. page supprimer Formation :

Après avoir sélectionner une formation dans la page Gestion Formation, l'administrateur clique dur le bouton « SUP », la boîte de dialogue « Confirmation » s'affiche, l'administrateur choisit « Oui » pour supprimer, ou « Non » pour ne pas supprimer la formation.

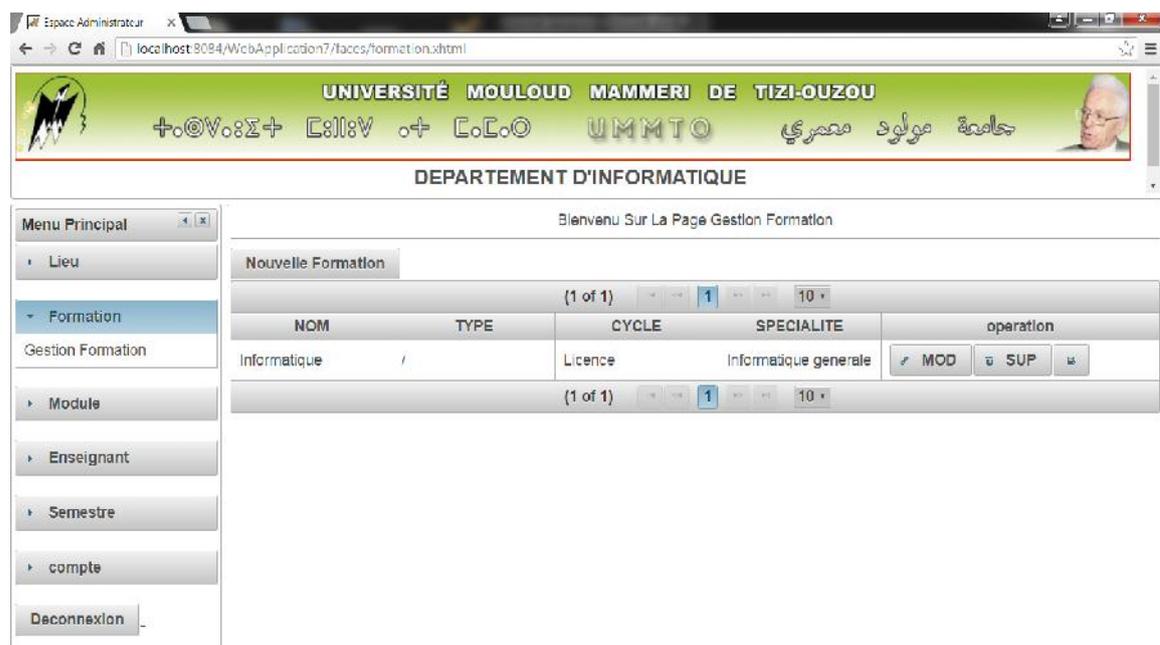
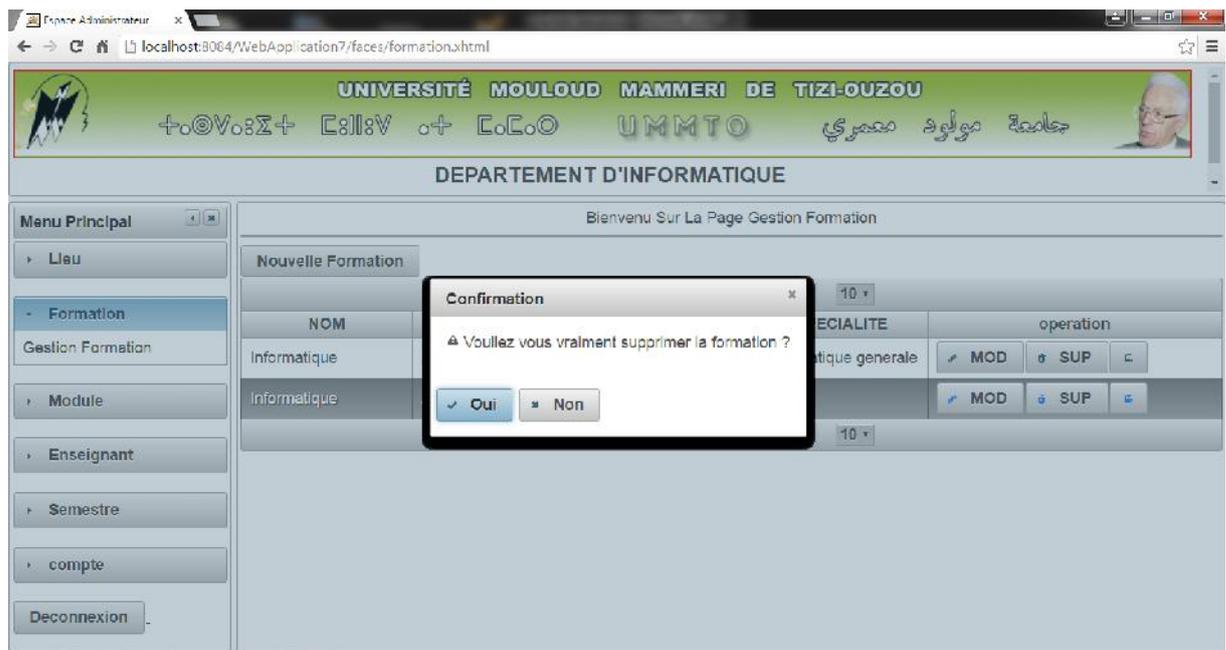


Figure V.8 : page supprimer Formation

### V.9. Page Gestion Enseignant :

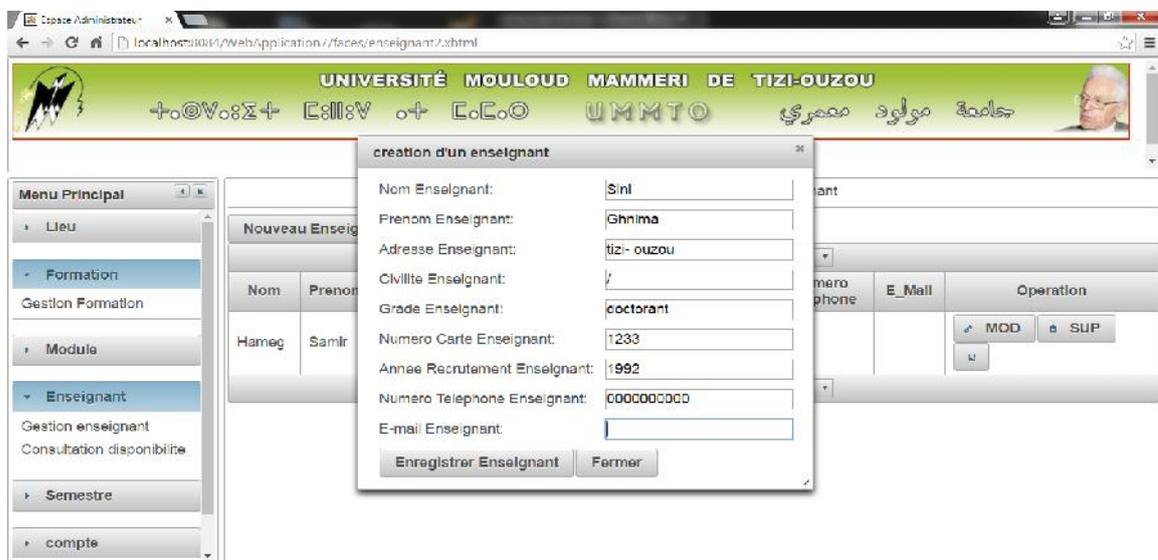
Cette fenêtre s’affiche on cliquant premièrement sur le lien »Enseignant » puis sur le lien « Gestion Enseignant » dans l’espace administrateur .à partir de là, l’administrateur peut ajouter, modifier, afficher et supprimer un enseignant.



Figure V.9 : Page Gestion Enseignant

#### V.10. Page ajouter Enseignant :

Après avoir cliquer sur le bouton « Nouveau Enseignant » dans la page Gestion Enseignant, le formulaire « Creation d'un enseignant » s'affiche, l'administrateur doit remplir les champs du formulaire puis il clique sur le bouton « Enregistrer Enseignant ».



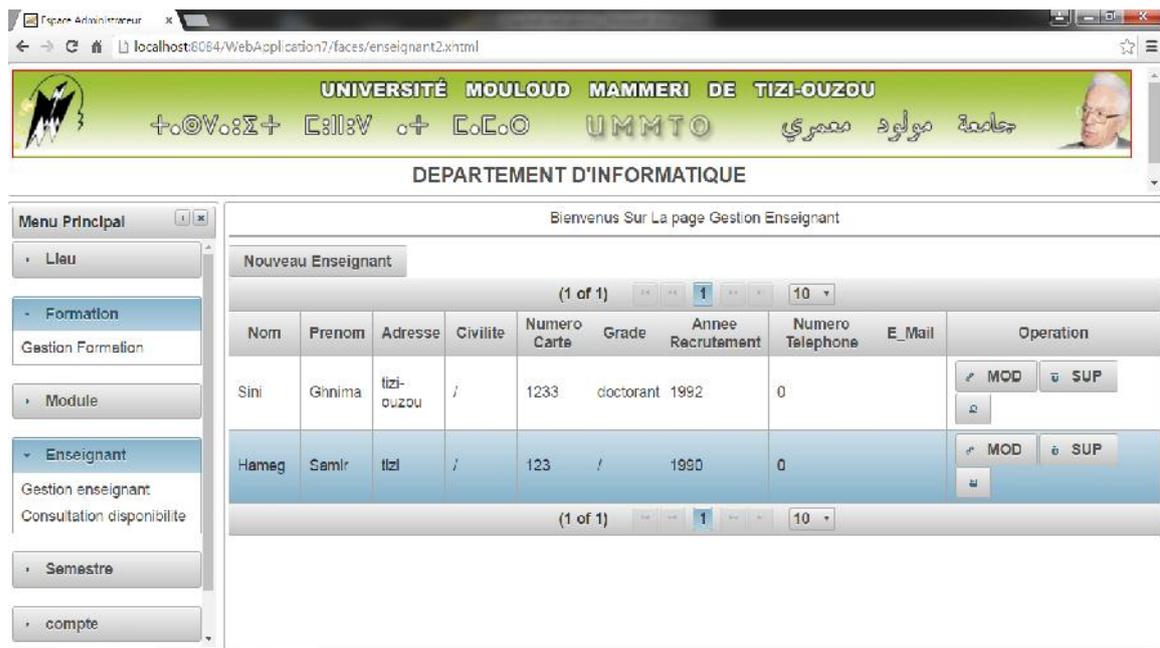


Figure V.10 : page ajouter Enseignant

**V.11. Page afficher Enseignant :**

Après avoir sélectionner un enseignant dans la page Gestion Enseignant, l'administrateur clique sur le bouton d'affichage, la fenêtre « Edition d'un enseignant » s'affiche ,elle contient toute les informations de l'enseignant.



Figure V.11 : page afficher Enseignant

### V.12. Page Gestion Disponibilite Enseignant :

Cette fenêtre s'affiche on cliquant sur le lien »Enseignant » dans l'espace enseignant .à partir de là, l'administrateur peut ajouter, modifier, afficher et supprimer sa disponibilité.



Figure V.12 : page Gestion Disponibilite Enseignant.

### V.13. Page saisie disponibilite Enseignant :

Après avoir cliquer sur le bouton « Nouvelle Disponibilite » dans la page Gestion Disponibilite Enseignant, le formulaire « Creation d'un disponibilité » s'affiche, l'enseignant doit remplir les champs du formulaire puis il clique sur le bouton « Enregistrer Disponibilite».



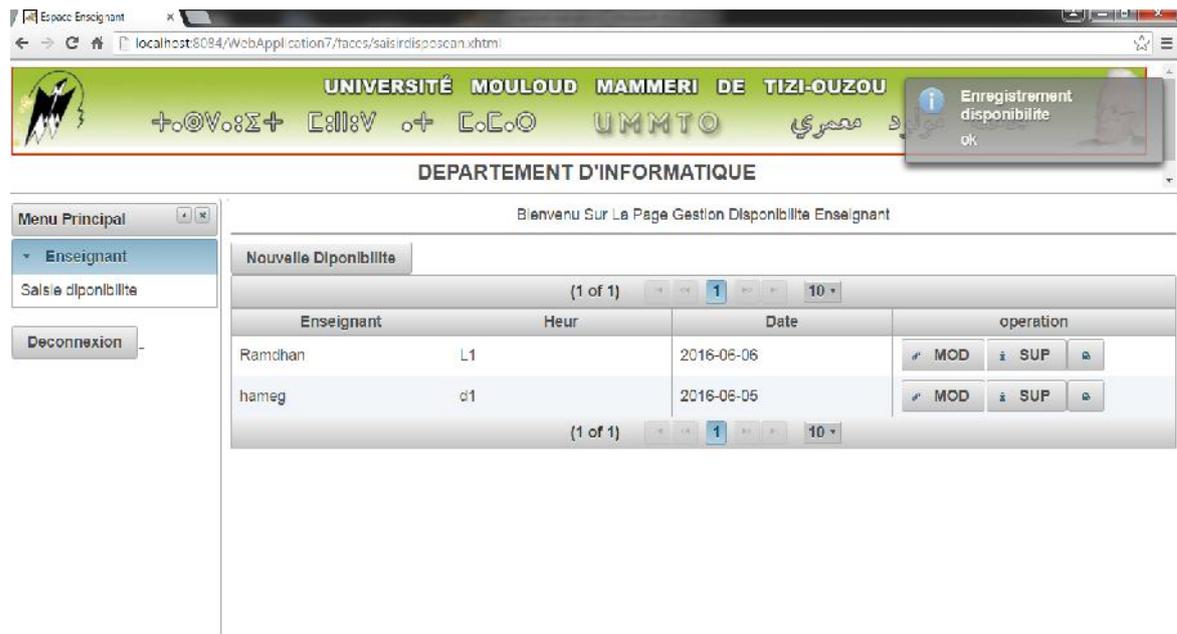


Figure V.13 : page saisie disponibilité Enseignant

#### V.14. Page Consulter Disponibilité Enseignant :

Cette fenêtre s'affiche on cliquant premièrement sur le lien »Enseignant » puis sur le lien «Consultation Disponibilité » dans l'espace administrateur .à partir de là, l'administrateur peut consulter la liste des enseignants disponibles.



Figure V.14 : page Consulter disponibilité Enseignant

**VI. Conclusion :**

Dans ce chapitre, nous avons présenté d'abord l'environnement et les outils utilisés pour implémenter notre application. Enfin, nous avons présenté quelques interfaces de notre application.

Nous avons en outre explicité les composants de la base de données, puis pour terminer nous avons présenté quelques interfaces de l'application.

# Conclusion Générale

L'objectif de cette étude est de concevoir et réaliser une application web pour la gestion des activités pédagogiques de département d'Informatique, cette application doit effectuer les tâches administratives récurrentes, qui sont la plupart du temps réalisées manuellement ou qu'un ordinateur peut les faire en une fraction de seconde.

Dans ce travail, nous avons voulu trouver des solutions à des problèmes de performance, de temps et de disponibilité.

L'informatisation permet aussi d'offrir un espace de travail convivial, stockage conséquent de l'information en limitant l'espace d'archivage, ainsi un bon fonctionnement du département.

Nous avons, au cours de ce mémoire, contribué à la réalisation d'une application de gestion du département d'Informatique ummto, Pour la réalisation de cette dernière nous avons jugé utile de décomposer notre travail en quatre chapitres

- Le premier chapitre concerne la présentation de langage de programmation javaEE.
- Le deuxième chapitre concerne l'organisme d'accueil.
- Le troisième chapitre concerne l'analyse et la conception de notre système en utilisant L'extension d'UML ainsi que la conception et l'implémentation de notre base de données en recensant les différentes informations manipulées.

Et enfin nous avons terminé par la présentation des différents outils de développement utilisés lors de la mise en œuvre effective de l'application, ainsi que la description de son fonctionnement illustré par ses différentes interfaces.

Le travail présenté nous a permis d'affronter des problèmes concrets, en mettant en pratique nos connaissances acquises lors de notre cursus de Master en informatique, et nous a permis en outre, de bénéficier d'une première expérience dans le monde professionnel. Arrivés à terme de ce travail, nous souhaitons que notre application réponde aux besoins des utilisateurs.

# Bibliographie/webliographi

- [1] : [https://fr.wikipedia.org/wiki/Java\\_EE](https://fr.wikipedia.org/wiki/Java_EE).
- [2] : Mémoire fin d'étude (Conception et réalisation d'une application Java EE pour gestion du matériel) promotion 2014/2015.
- [3] : <http://gerardnico.com/wiki/language/java/j2ee>
- [4] : Développement Web et J2EE (P. André)
- [5] : <http://www.jmdoudoux.fr/java/dej/chap-j2ee-javaee.htm>
- [6] : Java Server Faces Tutorial
- [7] : <http://www.jmdoudoux.fr/java>
- [8] : Java EE Java Enterprise Edition /Pierre-Yves Gibello
- [9] : <http://www.jmdoudoux.fr/java/dej/chap-frameworks.htm>
- [10] : <html://fr.Wikipedia.org/Wiki/Hibernate>
- [11] : Programmation Web Avancée Hibernate /Thierry Hamon
- [12] : [http://www.tutorialspoint.com/hibernate/hibernate\\_tutorial.pdf](http://www.tutorialspoint.com/hibernate/hibernate_tutorial.pdf)
- [13] : Mémoire fin d'étude (Développement d'un système d'information Pour le suivi des enseignements) Promotion 2008 /2009
- [14] : <http://www.ummtto.dz>
- [15] : Mémoire fin d'étude Conception et réalisation d'une application pour la gestion d'un cabinet médical) promotion 2014/2015
- [16] : Modélisation avec UML
- [17] : Mémoire fin d'étude (Conception et réalisation d'une Application Java EE sur la gestion des achats locaux et importation) Promotion:2014/2015
- [18] : Mémoire fin d'étude (Conception et réalisation d'un service web pour la gestion des écarts des stations de service) promotion 2014/2015.
- [19], [22] : Conception et réalisation d'un web service pour la gestion des dossiers médicaux promotion (2009/2010)

**[20] : Java EE Approfondi (cours de 2eme année ingénieur Spécialisation »Génie Informatique)**

**[21] : Java Server Faces(JSF) Tutorial**

**[23] : <http://www.oracle.com>**

**[24] : Mémoire fin d'étude (*Conception* et réalisation d'une Application Java EE sur la gestion des achats locaux et importation) promotion 2009/2010**

**[25]: SQL Data Definition Database Systems /Natasha Alechina**

**[26]:[http://www.tutorialspoint.com/xhtml/what\\_is\\_xhtml.htm](http://www.tutorialspoint.com/xhtml/what_is_xhtml.htm)**

## I. Introduction :

Pour faire face à la complexité croissante des logiciels, des nouvelles méthodes et outils de modélisation ont été créés.

La modélisation par objet saisit la structure statique d'un système, en montrant les objets dans le système, leurs relations, ainsi que les attributs et les opérations qui caractérisent chaque classe. Ce mode de conception de système prend son intérêt par rapport à la programmation traditionnelle puisque le modèle objet correspond étroitement au monde réel.

Il est par conséquent plus souple pour les modifications. Dans ce contexte et devant le foisonnement de nouvelles méthodes de conception « orientée objet », l'Object Management Group (OMG) a eu comme objectif de définir une notation standard utilisable dans les développements informatiques basés sur l'objet. C'est ainsi qu'apparu l'UML.

## II. UML:

### II.1. Historique

Au début des années 90, une cinquantaine de méthode objet ont vue le jour. Ce qui est un signe de l'intérêt du sujet, et également de confusion. Toutes ces méthodes utilisaient à peut près les mêmes concepts de classes, d'association, de partitions en sous-systèmes ...

En octobre 1994, Grady Booch et Jim Rumbaugh fondèrent la Rational Software Corporation pour unir leurs efforts en vue de créer une norme industrielle unique à partir de leurs méthodes. C'est ainsi que est née la méthode unifiée (Unified Method) en octobre 1995. Ils sont ensuite rejoints par Ivar Jacobson, l'inventeur des cas d'utilisation (Uses cases), et ont publié UML 0.9 en juin 1996, puis UML 1.0 en janvier 1997.

La version 1.1, mise au point avec d'autres partenaires, devient un standard en novembre 1997 lorsqu'elle est acceptée par l'OMG (Object Management Groupe). La poursuite du développement d'UML fut ensuite intégralement remise aux mains l'OMG. En juillet 1998, l'OMG publiait UML 1.2, puis UML 1.3 en juin 1999. En mai 2002, UML 1.4, qui présentait de petites améliorations et quelques extensions, voyait le jour. UML 1.5 a été publié en mars 2003 avec également quelques correctifs.

UML 2.0, publié en 2005, présente quand à lui de profondes modifications et de variables extensions par rapport aux versions 1.x, comme par exemple le diagramme d'activité et le diagramme de séquences.

## II.2. définition :

L'UML (Unified Modeling Language), est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.

L'UML est le résultat de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard adopté par l'Object Management Group (OMG).

## II.3. La modélisation UML :

UML fournit un ensemble d'outils permettant de représenter des éléments du monde objet (classes, objets, ...) ainsi que les liens qui les relient. Toutefois, étant donné qu'une seule représentation est trop subjective, UML fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux vues. Une vue est constituée d'un ou plusieurs diagrammes.

## II.4. Formalisme d'UML :

UML propose 13 types de diagrammes. UML n'étant pas une méthode, leur utilisation est laissée à l'appréciation de chacun, même si le diagramme de classes est généralement considéré comme l'élément central d'UML, on peut se contenter de modéliser partiellement un système.

Il se décompose en plusieurs sous-ensembles :

- **Les vues** : Les vues sont les observables du système. Elles décrivent le système d'un point de vue donné, qui peut être organisationnel, dynamique, temporel, architectural, logique, etc. En combinant toutes ces vues, il est possible de définir (ou retrouver) le système complet.
- **Les diagrammes** : Les diagrammes sont des éléments graphiques. Ceux-ci décrivent le contenu des vues, qui sont des notions abstraites. Les diagrammes peuvent faire partie de plusieurs vues.
- **Les modèles d'élément** : Les modèles d'élément sont les briques des diagrammes UML, ces modèles sont utilisés dans plusieurs types de diagramme. Exemple d'élément : cas d'utilisation, classe, séquence, etc.

➤ **UML suit une démarche en trois étapes :**

- Analyse
- Conception
- Implémentation

➤ **Les briques d'UML : composées de :**

- Les éléments
- Les relations
- Les diagrammes

**II.5. Présentation générales de différents diagrammes UML :**

Un digramme est la représentation graphique d'un ensemble d'éléments que l'on représente en général par un graphe reliant les sommets (éléments) avec des arcs (relations). Il donne ainsi à l'utilisateur le moyen de visualiser et de manipuler des éléments de modélisation. UML s'articule autour de treize(13) types de diagrammes dans sa version 2, chacun d'eux étant dédié à la représentation des concepts particuliers d'un système logiciel.

Ces diagrammes sont regroupés dans deux grands ensembles

### II.5.1 Selon les trois points de vue de modélisation :

Il existe trois points de vue classiques de modélisation : fonctionnelle, statique et dynamique.

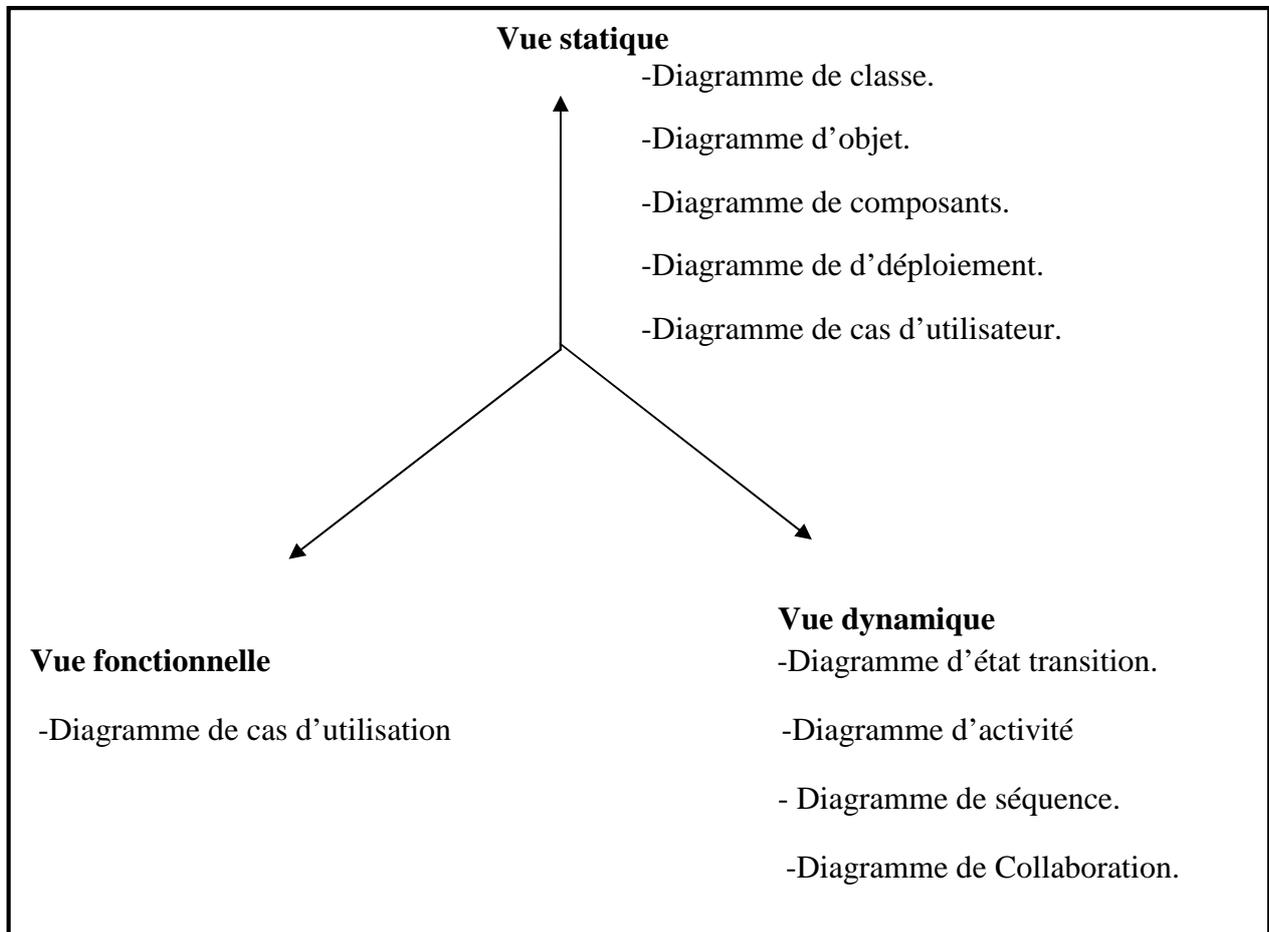


Figure II.5.1 : Composantes de la modélisation UML.

### II.5.2. Selon les étapes :

#### II.5.2.1. Diagrammes statiques (structurels) :

- **Diagramme de classe** : Ce diagramme exprime la structure statique du système en termes de classes et de relations entre ces classes. L'intérêt du diagramme de classe est de modéliser les entités du système d'information.

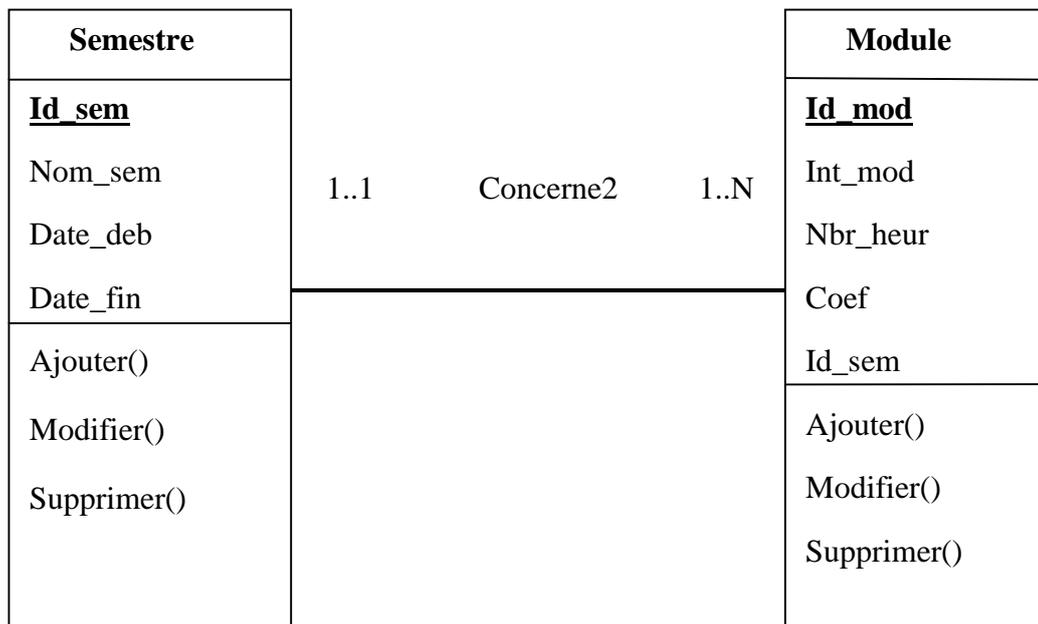
La description du diagramme des classes est fondée sur :

- Le concept d'objet,

- Le concept de classe comprenant les attributs et les opérations,
- Les différents types d'association entre classes.

✓ **Concepts de base :**

- **Classe** : c'est un ensemble d'éléments ayant les mêmes attributs, les mêmes opérations, les mêmes relations et la même sémantique.
- **Objet** : un objet est une instance d'une classe au cours d'exécution d'un Programme.
- **Association** : elle permet d'ajouter des attributs et des opérations à l'association. Elle est représentée dans l'UML par une classe reliée par un trait en pointillée sur l'association concernée.
- **Attribut** : Un attribut décrit une donnée de la classe, Pour chaque objet d'une classe l'attribut prend une valeur (sauf cas d'attribut multivalué).
- **Opération** : est un service offert par la classe (un traitement que les objets correspondant peuvent effectuer).
- **interface** : Une interface est un ensemble d'opérations utilisées pour définir un service d'une classe ou d'un composant.



**Figure II.5.2.1 : Représentation de diagramme de classe en UML**

- **Diagramme d'objet** : Ce diagramme sert à représenter les instances de classes (objets) utilisées dans le système.
- **Diagramme de composants** : Ce diagramme permet de montrer les composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre (fichiers, bibliothèques, bases de données...)
- **Diagramme de déploiement** : Ce diagramme sert à représenter les éléments matériels (ordinateurs, périphériques, réseaux, systèmes de stockage...) et la manière dont les composants du système sont répartis sur ces éléments matériels et interagissent avec eux.
- **Diagramme de cas d'utilisation** : (use-cases) Ce diagramme permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système.
- **Diagramme états-transitions**: Ce diagramme représente le comportement d'une classe ou d'un composants en termes d'états.
- **Diagramme d'activité** : Ce diagramme permet de décrire sous forme de flux ou d'enchaînement d'activités le comportement du système ou de ses composants.
- **Diagramme de séquence**:Ce diagramme s'agit ici de décrire la séquence d'interaction entre l'acteur et le logiciel pour réaliser le cas d'utilisation Pour chaque cas d'utilisation on a élaboré les différents diagrammes de séquences.
- **Diagramme de communication** (depuis UML 2.x): (*Communication Diagram*) : Ce diagramme représentation simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets.
- **Diagramme global d'interaction** (depuis UML 2.x) : Ce diagramme permet de décrire les enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences (variante du diagramme d'activité).
- **Diagramme de temps** (depuis UML 2.x) : Ce diagramme permet de décrire les variations d'une donnée au cours du temps.
- **Diagramme de paquetage (nouveau dans UML 2)** :  
Ce diagramme montre l'organisation logique du modèle et les relations entre packages. Chaque paquetage représente un ensemble homogène d'éléments du système (classe, composants...).
- **Diagramme de structure composite (nouveau dans UML 2)** :  
Ces diagrammes représente les composants physiques d'une application. Un composant peut être du code, un script...

### II.5.2.2. Diagrammes de comportement (dynamique):

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

➤ **Diagramme des cas d'utilisation** : Ce diagramme représente les cas d'utilisations identifiés et l'acteur associé à chacun. Ils permettent de représenter les processus du système.

❖ **Caractéristiques d'un cas d'utilisation** :

- **Les acteurs** : Un acteur représente un ensemble de rôles joués par les utilisateurs des cas d'utilisation en interaction avec ces cas d'utilisation. En règle générale, un acteur représente un rôle qu'un homme, une machine ou même un autre système joue avec le système.

➤ **Représentation du diagramme des cas d'utilisation**

La représentation d'un cas d'utilisation met en jeu trois concepts :

- L'acteur
- Le cas d'utilisation
- L'interaction entre l'acteur et le cas d'utilisation

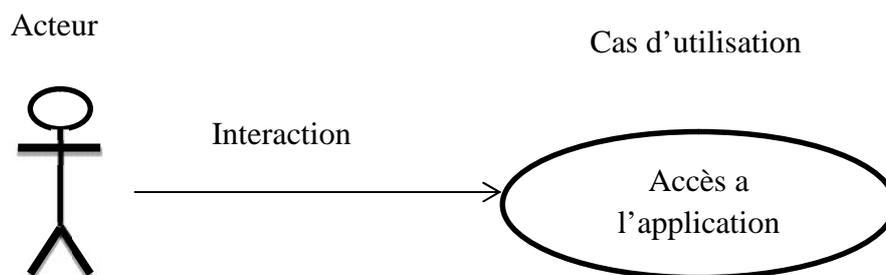


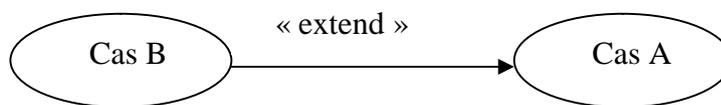
Figure II.5.2.2 : Représentation d'un cas d'utilisation en UML

**❖ Relation entre cas d'utilisation:**

On a trois (03) types de relations entre cas d'utilisation :

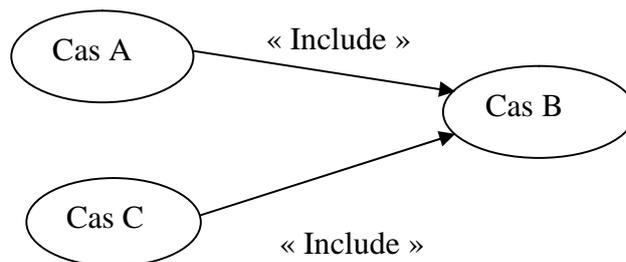
**L'extension « extend » :** Elle indique que le cas d'utilisation source ajoute son comportement au cas d'utilisation destination. L'extension peut être soumise à des conditions.

**Formalisme :** Une relation d'extension d'un cas d'utilisation A par rapport à un cas d'utilisation B signifie qu'une instance de A peut être étendue par le comportement décrit dans B.



**L'inclusion « include » :** Elle indique que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination. Cette relation permet de décomposer des comportements et de définir les comportements partageables entre plusieurs cas d'utilisations.

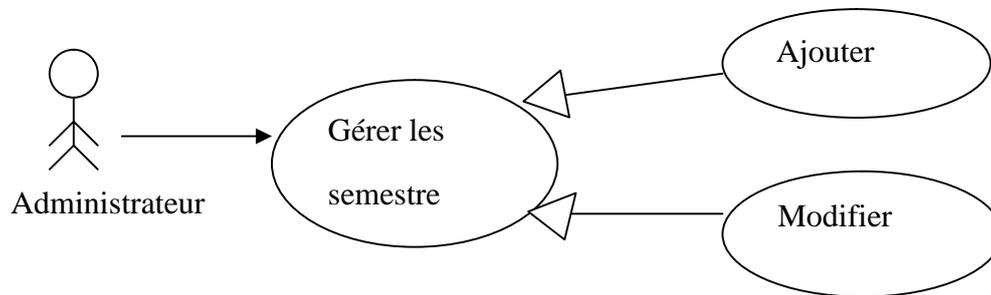
**Formalisme :** Une relation d'inclusion d'un cas d'utilisation A par rapport à un cas d'utilisation B signifie qu'une instance de A contient le comportement décrit dans B.



Les cas A et C ont recourt au cas B

**La généralisation** : Cette relation introduit la notion d'héritage, cela permet de décrire différentes variantes d'un cas général.

**Formalisme :**



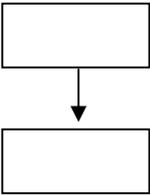
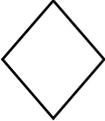
➤ **Diagramme d'état-transition (machine d'état) :**

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

➤ **Diagramme d'activités :**

Ce diagramme Il montre l'enchaînement des actions et décisions au sein d'une activité «organigramme amélioré».Il permet aussi de représenter les flots de contrôle et les flots de données.

Pour élaborer le diagramme d'activité nous allons utiliser le formalisme suivant :

Symbole	Désignation
	Etat initial
	Message
	Activité
	Transition d'état
	Etat final
	Nœud de décision(Choix)

**Tableau 1 : formalisme du diagramme d'activité.**

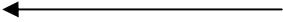
### ❖ Concepts de base :

- **Activité** : représente une exécution d'un mécanisme, ou un déroulement d'étapes séquentielles.
- **Etat** : représente une conjonction instantanée des valeurs des attributs d'un objet.
- **État initial** : L'état initial est un pseudo état qui indique l'état de départ, Lorsqu'un objet est créé, il entre dans l'état initial.
- **État final** : L'état final est un pseudo état qui indique que le diagramme d'états-transitions, est terminé.
- **Action** : Une action correspond à un traitement qui modifie l'état du système. Cette action peut être appréhendée soit à un niveau élémentaire proche d'une instruction en termes de programmation soit à un niveau plus global correspondant à une ou plusieurs opérations.
- **La transition** : représente le passage instantané d'un état vers un autre, c'est aussi l'arrivée d'un événement qui conditionne la transition , un événement peut être une expression logique, exprimée en langage naturel entre deux crochets.
- **Nœud de décision(Choix)** : C'est un élément physique. Lors d'une exécution, il représente une ressource ayant une capacité de calcul. En règle général, il a moins de la mémoire et souvent de capacité de traitement .Un nœud est représenté par un cube.

➤ **Diagramme de séquence :**

Ce diagramme permet de représenter les interactions entre objets selon un point de vue temporel, la communication se fait par envoi de messages.

Pour élaborer le diagramme de séquence nous allons utiliser le formalisme suivant :

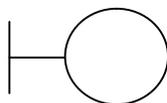
Symbole	Désignation
	Acteur
	Objet
	Message
	Réponse

**Tableau 2: Le formalisme du diagramme de séquence.**

Les classes d'analyse peuvent être répertoriées dans trois catégories d'objets :

- **Objet interface :** ils représentent l'interface entre l'acteur et le système ou les pages Web complètes dans le cas d'application Web.

L'icône :



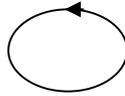
- **Objet entité :** ce sont des objets d'écrits ans un cas d'utilisation mais qui lui survivrons, c'est-à dire qui se trouveront dans d'autre cas d'utilisation.

L'icône :



- **Objets contrôle** : ils représentent le processus, c'est-à-dire les activités systèmes tel qu'un calcul ou une recherche, ils dirigent les objets entité et interface.

L'icône :



❖ **Diagramme de communication (anciennement appelé collaboration) :**

Ce diagramme est une autre représentation des scénarios des cas d'utilisation qui se concentre sur les échanges de messages entre les objets dans le plan au sein d'une interaction.

❖ **Diagramme global d'interaction (nouveau dans UML 2) :**

Il fusionne les diagrammes d'activité et de séquence pour combiner des fragments d'interaction avec des décisions et des flots.

❖ **Diagramme de temps (nouveau dans UML 2) :**

Il fusionne les diagrammes d'états et de séquence pour décrire les variations de l'état d'un objet au cours du temps.

## II.6. Les points forts et points faible d'UML

### II.6.1. Les points forts :

- UML est un langage formel et normalisé, gain de précision, gage de stabilité encourage l'utilisation d'outils
- UML est un support de communication performant, il cadre l'analyse, il facilite la compréhension de représentations abstraites complexes, son caractère polyvalent et sa souplesse font un langage universel.
- UML permet de modéliser de manière claire est plus intuitive, plus expressive, et précise la structure et le comportement indépendant de toute méthode ou de tout langage de programmation.
- UML est en particulier conçue pour être lisible sur des supports courants et variés, comme le papier, les écrans d'ordinateur, les tableaux blancs, etc.

### **II.6.2. Les points faibles :**

- La mise en pratique d'UML nécessite un apprentissage et passe par une période d'adaptation.
- La nécessité de s'accorder sur des modes d'expression communs est vitale en informatique. UML n'est pas à l'origine des concepts objets, mais en constitue une étape majeure, car il unifie les différentes approches et en donne une définition plus formelle.
- Le processus (non 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.

### **III. Conclusion :**

L'UML, comme l'on a vu à travers ce chapitre, ne propose pas une démarche objet mais une notation adapté au monde de développement orienté objet. Il nous a donc permis de s'initier aux techniques de modélisation objet. La notation UML peut s'adapter a tous les projets informatiques.