

Ministère de l'enseignement supérieur de la  
recherche et de l'innovation

Université Assane Seck de Ziguinchor

UFR sciences et technologies

Département informatique



## Mémoire de fin d'études

Pour l'obtention du diplôme de master

Mention : Informatique ; Spécialité : génie logiciel

Sujet :

Automatisation de la gestion des salaires et dématérialisation du  
bulletin de salaire du personnel de l'UASZ

Présenté par :

M. Camir Augustin N. MALACK

### Membres du jury

- Dr. Marie NDIAYE (**Rapporteur**)
- Dr. Youssou DIENG (**Examineur**)
- M. Alain Charles GOMIS (**DRH**)
- M. Al Ousseynou FALL (**CRI**)

### Sous la direction de :

- Dr. Ibrahima DIOP
- Dr. Youssou FAYE

### Sous la supervision de :

- Pr. Salomon SAMBOU

2017-2018

## **RESUME**

Pour automatiser certaines tâches, beaucoup de structures se procurent des logiciels, qui souvent sont faits pour répondre à une variété de besoins d'entreprises quel que soit leur secteur d'activités. Cependant, dans la plupart des cas, ces logiciels répondent partiellement aux besoins de ces dites structures et certaines fonctionnalités restent inutilisées. Ainsi, pour répondre à un besoin spécifique d'une entreprise, il va falloir procéder à un travail d'adaptation, parfois difficile, notamment quand le code source n'est pas accessible.

L'UASZ à travers sa direction des ressources humaines utilise depuis longtemps ce genre d'application pour la gestion des bulletins de salaire de son personnel. En dépit des manquements notés, ce logiciel, non seulement, ne répond pas totalement aux besoins de la direction des ressources, mais présente l'inconvénient majeur d'avoir un code source inaccessible rendant ainsi le processus d'adaptation quasi impossible.

C'est ainsi que dans ce mémoire, nous proposons une application afin de répondre aux préoccupations de la direction des ressources. Après une étude et des séries d'entretiens, menés conjointement avec la direction des ressources, nous avons utilisé la méthodologie agile Scrum pour le processus de développement de notre application. Cependant, une spécification claire des besoins nous a permis de formaliser les données avec UML (Unified Modeling Language) et de les stocker dans MySQL (My Structured Query Language). Une implémentation avec java, nous a permis de disposer d'une application sécurisée qui va non seulement produire les bulletins de salaire, mais aussi d'envoyer mensuellement une copie par e-mail et des notifications par sms afin d'informer de la disponibilité du bulletin de salaire.

## DEDICACES

*C'est avec une profonde gratitude et sincères mots, que je dédie ce modeste travail de fin d'études à mes chers parents qui ont sacrifié leur vie pour ma réussite. Ils m'ont éclairé le chemin par leur conseil judicieux. J'espère qu'un jour, je pourrais leur rendre un peu de ce qu'ils ont fait pour moi, que Dieu leur accorde bonheur et longue vie.*

*Ce travail est dédié aussi à mes frères et sœurs pour leur amour, leur soutien et leurs encouragements.*

*Merci à vous tous.*

## **REMERCIEMENTS**

*Je ne saurais introduire ce mémoire sans remercier toutes les personnes de bonne volonté qui n'ont ménagé aucun effort pour la réalisation de ce modeste travail de mémoire de fin d'études, à savoir mes encadreurs Dr. Ibrahima DIOP et Dr. Youssou FAYE. Je vous remercie pour le temps que vous avez consacré à m'apporter les outils méthodologiques indispensables à la conduite de ce projet.*

*Je remercie l'ensemble du personnel de la Direction des Ressources Humaines, particulièrement son directeur M. Alain Charles GOMIS, qui m'a permis de comprendre la gestion des salaires et les calculs de l'abattement et des impôts, ainsi que sa disponibilité pour mener à bien ce travail.*

*Un grand merci au chef de la division soldes et dépenses du Personnel Mme POUYE, pour son précieux temps qu'elle m'a accordé et qui m'a permis de comprendre les forces et les faiblesses de l'existant.*

*Au terme de ce modeste travail, nous tenons aussi à remercier chaleureusement et respectivement tous ceux qui y ont contribué de près ou de loin, particulièrement le chef de Département d'Informatique Dr. Youssou DIENG pour avoir mis en place ce cadre de partenariat entre le Département d'Informatique et la DRH. C'est ce qui m'a permis d'avoir ce sujet de stage.*

*Des remerciements vont également à l'égard de tous les enseignants du département d'Informatique, les intervenants externes et l'équipe pédagogique pour toutes les connaissances théoriques et pratiques et l'accompagnement qu'ils m'ont apportés tout au long de mon cursus universitaire.*

*Enfin, je remercie mes amis et camarades de promotion pour ces années passées ensemble.*

*Merci à tous.*

## SOMMAIRE

RESUME.....	i
DEDICACES .....	ii
REMERCIEMENTS .....	iii
SOMMAIRE .....	iv
LISTE DES FIGURES.....	v
LISTE DES TABLEAUX.....	vii
LISTE DES ABREVIATIONS.....	viii
INTRODUCTION GENERALE.....	12
CHAPITRE I : CONTEXTE JUSTIFICATIF DU SUJET.....	14
I- La gestion des salaires à l'Université ASSANE SECK de Ziguinchor.....	14
II- Problèmes rencontrés dans la gestion des salaires à l'UASZ.....	18
III- Problématique du sujet .....	19
CHAPITRE II : PROCESSUS DE DEVELOPPEMENT DE L'APPLICATION .....	22
I- Présentation de la méthodologie SCRUM .....	22
II- Les cycles de la méthodologie SCRUM .....	24
III- Organisation de l'équipe du projet selon le Framework SCRUM.....	26
CHAPITRE III : SPECIFICATION ET ANALYSE DES BESOINS FONCTIONNELS.....	28
I- La spécification des besoins.....	28
II- Analyse des besoins fonctionnels du système.....	41
CHAPITRE IV : CONCEPTION DU SYSTEME.....	50
I- Conception générale.....	50
II- Conception préliminaire.....	56
CHAPITRE V : IMPLEMENTATION ET PRESENTATION DE L'APPLICATION.....	60
I- Implémentation de l'application.....	60
II- Présentation de l'Application.....	77
CONCLUSION GENERALE .....	87
BIBLIOGRAPHIE ET WEBOGRAPHIE .....	88

## LISTE DES FIGURES

Figure 1 : fonctionnement du Framework Scrum .....	25
Figure 2 : Diagramme de cas d'utilisation du CSDSP .....	32
Figure 3 : Diagramme de cas d'utilisation du DRH.....	33
Figure 4 : Diagramme d'activités du cas « s'authentifier » .....	42
Figure 5 : Diagramme de séquence du cas « s'authentifier ».....	43
Figure 6 : diagramme d'activités du cas « ajouter agent ».....	44
Figure 7 : Diagramme de séquence du cas « ajouter agent » .....	45
Figure 8 : Diagramme d'activités du cas « attribuer grade » .....	46
Figure 9 : Diagramme de séquence du cas « attribuer grade » .....	47
Figure 10 : Diagramme d'activités du cas « éditer bulletin de paie provisoire ».....	48
Figure 11 : Diagramme de séquence du cas « éditer bulletin de paie provisoire ».....	49
Figure 12 : schéma du modèle MVC .....	51
Figure 13 : Schéma de l'architecture 3 tiers.....	51
Figure 14 : Schéma de l'architecture de notre système.....	52
Figure 15 : Diagramme de package.....	55
Figure 16 : diagramme de déploiement.....	56
Figure 17 : Diagramme de classe des classes participantes aux fonctionnalités ajout, modification et désactivation d'un agent.....	57
Figure 18 : Diagramme de classe des classes participantes aux fonctionnalités d'édition et de la validation d'un bulletin de salaire provisoire.....	58
Figure 19 : Architecture de Spring.....	63
Figure 20 : Représentation de Spring boot par rapport Spring .....	65
Figure 21 : Arborescence des fichiers de l'application .....	68
Figure 22 : Arborescence des packages du projet .....	69
Figure 23 : Exemple de la classe « Agent » .....	71
Figure 24 : Exemple de l'interface « FonctionRepository » .....	72
Figure 25 : Exemple avec l'interface « BulletinService » .....	73
Figure 26 : Exemple avec la classe « BulletinServiceImpl » .....	74

## Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

Figure 27 : Exemple avec la classe « StatutController ».....	75
Figure 28 : Exemple avec la classe « SecuriteConfig » .....	76
Figure 29 : la liste des différents menus et sous menus de l'application.....	77
Figure 30 : Page d'authentification .....	78
Figure 31 : Page d'accueil du DRH.....	79
Figure 32 : Page pour la saisie des informations sur l'identité de l'agent	<b>Erreur ! Signet non défini.</b>
Figure 33 : Page pour la saisie des informations sur l'adresse de l'agent.....	80
Figure 34 : Page pour la saisie des informations sur l'identité de l'agent .....	81
Figure 35 : Page de l'agent .....	81
Figure 36 : Page pour l'attribution d'un grade à un agent .....	82
Figure 37 : Page pour le choix du grade de l'agent.....	83
Figure 38 : Page pour lister les grades de l'agent .....	83
Figure 39 : Page pour éditer les bulletins de salaire.....	84
Figure 40 : Page pour vérifier le bulletin de salaire d'un agent .....	85
Figure 41 : Interface pour la saisie du mot de passe pour consulter le bulletin .....	86
Figure 42 : Bulletin de salaire d'un agent de l'UASZ.....	86

## **LISTE DES TABLEAUX**

Tableau 1 : procédure de la gestion des salaires .....	16
Tableau 2 : Equipe de pilotage du projet.....	27
Tableau 3 : les acteurs de notre système .....	29
Tableau 4 : les fonctionnalités de notre système.....	29
Tableau 5 : description du cas d’utilisation « s’authentifier » .....	34
Tableau 6 : description du cas d’utilisation « ajouter agent ».....	34
Tableau 7 : description du cas d’utilisation « modifier agent » .....	35
Tableau 8 : description du cas d’utilisation « désactiver agent ».....	36
Tableau 9 : description du cas d’utilisation « attribuer grade » .....	36
Tableau 10 : description du cas d’utilisation « affichage des différents agents qui bénéficient d’avancement automatique ».....	37
Tableau 11 : description du cas d’utilisation «édition des bulletins de paie provisoire ».....	38
Tableau 12 : description du cas d’utilisation « validation des bulletins de paie provisoire ».....	38
Tableau 13 : description du cas d’utilisation « édition bulletin de paie finale ».....	39
Tableau 14 : description du cas d’utilisation « envoi par email les bulletins de salaire » .....	40
Tableau 15 : description du cas d’utilisation « envoi des sms » .....	40

## **LISTE DES ABREVIATIONS**

ASF:	Apache Software Foundation
API:	Application Programming Interface
CSDSP :	Chef de Division Solde et Dépense du Personnel
CRI :	Centre des Ressources Informatiques
CSS :	Cascading Style Sheets
DAF :	Direction des Affaires Financières
DAO :	Data Access Objet
DRH :	Directeur des Ressources Humaines
EDI:	Environnement de Développement Intégré
EJB :	Enterprise Java Beans
GNU :	General Public License
HTML:	Extensible HyperText Markup Language
HTTP:	HyperText Transfer Protocol
IRP :	Impôt sur le Revenu Projeté
J2EE :	Java Enterprise Edition
JPA :	Java Persistence API
JDBC:	Java DataBase Connectivity
JDO:	Java Data Objects
JS :	Java Script
JSP:	JavaServer Pages

## **Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»**

JVM:	Java Virtual Machine
MVC :	Modèle Vue et Contrôleur
MySQL:	My Structured Query Language
ORM:	Object-Relational Mapping
PATS :	Personnel Administratif Technique et de Service
PATSFn :	Personnel Administratif Technique et de Service Fonctionnaire
PER :	Personnel Enseignant et Recherche
PV:	Procès-Verbal
POM:	Project Object Model
RAM:	Random Access Memory
SG :	Secrétaire Général
SGBD :	Système de Gestion des Bases de Données
SGBDR :	Système de Gestion des Bases de Données Relationnelles
SQL:	Structured Query Language
UASZ :	Université Assane Seck de Ziguinchor
UFR :	Unité de Formation et de Recherche
UML:	Unified Modeling Language
XML :	Extensible Markup Language

## **INTRODUCTION GENERALE**

La gestion des salaires s'articule autour de la réalisation des bulletins de paie et du traitement des salaires, dans le respect d'un cadre juridique et de certaines obligations légales. Le bulletin de paie est un document périodique obligatoire devant répondre à certaines règles. Et le traitement de salaires est un processus permettant le calcul et la fiabilité de tous les éléments qui participent à la rémunération du salarié.

Cependant, les grandes structures comme l'Université Assane Seck de Ziguinchor (UASZ) utilisent les logiciels des paies pour gérer la rémunération de leurs agents. Un logiciel de paie est une application informatique qui gère l'intégralité du processus d'obtention des salaires d'une entreprise, de l'entrée des salariés jusqu'à leur sortie.

Beaucoup de logiciels de gestion de salaire qu'on trouve sur le marché sont conçus de façon à s'adapter dans n'importe quelle structure, alors que les réalités peuvent parfois varier d'une organisation à une autre. Ainsi, ces logiciels du marché n'arrivent pas à gérer certaines tâches et présentent parfois beaucoup de manquement.

L'UASZ qui en possède un de ces logiciels et dont elle s'en sert depuis pour gérer les salaires de son personnel, rencontre beaucoup de problèmes, car certaines tâches censées être automatiques sont faites de façon manuelle et il existe encore des besoins auxquels le logiciel ne peut pas satisfaire.

Pour pallier à cela, l'UASZ à travers la direction des ressources humaines en partenariat avec le département Informatique, a décidé de concevoir et d'implémenter un système d'information optimale et sûr pour l'automatisation de la gestion des salaires et la dématérialisation du bulletin de salaire de son personnel. Cette application permettra de produire les bulletins de salaire des agents de manière automatique, mais aussi de pouvoir envoyer les bulletins de paie par e-mails pour chaque agent et notifier l'agent par sms pour lui informer de la disponibilité du bulletin dans sa boîte e-mail.

Ainsi, pour la mise en œuvre de cette application, nous avons d'abord fait une étude du logiciel qui existe en ayant des séries d'entretiens avec le DRH, et ce dernier nous éclaircit sur les problèmes

auxquels il est confronté avec le logiciel de gestion de paie. Et faces à ces difficultés, le DRH a exprimé ses besoins en précisant ses attentes sur le nouveau logiciel. Un logiciel qui doit répondre à toutes les normes de sécurité, doit être capable de produire des bulletins de salaires pour chaque agent et communiquer à travers des e-mails et sms. Afin de réaliser les fonctionnalités qui doivent exister sur le futur logiciel, nous avons utilisé UML (Unified Modeling Language) pour formaliser les données qui seront stockées dans un système de gestion des bases de données (SGBD). Et pour permettre, l'interaction avec ces données nous avons travaillé avec des technologies informatiques adaptées qui nous ont permis d'atteindre nos objectifs et mener à bien notre travail.

A cet effet, nous avons réparti notre mémoire en cinq chapitres principaux à commencer par :

- ❖ **Chapitre I : *contexte justificatif du sujet*** expose les problèmes des bulletins de salaire et en dégage la problématique
- ❖ **Chapitre II : *le processus de développement de l'application*** nous permet de faire connaissance du Framework Scrum
- ❖ **Chapitre III : *la spécification et l'analyse des besoins fonctionnels*** identifie les acteurs, propose les fonctionnalités et analyse les besoins que doit satisfaire ce système en question
- ❖ **Chapitre IV : *la conception du système*** aborde la conception générale et la conception détaillée du système
- ❖ **Chapitre V : *implémentation et la présentation de l'application*** présentent les technologies utilisées pour la réalisation du système d'information et l'application elle-même.

## CHAPITRE I : CONTEXTE JUSTIFICATIF DU SUJET

Le présent chapitre se décompose en trois sections principales. Ainsi, dans un premier temps, nous parlons brièvement de la gestion des salaires à l'UASZ. Nous détaillons ensuite les problèmes rencontrés dans la gestion des salaires à l'UASZ. Et enfin, nous abordons enfin la problématique du sujet.

### ***I- La gestion des salaires à l'Université ASSANE SECK de Ziguinchor***

L'Université Assane SECK de Ziguinchor (UASZ) est créée par le décret 2008-537 du 22 mai 2008. Elle a démarré ses activités en février 2007 avec (03) Unités de Formation et de Recherche (UFR) et des Centres et Services communs. Chaque UFR est composée de plusieurs départements et/ou filières. Aujourd'hui, L'UASZ compte moins de quatre mille étudiants (4 000) répartis dans quatre (04) UFR que sont :

- ❖ l'UFR Sciences Économiques et Sociales ;
- ❖ l'UFR Sciences et Technologies ;
- ❖ l'UFR Lettre, Arts et Sciences Humaines ;
- ❖ l'UFR Sciences de la Santé.

Les centres et services communs quant à eux ne cessent d'augmenter d'année en année. Et on distingue ainsi :

- ❖ le Service de Formation Ouverte et à Distance ;
- ❖ le Centre des Ressources Informatiques ;
- ❖ la Bibliothèque Universitaire ;
- ❖ la Direction du Service aux Étudiants ;
- ❖ la Direction des Affaires Financières
- ❖ la Direction Centrale de la Scolarité ;
- ❖ la Direction de la Gestion du Patrimoine et de Maintenance ;

## **Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l’UASZ»**

- ❖ la Direction de l’Environnement et de la Sécurité ;
- ❖ la Direction des Ressources Humaines.

Ainsi depuis sa création jusqu’à nos jours, l’UASZ a noté une évolution dans tous les domaines, ce qui a impliquée une augmentation du personnel travaillant dans la structure, pour son bon fonctionnement. Et selon le code du travail, chaque personne qui travaille au sein de la structure a le droit de recevoir son salaire décrit dans un bulletin de salaire à chaque fin du mois. Pour dire que le nombre de bulletins de salaire que l’UASZ produit par mois pour ses salariés n’a pas cessé de croître durant ces dernières années et continuera de croître pour les années à venir.

La gestion des salaires à l’UASZ distingue trois (03) types de personnels (et donc trois types de bulletins de salaire), à savoir :

- ❖ Le Personnel Enseignant et Recherche (PER) ;
- ❖ Le Personnel Administratif Technique et Service (PATS) ;
- ❖ Les Fonctionnaires Personnels Administratifs Technique et Service (PATSFn).

La Direction des Ressources Humaines est la composante essentielle de l’Université qui se charge de la gestion des salaires. Cette direction emploie trois (3) personnes que sont le directeur des ressources humaines (DRH), le chef de la division solde et dépense du personnel (CSDSP) et la secrétaire. [1]

Rattachée au secrétariat général, la Direction des Ressources Humaines, par le biais de son Directeur, est chargée de la bonne gestion de son personnel. Ainsi, son directeur est chargé d’appliquer les textes statutaires et réglementaires en matière de personnel. Il veille à la mise en œuvre d’une politique et de moyens adéquats en vue d’encourager le personnel compétent à rester au service de l’Université. Il tient le secrétariat des différentes commissions d’avancement du personnel. Il est chargé d’établir et d’assurer le suivi des relations professionnelles entre l’Université Assane SECK et les services de l’Etat chargés de la fonction publique, de l’emploi et du travail et le ministère chargé de l’enseignement supérieur. Il se charge de diffuser les informations relatives au statut, de promouvoir le personnel d’enseignement et de recherche, de centraliser les dossiers venant des établissements et instituts et de les acheminer vers les structures compétentes. Il administre les salaires et tient les fichiers de salaire qui constitue notre cadre de

**Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l’UASZ»**

travail de mémoire. Pour la tenue des fichiers de salaires, le directeur des ressources humaines est aidé par le chef de la division administrative du personnel qui par ailleurs assure son intérim en cas d’absence. [1]

Dans le tableau, ci-dessous, nous allons décrire les différentes étapes ainsi que les acteurs qui interviennent dans la procédure de génération des bulletins de salaire du personnel de l’UASZ.

***Tableau 1 : Procédure de gestion des salaires***

<i>Fait générateur</i>	<i>Intervenant</i>	<i>Description</i>
<b>Établissement des bulletins de paie</b>	DRH	<ul style="list-style-type: none"> <li>- Reçoit jusqu’au 14 du mois les papiers justificatifs de modification de salaire du personnel</li> <li>- Les transmettent au directeur Financier et Comptable (DFC)</li> </ul>
<b>Étude des modifications</b>	Cellule de traitement des salaires	<ul style="list-style-type: none"> <li>- se réunit tous les 15 du mois sur convocation du DRH pour examiner les différents cas relatifs aux salaires</li> <li>- Établis un PV et le transmet à la division solde et dépense du personnel le jour suivant pour la prise en charge des modifications</li> </ul>
<b>Réception PV</b>	Chef de division solde et dépense du personnel	<ul style="list-style-type: none"> <li>- effectue la saisie informatique des éléments variables de paie de chaque employé ;</li> <li>- procède au calcul des salaires ;</li> <li>- s'assure que tous les montants des bulletins sont correctement calculés ;</li> <li>- procède à l'édition provisoire des bulletins de paie</li> </ul>

**Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»**

		<ul style="list-style-type: none"> <li>- transmets les états au DRH via le DFC pour validation par la cellule au plus tard le 20 du mois</li> </ul>
<b>Réception des états</b>	DRH	<ul style="list-style-type: none"> <li>- Vérifie la conformité des états par rapport aux dispositions prises par la cellule en s'assurant que tous les éléments variables approuvés ont été pris en compte dans les bulletins édités par la division</li> <li>- S'assure d'inexistence d'autres erreurs</li> <li>- Retourne le document à la division solde et dépense du personnel au plus tard le 22 du mois</li> </ul>
<b>Impression des bulletins de paies</b>	DFC	<ul style="list-style-type: none"> <li>- au plus tard le 25 du mois, avec le CRI arrête les salaires et imprime les différents états à savoir : <ul style="list-style-type: none"> <li>✓ L'état des salaires,</li> <li>✓ Les états de paiement par banque,</li> <li>✓ Les états des cotisations,</li> <li>✓ Les mandats correspondants</li> </ul> </li> <li>- Transmets le dossier au Recteur via le SG pour signature</li> <li>- Après signature transmet à l'agence comptable tous les documents pour paiement</li> <li>- Archive les états mensuels de paie</li> </ul>

Tout le processus décrit ci-dessus dans le tableau est décrit de manière théorique, mais dans son élaboration, c'est-à-dire dans la pratique la direction des ressources humaines est confrontée à pas mal de difficultés pour l'élaboration des bulletins de salaire. Nous allons voir les problèmes rencontrés dans la gestion des salaires dans la partie qui suit.

## **II- Problèmes rencontrés dans la gestion des salaires à l'UASZ**

Pour la gestion des salaires à l'UASZ, la DRH dispose d'un logiciel développé par l'université de Cheikh Anta Diop de Dakar (UCAD). Les conséquences directes de l'utilisation de ce logiciel générique sont l'absence de certaines fonctionnalités pour le traitement automatique des bulletins de salaire et la présence de paramètres ou d'informations non-utiles pour le cas de l'UASZ. Ce qui est la cause de beaucoup d'autres problèmes dans la gestion des salaires à l'UASZ, comme :

- ❖ La saisie manuelle de beaucoup d'informations (intérêts, primes, changement d'échelons ou de grades, etc.).
- ❖ Une gestion des droits d'accès limitée (actuellement, le logiciel se trouve dans la machine du Chef de Division Solde et Dépenses du Personnel).
- ❖ Une ergonomie restreinte.
- ❖ Une lourdeur dans la gestion des rajouts mensuels de distribution des salaires (actions manuelles).
- ❖ L'utilisation du papier pour l'impression.
- ❖ La distribution des bulletins de salaire se fait de manière trimestrielle.
- ❖ L'automatisation de toute la procédure de gestion des salaires pas encore à l'ordre du jour (la procédure énoncée dans le tableau 1).

Parfois, on est obligé de faire appel à l'ingénieur informatique du CRI de l'UASZ qui utilise son savoir-faire pour résoudre certaines tâches et il arrive même qu'il contacte les concepteurs du logiciel qui lui donnent des directives pour corriger des erreurs.

Avec un personnel qui ne cesse d'augmenter, les tâches deviennent plus fréquentes et complexes. Ainsi, l'UASZ pour s'adapter à sa croissance nécessite une solution à la hauteur de ses attentes pour la gestion des salaires de son personnel exigeant. Ce qui constitue la problématique de notre stage à la DRH que nous présentons dans la partie suivante.

### **III- Problématique du sujet**

Depuis trop longtemps, l'UASZ rencontre beaucoup de problèmes avec leur logiciel de gestion de salaire, car ce logiciel présente beaucoup de manquements et certaines tâches censées être automatiques sont faites de façon manuelle.

Ainsi, dans cette partie, nous allons présenter des solutions aux problèmes liées à la gestion des salaires puis nous allons proposer une méthodologie de gestion de projet appropriée.

#### **1. Solution proposée**

La solution que nous proposons est un logiciel adapté aux besoins spécifiques de la gestion des salaires à l'UASZ. Dans un premier temps, le logiciel va permettre de :

- ❖ **Automatiser la production des bulletins de salaire des agents** : chaque type d'agents aura son propre modèle de bulletin de paie puisque les agents n'ont pas le même système de graduation. Les bulletins de salaire auront des informations détaillées de chaque élément de salaire disponible sur le bulletin ce qui permettra aux agents de pouvoir comprendre le calcul de leur salaire. Et enfin, chaque agent aura ces bulletins de paies disponibles dans son e-mail de l'UASZ et il l'aura la possibilité de l'imprimer à tout instant et en chaque lieu, ainsi qu'un sms de notification est envoyé à l'agent. Pour préserver la sécurité des bulletins même après l'envoi des bulletins par mails un code de sécurité est attribué à chaque bulletin.
- ❖ **Automatiser la procédure de gestion des salaires** : toute la procédure entière de la gestion des salaires sera automatisée cela permettra d'imprimer des PV à partir du logiciel.

#### **2. Objectifs spécifiques du sujet**

## Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

Les besoins principaux à couvrir par l'outil de gestion automatique des salaires sont donc les suivants :

- ❖ **La gestion des informations des agents** : chaque agent appartient un type, il peut être PER, PATS ou PATSFn. La catégorie de l'agent lui procure certaines informations qu'elles partagent et d'autres qui lui sont spécifiques.
- ❖ **La gestion des éléments de salaire** : les éléments de salaire (les primes, les éléments de calcul de l'impôt, ...) seront paramétrables automatiquement pour éviter des erreurs de saisie. Par exemple avec l'ancien logiciel le, nombre de parts de l'agent était saisi, mais dans notre application elle est lié automatiquement à sa situation familiale.
- ❖ **Les Calculs du salaire brut au salaire net** : le paramétrage des éléments pour le calcul de l'impôt va permettre au DRH de pouvoir configurer automatiquement l'application après le vote de chaque nouvelle loi sur l'impôt national.
- ❖ **La Production des bulletins de salaire** : la production des bulletins de salaire se fait de manière trimestrielle, alors que la paie des agents se fait de manière mensuelle.
- ❖ **Envoie par e-mail des bulletins de salaire** : après que le logiciel ait produit les bulletins de salaire de ses agents, il doit être en mesure de les envoyés par e-mails pour chaque agent qui travaille dans la structure.
- ❖ **Notification par sms de l'envoi des bulletins de salaire** : après qu'un bulletin de salaire est envoyé par e-mail, une notification par sms est directement envoyée dans le téléphone portable de l'agent.
- ❖ **Test logiciel** : le logiciel sera testé pendant six (06) mois avant son utilisation, pour pouvoir corriger d'éventuelles erreurs.
- ❖ **Documentation du code source et manuel d'utilisation** : après la conception du logiciel, le code sera bien commenté pour assurer la maintenance du logiciel. Ainsi un manuel d'utilisation sera aussi élaboré pour la bonne maîtrise du logiciel par la DRH.
- ❖ **Automatisation de la procédure de gestion des salaires** : comme le logiciel ne produit que les bulletins de salaire, on va lui ajouter la possibilité d'automatiser toute la procédure pour l'obtention des salaires, c'est-à-dire : l'archivage des papiers justificatifs de modification de salaire, l'ajout ou la modification des éléments variables de paie, la génération automatique des PV par la Cellule de traitement des salaires, etc.

## Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

Pour réaliser tout le travail énoncé ci-dessus nous avons besoin de travailler en parfaite collaboration avec le DRH, c'est pour cela qu'on a adopté une méthodologie de développement qui porte le **client (DRH) au centre**, le DRH est impliqué dans la réalisation du début à la fin du projet. Ainsi pour ce faire nous allons utiliser l'une des méthodologies agiles qui est le **Scrum** qu'on va détailler dans le chapitre suivant.

## CHAPITRE II : PROCESSUS DE DEVELOPPEMENT DE L'APPLICATION

Un processus de développement est un ensemble d'activités structuré à réaliser pour atteindre l'objectif d'un projet de système informatique, dont les activités varient en fonction de l'organisation du projet, et du type de système à développer.

La méthode SCRUM est une des nombreuses méthodes de gestion de projet, qui permet entre autre de développer des produits complexes. Lancée en 1993 par Jeff Sutherland, SCRUM fait partie du groupe de méthodes AGILE qui sont, de manière générale, "des groupes de pratiques de pilotage et de réalisation de projets" destinés de base à l'informatique. Cette méthode fait partie des méthodes de management modernes et figure parmi les plus utilisées, ce qui prouve son efficacité et son rendement. Elle est utilisée par de nombreuses entreprises telles qu'on peut par exemple penser à développer une nouvelle application mobile, bureau ou encore un site Web en appliquant la méthode SCRUM. Pour information, SCRUM n'est pas un acronyme mais veut dire "Mêlée" en anglais, ce qui est une belle image pour illustrer l'esprit de groupe pour avancer ensemble vers un même objectif (ce qui permet de bien résumer l'esprit SCRUM).

La méthodologie SCRUM est plus qu'une technique, elle est un changement dans la façon d'aborder le développement logiciel en passant par une plus grande participation des intervenants externes et internes. C'est un engagement quotidien de toutes les parties prenantes vers un avenir où la qualité logicielle et l'authenticité des rapports humains n'auront pas de compromis sinon que la synchronicité et le succès de leurs entreprises. [2]

Nous allons voir au long de ce chapitre le processus de développement de notre application avec la méthodologie Scrum, qui est une implémentation de la méthode agile.

### ***I- Présentation de la méthodologie SCRUM***

Scrum est un schéma d'organisation de développement de produits complexes. Il est défini par ses créateurs comme un « cadre de travail holistique itératif qui se concentre sur les buts communs en livrant de manière productive et créative des produits de la plus grande valeur possible ». Le Framework s'appuie sur le découpage d'un projet en boîtes de temps, nommées « sprints ». Les sprints peuvent durer entre quelques heures et un mois (avec une préférence pour deux semaines).

Chaque sprint commence par une estimation suivie d'une planification opérationnelle. Le sprint se termine par une démonstration de ce qui a été achevé. Avant de démarrer un nouveau sprint, l'équipe réalise une rétrospective. Cette technique analyse le déroulement du sprint achevé, afin d'améliorer ses pratiques. Le flot de travail de l'équipe de développement est facilité par son auto-organisation, il n'y aura donc pas de gestionnaire de projet. [3]

Scrum définit trois rôles : le propriétaire du produit (product owner), le maître mêlé (scrum master) et le développeur. Il est à noter que le rôle de développeur couvre plusieurs métiers d'une organisation traditionnelle.

**Propriétaire du produit** : le propriétaire du produit (**product owner**) est le représentant des clients et des utilisateurs. Il est responsable de maximiser la valeur du produit et du travail de l'équipe de développement. Il s'agit d'une personne et non d'un comité. Il est seul à diriger l'activité de l'équipe de développement à qui il n'est pas permis de suivre les instructions d'une autre personne. De ce fait, cet acteur se charge de différents rôles et responsabilités :

- ❖ Il explicite les éléments (items) du carnet du produit.
- ❖ Il définit l'ordre dans lequel les fonctionnalités seront développées et prennent les décisions importantes concernant l'orientation du projet.
- ❖ Il s'assure que le carnet du produit est visible et compris de l'équipe de développement.
- ❖ Il participe avec l'équipe à la définition de l'objectif du Sprint au début de celui-ci (Sprint Planning). L'objectif peut être fonctionnel, technique ou organisationnel. Si l'objectif devient obsolète pendant le sprint, il a alors la possibilité d'interrompre le sprint en cours.
- ❖ Dans l'idéal, le propriétaire du produit travaille dans la même pièce que l'équipe. Il est important qu'il reste très disponible pour répondre aux questions de l'équipe et pour lui donner son avis sur divers aspects du logiciel. [3]

**Maître de mêlée** : Le maître de mêlée (**scrum master**) est responsable de la compréhension, de l'adhésion et de la mise en œuvre du Framework. C'est un leader au service de l'équipe, il assiste chaque rôle de l'équipe scrum dans son activité et promeut le changement des interactions entre les rôles dans le but de maximiser la valeur de ce que produit l'équipe. Son autorité s'exerce sur le processus de développement (définition de la durée des Sprints, ...), mais il ne dispose d'aucune autorité sur les autres membres de l'équipe scrum. Ce n'est pas un chef de projet, ni un développeur,

ni un intermédiaire de communication avec les clients. Le rôle de scrum master ne peut pas être cumulé avec celui de propriétaire du produit. Les attributions du « chef de projet » présent dans d'autres approches sont distribuées dans les différents rôles de l'équipe scrum. L'existence du rôle de chef de projet dans un contexte scrum est le signe d'une méconnaissance fondamentale de scrum et est perçue comme contre-productive et menant à de piètres résultats. En tant que facilitateur, il aide l'équipe à déterminer quelles interactions avec l'extérieur lui sont utiles, et lesquelles sont freinantes. Il aide alors à maximiser la valeur produite par l'équipe. Parmi ses attributions :

- ❖ communiquer la vision et les objectifs à l'équipe
- ❖ faciliter les rituels de scrum
- ❖ coacher l'équipe de développement
- ❖ écarter les éléments pouvant perturber l'équipe [3]

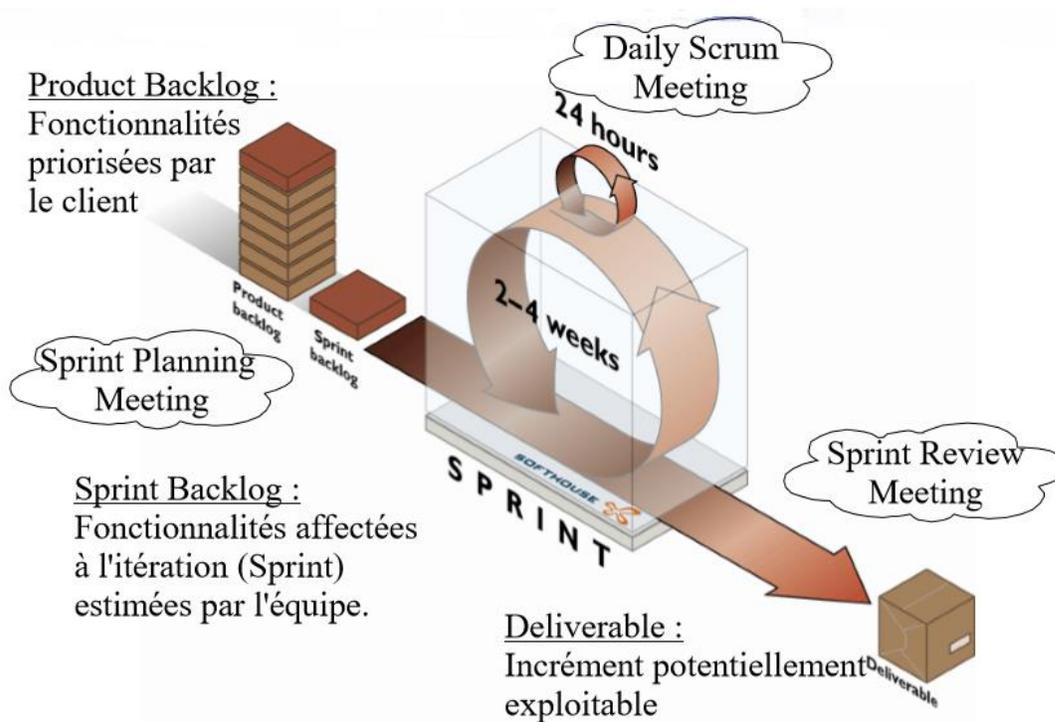
**Équipe de développement** : L'équipe de développement (**scrum team**) a pour responsabilité de livrer à chaque fin d'itération une nouvelle version de l'application enrichie de nouvelles fonctionnalités et respectant le niveau de qualité nécessaire pour être livrée. L'équipe ne comporte pas de rôles prédéfinis ; elle est « structurée et habilitée par l'entreprise à organiser et gérer son propre travail ». Elle est auto-organisée et choisit la façon d'accomplir son travail, sans que ce soit imposé par une personne externe. Il n'y a pas non plus de notions de hiérarchie interne : toutes les décisions sont prises ensemble. Ce mode d'organisation a pour objectif d'augmenter l'efficacité de travail de l'équipe. Elle est pluridisciplinaire et comporte toutes les compétences pour réaliser son projet, sans faire appel à des personnes externes à celle-ci. L'objectif de l'équipe est de livrer le produit par petits incréments. Ainsi, à tout instant, il existe une version du produit « potentiellement utilisable » disponible. L'équipe s'adresse directement au propriétaire du produit, et ne prend ses instructions que de lui. Son activité est issue du carnet de produit uniquement. [3]

## ***II- Les cycles de la méthodologie SCRUM***

## Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

La méthodologie Scrum est l'approche agile la plus utilisée des approches agiles existantes et est simple à comprendre. Nous allons expliquer le mode de fonctionnement du Framework scrum dans l'image suivante :

- ❖ On commence toujours par un carnet du produit (Product backlog) – Une liste bien structurée de tout ce qui devrait être implémenté dans le produit. Cela constituera notre seule source de spécifications.
- ❖ Le carnet de produit sera ensuite décomposé en groupes de spécifications appelés carnets de sprint (sprint backlog) – Cette tâche est réalisée par l'équipe de développement durant un sprint planning où le Product owner explique les spécifications qui sont sur le carnet du produit.



**Figure 1 : fonctionnement du Framework Scrum [4]**

- ❖ La Réunion de planification d'un Sprint (sprint planning meeting) : toute l'équipe scrum est présente à cette réunion, qui ne doit pas durer plus de 8 heures pour un sprint d'un mois. Pour un sprint plus court, la durée est en général réduite, mais peut durer 8h. À l'issue de

cette réunion, l'équipe a décidé des éléments du carnet du produit qu'elle traitera dans le cadre de la prochaine itération, et comment elle s'organisera pour y parvenir.

- ❖ Chaque carnet de sprint est délivré en incréments appelés sprints. Ceux-ci peuvent durer entre une (01) et quatre (04) semaine, pas plus. L'idée de sprints étant de délivrer le carnet de produit en incréments.
- ❖ Une fois le sprint démarré, une réunion quotidienne qu'on appelle sprint quotidien (daily sprint meeting) est établie entre l'équipe. Cette réunion concerne uniquement l'équipe de développement et elle ne devrait pas dépasser les quinze (15) minutes.
- ❖ À la fin de chaque sprint, il est nécessaire de faire une revue de sprint (sprint review meeting) de deux (02) à quatre (04) heures. C'est à ce niveau-là que l'équipe Scrum ainsi que les parties prenantes parcourent ce qui a été implémenté au cours du sprint.
- ❖ Un sprint rétrospectif (rétrospective de sprint) est tenue afin de permettre à l'équipe Scrum d'auto évaluée et de créer un plan comprenant toutes les améliorations à prendre en considération lors du sprint suivant.

### ***III- Organisation de l'équipe du projet selon le Framework***

#### ***SCRUM***

##### ***1. L'équipe de pilotage du projet***

Le tableau suivant résume les différents membres de l'équipe de pilotage du projet selon le Framework scrum.

**Tableau 2 : Equipe de pilotage du projet**

Personne	Rôles	Fonction
<b>M. Alain Charles GOMIS</b>	Product Owner	Directeur des Ressources Humaine de l'UASZ
<b>Dr. Ibrahima DIOP</b>	Scrum Master	Enseignant-chercheur au Département d'Informatique de l'UASZ
<b>M. Camir A. N. MALACK</b>	SCRUM Team	Etudiant en Master 2 au Département d'Informatique de l'UASZ
<b>Dr. Youssou FAYE</b>	SCRUM Team	Enseignant-chercheur au Département d'Informatique de l'UASZ
<b>Dr. Youssou DIENG</b>	SCRUM Team	Enseignant-chercheur au Département d'Informatique de l'UASZ

## **2. Le déroulement des événements du projet**

Nous avons appliqué la méthodologie scrum à notre manière vu que notre Product Owner (DRH) n'est pas tout le temps disponible. Malgré cela, nous avons eu à travailler au maximum avec le DRH en organisant des rencontres dès que possible dans son bureau. Et pour chaque rencontre, on vérifie le travail fait précédemment puis on se fixe un nouvel objectif. Chaque deux (02) semaines, on fait un compte-rendu au Scrum Master. Nous avons même eu une rencontre avec le Product Owner (DRH), le Scrum Master et le Scrum Team pour la vérification de l'avancement du projet.

Les méthodes agiles impliquent au maximum le demandeur (client) et permettent une grande réactivité à ses demandes. Le Framework Scrum propose la nécessité de développer par étapes courtes, d'inspecter à la fois l'efficacité des pratiques mises en œuvre et le produit qui en résulte, et adapter ainsi les objectifs du produit. Et tout cela est très important pour à la conception d'une application, qui est l'objet du Chapitre III.

## CHAPITRE III : SPECIFICATION ET ANALYSE DES BESOINS FONCTIONNELS

Dans ce chapitre, nous présenterons les objectifs de notre travail, ce qui nous amène à parler d'abord d'un Langage de modélisation qui s'appelle UML (Unified Modeling Language) qui se traduit par « Langage de modélisation unifié », c'est une norme du langage de modélisation objet. Elle nous permet de décrire de manière visuelle et graphique les besoins et, les solutions fonctionnelles et techniques d'un projet logiciel. On utilise pour cela des diagrammes. UML est une démarche qui se base sur une approche objet.

L'approche objet est centrée sur le diagramme de classes qui décrit aussi bien des actions que des informations dans une même entité. Les autres diagrammes nous aident à voir plus clair dans les besoins et dans la solution qui est à développer. Ils permettent de compléter le diagramme de classes.

Par la suite, nous traduisons les spécifications des besoins fonctionnels en diagrammes fonctionnels UML, ensuite nous allons parler de l'analyse des besoins et enfin, nous présenterons la modélisation de l'application.

### ***I- La spécification des besoins***

Cette phase consiste à comprendre le contexte du sujet. Il s'agit de déterminer les fonctionnalités et les acteurs les plus pertinents, et d'identifier les cas d'utilisation initiaux de notre application.

#### **1. Identification des acteurs du système**

Un acteur représente un rôle joué par une personne qui interagit avec le système. Il est principal quand il sollicite le système pour obtenir un résultat perceptible et est secondaire quand il est sollicité pour des informations complémentaires nécessaires au déroulement du cas d'utilisation. Les acteurs se recrutent parmi les utilisateurs du système et aussi parmi les responsables de sa configuration et de sa maintenance.

## Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

Dans le tableau, ci-dessous, nous allons définir les acteurs potentiels qui vont interagir avec l'application.

**Tableau 3 : Les acteurs de notre système**

<u>Acteurs</u>	<u>Descriptions</u>	<u>Rôles</u>
<b>DRH</b>	Directeur des Ressources Humaines	Gère l'administration des salaires
<b>CSDP</b>	Chef de la Division Soldes et Dépenses du Personnel	S'occupe du traitement individuelle des salaires, établit les états de paiement (les banques, organisationnelles, syndicat, coopérative)
<b>Administrateur</b>	Centre des ressources informatiques	Assure l'administration du système

## 2. Identification des fonctionnalités

Les besoins fonctionnels ou besoin métiers représentent les actions que le système doit exécuter, il ne devient opérationnel que s'ils sont satisfaits. Ainsi, suite à la description de la procédure d'obtention des bulletins de salaire décrite dans le chapitre 1, nous avons pu identifier un certain nombre de fonctionnalités que nous allons lister dans le tableau ci-dessous.

Dans le tableau suivant, chaque fois qu'on voit le mot « **Gérer** » signifie : **ajouter, modifier, afficher** et **supprimer**.

**Tableau 4 : Les fonctionnalités de notre système**

<u>Fonctionnalité du système</u>	<u>Module</u>	<u>Acteurs</u>
<b>S'Authentifier</b>		Tous les utilisateurs

Gérer un agent		DRH, CSDP
Affichage des agents qui bénéficient d'avancement automatique		DRH, CSDP
Gérer un grade		DRH, CSDP
Gérer un statut		DRH, CSDP
Gérer une fonction		DRH, CSDP
Gérer une situation familiale		CSDP, DRH
Gérer un service		CSDP, DRH
Gérer une banque		CSDP, DRH
Gérer une indemnité par statut		DRH
Gérer une indemnité par statut		DRH
Gérer une augmentation		DRH
Gérer une prime particulière		DRH
Gérer une prime spéciale		DRH
Gérer une retenue (ou cotisation)		DRH
Gérer un prestataire		DRH
Gérer un nombre de part		DRH
Attribuer grade		CSDP, DRH
Attribuer statut		CSDP, DRH
Attribuer fonction		CSDP, DRH
Attribuer prime particulière		CSDP, DRH

<b>Attribuer retenue ou cotisation</b>		CDSDP, DRH
<b>Attribuer situation familiale</b>		CDSDP, DRH
<b>Gérer un abattement</b>		DRH
<b>Gérer un impôt sur le revenu projeté (IRP)</b>		DRH
<b>Gérer une réduction d'impôts par charge de famille</b>		DRH
<b>Gérer un Trimf</b>		DRH
<b>Édition bulletin de paie provisoire</b>		DRH
<b>Validation bulletin de paie provisoire</b>		DRH
<b>Édition bulletin de paie finale</b>		DRH
<b>Imprimer le mandat, les états les cotisations.</b>		CDSDP, DRH
<b>Envoi par e-mail les bulletins de salaire au personnel</b>		CDSDP, DRH
<b>Envoi des sms au personnel</b>		CDSDP, DRH
<b>Ajouter, Lister, Modifier, Supprimer un utilisateur</b>		Administrateur

### **3. Diagrammes de cas d'utilisation**

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. Il donne une vue du système dans son environnement extérieur et définit la relation entre l'utilisateur et les éléments que le système met en œuvre.

Pour notre application, nous allons représenter deux diagrammes de cas d'utilisation, d'une part celui du CSDP et d'autre part celui du DRH.

## Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

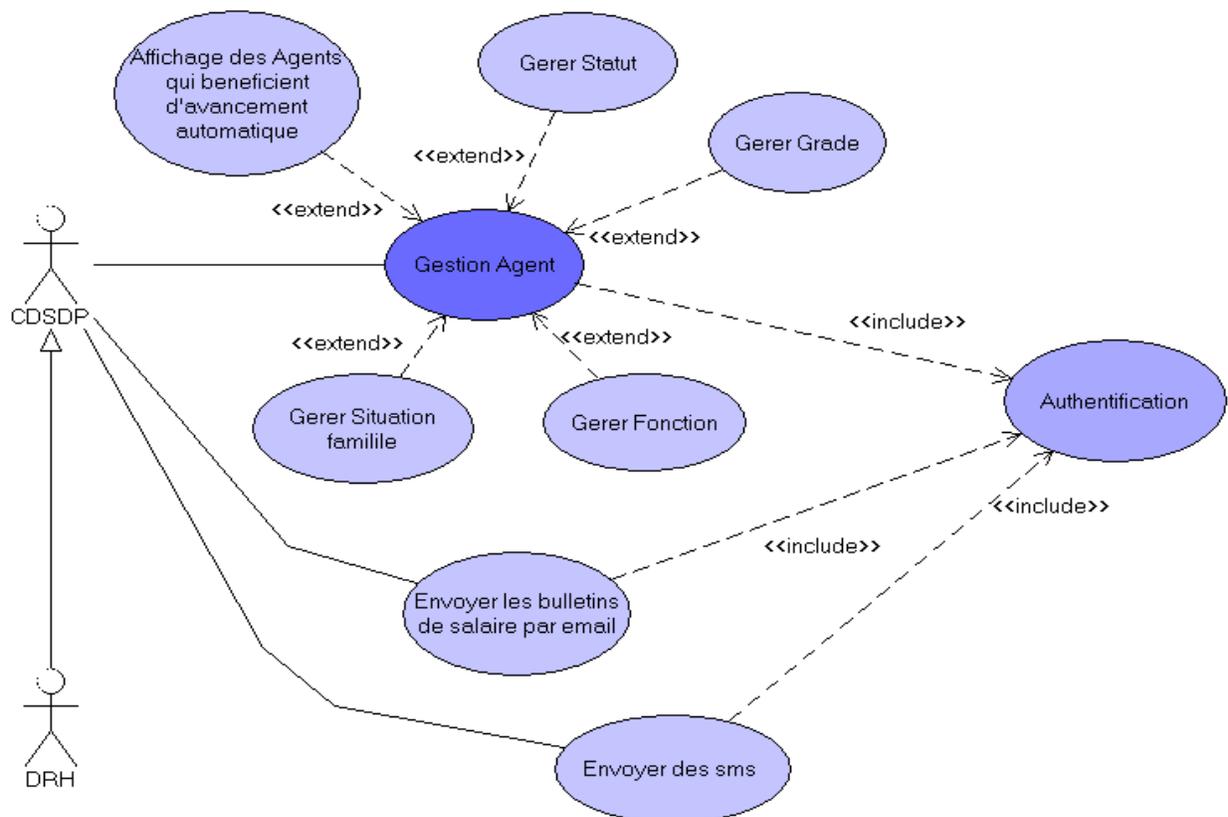
Dans les représentations de nos cas d'utilisation, chaque fois qu'on voit le mot « **gérer** » suivi de quelque chose dans un cas d'utilisation cela veut tout simplement dire, que ce cas d'utilisation se généralise pour les cas d'utilisation : **ajouter, modifier, afficher** et **supprimer**.

Par exemple pour le cas d'utilisation « **gérer statut** » on le généralise pour les cas d'utilisation :

- ❖ Ajouter statut
- ❖ Lister statut
- ❖ Modifier statut
- ❖ Supprimer statut

### a. Diagramme de cas d'utilisation du CSDSP

Le diagramme de cas d'utilisation du CSDSP permet de représenter les fonctionnalités auxquelles cet acteur a accès à notre système.

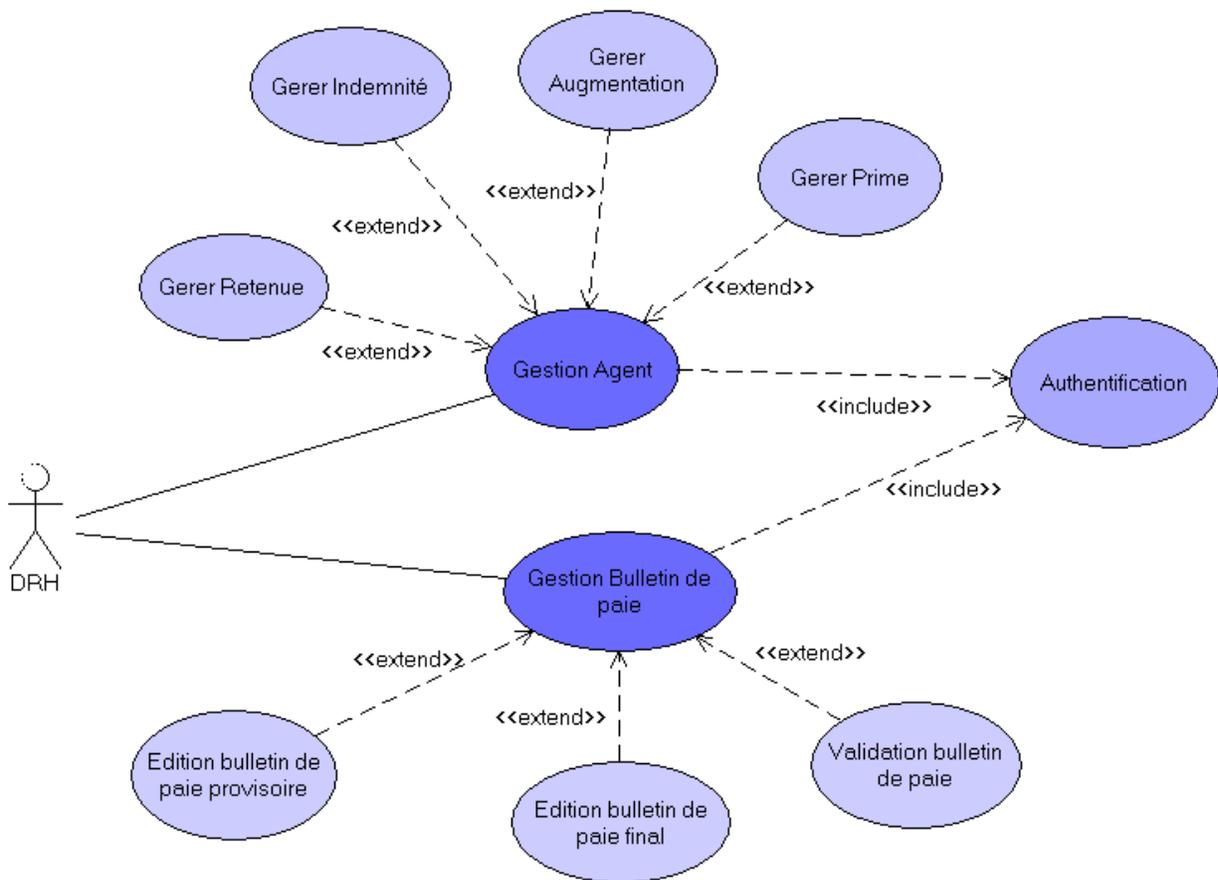


**Figure 2 : Diagramme de cas d'utilisation du CSDSP**

Dans le diagramme, ci-dessus, nous voyons que le CSDSP a beaucoup de fonctionnalités malgré qu’on ne les ait pas tous représentés dans ce diagramme. On note aussi la présence du DRH dans ce diagramme destiné au CSDSP, cela est du fait que tout ce que le CSDSP peut faire dans l’application le DRH peut le faire aussi, mais l’inverse n’est pas vrai, et cela, on va le voir dans le diagramme de cas d’utilisation suivante qui est celui du DRH.

***b. Diagramme de cas d’utilisation du DRH***

Le diagramme de cas d’utilisation du DRH que nous présentons ici montre quelques fonctionnalités que cet acteur manipule dans notre application.



***Figure 3 : Diagramme de cas d’utilisation du DRH***

Pour ce diagramme de cas d’utilisation, on voit le DRH en plus de s’occuper de toutes les fonctionnalités qui touchent l’agent, il est aussi responsable de toutes les fonctionnalités qui se

rapportent au bulletin de paie du personnel. Donc on peut dire qu'il est l'élément central de notre application.

#### 4. Description de cas d'utilisation

Dans cette partie, nous avons choisi de décrire quelques cas d'utilisation.

##### *a. Description des cas d'utilisation « s'authentifier »*

Le tableau suivant permet de décrire le cas d'utilisation de l'authentification.

**Tableau 5 : Description du cas d'utilisation « s'authentifier »**

Description du cas d'utilisation « s'authentifier »	
<b>Titre</b>	S'authentifier
<b>Résumé</b>	Ce cas permet d'accéder au système.
<b>Acteur (s)</b>	DRH, CSDSP, Administrateur
<b>Pré condition</b>	chaque utilisateur doit avoir un compte.
<b>Scénario nominal</b>	<ol style="list-style-type: none"><li>1. Le système affiche le formulaire de connexion</li><li>2. L'utilisateur saisit son login et son mot passe.</li><li>3. Le système vérifie les informations saisies par l'utilisateur, et renvoie vers sa page d'accueil selon son profil</li></ol>
<b>Post condition</b>	Accéder à la page des fonctionnalités selon le profil de l'utilisateur
<b>Exception</b>	utilisateur saisit un login ou un mot de passe incorrecte.

##### *b. Description des cas d'utilisation « ajouter agent »*

Le tableau suivant permet de décrire le cas d'utilisation de l'ajout d'un agent

**Tableau 6 : Description du cas d'utilisation « ajouter agent »**

Description du cas d'utilisation « ajouter agent »	
<b>Titre</b>	Ajouter agent

<b>Résumé</b>	Permettre d'ajouter des agents
<b>Acteur (s)</b>	DRH
<b>Pré condition</b>	Affichage de la page d'accueil du DRH
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur clique sur le bouton « ajouter agent »</li> <li>2. Le système lui affiche le pop-up d'ajout d'un agent</li> <li>3. L'utilisateur renseigne les informations de l'agent puis clique sur « valider »</li> <li>4. Le système vérifie les informations saisies puis les enregistre.</li> </ol>
<b>Post condition</b>	Message de confirmation de l'enregistrement effectué
<b>Exception</b>	les données saisies sont invalides.

**c. Description des cas d'utilisation « modifier agent »**

Le tableau suivant permet de décrire le cas d'utilisation de la modification d'un agent.

**Tableau 7 : Description du cas d'utilisation « modifier agent »**

<b>Description du cas d'utilisation « modifier agent »</b>	
<b>Titre</b>	Modifier agent
<b>Résumé</b>	Permettre de modifier les données d'un agent
<b>Acteur (s)</b>	CDS DP
<b>Pré condition</b>	Authentification, affichage des agents du système.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur recherche l'agent par (nom, matricule, ...)</li> <li>2. Le système lui affiche l'agent</li> <li>3. L'utilisateur clique sur « détails »</li> <li>4. Le système lui redirige vers la page « Agent »</li> <li>5. L'utilisateur clique sur la partie à modifier puis renseigne les informations à modifier</li> <li>6. L'utilisateur clique sur le bouton « modifier »</li> <li>7. Le système vérifie les informations puis les enregistre</li> </ol>
<b>Post condition</b>	Message de confirmation de l'opération

<b>Exception</b>	les données saisies sont incorrectes.
------------------	---------------------------------------

***d. Description des cas d'utilisation « désactiver agent »***

Le tableau suivant permet de décrire le cas d'utilisation de la désactivation d'un agent.

***Tableau 8 : Description du cas d'utilisation « désactiver agent »***

<b>Description du cas d'utilisation « Désactiver un agent »</b>	
<b>Titre</b>	Désactiver un agent
<b>Résumé</b>	Il permet de bloquer un agent non-actif dans le système
<b>Acteur (s)</b>	DRH
<b>Pré condition</b>	Authentification, affichage des agents du système.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur recherche l'agent par (nom, matricule, ...)</li> <li>2. Le système lui affiche l'agent</li> <li>3. L'utilisateur clique sur « désactiver »</li> <li>4. Le système lui envoie un pop-up de confirmation</li> <li>5. L'utilisateur clique sur le bouton « confirmer »</li> <li>6. Le système bloque l'agent</li> </ol>
<b>Post condition</b>	Une possibilité de l'activer encore
<b>Exception</b>	Problème lié à la connexion

***e. Description des cas d'utilisation « Attribuer un grade »***

Le tableau suivant permet de décrire le cas d'utilisation d'attribution d'un grade à un agent.

***Tableau 9 : Description du cas d'utilisation « attribuer grade »***

<b>Description du cas d'utilisation « Attribuer grade »</b>	
<b>Titre</b>	Attribuer grade
<b>Résumé</b>	Il consiste à attribuer un grade à un agent
<b>Acteur (s)</b>	DRH, CSDP

<b>Pré condition</b>	Authentification, création des grades, affichage des agents du système.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur recherche l'agent par matricule</li> <li>2. Le système lui affiche l'agent</li> <li>3. L'utilisateur clique sur « grade »</li> <li>4. Le système lui affiche un formulaire qui contient tous les grades</li> <li>5. L'utilisateur choisit le grade puis clique sur le bouton « valider »</li> <li>6. Le système vérifie les informations puis les enregistre</li> </ol>
<b>Exception</b>	Problème lié à la connexion

***f. Description des cas d'utilisation « Affichage des différents agents qui bénéficient d'avancement automatique »***

Le tableau suivant permet de décrire le cas d'utilisation de l'affichage des différents agents qui bénéficient d'avancement automatique.

***Tableau 10 : Description du cas d'utilisation « affichage des différents agents qui bénéficient d'avancement automatique »***

<b>Description du cas d'utilisation « Affichage des différents agents qui bénéficient d'avancement automatique »</b>	
<b>Titre</b>	Affichage des différents agents qui bénéficient d'avancement automatique
<b>Résumé</b>	Il permet d'afficher les agents qui doivent bénéficier d'avancement automatique sur leur grade
<b>Acteur (s)</b>	DRH, CDSDP
<b>Pré condition</b>	Authentification
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur clique sur la barre des menus et choisit l'option « Affichage d'agents qui bénéficient d'avancement automatique »</li> <li>2. Le système lui affiche tous les agents</li> </ol>
<b>Exception</b>	Problème lié à la connexion

**g. Description des cas d'utilisation « Edition des bulletins de paie provisoire »**

Le tableau suivant permet de décrire le cas d'utilisation de l'édition des bulletins de paie provisoire.

**Tableau 11 : Description du cas d'utilisation « édition des bulletins de paie provisoire »**

<b>Description du cas d'utilisation « Édition bulletin de paie provisoire »</b>	
<b>Titre</b>	Édition bulletin de paie provisoire
<b>Résumé</b>	Il permet d'éditer les bulletins de paie provisoire de tous les agents
<b>Acteur (s)</b>	<b>DRH</b>
<b>Pré condition</b>	Authentification
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur choisit l'option « bulletin provisoire » puis clique sur « ajouter »</li> <li>2. Le système lui affiche un pop-up d'édition des bulletins de paie provisoire</li> <li>3. L'utilisateur renseigne les informations (mois, ....) puis clique sur « valider »</li> <li>4. Le système vérifie les informations puis les enregistre</li> </ol>
<b>Post condition</b>	Message de confirmation de l'opération effectuée
<b>Exception</b>	<b>Problème lié à la connexion</b>

**h. Description des cas d'utilisation « validation bulletin de paie provisoire »**

Le tableau suivant permet de décrire le cas d'utilisation de la validation d'un bulletin de paie provisoire.

**Tableau 13 : Description du cas d'utilisation « validation des bulletins de paie provisoire »**

Description du cas d'utilisation « Validation bulletin de paie provisoire »	
<b>Titre</b>	Validation ou non du bulletin de paie provisoire
<b>Résumé</b>	Il permet de valider les bulletins de paie provisoire
<b>Acteur (s)</b>	DRH
<b>Pré condition</b>	Authentification, Affichage des bulletins de paie provisoire
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur clique sur l'agent</li> <li>2. Le système lui affiche un pop-up avec toutes les informations du bulletin de paie de l'agent</li> <li>3. L'utilisateur clique sur « valider »</li> <li>4. Le système enregistre puis colorie l'agent en couleur verte</li> </ol>
<b>Post condition</b>	Message de confirmation de l'opération effectuée
<b>Exception</b>	Problème lié à la connexion

***i. Description des cas d'utilisation « Edition bulletin de paie finale »***

Le tableau suivant permet de décrire le cas d'utilisation de l'édition d'un bulletin de paie finale.

**Tableau 14 : Description du cas d'utilisation « édition bulletin de paie finale »**

Description du cas d'utilisation « Édition bulletin de paie finale »	
<b>Titre</b>	Édition bulletin de paie
<b>Résumé</b>	Il permet d'éditer les bulletins de paie finale
<b>Acteur (s)</b>	DRH
<b>Pré condition</b>	Authentification, Validation des bulletins provisoires
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur choisit l'option « bulletin final » puis clique sur « ajouter »</li> <li>2. Le système lui affiche un pop-up d'édition des bulletins de paie finale</li> <li>3. L'utilisateur renseigne les informations (mois, ....) puis clique sur « valider »</li> <li>4. Le système vérifie les informations puis les enregistre</li> </ol>

**Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»**

<b>Post condition</b>	Message de confirmation de l'opération effectuée
<b>Exception</b>	Problème lié à la connexion

***j. Description des cas d'utilisation « Envoi par email les bulletins de salaire »***

Le tableau suivant permet de décrire le cas d'utilisation de l'envoi par email les bulletins de salaire.

***Tableau 15 : Description du cas d'utilisation « envoi par email les bulletins de salaire »***

<b>Description du cas d'utilisation «Envoi par email les bulletins de salaires»</b>	
<b>Titre</b>	Envoi par email les bulletins de salaire
<b>Résumé</b>	Permettre d'envoyer par email les bulletins de salaires au personnel
<b>Acteur (s)</b>	CSDSP, DRH
<b>Pré condition</b>	Authentification, Affichage des bulletins de paie finale
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur clique sur le menu déroulant puis choisit l'option « envoyer mail »</li> <li>2. Le système lui affiche tous les agents qui bénéficient des bulletins de paie finale</li> <li>3. L'utilisateur choisit l'option « envoie mail groupé »</li> <li>4. Le système lui envoie un pop-up de confirmation</li> <li>5. L'utilisateur clique sur « valider »</li> <li>6. Le système envoie les mails aux agents avec comme pièce jointes leur bulletin de salaire</li> </ol>
<b>Post condition</b>	La liste des agents dont les emails ont été envoyés
<b>Exception</b>	Problème lié à la connexion

***k. Description des cas d'utilisation « Envoi des sms »***

Le tableau suivant permet de décrire le cas d'utilisation de l'envoi des sms.

***Tableau 16 : Description du cas d'utilisation « envoi des sms »***

Description du cas d'utilisation « Envoi des sms »	
<b>Titre</b>	Envoi des sms au personnel
<b>Résumé</b>	Permettre d'envoyer des sms au personnel pour leur prévenir de la disponibilité de leur salaire
<b>Acteur (s)</b>	CSDSP, DRH
<b>Pré condition</b>	Authentification, Affichage des bulletins de paie finale, Envoi par mail les bulletins de salaire
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur clique sur le menu déroulant puis choisit l'option « envoyer sms »</li> <li>2. Le système lui affiche tous les agents qui bénéficient des bulletins de paie finale</li> <li>3. L'utilisateur choisit l'option « envoie sms groupé »</li> <li>4. Le système lui envoie un pop-up de confirmation</li> <li>5. L'utilisateur clique sur « valider »</li> <li>6. Le système envoie les sms dans les téléphones des agents</li> </ol>
<b>Post condition</b>	Message de confirmation de l'opération effectuée
<b>Exception</b>	Problème lié à la connexion

## **II- Analyse des besoins fonctionnels du système**

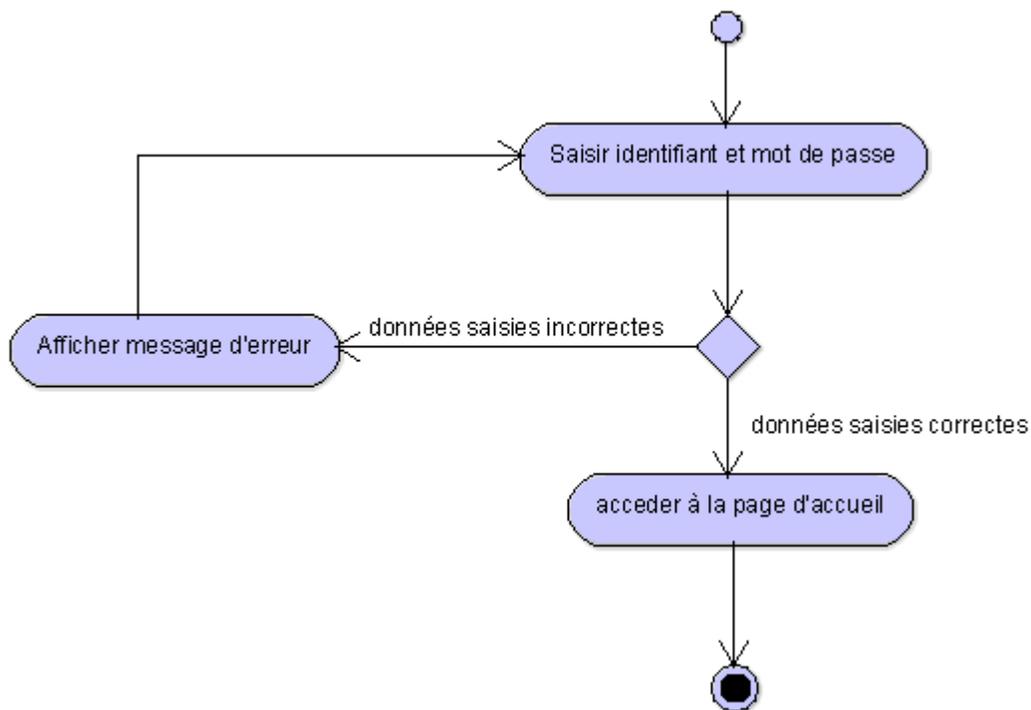
Dans cette partie, nous allons faire l'analyse du système que nous voulons mettre en place, en décrivant quelques fonctionnalités importantes. Cependant, pour chaque cas d'utilisation décrite, nous allons proposer un diagramme d'activités et un diagramme de séquence.

### **1. Les activités de l'authentification**

Le cas d'utilisation « authentifier » est composé de plusieurs enchaînements que l'on va représenter dans le diagramme d'activité et parmi ces scénarios on va choisir un seul pour créer le diagramme de séquence.

#### **a. Diagramme d'activités du cas « authentifier »**

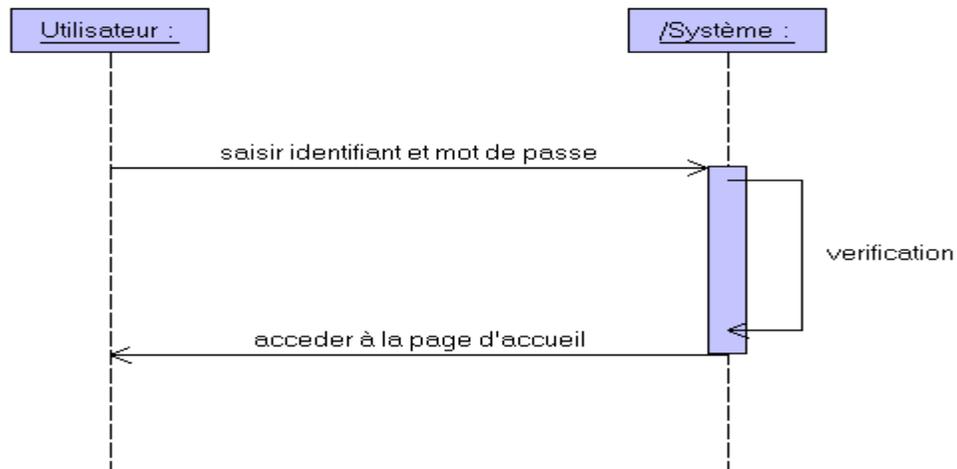
L'utilisateur saisit son login puis son mot de passe, le système vérifie les informations saisies si elles ne sont pas correctes un message d'erreur est envoyé à l'utilisateur et ensuite, il pourra saisir de nouveau son login ou son mot de passe. Et si les informations saisies sont correctes, le système lui affichera la page d'accueil.



***Figure 4 : Diagramme d'activités du cas « s'authentifier »***

***b. Diagramme de séquence du cas « authentifier »***

L'utilisateur saisit son login et son mot de passe, le système vérifie les informations saisies avant d'afficher la page d'accueil.



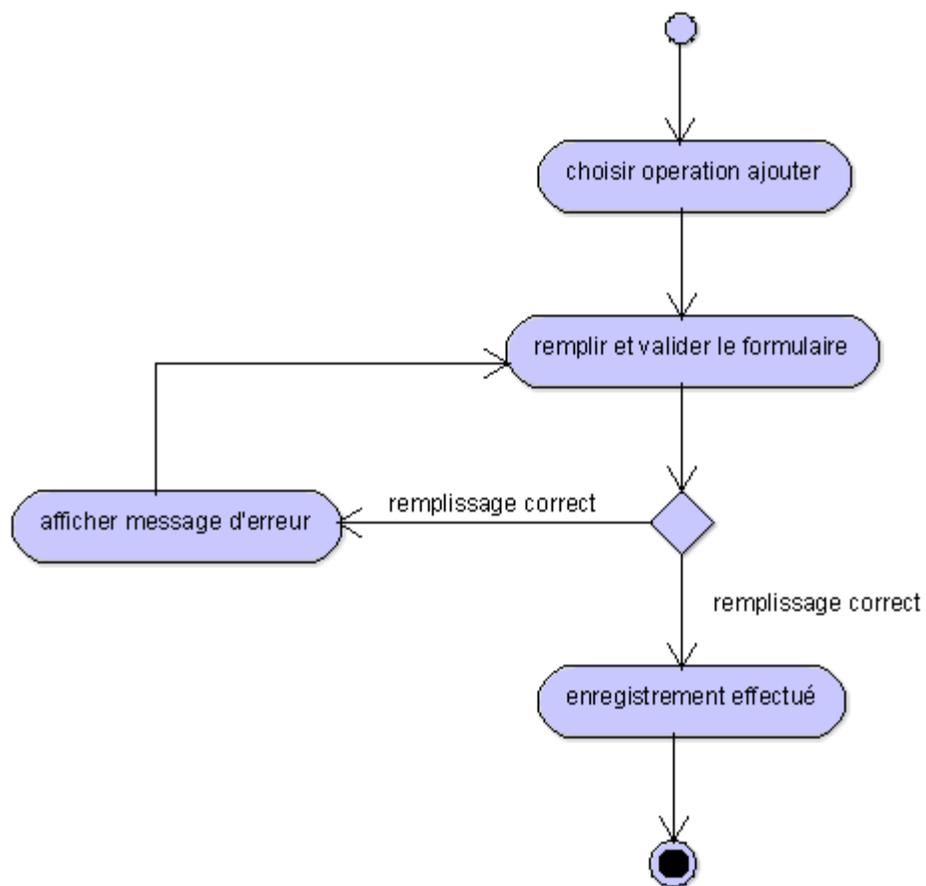
**Figure 5 : Diagramme de séquence du cas « s'authentifier »**

## **2. Les activités de l'ajout d'un agent**

Pour représenter les activités du cas d'utilisation « ajouter agent », nous allons d'abord proposer un diagramme d'activités qui va illustrer les enchaînements possibles pour ce cas, puis on va ensuite prendre un scénario pour réaliser le diagramme de séquence.

### **a. Le diagramme d'activité du cas « ajouter agent »**

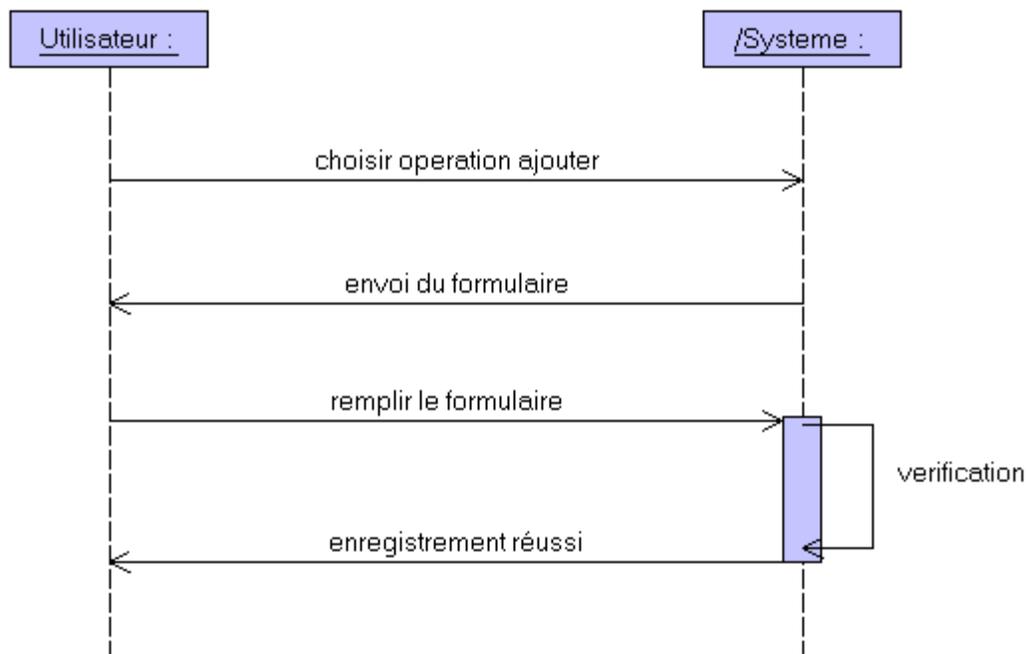
L'utilisateur choisit l'opération « ajouter agent ». Le système lui fournit un formulaire. Il remplit et valide le formulaire. Le système vérifie le remplissage du formulaire, si ce dernier est incorrect le formulaire est réaffiché avec un message d'erreur et s'il est correct le système enregistre les informations saisies.



***Figure 6 : diagramme d'activités du cas « ajouter agent »***

***b. Diagramme de séquence du cas « ajout agent »***

L'utilisateur est sur la page d'accueil et choisi l'opération « ajout un agent ». Le système lui affiche un formulaire d'ajout. Il remplit le formulaire et le valide, le système vérifie les informations saisies et les enregistrées.



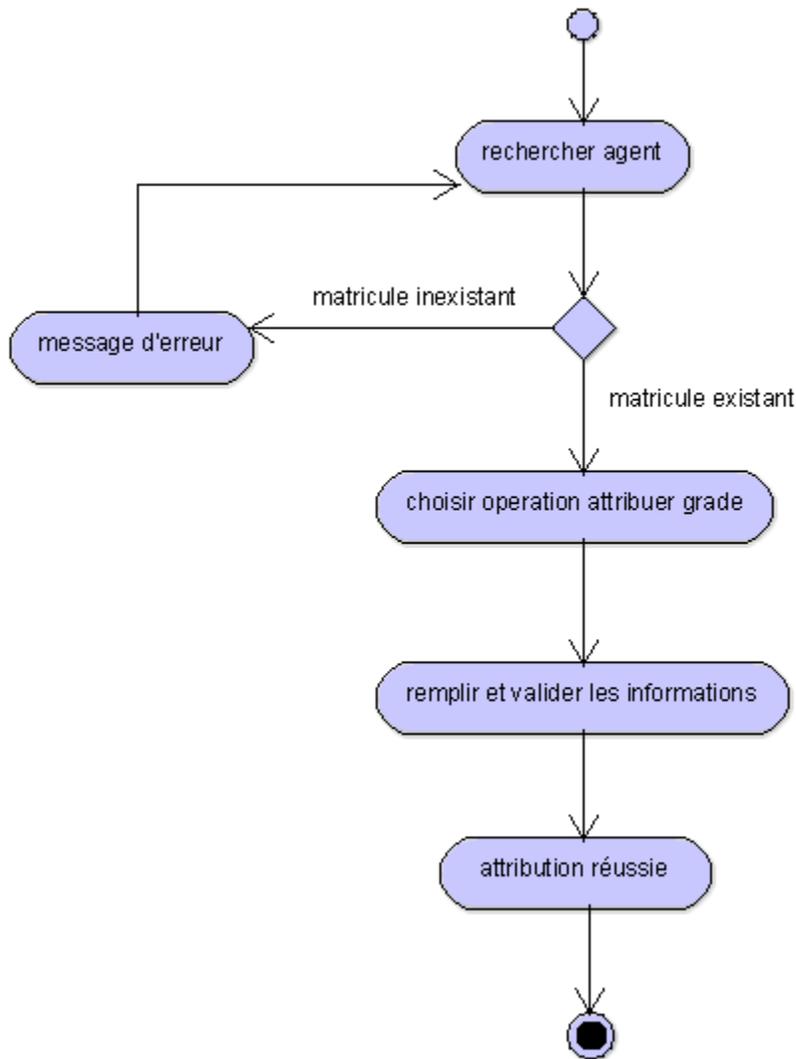
**Figure 7 :** Diagramme de séquence du cas « ajouter agent »

### **3. Les activités de l'attribution d'un grade à un agent**

Les activités de la fonctionnalité « attribuer grade » se décomposeront comme suit: d'abord, nous proposons un diagramme d'activité qui décrira tous les scénarios possibles puis nous allons prendre un de ces scénarios pour faire le diagramme de séquence.

#### **a. Diagramme d'activités du cas « attribuer un grade »**

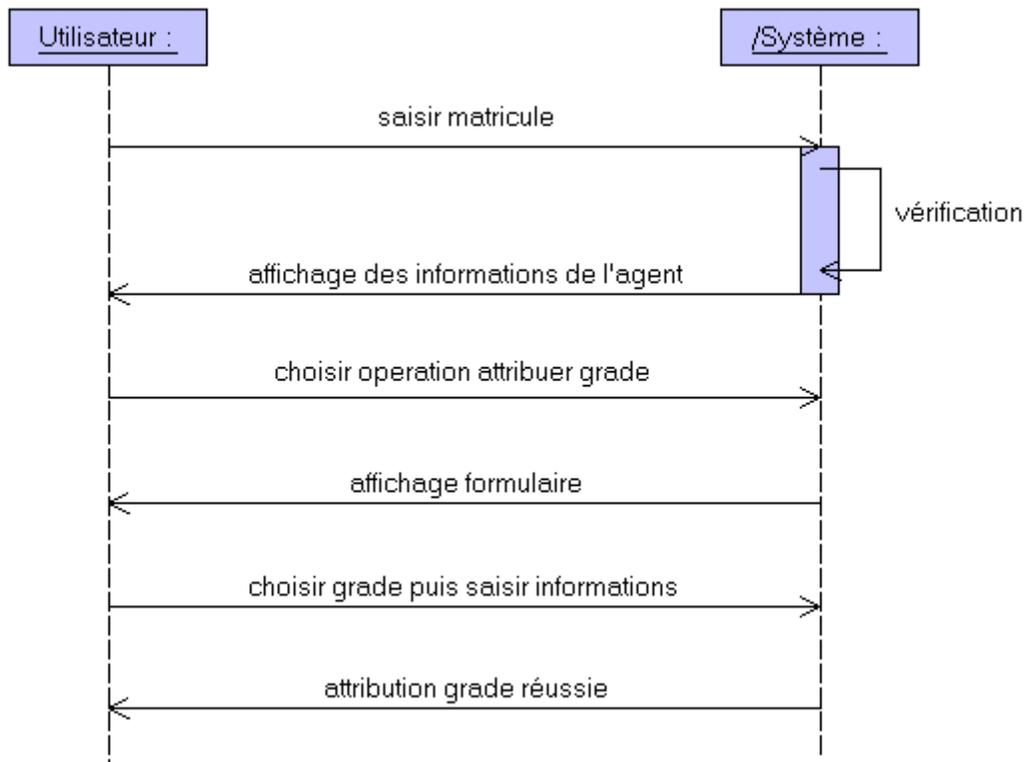
Pour effectuer l'attribution d'un grade à un agent, l'utilisateur accède à la page d'accueil et il donne l'identifiant (matricule) de l'agent. Ce dernier doit impérativement exister dans la base de données. S'il donne par exemple le bon matricule, les informations de l'agent s'affichent. Il choisit l'opération « attribuer grade », alors le système lui affiche un formulaire de tous les grades disponibles. Il choisit un puis valide. Le système procède à la vérification des données. Si elles sont conformes aux règles établies alors l'attribution est prise en compte sinon l'opération échoue.



**Figure 8 :** Diagramme d'activités du cas « attribuer grade »

**b. Diagramme de séquence du cas « attribuer grade »**

L'utilisateur accède à la page d'accueil puis saisit le matricule de l'agent. Le système vérifie le matricule et affiche les informations supplémentaires de l'agent. L'utilisateur choisit l'opération « attribuer grade ». Le système lui affiche un formulaire avec tous les grades disponibles. L'utilisateur choisit un grade puis valide les autres informations. Le système procède à une vérification des données remplies et les intègre. Un message de confirmation de l'attribution de grade est affiché à l'utilisateur.



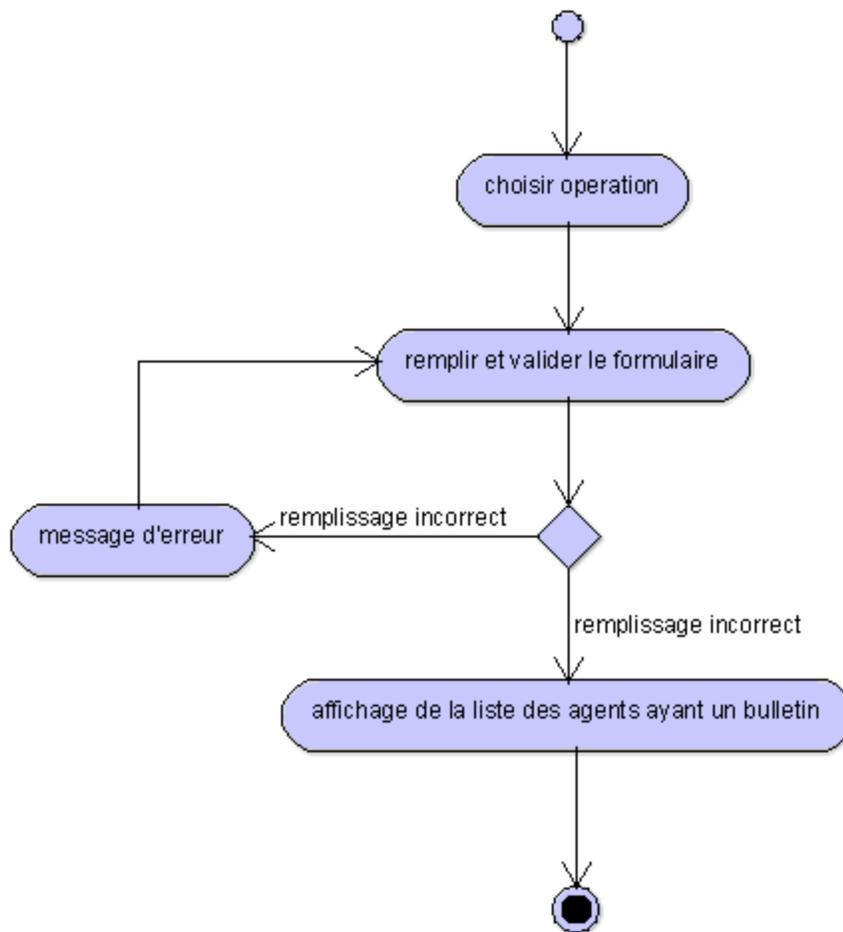
**Figure 9 : Diagramme de séquence du cas « attribuer grade »**

#### **4. Les activités de l'édition d'un bulletin de paie provisoire**

Le cas d'utilisation « éditer un bulletin de paie provisoire » est composé de plusieurs enchaînements que l'on va représenter dans le diagramme d'activité et parmi ces scénarios on va choisir un seul pour créer le diagramme de séquence.

##### **a. Diagramme d'activités du cas « éditer un bulletin de paie provisoire »**

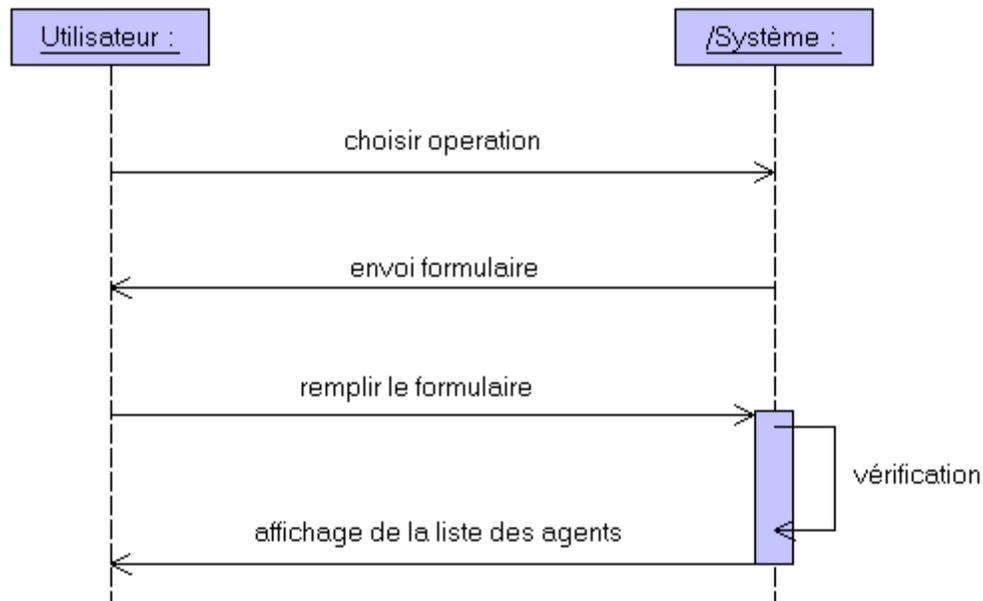
Le DRH qui est le seul utilisateur à pouvoir accéder à cette fonctionnalité choisie l'opération « éditer bulletin de paie provisoire ». Le système lui fournit un formulaire. Il remplit et valide le formulaire. Le système vérifie le remplissage du formulaire, si ce dernier est incorrect le formulaire est réaffiché avec un message d'erreur, et s'il est correct le système enregistre les informations saisies puis lui affiche la liste de tous les agents bénéficiant d'un bulletin provisoire.



**Figure 10 : Diagramme d'activités du cas « éditer bulletin de paie provisoire »**

**b. Diagramme de séquence du cas « éditer un bulletin de paie provisoire »**

L'utilisateur est sur la page d'accueil et choisi l'opération « éditer un bulletin de paie provisoire ». Le système lui affiche un formulaire. Il remplit le formulaire et le valide, le système vérifie les informations saisies et les enregistre, puis affiche la liste des agents ayant leur bulletin provisoire.



**Figure 11** : Diagramme de séquence du cas « éditer bulletin de paie provisoire »

Ce chapitre a permis de faire les spécifications et l'analyse des besoins fonctionnels du système « Gestion automatique des bulletins de salaire de l'UASZ ». Le chapitre suivant entame une autre étape importante qui est la conception de notre application.

## CHAPITRE IV : CONCEPTION DU SYSTEME

La conception de notre système constitue la phase essentielle de celui-ci, car elle fait partie des phases les plus importantes pour la concrétisation d'une application informatique. Ainsi, dans ce chapitre, nous parlons de la conception générale de notre application qui s'appuie sur l'architecture d'un logiciel, le diagramme de package et le diagramme de déploiement. Puis on abordera la conception détaillée de l'application avec les diagrammes de classes participant aux fonctionnalités et nous proposons un dictionnaire de données.

### *I- Conception générale*

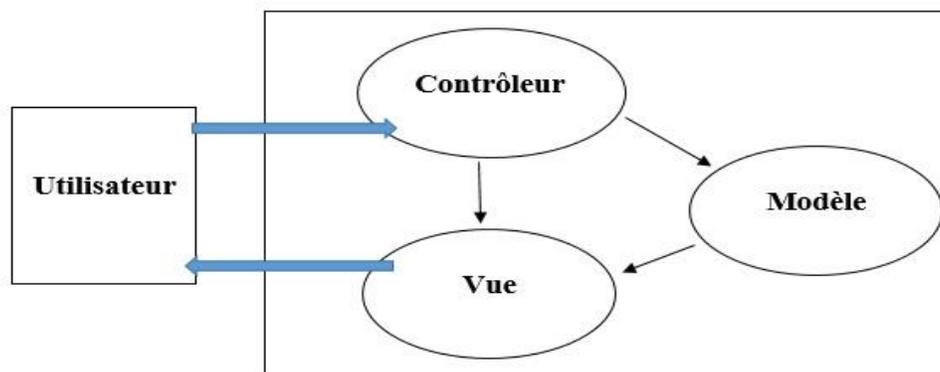
#### **1. Architecture du système**

Dans le développement logiciel en général et dans le développement web en particulier, les mots MVC et 3-Tiers sont souvent cités dans les cours de génie logiciel. Et les gens ont tendance à les confondre, mais ils n'ont pas tort car ces deux pratiques sont à la fois différentes et similaires.

##### *a. Modèle MVC*

MVC (Modèle-Vue-Contrôleur) est un modèle de conception qui impose la séparation entre données, traitements et présentation. C'est pour cette raison que l'application est divisée en trois composants fondamentaux : le modèle, la vue et le contrôleur. Chacun de ces composants tient un rôle bien défini. Ainsi, ce modèle se décompose en trois composants :

- ❖ **Le Modèle** : décrit les données manipulées par l'application et définit les méthodes d'accès.
- ❖ **La Vue** : définit l'interface utilisateur et la présentation.
- ❖ **Le Contrôleur** : prend en charge la gestion des événements de synchronisation pour mettre à jour la vue ou le modèle. [5]

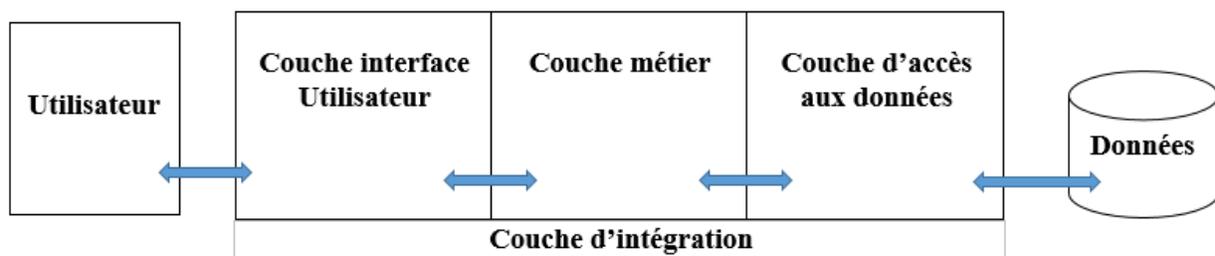


**Figure 12** : schéma du modèle MVC

### **b. Architecture 3 tiers**

L'architecture 3-Tiers est un modèle logique d'architecture applicative qui vise à séparer très nettement trois couches logicielles au sein d'une même application, à modéliser et à présenter cette application comme un empilement de trois couches, étages ou niveaux dont le rôle est clairement défini :

- ❖ **La présentation des données (couche Web)**: correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec utilisateur.
- ❖ **Le traitement métier des données (couche Métier)**: correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative.
- ❖ **L'accès aux données persistantes (couche DAO)** : correspondant aux données qui sont destinées à être conservées sur une durée ou de manière définitive (stockage sur un support physique). [5]

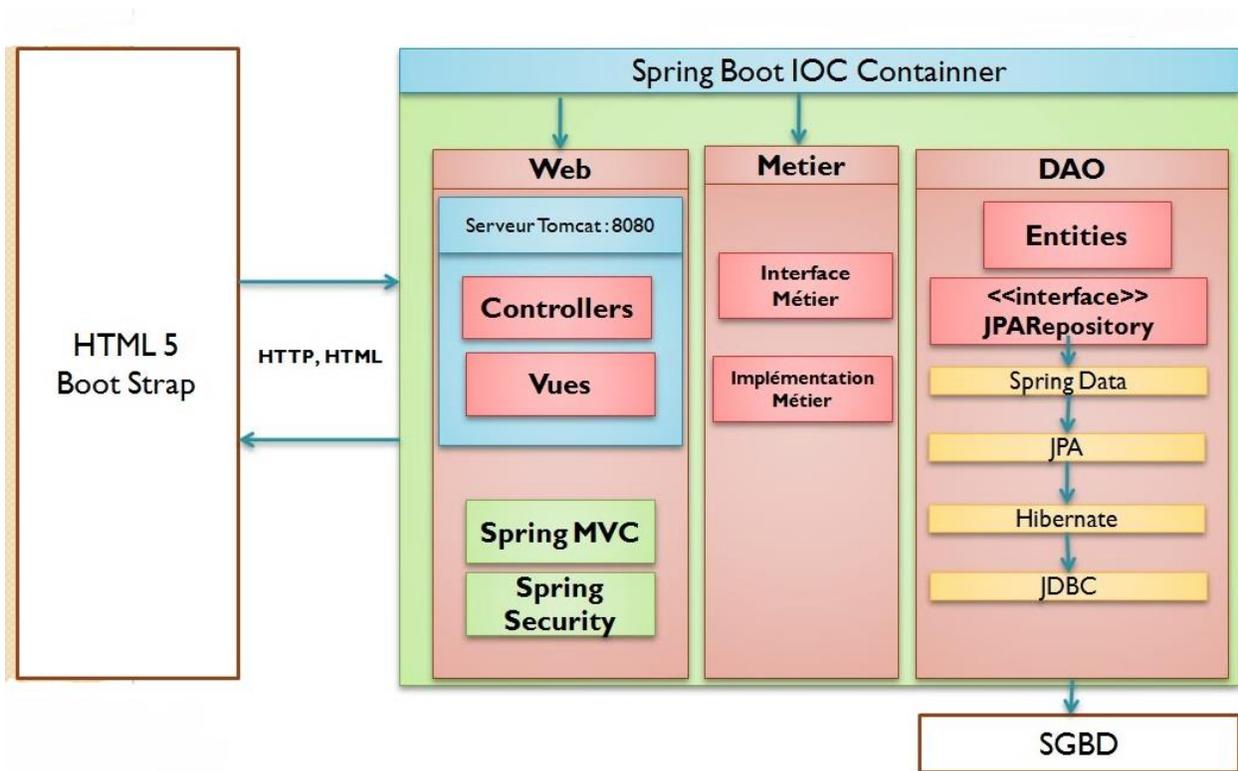


**Figure 13** : Schéma de l'architecture 3 tiers

Une telle architecture, appelée 3-Tiers ou à 3 niveaux cherche à respecter le modèle MVC.

Pour qu'une application 3-Tiers respecte le modèle MVC, il faut lui ajouter une *couche de contrôle* entre *l'interface utilisateur* et la *couche métier*.

L'architecture web de notre futur système sera détaillée dans le schéma à trois couches de la façon suivante :



**Figure 14 :** Schéma de l'architecture de notre système [6]

## 2. Diagramme de package

Un diagramme de packages est un diagramme UML qui fournit une représentation graphique de haut niveau de l'organisation d'une application, et nous aide à identifier les liens de généralisation et de dépendance entre les packages. La représentation ci-dessous est le diagramme de package de notre système.

**Authentication** : elle utilise la gestion des profils pour distinguer et vérifier les utilisateurs qui se connectent.

**Gestion des profils** : dans ce package des fonctionnalités telles qu' « ajouter un utilisateur », « modifier un utilisateur », « supprimer un utilisateur » ou « afficher la liste des utilisateurs » sont développés.

**Gestion des menus** : toutes les interfaces à partir desquelles les utilisateurs interagissent avec le système sont contenues dans ce paquetage.

**Gestion des agents** : toutes les fonctionnalités concernant les agents comme « ajouter un agent », « modifier un agent », « désactiver un agent » ou « afficher la liste des agents » y sont développées.

**Gestion des Informations de salaire** : ce package est composé par d'autres packages qui sont :

- ❖ **Statut** : dans ce package, on a les fonctionnalités « ajouter un statut », « modifier un statut », « supprimer un statut » ou « afficher la liste des statuts »
- ❖ **Fonction** : qui est constituée des fonctionnalités « ajouter une fonction », « modifier une fonction », « supprimer une fonction » ou « afficher la liste des fonctions »
- ❖ **Situation familiale** : dans ce package, on a les fonctionnalités « ajouter une situation familiale », « modifier une situation familiale », « supprimer une situation familiale » ou « afficher la liste des situations familiales »
- ❖ **Banque** : qui est constituée des fonctionnalités « ajouter une banque », « modifier une banque », « supprimer une banque » ou « afficher la liste des banques »

**Gestion des Éléments de salaire** : ce package est composé des fonctionnalités comme « ajouter un grade », « modifier un grade », « supprimer un grade » ou « afficher la liste des grades ».

**Gestion des Primes** : ce package est composé par d'autres packages qui sont :

- ❖ **Prime particulière** : dans ce package, on a les fonctionnalités « ajouter une prime particulière », « modifier une prime particulière », « supprimer une prime particulière » ou « afficher la liste des primes particulières »
- ❖ **Prime spéciale** : qui est constitué des fonctionnalités « ajouter une prime spéciale », « modifier une prime spéciale », « supprimer une prime spéciale » ou « afficher la liste des primes spéciales »

## Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

- ❖ **Indemnité** : dans ce package, on a les fonctionnalités « ajouter une indemnité », « modifier une indemnité », « supprimer une indemnité » ou « afficher la liste des indemnités »
- ❖ **Augmentation** : qui est constitué des fonctionnalités « ajouter une augmentation », « modifier une augmentation », « supprimer une augmentation » ou « afficher la liste des augmentations »

**Gestion des Retenues** : ce package est composé par d'autres packages qui sont :

- ❖ **Cotisation** : dans ce package, on a les fonctionnalités « ajouter une prime cotisation », « modifier une cotisation », « supprimer une cotisation » ou « afficher la liste des cotisations »
- ❖ **Prestataire** : qui est constitué des fonctionnalités « ajouter un prestataire », « modifier un prestataire », « supprimer un prestataire » ou « afficher la liste des prestataires »

**Gestion des Impôts** : ce package est composé par d'autres packages qui sont :

- ❖ **Abattement** : dans ce package, on a les fonctionnalités « ajouter un abattement », « modifier un abattement », « supprimer un abattement » ou « afficher la liste des abattements »
- ❖ **Trimf** : qui est constitué des fonctionnalités « ajouter un Trimf », « modifier un Trimf », « supprimer un Trimf » ou « afficher la liste des Trimf »
- ❖ **Impôt RP (Revenu Projeté)** : dans ce package, on a les fonctionnalités « ajouter un impôt RP », « modifier un impôt RP », « supprimer un impôt RP » ou « afficher la liste des impôts RP »
- ❖ **Réduction ICF (Impôt par Charge de Famille)** : qui est constitué des fonctionnalités « ajouter une réduction ICF », « modifier une réduction ICF », « supprimer une réduction ICF » ou « afficher la liste des réductions ICF »

**Gestion Bulletin de salaire** : ce package est composé des fonctionnalités comme « éditer un bulletin de salaire provisoire », « éditer un bulletin de salaire final », « valider un bulletin de salaire », « envoyer bulletin par e-mail », « envoyer sms », « archiver bulletin de salaire ».

Dans la figure suivante, nous avons la représentation du diagramme de package.

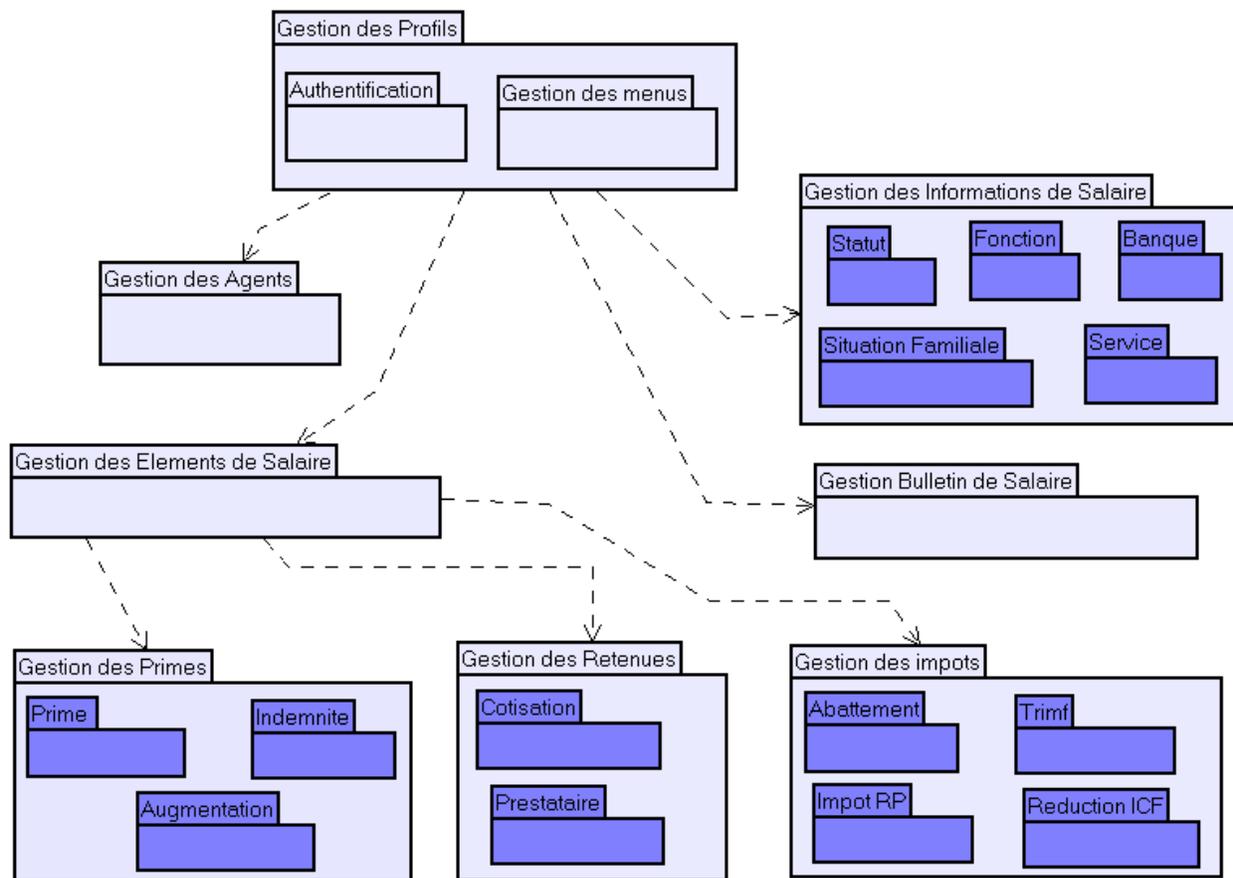
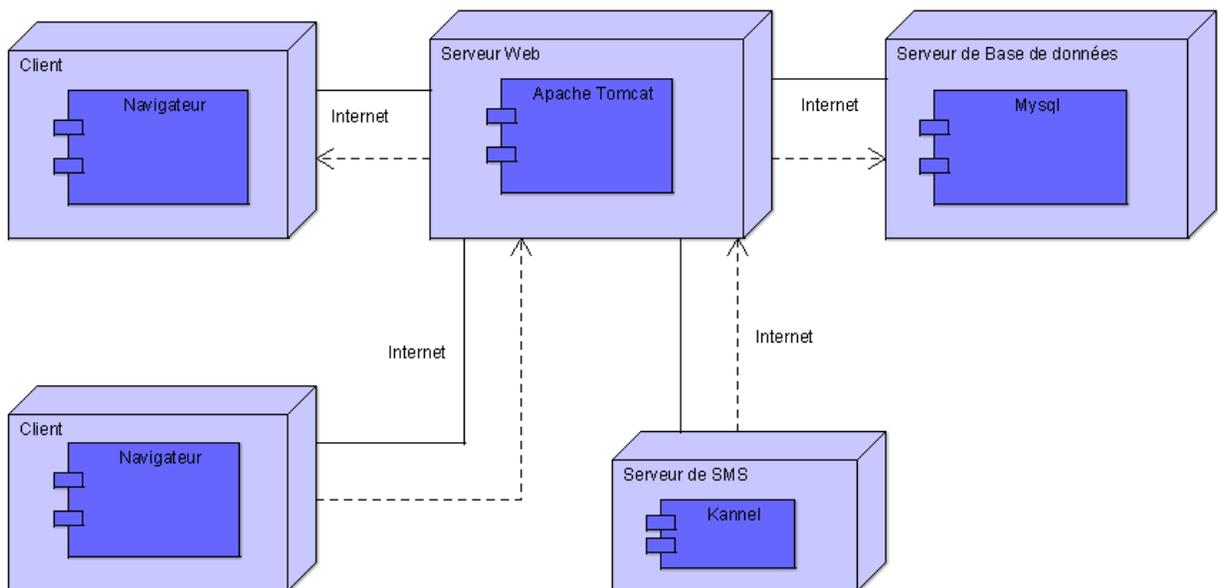


Figure 15 : Diagramme de package

### 3. Diagramme de déploiement

Le diagramme de déploiement permet de représenter l'architecture physique supportant l'exploitation du système. Cette architecture comprend des nœuds correspondant aux supports physiques (serveurs, routeurs...) ainsi que la répartition des artefacts logiciels (bibliothèques, exécutables...) sur ces nœuds. De ce fait, notre application sera déployée sur des machines de 2Go de RAM et 2.00GHZ de vitesse processeur pour la plupart. Les serveurs sont contenus dans des machines de 8Go de RAM. Ils empruntent le réseau internet.

Dans la représentation, ci-dessous, nous allons illustrer le diagramme de déploiement de notre application.



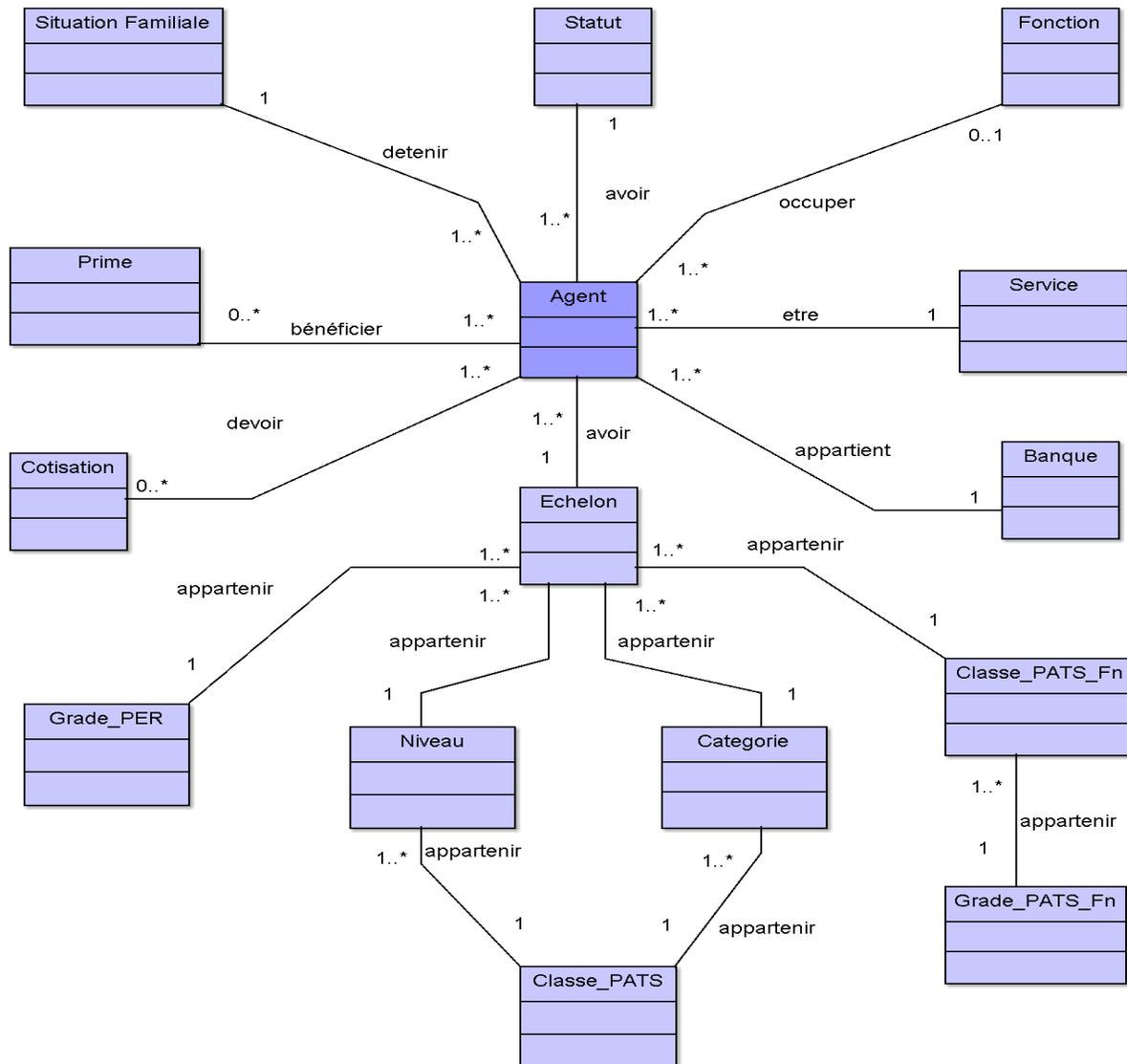
***Figure 16 : diagramme de déploiement***

## ***II- Conception préliminaire***

La phase de conception préliminaire du projet est l'ultime phase de la conception de notre système d'information, elle consiste à construire les classes qui interviennent pour le codage de la solution. Pour cela, nous allons utiliser le diagramme de classe qui permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d'association.

## 1. Les classes participantes aux fonctionnalités d'ajout, de modification, de désactivation d'un agent

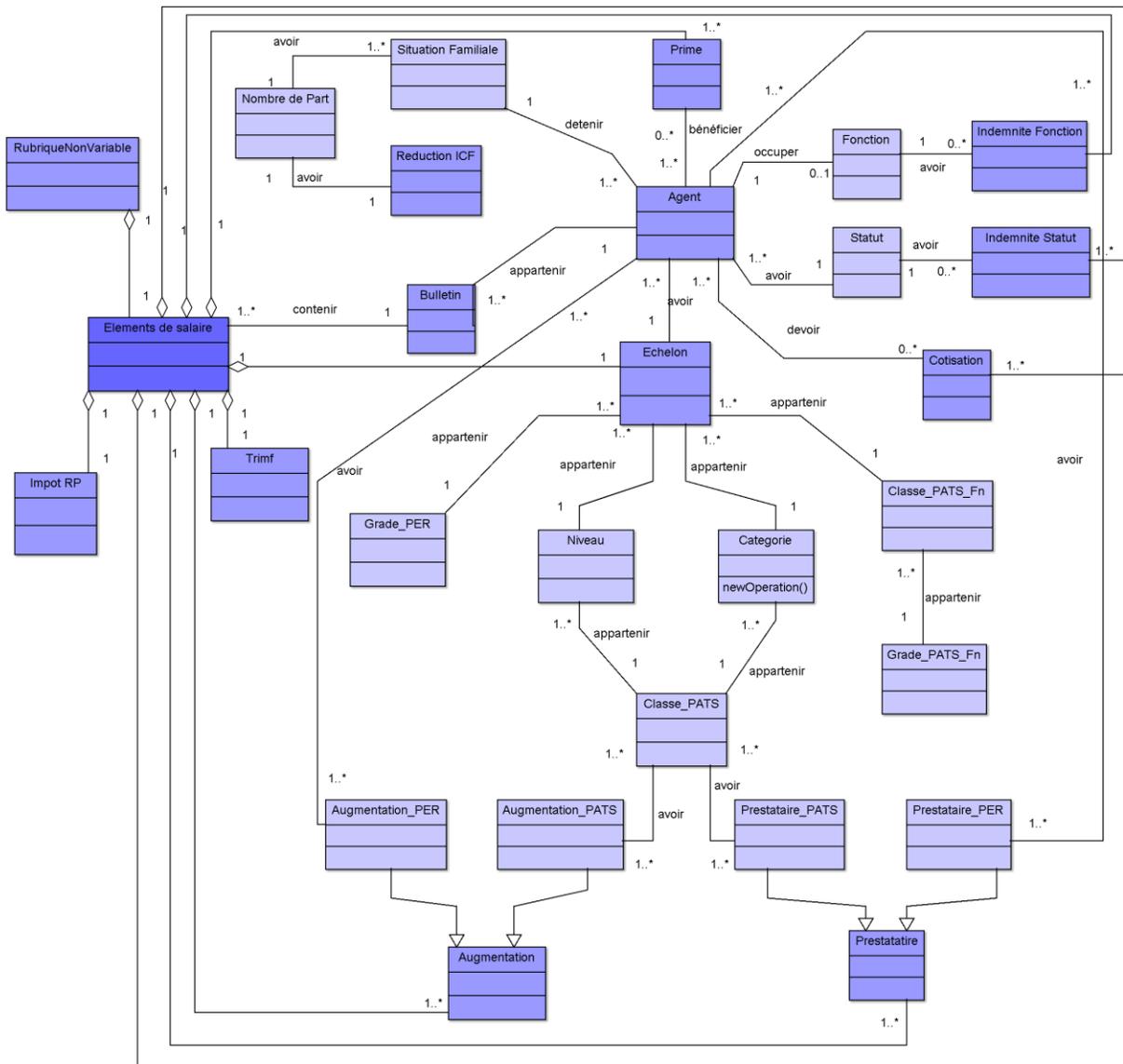
Dans la figure suivante, nous allons représenter les différentes classes nécessaires à l'ajout, la modification et la désactivation d'un agent dans notre application.



**Figure 17** : Diagramme de classe des classes participantes aux fonctionnalités ajout, modification et désactivation d'un agent.

## 2. Les classes participantes aux fonctionnalités d'édition et de la validation d'un bulletin de salaire provisoire

Dans la figure suivante, on va représenter les classes nécessaires pour l'obtention d'un bulletin de salaire.



**Figure 18 :** Diagramme de classe des classes participantes aux fonctionnalités d'édition et de la validation d'un bulletin de salaire provisoire.

**Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»**

Ce chapitre a permis de faire la conception du système « Gestion automatique des bulletins de salaire de l'UASZ ». Dans le prochain chapitre, nous entamons la phase d'implémentation de cette application qui utilise des technologies spécifiques pour les besoins.

## CHAPITRE V : IMPLEMENTATION ET PRESENTATION DE L'APPLICATION

Dans ce chapitre nous traitons en premier lieu les différentes parties d'implémentation de l'application en utilisant les résultats fournis dans la partie conception. Pour ce faire nous parlons de l'environnement de développement et les langages utilisés. Ensuite nous passerons par la présentation de l'organigramme d'interaction et les différentes interfaces de notre application

### *I- Implémentation de l'application*

#### 1. Les serveurs utilisés pour la mise en œuvre de l'application

##### *a. Apache Tomcat*

Apache Tomcat, souvent appelé serveur Tomcat, est un conteneur de servlets Java open-source développé par Apache Software Foundation (ASF). Tomcat implémente plusieurs spécifications Java EE, notamment Java Servlet, JavaServer Pages (JSP), Java EL et WebSocket, et fournit un environnement de serveur Web HTTP « pur Java » dans lequel le code Java peut s'exécuter. [7]

Tomcat est un serveur HTTP à part entière. De plus, il gère les servlets et les JSP (par un compilateur Jasper compilant les pages JSP pour en faire des servlets). Tomcat a été écrit en langage Java. Il peut donc s'exécuter via la machine virtuelle Java sur n'importe quel système d'exploitation la supportant. [8]

Catalina est le conteneur de servlets utilisé par Tomcat. Il est conforme aux spécifications servlet d'Oracle Corporation et les JavaServer Pages (JSP). Coyote est le connecteur HTTP de Tomcat, compatible avec le protocole HTTP 1.1 pour le serveur web ou conteneur d'application. Jasper est le moteur JSP d'Apache Tomcat. Tomcat 5.x utilise Jasper 2, qui est une implémentation de la spécification JavaServer Pages 2.3 de Sun Microsystems. Jasper parse les fichiers JSP afin de les

compiler en code Java en tant que servlets (gérés par Catalina). Pendant son exécution, Jasper est capable de détecter et recompiler automatiquement les fichiers JSP modifiés. [8]

### ***b. SGBD (MySQL)***

En informatique, un système de gestion de bases de données (SGBD) est un logiciel système destiné à stocker et à partager des informations dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations.

Un SGBD permet d'inscrire, de retrouver, de modifier, de trier, de transformer ou d'imprimer les informations de la base de données. Il permet d'effectuer des comptes-rendus des informations enregistrées et comporte des mécanismes pour assurer la cohérence des informations, éviter des pertes d'informations dues à des pannes, assurer la confidentialité et permettre son utilisation par d'autres logiciels. [9]

**MySQL** est un système de gestion bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de bases de données les plus utilisées au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server.

MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multithread et multi-utilisateur.

C'est un logiciel libre, open source développée sous double licence selon qu'il est distribué avec un produit libre ou avec un produit propriétaire. Dans ce dernier cas, la licence est payante, sinon, c'est la licence publique générale GNU (GPL) qui s'applique. [10]

## **2. Les technologies utilisées pour la mise en œuvre de l'application**

## a. Technologies coté Back end

### i. Spring

Spring est un Framework bien connu des développeurs Java pour les nombreuses fonctionnalités qu'il apporte sur les aspects web, sécurité, batch ou encore accès aux données dans le cadre du développement d'une application

Spring est un conteneur dit « léger », c'est-à-dire une infrastructure similaire à un serveur d'applications J2EE. Il prend donc en charge la création d'objet et la mise en relation d'objet par l'intermédiaire d'un fichier de configuration qui décrit les objets à fabriquer et les relations de dépendance entre ces objets. Le gros avantage par rapport aux serveurs d'application est qu'avec Spring, les classes n'ont pas besoin d'implémenter une quelconque interface pour être prises en charge par le Framework (au contraire des serveurs d'applications J2EE et des EJBs). C'est en ce sens que Spring est qualifié de conteneur « léger ».

Spring s'appuie principalement sur l'intégration de trois concepts clé :

L'inversion de contrôle est assurée de deux façons différentes : la recherche de dépendances et l'injection de dépendances

La recherche de dépendance : consiste pour un objet à interroger le conteneur, afin de trouver ses dépendances avec les autres objets. C'est un cas de fonctionnement similaire aux EJBs.

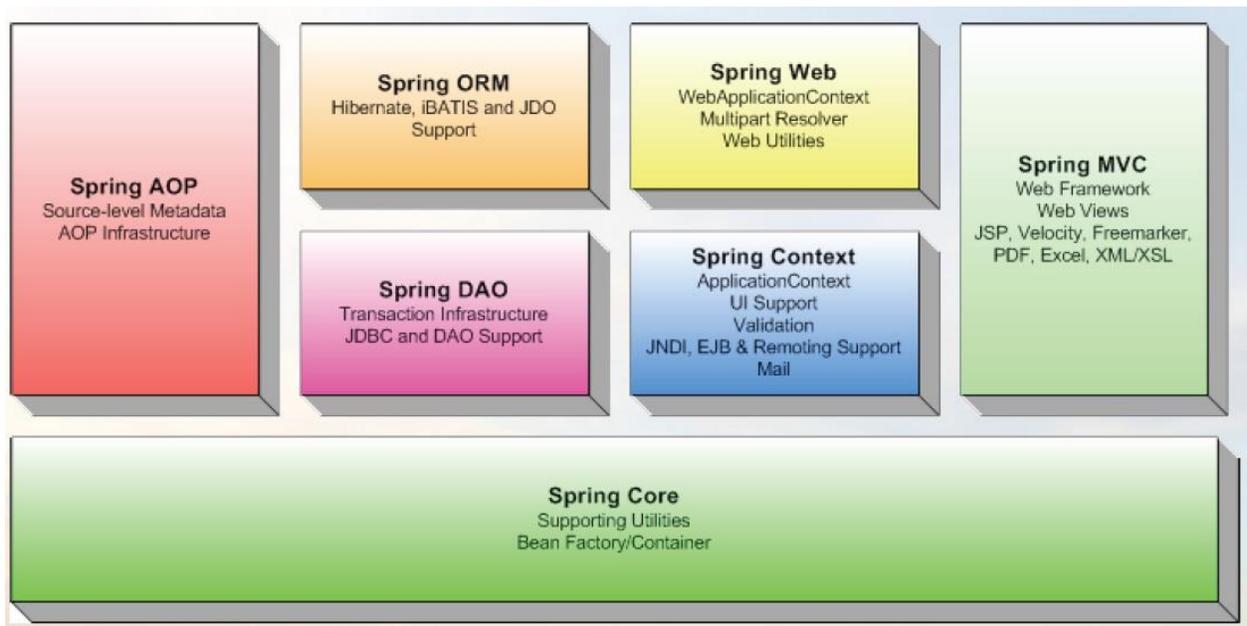
L'injection de dépendances : cette injection peut être effectuée de trois manières possibles :

- ❖ L'injection de dépendance via le constructeur.
- ❖ L'injection de dépendance via les modificateurs (setters).
- ❖ L'injection de dépendance via une interface.

La programmation orientée aspect (POA) qui est un paradigme de programmation qui permet de traiter séparément les préoccupations transversales (en anglais, *cross-cutting concerns*), qui relèvent souvent de la technique, des préoccupations métier, qui constituent le cœur d'une application. [11]

## Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

La couche d'abstraction permet d'intégrer d'autres Framework et bibliothèques avec une plus grande facilité. Cela se fait par l'apport ou non de couches d'abstraction spécifiques à des Framework particuliers. Il est ainsi possible d'intégrer un module d'envoi de mails plus facilement.



**Figure 19 : Architecture de Spring**

Le Framework est organisé en modules, reposant tous sur le module Spring Core :

**Spring Core** : implémente notamment le concept d'inversion de contrôle (injection de dépendance). Il est également responsable de la gestion et de la configuration du conteneur.

**Spring Context** : ce module étend Spring Core. Il fournit une sorte de base de données d'objets, permet de charger des ressources (telles que des fichiers de configuration) ou encore la propagation d'événements et la création de contexte comme par exemple le support de Spring dans un conteneur de Servlet.

**Spring AOP** : permet d'intégrer de la programmation orientée aspect.

**Spring DAO** : ce module permet d'abstraire les accès à la base de données, d'éliminer le code redondant et également d'abstraire les messages d'erreur spécifiques à chaque vendeur. Il fournit en outre une gestion des transactions.

**Spring ORM** : cette partie permet d'intégrer des Framework de mapping Object/Relationnel tel que Hibernate, JDO ou iBatis avec Spring. La quantité de code économisé par ce package peut être très impressionnante (ouverture, fermeture de session, gestion des erreurs)

**Spring Web** : ensemble d'utilitaires pour les applications web. Par exemple une servlet qui démarre le contexte (le conteneur) au démarrage d'une application web. Permet également d'utiliser des requêtes http de type multipart. C'est aussi ici que se fait l'intégration avec le Framework Struts.

**Spring Web MVC** : implémentation du modèle MVC. [12]

## *ii. Spring boot*

Spring Boot est un projet ou un micro Framework qui a notamment pour but de faciliter la configuration d'un projet Spring et de réduire le temps alloué au démarrage d'un projet. Spring Boot soutient des conteneurs embarqués (Embedded containers). Cela permet des applications web d'exécuter indépendamment sans déploiement sur Web Server. Pour arriver à remplir cet objectif, Spring Boot se base sur plusieurs éléments :

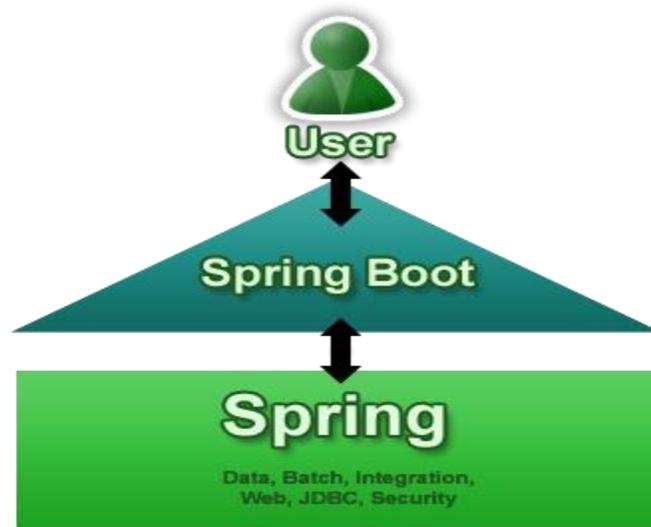
Un site web (<https://start.spring.io/>) qui vous permet de générer rapidement la structure de votre projet en y incluant toutes les dépendances Maven nécessaires à votre application. Cette génération est aussi disponible via le plugin Eclipse STS.

L'utilisation de « Starters » pour gérer les dépendances. Spring a regroupé les dépendances Maven de Spring dans des « méga dépendances » afin de faciliter la gestion de celles-ci. Par exemple, si vous voulez ajouter toutes les dépendances pour gérer la sécurité, il suffit d'ajouter le starter « Spring-boot-starter-Security ».

L'auto-configuration, qui applique une configuration par défaut au démarrage de votre application pour toutes dépendances présentes dans celle-ci. Cette configuration s'active à partir du moment où vous avez annoté votre application avec « @EnableAutoConfiguration » ou « @SpringBootApplication ». Bien entendu cette configuration peut-être surchargée via des propriétés Spring prédéfinie ou via une configuration Java. L'auto-configuration simplifie la configuration sans pour autant vous restreindre dans les fonctionnalités de Spring. Par exemple, si

vous utilisez le starter « Spring-boot-starter-Security », Spring Boot vous configurera la sécurité dans votre application avec notamment un utilisateur par défaut et un mot de passe généré aléatoirement au démarrage de votre application.

En plus de ces premiers éléments qui facilitent la configuration d'un projet, Spring Boot offre d'autres avantages notamment en termes de déploiement applicatif. Habituellement, le déploiement d'une application Spring nécessite la génération d'un fichier .war qui doit être déployé sur un serveur comme un Apache Tomcat. Spring Boot simplifie ce mécanisme en offrant la possibilité d'intégrer directement un serveur Tomcat dans votre exécutable. Au lancement de celui-ci, un Tomcat embarqué sera démarré afin de faire tourner votre application. [13]



***Figure 20 : Représentation de Spring boot par rapport Spring***

### ***iii. Thymeleaf***

Thymeleaf est un moteur de Template Java moderne côté serveur pour les environnements web et autonomes. L'objectif principal de Thymeleaf est d'apporter des modèles naturels élégants à votre flux de travail de développement - HTML qui peut être correctement affiché dans les navigateurs

et aussi travailler comme des prototypes statiques, permettant une collaboration plus forte dans les équipes de développement.

Avec des modules pour Spring Framework, une multitude d'intégrations avec vos outils préférés, et la possibilité de brancher vos propres fonctionnalités, Thymeleaf est idéal pour le développement web JVM HTML5 moderne. [14]

### ***b. Technologies coté Front end***

#### ***i. Bootstrap***

Bootstrap est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur ... etc. ...) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub. [15]

#### ***ii. JQuery***

JQuery est une bibliothèque JavaScript rapide, compacte et riche en fonctionnalités. Il rend les choses comme la lecture et la manipulation de documents HTML, la gestion des événements, l'animation et Ajax beaucoup plus simple avec une API facile à utiliser qui fonctionne sur une multitude de navigateurs. [16]

### **3. Les autres outils utilisés pour la mise en œuvre**

#### ***a. IntelliJ***

IntelliJ IDEA est un environnement de développement intégré (IDE) Java pour le développement de logiciels informatiques. Il est développé par JetBrains (anciennement connu sous le nom d'IntelliJ), et est disponible sous la forme d'une édition communautaire sous licence Apache et dans

une édition commerciale exclusive. Les deux peuvent être utilisés pour le développement commercial. [17]

### ***b. ArgoUML***

**ArgoUML** est un logiciel libre de création de diagrammes UML. Programmé en Java, il est édité sous licence EPL 1.0. Il est multilingue, supporte la génération de code et l'ingénierie inverse. [18]

### ***c. TopCased***

TOPCASED est un acronyme pour Toolkit in Open Source for Critical Applications & Systems Development (boîte à outils open source pour le développement d'applications critiques et de systèmes). Le projet a été abandonné en juillet 2015.

TOPCASED est un logiciel d'ingénierie assistée par ordinateur. Il contient un IDE basé sur le Framework de la plateforme de développement Eclipse, à laquelle il ajoute des fonctionnalités essentiellement liées à la mise en œuvre de la première branche du cycle en V pour l'ingénierie du logiciel, du matériel ou de systèmes mixtes logiciel/matériel. [19]

## **4. L'arborescence des fichiers de notre application**

Les projets Spring boot sous IntelliJ reprennent tous la même arborescence.

**Dossier .idea** : contient quelques sous-dossiers et divers fichiers XML contenant principalement des informations IntelliJ IDEA internes et des paramètres généraux. Normalement, il n'est pas nécessaire d'éditer le contenu de ce dossier.

**Dossier src** : c'est ce dossier qui nous intéresse, il contient tous les fichiers source qui composent l'application (activités, classes auxiliaires, etc.). Vous pouvez créer n'importe quelle hiérarchie de sous-dossiers dans le dossier src pour mieux refléter la structure et la complexité de votre application. Le répertoire src contient plusieurs sous-répertoires, chacun avec une utilité précise :

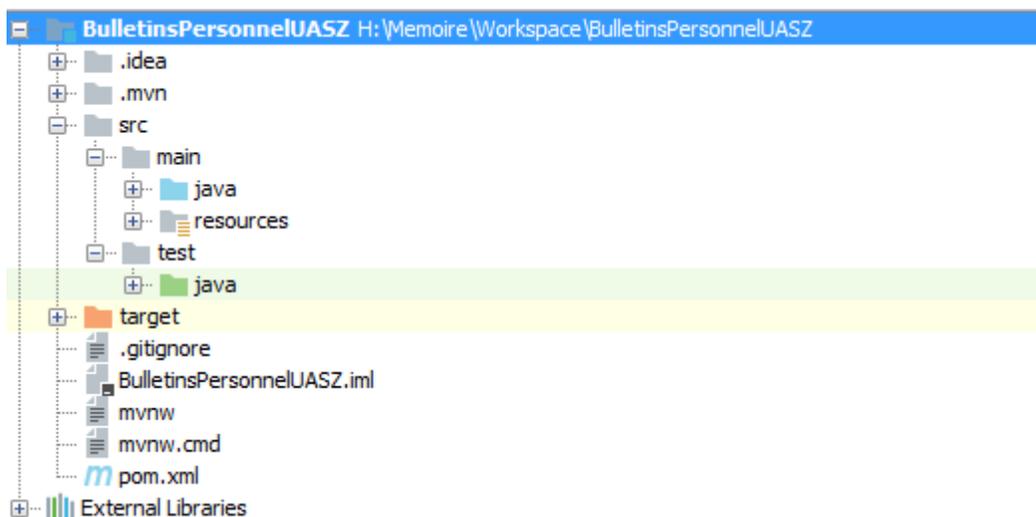
## Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

- ❖ **src/main/java** : notre code java va être ici. Dans ce dossier, on a la partie Back-end de notre application
- ❖ **src/main/ressources** : les autres ressources dont notre application a besoin. Dans ce package, se trouve la partie Front-end de notre application
- ❖ **src/test/java** : les tests unitaires

**Dossier target** : contient toutes les éléments générés du projet

**Dossier lib (ou external librairies)**: contient toutes les bibliothèques de classes (fichiers .jar) que vous voulez référencer à partir des fichiers source de l'application. Vous pouvez simplement glisser et déposer les fichiers .jar du disque dans ce dossier.

**Le fichier POM** (appelé POM pour Project Object Model) contient une description détaillée de votre projet, avec en particulier des informations concernant le versionnage et la gestion des configurations, les dépendances, les ressources de l'application, les tests, les membres de l'équipe, la structure et bien plus encore. Le POM prend la forme d'un fichier XML (*pom.xml*) qui est placé dans le répertoire de base du projet.

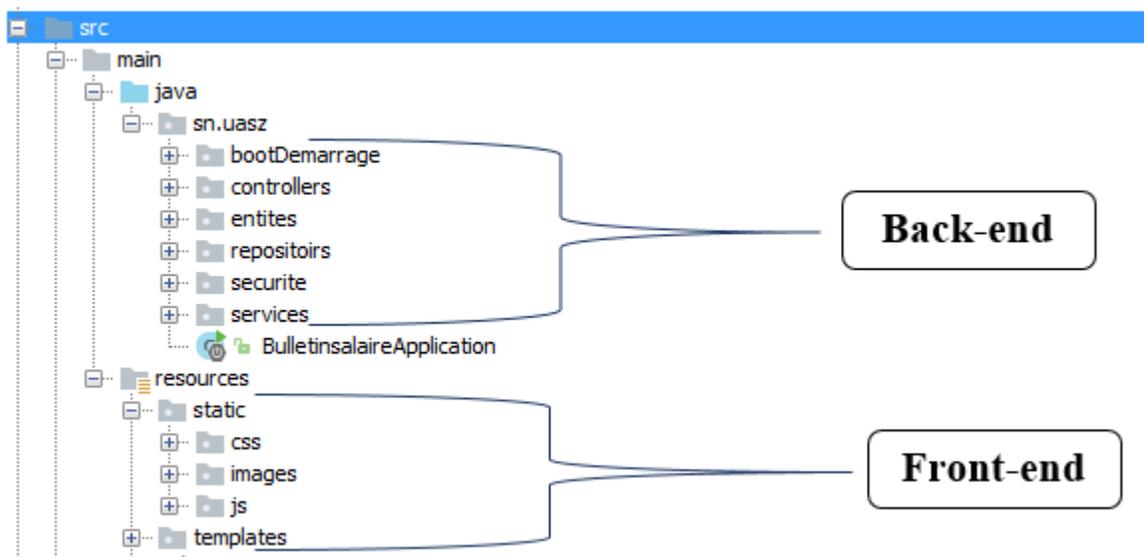


**Figure 21** : Arborescence des fichiers de l'application

## 5. L'arborescence des packages

## Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

Un projet respectant une architecture 3-Tiers doit pouvoir se distinguer dès le départ par son arborescence de package. Trop de projets consistent à mettre l'intégralité des sources dans un seul et même package. Ils rendent ainsi difficile, voire impossible, de bien différencier les sources utilisées pour la partie métier, de celles utilisées pour l'interface. Une bonne arborescence permet de classer les sources de son projet et de faciliter la reprise de code par d'autres développeurs. [20] L'arborescence présentée ci-dessous est celle que nous avons utilisée pour réaliser notre application :



**Figure 22** : Arborescence des packages du projet

### **a. Les packages du Back-end**

#### **i. Entités**

Il s'agit du package principal. L'ensemble des entités de notre application appelé beans se trouve dans le package. Les entités doivent respecter les conventions de nommage des attributs et des accesseurs. Ainsi que quelques annotations :

## Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

**@Entity** : est la première annotation indispensable. Elle se place avant la ligne qui déclare la classe et indique que la classe en question doit être gérée par la couche de persistance JPA. En l'absence de cette annotation, toutes les autres annotations JPA seraient ignorées.

**@Table** : désigne la table de la base de données dont la classe est une représentation. Son principal argument est **name** qui désigne le nom de la table. En l'absence de cet argument, la table portera le nom de la classe.

**@Id** : désigne le champ dans la classe qui est image de la clé primaire de la table. Cette annotation est obligatoire.

**@Column** : fait le lien entre un champ de la classe et la colonne de la table dont le champ est l'image. L'attribut **unique=true** indique que le nom d'une personne doit être unique. Cela va se traduire dans la base de données par l'ajout d'une contrainte d'unicité sur la colonne

**@NotNull** : indique que la colonne associée au champ ne peut avoir la valeur NULL et que donc le champ doit avoir nécessairement une valeur.

Le code ci-dessous est exemple avec la classe « agent ».

```
Agent
1 package sn.uasz.entites;
2
3 import ...
13
14 /**
15  * Created by mirk on 2017-04-24.
16  */
17 @Entity
18 @Table(name = "Agent")
19 public class Agent implements Serializable {
20
21     @Id
22     @Column(unique = true)
23     private String matricule;
24
25     @NotNull
26     private String nom;
27
28     @NotNull
29     private String type;
30
31     @NotNull
32     private String prenom;
33
34     private String nomMarital;
35
36     @NotNull
37     private String civilite;
38
39     private boolean active;
40
```

**Figure 23** : Exemple de la classe « Agent »

## ii. Repositoires

Dans ce package, on met l'ensemble des interfaces définissant les méthodes d'accès disponibles pour l'accès au donnée. Ce package est indépendant de l'implémentation choisie.

**@Repository** est une annotation Spring qui indique que la classe décorée est une référentielle de la base de données. Un référentiel est un mécanisme d'encapsulation du stockage, de la récupération et du comportement de recherche qui émule une collection d'objets.

Le code ci-dessous est un exemple avec l'interface « fonctionRepository ».

```
FonctionRepository
1 package sn.uasz.repositoires;
2
3 import ...
4
5 /**
6  * Created by mirk on 2017-04-21.
7  */
8
9 @Repository
10 public interface FonctionRepository extends JpaRepository<Fonction, Long> {
11
12     /*Rechercher une fonction par nom*/
13     Fonction findByNom(String nom);
14
15     /*Rechercher une liste de fonction par type*/
16     List<Fonction> findByType(String type);
17
18     /*Rechercher une liste de fonction de fonction ayant un typeAou typeB */
19     @Query("SELECT f FROM Fonction f WHERE f.type = ?1 OR f.type = ?2")
20     List<Fonction> findByTypeAndType(String type1, String type2);
21
22 }
```

**Figure 24** : Exemple de l'interface « *FonctionRepository* »

### iii. Service

Dans ce package, on met l'ensemble des interfaces définissant les services disponibles de la partie métier (les managers). L'ensemble des managers implémentant les services sont dans ce package.

Le code ci-dessous est un exemple avec l'interface « *bulletinService* ».

## Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

```
BulletinService
1 package sn.uasz.services;
2
3 import ...
4
5 /**
6  * Created by mirk on 2017-06-16.
7  */
8
9 public interface BulletinService {
10
11     /*Ajouter un bulletin*/
12     public void addBulletin(Bulletin bulletin);
13
14     /*Afficher un bulletin selon son numero*/
15     public Bulletin getBulletin(String numero);
16
17     /*Afficher tout les bulletins selon un mois et une année données*/
18     public List<Bulletin> getBulletins(Long mois,Long annee);
19
20     /*Modifier un bulletin*/
21     public void updateBulletin(Bulletin bulletin);
22
23     /*Supprimer un bulletin*/
24     public void deleteBulletin(String numero);
25
26     /*Afficher tout les bulletins*/
27     public List<Bulletin> getBulletins();
28
29     /*Afficher le bulletin d'un agent*/
30     public String bulletinAgent(String numero,Bulletin bulletin) throws FileNotFoundException, DocumentException;
31
32
33 }
```

**Figure 25 : Exemple avec l'interface « BulletinService »**

Les annotations suivantes sont indispensables à l'implémentation de l'interface « bulletinService ».

**@Service** annote les classes au niveau de la couche de service.

**@Transactional** est une annotation qui vous permet de travailler avec les transactions des bases de données de façon déclarative.

Le code ci-dessous permet d'implémenter l'interface « bulletinService » d'où le nom de la classe est « bulletinServiceImpl ».

```
BulletinServiceImpl
1 package sn.uasz.services.impl;
2
3 import ...
4
22
23 /**
24  * Created by mirk on 2017-06-16.
25  */
26 @Service
27 @Transactional
28 public class BulletinServiceImpl implements BulletinService {
29
30     private BulletinRepository bulletinRepository;
31     private AgentEchelonService agentEchelonService;
32     private ElementSalaireService elementSalaireService;
33
34     @Override
35     public void addBulletin(Bulletin bulletin) { bulletinRepository.save(bulletin); }
36
37
38     @Override
39     public Bulletin getBulletin(String numero) { return bulletinRepository.findOne(numero); }
40
41
42     @Override
43     public List<Bulletin> getBulletins(Long mois, Long annee) {
44         return bulletinRepository.findByMoisAndAnnee(mois, annee);
45     }
46
47
48     @Override
49     public void deleteBulletin(String numero) { bulletinRepository.delete(numero); }
50
51
52     @Override
53     public List<Bulletin> getBulletins() { return bulletinRepository.findAll(); }
54
55
56
```

**Figure 26** : Exemple avec la classe « *BulletinServiceImpl* »

#### iv. Controller

Dans ce package, on met l'ensemble des sources en rapport avec la partie web et l'interface de l'application.

**@Controller** marque une classe en tant que contrôleur Spring Web MVC, c'est-à-dire annoter les classes au niveau des couches de présentation.

**@Autowired** ne gère que le câblage. Cette annotation permet de spécifier une variable d'instance à renseigner par Spring ou de récupérer automatique un Bean de Spring.

**@RequestMapping** a pour rôle principal de préciser quelle méthode doit être appelée pour une URI donnée.

Le code ci-dessous est un exemple de controller avec la classe « *statutController* ».

```
1 package sn.uas2.controllers;
2
3 import ...
4
5 /**
6  * Created by mirk on 2017-05-18.
7  */
8 @Controller
9 public class StatutController {
10
11     private StatutService statutService;
12
13     @Autowired
14     public void setStatutService(StatutService statutService) { this.statutService = statutService; }
15
16     /*Affichage de la liste des statuts*/
17     @RequestMapping("/Statut")
18     public String statut(Model model){
19         List<Statut> statuts=statutService.getStatuts();
20         model.addAttribute("statuts",statuts);
21         return "Drh_Statut";
22     }
23
24     /* Debut Gestion des Statuts */
25
26     /*Ajouter un Statut*/
27     @RequestMapping(value = "/AjouterStatut")
28     public String ajouterStatut(Statut statutAjt){
29         statutService.addStatut(statutAjt);
30         return "redirect:/Statut";
31     }
32
33 }
34
35
36
37
38
39
40
41
42
43
44
```

Figure 27 : Exemple avec la classe « StatutController »

## v. Security

Dans ce package, on gère toute la sécurité (protection des URL de l'application, validation du nom d'utilisateur et des mots de passe soumis, redirection vers le formulaire de connexion, etc.) au sein de l'application.

@EnableWebSecurity et WebSecurityConfigurerAdapter fonctionnent ensemble pour fournir une sécurité Web. En étendant WebSecurityConfigurerAdapter et seulement quelques lignes de code, nous sommes en mesure de fournir les services de sécurité suivants :

- ❖ *L'authentification*, consiste à assurer que seules les personnes autorisées aient accès aux ressources.
- ❖ *Le contrôle d'accès*, permet l'accès à des ressources uniquement aux personnes autorisées.

## Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

- ❖ *L'intégrité*, garantit que les données qui ont été envoyées, reçues ou stockées sont complètes et n'ont pas été modifiées par une entité tierce (accidentellement ou intentionnellement).
- ❖ *La confidentialité*, permet la protection des communications ou des données stockées contre l'interception et la lecture par des personnes non autorisées.

Le code ci-dessous en est un exemple de sécurité avec la classe « SecuriteConfig »

```
SecuriteConfig
package sn.uasz.securite;

import ...

/**
 * Created by mirk on 2017-03-15.
 */
@Configuration
@EnableWebSecurity
public class SecuriteConfig extends WebSecurityConfigurerAdapter{

    @Autowired
    private DataSource dataSource;

    @Override
    protected void configure(AuthenticationManagerBuilder auth) throws Exception {

        auth.jdbcAuthentication()
            .dataSource(dataSource)
            .usersByUsernameQuery("select username as principal, password as credentials, active from utilisateur where username=?")
            .authoritiesByUsernameQuery("select username as username, role as role from utilisateur_roles where username=?")
            .rolePrefix("ROLE_")
            .passwordEncoder(new BCryptPasswordEncoder());
    }

    @Bean
    public PasswordEncoder passwordEncoder(){
        PasswordEncoder encoder=new BCryptPasswordEncoder();
        return encoder;
    }
}
```

**Figure 28** : Exemple avec la classe « SecuriteConfig »

### **b. Les packages du Front-end**

#### **i. Static**

Dans ce package, on met les sous-dossiers qui auront des fichiers qui vont interagir avec les pages web de notre application.

- ❖ **CSS** : pour les fichiers utilisant le Langage CSS (Bootstrap, ..).
- ❖ **JS** : pour les fichiers utilisant le Langage de programmation de script (jQuery,..)
- ❖ **Images** : pour toutes les images qu'on aura besoin dans l'application

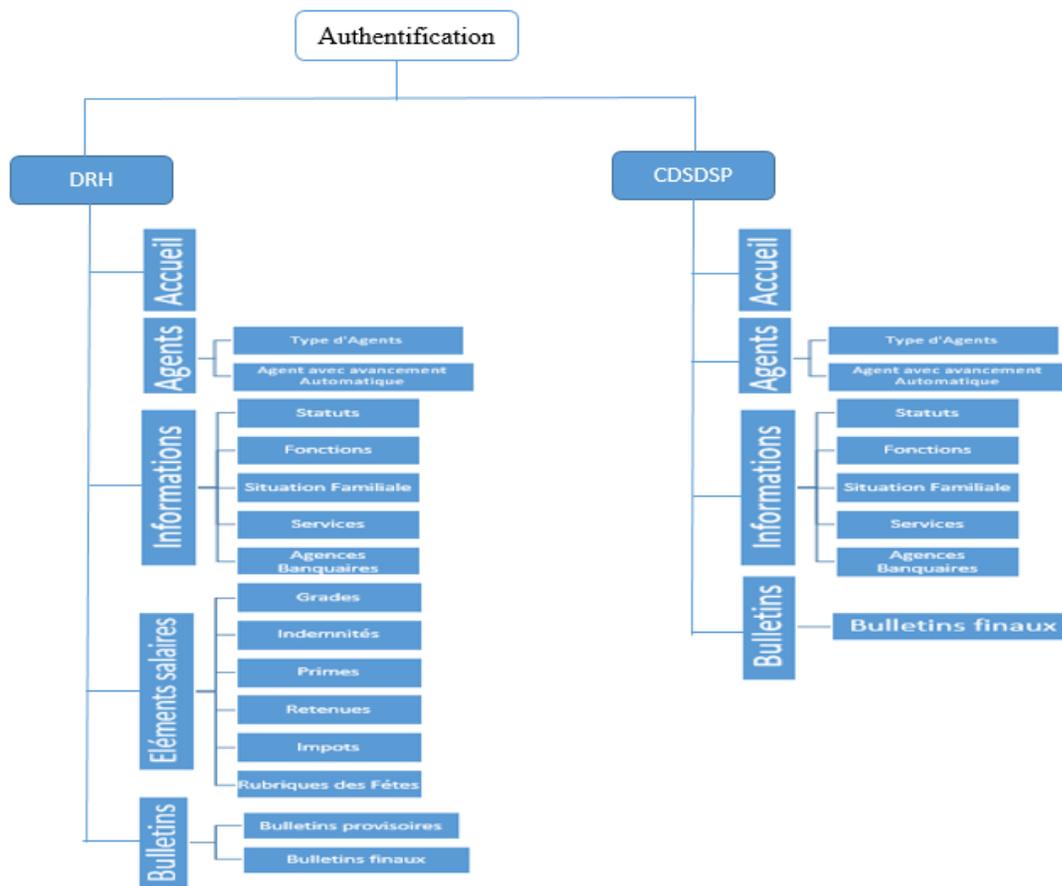
#### **ii. Template**

Dans ce package, on met toutes les pages Web de notre application, c'est-à-dire toutes les fichiers Html que nous aurons à utiliser durant tout le projet.

## II- Présentation de l'Application

Au terme de notre travail, nous avons obtenus une application qui est composée de plusieurs fonctionnalités accessibles à travers les menus et sous-menus que nous présentons dans **la figure 29**.

Cependant, nous allons représenter un seul scénario qui est celui de l'ajout d'un nouveau agent jusqu'à la production de son bulletin de salaire, en passant par l'attribution d'un grade à l'agent.

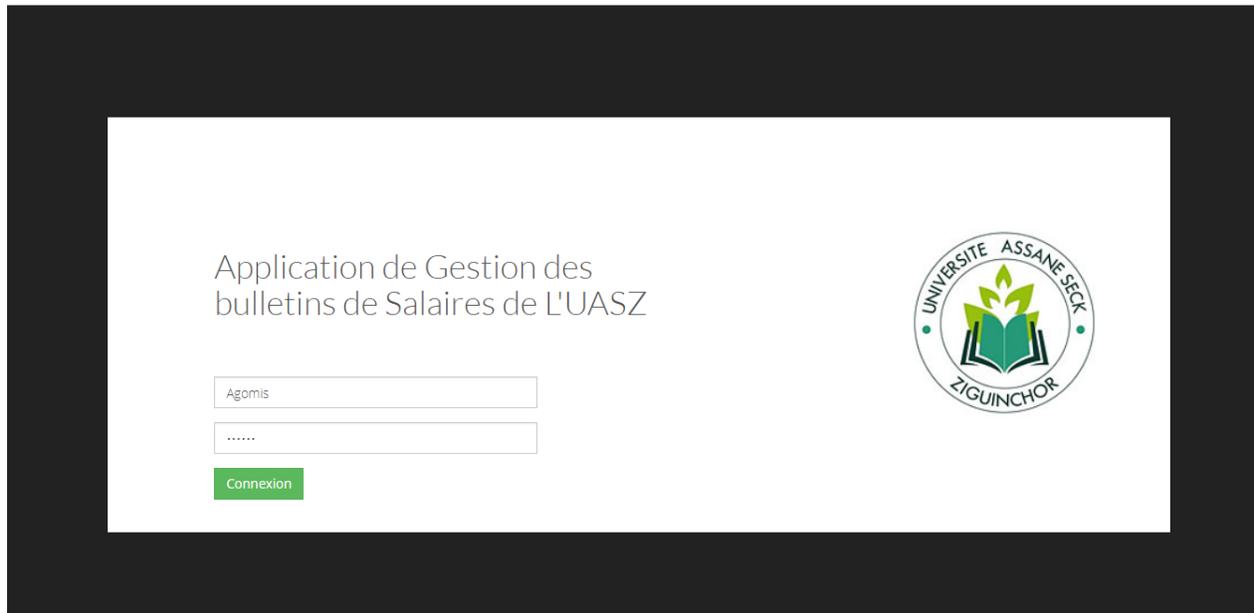


**Figure 29** : La liste des différents menus et sous menus de l'application

### 1. Page Authentification

## Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

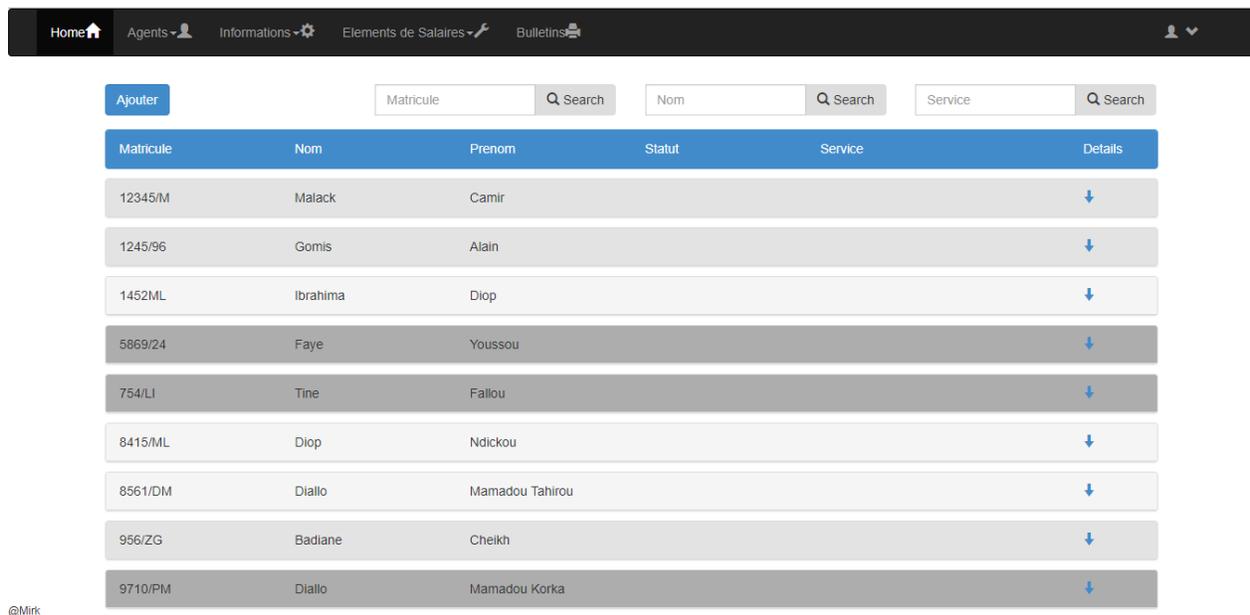
L'authentification apparaît lorsqu'un utilisateur saisit l'adresse de l'application dans un navigateur web. Elle permet à un utilisateur de se connecter à l'application après avoir saisi un bon login et bon mot de passe dans les champs correspondants (voir **Figure 30**).



***Figure 30 : Page d'authentification***

## 2. **Page d'accueil**

Cette interface de la **Figure 31** est la page d'accueil du DRH. À partir de cette page, le DRH peut naviguer dans toute l'application en passant par la liste des agents qui dispose de filtres sur le matricule, nom et le service des agents, comme le montre la **Figure 31** suivante. Il peut aussi à partir de cette liste voir les détails de chaque agent et à partir de là rajouter d'autres informations sur les agents. De plus, le DRH peut ajouter un nouveau agent avec le bouton « *Ajouter* » qui se trouve dans cette même figure, mais en haut et à gauche de la liste des agents.



The screenshot shows the home page of a web application for HR management. At the top, there is a navigation bar with links for Home, Agents, Informations, Elements de Salaires, and Bulletins. Below the navigation bar, there are search filters for Matricule, Nom, and Service, each with a search button. The main content area displays a table with the following columns: Matricule, Nom, Prenom, Statut, Service, and Details. The table contains ten rows of agent data, each with a downward arrow in the Details column.

Matricule	Nom	Prenom	Statut	Service	Details
12345/M	Malack	Camir			↓
1245/96	Gomis	Alain			↓
1452ML	Ibrahima	Diop			↓
5869/24	Faye	Youssou			↓
754/LI	Tine	Fallou			↓
8415/ML	Diop	Ndickou			↓
8561/DM	Diallo	Mamadou Tahirou			↓
956/ZG	Badiane	Cheikh			↓
9710/PM	Diallo	Mamadou Korke			↓

**Figure 31 : Page d'accueil du DRH**

### 3. Ajout d'un agent

L'ajout d'un agent dans notre application nécessite un processus que nous détaillerons dans les trois points suivants :

- ❖ Cette première fenêtre permet au DRH de saisir les informations concernant l'identité de l'agent (**Erreur ! Source du renvoi introuvable.**) composé du matricule, prénom, nom, ... Après la saisie de ces premières informations sur l'identité, le DRH doit appuyer sur le bouton « *Suivant* » pour aller à la deuxième fenêtre (voir **Figure 33**).
- ❖ Cette deuxième fenêtre permet au DRH de renseigner les informations concernant l'adresse de l'agent, les numéros de téléphones fixe ou mobile ainsi que l'e-mail de l'agent. Après la saisie de ces informations de la deuxième fenêtre, le DRH doit appuyer sur le bouton « *Suivant* » pour aller à la troisième fenêtre (voir **Figure 34**).
- ❖ Cette troisième (3ième) fenêtre permet à l'utilisateur de renseigner les données de l'agent au sein de l'Université à savoir sa catégorie (ou type), les dates de prise de service, .... Après la saisie de ces informations de la troisième fenêtre, le DRH doit appuyer sur le bouton « *Terminer* » pour aller à la page gestion des informations de l'agent (voir **Figure 35**).

## Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

The screenshot shows a web application interface with a table of agents in the background and a modal form titled "Ajouter une Identité Agent" in the foreground. The table lists agents with columns for Matricule and Nom. The modal form contains the following fields:

Code Matricule	1120/ML
Prenom	Lucien Andre Marie
Nom	Malack
Civilite	Monsieur
Nom Marital	
Date de Naissance	02/08/1976
Lieu de Naissance	Dakar
Nationalite	Sénégalaise

Buttons: Annuler, Suivant

**Figure 32** : Page pour la saisie des informations sur l'identité de l'agent

The screenshot shows a web application interface with a table of agents in the background and a modal form titled "Ajouter une Adresse Agent" in the foreground. The table lists agents with columns for Matricule and Nom. The modal form contains the following fields:

Adresse	Tilène
Portable	775660400
Email	l.malack@univ-zig.sn
Tel Bureau	
Tel Fixe	339901041

Buttons: Retour, Suivant

**Figure 33** : Page pour la saisie des informations sur l'adresse de l'agent

## Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

Matricule	Nom		
12345/M	Malack		
1245/96	Gomis		
1452/ML	Ibrahima		
5869/24	Faye		
754/LI	Tine		
8415/ML	Diop		
8561/DM	Diallo	Mamadou Tahirou	
956/ZG	Badiane	Cheikh	
9710/PM	Diallo	Mamadou Korka	

**Figure 34** : Page pour la saisie des informations sur l'identité de l'agent

Ainsi, après l'insertion de ces premières informations de l'agent dans la base de données, l'interface de la **Figure 35** apparaît automatiquement pour permettre au DRH (aidé par le CDSDSP) de rentrer les autres informations complémentaires (Voir la section 4. Suivante), qui seront décisives pour le calcul de son salaire et l'établissement de son bulletin de paie.

Malack Lucien Andre Marie Matricule : 1120/ML

- Identite de l'Agent
- Adresse de l'Agent
- Informations Professionnelles de l'Agent
- Situation Familiale
- Grade
- Statut
- Fonction
- Service
- Banque
- Prime Particuliere
- Retenue et Cotisation

**Figure 35** : Page de gestion des informations de l'agent

## Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

Dans cette **Figure 35**, nous pouvons voir toutes les rubriques d'informations pour la gestion du salaire d'un agent de l'UASZ. Dans la suite, nous avons choisi de vous montrer la rubrique grade, en ajoutant le grade d'un agent.

### 4. Attribution de Grade à un Agent

La **Figure 36** représente l'attribution d'un grade à un agent. Pour cela, le DRH choisit d'abord la rubrique « Grade » au niveau de la liste proposée et ensuite appuie sur le bouton « Ajouter », pour ajouter le grade de l'agent comme le montre la **Figure 36**.

Malack Lucien Andre Marie Matricule : 1120/ML

Classe	Niveau	Categorie	Echelon	Date Debut	Date Fin	Actif
<input type="button" value="Ajouter"/>						

**Figure 36** : Page pour l'attribution d'un grade à un agent

La fenêtre de la **Figure 37** montre les informations que le DRH pour renseigner le grade d'un agent. Ce grade sera automatiquement activé pour l'agent si on appuie sur le bouton « Ajouter » sans désactiver la case à cocher « Actif » qui est par défaut cochée.

## Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

Choisir Echelon PATS

Classe  
Classe 1

Niveau  
Niveau 1

Categorie  
Categorie 1

Echelon  
Echelon 1

Date Debut  
03/20/2012

Actif

Ajouter Annuler

**Figure 37** : Page pour le choix du grade de l'agent

Une fois que le DRH aura validé les informations sur le grade de l'agent, il aura la possibilité de les modifier ou de les supprimer, comme le montre la **Figure 38**.

Elements de Salaires

Malack Lucien Andre Marie Matricule : 1120/ML

Classe	Niveau	Categorie	Echelon	Date Debut	Date Fin	Actif
Classe 1	Niveau 1	Categorie 1	Echelon 1	20-03-2012		Oui

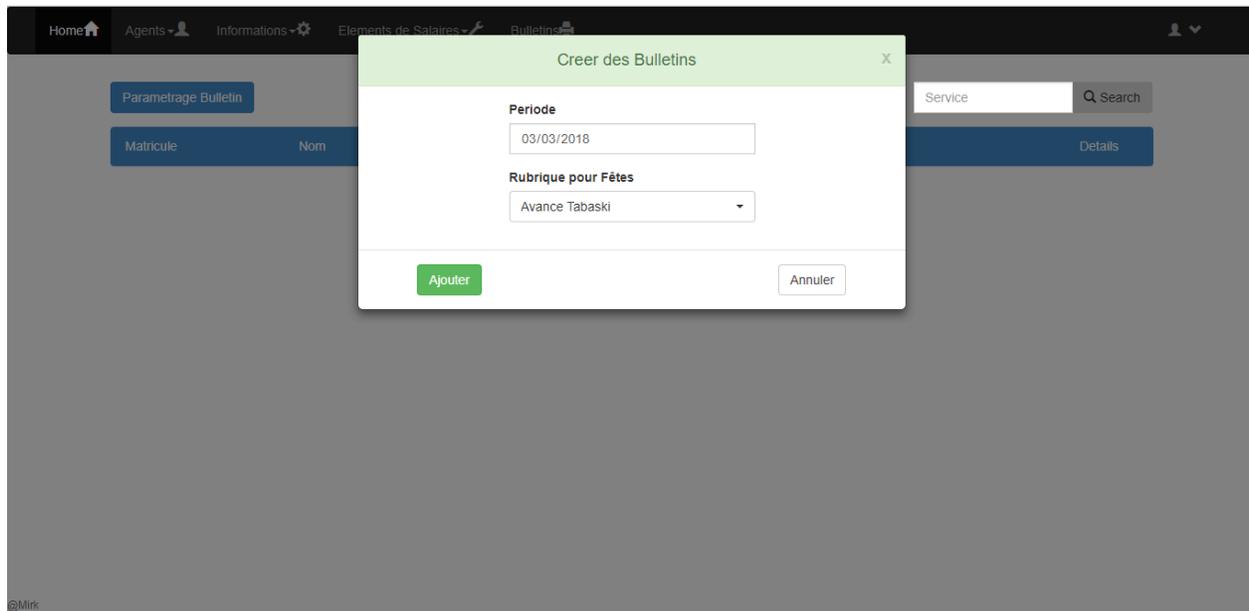
Ajouter

**Figure 38** : Page pour lister les grades de l'agent

## 5. Edition des Bulletins de salaire et Envoie par email

## Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

Cette fenêtre de la **Figure 39** permet au DRH de créer de nouveaux bulletins de paie pour tous les agents en renseignant la période des bulletins de salaire et les rubriques pour fêtes s'ils existent.



**Figure 39** : Page pour éditer les bulletins de salaire

Après la génération des bulletins de salaire d'un mois, on peut retourner sur la liste des agents et voir pour un agent donné son bulletin de salaire, comme le montre la **Figure 40** et ainsi le télécharger ou bien l'envoyer par e-mail à l'agent. Il existe aussi la possibilité d'envoyer par e-mail les bulletins de salaire à tous les agents de l'UASZ (ou à un ensemble réduit d'agents).

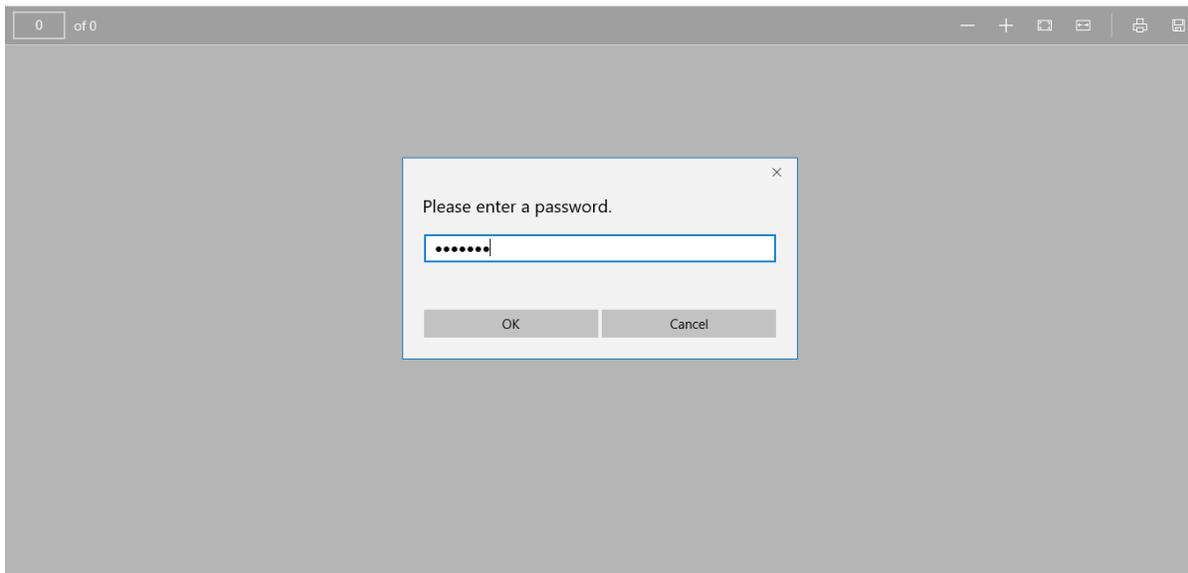
Libelle Rubrique	Base	Taux ou %	Gains	Retenues	Totaux
Solde Mensuelle			200000		
Augmentation avant 2000			23515		
Augmentation salaire 01/2002			9000		
Augmentation salaire 01/2000			5456		
Prime d'Assudiete	200000	5.0	10000		
Indemnité Complémentaire			25000		
Indemnité de Transport			50000		
Prime de risque	200000	10.0	20000		
Indemnité de Logement			50000		
Prime Administrative Spéciale			50000		
**Brut Imposable**					342971
Impôts sur le Revenu				57391	
Tremf				500	
Avance Tabeski			50000		
IPRES General	200000	5.5	11200		
Cotisation SATUC			10000		
Mutuelle Sante			10000		
Retenue Caisse de Solidarité			7000		
<b>Total Gains</b>			<b>492971</b>		
<b>Total Retenues</b>				<b>96091</b>	
<b>Net à Payer</b>					<b>396680</b>

**Figure 40** : Page pour vérifier le bulletin de salaire d'un agent

## 6. Consultation des bulletins avec mot de passe

Après chaque édition du bulletin de salaire, ce dernier est crypté et ne s'ouvre que si on donne le mot de passe qu'on demande. Donc, après chaque téléchargement d'un bulletin dans son e-mail, il faut impérativement avoir le mot de passe pour le consulter, voir la **Figure 41** et la **Figure 42**.

# Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»



**Figure 41 : Interface pour la saisie du mot de passe pour consulter le bulletin**

 <p>UNIVERSITÉ ASSANE SEKOU ZIGUINCHOR RECTORAT B.P. 523 – TEL : 33 991 68 09</p>	<b>Matricule :</b> 1120/ML <b>Nom :</b> Malack <b>Prénom :</b> Lucien Andre Marie <b>Grade :</b> Classe 1 Niveau 1 Catégorie 1 Echelon 1	<b>Catégorie :</b> PATS <b>Statut :</b> Coordonnateur Sportif <b>Fonction :</b> Neant <b>Service :</b> Direction des Services aux Etudiants <b>Periode :</b> 3 / 2018																																																																																																																																									
	<table border="1"> <thead> <tr> <th>Libelle Rubrique</th> <th>Base</th> <th>Taux %</th> <th>Gains</th> <th>Retenus</th> <th>Totaux</th> </tr> </thead> <tbody> <tr> <td>Solde Mensuelle</td> <td></td> <td></td> <td>200000</td> <td></td> <td></td> </tr> <tr> <td>Augmentation avant 2000</td> <td></td> <td></td> <td>23515</td> <td></td> <td></td> </tr> <tr> <td>Augmentation salaire 01/2002</td> <td></td> <td></td> <td>9000</td> <td></td> <td></td> </tr> <tr> <td>Augmentation salaire 01/2000</td> <td></td> <td></td> <td>5456</td> <td></td> <td></td> </tr> <tr> <td>Prime d'Assuidiete</td> <td>200000</td> <td>5.0</td> <td>10000</td> <td></td> <td></td> </tr> <tr> <td>Indemnité Complémentaire</td> <td></td> <td></td> <td>25000</td> <td></td> <td></td> </tr> <tr> <td>Indemnité de Transport</td> <td></td> <td></td> <td>50000</td> <td></td> <td></td> </tr> <tr> <td>Prime de risque</td> <td>200000</td> <td>10.0</td> <td>20000</td> <td></td> <td></td> </tr> <tr> <td>Indemnité de Logement</td> <td></td> <td></td> <td>50000</td> <td></td> <td></td> </tr> <tr> <td>Prime Administrative Spéciale</td> <td></td> <td></td> <td>50000</td> <td></td> <td></td> </tr> <tr> <td>**Brut Imposable**</td> <td></td> <td></td> <td></td> <td></td> <td>342971</td> </tr> <tr> <td>Impots sur le Revenu</td> <td></td> <td></td> <td></td> <td>57391</td> <td></td> </tr> <tr> <td>Trimf</td> <td></td> <td></td> <td></td> <td>500</td> <td></td> </tr> <tr> <td>Avance Tabaski</td> <td></td> <td></td> <td>50000</td> <td></td> <td></td> </tr> <tr> <td>IPRES General</td> <td>200000</td> <td>5.6</td> <td></td> <td>11200</td> <td></td> </tr> <tr> <td>Cotisation SATUC</td> <td></td> <td></td> <td></td> <td>10000</td> <td></td> </tr> <tr> <td>Mutuelle Sante</td> <td></td> <td></td> <td></td> <td>10000</td> <td></td> </tr> <tr> <td>Retenue Caisse de Solidarité</td> <td></td> <td></td> <td></td> <td>7000</td> <td></td> </tr> <tr> <td colspan="6" style="text-align: center;">Numero Compte : 14555A5555LM CBEAO Agence Ziguinchor Boucotte</td> </tr> <tr> <td><b>Brut imposable cumule</b></td> <td><b>date d'Embauche</b></td> <td><b>Situation familiale</b></td> <td><b>Nombre de parts</b></td> <td><b>total Gains</b></td> <td><b>Total Retenus</b></td> <td><b>Net à Payer</b></td> </tr> <tr> <td></td> <td>2012-03-20</td> <td>M</td> <td>2.0</td> <td>492971</td> <td>96091</td> <td>396880</td> </tr> </tbody> </table>						Libelle Rubrique	Base	Taux %	Gains	Retenus	Totaux	Solde Mensuelle			200000			Augmentation avant 2000			23515			Augmentation salaire 01/2002			9000			Augmentation salaire 01/2000			5456			Prime d'Assuidiete	200000	5.0	10000			Indemnité Complémentaire			25000			Indemnité de Transport			50000			Prime de risque	200000	10.0	20000			Indemnité de Logement			50000			Prime Administrative Spéciale			50000			**Brut Imposable**					342971	Impots sur le Revenu				57391		Trimf				500		Avance Tabaski			50000			IPRES General	200000	5.6		11200		Cotisation SATUC				10000		Mutuelle Sante				10000		Retenue Caisse de Solidarité				7000		Numero Compte : 14555A5555LM CBEAO Agence Ziguinchor Boucotte						<b>Brut imposable cumule</b>	<b>date d'Embauche</b>	<b>Situation familiale</b>	<b>Nombre de parts</b>	<b>total Gains</b>	<b>Total Retenus</b>	<b>Net à Payer</b>		2012-03-20	M	2.0	492971	96091
Libelle Rubrique	Base	Taux %	Gains	Retenus	Totaux																																																																																																																																						
Solde Mensuelle			200000																																																																																																																																								
Augmentation avant 2000			23515																																																																																																																																								
Augmentation salaire 01/2002			9000																																																																																																																																								
Augmentation salaire 01/2000			5456																																																																																																																																								
Prime d'Assuidiete	200000	5.0	10000																																																																																																																																								
Indemnité Complémentaire			25000																																																																																																																																								
Indemnité de Transport			50000																																																																																																																																								
Prime de risque	200000	10.0	20000																																																																																																																																								
Indemnité de Logement			50000																																																																																																																																								
Prime Administrative Spéciale			50000																																																																																																																																								
**Brut Imposable**					342971																																																																																																																																						
Impots sur le Revenu				57391																																																																																																																																							
Trimf				500																																																																																																																																							
Avance Tabaski			50000																																																																																																																																								
IPRES General	200000	5.6		11200																																																																																																																																							
Cotisation SATUC				10000																																																																																																																																							
Mutuelle Sante				10000																																																																																																																																							
Retenue Caisse de Solidarité				7000																																																																																																																																							
Numero Compte : 14555A5555LM CBEAO Agence Ziguinchor Boucotte																																																																																																																																											
<b>Brut imposable cumule</b>	<b>date d'Embauche</b>	<b>Situation familiale</b>	<b>Nombre de parts</b>	<b>total Gains</b>	<b>Total Retenus</b>	<b>Net à Payer</b>																																																																																																																																					
	2012-03-20	M	2.0	492971	96091	396880																																																																																																																																					

**Figure 42 : Bulletin de salaire d'un agent de l'UASZ en PDF**

## **CONCLUSION GENERALE**

Rappelons que l'objectif de ce travail était de concevoir et d'implémenter un système d'informations pour l'automatisation de la gestion des salaires et la dématérialisation du bulletin de salaire du personnel de l'UASZ.

Pour cela, nous avons réalisé ce système (ou cette application), qui répond aux attentes de la DRH pour calculer les salaires et produire les bulletins de paie du personnel. Ce logiciel permet aussi aux agents de l'UASZ de recevoir leur bulletin de salaire chaque mois dans leur boîte e-mail (avec plus de sécurité) après avoir reçu une notification sms pour leur informer de la disponibilité de ce dernier.

À cet effet, pour présenter les objectifs et les résultats de notre travail, ce mémoire a été découpé en cinq chapitres principaux. Dans le chapitre 1, nous avons présenté le contexte justificatif du sujet, les solutions aux problèmes auxquels la DRH rencontre dans la gestion des salaires de son personnel. Dans le deuxième chapitre, nous avons fait la connaissance de la méthodologie Scrum. Et cela, nous a permis au troisième chapitre de spécifier et d'analyser les besoins fonctionnels du système. La conception de notre application est faite dans le quatrième chapitre. Enfin dans le cinquième chapitre, nous avons présenté l'implémentation de l'application et puis nous avons présenté les interfaces de l'application produite.

En perspective, notre application peut être améliorée en enrichissant certaines fonctionnalités et en ajoutant de nouvelles fonctionnalités telles que :

- ❖ la description (ou l'explication) des éléments de salaire dans le bulletin de paie de l'agent (au verso du bulletin de salaire), ce qui va permettre à chaque agent de pouvoir recalculer son bulletin de salaire pour la vérification de la conformité de son solde net ;
- ❖ l'automatisation de la procédure de calcul du salaire et d'obtention du bulletin de salaire.

## BIBLIOGRAPHIE ET WEBOGRAPHIE

- [1] Mme Amy Diouf BADJI « Automatisation de la gestion des congés au niveau de la Direction des Ressources Humaines de l'Université Assane SECK de Ziguinchor », Mémoire de Master 2 MIAGE année scolaire 2014/2015 à l'UASZ.
- [2] <https://www.supinfo.com/articles/single/2912-pourquoi-utiliser-methode-scrum-votre-projet>
- [3] <http://projet.eu.org/pedago/sin/ICN/2nde/12-agile.pdf>
- [4] [http://www.dphu.org/uploads/attachements/books/books\\_1800\\_0.pdf](http://www.dphu.org/uploads/attachements/books/books_1800_0.pdf)
- [5] <http://www.beep.ird.fr/collect/upb/index/assoc/ESI-2006-ZAN-MIG/ESI-2006-ZAN-MIG.pdf>
- [6] <https://fr.slideshare.net/mohamedyoussefi9/mohamed-youssefi-support-architectures-logicielles-distribues-bases-sue-les-micro-services-avec-spring-boot>
- [7] [https://fr.wikipedia.org/wiki/Apache\\_Tomcat](https://fr.wikipedia.org/wiki/Apache_Tomcat)
- [8] [https://en.wikipedia.org/wiki/Apache\\_Tomcat](https://en.wikipedia.org/wiki/Apache_Tomcat)
- [9] [https://fr.wikipedia.org/wiki/Syst%C3%A8me\\_de\\_gestion\\_de\\_base\\_de\\_donn%C3%A9es](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es)
- [10] <https://fr.wikipedia.org/wiki/MySQL>
- [11] <https://spring.io/projects/spring-data>
- [12] <http://zekey.developpez.com/articles/spring/>
- [13] <http://www.groupeafg.com/spring-boot-kesako/>
- [14] <https://www.thymeleaf.org/>
- [15] [https://fr.wikipedia.org/wiki/Bootstrap\\_\(framework\)](https://fr.wikipedia.org/wiki/Bootstrap_(framework))
- [16] <https://jquery.com/>
- [17] <https://fr.wikipedia.org/wiki/Intellij>
- [18] <https://fr.wikipedia.org/wiki/Topcased>

**Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»**

[19] <https://fr.wikipedia.org/wiki/ArgoUML>

[20] <https://o7planning.org/fr/11267/le-tutoriel-de-spring-boot-pour-les-debutants#a5292673>

**NB**: Tous ces liens, utilisés dans cette partie, ont été consultés en mars 2018.

## ANNEXE

```
--  
-- Structure de la table `agent`  
--  
CREATE TABLE IF NOT EXISTS `agent` (  
  `matricule` varchar(255) NOT NULL,  
  `active` bit(1) NOT NULL,  
  `adresse` varchar(255) NOT NULL,  
  `civilite` varchar(255) NOT NULL,  
  `date_naissance` date NOT NULL,  
  `date_prise_service` date NOT NULL,  
  `date_recrutement` date NOT NULL,  
  `date_recrutement_initial` date NOT NULL,  
  `date_retraire` date DEFAULT NULL,  
  `date_sortie` date DEFAULT NULL,  
  `date_titularisation` date NOT NULL,  
  `email` varchar(255) NOT NULL,  
  `lieu_naissane` varchar(255) NOT NULL,  
  `nationalite` varchar(255) NOT NULL,  
  `nom` varchar(255) NOT NULL,  
  `nom_marital` varchar(255) DEFAULT NULL,  
  `portable` varchar(255) NOT NULL,  
  `prenom` varchar(255) NOT NULL,  
  `tel_bureau` varchar(255) DEFAULT NULL,  
  `tel_fixe` varchar(255) DEFAULT NULL,  
  `type` varchar(255) NOT NULL,  
  PRIMARY KEY (`matricule`),  
  UNIQUE KEY `UK_pxogqxl64ae07j2loxltgvrlx` (`email`),  
  UNIQUE KEY `UK_d13aufuws6xlx8kx12sfvlumb` (`portable`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--  
-- Structure de la table `agent_categorie`  
--  
CREATE TABLE IF NOT EXISTS `agent_categorie` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `date_debut` date NOT NULL,  
  `date_fin` date DEFAULT NULL,  
  `matricule_agent` varchar(255) DEFAULT NULL,  
  `code_categorie` bigint(20) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `FKhq0s7w0bji74xt9ufaxyde3ga` (`matricule_agent`),  
  KEY `FKkgagw6uidyqpka6qtwmo7h9ry` (`code_categorie`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;  
  
--  
-- Structure de la table `echelon`  
--  
CREATE TABLE IF NOT EXISTS `echelon` (  
  `code` bigint(20) NOT NULL AUTO_INCREMENT,  
  `indice` varchar(255) DEFAULT NULL,  
  `nom` varchar(255) NOT NULL,  
  `solde` bigint(20) NOT NULL,  
  `type` varchar(255) NOT NULL,  
  `code_patscategorie` bigint(20) DEFAULT NULL,  
  `code_patsfnclasse` bigint(20) DEFAULT NULL,  
  `code_patsniveau` bigint(20) DEFAULT NULL,  
  `code_pergrade` bigint(20) DEFAULT NULL,  
  PRIMARY KEY (`code`),  
  KEY `FK4wrgh8929vxot4qj842k0hbt` (`code_patscategorie`),  
  KEY `FKptgs5rc60abl489186qja0wy` (`code_patsfnclasse`),  
  KEY `FKbxqylnjdt1245xhv0ao01eub2` (`code_patsniveau`),  
  KEY `FK57qawanqdvio5ve68kovsyklc` (`code_pergrade`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

```
--  
-- Structure de la table `agent_fonction`  
--  
CREATE TABLE IF NOT EXISTS `agent_fonction` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `date_debut` date NOT NULL,  
  `date_fin` date DEFAULT NULL,  
  `en_vigueur` bit(1) NOT NULL,  
  `matricule_agent` varchar(255) DEFAULT NULL,  
  `code_fonction` bigint(20) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `FKsd7jdhxsxu38g1vwe3o3j702t` (`matricule_agent`),  
  KEY `FKsdefk5741hc2fny7p6wcl6c41` (`code_fonction`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;  
  
--  
-- Structure de la table `agent_prime_particuliere`  
--  
CREATE TABLE IF NOT EXISTS `agent_prime_particuliere` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `date_debut` date NOT NULL,  
  `date_fin` date DEFAULT NULL,  
  `en_vigueur` bit(1) NOT NULL,  
  `matricule_agent` varchar(255) DEFAULT NULL,  
  `code_primeparticuliere` bigint(20) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `FK3loqo18jlv36rhmurfq3jmndm` (`matricule_agent`),  
  KEY `FK3d5q22hne4icnmbg423htxiq5` (`code_primeparticuliere`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

```
--  
-- Structure de la table `agent_situation_familiale`  
--  
  
CREATE TABLE IF NOT EXISTS `agent_situation_familiale` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `date_debut` date NOT NULL,  
  `date_fin` date DEFAULT NULL,  
  `en_vigueur` bit(1) NOT NULL,  
  `matricule_agent` varchar(255) DEFAULT NULL,  
  `code_situationfamiliale` bigint(20) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `FKjhlmqdwd6yh6mxiaqj3ds5k7j` (`matricule_agent`),  
  KEY `FKltkofujndrqfscachphw3ogba` (`code_situationfamiliale`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;  
  
--  
-- Structure de la table `bulletin`  
--  
  
CREATE TABLE IF NOT EXISTS `bulletin` (  
  `numero` varchar(255) NOT NULL,  
  `annee` bigint(20) DEFAULT NULL,  
  `mois` bigint(20) DEFAULT NULL,  
  `periode` date DEFAULT NULL,  
  `matricule_agent` varchar(255) DEFAULT NULL,  
  PRIMARY KEY (`numero`),  
  KEY `FK2mxkfnb2hh4n3o60jn3a29umq` (`matricule_agent`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--  
-- Structure de la table `element_salaire`  
--  
CREATE TABLE IF NOT EXISTS `element_salaire` (  
  `id` bigint(20) NOT NULL AUTO_INCREMENT,  
  `base` bigint(20) DEFAULT NULL,  
  `code` bigint(20) DEFAULT NULL,  
  `gains` bigint(20) DEFAULT NULL,  
  `libelle` varchar(255) NOT NULL,  
  `retenues` bigint(20) DEFAULT NULL,  
  `taux` double NOT NULL,  
  `totaux` bigint(20) DEFAULT NULL,  
  `numero_bulletin` varchar(255) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `FKgkiepjd75o0fo244q54oowf0p` (`numero_bulletin`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

## **Table des matières**

RESUME.....	i
DEDICACES.....	ii
REMERCIEMENTS .....	iii
SOMMAIRE .....	iv
LISTE DES FIGURES.....	v
LISTE DES TABLEAUX .....	vii
LISTE DES ABREVIATIONS .....	viii
INTRODUCTION GENERALE .....	12
CHAPITRE I : CONTEXTE JUSTIFICATIF DU SUJET .....	14
I- La gestion des salaires à l’Université ASSANE SECK de Ziguinchor .....	14
II- Problèmes rencontrés dans la gestion des salaires à l’UASZ .....	18
III- Problématique du sujet.....	19
1. Solution proposée .....	19
2. Objectifs spécifiques du sujet.....	19
CHAPITRE II : PROCESSUS DE DEVELOPPEMENT DE L’APPLICATION.....	22
I- Présentation de la méthodologie SCRUM.....	22
II- Les cycles de la méthodologie SCRUM.....	24
III- Organisation de l’équipe du projet selon le Framework SCRUM.....	26
1. L’équipe de pilotage du projet .....	26
2. Le déroulement des événements du projet.....	27
CHAPITRE III : SPECIFICATION ET ANALYSE DES BESOINS FONCTIONNELS .....	28
I- La spécification des besoins .....	28
1. Identification des acteurs du système .....	28
2. Identification des fonctionnalités.....	29
3. Diagrammes de cas d’utilisation .....	31
4. Description de cas d’utilisation .....	34
II- Analyse des besoins fonctionnels du système .....	41
1. Les activités de l’authentification.....	41
2. Les activités de l’ajout d’un agent.....	43

# Mémoire de master : «Automatisation de la Gestion des Salaires et Dématérialisation du Bulletin de Salaire du Personnel de l'UASZ»

3. Les activités de l'attribution d'un grade à un agent.....	45
4. Les activités de l'édition d'un bulletin de paie provisoire.....	47
CHAPITRE IV : CONCEPTION DU SYSTEME .....	50
I- Conception générale .....	50
1. Architecture du système .....	50
2. Diagramme de package .....	52
3. Diagramme de déploiement .....	55
II- Conception préliminaire.....	56
1. Les classes participantes aux fonctionnalités d'ajout, de modification, de désactivation d'un agent.....	57
2. Les classes participantes aux fonctionnalités d'édition et de la validation d'un bulletin de salaire provisoire.....	58
CHAPITRE V : IMPLEMENTATION ET PRESENTATION DE L'APPLICATION .....	60
I- Implémentation de l'application .....	60
1. Les serveurs utilisés pour la mise en œuvre de l'application.....	60
2. Les technologies utilisées pour la mise en œuvre de l'application.....	61
3. Les autres outils utilisés pour la mise en œuvre .....	66
4. L'arborescence des fichiers de notre application .....	67
5. L'arborescence des packages .....	68
II- Présentation de l'Application .....	77
1. Page Authentification.....	77
2. Page d'accueil.....	78
3. Ajout d'un agent.....	79
4. Attribution de Grade à un Agent.....	82
5. Edition des Bulletins de salaire et Envoie par email.....	83
6. Consultation des bulletins avec mot de passe .....	85
CONCLUSION GENERALE .....	87
BIBLIOGRAPHIE ET WEBOGRAPHIE .....	88
ANNEXE .....	90
Table des matières .....	95