

## Table des matières

|  |           |
|--|-----------|
| Table des matières .....   |           |
| Liste des Figures :.....   |           |
| Liste des tableaux :.....  |           |
| Dédicace :.....  |           |
| Remerciements.....   |           |
| Liste des abréviations :.....                                      |           |
| Introduction : .....   |           |
| <b>PARTIE I : CONTEXTE GENETAL DU PROJET .....</b>                 | <b>9</b>  |
| Chapitre 1 : Présentation de l'organisme d'accueil .....           |           |
| 1 Présentation générale de la TGR :.....                           | 10        |
| 2 Missions principales de la TGR : .....                           | 10        |
| 3 Organisation de la TGR :.....                                    | 11        |
| 3.1 Organigramme de la TGR :.....                                  | 12        |
| 3.1.1 Service de la production de la paie :.....                   | 13        |
| Chapitre 2 : Présentation du projet.....                           |           |
| 1 Contexte et but du projet : .....                                | 15        |
| 2 Démarche Suivie et planification : .....                         | 15        |
| 2.1 Processus de développement :.....                              | 15        |
| 2.2 Planification du projet : .....                                | 17        |
| <b>PARTIE II : ANALYSE ET CONCEPTION DU PROJET .....</b>           | <b>18</b> |
| Chapitre 3 : Analyse et Spécification :.....                       |           |
| 1 Fonctionnalités principales :.....                               | 19        |
| 2 Analyse du contexte : .....                                      | 21        |
| 2.1 Identification des acteurs :.....                              | 21        |
| 2.1.1 Interaction acteur – système et les messages échangés :..... | 21        |
| Chapitre 4 : Conception du Système .....                           |           |
| 1 Formalisme UML :.....  | 23        |
| 2 Diagrammes de cas d'utilisation : .....                          | 24        |
| 3 Description de quelques cas d'utilisation :.....                 | 27        |
| 4 Diagramme de packages :.....                                     | 28        |
| 5 Diagrammes de séquences :.....                                   | 29        |
| 6 Diagramme de classes : .....                                     | 32        |
| 6.1 Diagramme de classes et package UC du système GIF : .....      | 32        |
| 7 Architecture Technique du Système :.....                         | 35        |
| <b>PARTIE III : MISE EN ŒUVRE DU PROJET .....</b>                  | <b>36</b> |
| Chapitre 5 : Technologies et outils utilisés.....                  |           |

|      |   |    |
|------|---|----|
| 1    | Technologie et Architecture Java EE : ..... | 37 |
| 2    | JSP : .....                                 | 38 |
| 3    | JSTL : .....                                | 39 |
| 4    | Hibernate Framework : .....                 | 39 |
| 4.1  | JPA et Hibernate : .....                    | 41 |
| 5    | Design pattern MVC : .....                  | 41 |
| 6    | Spring Framework : .....                    | 42 |
| 6.1  | Spring MVC : .....                          | 42 |
| 6.2  | Principe de fonctionnement : .....          | 43 |
| 7    | Eclipse IDE : .....                         | 44 |
| 8    | Oracle: .....                               | 45 |
| 9    | Le serveur apache Tomcat : .....            | 45 |
| 10   | Subversion : .....                          | 46 |
| 10.1 | Jasper Report Framework : .....             | 46 |

## Chapitre 6 : Mise en œuvre de la solution GIF .....

|     |  |    |
|-----|--|----|
| 1   | Architecture applicative : .....                           | 48 |
| 2   | Réalisation des couches applicatives : .....               | 49 |
| 2.1 | La couche base de données : .....                          | 49 |
| 2.2 | La couche DAO : .....                                      | 49 |
| 2.3 | La couche service : .....                                  | 49 |
| 2.4 | La couche Controller : .....                               | 49 |
| 2.5 | La couche Présentation : .....                             | 49 |
| 3   | Les écrans de la réalisation : .....                       | 49 |
| 3.1 | Page de démarrage : .....                                  | 49 |
| 3.2 | Page des statistiques : .....                              | 50 |
| 3.3 | Page gestion des formateurs (liste des formateurs) : ..... | 51 |
| 3.4 | Page, ajouter formateur : .....                            | 53 |

Conclusion et perspectives : .....

Webographie : .....

Bibliographie : .....

### Liste des Figures :

|   |    |
|---|----|
| Figure 1 : Organigramme de la Trésorerie Générale du Royaume .....                | 12 |
| Figure 2 : Organigramme Direction de l'Appui et de la Gestion des Ressources..... | 14 |
| Figure 3 : diagramme de Gantt, planning du projet .....                           | 17 |
| Figure 4 : Boîte noire du système GIF .....                                       | 22 |
| Figure 5 : cas d'utilisation Administrateur « gestion globale » .....             | 26 |
| Figure 6 : cas d'utilisation « imprimer état » .....                              | 26 |
| Figure 7 : cas d'utilisation « attribuer séance de formation » .....              | 27 |
| Figure 8 : diagramme de packages « des cas d'utilisation » .....                  | 29 |
| Figure 9 : diagramme de séquence « imprimer état » .....                          | 30 |
| Figure 10 : Scénario « attribuer séance de formation ».....                       | 31 |
| Figure 11 : scénario « ajouter formateur » selon le design pattern MVC .....      | 31 |
| Figure 12 : diagramme de classes du système GIF .....                             | 33 |
| Figure 13 : exemple diagramme de packages Formation (architecture logique) .....  | 34 |
| Figure 14 : architecture technique du système .....                               | 35 |
| Figure 15 : Architecture de la technologie JEE.....                               | 38 |
| Figure 16 : l'emplacement de Hibernate entre le SGBD et l'Application .....       | 41 |
| Figure 17 : principe de base du MVC .....   | 43 |
| Figure 18 : principe de fonctionnement MVC avec Spring MVC .....                  | 44 |
| Figure 19 : fichier de configuration web.xml.....                                 | 44 |
| Figure 20 : fichier de la servlet spring-servlet.xml .....                        | 44 |
| Figure 21 : exemple « commit » du projet GIF sur le serveur svn de assembla ..... | 46 |
| Figure 22 : Architecture applicative du système GIF .....                         | 49 |
| Figure 23 : page d'authentification .....   | 50 |
| Figure 24 : page d'accueil « statistiques & tableau de bord ».....                | 51 |
| Figure 25 : gestion des formateurs (liste des formateurs) .....                   | 52 |
| Figure 26 : gestion des formateurs (Ajouter un formateur).....                    | 53 |

**Liste des tableaux :**

|  |    |
|--|----|
| Tableau 1 : liste des abréviations .....   | 8  |
| Tableau 2 : Les messages échangés entre le système et les acteurs .....              | 22 |
| Tableau 3 : les cas d'utilisations selon les acteurs .....                           | 25 |
| Tableau 4 : description du cas d'utilisation « imprimer état » .....                 | 28 |
| Tableau 5 : description du cas d'utilisation « attribuer séance de formation » ..... | 28 |
| Tableau 6 : description des classes.....   | 35 |
| Tableau 6 : les tags Library du jstl .....   | 39 |
| Tableau 7 : correspondance du modèle objet au modèle BDR .....                       | 40 |

**Dédicace :**

*Je dédie ce travail :*

*À mes chers parents qui m'ont soutenu tout au long de mes études ainsi que dans ma vie entière, pour les sacrifices qu'ils ne cessent de m'offrir, je vous dédie ce travail avec tout le respect que je vous garderai.*

*À ma chère sœur, à toute ma famille et à tous mes amis.*

*À mes professeurs et à mes encadrants.*

*À tous le Personnel de la trésorerie générale du royaume*

*Mohammed Amine*

**Remerciements**

Je tiens d'abord à remercier le corps professoral du Département Informatique de la Faculté des Sciences et Techniques Fès et spécialement mon encadrant pédagogique Professeur AICHA Majda ainsi que tout le Staff Administratif.

Un Grand Merci au Professeur BENABBOU Rachid et tous les Professeurs et Responsables de notre formation pour leurs efforts Afin que notre formation soit une réussite et un tournant décisif dans notre vie.

Merci à vous les membres du jury qui a accepté de juger ce travail pour votre disponibilité et écoute attentive.

Un Merci particulier à Monsieur SOUFAIH Hamid pour m'avoir accueilli au sein de son service à la trésorerie générale du royaume ainsi que pour sa patience tout au long du stage.

Je Remercie Mr. BAROUAGUE Mohamed, Mr. BENNANI Mohamed, Mme. Fatma et tous le personnel de la trésorerie générale du royaume pour ce qu'ils m'ont fait découvrir et pour les bons moments passés avec eux.

Je m'adresse aussi à remercier les gens des communautés web des Développeurs, stackoverflow, developpez.net et tous les autres.

Ainsi que toute personne qui a contribué de près ou de loin à l'élaboration de ce travail

Enfin je remercie ma famille et mes amis pour leur chaleureux soutien.

**Liste des abréviations :**

| Abréviation | Désignation                            |
|-------------|--|
| GIF         | Gestion des Indemnités des Formations  |
| TGR         | Trésorerie Générale du Royaume         |
| EDI         | Echange des Données Informatisé        |
| UML         | Unified Modeling Language              |
| CRUD        | Create Read Update Delete              |
| UC          | Use Case                               |
| MVC         | Modèle-Vue-Contrôleur                  |
| DAO         | Data Access Object                     |
| JPA         | Java Persistence API                   |
| API         | Application Programming Interface      |
| EJB         | Enterprise Java Beans                  |
| SDK         | Software Development Kit               |
| JSP         | Java Server Page                       |
| JEE         | Java Enterprise Edition                |
| IHM         | Interaction Homme Machine              |
| XML         | eXtensible Markup Language             |
| EL          | Expression Language                    |
| ORM         | Object Relational Mapping              |
| JDBC        | Java DataBase Connectivity             |
| POJO        | Plain Old Java Object                  |
| SQL         | StructuredQuery Language               |
| SGBD        | Système de gestion de bases de données |
| HTML        | HypertextMarkup Language               |

**Tableau 1 : liste des abréviations**

### **Introduction :**

Le développement de tout organisme repose le plus souvent sur son aptitude à gérer ses ressources humaines, matérielles et financières ainsi que l'atmosphère actuelle qui règne au sein du secteur des technologies de l'information poussent les organismes privées et publiques, qui se base sur les technologies et les systèmes d'informations, à diminuer les coûts de leurs solutions.

Les organismes conscients de l'importance des solutions informatique d'un coté et les formations d'un autre, comme étant un pilier de toute stratégie de gestion des ressources humaines d'une part, et un enjeu clé pour accompagner les changements émanant des évolutions d'autre part, - ces organismes - ont développés leurs politiques de formations qui leurs permettent de diagnostiquer et par conséquent enrichir leurs référentiels de compétences. Cet enrichissement permet d'avoir un rendement important et de palier aux dysfonctionnements à court, moyen et long terme.

Dans ce sens et conformément à sa stratégie de veille technologique et économique, la Trésorerie générale du Royaume a jugé nécessaire l'élaboration de sa politique lui permettant de diagnostiquer les compétences disponibles afin de les faire évoluer.

Ainsi, mon projet a pour but de doter les responsables de la formation continue au sein de l'administration d'un système GIF leurs permettant la gestion des différentes composantes de la formation à savoir, la gestion des formations, des formateurs et leurs indemnités.

Ce rapport présente le travail réalisé durant les deux mois de ma période de stage. Il comporte 3 parties.

La première partie, composée de deux chapitres, définit le contexte général du projet. Il débute par la présentation de l'organisme d'accueil, ensuite le deuxième chapitre décrit le projet ainsi que la démarche suivie.

La deuxième partie concerne les concepts de modélisation du projet. Ainsi, le premier chapitre est consacré à l'analyse et la spécification des besoins afin d'énumérer les fonctionnalités principales du système à développer et de délimiter le périmètre fonctionnel de ce futur système. Le deuxième chapitre fera l'objet de la phase de conception du projet en le formalisme UML.

La dernière partie du rapport aborde l'aspect technique du projet. Dans le premier chapitre de cette partie je vais exposer les différentes technologies et Framework que j'ai utilisée aussi les outils qui mon aidé de mettre en œuvre mon projet. Quant au dernier chapitre, il sera réservé à la phase de réalisation.

Une synthèse du travail réalisé sera présentée à la fin du rapport comme Conclusion.

## **PARTIE I : CONTEXTE GENETAL DU PROJET**

Cette partie comprend deux chapitres :

Le premier chapitre introduit l'organisme d'accueil « la Trésorerie Générale du Royaume », ses activités, et son organigramme.

Le deuxième chapitre présente le sujet du projet et présente son objectif. Il précise également la démarche suivie dans la conduite du projet et le planning général de la réalisation.



## Chapitre 1 : Présentation de l'organisme d'accueil

---

Dans ce chapitre je présente l'organisme d'accueil : la Trésorerie Générale du Royaume. J'exposerai d'abord, les activités de la TGR puis son organisation ainsi que le service ou j'ai réalisé mon projet.

### INTRODUCTION :

Comme je l'ai indiqué auparavant, mon projet s'est déroulé au sein de la TGR, dont le fonctionnement est axé autour de plusieurs activités, et se base sur plusieurs systèmes d'informations matures et puissants.

## Présentation générale de la TGR :

La Trésorerie Générale du Royaume constitue l'une des administrations les plus importantes du Ministère des Finances et à travers laquelle transite l'ensemble des flux financiers et comptables de l'Etat et des collectivités locales.

Elle est également au centre d'un maillage institutionnel constitué d'administrations publiques, d'établissements publics, de collectivités locales et d'autres grandes institutions financières tous concernés par la gestion des deniers publics. La TGR a initié, depuis 8 ans, un grand projet de modernisation dont la vision stratégique est sous-tendue par deux objectifs fondamentaux à savoir :

- La contribution à l'amélioration substantielle de la gestion des finances publiques.
- L'amélioration du service rendu aux clients et partenaires.

## Missions principales de la TGR :

### *a) Le recouvrement des impôts et produits divers :*

La TGR assure, par le biais de son vaste réseau de comptables publics, la perception des recettes fiscales et non fiscales, à travers notamment :

- La gestion du contentieux administratif et judiciaire relatif au recouvrement et
- L'assistance des percepteurs en la matière.
- La prise en charge des ordres de recettes au titre du budget général de l'Etat, des budgets annexes et des comptes spéciaux du Trésor.
- La centralisation des prises en charges et des recouvrements au titre des amendes et condamnations pécuniaires.
- La gestion des comptes de prêts et d'avances accordées par le trésor et de «fonds de roulement» consentis par des organismes de financement des projets publics.
- L'élaboration des statistiques concernant la situation du recouvrement de créances publiques.

- La production du compte de gestion «recettes» du Trésorier Général du Royaume en tant que comptable principal.

*b) Le contrôle et le paiement des dépenses publiques :*

La TGR assure le contrôle et le règlement des dépenses publiques. Ainsi, en tant qu'institution investie de mission de contrôle, suite au rapprochement avec le Contrôle Général des Engagements de Dépenses (CGED), la TGR est chargée de contrôler la régularité des engagements de la quasi-totalité des dépenses du budget général de l'Etat. En tant que comptable supérieur de l'Etat, elle assure le règlement des dites dépenses. En effet, au vu des ordonnances et mandats transmis par les ordonnateurs accrédités et après visa des contrôleurs, les comptables procèdent au règlement de ces créances.

La Trésorerie Principale assure également par le biais de la Paierie Principale des Rémunérations (PPR), le contrôle et la gestion de la paie de plus que 700.000 fonctionnaires.

*c) La gestion des finances locales :*

A travers son réseau de trésoriers, receveurs communaux et percepteurs communaux, la TGR effectue le suivi et la centralisation de l'exécution des budgets de 1786 collectivités locales, de 86 groupements et de 41 arrondissements, à travers un réseau constitué de 16 Trésoreries Régionales, 4 Trésoreries Préfectorales, 29 Trésoreries Provinciales dont 3 Trésoreries provinciales recettes communales, 8 Trésoreries Communales, 21 Recettes Communales et 115 Perceptions Recettes Communales.

En effet, la TGR assure le recouvrement de leurs créances, le règlement des leurs dépenses et la paie de leur personnel.

*d) L'activité bancaire :*

A travers sa mission de collecte et de gestion de l'épargne, la TGR participe au financement de la trésorerie de l'Etat. A ce titre, elle assure la gestion des comptes des entreprises et établissements publics qui sont soumis à l'obligation de dépôt, auprès d'elle, de leurs fonds. La TGR assure également la gestion de dépôt des particuliers et des personnes morales privées.

*e) La production de l'information financière et comptable :*

La TGR assure la centralisation des opérations comptables de l'Etat et des collectivités locales et de ce fait elle constitue une référence en matière de production et de valorisation de l'information comptable de l'Etat et des collectivités locales.

La production de l'information comptable permet ainsi de :

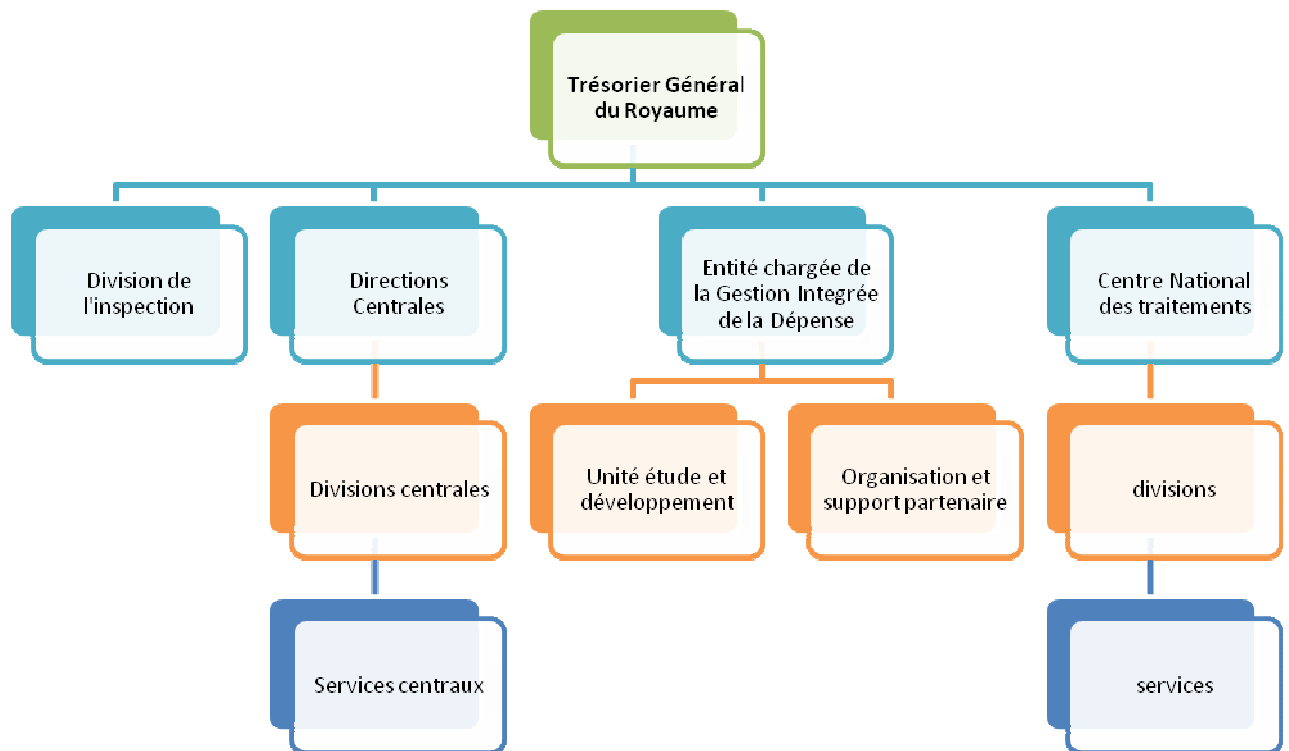
- Décrire précisément les opérations budgétaires et financières.
- Restituer rapidement une information fiable et indispensable à la prise de décision.
- Satisfaire les besoins et les échéances du contrôle de la Cour des Comptes et du Parlement.

## **Organisation de la TGR :**

La TGR est organisée en services centraux et services extérieurs. Le nombre des fonctionnaires au sein de ces deux services s'élève à 6800 répartis sur des échelles et des catégories diverses :

- Trésoreries des chancelleries diplomatiques et consulaires
- Trésoreries régionales
- Trésoreries provinciales
- Trésoreries communales
- Perceptions
- Recettes Communales
- trésoreries ministérielles

## Organigramme de la TGR :



**Figure 1 : Organigramme de la Trésorerie Générale du Royaume**

Les services centraux sont placés sous l'autorité du trésorier général du royaume de la trésorerie, assisté par des directeurs centraux et des chefs de divisions. L'organisation des services centraux repose sur 4 directions, groupant 16 divisions, ainsi qu'un centre national de traitement chargé des opérations comptables avec la centralisation financière et comptable. Une division d'inspection et une entité chargée de la gestion intégrée de la dépense rattachées directement au trésorier.

## Service de la production de la paie :

Mon projet s'est déroulé dans le service de la production de la paie (SPP). Ce service appartient à la division de la production informatique composée de quatre services :

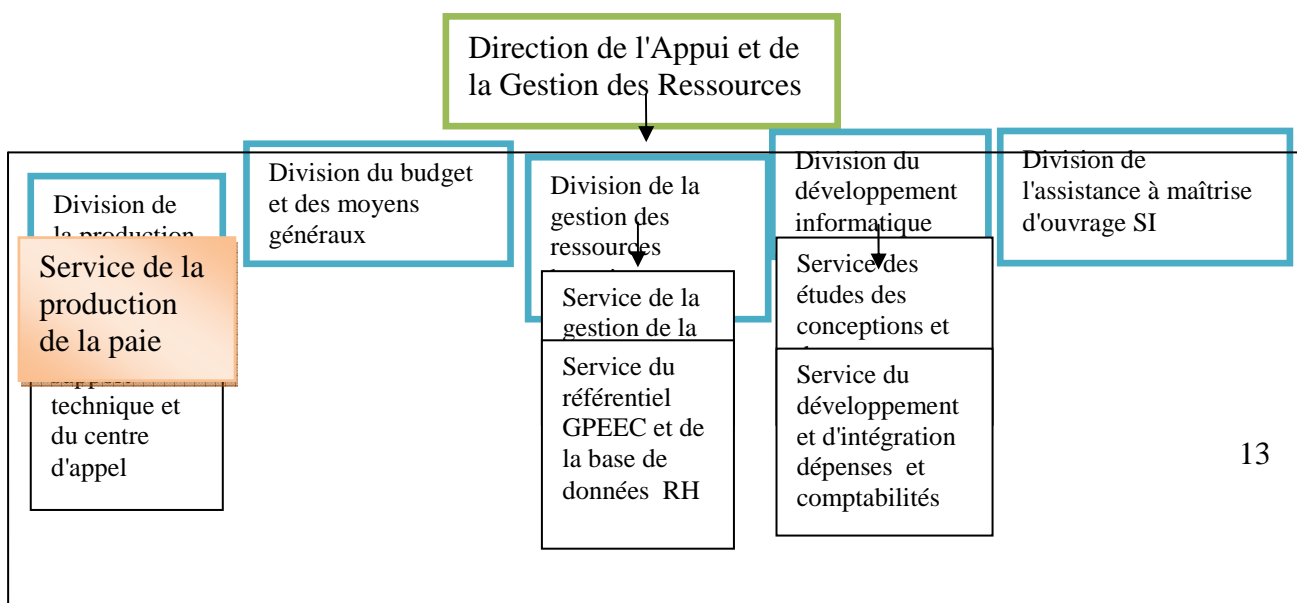
- Service du support techniques et du centre d'appel
- Service de la messagerie de l'administration des réseaux et de la téléphonie
- Service de l'administration du portail de la télé service et de l'EDI (Echange de Données Informatisé)
- Service de l'administration et de l'exploitation des systèmes.

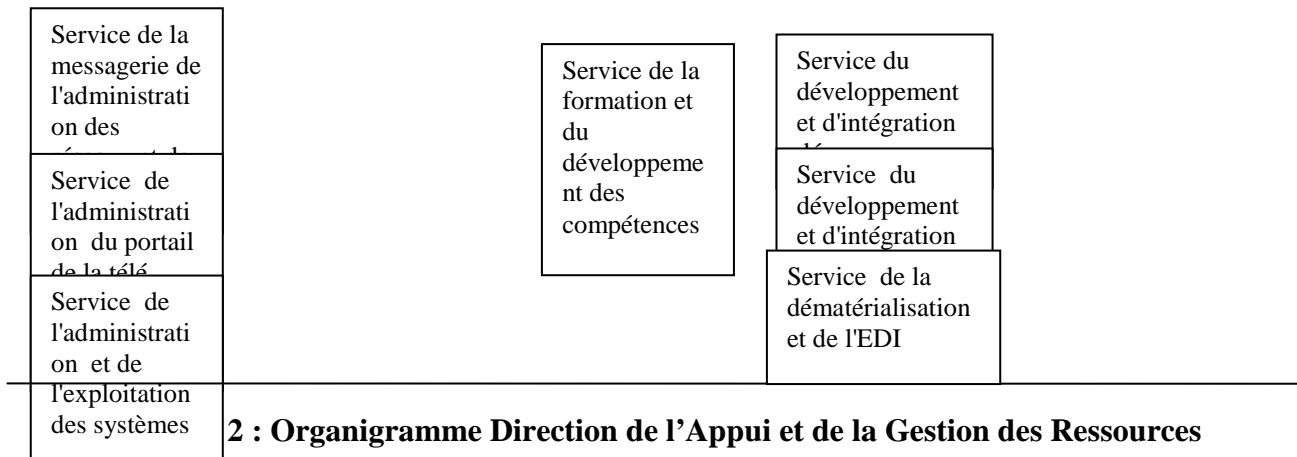
La division de la production informatique est sous l'autorité de la direction de l'appui et de la gestion des ressources.

### Le SPP est chargé :

- De la gestion de la paie et suivi administratif des salariés.
- D'animer ou de participer au pilotage des systèmes de gestion intégrée.
- Participer à concevoir, d'organiser et de mettre en œuvre les systèmes d'information de la Trésorerie Générale du Royaume.
- D'assurer la programmation et la gestion des ressources budgétaires déléguées ou affectées à la Trésorerie.

L'organigramme dans la page suivante montre l'emplacement de du service production de la paie dans la hiérarchie de la direction de l'Appui et de la Gestion des Ressources à la TGR.





## CONCLUSION :

Dans le présent chapitre, j'ai présenté l'organisme d'accueil en termes d'activités et d'organisation. Le service de la production de la paie, dans lequel s'est déroulé mon projet, a été situé dans l'organigramme de la TGR avant la présentation du projet GIF (Gestion des indemnités des formations).

Le chapitre suivant présente le sujet du projet, et la démarche suivie pour sa réalisation.

## Chapitre 2 : Présentation du projet

Dans ce chapitre, je vais introduire le sujet du projet ainsi que son objectif et le but visé. Je termine par présenter la démarche adoptée et le planning général suivi pour la réalisation de mon projet.

## INTRODUCTION :

Au bout des quatre dernières années la Trésorerie Générale du Royaume a complètement changé sa politique envers ses ressources humaines, elle a amélioré ses relations au niveau du siège et de tout le réseau national, cela s'inscrit dans la modernisation de la trésorerie.

Cette action stratégique avait comme buts principaux :

- L'amélioration de la qualité du service rendu à la clientèle.
- Le développement des compétences liées à l'exercice des métiers du réseau.
- Elever le professionnalisme.
- Le partage d'expériences et des méthodes du travail.
- La sensibilisation concernant la sécurité, l'optimisation d'utilisation des ressources et des systèmes d'information.
- Minimiser les impacts.
- La prévention des risques ainsi que l'adaptation aux mesures internationales.

Cette démarche était la solution indispensable pour la valorisation des ressources humaines -la véritable richesse de l'organisation - et qui a ciblé plus de 1500 fonctionnaires (responsables, cadres et agents) à travers des formations planifiées.

Ces formations qui concernent tous les paliers et les services de l'organisation et qui abordent de différents modules dont il y a plusieurs thèmes qui sont animés par des individus interne ou externe qui passent par des marchés publics.

## **Contexte et but du projet :**

Consciente de l'importance de l'informatique, la TGR accorde aujourd'hui une attention particulière à son système d'information en investissant dans des progiciels de gestion complets acquises récemment tel que GID (gestion intégrée de dépenses), GIR (gestion intégrée de recette) et PGCE (projet gestion de comptabilité de l'état), @UJOUR (qui permet une gestion performante et intégrée de tous les aspects de la rémunération du personnel de l'Etat, en acceptant la prise en compte rapide des réformes grâce à son système de paramétrage) qui facilitent et automatisent une grande partie du travail et qui aident à offrir des statistiques accompagnées des tableaux de bords, ils participent même à l'aide de prise de décisions.

Dans notre projet, le service de la formation et du développement des compétences à la DRH (division de la gestion des ressources humaines), chargé de la mission de gestion des formations, doit pouvoir gérer les formations et les planifications des cycles et séances de formations ainsi que les formateurs en fin du compte il doit avoir la possibilité de générer un état des sommes dues pour frais d'animation finale au profit de chaque formateur ayant participé à l'animation d'une séance de formation faisant partie d'un cycle de formation dans un trimestre glissant (90 jours), en addition de la gestion complète des indemnités (rémunérations) des formateurs en respectant des contraintes.

avant, cette tâche était effectuée manuellement en saisissant les coordonnées de chaque formateur et des informations concernant le thème de formation et finalement l'imprimer à partir d'un document Word, cela augmentait considérablement la complexité de cette gestion, l'accès ou la saisie de certaines données demandait beaucoup de concentration aussi la perte du temps avec un risque de perte des données considérablement élevé aussi, donc pas de sécurité soit au niveau des problèmes cités ou au niveau de traçabilité efficace de l'information ce qui oblige de trouver une solution qui permet de gérer les données d'une façon simple, diminuer le temps et de centraliser les données aussi les rendre facilement accessible et modifiable avec une interface IHM ergonomique et légère.

## **Démarche Suivie et planification :**

### **Processus de développement :**

Comme c'est le cas pour l'élaboration de tout projet, j'ai été amené à suivre une démarche pour la conduite du projet, pour cela j'ai choisi de répartir mon travail en quatre phases Primaires :

- Exploration : Dans cette phase on détermine la vision finale du projet et son contexte ainsi que le champ et les limites du projet.
- Elaboration : Phase de spécification des besoins, de planification et de définition de l'architecture du futur système en terme Analyse et Conception des modèles Statiques et Dynamiques. Dans cette phase on définit également les critères qualité à satisfaire dans le futur système.
- Construction : Le Système est construit itérativement. Chaque itération est un mini-projet à part avec tout ce qu'il nécessite en terme d'analyse, de conception, d'implémentation et de tests. Chaque itération a pour but de réaliser complètement un ou plusieurs cas d'utilisations.
- Déploiement : comme son nom l'indique cette phase de déploiement du produit couvre la formation des utilisateurs, la documentation du produit, ...)



## Planification du projet :

Pour mener à bien mon projet j'ai élaboré un plan de travail grâce au diagramme de Gantt qui est relatif à la gestion du projet et qui me permettra dans une courte durée, d'obtenir de bons résultats alors là j'ai planifié mes différentes tâches en accordant à chacune une durée qui devra être précise.

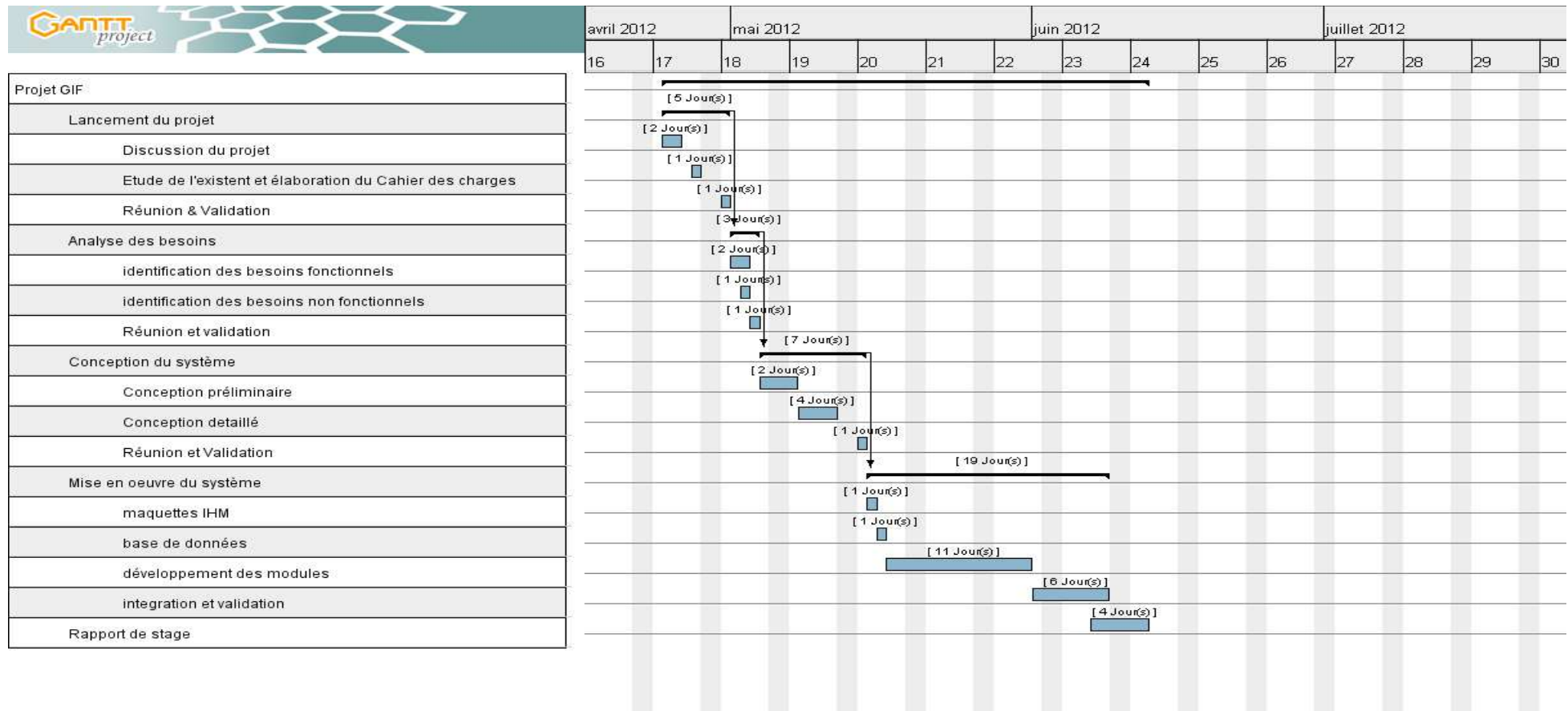


Figure 3 : diagramme de Gantt, planning du projet



## CONCLUSION :

Dans ce chapitre, j'ai présenté le projet GIF dans une introduction suivie d'une petite explication concernant le déroulement des formations au sein de la TGR ce qui nous a mené vers le but du projet. La présentation de la démarche adoptée et le planning ont été le sujet de la seconde partie de ce chapitre.

Dans le chapitre suivant qui inclut dans la deuxième partie du rapport est consacré à la spécification, l'analyse et la conception de la solution GIF.

## PARTIE II : ANALYSE ET CONCEPTION DU PROJET

Cette partie traite les deux phases les plus importantes et plus critiques de tout cycle de développement. Ce sont l'analyse et la conception.

Un premier chapitre, présente la phase d'analyse dont le but est de délimiter le périmètre du projet en identifiant les fonctionnalités auxquelles le système doit répondre, ainsi que les entités externes (acteurs) qui réagissent avec le système.

Les résultats de la phase d'analyse sont la base de la phase de conception traitée par le deuxième chapitre. Ce dernier commence par les cas d'utilisation réalisés à l'aide de la conception avec la méthode UML ce qui permet de bien maîtriser la fonctionnalité du système.



## Chapitre 3 : Analyse et Spécification :

---

### INTRODUCTION :

Lors de cette phase j'ai eff développé, Ces besoins on chargé de la gestion des états des sommes dues pour frais d'animations lors des réunions et quelques formateurs qui ont participé à l'animation de quelques séances de formation. Cette première phase du cycle de développement avait comme résultat, le cahier des charges utilisateur et le cahier des charges fonctionnel. Dans ce présent chapitre je vais présenter l'essentiel de ces deux livrables.

Ce chapitre décrit la phase d'analyse et de spécification du projet. Dans un premier temps, je vais identifier les fonctionnalités auxquelles le système de gestion des indemnités des formations doit répondre.

La seconde partie traite l'analyse du contexte du projet qui permet d'identifier les acteurs et les interactions entre le système et le monde extérieur afin de délimiter le périmètre fonctionnel.

## Fonctionnalités principales :

Cette partie est réservée à l'énumération ainsi qu'à la description des différentes fonctionnalités que le système de gestion des indemnités des formations doit assurer. Une analyse profonde du cahier des charges utilisateur détaillé que j'ai établi m'a mené de dégager les fonctionnalités suivantes :

- **Gestion des formateurs :**

Les données des formateurs sont saisies dans l'application intranet par l'administrateur ou par le formateur lui-même (s'inscrire), ensuite enregistré directement dans la base de données. Ces informations doivent pouvoir être éditées par les deux profils utilisateurs de l'application et supprimées seulement par l'administrateur. La consultation et le suivi des informations et aussi possible, ainsi que l'option de la recherche multicritères qui est valable pour l'administrateur, dans les détails de chaque formateur l'administrateur doit pouvoir lui affecter une séance de formation dans la même page sans oublier les RIB des comptes en banque sont ajoutés pour chaque formateur.

- **Gestion des services :**

Les formateurs ainsi que les états des sommes dues pour frais d'animation sont rattachés à un service à la TGR, pour une raison de traçabilité et d'organisation le système doit reconnaître à quel service appartient chacun d'eux.

Les services sont rattachés à des divisions et chaque division relève d'une trésorerie qui se situe elle-même dans une ville, ces trois derniers locaux doivent aussi être ajoutés, modifié ou supprimé en respectant les relations entre eux.

- **Gestion des formations :**

La gestion des formations s'articule sur l'ajout, la modification et la suppression des formations qui sont organisées selon :

***Des modules :***

Modules regroupant plusieurs thèmes (Sujets) de formation.

***Les thèmes (sujets) de formation :***

Sujet - titre- exacte de la formation, introduit lors de l'animation d'une séance dans un cycle de formation qui s'est déroulé dans une heure et une date précise (date début et date fin) à un endroit qui est pour la plupart du temps le siège la TGR - un module supplémentaire qui va pouvoir gérer les locaux de formations, est ajouté à l'application.

***Les cycles de formation :***

Un cycle de formation concerne un thème (sujet) de formation il est caractérisé aussi par la date de début, date de fin et le local (trésorerie) ou ses séances sont animées, il regroupe plusieurs séances de formation.

***Les séances de formation :***

Une séance de formation est animée par un formateur et destiné à des bénéficiaires, elle est planifiée dans une date précise ainsi qu'une heure de début et une heure de fin pour déduire la durée de cette séance, c'est à base de cette durée que le formateur va être rémunéré.

- **Gestion des bénéficiaires :**

Les bénéficiaires sont les personnels qui vont assister à une formation, la gestion des bénéficiaire est simple il suffit d'implémenter les opérations du CRUD et les relier à la séance de formation organisée par un certain formateur dans laquelle ils ont participés, ainsi que la présence qui est aussi une information importante doit être introduite au système.

- **Traitement des indemnités et grades :**

Les rémunérations (indemnités) sont attribuées aux formateurs pour chaque heure de formation animée en respectant certaines contraintes.

Les contraintes sont axées sur :

Le cout de la rémunération est variable en mesure du grade de l'animateur (formateur), un animateur ne sera rémunéré que sur 12 heures ou moins de formation par trimestre glissant - au cas où il dépasse 12 heures de formations dans un trimestre glissant, les heures supplémentaires ne seront pas incluses (ne seront pas calculées) -.

Ces rémunérations peuvent avoir des états soit envoyé ou non envoyé et comme on l'a déjà noté la rémunération est basée sur le grade du formateur, alors les grades doivent aussi pouvoir être ajouté, modifié ou supprimé.

- **Gestion des états :**

Une fois le formateur est sélectionné, son prime du trimestre courant est calculé au cas où il mérite, par la suite le système doit générer un état des sommes dues pour frais d'animation divisé en deux parties.

La première partie contient les informations de base du formateur, son RIB, le nombre d'heures de formation, le thème, lieu et la date de la formation en bas la somme calculée du prime en chiffre et en lettres avec la signature du service et la division chargée de l'opération.

La deuxième partie est un reçu destiné au formateur qui comporte comme informations la somme calculée en chiffres et lettres, le lieu et la date accompagnées de la signature du responsable.

Le système doit aussi permettre le remplissage d'un tableau contenant les états élaborés et envoyés, plus un autre contenant ceux qui ne sont pas encore traités.

De plus il y a des cas où celui qui est chargé de la mission de gestion (administrateur) devrait envoyer un bordereau contenant les noms de plusieurs formateurs pour les signer à la fois, l'application doit prendre en charge de regrouper ces noms dans un seul bordereau.

- **Sécurité de l'application :**

Chaque utilisateur de l'application doit être identifié par un login et un mot de passe pour s'authentifier au système afin qu'il puisse accéder à sa partie d'administration.

- **Statistiques et tableau de bord :**

Le système doit offrir des statistiques dans la page d'accueil à partir des informations enregistrées dans la base de données, sous forme d'un tableau de bord avec des diagrammes et chartes graphiques offrant une vue générale et simple sur les transactions hebdomadaires, mensuelles et annuelles.

## **Analyse du contexte :**

L'analyse du contexte a pour objectif l'identification des différentes entités qui interagiront avec le système.

### **Identification des acteurs :**

Un acteur peut consulter et/ou modifier directement l'état du système, par émission et/ou réception de messages éventuellement porteurs de données.

Les acteurs que j'ai identifiés à partir de la description des fonctionnalités ci-dessus sont les suivants :

#### **Le formateur :**

- Le formateur participe à l'animation des formations.
- Il reçoit sa rémunération.

#### **L'administrateur :**

- L'administrateur s'occupe de la gestion de l'application, il a tous les droits devant les autres acteurs.

#### **Le système :**

- Le système sera chargé de la génération des statistiques et les données du tableau de bord.
- Calculer la somme pour frais d'animation.
- La génération des états.

### **Interaction acteur – système et les messages échangés :**

Un message représente ici la spécification de haut niveau d'une interaction (communication) entre acteurs et système, avec l'intention (métier) de déclencher une activité chez le récepteur. La réception d'un message est normalement considérée comme un événement. Il faut se demander :

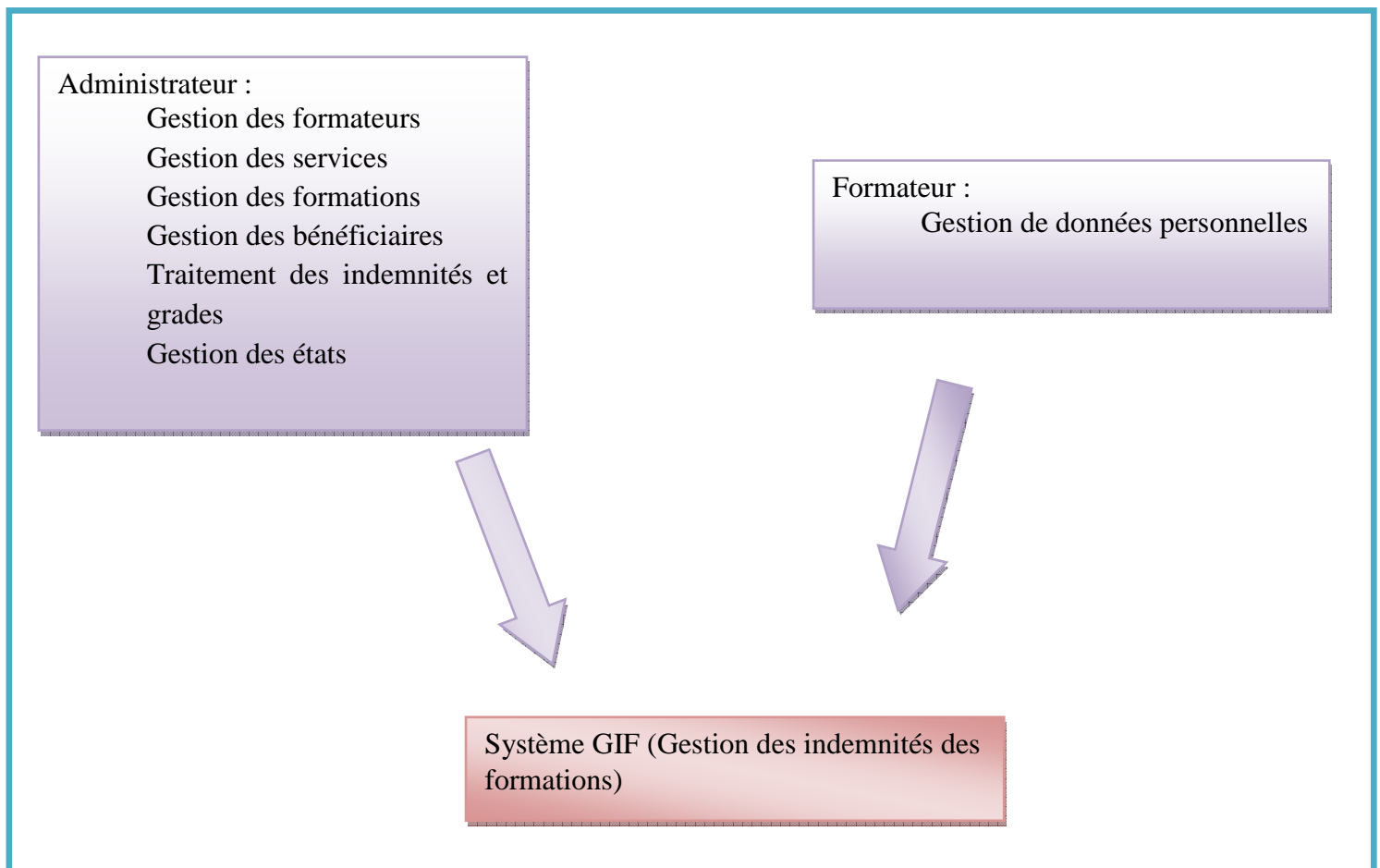
- Pour chaque acteur, quels sont les messages qui déclenchent un comportement attendu par l'acteur du système, dans le cadre de son activité.
- Pour le système, quels sont les messages émis à l'attention d'un acteur particulier et qui portent une information utilisée par ce destinataire.

L'identification des interactions avec le système consiste à l'identification des différents messages échangés par ces acteurs avec le système à développer.

Les messages que j'ai identifiés après l'analyse des rôles des acteurs listés ci-dessus sont résumés dans le tableau de la page suivante :

| NOM ACTEUR              | LE MESSAGE EMIS PAR LE SYSTEME   | LE MESSAGE REÇU PAR SYSTEME   |
|-------------------------|--|---|
| <b>L'administrateur</b> | <ul style="list-style-type: none"> <li>• <i>Statistiques et tableau de bord.</i></li> <li>• <i>calcule des rémunérations de formations.</i></li> <li>• <i>Génération des états.</i></li> </ul> | <i>Création, modification, suppression des formateurs, bénéficiaires, formations, services, grades, indemnités, trésoreries</i> |
| <b>Le formateur</b>     | <ul style="list-style-type: none"> <li>• <i>L'état de son prime (Envoyé ou non).</i></li> <li>• <i>Somme de ses indemnités.</i></li> </ul>   | <ul style="list-style-type: none"> <li>• <i>informations personnels.</i></li> </ul>   |

**Tableau 2 : Les messages échangés entre le système et les acteurs**



Les différentes entités composant l'environnement d'interactions du système, à savoir : les acteurs humains, les acteurs systèmes et les messages échangés entre eux.

Le chapitre suivant est consacré à la présentation de la phase de conception du système GIF, et donc je vais présenter la structure interne du système.

## Chapitre 4 : Conception du Système

---

**INTRODUCTION :** Ce chapitre traite la conception du système. Je vais présenter d'abord les diagrammes des cas d'utilisation réalisé à l'aide du formalisme UML, qui va aussi aider à maîtriser la fonctionnalité du système, il va aussi permettre de le décomposer en des modules qui respectent l'architecture technique que je compte implémenter.

Vu la taille du système à développer et la complexité de la TGR concernant le développement des applications, j'ai jugé très utile la décomposition du système GIF en des sous modules afin de bien maîtriser les aspects de modélisation et de faciliter l'implémentation en respectant les contraintes de l'architecture technique aussi pour livrer chaque partie développer le plus tôt possible.

## Formalisme UML :

Dans l'optique de concevoir une application qui doit être modulaire, facilement extensible et orienté objet, le formalisme UML s'est imposé comme l'outil le plus adéquat pour la modélisation de ce projet. En effet, UML va permettre de mener la phase d'analyse et de conception tout en bénéficiant de la puissance et de la simplicité de l'orienté objet. Rappelons-nous que la plate forme technique du projet GIF se base sur l'architecture JEE.

C'est pourquoi j'ai opté pour UML comme langage de modélisation car la notation UML est la plus appropriée pour des projets orientés objet. Ce choix peut être justifié également par plusieurs raisons :

- La notation UML facilite la compréhension et la communication d'une modélisation objet.
- UML est aujourd'hui un standard, adopté par les grands constructeurs de logiciel du marché.
- UML facilite la séparation et la décomposition.

Pour organiser le modèle de spécification fonctionnel d'une part et respecter la décomposition d'autre part, j'ai opté pour l'utilisation des packages, qui représentent les besoins d'une entreprise vis-à-vis d'un système informatique.

## Diagrammes de cas d'utilisation :

Le diagramme de cas d'utilisation décrit le comportement du système du point de vue utilisateur sous forme d'actions et de réactions. Il existe deux concepts fondamentaux dans la modélisation par les cas d'utilisation :

- Les acteurs qui agissent sur le système
- Les cas d'utilisations qui représentent les façons dont le système est manipulé par les acteurs.

Chaque cas d'utilisation indique une fonctionnalité du système déclenché par un acteur externe au système. Ce genre de diagramme permet de mettre en place et de comprendre les besoins des utilisateurs.

Les tableaux suivants regroupent tous les cas d'utilisations du système selon les acteurs.

|                              |  |
|------------------------------|--|
| <b>Nom cas d'utilisation</b> | Gestion des formateurs   |
| <b>Acteurs</b>               | Administrateur   |
| <b>Messages émis/reçus</b>   | Émis : ajout, édition, suppression des formateurs.<br>Reçu : existence/validation. |

|                              |   |
|------------------------------|---|
| <b>Nom cas d'utilisation</b> | Gestion des formations  |
| <b>Acteurs</b>               | Administrateur  |
| <b>Messages émis/reçus</b>   | Émis : ajout, édition, suppression des formations.<br>Reçu : existence/validation |

|                              |  |
|------------------------------|--|
| <b>Nom cas d'utilisation</b> | Gestion des bénéficiaires  |
| <b>Acteurs</b>               | Administrateur   |
| <b>Messages émis/reçus</b>   | Émis : ajout, édition, suppression d'un bénéficiaire.<br>Reçu : existence/validation |

|                              |  |
|------------------------------|--|
| <b>Nom cas d'utilisation</b> | Gestion des grades et indemnités   |
| <b>Acteurs</b>               | Administrateur   |
| <b>Messages émis/reçus</b>   | Émis : ajout, édition, suppression des grades et indemnités.<br>Reçu : existence/validation. |

|                              |  |
|------------------------------|--|
| <b>Nom cas d'utilisation</b> | Gestion des services, divisions, trésoreries et ville  |
| <b>Acteurs</b>               | Administrateur   |
| <b>Messages émis/reçus</b>   | Émis : ajout, édition, suppression des services, divisions, trésoreries et ville<br>Reçu : existence/validation. |

|                              |   |
|------------------------------|---|
| <b>Nom cas d'utilisation</b> | ajout/modification d'un formateur   |
| <b>Acteurs</b>               | Formateur   |
| <b>Message émis/reçu</b>     | Émis : ajout, édition des données d'un formateur.<br>Reçu : existence/validation. |

|                              |   |
|------------------------------|---|
| <b>Nom cas d'utilisation</b> | Statistiques                                      |
| <b>Acteurs</b>               | Système   |
| <b>Messages émis/reçus</b>   | Émis : Statistiques et tableau de bord.<br>Reçu : |

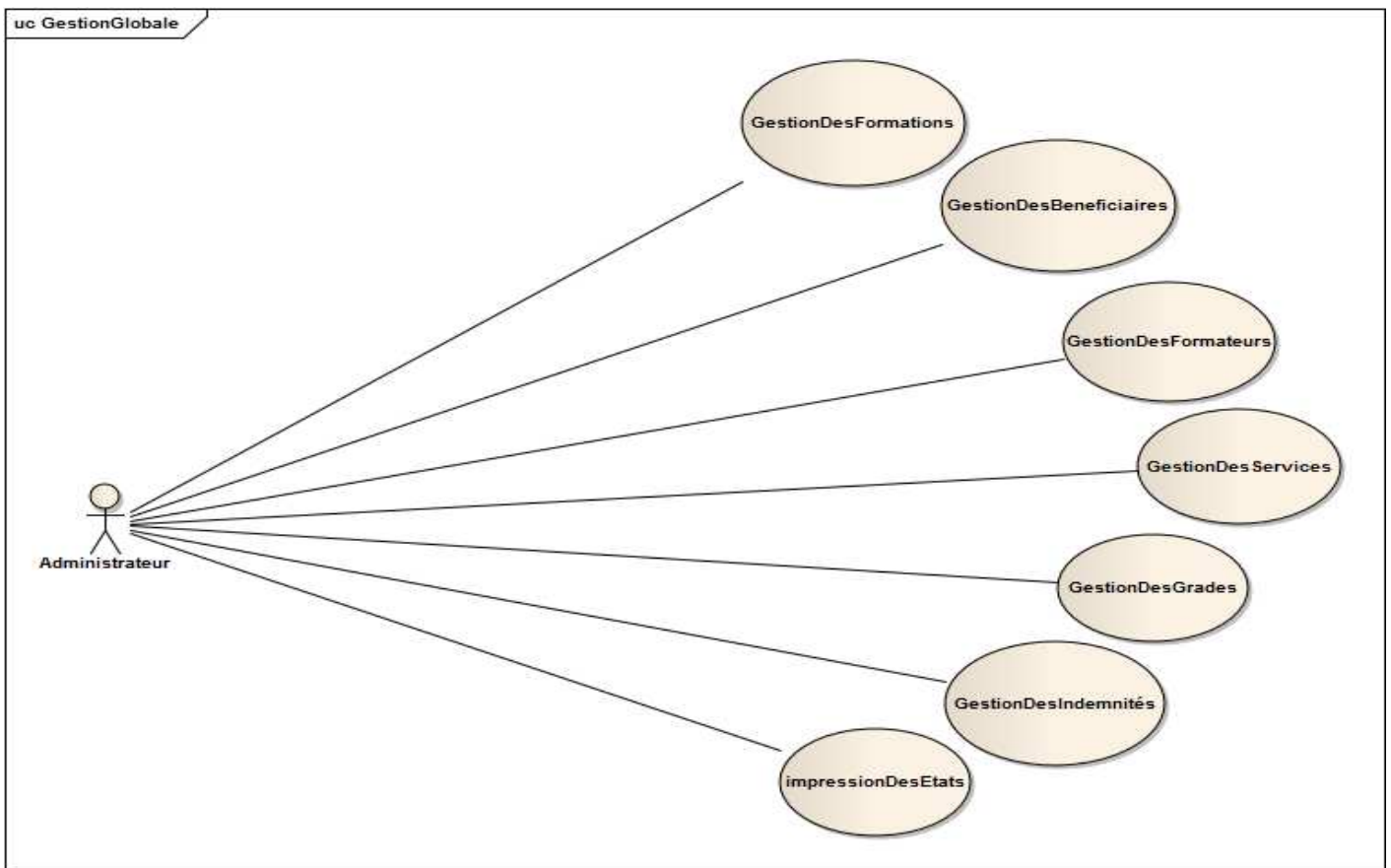
**Tableau 3 : les cas d'utilisations selon les acteurs**

D'après ce qui précède on obtient les diagrammes des cas d'utilisation importants suivante présentés divisés en blocs correspondants aux différents acteurs :

*a) Diagramme cas d'utilisation globale relatif à l'administrateur :*

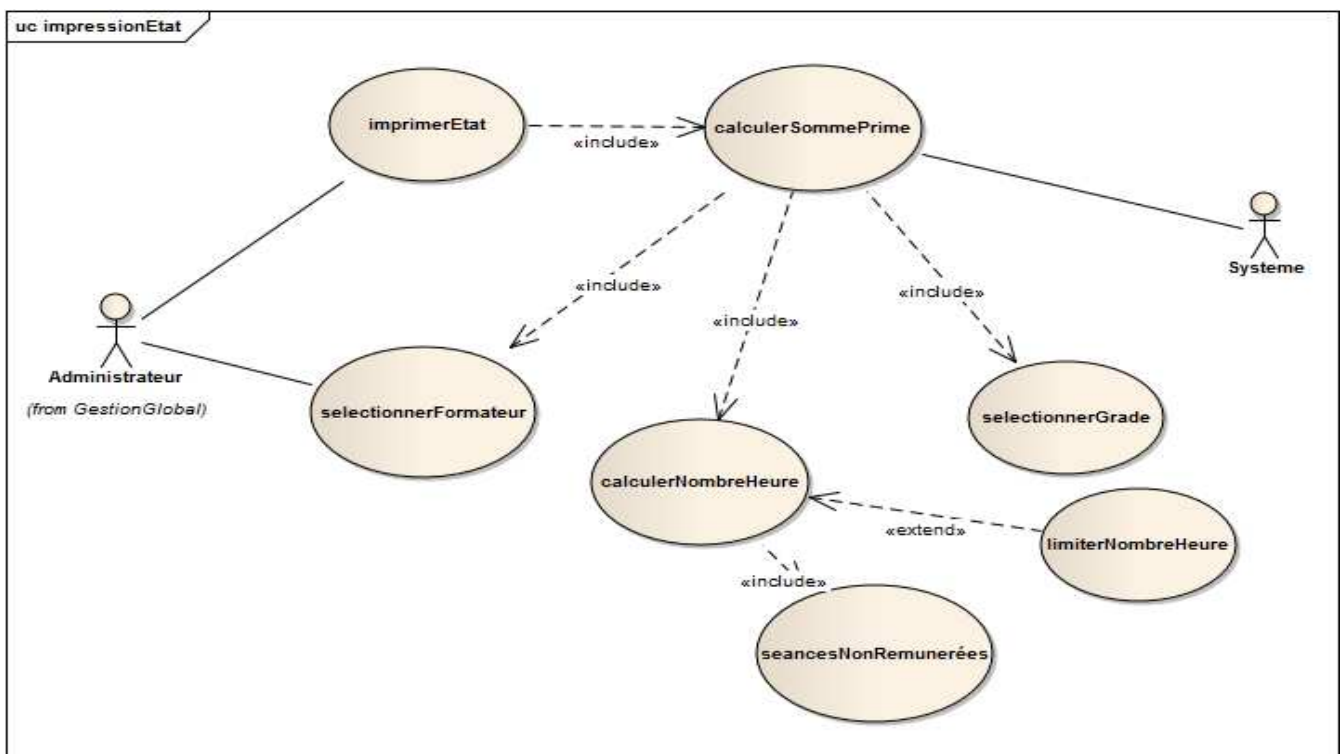
Ces cas d'utilisation englobe l'ensemble des opérations majeurs effectuées par l'administrateur, c'est l'axe de du système GIF.





**Figure 5 : cas d'utilisation Administrateur « gestion globale »**

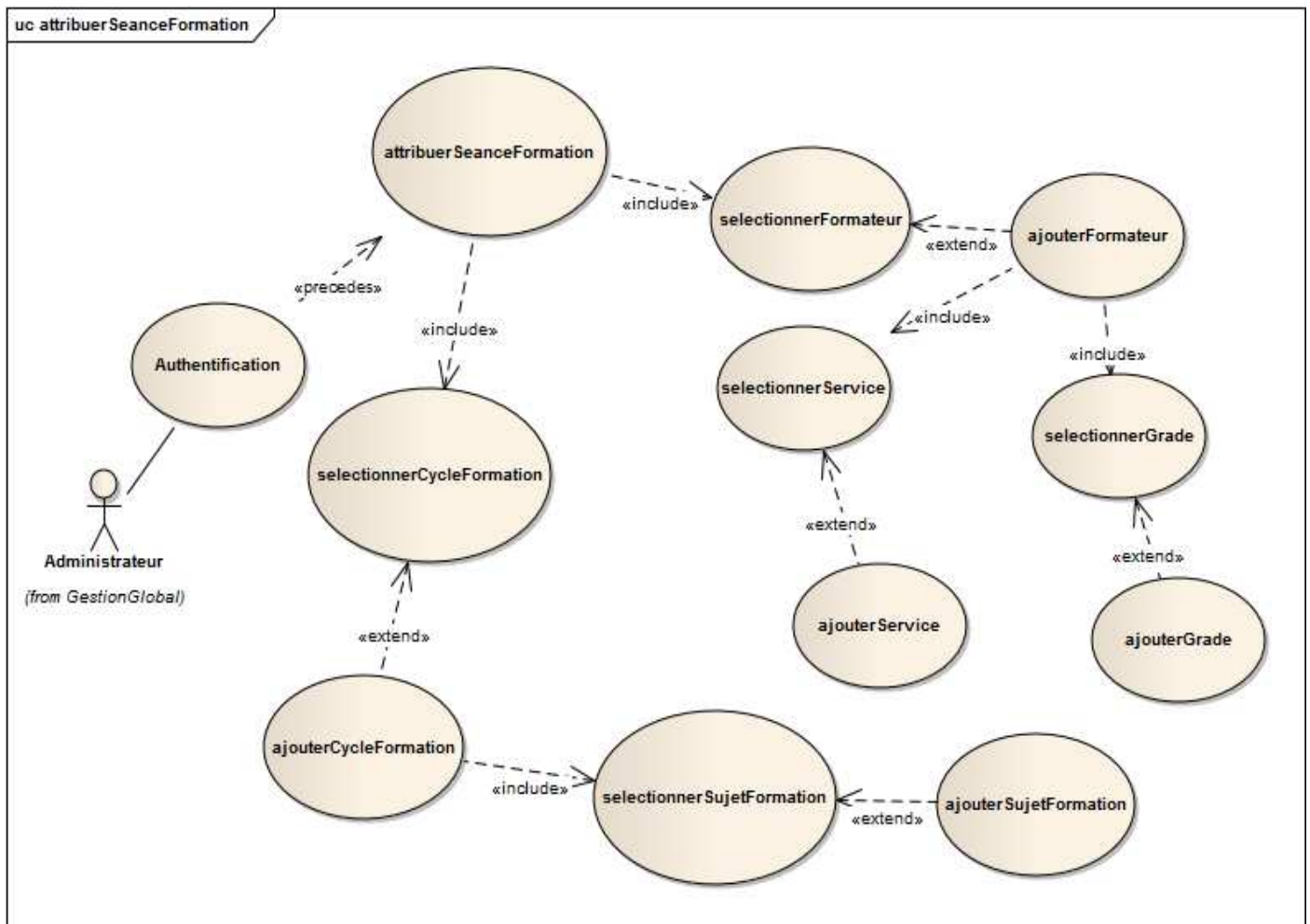
*b) Diagramme cas d'utilisation «imprimer état » relatif à l'administrateur :*



**Figure 6 : cas d'utilisation « imprimer état »**

## c) Diagramme cas d'utilisation, « attribuer séance de formation » :

Figure 7 : cas d'utilisation « attribuer séance de formation »



## Description de quelques cas d'utilisation :

Je vais décrire ci-dessous les deux derniers cas d'utilisation (importants) dans des tableaux: Le premier tableau concerne le cas d'utilisation « imprimer état », quant au deuxième, il donne une description du cas d'utilisation « attribuer séance de formation », chaque tableau illustre le nombre du cas d'utilisation, les acteurs impliqués, la description et le scénario du déroulement de cet cas d'utilisation, ces informations sont résumé à partir des descriptions préliminaires (description, scénario nominale, scénarios alternatifs..) des scénarios durant la phase capture des besoins du cahier des charges.

## a) Description du UC « imprimer état » :

|                   |                 |
|-------------------|-----------------|
| Cas d'utilisation | - Imprimer état |
|-------------------|-----------------|

|                    |   |
|--------------------|---|
| <b>Acteurs</b>     | - Administrateur, Système   |
| <b>Description</b> | - Permet à l'administrateur d'imprimer l'état des sommes dues pour frais d'animation au profit d'un formateur.  |
| <b>Scénarios</b>   | <ul style="list-style-type: none"> <li>• L'administrateur sélectionne le formateur à qui il veut imprimer l'état des sommes dues pour frais d'animation.</li> <li>• Le système récupère l'indemnité mérité à base du grade du formateur.</li> <li>• Le système calcule le nombre des heures animé durant les séances dont le formateur n'est pas encore payé du cycle de formation dans un trimestre glissant.</li> <li>• Après le système donne comme résultat le fichier à imprimer qui englobe toutes les informations nécessaires.</li> </ul> |

**Tableau 4 : description du cas d'utilisation « imprimer état »**

*b) Description du UC « attribuer séance de formation » :*

|                          |   |
|--------------------------|---|
| <b>Cas d'utilisation</b> | - Attribuer séance de formation   |
| <b>Acteurs</b>           | - Administrateur  |
| <b>Description</b>       | - Permet à l'administrateur d'attribuer une séance de formation à un formateur, après authentification.   |
| <b>Scénario</b>          | <ul style="list-style-type: none"> <li>• La première étape consiste sur la sélection du formateur à qui on va attribuer la séance de formation.</li> <li>• Au cas où le formateur ne fait pas encore partie du système on doit l'ajouter, l'ajout d'un formateur implique, plus la saisie de ses données personnelles, la sélection du grade et le service auquel il appartient</li> <li>• La deuxième étape est basée sur la sélection du cycle de formation, comme je l'ai déjà noté, un cycle de formation concerne</li> </ul> |

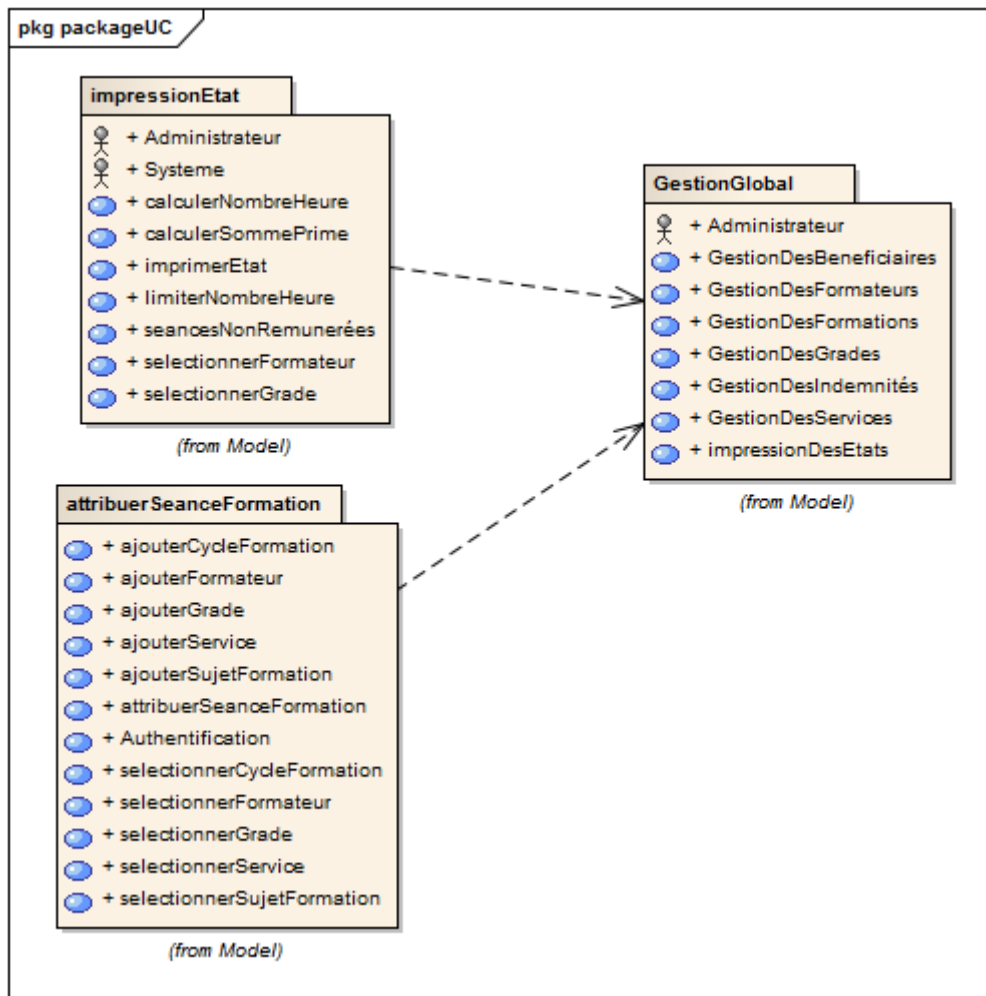
**Tableau 5 : description du cas d'utilisation « attribuer séance de formation »**

## Diagramme de packages :

Package (ou paquetage) : mécanisme général de regroupement d'éléments tels que classes, interfaces, mais aussi acteurs, cas d'utilisation, etc. les packages peuvent être imbriqués dans d'autres package.

La figure suivante illustre un exemple de diagramme de packages regroupant les cas d'utilisations du système GIF vues dernièrement.

*a) Diagramme de packages basé sur les cas d'utilisations :*



**Figure 8 : diagramme de packages « des cas d'utilisation »**

Comme on peut voir dans la figure ci-dessus, les packages « attribuer séance de formation » et « imprimer état » dépendent logiquement du package gestion global.

Dans le chapitre du diagramme de classe, on verra un exemple du diagramme de package concernant l'architecture logique du système basé sur les interfaces les classe du modèle MVC.

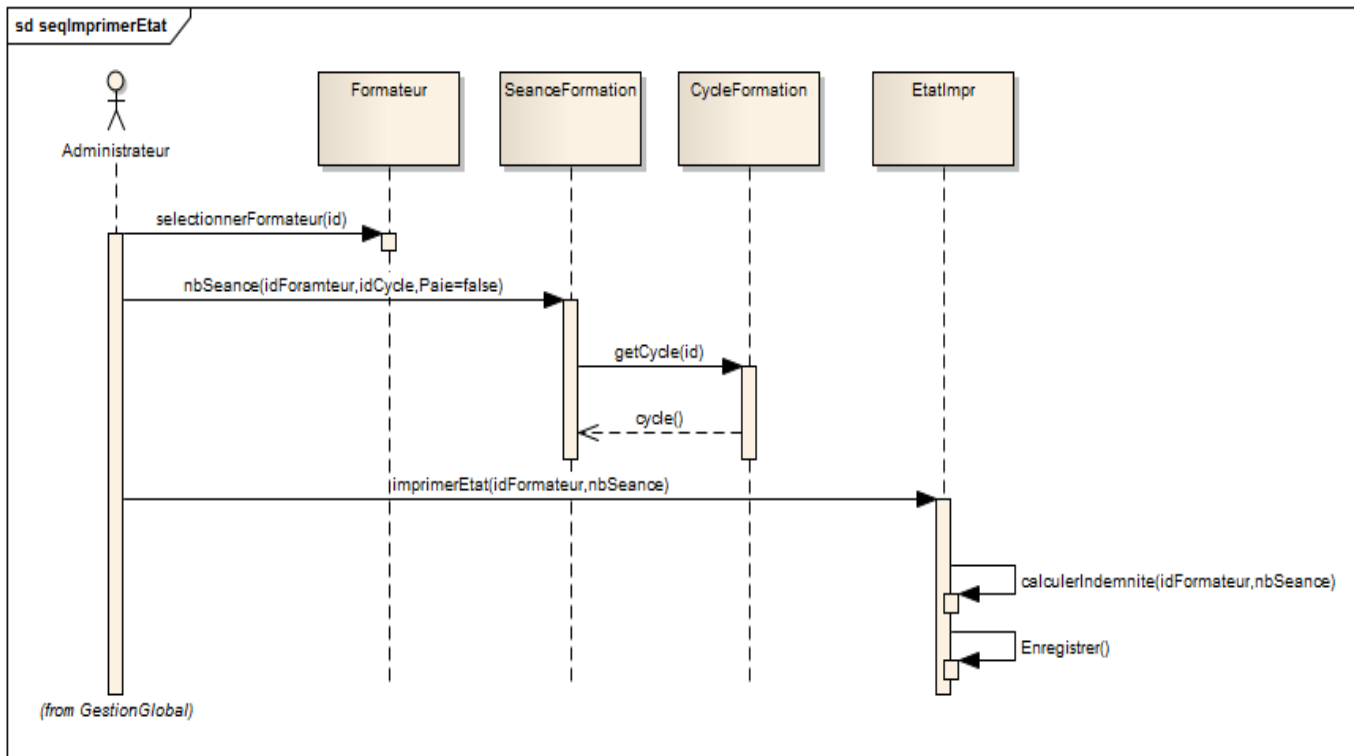
## Diagrammes de séquences :

Afin de bien illustrer les cas d'utilisations déjà élaborés, et dans le but de mieux représenter les interactions entre les objets de notre projet selon un point de vue temporel, j'ai implémenté les diagrammes de séquences suivants :

### *a) Scénario « imprimer état » :*

Avant l'impression, l'administrateur va choisir le formateur, après il demande au système de calculer la somme des heures non rémunérées des séances d'un cycle précis, de son rôle l'objet séance envoie un message au cycle de formation en lui demandant de retourner le cycle concerné pour valider, par la suite et à base du résultat retourné l'administrateur demande au système d'afficher le résultat finale, en fin ce dernier calcule la somme de la rémunération à base de l'indemnité mérité.

**Figure 9 : diagramme de séquence « imprimer état »**



*b) Scénario « attribuer séance de formation » :*

Sur ce diagramme je vais illustrer l'interaction entre les objets concernant l'attribution d'une séance de formation, supposons qu'on va commencer dès la partie d'ajout d'un cycle de formation, ensuite, l'administrateur choisi le formateur à qui il va attribuer une séance de formation de cycle de formation, ensuite il saisie les données concernant le planning de cette séance de formation et choisi le cycle.

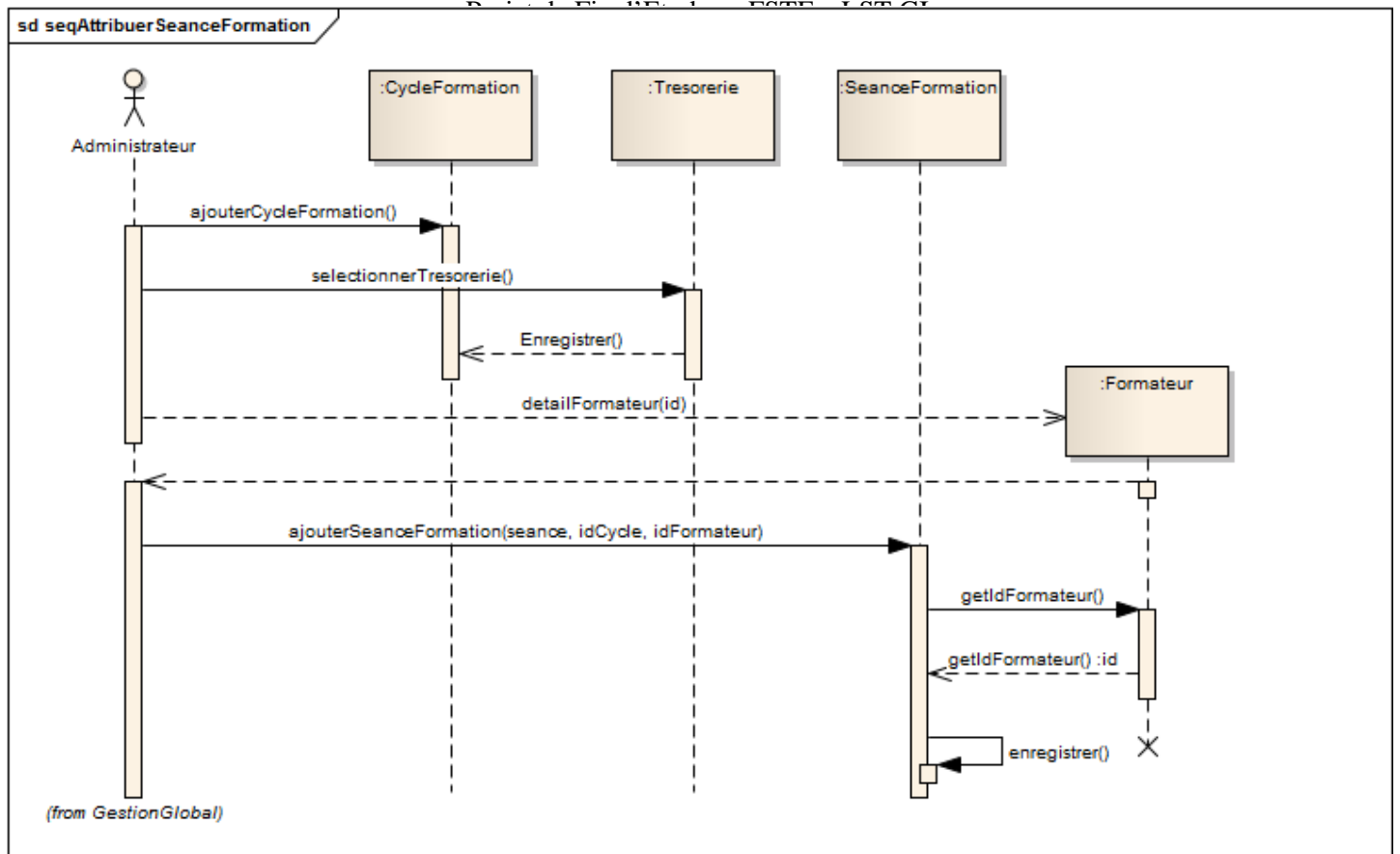


Figure 10 : Scénario « attribuer séance de formation »

c) Scénario, ajouter formateur selon le design pattern MVC à implémenter :

Selon l'architecture logique du système GIF qui suit la convention du design pattern MVC à base du Framework SPRING MVC, le diagramme de séquence suivant à titre d'exemple illustre cette implémentation :

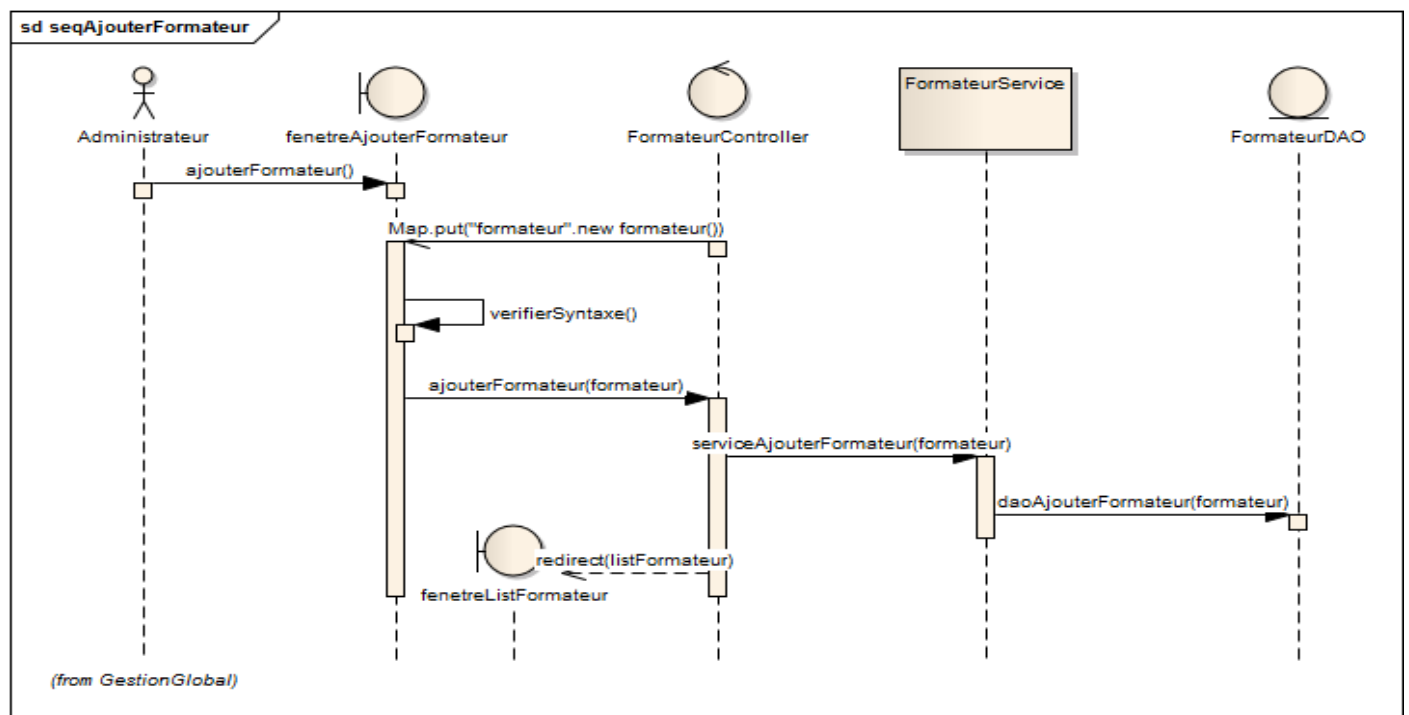


Figure 11 : scénario « ajouter formateur » selon le design pattern MVC

## **Diagramme de classes :**

Le diagramme de classes représente la structure statique du système en termes de classes et de relations. Je commence alors par présenter le diagramme de classes, puis je donnerai une description détaillée des différentes classes contenues dans ce diagramme.

## **Diagramme de classes et package UC du système GIF :**

Le diagramme que j'ai réalisé ci-dessous pourra être divisé en plusieurs sous-diagrammes pour le comprendre facilement et rapidement, en plus ses classes sont organisées en des packages de deux niveaux dans le code implémenté, j'ai donné un exemple ci-dessous après le diagramme de classe qui illustre la méthode que j'ai choisie, cet exemple concerne le package « formation » et les sous packages accompagnés « dao », « service », « entity » et « Controller » contenant les interfaces et leurs implémentations, selon l'architecture logique souvent utilisée et proposée par le Framework SPRING MVC3.

Dans ce cas les packages sont des espaces de noms, qui respectent des contraintes par exemple deux éléments ne peuvent pas porter le même nom au sein du même package. En revanche, deux éléments appartenant à des packages différents peuvent porter le même nom, et pour que ces éléments (packages) soient visibles aux autres packages on doit importer les uns dans les autres selon le besoin et la relation de dépendance entre eux.



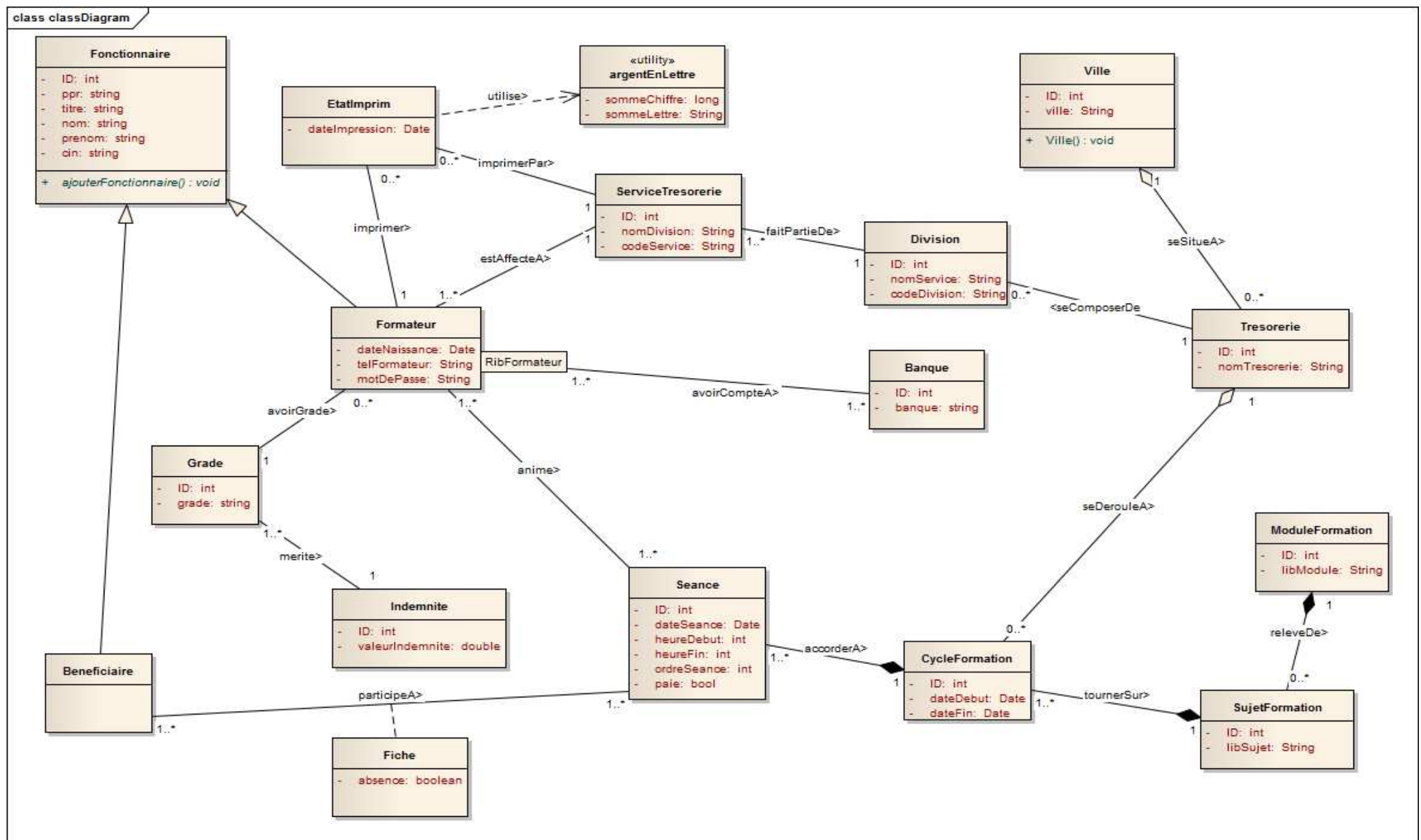
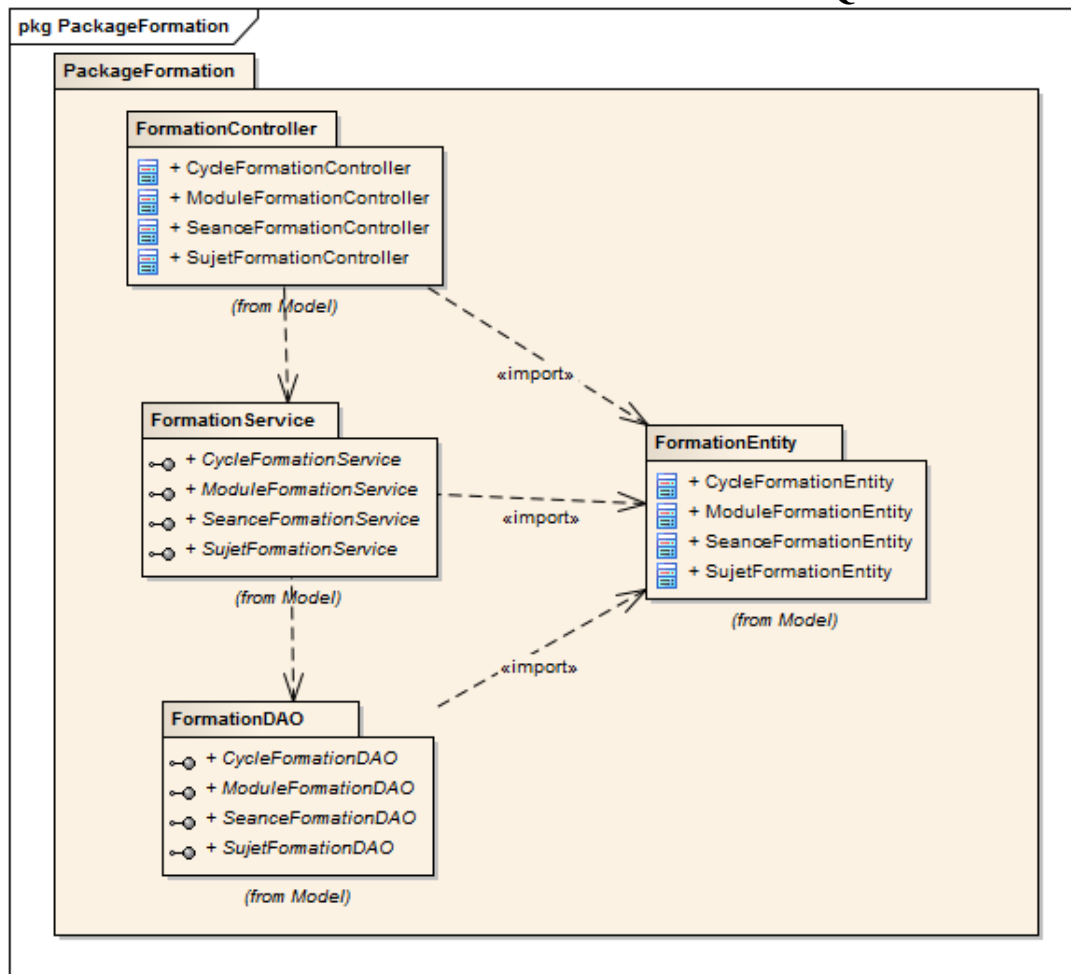


Figure 12 : diagramme de classes du système GIF



**UNIVERSITE SIDI MOHAMED BEN ABDELLAH  
FACULTE DES SCIENCES ET TECHNIQUES FES**



**Figure 13 : exemple diagramme de packages Formation (architecture logique)**

| Classe            | Description   |
|-------------------|---|
| Fonctionnaire     | Classe mère qui regroupe les attributs d'un formateur et bénéficiaire.                |
| Formateur         | Classe fille hérite de fonctionnaire qui représente le formateur.                     |
| Beneficiaire      | Classe fille hérite de fonctionnaire qui représente un bénéficiaire d'une formations. |
| EtatImpr          | Class qui représente l'Etat à un imprimer.  |
| ServiceTresorerie | Classe qui représente le service qui dérive d'une division.                           |
| Division          | Classe qui représente la division qui se trouve dans une trésorerie.                  |
| Grade             | Classe concerne le grade d'un formateur.  |
| Indemnité         | Classe qui relie l'indemnité mérité au grade du formateur.                            |
| Seance            | Classe qui représente la séance de formation à planifier d'un cycle.                  |
| CycleFormation    | Classe qui représente le cycle de formation qui concerne un sujet.                    |
| SujetFormation    | Classe du sujet de formation, représente le sujet à animer durant un cycle.           |
| ModuleFormation   | Classe qui contient les noms des modules qui regroupent les sujets de formation.      |
| Tresorerie        | Classe qui représente les trésoreries (lieu de déroulement des formations).           |
| Ville             | Classe qui représente une ville où se trouvent les trésoreries.                       |
| Banque            | Classe qui représente les banques où les formateurs disposent des                     |

**UNIVERSITE SIDI MOHAMED BEN ABDELLAH  
FACULTE DES SCIENCES ET TECHNIQUES FES  
DEPARTEMENT D'INFORMATIQUE**

|                |  |
|----------------|--|
|                | <i>comptes.</i>  |
| RibFormateur   | <i>Classe qui relie le formateur à sa banque et qui comporte l'attribue RIB (numéro de compte du formateur).</i> |
| ArgentEnLettre | <i>Classe « utility » qui joue le rôle de convertir les sommes d'argents à partir des chiffres en lettres.</i>   |

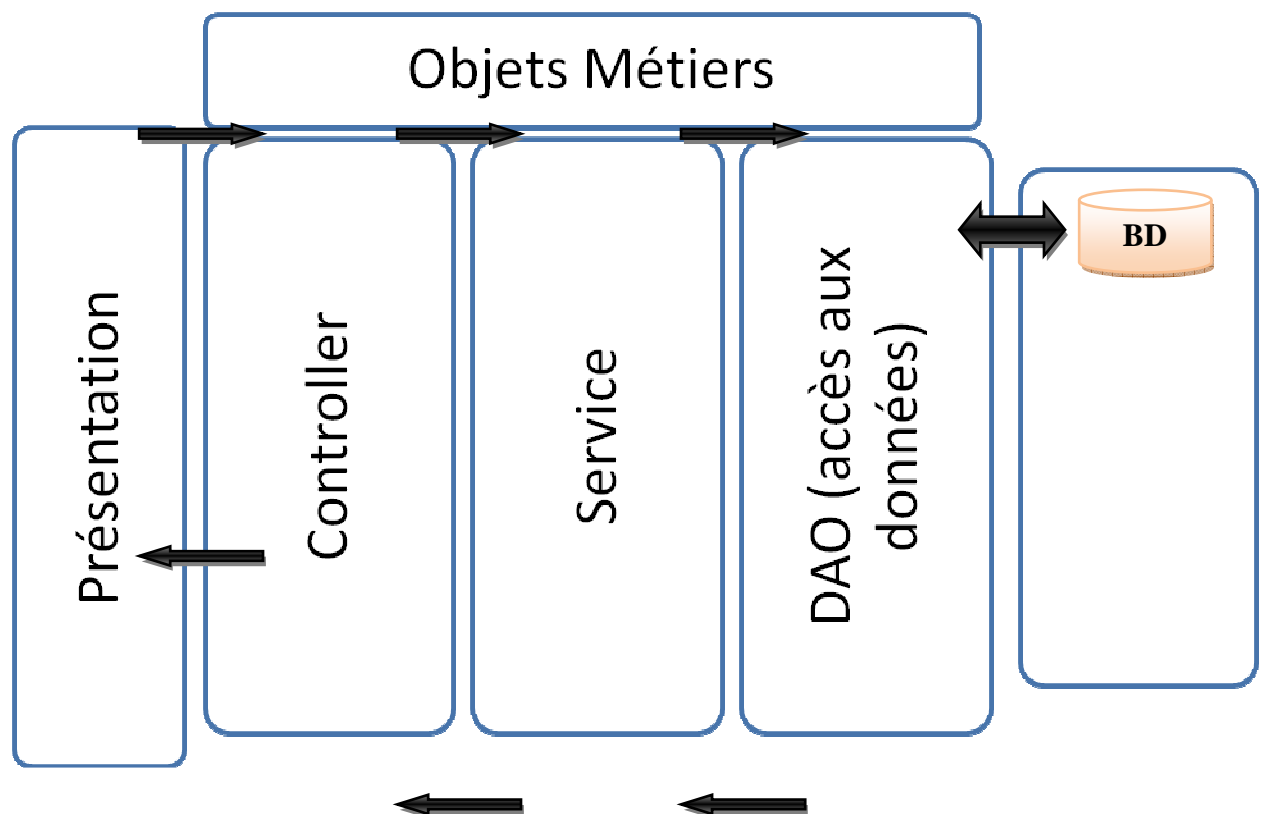
**Tableau 6 : description des classes**

Avant de terminer avec cette partie j'aimerais bien noter que, à partir de ces classes qu'on va extraire notre modèle relationnel de la base de données et je vais les mapper grâce aux technologies de Mapping Hibernate avec JPA à la base de données.

## Architecture Technique du Système :

Après avoir achevé la conception du système, j'ai en place l'architecture technique du système qui permet de le décomposer en plusieurs couches. Cette décomposition permet de maîtriser le développement du système.

L'architecture technique du système que j'ai mis en place est schématisée comme suit :



**Figure 14 : architecture technique du système**

**UNIVERSITE SIDI MOHAMED BEN ABDELLAH  
FACULTE DES SCIENCES ET TECHNIQUES FES  
DEPARTEMENT D'INFORMATIQUE**

Cette architecture comporte cinq couches, dont chacune à un rôle bien précis, je décris leurs utilités en commençant de droit à gauche :

**Couche BD** : couche base de données, assure le stockage des données manipulées par les utilisateurs dans ce cas c'est le SGBD qui intervient dans la manipulation.

**Couche DAO** : couche accès aux données, chacune des méthodes accéderont à la base de données, pour mettre à jour ou récupérer des enregistrements de la base de données.

**Couche Service** : chacune des méthodes de cette couche représentera un processus métier. L'implémentation de ces méthodes, appelleront les méthodes de la couche DAO.

**Couche Controller** : permet de lier la partie logique et la présentation il s'agit de la gestion des événements.

**Couche Présentation** : elle regroupe les interfaces de l'application du Système qui reçoivent les demandes des utilisateurs.

**CONCLUSION :**

Dans ce chapitre, nous avons exploré les différents diagrammes UML qui ont été élaboré à base de la phase analyse et spécification du système étudié, aussi, la mis en place de l'architecture technique du système GIF.

La conception et l'architecture technique dans ce chapitre est la base de la phase de construction du système, qui fera l'objet de la partie suivante.

*Rapport-gratuit.com*  
LE NUMERO 1 MONDIAL DU MÉMOIRES 

**PARTIE III : MISE EN ŒUVRE DU PROJET**

La dernière phase du cycle de développement adopté est la réalisation du système et son intégration.

Ainsi, cette partie est composée de deux chapitres : le premier passe en revue les différentes technologies et outils utilisés pour la réalisation.

Le deuxième chapitre présente l'architecture applicative du système suivie par l'explication du déroulement des différentes étapes de la phase réalisation. La dernière partie du chapitre est consacrée à la présentation des différentes interfaces permettant l'intégration du système.

## **Chapitre 5 : Technologies et outils utilisés**

---

J'ai réservé le présent chapitre pour la présentation de l'infrastructure à laquelle le projet GIF est basé et que je compte déployer, selon les axes suivants :  
Architecture, Technologies, Modèle et Outils

**INTRODUCTION :**  
L'infrastructure du projet GIF est basée sur l'architecture et les technologies Java EE. Ainsi les Framework et les design patterns open source supportés.

## **Technologie et Architecture Java EE :**

Le Java Enterprise Edition, comme son nom l'indique, a été créé pour le développement d'applications d'entreprises. Ses spécifications ont été pensées afin notamment de faciliter le travail en équipe sur un même projet : l'application est découpée en couches, et le serveur sur lequel tourne l'application est lui-même découpé en plusieurs niveaux. Pour faire simple, Java EE fournit un ensemble d'extensions au Java standard afin de faciliter la création d'applications centralisées.

On parle généralement de «plate-forme JEE» pour désigner l'ensemble constitué des services (API) offerts et de l'infrastructure d'exécution. JEE comprend notamment :

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

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

Dans la mesure où JEE s'appuie entièrement sur le Java, il bénéficie des avantages et inconvénients de ce langage, en particulier une bonne portabilité et une maintenabilité du code.

Les API de JEE peuvent se répartir en trois grandes catégories :

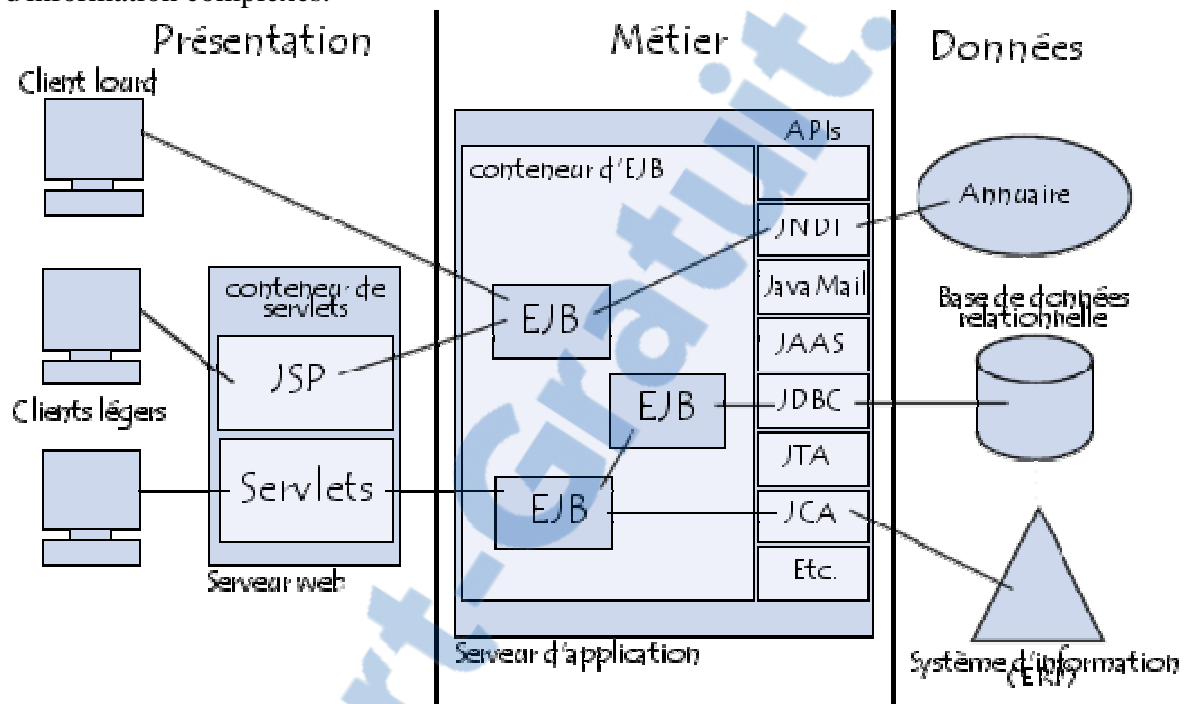
Les composants. On distingue habituellement deux familles de composants :

Les composants web : Servlets et JSP (Java Server Pages). Il s'agit de la partie chargée de l'interface avec l'utilisateur (on parle de logique de présentation).

**UNIVERSITE SIDI MOHAMED BEN ABDELLAH**  
**FACULTE DES SCIENCES ET TECHNIQUES FES**  
**DEPARTEMENT D'INFORMATIQUE**

Les composants métier : EJB (Enterprise Java Beans). Il s'agit de composants spécifiques chargés des traitements des données propres à un secteur d'activité (on parle de logique métier ou de logique applicative) et de l'interfaçage avec les bases de données.

L'architecture JEE permet ainsi de séparer la couche présentation, correspondant à l'interface homme-machine (IHM), la couche métier contenant l'essentiel des traitements de données en se basant dans la mesure du possible sur des API existantes, et enfin la couche de données correspondant aux informations de l'entreprise stockées dans des fichiers, dans des bases de données relationnelles ou XML, dans des annuaires d'entreprise ou encore dans des systèmes d'information complexes.



**Figure 15 : Architecture de la technologie JEE**

## JSP :

Le Java Server Pages ou JSP 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. Depuis la version 2.0 des spécifications, la syntaxe JSP est complètement conforme au standard XML.

La syntaxe du JSP ajoute des balises XML, appelées actions JSP, qui peuvent être utilisées pour appeler des fonctions. De plus, cette technique permet la création de bibliothèques de balises JSP (taglib) qui agissent comme des extensions au HTML ou au XML. Les bibliothèques de balises offrent une méthode indépendante de la plate-forme pour étendre les fonctionnalités d'un serveur HTTP. Il existe aussi un langage de script particulier, appelé Expression Language (EL) destiné à réduire l'injection de code java au sein des pages JSP ainsi qu'à étendre les possibilités des taglibs.

Les JSP sont compilées par un compilateur JSP pour devenir des servlets Java. Un compilateur JSP peut créer un servlet Java en code source Java qui peut à son tour être compilé par le compilateur Java, ou peut créer le pseudo-code Java interprétable directement.

**UNIVERSITE SIDI MOHAMED BEN ABDELLAH**  
**FACULTE DES SCIENCES ET TECHNIQUES FES**  
**DEPARTEMENT D'INFORMATIQUE**

Dans les deux cas, il est bon de comprendre comment le compilateur JSP transforme la page en servlet Java. Voir l'exemple de page JSP fourni en fin d'article, avec le servlet créé et la page HTML résultante.

## **JSTL :**

La JSTL est une implémentation de Sun qui décrit plusieurs actions basiques pour les applications web JEE. Elle propose ainsi un ensemble de bibliothèques de tags pour le développement de pages JSP.

De nombreux Frameworks facilitent le développement d'application JEE (Spring ou d'autres). La plupart de ces Frameworks proposent également des bibliothèques de tags JSP facilitant la création de pages JSP. Il en résulte une multitude de bibliothèques différentes pour des fonctionnalités similaires. La JSTL propose une bibliothèque standard pour la plupart des fonctionnalités de base d'une application JEE.

Alors, Le but de la JSTL est de simplifier le travail des auteurs de page JSP, c'est à dire la personne responsable de la couche présentation d'une application web JEE.

En effet, un web designer peut avoir des problèmes pour la conception de pages JSP du fait qu'il est confronté à un langage de script complexe qu'il ne maîtrise pas forcément.

La JSTL permet de développer des pages JSP en utilisant des balises XML, donc avec une syntaxe proche des langages utilisés par les web designers, et leur permet donc de concevoir des pages dynamiques complexes sans connaissances du langage Java.

Sun a donc proposé une spécification pour une bibliothèque de tags standards : la Java Standard Tag Library (JSTL). C'est à dire qu'il spécifie les bases de cette bibliothèque, mais qu'il laisse l'implémentation libre (de la même manière que pour les serveurs JEE qui sont des implémentations de la spécification JEE).

|  | <b>URI</b>  | <b>Préfixe</b> |
|--|---|----------------|
|  | <a href="http://java.sun.com/jsp/jstl/core">http://java.sun.com/jsp/jstl/core</a>           | c              |
|  | <a href="http://java.sun.com/jsp/jstl/fmt">http://java.sun.com/jsp/jstl/fmt</a>             | fmt            |
|  | <a href="http://java.sun.com/jsp/jstl/xml">http://java.sun.com/jsp/jstl/xml</a>             | x              |
|  | <a href="http://java.sun.com/jsp/jstl/functions">http://java.sun.com/jsp/jstl/functions</a> | fn             |

**Tableau 6 : les tags Library du jstl**

## **Hibernate Framework :**

Hibernate est une solution open source de type ORM (Object Relational Mapping) qui permet de faciliter le développement de la couche persistance d'une application. Hibernate permet donc de représenter une base de données en objets Java et vice versa.

il facilite la persistance et la recherche de données dans une base de données en réalisant lui même la création des objets et les traitements de remplissage de ceux-ci en accédant à la base de données. La quantité de code ainsi épargnée est très importante d'autant que ce code est généralement fastidieux et redondant.

De plus il est très populaire notamment à cause de ses bonnes performances et de son ouverture à de nombreuses bases de données.

**UNIVERSITE SIDI MOHAMED BEN ABDELLAH**  
**FACULTE DES SCIENCES ET TECHNIQUES FES**  
**DEPARTEMENT D'INFORMATIQUE**

La persistance est le mécanisme de sauvegarde (sur disque ou dans une base de données relationnelle) et de restauration des données. Un programme informatique peut alors être fermé sans que ses données ne soient perdues pour autant.

Dans le monde Java, l'API JDBC (Java DataBase Connectivity) a très vite répondu à ce besoin. Cependant, elle implique que le développeur implémente lui-même l'interaction avec les bases de données et les requêtes SQL qui sont nécessaires à la persistance des données de son application. De plus, la transcription des objets Java en données relationnelles dans la base de données doit être faite manuellement et est souvent fastidieuse.

- **La notion de Mapping relationnel - objet**

Cette problématique est résolue par la notion de Mapping relationnel-objet (également appelé ORM pour Object Relational Mapping), qui permet de rendre transparent le pont entre le monde orienté objet du Java et le monde relationnel des SGBD (Systèmes de Gestion de Base de Données). Le mapping relationnel-objet permet ainsi de faire automatiquement les transcriptions suivantes :

|  | Modèle base de données relationnel |
|--|------------------------------------|
|  | Table                              |
|  | Colonne                            |
|  | Enregistrement                     |
|  | Clé étrangère                      |

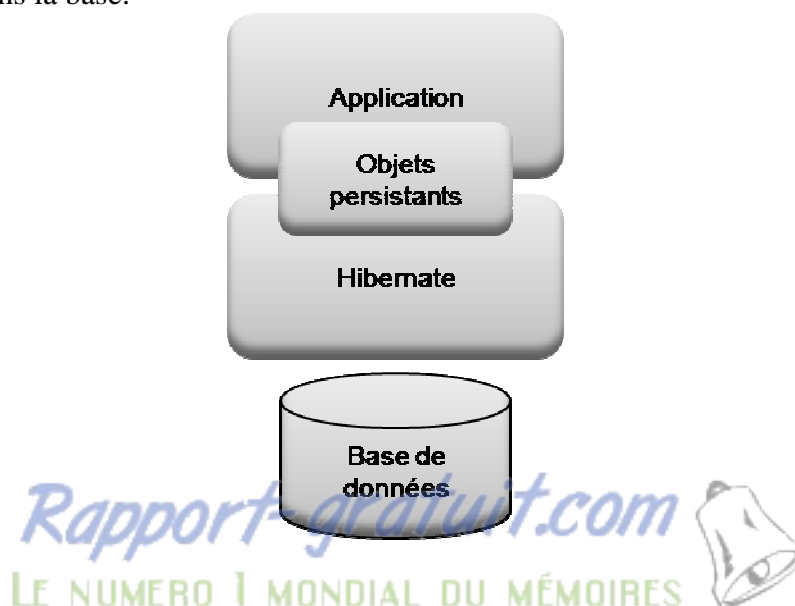
**Tableau 7 : correspondance du modèle objet au modèle BDR**

On parle alors de :

**Classe persistance** : classe dont les instances peuvent être rendues persistantes, c'est-à-dire enregistrées en base de données.

**Instance persistante** : instance d'une classe persistante dont les données sont présentes dans la base de données.

**Instance transiente ou volatile** : instance d'une classe persistante dont les données ne sont pas présentes dans la base.



**Figure 16 : l'emplacement de Hibernate entre le SGBD et l'Application**

## **JPA et Hibernate :**

En Java, nous disposons de la Java Persistence API (abrégée JPA), une interface de programmation qui constitue une couche au-dessus de JDBC. Elle permet de rendre transparente la couche JDBC et de faire le lien entre une base de données relationnelle et des objets Java. Ainsi, les accès directs à la base de données sont remplacés par des appels à des méthodes objet de haut niveau et l'application manipule uniquement des objets, s'affranchissant de l'aspect relationnel des bases de données.

Le framework Hibernate constitue l'une des principales implémentations de cette interface. Pour être exact, Hibernate a précédé JPA et a fortement inspiré l'API. Pour utiliser Hibernate dans un projet Java, nous pouvons utiliser, au choix, des fichiers XML ou des annotations Java directement intégrées dans le code. A la différence du mapping par XML, tous les attributs sont mappés par défaut par annotations.

## **Design pattern MVC :**

Le design pattern Modèle-Vue-Contrôleur (MVC) est un pattern architectural qui sépare les données (le modèle), l'interface homme-machine (la vue) et la logique de contrôle (le contrôleur).

Ce modèle de conception impose donc une séparation en 3 couches :

**Le modèle :** Il représente les données de l'application. Il définit aussi l'interaction avec la base de données et le traitement de ces données.

**La vue :** Elle représente l'interface utilisateur, ce avec quoi il interagit. Elle n'effectue aucun traitement, elle se contente simplement d'afficher les données que lui fournit le modèle. Il peut tout à fait y avoir plusieurs vues qui présentent les données d'un même modèle.

**Le contrôleur :** Il gère l'interface entre le modèle et le client. Il va interpréter la requête de ce dernier pour lui envoyer la vue correspondante. Il effectue la synchronisation entre le modèle et les vues.

La synchronisation entre la vue et le modèle se passe avec le pattern Observer. Il permet de générer des événements lors d'une modification du modèle et d'indiquer à la vue qu'il faut se mettre à jour.

Ce modèle de conception permet principalement 2 choses :

Le changement d'une couche sans altérer les autres. C'est-à-dire que comme toutes les couches sont clairement séparées, on doit pouvoir en changer une pour, par exemple, remplacer Swing par SWT sans porter atteinte aux autres couches. On pourrait aussi donc changer le modèle sans toucher à la vue et au contrôleur. Cela rend les modifications plus simples.

La synchronisation des vues. Avec ce design pattern, toutes les vues qui montrent la même chose sont synchronisées.

Il faut tout de même garder en mémoire, que la mise en œuvre de MVC dans une application n'est pas des plus simples. En effet, ce modèle de conception introduit tout de même un



**UNIVERSITE SIDI MOHAMED BEN ABDELLAH  
FACULTE DES SCIENCES ET TECHNIQUES FES  
DEPARTEMENT D'INFORMATIQUE**

niveau de complexité assez élevé. De plus, implémenter MVC dans votre application nécessite une bonne conception dès le départ. Ce qui peut prendre du temps. Ce pattern n'est donc à conseiller que pour les moyennes et grandes applications.

## **Spring Framework :**

Spring est un framework créé et supporté par l'entreprise SpringSource (qui fait maintenant partie de VMware). Il permet de simplifier le développement d'applications Java et est devenu un standard dans l'industrie du développement logiciel basé sur la plateforme Java, surtout dans le développement Java EE.

Spring est un conteneur léger qui facilite le développement avec des POJO (Plain Old Java Object), c'est-à-dire avec des classes Java qui n'ont pas besoin de s'exécuter dans un conteneur spécifique ou dans un serveur d'applications. Il se compose d'un noyau (core) et de plusieurs modules qui ajoutent des fonctionnalités.

SPRING est effectivement un conteneur dit « léger », c'est-à-dire une infrastructure similaire à un serveur d'application JEE. Il prend donc en charge la création d'objets et la mise en relation d'objets par l'intermédiaire d'un fichier de configuration qui décrit les objets à fabriquer et les relations de dépendances entre ces objets.

## **Spring MVC :**

En particulier le module Spring MVC permet d'implémenter naturellement un Modèle Vue Contrôleur. L'utilisation des fichiers de configuration XML en font une remarquable implémentation facilement maintenable et évolutive.

Dans Spring MVC, les contrôleurs sont chargés d'appeler les différentes méthodes nécessaires de la classe métier et de transmettre à la vue le modèle contenant toutes les données indispensables à génération de celle-ci. A partir des fichiers de configuration, les requêtes http sont redirigées par un dispatcher vers le contrôleur adéquat. Il existe différents types de contrôleurs qui permettent de rediriger les requêtes selon différents critères :

- Adresse URL complète
- Paramètres passés dans l'URL
- Utilisateur connecté à l'application

De plus, Spring MVC offre la possibilité de développer ses propres types de contrôleur afin de pouvoir rediriger des adresses Url plus complexes. Enfin, Spring MVC autorise l'utilisation des expressions régulières pour la redirection des URL.

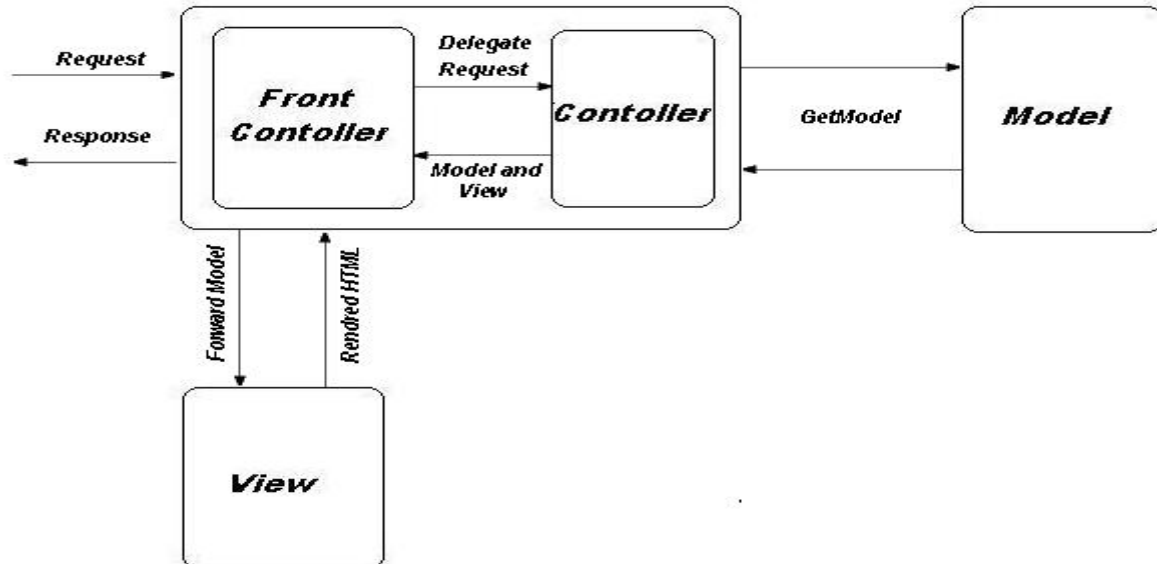
Le passage du modèle à la vue se fait par l'intermédiaire d'une map d'Object triée par mot clé. La vue y accède en utilisant le mot clé de l'Object souhaité.

Pour finir, à la différence de l'implémentation native de Tomcat avec les servlets, le contrôleur va faire appel au nom d'une vue et non plus directement à une page Jsp. L'appel à la vue sera traité par un résolveur de nom (configurable par XML) qui va se charger de renvoyer la bonne vue, celle-ci n'étant pas forcément une page JSP. En effet, Spring MVC permet l'utilisation de différents formats pour les vues : JSP, PDF, WORD, Excel, etc.

**UNIVERSITE SIDI MOHAMED BEN ABDELLAH  
FACULTE DES SCIENCES ET TECHNIQUES FES  
DEPARTEMENT D'INFORMATIQUE**

**Principe de fonctionnement :**

Comme tout autre MVC Framework, Spring MVC se base sur le principe décrit par le schéma ci-dessous :

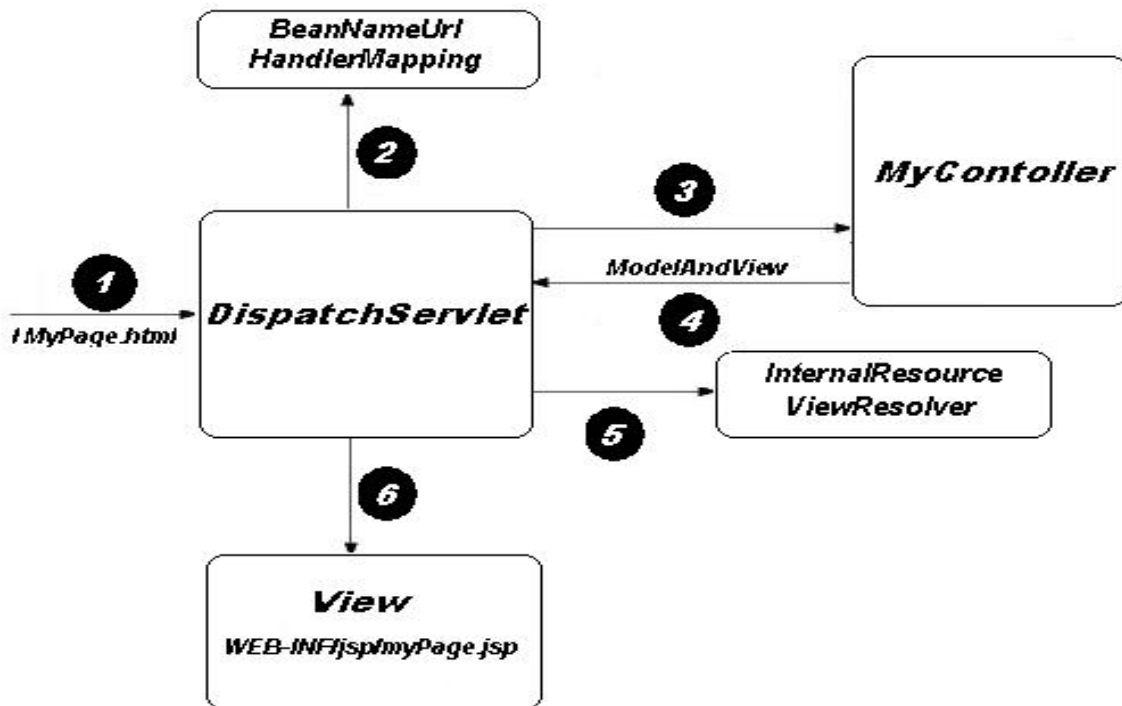


**Figure 17 : principe de base du MVC**

Passant maintenant aux détails : ci-dessous les étapes et la cinématique de la gestion d'une requête par Spring MVC.

- Le `DispatcherServlet` reçoit une requête dont l'URI-pattern est `'/myPage.html'`.
- Le `DispatcherServlet` consulte son `Handler Mapping` (Exemple : `BeanNameUrlHandlerMapping`) pour connaître le contrôleur dont le nom de bean est `'/myPage.html'`.
- Le `DispatcherServlet` dispatche la requête au contrôleur identifié (Exemple : `MyController`)
- Le contrôleur retourne au `DispatcherServlet` un objet de type `ModelAndView` possédant comme paramètre au minimum le nom logique de la vue à renvoyer.
- Le `DispatcherServlet` consulte son `View Resolver` lui permettant de trouver la vue dont le nom logique est `'myPage'`. Ici le type de `View Resolver` choisit est `InternalResourceViewResolver`.
- Le `DispatcherServlet` forwarde la requête à la vue associé. Ici la page `/WEB-INF/jsp/myPage.jsp`

**UNIVERSITE SIDI MOHAMED BEN ABDELLAH  
FACULTE DES SCIENCES ET TECHNIQUES FES  
DEPARTEMENT D'INFORMATIQUE**



**Figure 18 : principe de fonctionnement MVC avec Spring MVC**

```

<servlet>
  <servlet-name>spring</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-
class>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>spring</servlet-name>
  <url-pattern>/</url-pattern>
</servlet-mapping>
  
```

```

<bean id="viewResolver"
  class="org.springframework.web.servlet.view.UrlBasedViewResolver">
  <property name="viewClass"
    value="org.springframework.web.servlet.view.JstlView" />
  <property name="prefix" value="/WEB-INF/jsp/" />
  <property name="suffix" value=".jsp" />
</bean>
  
```

Les espaces doivent être bien respectés entre les balises XML des fichiers de configuration spring et JEE.

## Eclipse IDE :

**UNIVERSITE SIDI MOHAMED BEN ABDELLAH**  
**FACULTE DES SCIENCES ET TECHNIQUES FES**  
**DEPARTEMENT D'INFORMATIQUE**

Eclipse est un projet de la Fondation Eclipse visant à développer tout un environnement de développement libre, extensible, universel et polyvalent.

Son objectif est de produire et fournir divers outils gravitant autour de la réalisation de logiciel, englobant les activités de codage logiciel proprement dites (avec notamment un environnement de développement intégré) mais aussi de modélisation, de conception, de test, de reporting, etc. Son environnement de développement notamment vise à la généricité pour lui permettre de supporter n'importe quel langage de programmation.

Le projet Eclipse est pour cela organisé en un ensemble cohérent de projets logiciels distincts, sa spécificité tenant à son architecture totalement développée autour de la notion de plugin (en conformité avec la norme OSGi) : toutes les fonctionnalités de l'atelier logiciel doivent être développées en tant que plug-in bâti autour de l'IDE Eclipse Platform.

Eclipse recouvre donc notamment également à cet effet tout un framework de développement logiciel fournissant des briques logicielles à partir desquelles développer tous ces outils. C'est la raison pour laquelle Eclipse est présenté dans la littérature tout autant comme un EDI ou comme un Framework.

Il est possible de développer en JEE avec Eclipse, en partant par exemple du package "Eclipse for JavaEE Developers" ou alors, en installant WebTools.

- Ainsi, on a la possibilité d'utiliser les outils suivants :
- Assistants pour la génération d'applications Web (WAR) ou EAR.
- Assistants pour la création de Servlets, Filters, EJBs, etc.
- Editeur de fichiers HTML, JSP, CSS et JS.
- Support natif des serveurs les plus populaires (Tomcat, JBoss, WebSphere, JOnAs, etc.)
- Support de JSF
- Support de JPA.

## **Oracle:**

Oracle est un système de gestion de base de données relationnel (SGBDR) fourni par Oracle corporation. ORACLE Express Edition la version allégée gratuite d'Oracle. Il contient les principales fonctionnalités d'un SGBD ORACLE. Ce qui rend ce produit très intéressant pour les tests ou les petits projets.

Points forts :

- Moteur de Bases de données ORACLE.
- Produit gratuit.
- Interface d'administration web intégrée.

## **Le serveur apache Tomcat :**

Pour faire fonctionner une application web Java EE, nous avons besoin de mettre en place un serveur d'applications. Il en existe beaucoup sur le marché : j'ai choisi d'utiliser Tomcat, car c'est un serveur léger, gratuit, libre, multiplateforme et assez complet pour ce que nous allons

**UNIVERSITE SIDI MOHAMED BEN ABDELLAH  
FACULTE DES SCIENCES ET TECHNIQUES FES  
DEPARTEMENT D'INFORMATIQUE**

aborder. On le rencontre d'ailleurs très souvent dans des projets en entreprise, en phase de développement comme en production.

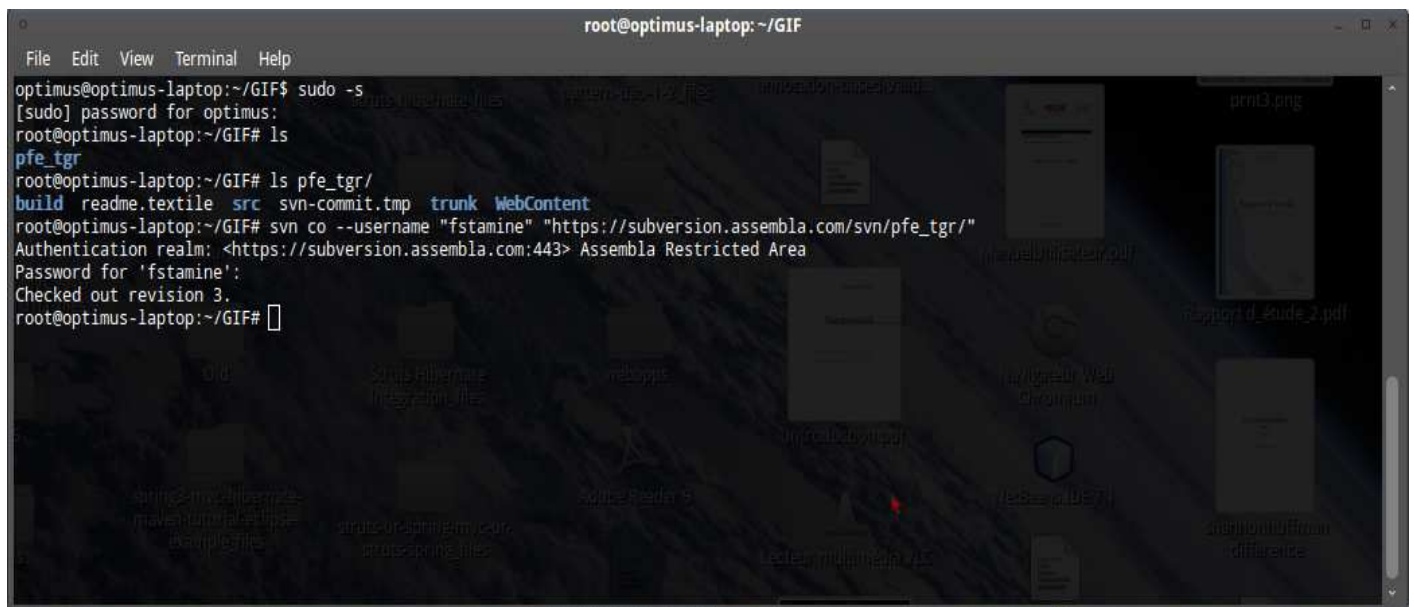
Pour information, Tomcat tire sa légèreté du fait qu'il n'est en réalité que l'assemblage d'un serveur web (gestion des requêtes/réponses HTTP) et d'un conteneur web. Mais il ne respecte pas entièrement les spécifications JAVA EE et ne supporte pas toutes ses technologies.

## Subversion :

Subversion (en abrégé svn) est un logiciel de gestion de versions, distribué sous licence Apache et BSD. Il a été conçu pour remplacer CVS. Ses auteurs s'appuient volontairement sur les mêmes concepts (notamment sur le principe du dépôt centralisé et unique) et considèrent que le modèle de CVS est le bon, et que seule son implémentation est en cause.

Je l'ai utilisé avec l'espace gratuit assembla qui permet de créer des repositories avec svn.

**Figure 21 : exemple « commit » du projet GIF sur le serveur svn de assembla**



## Jasper Report Framework :

J'ai choisi le Framework Jasper Reports pour le développement d'éditions. Jasper Reports est un Framework Java Open source pour l'édition. Il a la possibilité de produire un contenu riche à travers l'écran, l'imprimante ou à travers les fichiers PDF, HTML, XLS, CSV et XML. Cet outil permet :

- le formatage dynamique des éléments du document.
- la programmation en java.
- l'accès dynamique à la base de données.
- l'utilisation des fichiers de style pour la présentation.
- la possibilité d'utiliser les couleurs.
- l'insertion des images dynamiquement.
- la séparation du modèle de développement (la vue et les données).

Pour le modèle du fichier à générer JRXML je l'ai réalisé avec iReport.

**UNIVERSITE SIDI MOHAMED BEN ABDELLAH  
FACULTE DES SCIENCES ET TECHNIQUES FES  
DEPARTEMENT D'INFORMATIQUE**

**CONCLUSION :**

Au cours de ce chapitre, j'ai présenté les différentes technologies et outils que j'ai mis en œuvre pour la réalisation du système GIF.

Dans le chapitre suivant, je vais présenter l'architecture applicative du système, ainsi que l'implémentation de chacune des couches de cette architecture tout en utilisant les différents outils présentés ci-dessus et je vais terminer avec des prises d'écrans de la partie réalisation.

## Chapitre 6 : Mise en œuvre de la solution GIF

### INTRODUCTION

La mise en place de l'architecture GIF. dans le chapitre précédent, m'a permis de mettre en place l'architecture applicative du système GIF.

Ce chapitre a pour but la description de la phase mise en œuvre de la solution. Je vais présenter d'abord l'architecture applicative du système et je vais détailler ensuite la dernière phase du cycle de développement suivi : la réalisation et l'intégration représenté par des prises d'écrans de la solution

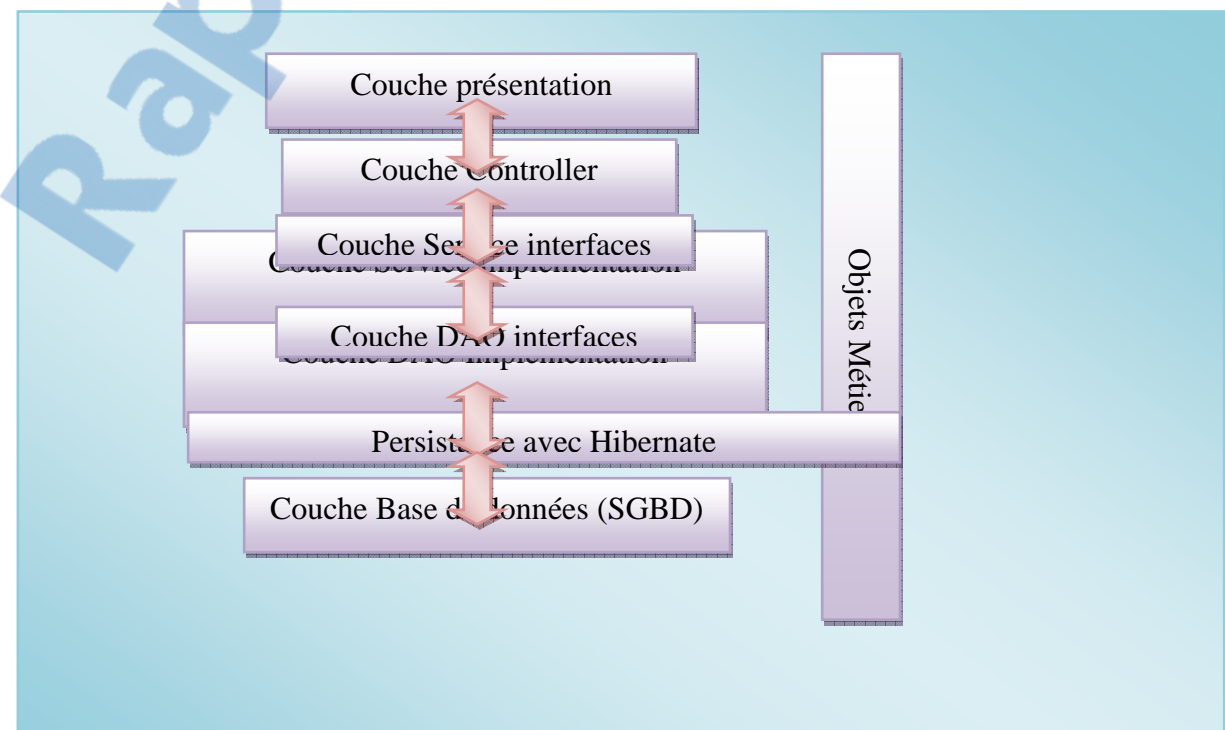
### Architecture applicative :

L'architecture applicative comporte les différentes couches applicatives du système, et associe à chaque couche l'outil et la technologie de réalisation adéquats.

Afin de bien maîtriser le développement du système, l'architecture applicative du système GIF a été décomposée en cinq couches :

- **Couche présentation**
- **Couche contrôleur**
- **Couche service**
- **Couche DAO**
- **Couche base de données**

Le schéma suivant montre l'architecture finale du système :



## **Réalisation des couches applicatives :**

### **La couche base de données :**

Cette couche comporte l'implémentation physique de la base de données, cette implémentation a été réalisée grâce à l'outil gratuit de Oracle « sql developer », d'ailleurs le schéma de la base de données a été extrait du diagramme de classes UML que j'ai élaboré lors de la phase conception.

### **La couche DAO :**

Cette couche permet l'accès aux données, pour être bien précise elle a pour but de transformer les données contenues dans une base de données en objets et inversement. La représentation du modèle de données objet en base de données (appelée persistance) est effectué à l'aide du Framework Hibernate et les classe métier annoté avec le JPA.

### **La couche service :**

La couche Service ajoute les règles et la logique métier de l'application. Elle fait appel à la couche DAO pour manipuler les données, c'est le Spring Framework qui prends en charge cette liaison mais après avoir implémenté les interfaces de couche service.

### **La couche Controller :**

Permet de lier la partie logique et la présentation il s'agit de la gestion des événements. Ici le contrôleur reçoit les requêtes à partir des liens saisies dans le navigateur (couche présentation), ensuite il décide quoi faire dans la méthode publique qui permet de filtrer la demande, aussi Spring intervient ici dans cette couche.

### **La couche Présentation :**

Cette couche représente les interfaces web développé à l'aide de HTML, JSP, JSTL et le Framework Spring surtout dans la partie du formulaire, tout ces bibliothèque de tags permet en particulier une certaine réutilisabilité, flexibilité et séparation de la présentation des données, ces interfaces reçoivent les demandes des utilisateurs.

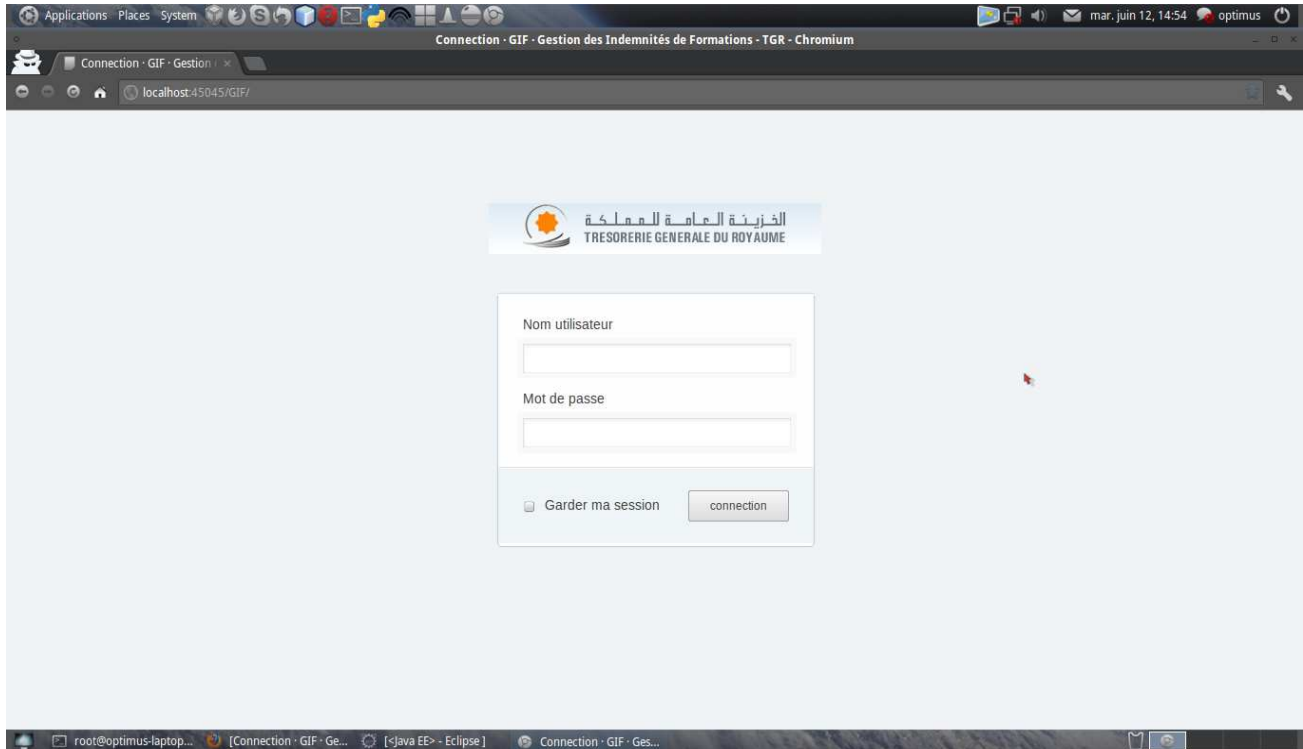
## **Les écrans de la réalisation :**

### **Page de démarrage :**



**UNIVERSITE SIDI MOHAMED BEN ABDELLAH  
FACULTE DES SCIENCES ET TECHNIQUES FES  
DEPARTEMENT D'INFORMATIQUE**

La figure ci-dessous présente la page d'accueil du système sous forme d'une page d'authentification puis que l'accès au système va être restreint et limité à la personne concernée de la gestion. Cette interface permet alors à l'utilisateur de saisir son login et son mot de passe pour se connecter et procéder à sa gestion.

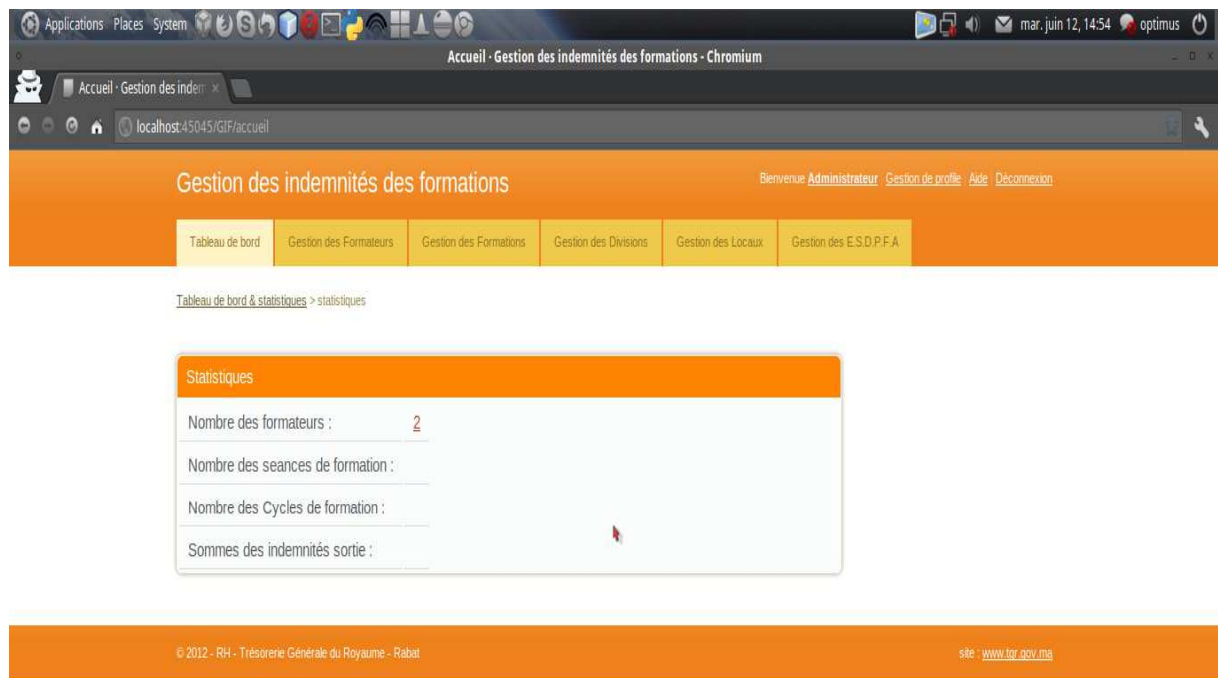


**Figure 23 : page d'authentification**

**Page des statistiques :**

**UNIVERSITE SIDI MOHAMED BEN ABDELLAH**  
**FACULTE DES SCIENCES ET TECHNIQUES FES**  
**DEPARTEMENT D'INFORMATIQUE**

La figure ci-dessous représente la page contenant les statistiques de l'application, elle apparaît une fois l'utilisateur est authentifié.



**Figure 24 : page d'accueil « statistiques & tableau de bord »**

**Page gestion des formateurs (liste des formateurs) :**

# UNIVERSITE SIDI MOHAMED BEN ABDELLAH

## FACULTE DES SCIENCES ET TECHNIQUES FES

### DEPARTEMENT D'INFORMATIQUE

La page liste des formateurs permet de paginer les formateurs saisis dans le système avec leurs informations en détail,

The screenshot shows a web application interface for 'Gestion des indemnités des formations'. The main content area displays a table titled 'Liste des Formateurs' with the following data:

| Titre, Nom & Prénom      | CIN     | PPR        | Téléphone | Service | Opérations   |
|--------------------------|---------|------------|-----------|---------|--|
| Mr. IBRAHIM Jalal        | J34343  | 34343999   | 034343    | RHE     | <a href="#">Supprimer</a> <a href="#">Modifier</a> |
| Mr. HOUSAM Mohamed Amlne | G434323 | 0034958837 | 051583079 | RHT     | <a href="#">Supprimer</a> <a href="#">Modifier</a> |

Below the table, it indicates 'page 1 de 1' and provides pagination controls: 'Précédent', '1', 'Suivant', and 'Voir tout'. To the right of the table, there is a 'Outils' section with buttons for 'Ajouter Nouveau', 'Selectionner tous', and 'Supprimer Selectionné(s)'. Below this, there is a 'Gestion des Grades' section with a 'Liste des grades' button. The footer of the application shows the copyright '© 2012 - RH - Trésorerie Générale du Royaume - Rabat' and the website 'site : www.tgr.gov.ma'.

Figure 25 : gestion des formateurs (liste des formateurs)

**UNIVERSITE SIDI MOHAMED BEN ABDELLAH  
FACULTE DES SCIENCES ET TECHNIQUES FES  
DEPARTEMENT D'INFORMATIQUE**

## **Page, ajouter formateur :**

Cette fenêtre permet d'ajouter un nouveau formateur au système afin de pouvoir lui attribuer des séances de formation au futur.

The screenshot shows a web browser window with the address bar displaying 'localhost:45045/GIF/formateur/nouveau'. The page title is 'TGR - Gestion des indemnités des formations - Accueil - Chromium'. The main content area displays a form titled 'Ajouter un Formateur' with the following fields and controls:

- Titre**: A dropdown menu with the text '--- Sélectionnez ---'.
- Nom (Obligatoire)**: A text input field.
- Prénom (Obligatoire)**: A text input field.
- PPR (Obligatoire)**: A text input field.
- CIN (Obligatoire)**: A text input field.
- Date de Naissance (Obligatoire)**: A date picker showing 'Juin 2012' with a calendar grid. The 12th is highlighted.
- Mot de passe (Obligatoire)**: A text input field.
- Numéro de Téléphone**: A text input field with the value '0'.
- Service et Grade (Obligatoire)**: Two dropdown menus, the first with '--- Sélectionnez service ---' and the second with '--- Sélectionnez grade ---'.

At the bottom right of the form are two buttons: 'Annuler' and 'Enregistrer'. The browser's status bar at the bottom shows the URL 'javascript:calendar.selectDate("2012","06","11")' and the system tray includes icons for 'root@optimus-laptop...', 'TGR - Gestion des ind...', and '[Java EE] - Eclipse'.

**Figure 26 : gestion des formateurs (Ajouter un formateur)**

## **CONCLUSION :**

Dans ce chapitre j'ai présenté la dernière phase du cycle de développement du système GIF. Cette phase a été décomposée en deux sous phases principale : l'architecture applicative du système et la réalisation de la solution représentée par des prises d'écrans.

**UNIVERSITE SIDI MOHAMED BEN ABDELLAH  
FACULTE DES SCIENCES ET TECHNIQUES FES  
DEPARTEMENT D'INFORMATIQUE**

**Conclusion et perspectives :**

Mon travail a consisté, dans un premier temps, en l'étude et l'analyse d'une partie de la politique de formation dans la TGR. Dans un second temps, j'ai modélisé le processus du système GIF qui est un des axes principaux de cette politique de formation.

Au cours de la période de mon projet de fin d'études, j'ai eu l'opportunité de mettre en exercice, différentes connaissances acquises durant mes études à la faculté des sciences et techniques de Fès. De plus, j'ai eu l'occasion d'acquérir de nouveaux concepts à savoir la technologie JEE, l'architecture MVC et beaucoup plus. Les différentes réunions et discussions avec le personnel de la trésorerie générale du royaume m'ont été bénéfiques et surtout le côté relationnel.

Les difficultés majeures que j'ai rencontrées durant ce projet résident essentiellement dans la nouveauté des technologies avec lesquelles j'ai travaillé.

Enfin, je précise que je n'ai pas encore terminé la partie réalisation de l'application la génération des états dans un fichier PDF et la gestion des bénéficiaires ainsi que ce système pourra se doter d'un tableau de bord contenant des statistiques pour aider le gestionnaire à la prise de décisions.

Et j'espère que la réalisation continuera sur la base de mon projet de fin d'études.

**UNIVERSITE SIDI MOHAMED BEN ABDELLAH  
FACULTE DES SCIENCES ET TECHNIQUES FES  
DEPARTEMENT D'INFORMATIQUE**

**Webographie :**

- <http://www.javabeat.net/2007/06/introduction-to-spring-mvc-web-framework-web-tier/>
- <http://stackoverflow.com/questions/6685202/advice-about-using-dao-spring-mvc-hibernate>
- <http://techtracer.com/2008/12/30/the-great-hibernate-tutorial-a-great-jump-start-for-beginners/>
- <http://viralpatel.net/blogs/>
- <http://www.roseindia.net>
- <http://krams915.blogspot.com>
- <http://jpgmr.wordpress.com/2009/12/08/spring-mvc-and-hibernate-made-simple-with-annotations/>
- <http://krams915.blogspot.com/2011/03/spring-hibernate-many-to-one.html>
- <http://www.mkyong.com/spring-mvc/spring-mvc-multiactioncontroller-annotation-example/>
- <http://www.mkyong.com/spring-mvc/spring-mvc-dropdown-box-example/>
- <http://www.stackoverflow.com/user/fstamine>

**Bibliographie :**

- *Polycopies des cours :*
- *Modélisation en UML (Professeur. Abderrahim BENABBOU).*
- *Gestion de configuration (Professeur. Azzedine ZAHI).*
- *[EYROLLES] Java Persistance et Hibernate*
- *[EYROLLES] UML 2.0 Modéliser une application web 4<sup>ème</sup> édition*
- *[EYROLLES] UML 2 en action*
- *[EYROLLES] Programmer en Java*