

# RAPPORT DE STAGE DE FIN D'ETUDES

Pour l'obtention de la

«Licence Appliquée en Sciences et Technologies de l'Information et de  
Communication (LASTIC)»

Présenté par :

**CHAIEB Salwa**

## Titre

Conception et réalisation d'une application d'aide à  
la génération des emplois du temps

Soutenu le : 19/09/2015

Devant le jury :

**Président:** Mr. BEN BRAIEK Ezzedine

**Encadreur :** Mme BEN HMIDA Faten

**Rapporteur :** Mme HOUAIDIA Chiraz

Année Universitaire : 2014 / 2015

# Remerciement

Je tiens à exprimer mes remerciements à tous ceux qui ont rendu ce travail possible. Leurs aides précieuses, leurs conseils fructueux et leurs encouragements, tout au long de l'élaboration de ce projet de fin d'études, m'a permis de le réaliser dans la meilleure considération.

Je veux rendre un hommage particulier à mes encadrants:

**Mme BEN HMIDA Faten & Mr SAIDANE Mhamed**

Pour leurs soutiens et leurs conseils précieux.

Aux membres du jury qui ont bien voulu me honorer de leur présence d'évaluer notre travail.

Un grand merci à toutes les personnes qui m'ont soutenue de près ou de loin au cours de la réalisation de ce modeste travail.

Merci pour tous

## Dédicace

*A ceux qui ont contribué à l'élaboration de ce  
travail  
Et ceux à qui je dois tant*

*Ma plus grande gratitude et tout mon amour à mes  
parents, qui ont su me faire confiance, me soutenir  
et m'encourager au cours de ma vie.*

*A mes enfants, mes sœurs et frères  
A TOUS QUI M'AIMENT*

*A tous mes amis en qui j'ai toujours trouvé le  
soutien et le réconfort.*

# Résumé

La génération des emplois du temps universitaire peut être décrite comme l'allocation des ressources (Séances, Salles, ...) à des types d'enseignement (Séances de cours, TD et TP), tout en essayant de satisfaire un ensemble de contraintes et en évitant le maximum possible les nombres des séances de chevauchements.

**Mots Clés :** Problème Emploi du temps, chevauchement,

## ملخص

يعتبر إحداث جداول الأوقات داخل أي مؤسسة جامعية من أهم الأحداث المنظمة لحسن سير السنة الجامعية من خلال توزيع الحصص، القاعات و نوعية الدروس (دروس نظرية، أشغال مسيرة، أشغال تطبيقية) مع محاولة عدم الوقوع في الخلط

الكلمات المفاتيح: مشكلة ، جدول اوقات ، تضارب

## Abstract :

Automatic generation of university Timetabling can be described as the allocation of resources (time slots, rooms ...) to events (lesson, TD and TP), while trying to meet a set of constraints.

**Keywords:** Problems, Schedule, compelled.

# Table des matières

<b>TABLE DES MATIERES .....</b>	<b>1</b>
<b>TABLES DES FIGURES .....</b>	<b>3</b>
<b>INTRODUCTION GENERALE .....</b>	<b>5</b>
CADRE GENERAL.....	5
TRAVAIL DEMANDE .....	5
PLAN DU RAPPORT.....	5
<b>CHAPITRE I : ETUDE, SPECIFICATION ET ANALYSE DES BESOINS .....</b>	<b>7</b>
1.1.    INTRODUCTION.....	8
1.2.    ETUDE DE L'EXISTANT.....	8
1.3.    CRITIQUE DE L'EXISTANT .....	9
1.4.    SPECIFICATION DES BESOINS.....	9
1.4.1.    Introduction.....	9
1.4.2.    Spécification des besoins.....	10
1.3.2.1    Les acteurs.....	10
1.4.3.    Les besoins fonctionnels .....	11
1.4.4.    Les diagrammes de cas d'utilisation.....	12
1.4.4.1.    Diagramme de Cas d'utilisation : L'administrateur .....	13
1.4.4.2.    Diagramme de Cas d'utilisation : Directeur de département.....	15
1.4.4.3.    Diagramme de Cas d'utilisation : Cas Enseignant .....	16
1.4.4.4.    Diagramme de Cas d'utilisation : Etudiant : .....	17
1.4    ANALYSE DE LA VUE DYNAMIQUE .....	18
1.4.1    Diagramme de séquence : « Authentification ».....	18
1.4.2    Diagramme de Séquence : « Ajout Département » :.....	18
1.4.3    Diagramme de séquence « Modifier un département » .....	19
1.4.4    Diagramme de séquence : « Supprimer un département ».....	19
1.4.5    Diagramme de séquence « Affectation en tant que Vacant » :.....	20
1.4.6    Diagramme de séquence : « Affectation Charge enseignant » :.....	21
1.4.7    Diagramme de séquence : « Consultation Emploi » :.....	21
<b>CHAPITRE II: CONCEPTION .....</b>	<b>22</b>
2.1    INTRODUCTION.....	23
2.2    DIAGRAMME DE CLASSE.....	24
2.3    LA DONNEE EST LA SUIVANTE MODELE LOGIQUE DE LA BASE DE DONNEES .....	26
<b>CHAPITRE III: REALISATION .....</b>	<b>29</b>
3.1    . INTRODUCTION.....	30
3.2    ENVIRONNEMENT DE TRAVAIL .....	30
3.2.1    Environnement matériel.....	30
3.2.2    Environnement logiciel.....	30
3.2.2.1    Choix des technologies de développement.....	30
3.2.2.2    Les Outils de développement .....	32
3.3    ETAPES DE REALISATION .....	33
3.3.1    L'architecture de l'application Web.....	33
3.4    . LES INTERFACES DE L'APPLICATION.....	34
<b>CONCLUSION.....</b>	<b>40</b>

<b>BIBLIOGRAPHIE .....</b>	<b>41</b>
<b>ANNEXE A .....</b>	<b>42</b>

## Tables des figures

Figure 1. 1 : Diagramme de Cas d'utilisation Général.....	11
Figure 1. 2: Diagramme de Cas d'utilisation Pour l'Authentification .....	12
Figure 1. 3: Diagramme de Cas d'utilisation Pour l'Administrateur .....	13
Figure 1. 4: Diagramme de Cas d'utilisation : Directeur de département.....	15
Figure 1. 5 : Diagramme de Cas d'utilisation Pour Enseignant .....	16
Figure 1. 6 :Diagramme de Cas d'utilisation Pour Etudiant .....	17
Figure 1. 7: Diagramme de séquence « Authentification ».....	18
Figure 1. 8 : Diagramme de séquence « Authentification ».....	18
Figure 1. 9: Diagramme de séquence « Modifier un département » .....	19
Figure 1. 10: Diagramme de séquence : « Supprimer un département » .....	19
Figure 1. 11 : Diagramme de séquence « Affectation en tant que Vacant».....	20
Figure 1. 12 : Diagramme de séquence : « Affectation Charge enseignant».....	21
Figure 1. 13 : Diagramme de séquence : « Consultation Emploi».....	21
Figure 2. 1 : Le modèle Conceptuel de Données .....	23
Figure 2. 2 : Diagramme de classe .....	25
Figure 3. 1 : Architecture d'une application Web .....	34
Figure 3. 2 : Page d'accueil.....	34
Figure 3. 3 : Authentification .....	35
Figure 3. 4 :Saisie d'une filière .....	35
Figure 3. 5 : Suppression d'une filière .....	36
Figure 3. 6: Interface saisie Salle .....	36
Figure 3. 7 : Interface saisie Type d'enseignement.....	37
Figure 3. 8 : Interface saisie grade .....	37
Figure 3. 9 : Interface d'ajout d'un enseignant .....	38

Figure 3. 10 : Interface d'ajout d'une matière.....	38
Figure 3. 11 : Interface de consultation des matières affectées.....	39



# **Introduction Générale**

## **Cadre général**

La gestion des emplois du temps est l'organisation des enseignements travaillant au sein d'un établissement. Il s'agit d'une gestion complexe vu les contraintes multiples et les paramètres dont il faut tenir compte. Trouver une salle libre au jour et à l'horaire où l'enseignant est disponible et où le groupe des étudiants ne suit pas un autre cours n'est pas une tâche aisée.

Cette tâche se complique davantage lorsqu'il s'agit d'un établissement dépassant ses capacités du point de vue effectif des étudiants et enseignants. Dans ce cas, le travail manuel de préparation de l'emploi du temps devient quasi-impossible et les responsables se trouvent obligés de faire fonctionner leurs établissements jusqu'aux heures tardives et aussi, à répartir les emplois du temps des enseignants sur plusieurs journées, ce qui les empêche de se concentrer sur leurs activités de recherche ou bien même de ramener des étudiants pour ne suivre qu'une seule séance pendant toute une journée.

Les contraintes ne se limitent pas à ce que nous venons de citer, elles sont plus nombreuses et nous les découvrirons dans la suite du présent rapport.

## **Travail demandé**

L'administration de l'Institut Supérieur des Sciences Appliquées et de Technologie de Sousse qui sera présenté dans l'annexe (Voir annexe A) avec toute son ancienneté et l'expérience de ses responsables et agents, n'a pas échappé à ces difficultés, ce qui fait de cette gestion l'une de ses préoccupations majeures.

Dans ce cadre, l'administration de l'ISSATS nous a confié le travail du développement d'un système permettant de préparer et de gérer les emplois aussi bien des enseignants, que des étudiants et aussi avoir tous les états possibles concernant l'occupation des salles.

Ce système ne sera pas automatique dans le sens où il n'y a pas de gestion automatique des emplois du temps mais il permettra de voir plus clair et d'aider à prendre les bonnes décisions.

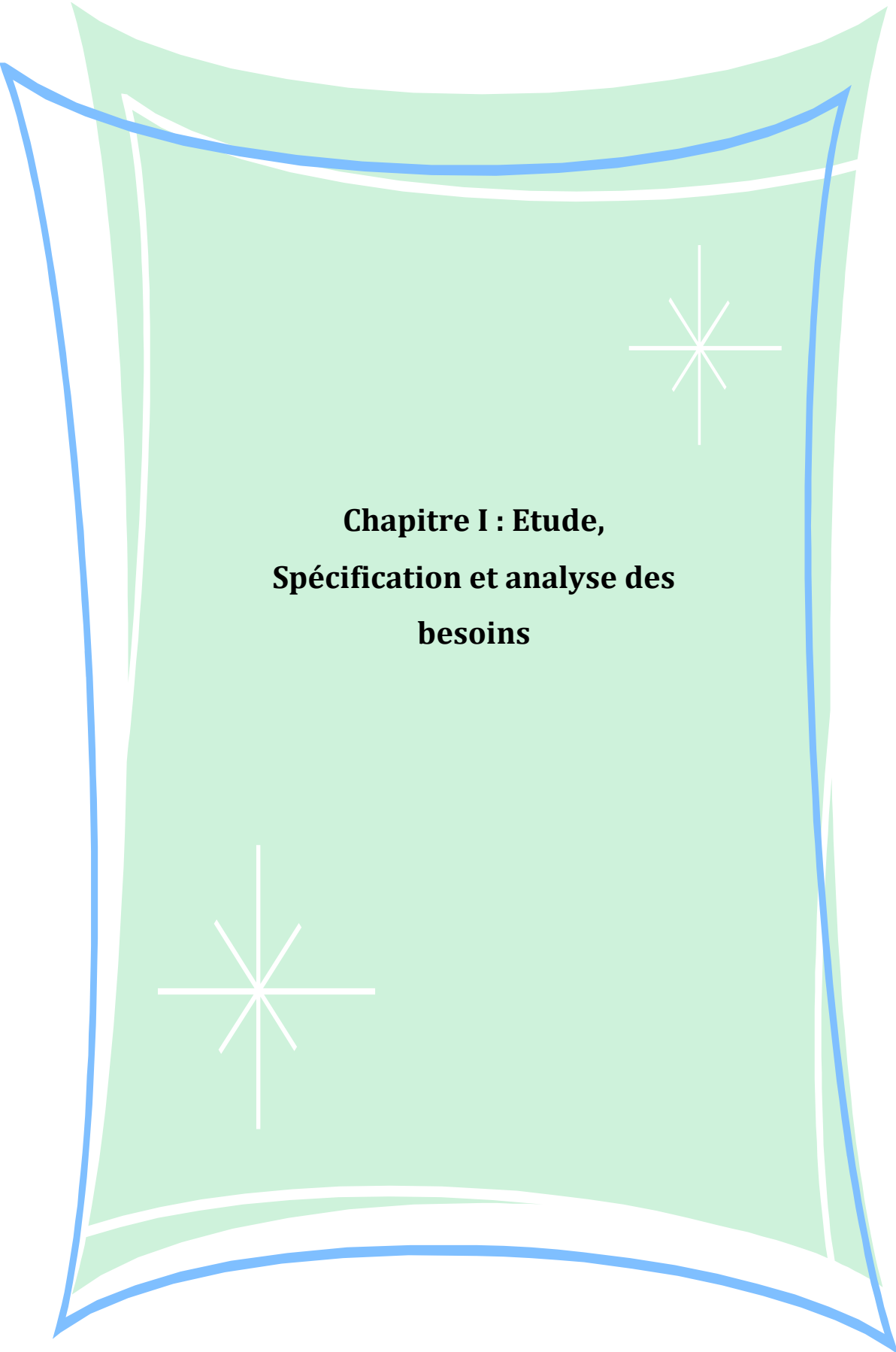
## **Plan du rapport**

La première partie de ce rapport présente le contexte de notre projet, le travail à réaliser et la solution que nous proposons pour réaliser ce travail.

La deuxième partie « Etude, Spécification et analyse des besoins » se focalise sur l'étude de l'existant, la présentation du projet à réaliser, ensuite, l'identification et l'analyse des besoins fonctionnels et non fonctionnels.

La troisième partie expose les différentes interfaces homme-machine développées ainsi que certains détails de l'implémentation. Les premières interfaces présentées sont parmi celles qui comportent des opérations d'ajout, de modifications et de suppression, les deuxièmes sont celles qui présentent une description des états possibles et des emplois prêts à être imprimés.

Nous clôturons ce rapport par une conclusion et les perspectives.



**Chapitre I : Etude,  
Spécification et analyse des  
besoins**

## **1.1. Introduction**

Nombreux sont les logiciels qui ont été conçus pour gérer les services d'enseignement tels que Salima, Fet. Malgré leur diversité, ces logiciels se basent sur les mêmes principes que ce soit au niveau des données, ou au niveau des traitements.

L'emploi du temps est l'un des exemples les plus fréquents des problèmes d'optimisation dans un établissement universitaire. Il peut se manifester sous des formes selon la spécificité de l'établissement.

Réaliser un emploi du temps pose un problème qu'on doit résoudre.

D'une manière générale, pour résoudre un problème donné, on commence par la recherche de la classe à laquelle appartient ce problème. Les problèmes relatifs aux emplois du temps augmentent en fonction de contraintes, ce qui rend difficile leur résolution. De tels problèmes peuvent être résolus par des approches classiques ou modernes.

Dans ce chapitre, nous décrivons le problème d'emploi du temps à l'ISSATS.

## **1.2. Etude de l'existant**

La tâche de planification des enseignements est confiée à des agents très anciens dans leur poste. Ils réalisent tout le travail manuellement et se chargent de toute la gestion de remplacement des séances non réalisées pour diverses raisons.

Grace à ses expériences, ils arrivent à tout faire et à satisfaire la plus part des vœux des enseignants mais, d'une part, cela lui nécessite un effort considérable.

L'emploi du temps est un plan représentatif qui nécessite :

- Un Enseignant qui va assurer un volume horaire par semaine qui est le D<sup>U</sup> convenant à son grade.
- Un cours va être assuré par un enseignant à un tel groupe dans un créneau horaire (séance) bien défini.
- Il consiste à gérer des emplois du temps tout en prenant compte des ressources humaines et matérielles disponibles (salles).

Elaborer un emploi du temps c'est établir le programme hebdomadaire pour tous les cours enseignés d'un ensemble d'enseignants et groupes en réduisant au minimum les chevauchements.

En pratique, certaines conditions s'imposent pour l'élaboration d'un emploi du temps. Le système doit spécifier l'affectation des enseignants, groupes et salles dans le temps tout en respectant le plus possible les contraintes imposées (chevauchements).

### **1.3. Critique de l'existant**

La critique de l'existant est une phase primordiale qui se fait après l'étude de l'existant. Cette étape a pour objectif la découverte et la précision des erreurs produites par l'utilisation manuelle fin d'apporter les solutions convenables.

A l'ISSATS, on effectue un emploi du temps en utilisant une base créée en Microsoft Access de façon purement manuelle. L'administrateur choisit les salles et les séances de cours, de TD ou de TP. Il se charge ensuite d'affecter les matières aux différents groupes qui existent à l'institut.

Au cours de l'affectation des emplois du temps, on peut rencontrer plusieurs difficultés dont la plus grande est la présence chevauchements que soit en termes de salles, groupes ou enseignants. Ces chevauchements sont autorisés par notre base.

Cette difficulté nous amène à traiter la gestion d'emplois du temps.

Au début de chaque semestre universitaire, l'administrateur définit les disponibilités possibles pour fixer un cours ; pour réaliser cela on devra vérifier la disponibilité des groupes dans une séance donnée et aussi faire une vérification de la disponibilité des enseignants et la salle selon son type.

Dans ce système manuel, l'administrateur peut risquer certaines erreurs tels que :

- L'affectation d'une salle à deux groupes le même jour et même séance.
- Lourdeur de la procédure de recherche d'une information telle que la disponibilité d'un enseignant, d'un groupe ou d'une salle.
- Impossibilité d'enseigner plus d'un cours /TD /TP en même temps par groupe.
- Affecter au même enseignant deux matières différentes à la même séance.

A la fin de chaque semaine, une nouvelle version d'emploi du temps format Excel de la semaine suivante sera affichée sur le site l'ISSATS

L'objectif de ce projet est d'analyser, de réaliser et de concevoir une application permettant la gestion des emplois du temps en essayant au maximum d'éviter les chevauchements possibles.

### **1.4. Spécification des Besoins**

#### **1.4.1. Introduction**

Une méthode de conception est une démarche générale reflétant une philosophie de présentation et de suivi du système. Elle propose des outils spécifiques permettant un suivi efficace de l'information relative au système. Et notre choix se porte sur le langage UML (Unified Modeling Language) qui facilite l'interactivité avec la base de données à l'aide des diagrammes de cas d'utilisation et des diagrammes de classes

L'UML (Unified Modeling Language) est un langage de modélisation orientée objet, elle est développée dans le but de définir la notion standard pour la modélisation des applications construites à l'aide des objets. Elle est utilisée pour spécifier un logiciel ou pour le concevoir, le modèle décrit les classes et les cas d'utilisation vus de l'utilisateur final du logiciel.

Le modèle produit par une conception orientée objet est en général une extension du modèle issu de la spécification, il l'enrichit de classe dites techniques qui n'intéressent pas l'utilisateur final du logiciel mais seulement ses concepteurs.

## 1.4.2. Spécification des besoins

### 1.3.2.1 Les acteurs

Un acteur est une entité externe qui interagit avec le système (opérateur, centre distant, autre système...). En réponse à l'action d'un acteur, le système fournit un service qui correspond à son besoin. Les acteurs peuvent être classés (hiérarchie).

➤ **L'administrateur** : c'est la personne primordiale dans le système de gestion des emplois du temps. Il permet de manipuler toutes les tâches proposées et possibles pour faire un emploi du temps ; tels que Ajout, Modifier, Supprimer et consulter.

- ✓ Ajout enseignant,
- ✓ Ajout filière
- ✓ Ajout niveau
- ✓ Ajout matière
- ✓ Ajout salle
- ✓ Changement d'emploi d'enseignant et/ou de groupe,....

➤ **L'enseignant** : Il s'agit du profil des enseignants de l'ISSATS qu'ils soient permanents ou vacataires.

Un enseignant peut :

- ✓ accéder au site après s'être identifié.
- ✓ Consulter son emploi du temps.

➤ **L'Etudiant**: toute personne qui suit une formation au sein de l'ISSATS : Chaque étudiant peut :

- ✓ Consulter son emploi après identification par groupe.

➤ **Le directeur de département** :

- ✓ accéder au site après avoir s'identifier.
- ✓ Affecter une matière à un enseignant en désignant son type (Cours /TD / TP)
- ✓ Consulter son emploi du temps.

### 1.4.3. Les besoins fonctionnels

- ✓ La première étape consiste à introduire toutes les informations nécessaires tels que la liste des filières, des départements, Matières, groupes, niveaux d'études, les salles avec leurs types (C/ TD /TP), les séances avec la description de l'heure du début et l'heure de fin.
- ✓ Chaque Enseignant est structuré en un seul département, qui regroupe chacun des enseignants spécifiques. Parmi ces enseignants, l'un d'eux est le directeur du département.
- ✓ Un enseignant est définit par son nom, prénom, département et grade.
- ✓ Chaque enseignant assure une ou plusieurs matières.
- ✓ Les étudiants quant à eux suivent plusieurs matières chaque semestre selon leur filière et leur niveau d'étude tout en respectant le plan d'étude.
- ✓ Une matière peut être enseignée par un ou plusieurs enseignants,
- ✓ Affecter toutes les matières aux groupes sans affecter les noms des enseignants : L'administrateur se charge d'affecter tous les matières à leurs groupes et aussi les salles avec Vacant.
- ✓ Enfin, on doit pouvoir imprimer toutes les emplois du temps des enseignants et des étudiants (Groupes).

La figure suivante illustre le diagramme des cas d'utilisation générale de notre application.

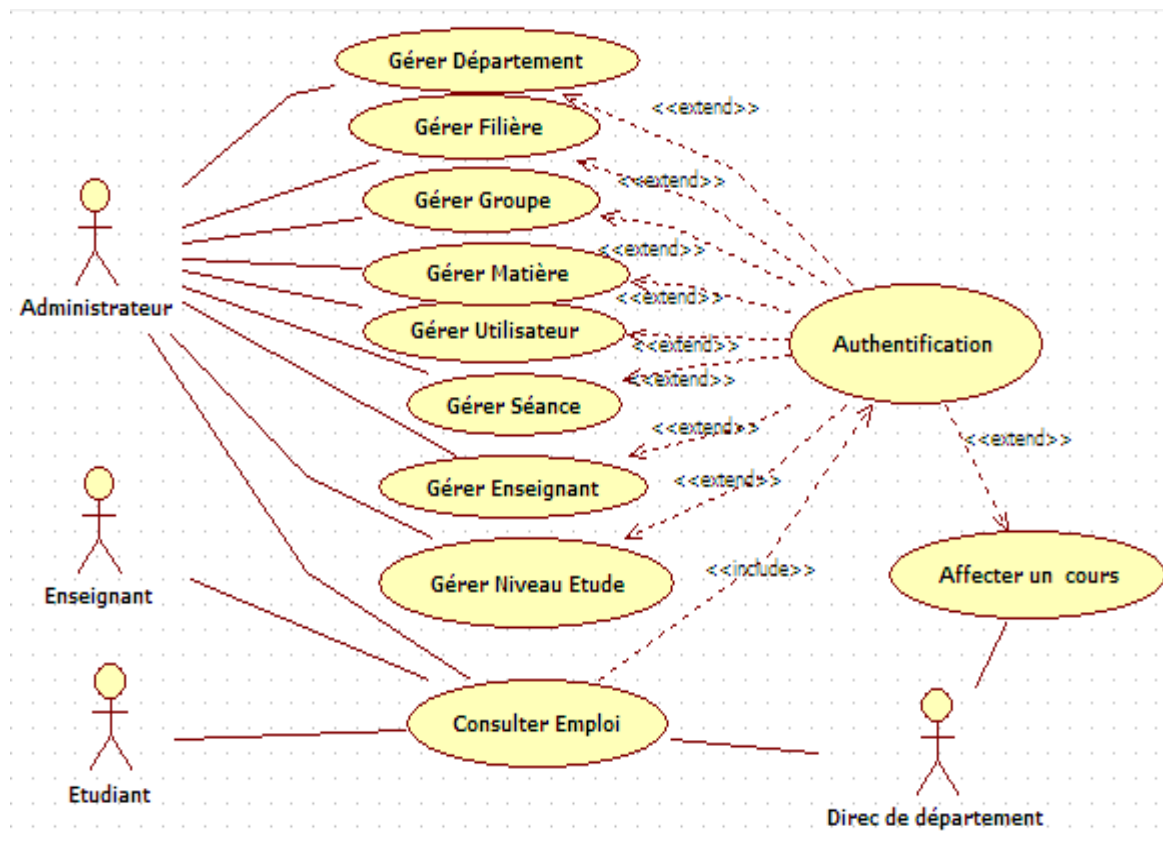


Figure 1. 1 : Diagramme de Cas d'utilisation Général

On entend dire par gérer les quartes actions principales sur un objet qui sont : ajouter, supprimer, modifier, consulter. Dans la conception textuelle et l'analyse des différents cas d'utilisation, on s'intéresse à l'action d'ajout.

#### 1.4.4. Les diagrammes de cas d'utilisation

Le diagramme de cas d'utilisation décrit la succession des opérations réalisées par un acteur. C'est le diagramme principal du modèle UML, celui qui assure la relation entre l'utilisateur et les objets que le système met en œuvre.

La description de l'interaction est réalisée suivant le point de vue de l'utilisateur, et les cas d'utilisation permettent de recueillir et de décrire les besoins des acteurs aux systèmes, il permet aussi de faciliter la structuration des besoins des utilisateurs et d'exprimer les limites et les objectifs du système.



Figure 1. 2: Diagramme de Cas d'utilisation Pour l'Authentification

#### Scénario du cas d'utilisation «Authentification» :

Scénario nominal	
Action de l'acteur	Réactions du système
1. L'utilisateur demande une connexion au système	2. Le Système affiche l'interface de connexion
3. L'administrateur saisit son login et son mot de passe	4. Le système vérifie si l'utilisateur existe dans sa base
5. Le système se connecte et ouvre la session	
Dans le cas où l'utilisateur saisit son login ou mot de passe erroné, le système demande de les retaper de nouveau.	



### 1.4.4.1. Diagramme de Cas d'utilisation : L'administrateur

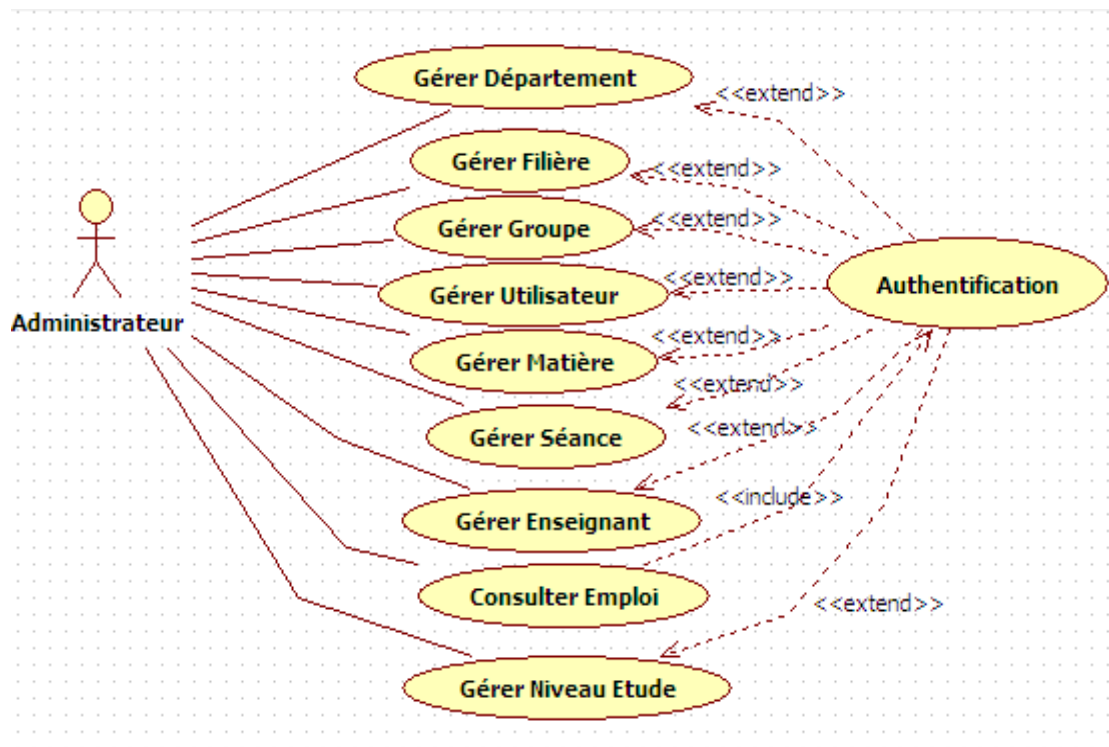


Figure 1. 3: Diagramme de Cas d'utilisation Pour l'Administrateur

#### Scénario du cas d'utilisation « Ajout »

Scénario nominal	
Action de l'acteur	Réactions du système
1. L'administrateur demande l'ajout d'un nouvel utilisateur	2. Le système affiche le formulaire du nouveau nom de l'utilisateur
3. L'administrateur remplit le formulaire du nouvel utilisateur et valide l'ajout	4. Le système vérifie et sauvegarde le nouveau nom de l'utilisateur.
<p>Dans le cas où l'administrateur entre un nom utilisateur existant : Le système demande de vérifier le nom du nouvel utilisateur.</p> <p>Toutes ces étapes seront appliquées avec les mêmes enchainements pour tous les scénarios d'ajout (Ajout département, Groupe, Filière, niveau étude....)</p>	

**Scénario du cas d'utilisation «Modifier» :**

<b>Scénario nominal</b>	
<b>Action de l'acteur</b>	<b>Réactions du système</b>
1. L'administrateur demande la modification d'un utilisateur  3. L'administrateur modifie les informations de l'utilisateur et valide la modification.	2. Le système affiche le formulaire de l'utilisateur  4. Le système sauvegarde les informations
Toutes ces étapes seront appliquées avec les mêmes enchainements pour tous les scénarios de modification (Département, Groupe, Filière,...)	

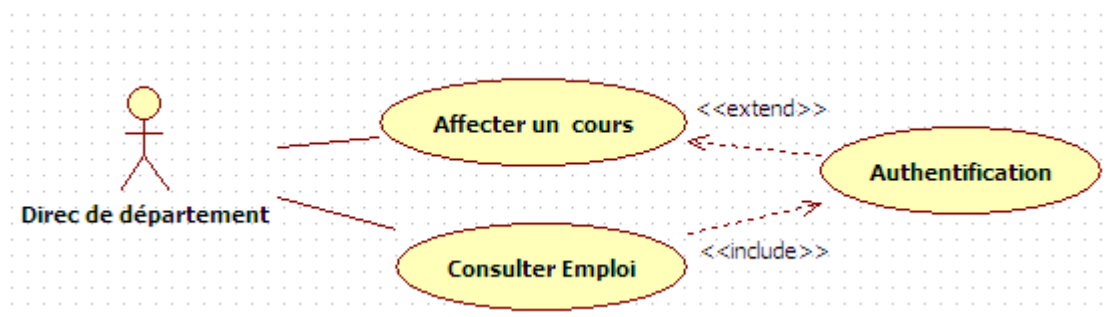
**Scénario du cas d'utilisation «Supprimer» :**

<b>Scénario nominal</b>	
<b>Action de l'acteur</b>	<b>Réactions du système</b>
1. L'administrateur demande la suppression d'un utilisateur  3. L'administrateur sélectionne l'utilisateur à supprimer et demande la suppression.	2. Le système affiche le formulaire de l'utilisateur  4. Le système supprime l'utilisateur. 5. Le système affiche la validation de Suppression
Toutes ces étapes seront appliquées avec les mêmes enchainements pour tous les scénarios de modification (Département, Groupe, Filière,...)	

**Scénario du cas d'utilisation «Affecter emploi en tant que Vacant» :**

<b>Scénario nominal</b>	
<b>Action de l'acteur</b>	<b>Réactions du système</b>
1. L'administrateur demande l'affectation d'une matière.	2. Le Système affiche le formulaire de saisie.  3. L'administrateur choisit le groupe, le type d'enseignement,  4. Le système affiche la liste des séances libres  5. L'administrateur choisit la séance  6. Le système affiche la liste des salles libres  7. L'administrateur choisit la salle selon le type d'enseignement et affecte le nom de l'enseignant « Vacant »  8. L'administrateur Valide  9. Le système sauvegarde.
3. L'administrateur choisit le groupe, le type d'enseignement,	
5. L'administrateur choisit la séance	
7. L'administrateur choisit la salle selon le type d'enseignement et affecte le nom de l'enseignant « Vacant »	
8. L'administrateur Valide	

**1.4.4.2. Diagramme de Cas d'utilisation : Directeur de département**

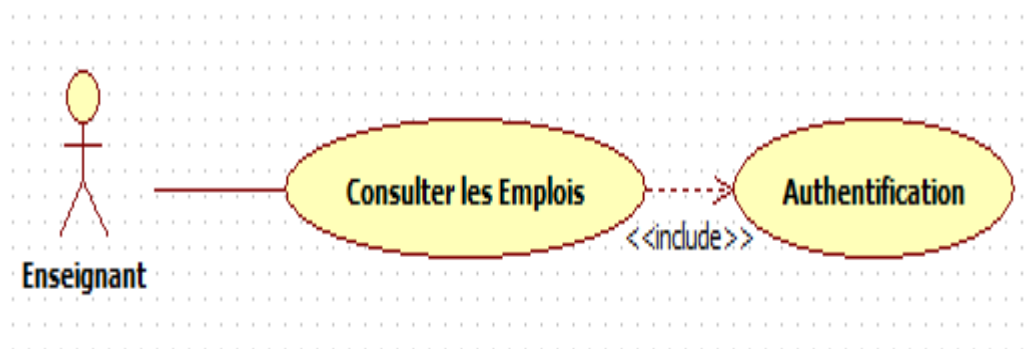


**Figure 1. 4: Diagramme de Cas d'utilisation : Directeur de département**

**Scénario du cas d'utilisation «Affecter matière à un enseignant» :**

<b>Scénario nominal</b>	
<b>Action de l'acteur</b>	<b>Réactions du système</b>
<ol style="list-style-type: none"> <li>1. Le directeur de département demande l'affectation d'une matière.</li>   <li>3. Le directeur choisit le groupe, le type d'enseignement,</li>   <li>5. Le directeur sélectionne le nom de l'enseignant et valide la saisie</li> </ol>	<ol style="list-style-type: none"> <li>2. Le Système affiche le formulaire d'affectation</li>   <li>4. Le système affiche la liste des enseignants accordé au département</li>   <li>6. Le système sauvegarde.</li> </ol>
<p>Cette affectation est faite seulement par les directeurs des départements.</p>	

**1.4.4.3. Diagramme de Cas d'utilisation : Cas Enseignant**



**Figure 1. 5 : Diagramme de Cas d'utilisation Pour Enseignant**

### Scénario du cas d'utilisation «Enseignant »

Scénario nominal	
Action de l'acteur	Réactions du système
1. L'enseignant demande la consultation de son emploi tout en saisissant le login et le mot de passe pour accéder à son espace.  L'enseignant consulte et/ou imprime	2. Le système vérifie le login et le mot de passe saisis par l'enseignant et lui affiche l'emploi du temps si la connexion est effectuée

#### 1.4.4.4. Diagramme de Cas d'utilisation : Etudiant :

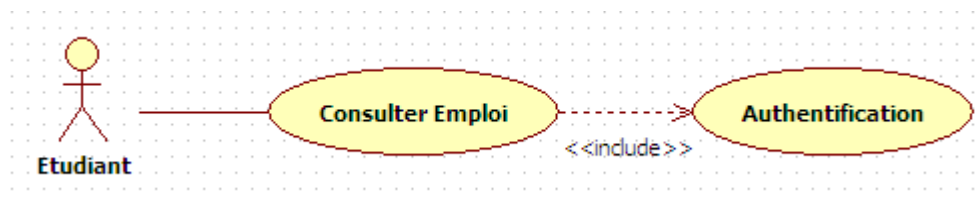


Figure 1. 6 :Diagramme de Cas d'utilisation Pour Etudiant

### Scénario du cas d'utilisation «Etudiant» :

Scénario nominal	
Action de l'acteur	Réactions du système
1. L'étudiant demande la consultation de son emploi tout en saisissant le login et le mot de passe du groupe pour accéder à son espace.  3. L'étudiant consulte et/ou imprime	2. Le système vérifie le login et le mot de passe saisis par l'étudiant et lui affiche l'emploi du temps si la connexion est effectuée.

## 1.4 Analyse de la Vue dynamique

Le diagramme de séquence UML est un diagramme qui permet de représenter les interactions entre les objets, suite à un événement externe en précisant la chronologie des échanges de messages.

Pour analyser les comportements des cas d'utilisations, nous utilisons les diagrammes de séquences

### 1.4.1 Diagramme de séquence : « Authentification »

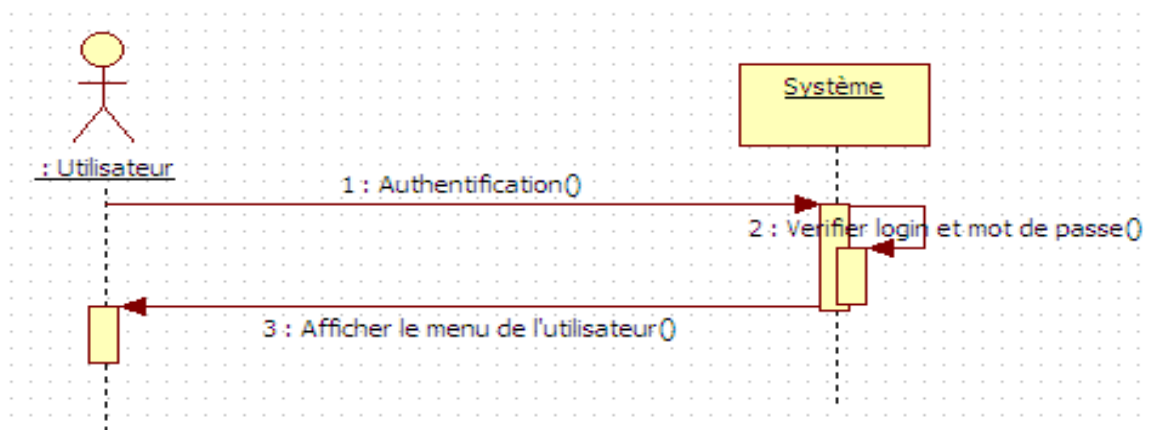


Figure 1. 7: Diagramme de séquence « Authentification »

### 1.4.2 Diagramme de Séquence : « Ajout Département » :

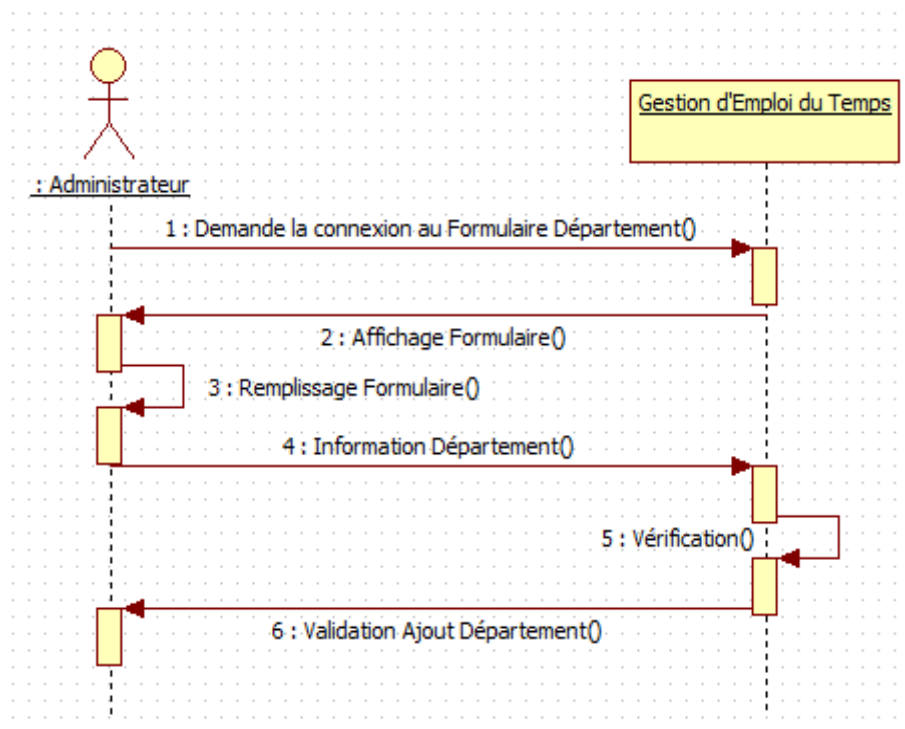


Figure 1. 8 : Diagramme de séquence « Authentification »

### 1.4.3 Diagramme de séquence « Modifier un département »

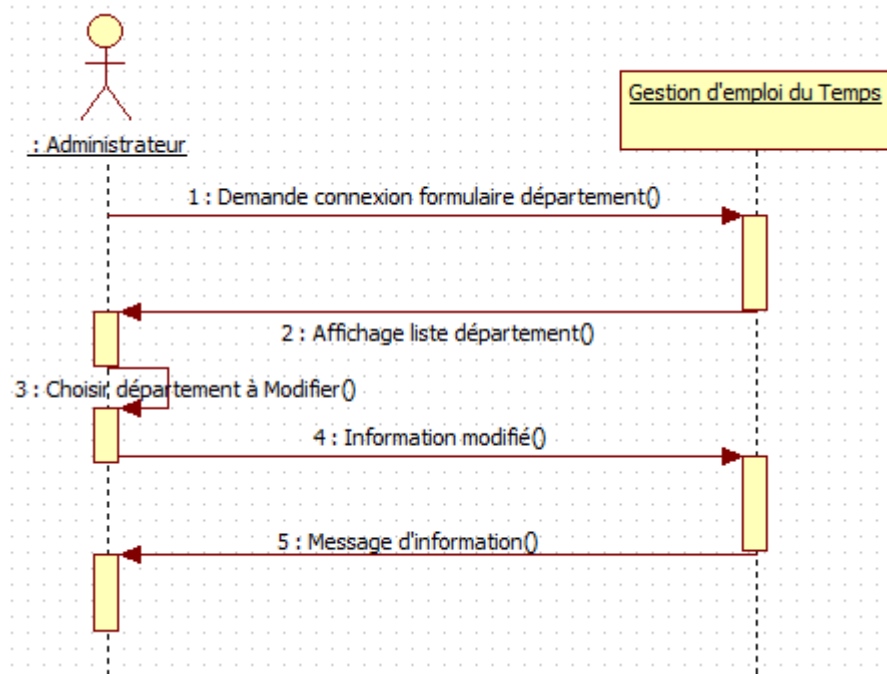


Figure 1. 9: Diagramme de séquence « Modifier un département »

### 1.4.4 Diagramme de séquence : « Supprimer un département »

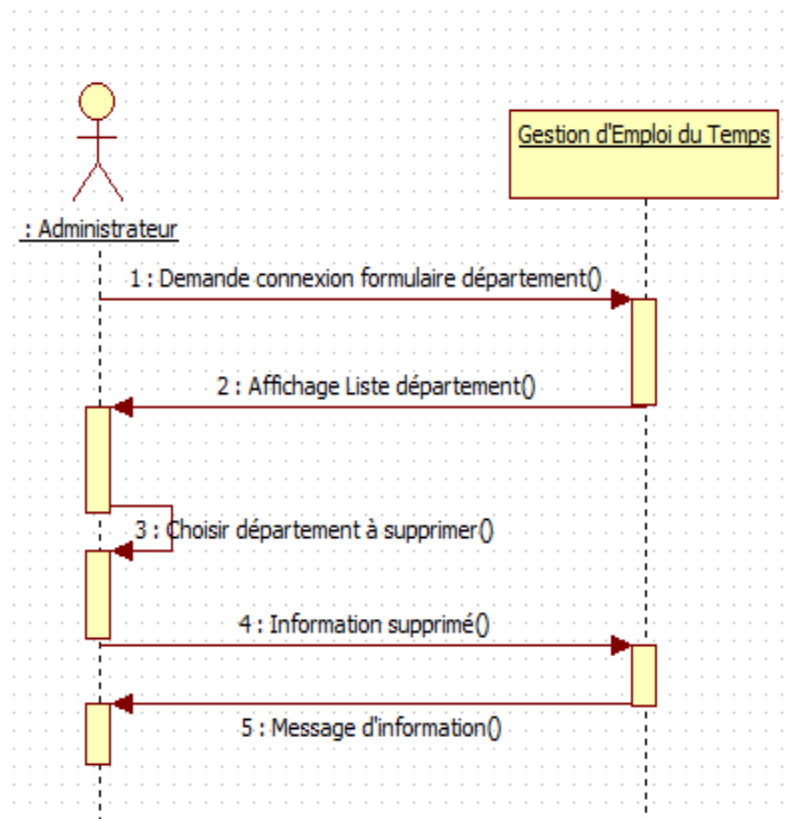


Figure 1. 10: Diagramme de séquence : « Supprimer un département »

### 1.4.5 Diagramme de séquence « Affectation en tant que Vacant » :

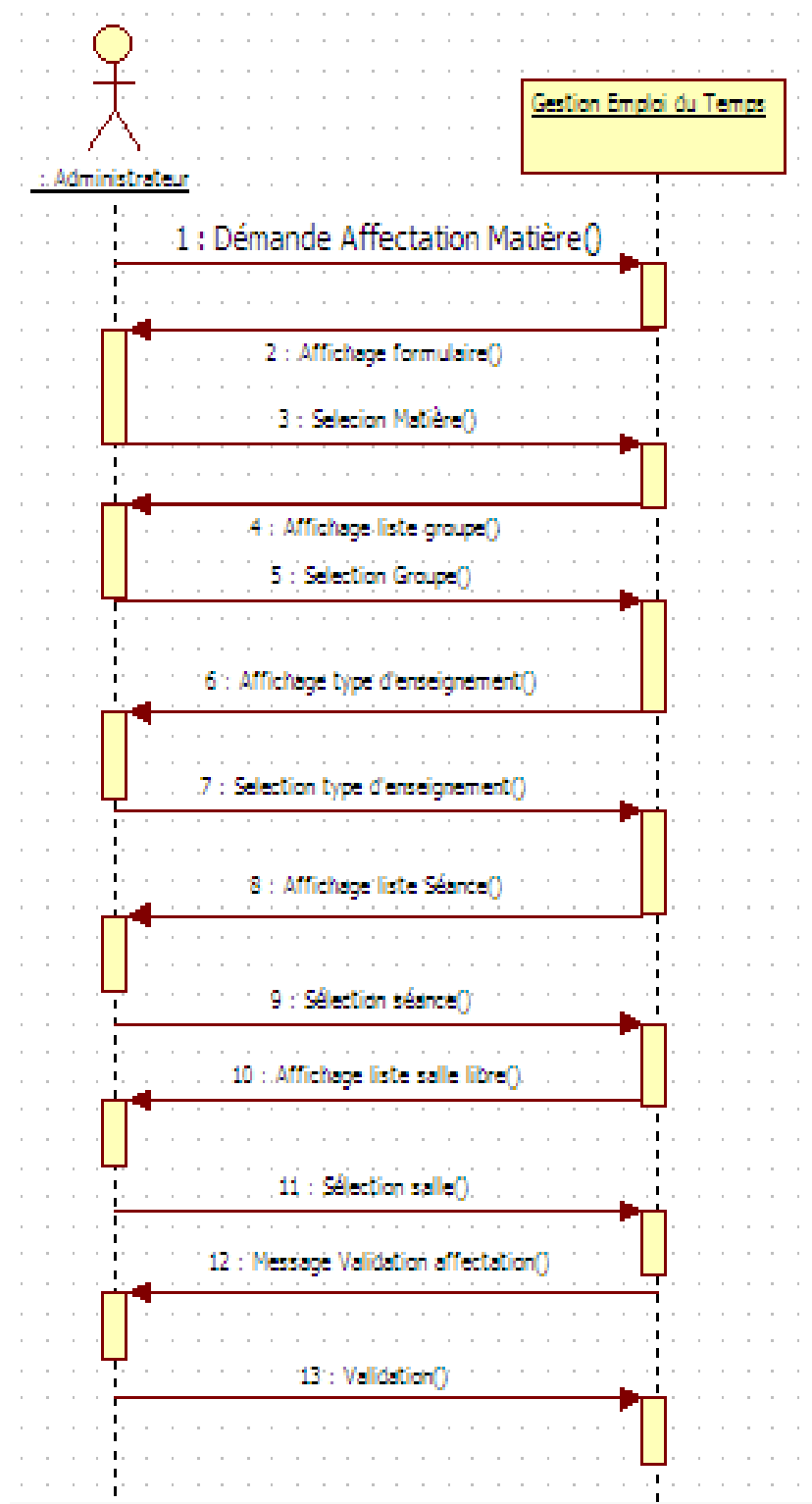


Figure 1. 11 : Diagramme de séquence « Affectation en tant que Vacant»



### 1.4.6 Diagramme de séquence : « Affectation Charge enseignant » :

Ce diagramme est effectué par les directeurs de départements

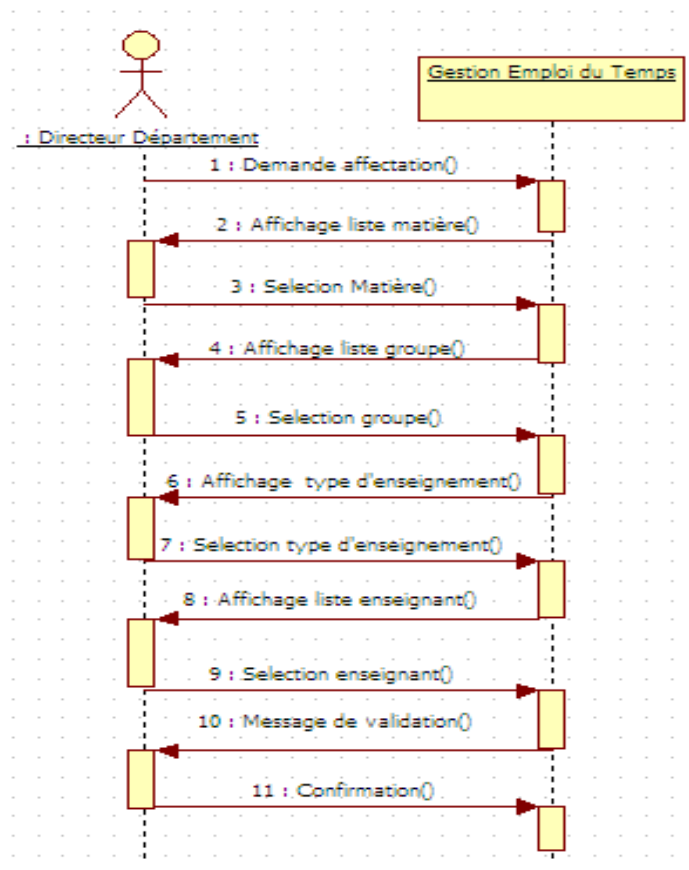


Figure 1. 12 : Diagramme de séquence : « Affectation Charge enseignant»

### 1.4.7 Diagramme de séquence : « Consultation Emploi » :

Ce diagramme est effectué par tous les enseignants et les étudiants après s’être s’identifier par un login et un mot de passe.

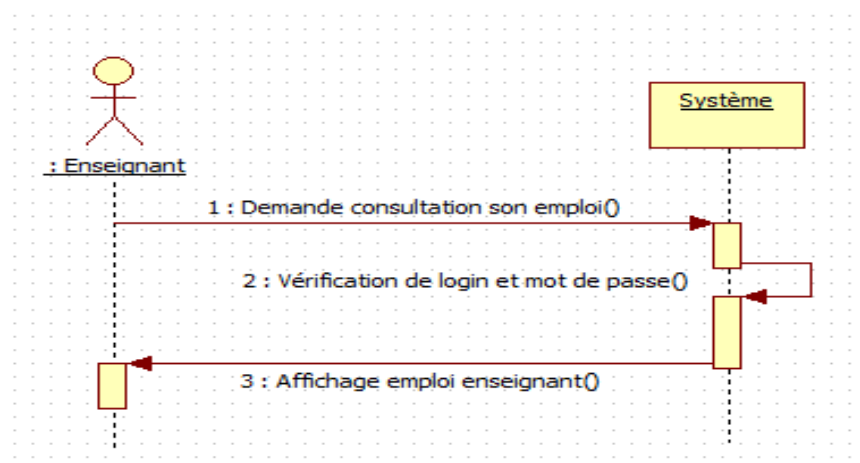


Figure 1. 13 : Diagramme de séquence : « Consultation Emploi»



**Chapitre II: Conception**

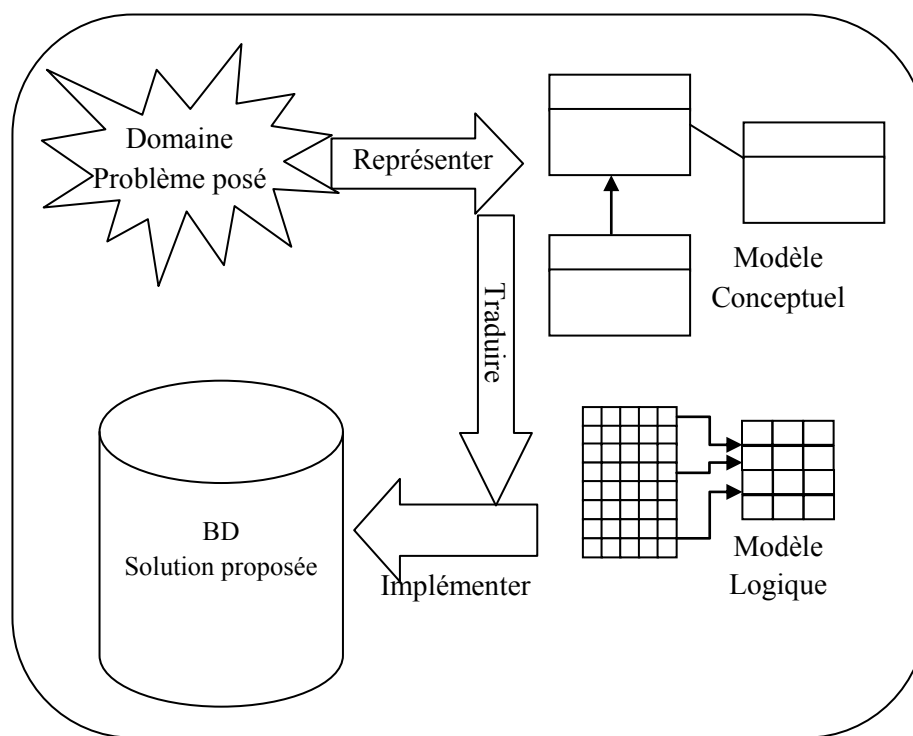
## 2.1 Introduction

La conception des bases de données est la tâche la plus ardue du processus de développement du système d'information

Recourir à une méthode de conception afin de faciliter la communication et la coopération entre les différents acteurs d'une application.

La conception d'une telle base de données consiste à suivre quatre étapes :

- ✓ Analyse de la situation existante et des besoins :  
Cette première étape repose sur l'analyse de l'existant et des besoins, elle est très délicate et fondamentale dans le processus de conception
- ✓ Création d'une série de modèles conceptuels (canonique et vues externes) qui permettent de représenter tous les aspects importants du problème
- ✓ Traduction des modèles conceptuels en modèle logique et optimisation (normalisation) de ce modèle logique
- ✓ Implémentation d'une base de données dans un SGBD, à partir du modèle logique



**Figure 2. 1 : Le modèle Conceptuel de Données**

Le modèle Conceptuel de Données (MCD) ou modèle Entité/Association est un modèle chargé de représenter, sous forme graphique, les informations manipulées par le système. Le MCD permet de décrire les données gérées sans tenir compte des choix d'organisation ou techniques.

Le MCD a pour objectif d'identifier, décrire et modéliser les entités et leurs associations à l'aide d'une présentation graphique

Certaines contraintes ne sont pas représentables par le seul formalisme de base (entité, association, propriétés, cardinalités) mais correspond à une règle que doit satisfaire le modèle pour être fidèle et cohérent avec l'activité à représenter.

Par exemple : Pour une telle séance d'emploi du temps, un enseignant ne fait un cours que dans une seule salle.

## 2.2 Diagramme de classe

Le diagramme de classe représente la description statique du système à développer en intégrant dans chaque classe la partie dédiée aux données et celle consacrée au traitement.

C'est un diagramme pivot de l'ensemble de modélisation d'un système, cette représentation est concentrée sur le concept de classe et d'associations, les traitements sont matérialisés par des opérations.

Une classe est une description abstraite d'un ensemble d'objet ayant des propriétés similaires, un comportement commun et des relations communes avec d'autres objets.

### Règle de gestion :

- Un enseignant peut enseigner une ou plusieurs matières à un ou plusieurs groupes.
- Un enseignant relève d'un seul département
- Un enseignant possède un seul grade et peut posséder une fonction (Directeur établissement, directeur de département).
- A un jour très bien déterminé et une séance précise on peut trouver au maximum un seul enseignant dans une salle.
- Un groupe peut être enseigné par un ou plusieurs enseignants.
- Un enseignant peut enseigner un ou plusieurs groupes.
- Un enseignant peut enseigner une ou plusieurs matières.
- Un groupe est connu par la filière et le niveau et la spécialité.

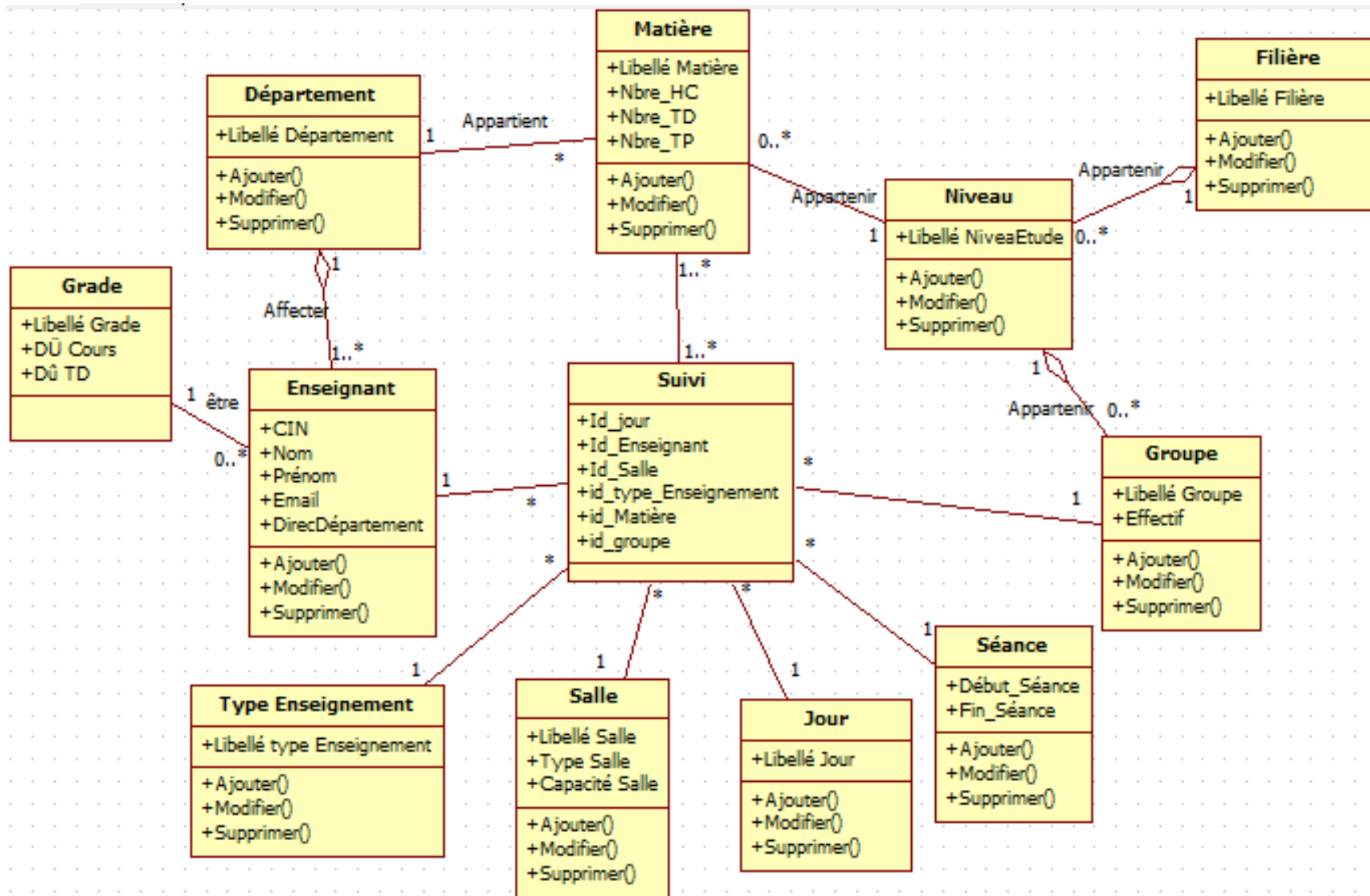


Figure 2. 2 : Diagramme de classe

## 2.3 La donnée est la suivante modèle logique de la base de données

- a) Table Département: (Id\_Département, Libellé Département)

Propriété		Type
Attribut	Désignation	
Id_Département	Identité de Département	Integer
Libellé Département	Nom de Département	String

- b) Table Filière : (Id\_Filière, Libellé Filière)

Propriété		Type
Attribut	Désignation	
Id_Filière	Identité de Filière	Integer
Libellé Filière	Nom de Filière	String

- c) Table Grade: (Id\_Grade, Libellé Grade, DÛ-C, DU-TD)

Propriété		Type
Attribut	Désignation	
Id_Grade	Identité de grade	Integer
Libellé Grade	Nom de grade	String
DÛ-C	Volume DÛ Cours	Float
DU-TD	Volume DÛ TD	Float

- d) Table enseignant : (Id\_Enseignant, Nom, Prénom, Id\_Grade, Id\_Département).

Propriété		Type
Attribut	Désignation	
Id_Enseignant	Identité de l'enseignant	Integer
Nom	Nom de l'enseignant	String
Prénom	Prénom de l'enseignant	String
Id_Grade	Identité de grade	Integer
Id_Département	Identité de département	Integer
DirécDépartement	Si L'enseignant est un directeur de département	Booléen

- e) Table Niveau : (Id\_Niveau, Libellé Niveau)

Propriété		Type
Attribut	Désignation	
Id_Niveau	Identité Niveau	Integer
Libellé Niveau	Nom de Niveau	String

f) Table Matière: (Id\_Matière, Libellé Matière, Nbre\_HC, Nbre\_HTD, Nbre\_HTP, Id\_Niveau, Id\_Departement)

Propriété		Type
Attribut	Désignation	
Id_Matière	Identité Matière	Integer
Libellé Matière	Nom de la matière	String
Nbre_HC	Volume horaire hebdomadaire cours	Float
Nbre_HTD	Volume horaire hebdomadaire TD	Float
Nbre_HTP	Volume horaire hebdomadaire TP	Float
Id_Niveau	Identité Niveau	Integer
Id_Departement	Identité Département	Integer

g) Table Groupe : (Id\_Groupe, Libellé Groupe, Id\_Niveau, Id\_Filière)

Propriété		Type
Attribut	Désignation	
Id_Groupe	Identité Groupe	Integer
Libellé Groupe	Nom de Groupe	String
Id_Niveau	Identité Niveau	Integer
Id_Filière	Identité de Filière	Integer

h) Table Salle : (Id\_Salle, Libellé Salle, Type Salle, Capacité)

Propriété		Type
Attribut	Désignation	
Id_Salle	Identité Salle	Integer
Libellé Salle	Nom de Salle	String
Type Salle	Type de Salle (C/TD/TP)	String
Capacité	Capacité de Salle	Integer

i) Table Jour : (Id\_Jour, Jour)

Id_Jour	Identité Jour	Integer
Jour	Jour de séance	String

j) Table TypeEnseignement : (Id-TypeEnseignement, Libellé TypeEnseignement)

Propriété		Type
Attribut	Désignation	
Id-TypeEnseignement	Identité de Type Enseignement	Integer
Libellé TypeEnseignement	Nom de Type Enseignement	String

k) Table Séance : (Id\_Séance, Libellé Séance, Heure\_Début, Heure\_Fin)

Propriété		Type
Attribut	Désignation	
Id_Séance	Identité Séance	Integer
Libellé Séance	Nom de Séance	String
Heure_Début	Heure Début Séance	String
Heure_Fin	Heure Début Séance	String

l) Table Suivi : (Id\_Enseignant, Id\_Groupe, Id\_Séance, Id\_Salle, Id-TypeEnseignement, Id\_Matière)

Propriété		Type
Attribut	Désignation	
<u>Id_Enseignant</u>	Identité de l'enseignant	Integer
Id_Groupe	Identité Groupe	Integer
<u>Id_Séance</u>	Identité Séance	Integer
<u>Id_Salle</u>	Identité Salle	Integer
Id-TypeEnseignement	Identité de Type Enseignement	Integer
Id_Matière	Identité Matière	Integer





## **Chapitre III: Réalisation**

## 3.1 . Introduction

Ce dernier chapitre présente la partie de la réalisation et la mise en œuvre des différents composants décrits au niveau du chapitre précédent. Dans un premier temps, on présente l'environnement matériel et logiciel. Ensuite, on décrit le travail réalisé en détaillant quelques captures d'écrans des fonctionnalités réalisées.

## 3.2 Environnement de travail

### 3.2.1 Environnement matériel

Pour développer cette application, on a un microordinateur avec un microprocesseur AMD E1-2100 APU with Radeon™ HD Graphics 1.00 GHz, Une mémoire virtuelle vive de 2.00G et un disque dur de 500Go

### 3.2.2 Environnement logiciel

#### 3.2.2.1 Choix des technologies de développement

##### ❖ SpringMVC :

SpringMVC est un Framework de développement java basé sur la notion de conteneur léger.

Spring est un projet open source de licence Apache dont le support et les évolutions sont réalisées par la société Interface21. Spring est composé de très nombreuses briques, mais l'on peut utiliser uniquement les parties qui nous intéressent sans se soucier des autres.

SpringMVC est un Framework de présentation, pour l'application WEB, suivant le modèle MVC, fondé sur le conteneur léger de SPRING. Le cœur de l'environnement Spring est un « conteneur léger ». Un conteneur léger sert à contenir un ensemble d'objets instanciés et initialisés, formant un contexte initial (ou une hiérarchie de contextes) pour une application.

Ce contexte initial est souvent construit à partir d'une description externe(XML) décrivant les objets à créer, les valeurs initiales et les dépendances entre objets. Les dépendances entre objets sont automatiquement créées à partir de la description (on parle d'injections de dépendances) et non par les objets eux-mêmes par programmation.

Dans le cas de SpringMVC le conteneur va servir à créer :

- Le contexte de l'application web
- Les objets traitant les requêtes (Controller)
- Les objets créant les pages HTML (View)
- Les objets donnés des formulaires (Command).
- Les liens avec les couches métiers et BD.

- La mapping des URL vers les contrôleurs
- Le mapping des vues, etc.

La vision de SpringMVC est le point d'entrée générique qui délègue les requêtes à des Controller. Ce dernier prend en charge une requête et utilise la couche métier pour y répondre. Il fabrique un modèle sous la forme d'une Map contenant les éléments de la réponse et choisit une View qui sera paramétrée par le Map pour donner la page qui sera affichée.

### ❖ **Spring IOC**

L'IOC est un Design-Pattern. Ce n'est pas une lubie, mais un enjeu primordial pour un développement propre et structuré

### ❖ **Hibernate**

Hibernate est un logiciel, écrit en java, qui permet de faire le mapping entre Objet java et Objets stockés en base relationnelle. Il en assure aussi la persistance. Il a besoin de plusieurs éléments pour fonctionner :

- Une classe de type javabean qui encapsule les données d'une occurrence d'une table.
- Un fichier de correspondance qui configure la correspondance entre la classe et la table des propriétés de configuration notamment des informations concernant la connexion à la base de données.
- Une fois ces éléments sont correctement définis, il est possible d'utiliser Hibernate dans le code de traitements à réaliser.

### ❖ **JPA : Java Persistence API**

La Java Persistence API (abrégée en JPA), est une interface de programmation Java permettant aux développeurs d'organiser des données relationnelles dans des applications utilisant la plateforme Java

Une entité JPA est, par définition, une classe Java qui doit avoir les propriétés suivantes :

Elle doit posséder un constructeur vide, public ou protected. Rappelons que ce constructeur vide existe par défaut si aucun constructeur n'existe dans la classe. Dans le cas contraire, il doit être ajouté explicitement.

Elle ne doit pas être finale, et aucune de ces méthodes ne peut être finale.

Une entité JPA ne peut pas être une interface ou une énumération.

Une entité JPA peut être une classe concrète ou abstraite.

Une classe possède des champs. Ces champs sont en général associés à des valeurs en base, rangées dans les colonnes d'une table. L'objet EntityManager est capable de positionner lui-même les valeurs de ces champs. Il le fait par injection, soit directement sur le champ, soit en passant par les getters / setters. Ce point n'est pas spécifié de façon très satisfaisante en JPA 1.0, et a été revu en JPA 2.0.

Il est possible de définir des entités JPA sur des classes qui héritent les unes des autres. Sur toute une hiérarchie de classes, on doit avoir une unique définition de clé primaire.

❖ **JSTL : Java server page Standard Tag Library :**

JSTL est l'acronyme de Java server page Standard Tag Library. C'est un ensemble de tags personnalisés développé sous la JSR 052 qui propose des fonctionnalités souvent rencontrées dans les JSP :

- Tag de structure (itération, conditionnement ...)
- Internationalisation
- Exécution de requêtes SQL
- Utilisation de documents XML

JSTL nécessite un conteneur d'applications web qui implémente l'API servlet 2.3 et l'API JSP 1.2.

L'implémentation de référence (JSTL-RI) de cette spécification est développée par le projet Taglibs du groupe Apache sous le nom " Standard ".

### 3.2.2.2 Les Outils de développement

❖ **Eclipse IDE**



L'Eclipse IDE est un environnement de développement intégré qui supporte une large variété des langages de programmation et d'outils de collaboration. L'éditeur intégré propose des fonctions de contrôles syntaxiques et sémantiques, d'avertissements et de conseil, de reprise de codes (« refactoring » : renommage, changement des méthodes, gestion des classes,...), de sauvegarde et de reprise.

❖ **MySQL**

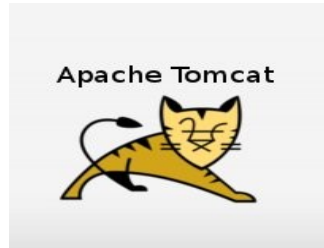


**MySQL** est une base de données relationnelle libre qui est très employée sur le Web, souvent en association avec PHP (langage) et Apache (serveur web). MySQL fonctionne indifféremment sur tous les systèmes d'exploitation. Il a le principe d'une base de données relationnelle et d'enregistrer

les informations dans des tables, qui représentent des regroupements de données par sujets. Les tables sont reliées entre elles par des relations.

Le langage SQL (Structured Query Language) est un langage reconnu par MySQL et les autres bases de données et permettant de modifier le contenu d'une base de données.

## ❖ Apache Tomcat



Tomcat est un serveur d'application Java. Ce qui signifie deux choses :

- Il est intégralement écrit en Java.
- Il est capable d'exécuter des applications qui sont développées en Java. ces applications sont destinées à traiter des requêtes web entrantes, et à générer une réponse adéquate.

Tomcat est un serveur d'applications qui a un rôle distinct de celui d'un serveur web.

## 3.3 Etapes de réalisation

### 3.3.1 L'architecture de l'application Web

Au sein des applications Web on retrouve généralement trois couches :

- La première couche est la présentation qui contient les servlets ou les contrôleurs en cas d'utilisation de Framework MVC, elle est chargée d'assurer le dialogue entre l'application et les services extérieurs.
- La deuxième couche est la couche métier qui contient les services et s'occupe des traitements.
- La dernière couche est la DAO qui s'occupe de l'accès aux bases de données.

La couche présentation (Vue) qui s'occupe de la saisie, le contrôle et l'affichage des résultats. Généralement la couche présentation respecte le pattern MVC qui fonctionne comme suit :

1. La vue permet de saisir les données, envoie ces données au contrôleur
2. Le contrôleur récupère les données saisies. Après la validation de ces données, il fait appel à la couche métier pour exécuter des traitements.
3. Le contrôleur stocke le résultat du modèle.
4. Le contrôleur fait appel à la vue pour afficher les résultats.
5. La vue récupère les résultats à partir du modèle et les affiche

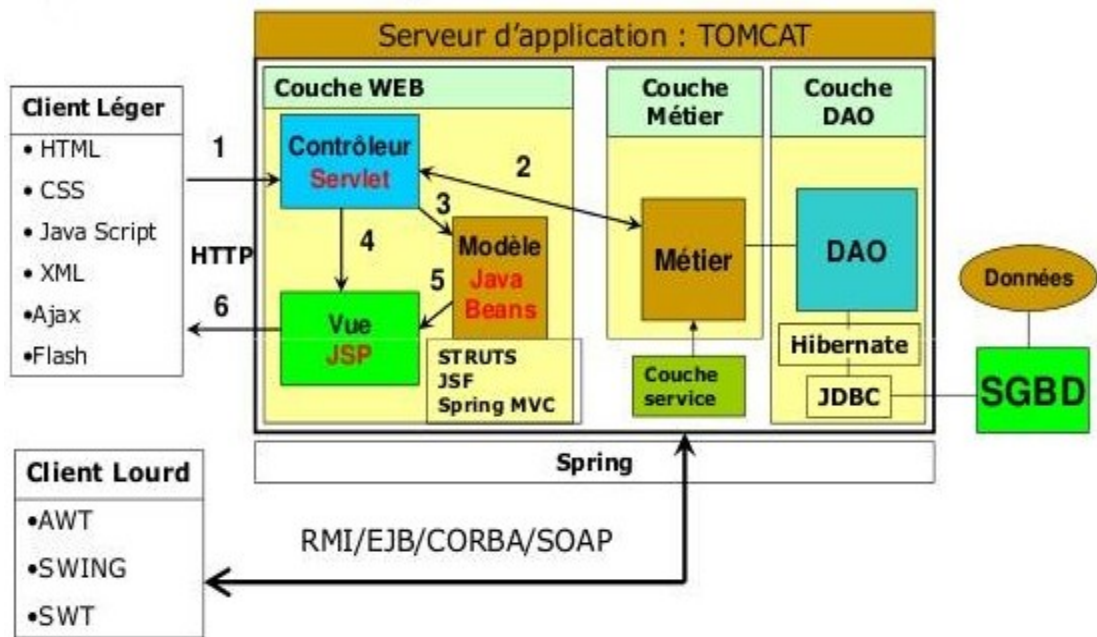


Figure 3. 1 : Architecture d'une application Web

### 3.4 . Les Interfaces de l'application

La figure 3.2 : présente la page d'accueil de l'application à laquelle peuvent se connecter étudiant ou enseignant.



Figure 3. 2 : Page d'accueil

La figure 3.3 : présente l'interface Authentification de l'application. Cette interface est décomposée deux champs pour la saisie du nom d'utilisateur (Login) et du mot de passe pour pouvoir accéder à l'application.

### Authentification Administrateur

Entrez votre login et mot de passe.

**Figure 3. 3 : Authentification**

La figure 3.4 : présente l'interface de saisie de base. Cette interface est accessible à l'administrateur seulement puisque il a tous les droits pour ajouter, modifier et supprimer tout type de données

INSTITUT SUPÉRIEURE DE SCIENCES APPLIQUÉES ET TECHNOLOGIES DE SOUSSE (ISSAT)admin

- Accueil
- Liste Filières
- Liste Niveaux
- Liste Départements
- Liste Grades
- Liste Types Enseignement
- Liste Salles
- Liste Séances
- Liste Matières
- Liste Groupes
- Liste Enseignant
- Emploi du temps

#### Nouvelle Filière

#### Liste des Filières :

Message de succès

Code	Filière	Actions
301	DNI: Informatique	<input style="background-color: #0070c0; color: white; padding: 2px 5px;" type="button" value="Editer"/> <input style="background-color: #d32f2f; color: white; padding: 2px 5px;" type="button" value="Supprimer"/>
307	Licence Appliquées en Informatique Industrielle: Systèmes et Automatismes	<input style="background-color: #0070c0; color: white; padding: 2px 5px;" type="button" value="Editer"/> <input style="background-color: #d32f2f; color: white; padding: 2px 5px;" type="button" value="Supprimer"/>
310	Licence Appliquées en Génie Mécanique: Conception et Fabrication	<input style="background-color: #0070c0; color: white; padding: 2px 5px;" type="button" value="Editer"/> <input style="background-color: #d32f2f; color: white; padding: 2px 5px;" type="button" value="Supprimer"/>

© Muhammad Usman 2012 - 2014 Powered by: Charisma

**Figure 3. 4 :Saisie d'une filière**

La figure 3.5 : présente l'interface de suppression d'une filière. Cette interface est accessible à l'administrateur seulement.

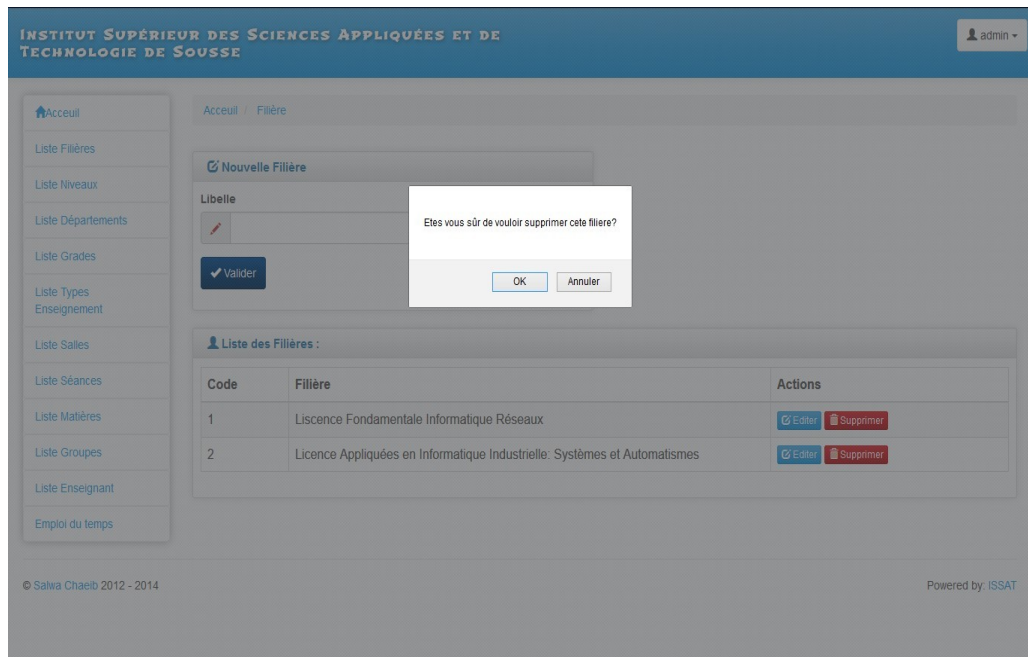


Figure 3. 5 : Suppression d'une filière

La figure 3.6 : présente l'interface de saisie d'une salle en précisant le type de salle et sa capacité. Cette interface est accessible à l'administrateur

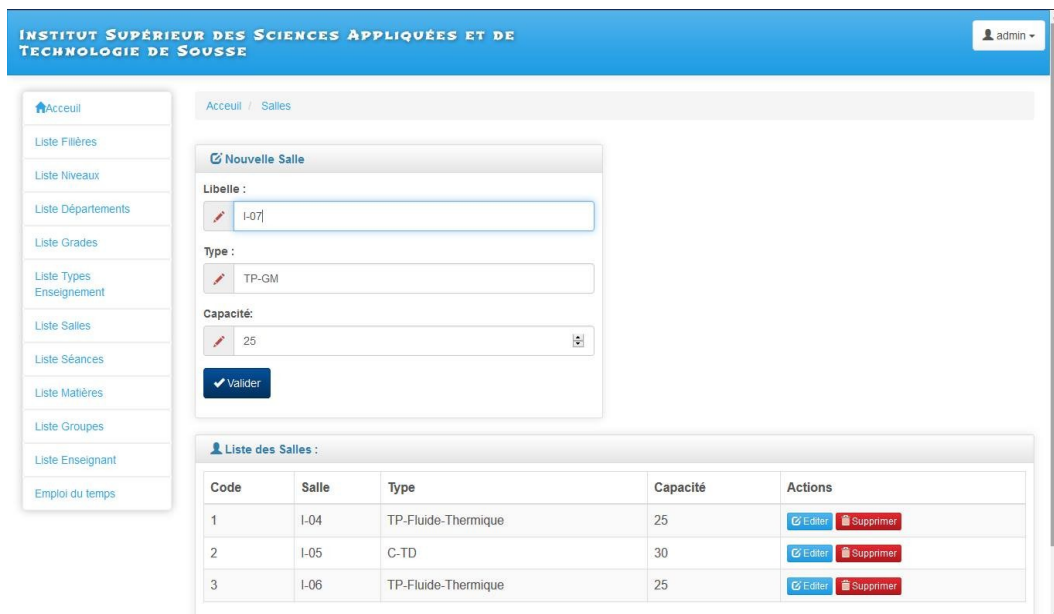
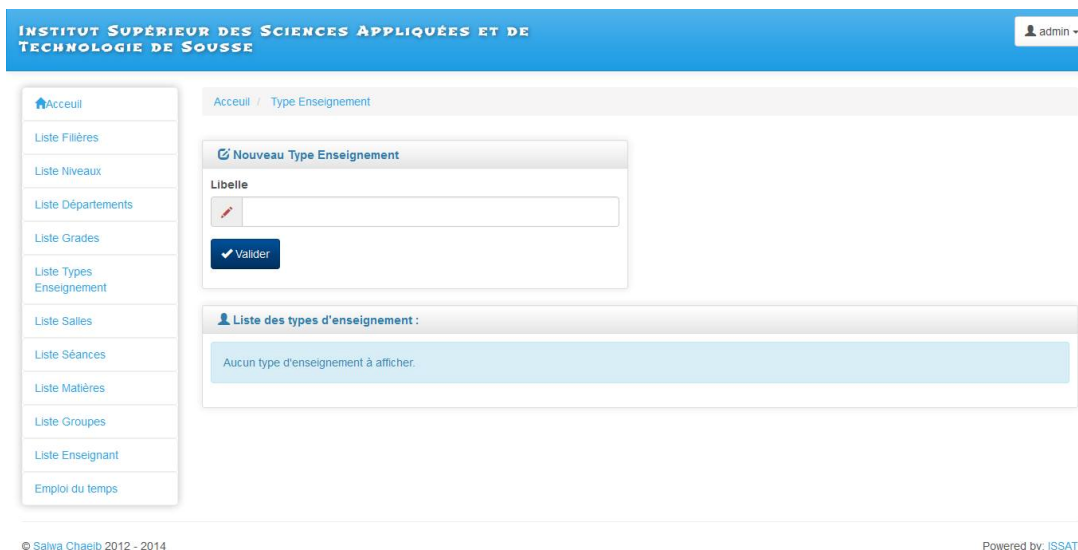


Figure 3. 6: Interface saisie Salle

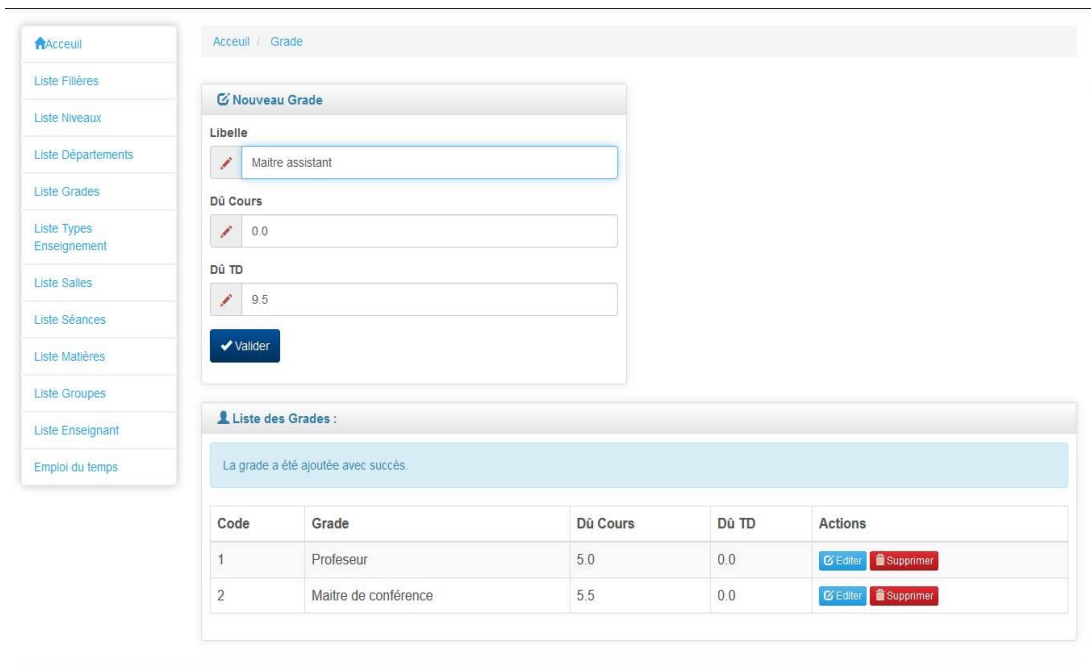


La figure 3.7 : présente l'interface de saisie d'un type d'enseignement. Cette interface est accessible à l'administrateur



**Figure 3. 7 : Interface saisie Type d'enseignement**

La figure 3.8 : présente l'interface de saisie d'un grade. Cette interface est accessible à l'administrateur



**Figure 3. 8 : Interface saisie grade**

La figure 3.9 : présente l'interface d'ajout d'un nouvel enseignant. Cette interface est accessible par l'administrateur seulement.

**Figure 3.9 : Interface d'ajout d'un enseignant**

La figure 3.10 : présente l'interface d'affectation d'une matière avec son type d'enseignement à un jour, un groupe, une salle et une séance.

**Figure 3.10 : Interface d'ajout d'une matière**

La figure 3.11 : présente l'interface de consultation de la liste des matières affectées.

INSTITUT SUPÉRIEUR DES SCIENCES APPLIQUÉES ET DE TECHNOLOGIE DE SOUSSE

admin

Accueil / Liste Affectation Matière

Recherche Affectation

Matières

Mathématique discrètes - Ing -Inf

Rechercher

Liste des Affectations des Matières :

Matiere	Groupe	Type Enseignement	Salle	Séance	Enseignant	Actions
Mathématique discrètes - Ing -Inf	FIA1-01	Cours	I-04	S1	Vacant	<a href="#">Editer</a> <a href="#">Supprimer</a>
Mathématique discrètes - Ing -Inf	FIA1-01	Travaux Pratique TP	I-06	S3	Vacant	<a href="#">Editer</a> <a href="#">Supprimer</a>

Figure 3. 11 : Interface de consultation des matières affectées.

## CONCLUSION

Au terme de ce projet de fin d'études, je tiens à souligner que sa réalisation était d'un très grand bénéfice pour moi car c'était une bonne occasion pour consolider mes connaissances théoriques dans le domaine de conception et la réalisation des applications informatiques.

Il est évidemment que ce projet n'est pas une œuvre parfaite mais j'ai tenu à ce qu'il soit à la hauteur de mes espérances professionnelles ainsi aux responsables de l'administration de l'ISSATS en espérant qu'elle trouve dans mon travail une bonne solution pour les différents problèmes en emploi du temps.

Le problème d'emploi du temps est un problème universel qui touche plusieurs secteurs et a une grande importance de façon que plusieurs études ont été effectuées afin d'avoir des solutions optimales.

En perspectives cette application pourrait être améliorée.

# BIBLIOGRAPHIE

## Livres :

- Langage de modélisation UML, Frédéric Julliard
- Servlets : Programmation d'applications Web avec Java(J2EE) Benjamin AUMAILLE
- Java 5 : entraînez-vous et maîtrisez le langage(Java 2 JDK 5.0 - J 2S25) , Alexandre Brillant
- Java la synthèse : concepts, architectures, Framework
- My SQL 5 : installation mise en œuvre administration programmation, Cyril Thibaud

## Site Web :

- <http://uml.free.fr/index-cours.html>
- <http://www.codejava.net/>
- <https://www.mysql.fr/>
- <http://www-igm.univ-mlv.fr/~dr/XPOSE2003/tomcat/tomcat.php?rub=5&id=1>
- <http://tomcat.apache.org/index.html>

## ANNEXE A

L'Institut Supérieur des Sciences Appliquées et de Technologie de Sousse est une institution d'enseignement supérieur, sous tutelle du Ministère de l'Enseignement Supérieur, de la Recherche Scientifique, il a pour mission s'assurer des formations en Licence, Cycle Préparatoires, Formation Ingénieur, Mastère Pro et Recherche.

Les 430 enseignants sont répartis sur 5 départements dont 200 permanents et 230 Vacataires.

Les départements existants sont :

- Département Informatique
- Département Génie mécanique
- Département Génie Electrique
- Département Energétique
- Département Administration

L'ISSATS possède :

- 14 filières :
  - 7 Licences Appliquées
  - 2 Licences Fondamentales
  - 1 Ingénieur (Génie Logiciel et Informatique Industrielle)
  - 2 mastères Recherche
  - 2 mastères professionnels
- 2974 étudiants répartis sur 110 groupes cours.
- 52 Salles C/TD et 52 Salles TP (Informatique, Electrique, Energétique et Mécaniques)
- 17554 ouvrages

# TABLE DES FIGURES

<b>Nom de la figure</b>	<b>Page</b>
Figure 1 : Organigramme de l'Ipelsht.....	7
Figure 2 : Exemple de feuille de remplissage des notes.....	10
Figure 3 : Exemple de feuille Excel remplie de notes.....	11
Figure 4 : Exemple de feuille Excel de gestion des notes.....	11
Figure 5 : Schéma descriptif du système de gestion des notes.....	18
Figure 6 : Cas d'utilisation global.....	24
Figure 7: Gérer les Utilisateurs.....	25
Figure 8 : Gérer les Classes.....	28
Figure 9 : Gérer les Spécialités.....	31
Figure 10 : Gérer les matières.....	37
Figure 11 : Gérer les étudiants.....	40
Figure 12 : Gérer les notes.....	42
Figure 13 : Gérer les résultats.....	45
Figure 14: Diagramme de Séquence de l'Authentification.....	48
Figure 15 : Diagramme de Séquence Ajouter un nouvel utilisateur.....	49
Figure 16 : Diagramme de Séquence Modifier un utilisateur.....	50
Figure 17 : Diagramme de Séquence Ajouter une note.....	51
Figure 18 : Diagramme de Séquence consulter et imprimer un résultat.....	52
Figure 19 : Diagramme des classes.....	54
Figure 20 : Page d'authentification administrateur.....	58
Figure 21 : Page d'ajout Utilisateurs.....	59
Figure 22 : Page de modification Administrateur.....	60
Figure 23 : Page de consultation d'utilisateur ajouté .....	61

Figure 24 : Interface d'accueil du chef département .....62

Figure 25 : Interface d'accueil de l'Agent.....63



# TABLE DES TABLEAUX

Nom du tableau	Page
Tableau 1 : Ajouter un utilisateur.....	26
Tableau 2 : modifier un utilisateur.....	26
Tableau 3 : Supprimer un utilisateur.....	27
Tableau 4 : Consulter liste utilisateurs.....	27
Tableau 5 : Ajouter une classe.....	29
Tableau 6 : Modifier une classe.....	29
Tableau 7 : Supprimer une classe.....	30
Tableau 8 : Consulter une classe.....	30
Tableau 9 : Ajouter une spécialité.....	32
Tableau 10 : Modifier une spécialité.....	32
Tableau 11 : Supprimer une spécialité.....	33
Tableau 12 : Consulter une spécialité.....	34
Tableau 13 : Ajouter un module.....	35
Tableau 14 : Modifier un module.....	35
Tableau 15 : Supprimer un module.....	36
Tableau 16 : Consulter un module.....	36
Tableau 17 : Ajouter une matière.....	37
Tableau 18 : Ajouter une matière.....	38
Tableau 19 : Supprimer une matière.....	39
Tableau 20 : Consulter une matière.....	39
Tableau 21 : Ajouter un étudiant.....	40
Tableau 22 : Modifier un étudiant.....	41
Tableau 23 : Supprimer un étudiant.....	41
Tableau 24 : Consulter un étudiant.....	42

Tableau 25 : Ajouter une note.....	43
Tableau 26 : Modifier une note.....	44
Tableau 27 : Supprimer une note.....	44
Tableau 28 : Consulter une note.....	45
Tableau 29 : Enregistrer un résultat.....	46
Tableau 30 : Imprimer un résultat.....	47