

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira De Bejaïa
Faculté de Sciences Exactes
Département d'Informatique



Mémoire de fin de cycle Master professionnel

Option : Génie Logiciel

T H E M E

**Conception et Réalisation d'une application de
gestion administrative De la SONATRACH-Bejaia**

**Cas d'études: "Service Gestion du département
Administration et Social."**

Présenté par :

- M^{lle} ABBACI Narimane
- M^{lle} ALITOUCHE Lamia

Encadré par :

- M^r ALLEM Khaled Univ. A/Mira Bejaia
- M^{me} BARCHICHE.A SONATRACH

Soutenu devant les membres du jury :

Président	M ^{me} BACHIRI Lyna	Univ. A/Mira Bejaia.
Examineur	M ^{me} EL BOUHISSI Houda	Univ. A/Mira Bejaia.
Examineur	M ^r ACHROUFENE Achour	Univ. A/Mira Bejaia.

Dédicaces

Je dédie ce modeste travail à :

Ma mère, ma première source de motivation, d'énergie, de courage, le symbole de la bonté par excellence. Ses prières m'ont été d'un grand aide pour mener à bien mes études. Maman merci d'être toujours là pour moi et de ne pas avoir cessé de croire en moi et de m'encourager.

Mon père, mon pilier, mon guide. Papa tu peux être fier, tu trouveras ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer et créer le climat parfait et affectueux à la poursuite de mes études.

Aucun hommage et dédicace ne pourrait être à la hauteur de l'amour dont ils ne cessent de me combler.

Que dieu leur procure bonne santé et longue vie.

A mon unique frère Mami, mon guide, mon modèle et ma fierté, à mes deux sœurs d'amour Basma et Leticia.

A la personne la plus chère à mes yeux, mon fiancé Salim pour tout son soutien, sa présence et son amour.

A ma meilleure amie Agnes.

Ils vont trouver ici l'expression de mes sentiments de respect et de reconnaissance.

M^{lle} ABBACI Narimane

Je dédie ce modeste travail

A mes chers parents pour leur soutien, leur patience,

Leur encouragement durant mon parcours scolaire.

A mon adorable sœur Souad

A mes frères Mohand et Sliman

A toute ma famille, de près ou de loin

A ma très chère amie Massylia

A tous ceux qui m'aiment

A tous ceux que j'aime

Je Dédie ce travail

M^{le}

Lamia

Alitouche

Remerciement

Nous tenons tout d'abord à remercier le Bon Dieu le tout puissant de nous avoir donné le courage, la force et la volonté durant toutes ces années d'étude et de nous avoir permis de réaliser ce travail dans de bonnes conditions.

Ce travail est le résultat d'une réflexion qui a bénéficié du soutien et des conseils attentifs de Madame **BARCHICHE.A** au sein de l'entreprise de la SONATRACH et de notre encadreur et promoteur Monsieur **ALLEM.K** qui ont bien voulu accepter de diriger ce projet. Nous voudrions également leurs témoigner l'expression de notre profonde gratitude et nos remerciements les plus distingués pour leur orientation, aide, confiance et leur patience.

Nos remerciements les plus sincères et notre loyale considération s'étendent également à tous nos enseignants durant nos années d'étude et aux personnes qui, de près ou de loin, nous ont apportés leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette année universitaire. Enfin, nous adressons nos plus sincères remerciements en particulier à nos parents, nos frères et sœurs et nos amis, qui nous ont toujours encouragés au cours de la réalisation de ce mémoire.

Merci à tous et à toutes.



Table des matières

Table des matières.....	I
Liste des tableaux.....	V
Liste des figures	VI
Liste des abréviations.....	VIII
Introduction générale	1
Chapitre I : Présentation de l'organisme d'accueil et étude de l'existant.....	3
I.1. Introduction	4
I.2. Présentation de la SONATRACH	4
I.2.1. Présentation de la Région Transport Centre	4
I.2.2. Département administration et social	7
I.3. Problématique	10
I.4. Solution envisagée.....	10
I.5. Conclusion	11
Chapitre II : Langages et méthodologie de conception	12
II.1. Introduction	13
II.2. Processus de développement.....	13
II.2.1. Scrum	13
II.2.2. Processus unifié.....	15
II.2.3. Pourquoi choisir scrum et UP.....	16
II.3. Langage de modélisation.....	17
II.3.1. OCL.....	18
II.4. Langage de programmation.....	18
II.5. Conclusion.....	18
Chapitre III : Sprint zéro	20

III.1. Introduction.....	21
III.2. Spécification des Besoins	21
III.2.1. Identification des acteurs	21
III.2.2. Réalisation du diagramme de contexte	22
III.2.3. Identification des messages échangés.....	22
III.2.4. Capture des besoins fonctionnels.....	23
III.2.5. Capture des besoins non fonctionnels.....	25
III.3. Pilotage du projet avec Scrum	25
III.3.1. Équipe et rôles.....	26
III.3.2. Le Backlog du produit	26
III.3.3. Product Backlog.....	26
III.3.4. Structure et découpage du projet.....	28
III.4. Prototypes et interfaces	29
III.5. Outils de développement.....	31
III.5.1. Papyrus.....	31
III.5.2. NetBeans	32
III.5.3. Oracle Database	32
III.6. Conclusion	33
Chapitre IV : Sprint 1 " Gestion des décisions "	34
IV.1. Introduction.....	35
IV.2. Backlog du sprint 1	35
IV.3. Spécifications fonctionnelles	39
IV.3.1. Diagramme de cas d'utilisation	39
IV.3.2. Diagramme d'activité global du sprint 1	40
IV.3.3. Diagramme d'activité « authentification »	41
IV.3.4. Description textuelle du cas d'utilisation « Établir une décision »	42
IV.4. Analyse	47
IV.4.1. Modèle du domaine	47
IV.4.2. Diagrammes de Séquence Système	48
IV.5. Conception.....	52

I.5.1. Diagrammes de séquences détaillés	53
Les diagrammes ci-dessous est une représentation graphique détaillée des interactions selon leur ordre chronologique. Dans ces diagrammes nous avons noté: IU : interface utilisateur V : Vue, C : Controller, M : Modèle et BD pour Base de données.	Tout
d'abord le diagramme de séquence détaillé du cas d'utilisation « S'authentifier » puis celui de « Établir une décision »	53
IV.4.4. Diagrammes de classes global du premier sprint	54
IV.6. Développement	56
IV.6.1. Schéma de la Base de Données	56
IV.6. Test	57
IV.6.1. L'interface d'authentification	57
IV.6.2. Interface de décision mutation.....	57
IV.6.3. Interface Attestation de travail.....	58
IV.7. Conclusion	59
Chapitre V : Sprint 2	60
" Calcul des éléments de la paie"	60
V.1. Introduction	61
V.2. Backlog du sprint 2	61
V.3. Spécifications fonctionnelles	63
V.3.1. Classification des cas d'utilisation par acteur	63
V.3.2. Diagramme de cas d'utilisation.....	64
V.3.3. Diagramme d'activités	65
V.3.4. Description textuelle des cas d'utilisation	66
V.4. Analyse.....	68
V.4.1. Modèle du domaine.....	68
V.4.2. Diagrammes de séquence système.....	69
V.5. Conception	74
V.5.1. Diagrammes de séquences détaillés.....	75
V.5.2. Diagramme de classes de conception.....	75
V.6. Développement	77

V.6.1. Schéma de la Base de Données.....	77
V.7. Test.....	78
V.7.1. Interface du gestionnaire administratif	79
V.7.2. Interface titre de congé.....	80
V.7.2. Interface des heures supplémentaires.....	80
V.8. Conclusion	81
Conclusion générale.....	82
Bibliographie.....	85
Annexe	i



Table n° 1 : Les éléments en communs entre UP et Scrum.....	17
Table n° 2 : Le Backlog du produit.....	26
Table n° 3 : Planification des sprints.	29
Table n° 4: Les Users Stories du sprint 1.	35
Table n° 5 : Description textuelle du cas d'utilisation « S'authentifier ».....	43
Table n° 6: Description textuelle du cas d'utilisation « Établir une décision ».....	44
Table n° 7: Description textuelle du cas d'utilisation « Valider les documents ».....	46
Table n° 8 : Les Users Story du sprint 2.	61
Table n° 9 : Classification des cas d'utilisation par acteur.	63
Table n° 10: Description textuelle du cas d'utilisation « Mettre à jour les éléments du calcul de la paie ».	67



Figure n° 1: Organigramme de la RTC.....	7
Figure n° 2: Organigramme du département Administration et Social.	8
Figure n° 3: Processus Scrum.....	15
Figure n° 4: Diagramme de contexte dynamique du système.	22
Figure n° 5 : Diagramme de cas d'utilisation global.	25
Figure n° 6 : Prototype de l'interface authentification.	29
Figure n° 7 : Prototype de l'interface d'accueil du gestionnaire administratif.....	30
Figure n° 8 : Prototype de l'interface d'accueil du cadre de gestion administrative.....	31
Figure n° 9 : Diagramme de cas d'utilisation « Établir une décision ».	40
Figure n° 10: Diagramme d'activité global du sprint 1.	41
Figure n° 11 : Diagramme d'activité « S'authentifier ».	42
Figure n° 12: Modèle du domaine du sprint 1.	47
Figure n° 13 : Diagramme de séquence système du cas « S'authentifier».....	48
Figure n° 14: Diagramme de séquence système du cas « Introduire les informations ».	49
Figure n° 15: Diagramme de séquence système du cas « Valider les documents ».	50
Figure n° 16: Diagramme de séquence système du cas « Établir une décision ».	51
Figure n° 17: Diagramme d'activité « Établir une décision de mutation ».	52
Figure n° 18: Diagramme de séquence détaillé du cas d'utilisation « S'authentifier ».	53
Figure n° 19: Diagramme de séquence détaillé du cas d'utilisation « Établir une décision ».	54
Figure n° 20 : Diagramme de classes de conception du sprint 1.	55
Figure n° 21: Schéma de la base de données du sprint 1.....	56
Figure n° 22 : Interface d'authentification.....	57
Figure n° 23 : Interface de décision de mutation.	58
Figure n° 24 : Aperçu d'avant génération en PDF d'une attestation de travail.	58
Figure n° 25 : Diagramme de cas d'utilisation « Calculer les éléments de la paie».	65
Figure n° 26: Vue d'ensemble des tâches effectuée par le gestionnaire administratif.	66

Figure n° 27 : Modèle du domaine du sprint 2.	68
Figure n° 28 : Diagramme de séquence système du cas « Calculer les frais de mission ».	70
Figure n° 29: Diagramme de séquence système du cas « Établir un titre de congé ».	71
Figure n° 30 : Diagramme de séquence système du cas « Établir un contrat de prêt ».	72
Figure n° 31: Diagramme de séquence système du cas « Calculer les heures supplémentaires ».....	73
Figure n° 32: Diagramme de séquence système du cas « Calcul des éléments de la paie ». .	74
Figure n° 33: Diagramme de séquence système du cas « Calcul des éléments de la paie ». .	75
Figure n° 34 : Diagramme de classes global du sprint 2.	77
Figure n° 35 : Schéma de la BDD du sprint 2.	78
Figure n° 36 : Interface du gestionnaire administratif.	79
Figure n° 37 : Interface de titre de congé.	80
Figure n°38 : Interface des heures supplémentaires.	81



Liste des abréviations

ASL	Administration et Social
IHM	Interface Homme Machine
OCL	Object Constraint Language
RTC	Région de Transport Centre
SONATRACH	Société Nationale pour le TRANsport par Canalisation des Hydrocarbures
SOPEG	SOciété PEtolière de Gérance
UML	Unified Modeling Langage
UP	Unified process



Introduction générale

De nos jours, les applications et logiciels informatiques sont devenus très importants et indispensables à la vie quotidienne de tout individu. Ce sont des outils qui facilitent la communication et le traitement de données d'une manière rapide et efficace. C'est la raison pour laquelle les entreprises investissent fortement dans des applications et logiciels, elles cherchent à se procurer les meilleurs outils qui vont leur permettre un maximum de coordination dans la gestion de leurs entreprises.

En étant conscientes de l'importance des applications informatiques dans la gestion d'une entreprise, nous nous sommes rapprochées d'une entreprise, et nous avons effectué un stage pratique au sein de celle-ci. Ce stage était dans le but de proposer une application qui réduirait la lourdeur des traitements et d'automatiser des tâches, ce qui est par conséquent bénéfique pour l'entreprise, et de développer une application qui répondrait à leurs besoins.

Le stage que nous avons effectué au sein de la Région Transport Centre (RTC) de Bejaïa (Société de la SONATRACH), nous a permis de mettre nos connaissances théoriques en pratique et de connaître et de comprendre son fonctionnement, le but de notre travail est de créer et de développer une application qui permettra au service Gestion du département Administration et Social (ASL) de mieux gérer les tâches qui lui sont confiées et d'automatiser les tâches administratives et les documents qui circulent dans ce service. Cette application va leur procurer une bonne gestion régulière, une facilité et rapidité de traitement.

Afin d'atteindre nos objectifs tracés, nous avons organisé et partitionné ce travail en cinq chapitres : Le premier chapitre sera consacré à la partie « **Présentation de l'organisme d'accueil et étude de l'existant** » son principal but est de présenter la société dans laquelle nous avons effectué notre stage ainsi que le service pour lequel nous avons développé cette application.

Le deuxième chapitre qui sera consacré à la partie « **Langages et méthodologie de conception** ». Dans cette partie nous allons présenter les langage UML et OCL que nous avons utilisé lors de la conception de cette application ainsi que la combinaison de la méthodes agile « SCRUM » avec le processus unifié (UP).

Quant au chapitre trois « **Sprint zéro** » sera la première partie dans l'application du cadre méthodologique adopté. Il traitera les fonctionnalités de l'application avec la spécification des besoins fonctionnels et non fonctionnels. Ce chapitre illustrera également le découpage de notre projet avec la planification des différentes activités qui lui sont affiliées.

Le quatrième chapitre comportera notre première itération, le sprint 1, intitulé « **Gestion des décisions** ». Dans ce sprint nous avons intégré le processus unifié à la méthode agile SCRUM et cela en passant par les quatre phases : spécification des besoins, analyse des besoins, conception, implémentation (développement) et test. Nous allons ensuite enchaîner avec le sprint 2 qui sera représenté dans le chapitre 5 « **Calcul des éléments de la paie** » qui comportera aussi les quatre phases du processus unifié. Ce sprint sera dans notre cas la dernière itération.

Chapitre I : Présentation de l'organisme d'accueil et étude de l'existant

I.1. Introduction

L'étude de l'existant est une étape initiale de la réalisation d'une application ou du logiciel informatique, elle est aussi une étape importante et décisive. C'est grâce à cette étude que nous arriverons à voir de plus près les spécificités de l'organisme et recenser ses besoins, ses problèmes et anomalies.

Notre étude se porte sur le département administration et social de la direction régionale de transport centre par canalisation de Bejaia et plus exactement le service Gestion. Cette direction est elle-même une région de l'activité transport de SONATRACH d'où la nécessité de présenter tout d'abord SONATRACH et suivre l'hierarchie de cette dernière jusqu'au département administration et sociale (ASL).

I.2. Présentation de la SONATRACH

SONATRACH acronyme de « **S**ociété **N**ationale pour le **TR**ansport par **C**analisation des **H**ydrocarbures», entreprise publique algérienne créée le 31 décembre 1963, c'est aussi un acteur majeur de l'industrie pétrolière surnommé la major africaine. SONATRACH est classée la première entreprise d'Afrique.



Ce groupe pétrolier et gazier intervient dans l'exploration, la production, le transport par canalisation, la transformation et la commercialisation des hydrocarbures et de leurs dérivés [1].

L'historique de la SONATRACH et ses activités de base ainsi que ses directions régionale de l'activité transport est présenté en annexe

I.2.1. Présentation de la Région Transport Centre

La direction régionale de transport centre de Bejaia (RTC) est l'une des cinq directions régionales de transport des hydrocarbures de la SONATRACH. Sa mission et son but principal est d'acheminer, transporter, stocker et livrer les hydrocarbures liquides et gazeux.

C'est une unité directement rattachée à la division exploitation, elle est chargée de l'exploitation de deux oléoducs¹, d'un gazoduc et d'un port pétrolier.

I.2.1.1. Structure générale de la RTC

A. Direction Régionale : dirigé par un directeur régional aidé par des assistants et un secrétariat.

B. Assistant de sécurité interne : a pour mission de protéger et de sauvegarder le patrimoine humain et matériel.

C. Secrétariat

D. Centre Informatique : il regroupe les moyens d'exploitation et de développement des applications informatiques, pour l'ensemble des structures de la RTC, ainsi que la gestion du réseau informatique interne.

E. Sous-direction Technique : elle a pour mission d'assurer la maintenance et la protection des ouvrages. Elle est organisée en six départements :

- Département maintenance ;
- Département protection cathodique ;
- Département approvisionnement et transport ;
- Département des travaux neufs ;
- Département méthode ;
- Département entretien lignes et bac.

F. Sous-direction Exploitation : elle est chargée de l'exploitation des installations, elle effectue des réparations en cas de fuite, de sabotage ou de panne pour les stations de pompage. Elle est composée de deux départements :

- Département exploitation liquide.
- Département exploitation gaz.

G. Sous-direction Administration et finances : elle a pour mission la gestion des ressources humaines et les moyens généraux, la gestion financière, le budget et le contrôle de

¹**Oléoducs** : Du mot latin oleum « huile » et ducere « conduire », est une canalisation destinée au transport du pétrole.

gestion et de prendre en charge les affaires juridiques de la RTC. Elle est organisée en six départements :

- Département administration et social ;
- Département ressources humaines et communication ;
- Département moyens généraux ;
- Département finance ;
- Département juridique ;
- Département budget et contrôle de gestion.

La figure ci-dessous explique graphiquement sous forme d'un organigramme les différents départements de la RTC.

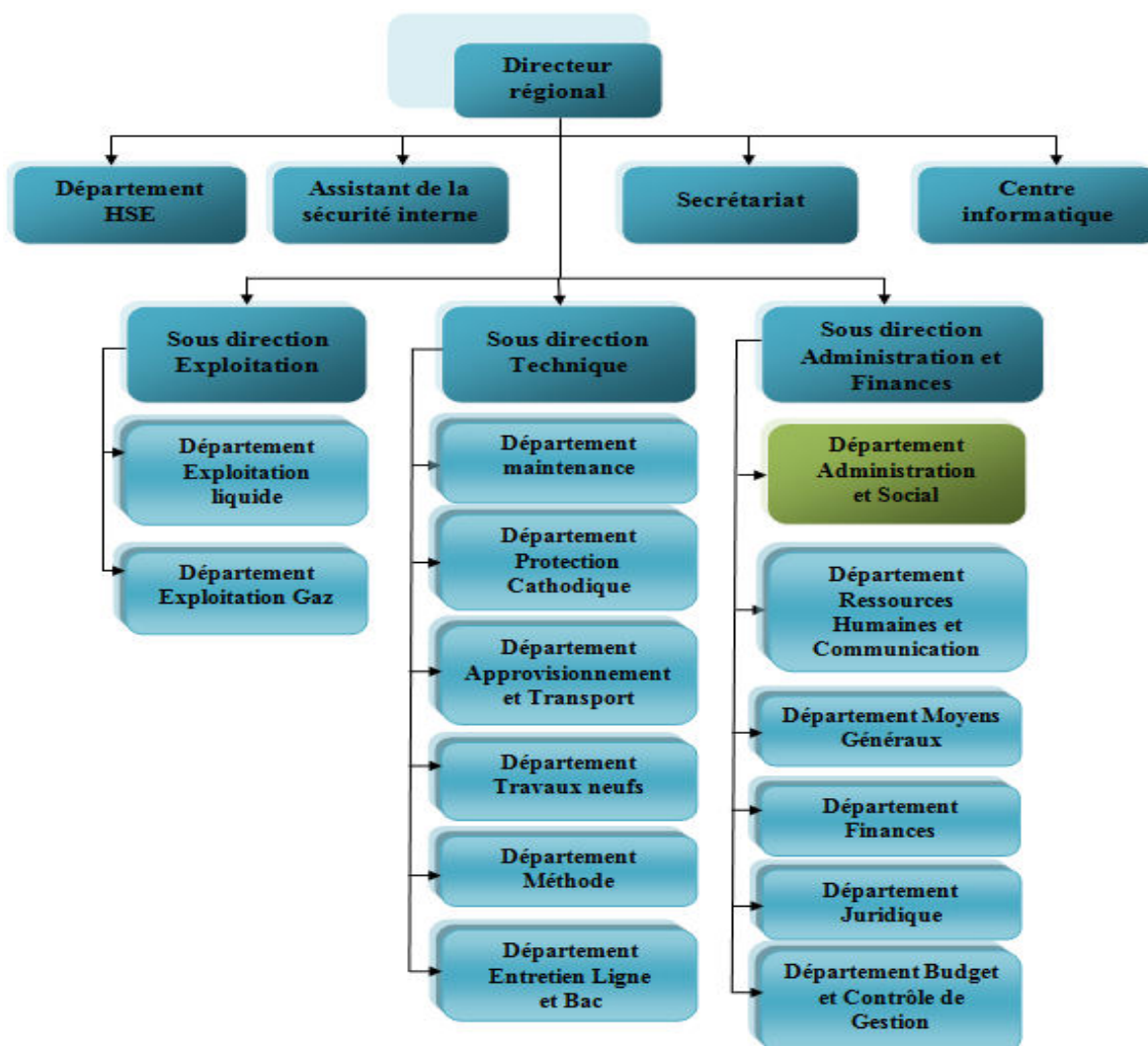


Figure n° 1: Organigramme de la RTC.

I.2.2. Département administration et social

Le département administration et social est l'un des départements de la sous direction administration et finances. Il prend en charge la gestion de la présentation sociale et la tenue du dossier administratif de chaque employé tout au long de son parcours à l'entreprise. Il intervient en cas de désaccord entre les employés et l'entreprise.

Le département ASL se constitue de six services représentés dans la figure ci-dessous. Dans notre étude nous nous intéresserons de plus près au service Gestion puisque c'est pour ce service qu'on fera cette application.

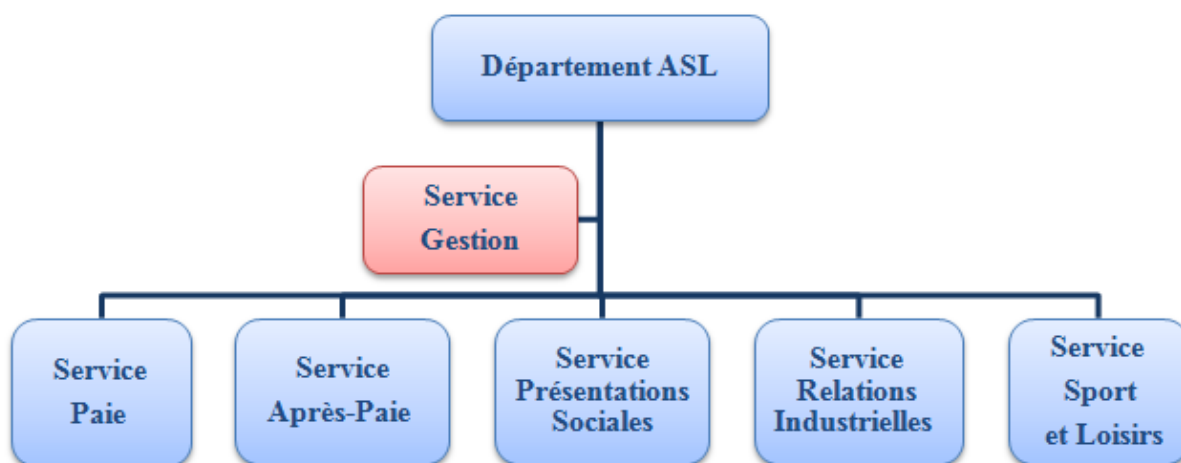


Figure n° 2: Organigramme du département Administration et Social.

I.2.2.1 Missions et objectifs du département ASL




Le département ASL a pour mission:

- Réalisation du suivi administratif de la gestion du personnel (contrats, visites médicales, déclarations aux organismes sociaux, gestion de tout type de congés...) et de la formation continue selon la législation sociale, la réglementation du travail et la politique des ressources humaines de la structure ;
- Réalisation des opérations de paie et de déclarations sociales, à partir de données collectées et vérifiées, en expliquant le contenu auprès du personnel et des organismes concernés ;
- Établir des bulletins de paie;
- Gestion, saisie et calcul des éléments de la paie : préparer les éléments fixes (ancienneté, congés) et variables (intéressement, prime, heures supplémentaires, maladie, accident, indemnités, traitements mensuels, retenues, frais de mission) de la paie, établir les déclarations obligatoires, régler les cotisations sociales;
- Assurer les relations avec les organismes sociaux : mutuelle, caisse primaire d'assurance maladie, médecine du travail, caisse d'assurance retraite et de la santé au travail...etc;

- Préparer tous les documents réglementaires liés à la gestion du personnel : établir les contrats de travail, avenants, soldes de tout compte, assurer les affiliations aux diverses caisses;
- Tenir à jour les dossiers administratifs de chaque salarié.

I.2.2.2. Service Gestion

Le service gestion est l'un des six services du département ASL, il a pour objectif principal la gestion de l'information et la coordination du travail des acteurs ou du personnel de l'organisation. Il est fondamental et essentiel pour l'entreprise puisque c'est lui qui prend en charge la gestion administrative de celle-ci au quotidien. Ce service recrute les postes de travail suivants :

-  **Chef de service gestion** : tient un rôle déterminant dans une entreprise, il doit ainsi planifier et diriger l'ensemble des activités d'un ou de plusieurs secteurs administratifs dont il a la charge ;
-  **Cadre de gestion administrative** : il s'occupe des pointages des cadres supérieurs, et aussi d'établir les décisions de tout type à savoir décision de mutation, détachement, retraite, etc. Dans ce service, il ya trois employés qui occupent ce poste;
-  **Gestionnaire administratif** : ce poste est pris en charge par trois employés et dont les tâches sont citées dans la section suivante.

I.2.2.3. Tâches d'un gestionnaire administratif du service Gestion

Les tâches d'un gestionnaire administratif, on peut citer quelques une :

- Collecte les temps de travail, les informations liées à la production et les autres éléments de paie, à partir des feuilles de présence et d'autres sources;
- Vérifie, calcule et met à jour les éléments de calcul de la paie, tels que les temps de présence, les heures supplémentaires, rémunérations, les vacations, les commissions, ajustements ou augmentations de salaire ;

- Met à jour les informations relatives aux salariés, notamment pour les salariés nouveaux embauchés, les salariés mutés ou ceux qui quittent l'entreprise ;
- Enregistre les jours de congés pris, ainsi que les indemnités de maternité et les dépenses afférentes ;
- Procède aux ajustements nécessaires suite à d'éventuelles erreurs ou divergences ;
- Établit le récapitulatif des ajustements effectués suite à d'éventuelles erreurs ou à des augmentations ;
- Fournit tout type de renseignements ou attestations nécessaires aux employés en matière de salaires ;
- Se tient informé des changements en matière d'imposition ou de déduction qui se répercutent sur l'établissement de la paie.

I.3. Problématique

L'étude détaillée du service gestion du département ASL nous a permis de détecter un certain nombre de problèmes qui ralentissent le bon fonctionnement de ce service. Ces problèmes sont:

- La difficulté des traitements évolue par rapport à l'évolution du nombre d'employés ;
- Le temps des traitements évolue par rapport à l'évolution du nombre d'employés ;
- Le taux d'erreurs est plus élevé lorsque le travail à faire n'est pas automatisé ;
- La possibilité de perte de données dans le cas où on les gère manuellement.

I.4. Solution envisagée

Une fois que nous avons recensé les anomalies et les difficultés que rencontrent les employés du service de gestion au quotidien, nous proposons d'implémenter et concevoir une application de gestion administrative nommée « EasyGest ». Cette application permettra aux futurs utilisateurs (le chef du service, le cadre de gestion administrative et le gestionnaire administratif) d'effectuer leurs tâches en toute sécurité, d'automatiser des fonctionnalités et

tâches administratives qui se faisaient manuellement auparavant, d'alléger et faciliter le traitement des données et le gain de temps. Ce qui va permettre une bonne gestion du service gestion et donc une évolution de ce dernier.

I.5. Conclusion

Suite à l'étude approfondie que nous avons réalisée sur la RTC, et plus précisément le service de gestion du département ASL, nous comprenons davantage la politique de travail adoptée au sein de cette entreprise, et plus exactement sur le processus de traitement des tâches de la gestion administrative. Nous sommes à présent en mesure de poursuivre la réalisation de notre application et ceci en passant à la sélection d'une certaine méthode de conception à suivre.

Chapitre II : Langages et méthodologie de conception

II.1. Introduction

Ce chapitre traite l'aspect méthodologique et décrit les outils utilisés au cours de la réalisation de ce projet. En effet, il concerne beaucoup plus le choix de méthode de conception et outils de développement adoptés.

Cette partie est organisée de la manière suivante: nous allons d'abord commencer par présenter de plus près la méthode agile SCRUM. Nous verrons ensuite les caractéristiques du processus unifié (UP) et enfin nous allons expliquer pourquoi nous avons opté et choisi de combiner entre ces deux méthodes citées auparavant. Pour terminer nous allons présenter le langage de modélisation UML et le langage de spécification OCL.

II.2. Processus de développement

Le développement d'un logiciel informatique nécessite le suivi d'un processus de développement qui définit l'ensemble des étapes à suivre au cours du développement d'une application informatique et puisqu'il n'existe pas un seul processus de développement, ni de processus standard ça reste donc une question de choix. De notre côté, nous avons choisi de combiner le processus unifié (UP) avec Scrum, et avant d'expliquer et d'argumenter notre choix, nous allons tout d'abord présenter brièvement Scrum et UP.

II.2.1. Scrum

Scrum est l'une des méthodes agile. Une méthode agile est une approche itérative et incrémentale, qui est menée dans un esprit collaboratif avec juste ce qu'il faut de formalisme. Elle génère un produit de haute qualité tout en prenant en considération l'évolution des besoins des clients [2].

Le terme Scrum vient de l'anglais qui signifie la mêlée du rugby, et c'est le nom de tout un processus de développement informatique. Scrum offre un cadre de gestion de projet agile des rôles et un rythme itératif.

Le principe sur lequel se base Scrum « Diviser pour régner » consiste à [3] :

- ✓ Découper le temps estimé pour la réalisation du projet en plusieurs blocs de temps appelés itération ou sprint, chaque itération va compter entre 2 à 4 semaines en fonction de la taille du projet ;
- ✓ Découper les besoins en sous besoins appelés « Users stories ». La particularité de ces Users stories est qu'elles vont être assez précises pour être estimées par l'équipe et qu'elles vont être assez petites pour être testées et développées durant une itération;
- ✓ Enfin diviser l'équipe en sous équipes qui comptent entre 6 à 10 personnes.

Les projets qui utilisent la méthode Scrum se forment autour d'une équipe auto-organisée et multifonctionnelle. Il existe trois rôles principaux [4] :

- **Responsable du produit** (ou ce qu'on appelle le **Product Owner**) : définit les spécifications fonctionnelles et communique la vision globale du produit à l'équipe. Autrement dit c'est le représentant du client ;
- **Manager de projet** (le **Scrum master**) : c'est le garant de l'application, son rôle principal est d'éliminer tous les obstacles qui peuvent empêcher l'équipe d'atteindre les objectifs fixés. Le Scrum Master conseille aussi le responsable du produit sur la façon d'exprimer ces besoins mais ce n'est en aucun cas le chef du projet, puisque l'enjeu de l'esprit SCRUM est d'auto organiser l'équipe ;
- **Équipe de développement** : c'est l'équipe qui est mandatée pour réaliser le produit. Généralement composée de 6 à 10 personnes mais pouvant aller jusqu'à 200 personnes. Afin de réaliser ce qui est dans le BACKLOG ² l'équipe de développement organise son travail en sprint³. C'est toute l'équipe qui est responsable du résultat final de chaque sprint. La manière dont sont exécutées les tâches est très libre mais cette liberté doit être néanmoins cadrée par l'obligation de répondre aux objectifs du sprint.

II.2.1.1. Cycle Projet Scrum

La figure ci-dessous représente et décrit en détails le processus à suivre lors de la réalisation d'un projet Scrum.

² **BACKLOG** : un référentiel des besoins, il contient l'ensemble des fonctionnalités du produit souhaité. Chaque fonctionnalité est représentée sous forme de user story.

³ **Sprint** : le cycle de vie SCRUM est rythmé par des itérations de quelques semaines, c'est ce qu'on appelle **les sprints**.

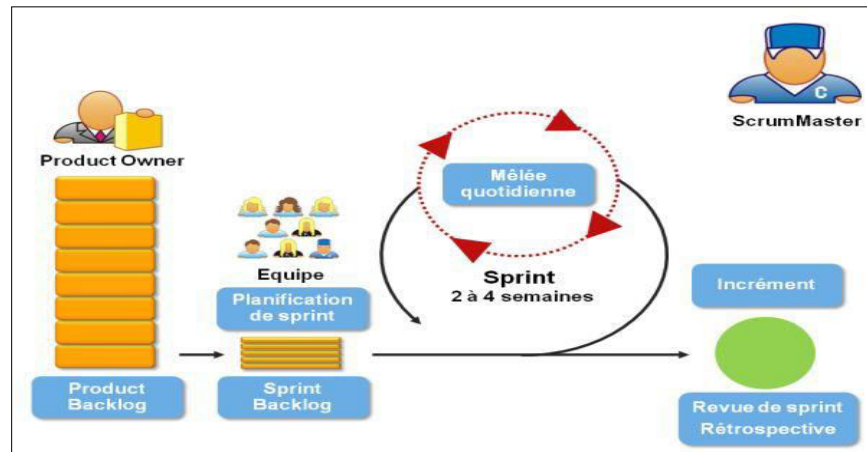


Figure n° 3: Processus Scrum [5].

II.2.2. Processus unifié

Le processus unifié (UP : Unified Process en anglais) est une méthodologie de développement basée sur un cycle de vie itératif et incrémental, piloté par des cas d'utilisation et orienté vers la diminution des risques. Le processus unifié permet d'affecter des tâches et des responsabilités au sein d'une organisation de développement logiciel. C'est aussi une approche à adapter pour des petits projets [6].

II.2.2.1. Cycle de vie du processus unifié

Le processus unifié répète un certain nombre de fois une série de cycles. Tout cycle se conclut par la livraison d'une version du produit aux clients et s'articule en quatre (4) phases : création, élaboration, construction et transition, chacune d'entre elles se subdivisant à son tour en itérations.

Chaque cycle se traduit par une nouvelle version du système. Ce produit se compose d'un corps de code source réparti sur plusieurs composants pouvant être compilés et exécutés et s'accompagne de manuels et de produits associés [7].

II.2.2.2. Caractéristiques du processus unifié

On peut citer quelques unes [8] :

- **UP est piloté par les cas d'utilisation** : les cas d'utilisation sont les véritables pilotes du projet quelle que soit l'activité et la phase ;
- **UP est centré sur l'architecture** : UP se soucie très tôt de l'architecture dans le cycle de vie d'une application ;
- **UP est itérative et incrémentale** : UP procède par itération, de cette manière il est possible de découvrir les erreurs et les incompréhensions plus tôt. Le travail itératif permet à l'équipe de capitaliser à chaque cycle les enseignements du cycle précédent ;
- **UP gère les besoins et les exigences** : les exigences du client changent au cours de la réalisation du projet. UP recommande de les organiser et de les documenter de manière à avoir une approche disciplinée et de pouvoir les filtrer et les trier par propriété et réaliser leur suivi ;
- **UP gère les risques de façon systématique et permanente** : la phase de pré-étude a pour but d'identifier les risques qui pourrait mettre le projet en péril. Tout au long du projet il est recommandé de maintenir une liste de risques classés selon leur danger.
- **UP se soucie en performance de la qualité** : il est nécessaire d'avoir un contrôle permanent et rigoureux des fonctionnalités, fiabilité, des performances de l'application et des performances du système. Pour cela, des tests sont imposés tout au long du cycle de vie. Ainsi les défauts sont vite détectés et le coût de leur correction est minimisé.

II.2.3. Pourquoi choisir scrum et UP

Pour la réalisation de notre projet, nous avons opté pour la combinaison entre SCRUM, une méthode agile qui a fait ses preuves dans plusieurs entreprises dans le monde, avec le processus unifié (UP). Ces deux méthodes partagent une même démarche: itérative, incrémentale et pilotée par des livraisons fréquentes ainsi que d'autres éléments mais dans des terminologies différentes, comme le montre le tableau suivant [9] :

Table n° 1 : Les éléments en communs entre UP et Scrum.

Scrum	UP
Backlog produit	Liste des exigences fonctionnelles ou non fonctionnelles
Backlog de Sprint	Plan d'itération
Produit partiel utilisable	« Prototype » en début de projet (phase d'Elaboration par exemple), « Produit » ensuite
Revue de Sprint	Réunion de fin d'itération.

Ces deux méthodes ont aussi des spécifications différentes l'une de l'autres. Nous trouvons par exemple :

- UP indique certaines dépendances entre activités optionnelles. Scrum rejette le concept d'ordre prévisible des étapes ;
- La planification dans UP est prédictive définissant un plan des exigences établi au début du projet, alors que dans Scrum la planification a plusieurs niveaux et elle peut être modifiée au court de la réalisation ;
- Dans UP l'équipe est dirigée par un chef de projet et les membres sont spécialisés. Tandis qu'en Scrum l'équipe est auto-organisée et multifonctionnelle et soutenue par un Scrum master.

II.3. Langage de modélisation

Même si les applications informatiques sont matérialisées, il est préférable de les représenter sous une forme graphique (modèles). Un modèle est une abstraction qui permet de mieux comprendre un objet réel et complexe. En ce qui concerne les langages de modélisation nous avons utilisé le Langage de Modélisation Unifié (UML) (version 2.5) qui est un langage connu auprès des développeurs informatiques qui permet de faire une représentation informatique (textuelle et graphique de chaque cas). Afin de spécifier notre application, nous avons utilisé le langage Object Constraint Language (OCL) qui est à son tour un langage d'expression des contraintes logicielles utilisé par UML.

II.3.1. OCL

OCL (Object Constraint Language) est un langage de spécification formelle, développé par IBM⁴ en 1995. Il fait partie d'UML depuis la version 1.1 et permet d'exprimer de l'information supplémentaire à propos d'un modèle (un ensemble de diagrammes UML).

OCL permet d'associer des contraintes à des contextes, n'importe quel élément du modèle peut servir de contexte : paquetage, composant, classe, interface, opération, attribut, etc [10].

II.4. Langage de programmation



Concernant l'implémentation de notre future application nous avons opté de programmer avec Java. Le langage Java est un langage de programmation créé en 1995 par Sun Microsystems. Il permet une programmation orientée-objet, avec une syntaxe proche de celle du langage C. Le langage Java a de nombreux avantages, on peut citer [11] :

- **La modularité** : écrire des portions de code génériques, utilisables par plusieurs applications ;
- **La portabilité** : un même programme compilé peut s'exécuter sur différents environnements ;
- **Une riche bibliothèque de classes** : comprenant la gestion des interfaces graphiques (fenêtres, boîtes de dialogue, contrôles, menus, graphisme), la gestion des exceptions;
- **Dynamique** : un programme java peut facilement s'enrichir de nouvelles fonctionnalités sans avoir besoin d'être arrêté.

II.5. Conclusion

Ce chapitre nous a permis de faire le point sur la méthode de conception à adopter afin de réaliser notre future application. C'est une étape qui va nous permettre de débiter et

⁴ **IBM : International Business Machines Corporation** est une société multinationale américaine présente dans les domaines du matériel informatique, du logiciel et des services informatiques.

commencer la réalisation et la création de notre application. En effet, une fois que les outils avec lesquels nous allons travailler seront prêts, nous serons donc en mesure de passer à l'étape de développement de l'application.

Chapitre III : Sprint zéro

III.1. Introduction

Une fois le choix du processus de développement est effectué, nous allons passer à présent à la première étape de la réalisation de notre projet qui est le sprint zéro. C'est au cours de ce sprint que nous allons préparer et mettre en place tout ce qui est nécessaire au lancement du développement de notre application.

Même si le sprint zéro n'a pas tout à fait les caractéristiques d'un sprint puisque il n'aboutit pas à une version livrable à sa fin, mais il reste crucial vu que c'est là où nous spécifierons les besoins et les fonctionnalités, identifierons les différents acteurs, élaborerons le Backlog du produit initial et planifierons les sprints.

III.2. Spécification des Besoins

La capture des besoins est une phase fondamentale dans la réalisation du projet. C'est l'étape qui nous permettrait de définir toutes les interactions entre le système et son environnement. Ainsi nous identifierons qui va interagir avec le système et quelles fonctionnalités le système va fournir aux acteurs.

III.2.1. Identification des acteurs

D'après [11], un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données. Voici donc les différents acteurs qui interagissent avec le système étudié :

- Le chef de service ASL ;
- Le cadre de gestion ;
- Le gestionnaire administratif ;
- L'administrateur système: celui-ci a accès à toutes les fonctionnalités de l'application. Il gère tous les comptes de tous les utilisateurs et les mots de passe ;

- L'opérateur de maintenance de l'application : il veille au bon fonctionnement et la maintenance de l'application.

III.2.2. Réalisation du diagramme de contexte

Le diagramme de contexte n'est pas désigné comme diagramme UML, il ne fait pas partie des 13 diagrammes officiels, mais il est utile pour la définition des acteurs avant de commencer la conception. Il présente le système à modéliser sous forme de boîte noire et les différents acteurs qui interagissent avec le système. La figure suivante représente le diagramme de contexte de l'application de gestion « EasyGest » [11].

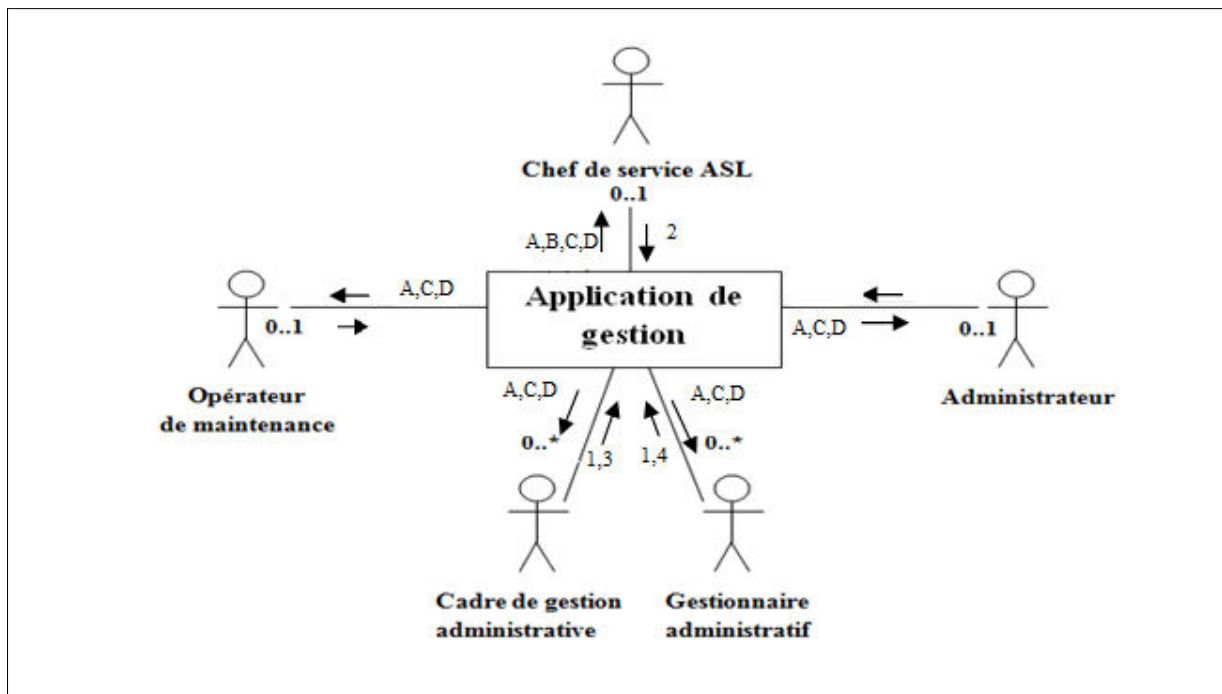


Figure n° 4: Diagramme de contexte dynamique du système.

III.2.3. Identification des messages échangés

Un message représente la spécification d'une communication unidirectionnelle entre objets qui transporte de l'information avec l'intention de déclencher une activité chez le

récepteur. Un message est normalement associé à deux occurrences d'événements: un événement d'envoi et un événement de réception [12].

III.2.3.1. Messages émis par le système

- A. L'affichage des interfaces à la demande de l'utilisateur ;
- B. Affiche un message demandant la validation des données ou des modifications effectuées ;
- C. Affiche un message confirmant que l'opération s'est bien effectuée ;
- D. Affiche des messages d'erreurs, demande de rectifier et de refaire le traitement.

III.2.3.2. Messages reçus par le système

- 1. La recherche et la confirmation des données personnelles de l'employé ;
- 2. Validation des données introduites auparavant;
- 3. Demande le suivi des déplacements de l'employé (mission, détachement, mutation..) ;
- 4. Demande de calculer le total des frais de mission, nombre des heures supplémentaires.

III.2.4. Capture des besoins fonctionnels

Cette section concerne l'étude du domaine de l'application. Il s'agit de déterminer les frontières, le rôle, les ressources disponibles et requises, les contraintes d'utilisation et de performance, etc. Cette étape a pour but d'éviter de développer un logiciel non adéquat.

La spécification fonctionnelle se traduit par un diagramme de cas d'utilisation qui donne une vue extérieure et globale du système et définit les liens entre les utilisateurs et toutes les fonctionnalités que propose ce système. Mais avant de passer au diagramme de cas d'utilisation, il faut tout d'abord identifier ces cas d'utilisation.

III.2.4.1. Identification des cas d'utilisation

Un cas d'utilisation représente un ensemble de séquences d'actions qui sont réalisées par le système. Il permet de décrire ce que le futur système devra faire, sans spécifier comment il le fera [11]. En se basant sur les renseignements recueillis au près des employés du service

gestion du département ASL (les gestionnaires administratifs et les cadres de gestion administrative), nous avons établi une liste des tâches effectuées par ces derniers. À partir de cette liste (qui est citée précédemment), nous allons déduire les différentes fonctionnalités du futur système. De ce fait nous avons distingué les cas d'utilisation suivants pour chaque acteur:

Acteur 1 : gestionnaire administratif

- ✓ Calculer les éléments de la paie (à savoir frais de missions, congés, heures supplémentaires, etc) ;
- ✓ Établir une ou plusieurs attestations de travail.

Acteur 2 : cadre de gestion administrative

- ✓ Établir tout types de décisions à savoir des décisions de mutation, détachement, retraite, promotion...etc ;
- ✓ Établir une ou plusieurs attestations de travail.

Acteur 3 : chef de service

- ✓ Validation des modifications et des sauvegardes effectuées par les gestionnaires administratifs et les cadres de gestion administrative.

III.2.4.2. Diagramme de cas d'utilisation

Avant de se lancer dans la réalisation d'un logiciel ou d'une application, Il faut comprendre, clarifier et bien structurer les attentes et les besoins du client. Le diagramme de cas d'utilisation est un schéma qui montre les cas d'utilisation reliés par des associations à leurs acteurs. Un cas d'utilisation doit être relié au moins à un acteur. La figure suivante représente les cas d'utilisation établis lors du sprint zéro.

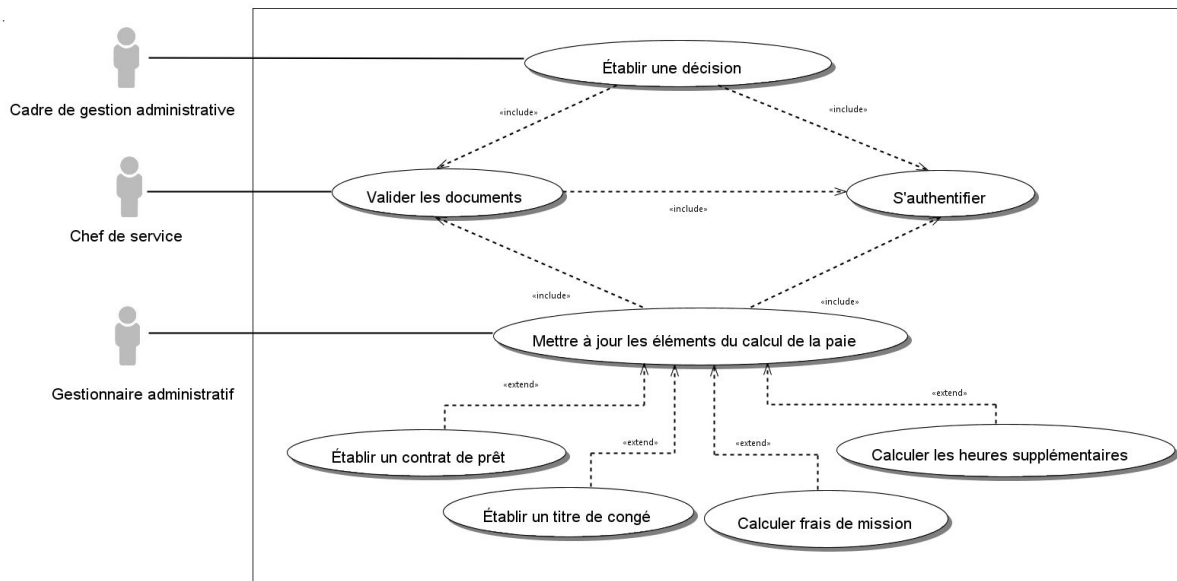


Figure n° 5 : Diagramme de cas d'utilisation global.

III.2.5. Capture des besoins non fonctionnels

Une fois les besoins fonctionnels sont identifiés, nous allons passer à présent à la spécification des besoins non-fonctionnels. Les principaux besoins non-fonctionnels de notre application sont les suivants :

- ☑ **Sécurité** : l'utilisateur doit s'authentifier afin d'accéder aux fonctionnalités de l'application ;
- ☑ **Temps de réponse** : l'application doit réagir en espace de quelques secondes ;
- ☑ **Fiabilité** : les accès doivent être extrêmement sûrs et sécurisés ;
- ☑ **Performance** : l'application doit répondre aux exigences des usagers ;
- ☑ **Convivialité** : l'application finale doit être conviviale, facile à utiliser et à manipuler par les utilisateurs.

III.3. Pilotage du projet avec Scrum

Dans le cadre d'un projet Scrum, nous sommes dans l'obligation de respecter le principe de « diviser pour régner ». Un principe sur lequel est basée la méthodologie de

SCRUM. Ce principe consiste à diviser l'équipe, le temps, les besoins (principe expliqué dans la section précédente II.2.1. Scrum). Donc pour la conduite de notre projet, nous avons appliqué ce principe de la manière citée ci-dessous.

III.3.1. Équipe et rôles

La réussite d'un projet Scrum en équipe est basée sur la coordination et la communication entre chaque membre de l'équipe. L'équipe est constituée du Product Owner, Scrum Master et l'équipe de développement. Dans notre cas l'équipe est constituée que de deux membres qui sont : ABBACI Narimane et ALITOUCHE Lamia, cela ne nous a pas empêché de former une équipe ou plutôt un binôme auto-responsable et multifonctionnel, et de respecter les autres principes de Scrum.

III.3.2. Le Backlog du produit

Après une réunion avec le Product Owner une première liste des exigences est établie. Cette liste est nommée «Backlog Product ». C'est un référentiel des besoins. Il est exprimé sous forme d'un ensemble de user story. Une fois que celui-ci est bien complet et clair, l'équipe de travail devra vérifier et discuter avec le Product Owner les estimations et les priorités établies pour chacune des fonctionnalités présentes.

III.3.3. Product Backlog

Le tableau suivant regroupe les Users stories qui composent le Product Backlog initial, chaque user story a un certain degré d'importance et de risque.

Table n° 2 : Le Backlog du produit.

Id	Thème (features)	Id U.S	User Story	Importance	Risque
1	Authentification	1.1	En tant qu'utilisateur, je dois m'authentifier pour accéder à mon espace au sein de	Moyenne	Faible

			l'application.		
2	Validation des documents, modification	2.1	En tant que chef de service, je souhaite valider des modifications, des documents.	Très élevée	Moyen
3	Gérer les décisions	3.1	En tant que cadre de gestion administrative, je souhaite établir une décision de retraite.	Élevée	Fort
		3.2	En tant que cadre de gestion administrative, je souhaite établir une décision de mutation.	Élevée	
		3.3	En tant que cadre de gestion administrative, je souhaite établir une décision de mise en disponibilité.	Élevée	
		3.4	En tant que cadre de gestion administrative, je souhaite établir une décision de détachement.	Élevée	
		3.5	En tant que cadre de gestion administrative, je souhaite établir une décision d'augmentation.	Élevée	
		3.6	En tant que cadre de gestion	Élevée	

			administrative, je souhaite établir une décision de promotion.		
4	Mettre à jour les éléments du calcul de la paie	4.1	En tant que gestionnaire administratif, je souhaite établir un titre de congé.	Élevée	Fort
		4.2	En tant que gestionnaire administratif, je souhaite établir un contrat de prêt.	Élevée	
		4.3	En tant que gestionnaire administratif, je souhaite calculer les frais de mission.	Élevée	
		4.4	En tant que gestionnaire administratif, je souhaite calculer les heures supplémentaires.	Élevée	

III.3.4. Structure et découpage du projet

Après la validation du Backlog Product de la part du Product Owner, nous passons à présent à la planification de notre travail. Nous avons constaté que les fonctionnalités sont liées donc elles vont être regroupées dans un seul release⁵. Ce release est constitué de deux sprints successifs, le premier pour la gestion des décisions et le second pour la gestion des éléments de la paie comme le résume le tableau suivant :

⁵ **Release** : est une période de temps à l'issue de laquelle une version du livrable est proposée. Ce release est constitué de plusieurs sprints.

Table n° 3 : Planification des sprints.

N° du Sprint	Les tâches à faire	La durée
1	Gestion des décisions	29/04/2018 au 27/05/2018
2	Gestion des éléments de la paie	28/05/2018 au 20/06/2018

III.4. Prototypes et interfaces

Voici pour une première vue un prototype de l'interface d'authentification. Cette dernière sera la première interface qui va s'afficher dès que l'utilisateur lance l'application afin de se connecter au système et accéder aux fonctionnalités de l'application qui lui sont autorisées.



Figure n° 6 : Prototype de l'interface authentification.

Voici ensuite une première vue de l'interface d'accueil du gestionnaire administratif. Cette dernière sera la première interface qui va s'afficher dès que l'utilisateur, qui est dans ce cas le gestionnaire administratif, s'authentifie et accède au système. Les tâches qui lui sont réservées vont être affichées sur cette interface.



Figure n° 7 : Prototype de l'interface d'accueil du gestionnaire administratif.

La figure ci-après de même représente une première vue de l'interface d'accueil du cadre de gestion administrative. Celle-ci sera la première interface qui va s'afficher dès que l'utilisateur, qui est dans ce cas le cadre de gestion administrative, s'authentifie. Toutes les tâches qui lui sont autorisées d'accomplir lui sont affichées sur cette interface.



Figure n° 8 : Prototype de l'interface d'accueil du cadre de gestion administrative.

III.5. Outils de développement

Il existe une multitude d'outils d'aide à la réalisation d'une application informatique, que ce soit pour la modélisation, le développement ou la gestion de la base de données. Dans notre cas nous avons opté et choisi de travailler à l'aide des outils que nous allons présenter dans ce qui suit.

III.5.1. Papyrus



Papyrus est un ensemble de plugins Eclipse faisant partie du projet Eclipse Modeling Project. Il s'agit d'un nouvel environnement pour éditer tout type de modèle EMF (Eclipse Modeling Framework) et tout particulièrement UML 2 (Unified Modeling Language la norme de l'OMG) [13].

Nous avons utilisé la version: Neon (4.6.0), afin de réaliser tous les diagrammes UML qui se trouvent dans ce rapport mais aussi pour représenter les contraintes OCL du diagramme de classes. C'est aussi un environnement qui permet la génération partielle du code, c'est une fonctionnalité que nous allons utiliser lors de l'implémentation de notre application.



III.5.2. NetBeans

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun⁶ en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2. En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le Groovy, le PHP et le HTML, ou d'autres (dont Python et Ruby). Il offre toutes les facilités d'un IDE moderne (éditeur en couleurs, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

⁶ **Sun Microsystems** était un constructeur d'ordinateurs et un éditeur de logiciels américain. Sun est racheté par Oracle Corporation le 20 avril 2009.

NetBeans constitue par ailleurs une plate forme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plate forme [14].

Nous avons développé notre application à l'aide de la version (8.2) de NetBeans. Elle nous a permis d'implémenter les interfaces et coder les programmes.

III.5.3. Oracle Database



Oracle Database est un système de gestion de base de données relationnelle (SGBDR) qui depuis l'introduction du support du modèle objet dans sa version 8 peut être aussi qualifié de système de gestion de base de données relationnel-objet (SGBDRO). Fourni par Oracle Corporation, il a été développé par Larry Ellison, accompagné entre autres, de Bob Miner et Ed Oates [15].

Nous allons travailler avec Oracle Database 11g Express Edition à l'aide de SQLDeveloper (version 18.1.0.095) comme gestionnaire de base de données, vu notre choix pour une base de données relationnelle-objet.

III.6. Conclusion

A ce stade du développement, nous avons pu atteindre notre objectif qui consiste à spécifier les besoins en identifiant les acteurs ainsi que les cas d'utilisation, et d'établir un plan de travail pour le développement de l'application et de présenter quelques prototypes d'interfaces. Ce qui nous permettrait d'enchaîner avec le premier sprint dans le chapitre suivant.

Chapitre IV : Sprint 1 " Gestion des décisions "

IV.1. Introduction

Le chapitre précédent concerne l'analyse des besoins et la planification du travail. Pour commencer ce chapitre et débiter ce premier sprint qui va se baser sur la gestion des décisions, nous avons dressé un tableau (le tableau ci-dessous) avec les fonctionnalités à développer dans ce sprint. Chaque fonctionnalités est représentée sous forme de user story. Elles sont aussi découpées en tâches qui sont affectées à un membre de l'équipe. Ces Users stories vont passer par les quatre étapes du processus UP, précisément la spécification fonctionnelle, la conception, le développement et les tests.

IV.2. Backlog du sprint 1

Nous allons commencer tout d'abord par définir les users stories qui composent le Product Backlog de ce sprint 1, chaque user story est décomposée en tâches qui à leur tours sont affectées aux membres de l'équipe, chaque membre a une tâche à réaliser (comme le montre le tableau suivant :

Table n° 4: Les Users Stories du sprint 1.

Id U.S	User Story	Id Taches	Tâches	Affectation	Durée
1.1	En tant qu'utilisateur, je dois m'authentifier pour accéder à mon espace au sein de la plateforme.	1.1. A	Réaliser les diagrammes de cas d'utilisation, description textuelle de la fonctionnalité «s'authentifier », diagramme de séquence, de classes de conception,	A.Narimane	8h
		1.1.B	Développer le cas «S'authentifier ».	A.Lamia	6 h

		1.1.C	Tester le cas « S’authentifier ».	A.Narimane	1 h
2.1	En tant que chef de service, je souhaite valider les modifications ou les documents.	2.1. A	Réaliser les diagrammes de cas d’utilisation, séquence, de classe de conception, description textuelle de la fonctionnalité «Valider les documents ».	A.Narimane	8h
		2.1. B	Développer le cas «Valider les documents ».	A.Lamia	24 h
		2.2.C	Tester le cas «Valider les documents ».	A.Narimane	2h
3.1	En tant que cadre de gestion administrative, je souhaite établir une décision de retraite	3.1. A	Réaliser les diagrammes de cas d’utilisation, séquence, de classes de conception, description textuelle de la fonctionnalité «Établir une décision de retraite ».	A.Lamia	8h
		3.1. B	Développer le cas «Établir une décision de retraite ».	A.Narimane	24 h
		3.1. C	Tester le cas «Établir une décision de retraite».	A.Lamia	3h
3.2	En tant que cadre de gestion administrative, je	3.2. A	Réaliser les diagrammes de cas d’utilisation, séquence, de classe de conception,	A.Narimane	8h

	souhaite établir une décision de mutation		description textuelle de la fonctionnalité «Établir une décision de mutation ».		
		3.2. B	Développer le cas «Établir une décision de mutation ».	A.Lamia	36 h
		3.2. C	Tester le cas «Établir une décision de mutation».	A.Narimane	3h
3.3	En tant que cadre de gestion administrative, je souhaite établir une décision de mise en disponibilité	3.3. A	Réaliser les diagrammes de cas d'utilisation, séquence, de classe de conception, description textuelle de la fonctionnalité «Établir une décision de mise en disponibilité».	A.Lamia	8h
		3.3. B	Développer le cas «Établir une décision de mise en disponibilité».	A.Narimane	24h
		3.3. C	Tester le cas «Établir une décision de mise en disponibilité».	A.Lamia	2h
3.4	En tant que cadre de gestion administrative, je souhaite établir une décision de détachement	3.4. A	Réaliser les diagrammes de cas d'utilisation, séquence, de classe de conception, description textuelle de la fonctionnalité «Établir une décision de détachement ».	A.Narimane	8 h

		3.4. B	Développer le cas «Établir une décision de détachement ».	A.Lamia	36 h
		3.4. C	Tester le cas «Établir une décision de détachement».	A.Narimane	2h
3.5	En tant que cadre de gestion administrative, je souhaite établir une décision d'augmentation	3.5. A	Réaliser les diagrammes de cas d'utilisation, séquence, de classe de conception, description textuelle de la fonctionnalité «Établir une décision d'augmentation ».	A.Lamia	8h
		3.5. B	Développer le cas «Établir une décision d'augmentation »	A.Narimane	36 h
		3.5. C	Tester le cas «Établir une décision d'augmentation»	A.Lamia	2h
3.6	En tant que cadre de gestion administrative, je souhaite établir une décision de promotion	3.6. A	Réaliser les diagrammes de cas d'utilisation, séquence, de classe de conception, description textuelle de la fonctionnalité «Établir une décision de promotion»	A.Narimane	8 h
		3.6. B	Développer le cas «Établir une décision de promotion »	A.Lamia	24 h

		3.7. C	Tester le cas «Établir une décision de promotion»	A.Narimane	2 h
--	--	--------	---	------------	-----

IV.3. Spécifications fonctionnelles

Dans ce premier sprint nous allons nous concentrer sur les tâches qu'effectue le cadre de gestion administrative. Ce dernier s'occupe principalement d'établir les décisions de différentes sortes telles que les décisions de mutation, détachement, retraite, promotion, etc ; chaque document établi nécessite une validation de la part du chef de service. De ce fait nous déduisons que ces cas d'utilisations sont reliés de la manière suivante :

- ✓ L'exécution de tous les cas d'utilisation inclut le cas d'utilisation « s'authentifier », par exemple si le cadre de gestion administrative veut établir une décision il doit tout d'abord s'authentifier ;
- ✓ Toute décision établie et toute modification apportée doit être validée par le chef de service, donc tous les cas d'utilisation incluent le cas « valider les documents » ;
- ✓ Il y a plusieurs type de décision (mutation, retraite, détachement, promotion, augmentation), il y a donc une relation de généralisation entre le cas d'utilisation « Établir une décision » et les autres cas d'établissement des autres décisions ;
- ✓ Le cas d'utilisation « établir une décision de retraite » inclut le cas d'utilisation « établir un solde de tout solde ».

IV.3.1. Diagramme de cas d'utilisation

La figure suivante représente un diagramme de cas d'utilisation. Ce diagramme est une description détaillée et organisée des cas d'utilisation identifiés auparavant, ainsi que les relations qui existent entre eux.

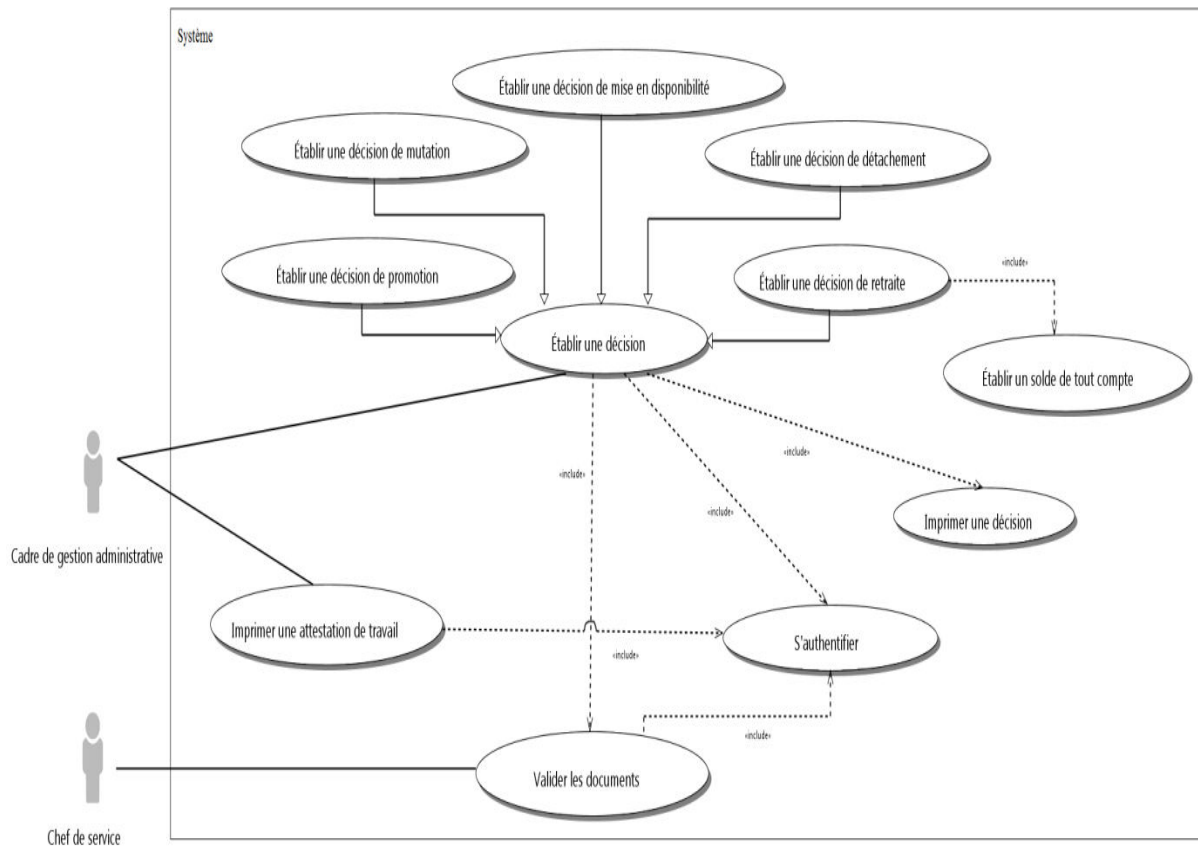


Figure n° 9 : Diagramme de cas d'utilisation « Établir une décision ».

IV.3.2. Diagramme d'activité global du sprint 1

La figure suivante représente une vue globale des tâches effectuées par les cadres de gestion administrative à partir de leur authentification jusqu'à l'établissement du document souhaité. La figure ci-après est sous forme d'un diagramme d'activité qui est plus simple interpréter.

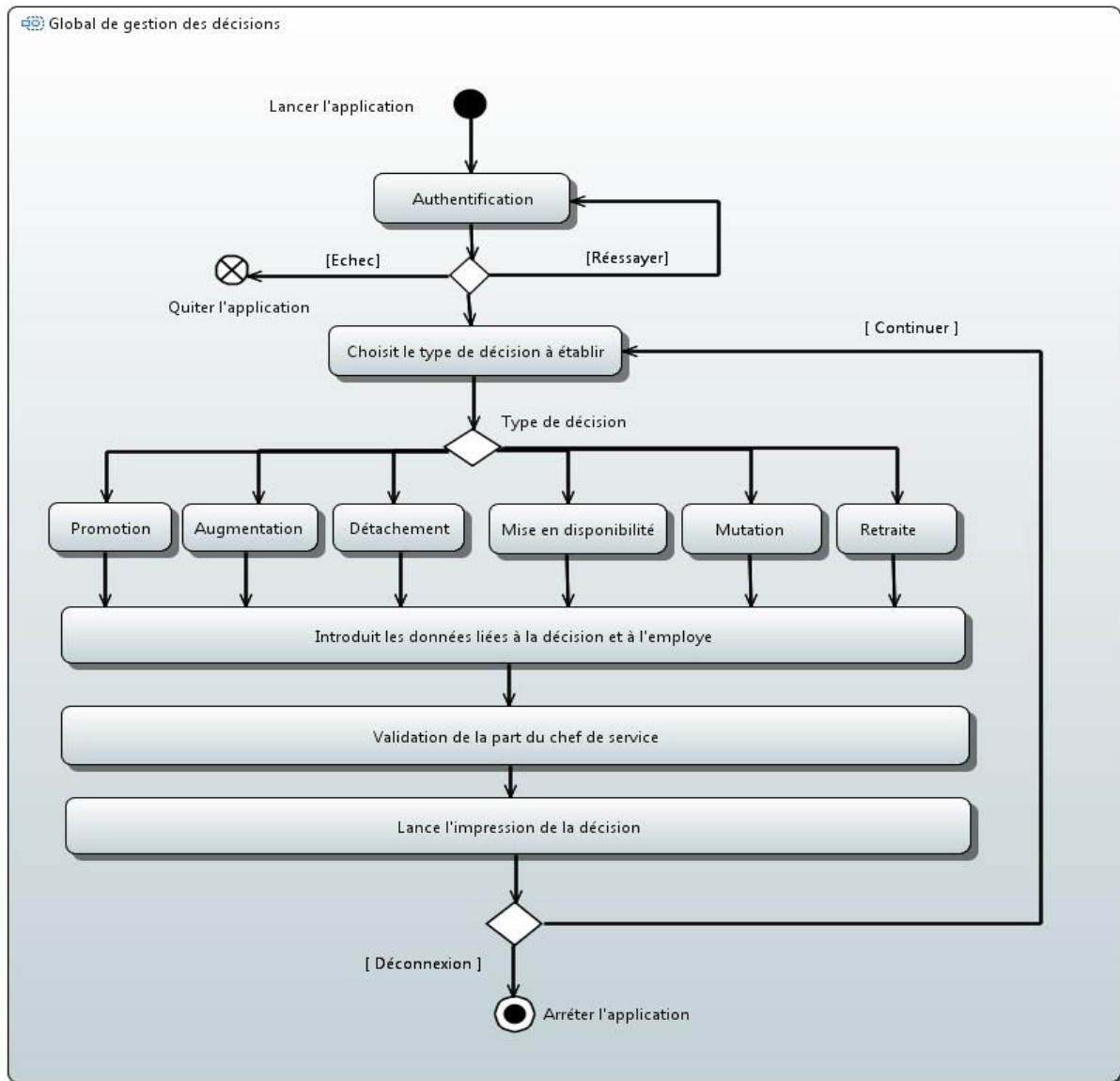


Figure n° 10: Diagramme d'activité global du sprint 1.

IV.3.3. Diagramme d'activité « authentification »

L'utilisateur doit s'authentifier pour pouvoir accéder aux différentes fonctionnalités de l'application. Il doit introduire l'identifiant et le mot de passe qui lui correspondent. Le système vérifie la validité de ces informations introduites, si ces dernières sont correctes le

Le système donne l'accès à l'utilisateur, dans le cas contraire le système affiche un message d'erreur demandant à l'utilisateur soit de rectifier ou de compléter les champs vides.

Nous avons choisi de présenter ce cas d'utilisation sous forme d'un diagramme d'activité pour mieux documenter et éclaircir le fonctionnement de ce cas d'utilisation et mettre en évidence les informations ainsi que l'enchaînement des étapes du scénario comme le montre la figure suivante.

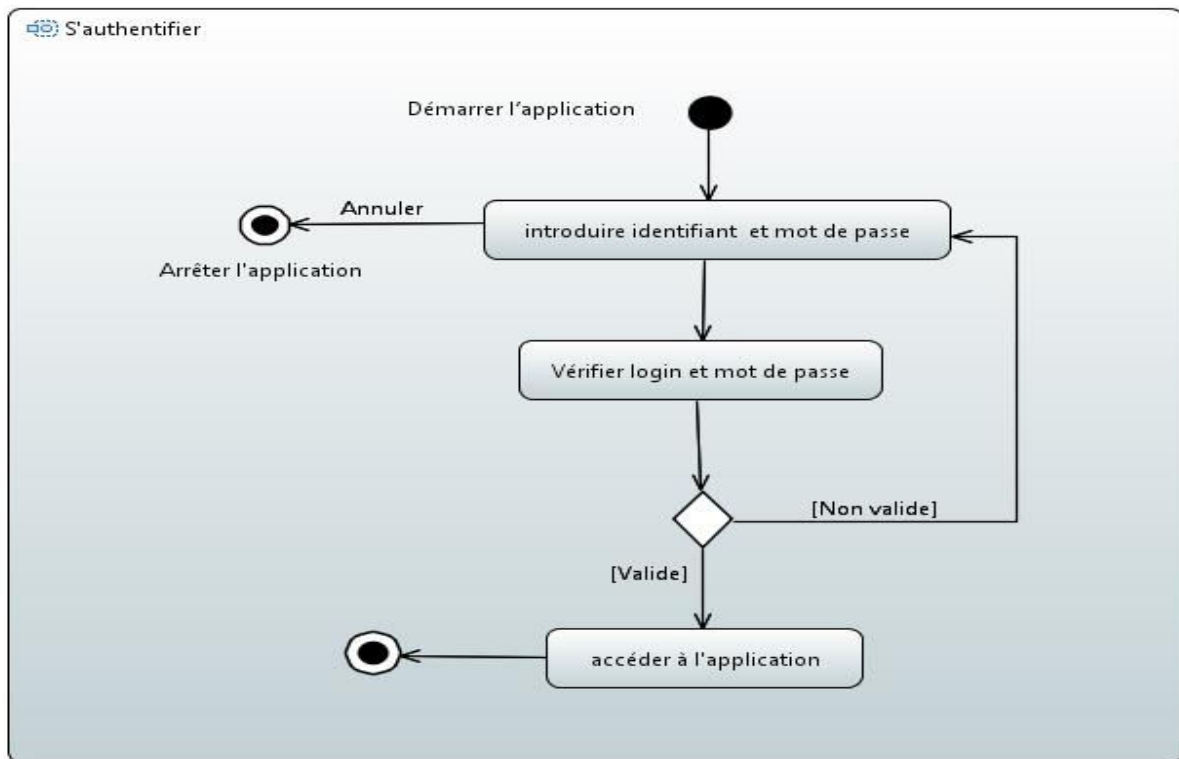


Figure n° 11 : Diagramme d'activité « S'authentifier ».

IV.3.4. Description textuelle du cas d'utilisation « Établir une décision »

Une fois les cas d'utilisation sont identifiés et organisés (définition des relations entre les cas d'utilisation), il est indispensable de décrire textuellement ces cas d'utilisation. Bien que le diagramme de cas d'utilisation décrit les grandes fonctions d'un système du point de vue des acteurs, mais n'expose pas de façon détaillée le dialogue entre les acteurs et les cas

d'utilisation. C'est pourquoi il est recommandé de rédiger une description textuelle, cette description permet de communiquer d'une manière plus facile avec les utilisateurs et de s'entendre sur la terminologie métier.

La description textuelle des cas d'utilisation se fera cas par cas comme le montre les tableaux suivants.

IV.3.4.1. Cas d'utilisation « s'authentifier »

Table n° 5 : Description textuelle du cas d'utilisation « S'authentifier ».

Cas d'utilisation	S'authentifier.
But et objectif	Connexion au système et accès sécurisé aux fonctions qui lui sont réservées.
Acteur	Gestionnaire, cadre de gestion administrative ou chef de service.
Pré-condition	<ul style="list-style-type: none">• L'utilisateur doit lancer l'application ;• L'utilisateur doit s'authentifier correctement en tant que gestionnaire, cadre de gestion ou chef de service.
Scénario nominal	<ol style="list-style-type: none">1. L'utilisateur accède à l'application ;2. Le système affiche l'interface d'authentification ;3. L'utilisateur saisie son identifiant et son mot de passe pour se connecter au système ;4. Une requête de recherche portant l'identifiant de l'utilisateur se déclenche dans la base de données afin de vérifier l'existence du compte et la conformité des informations saisies et d'afficher l'interface réservée à cet utilisateur ; [A]5. le système donne l'accès à l'utilisateur.
Enchaînement Alternatif [A]	<p>A1 : champs obligatoires vides</p> <p>5. Le système indique à l'utilisateur qu'il y a un champ obligatoire vide, le scénario reprend au point de 2 du scénario nominal.</p> <p>A2 : mot de passe erroné</p>

	5. Le système indique à l'utilisateur que le mot de passe introduit est erroné, le scénario reprend à partir de 2 du scénario nominal.
Enchaînement d'erreur [E]	E1 : identifiant n'existe pas Le système indique à l'utilisateur que l'identifiant n'est pas valide et n'existe pas, le scénario reprend au point 2 du scénario nominal.
Post-condition	l'utilisateur peut faire les tâches qui lui sont autorisées et se déconnecte et ferme son compte une fois son travail fini.
Besoin d'IHM	Formulaire d'authentification / Interface d'authentification.

IV.3.4.2. Cas d'utilisation « Établir une décision »

Table n° 6: Description textuelle du cas d'utilisation « Établir une décision ».

Cas d'utilisation	Établir une décision.
But et objectif	Permettre à l'utilisateur d'ajouter et saisir la décision prise à l'égard de l'employé concerné puis l'imprimer.
Acteur	Cadre de gestion.
Pré-condition	<ul style="list-style-type: none"> • L'utilisateur doit s'authentifier correctement; • La décision doit être prise d'abord par les autorités concernées et saisie par un autre service à travers un autre système.

<p align="center">Scénario nominal</p>	<ol style="list-style-type: none"> 1. L'utilisateur choisit l'interface d'établissement d'une décision ; 2. L'utilisateur sélectionne le type de décision à établir ; 3. L'utilisateur introduit le matricule de l'employé concerné par la décision; 4. Le système affiche automatiquement toutes les données personnelles qui ont été déjà saisies à travers un autre système ; 5. L'utilisateur saisi la date de la décision selon le type celle-ci et toutes les informations qui la concernent ; 6. Le système demande la vérification et la confirmation des informations introduites avant de sauvegarder ; 7. L'utilisateur confirme et enregistre les informations introduites ; [A1] [A2] 8. Le système sauvegarde provisoirement les modifications apportées en attendant la validation du chef de service ; 9. Le chef de service valide les modifications apportées concernant la décision ; [A3] 10. Le système fait une sauvegarde ; 11.L'utilisateur lance l'impression de la décision ; 12. Le système exécute la requête de l'utilisateur.
<p align="center">Enchaînement Alternatif [A]</p>	<p>A1 : champs d'informations incomplets ou informations erronées</p> <p>6. Le système affiche un message d'erreur à l'utilisateur, le scénario reprend au point 3 du scénario nominal.</p> <p>A2 : sauvegarde impossible des modifications apportées</p> <p>6. Le système affiche un message d'erreur, le scénario reprend au point 3 du scénario nominal.</p> <p>A3 : le chef de service ne valide pas la décision</p> <p>6. Le système affiche un message indiquant que la décision n'a pas été validée, le scénario reprend au point 3 du scénario nominal</p>
<p align="center">Enchaînement d'erreur [E]</p>	<p>E1 : annulation de la modification</p> <p>L'enchaînement E1 peut démarrer du point 2 au point 5.</p> <p>2.5 L'utilisateur annule la saisie et la modification et le cas</p>

	d'utilisation se termine en échec.
Post-condition	<ul style="list-style-type: none"> • La modification est sauvegardée ; • Établissement et impression de la décision une fois le document validé par le chef de service.
Besoin d'IHM	<ul style="list-style-type: none"> • Formulaire d'établissement de décision ; • L'interface de validation des décisions ; • Le système doit être lié à une imprimante.

IV.3.4.3. Cas d'utilisation « Valider les documents »

Table n° 7: Description textuelle du cas d'utilisation « Valider les documents ».

Cas d'utilisation	Valider les documents.
But et objectif	La confirmation des modifications et des mises à jour faites aux données concernant les employés.
Acteur	Chef de service gestion.
Pré-condition	L'utilisateur doit s'authentifier.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur demande l'interface de validation ; 2. Le système affiche l'interface demandée avec les documents à valider dans le cas où il y'a des documents à valider ; 3. L'utilisateur vérifie les informations puis valide ; [A1] 4. Le système prend en considération la validation et sauvegarde les modifications.
Enchaînement Alternatif [A]	<p>A1. L'utilisateur ne valide pas le document</p> <p>4. L'utilisateur introduit une note exprimant les raisons du non validation du document. Cette note s'affichera pour le cadre de gestion administrative ou le gestionnaire administratif. Le scénario reprend à l'étape d'introduction des informations par le cadre de gestion administrative ou bien le gestionnaire administratif.</p>

Post-condition	Impression des documents une fois validés par le chef de service.
Besoin d'IHM	L'interface de validation.

IV.4. Analyse

La phase analyse est une étape qui consiste à analyser et étudier les besoins du client et de l'utilisateur, c'est une sorte de reformulation des besoins du client sous une forme exploitable en conception. Elle permet de déterminer du « quoi » et du « quoi faire » d'une application.

IV.4.1. Modèle du domaine

La phase d'analyse du domaine permet d'élaborer la première version du diagramme de classes appelée modèle du domaine. Ce modèle doit définir les classes qui modélisent les entités ou concepts présents dans le domaine (ou métier) de l'application. Il s'agit donc de produire un modèle des objets du monde réel dans un domaine donné. Ces entités ou concepts peuvent être identifiés directement à partir de la connaissance du domaine ou par des entretiens avec des experts du domaine [17].

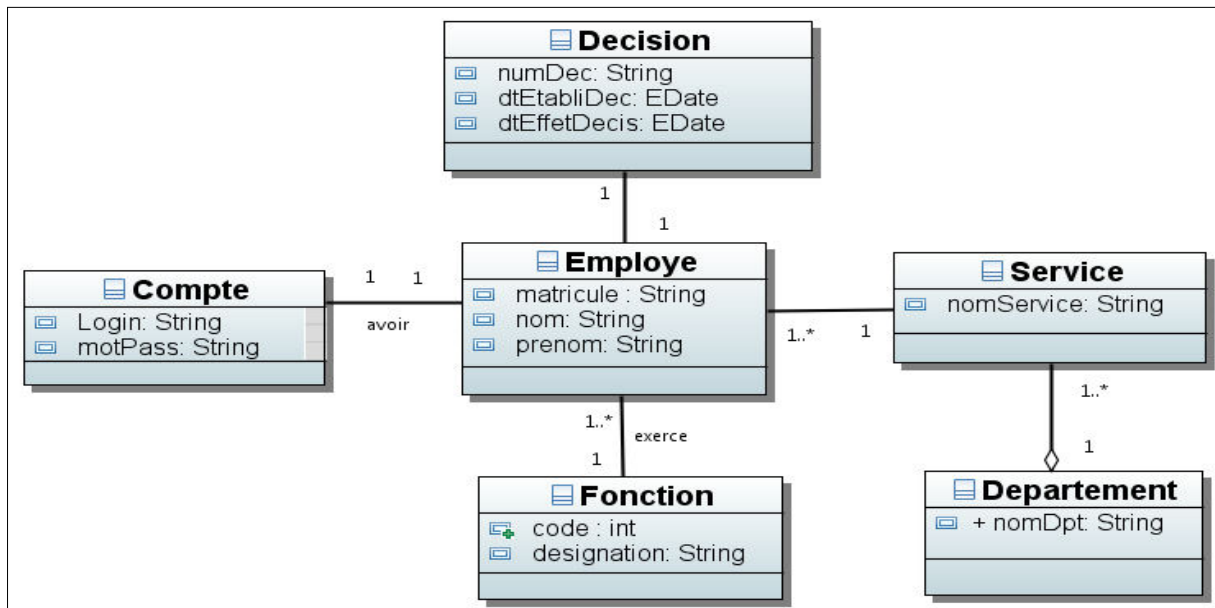


Figure n° 12: Modèle du domaine du sprint 1.

IV.4.2. Diagrammes de Séquence Système

La description textuelle est avantageuse mais présente des inconvénients. En effet, il est difficile de montrer les enchaînements se succèdent et à quel moment les acteurs secondaires sont sollicités. Il est donc recommandé de compléter la description textuelle par un ou plusieurs diagrammes dynamiques UML. Ce qui nous amène à faire appel aux diagrammes de séquence pour décrire les cas d'utilisation précédant.

IV.4.2.1. Diagramme de séquence système du cas « S'authentifier »

Le diagramme suivant a été référencé à partir de plusieurs autres diagrammes. Il représente l'enchaînement nominal du cas d'utilisation « s'authentifier ». Ce cas d'utilisation apparaît dans le cas où le système donne l'accès à l'utilisateur, dans un autre sens il apparaît dans le cas où l'identifiant et le mot de passe de l'utilisateur sont validés et celui-ci se connecte correctement à l'application. Ce cas d'utilisation est inclut dans quelques autres cas étudiés dans ce chapitre.

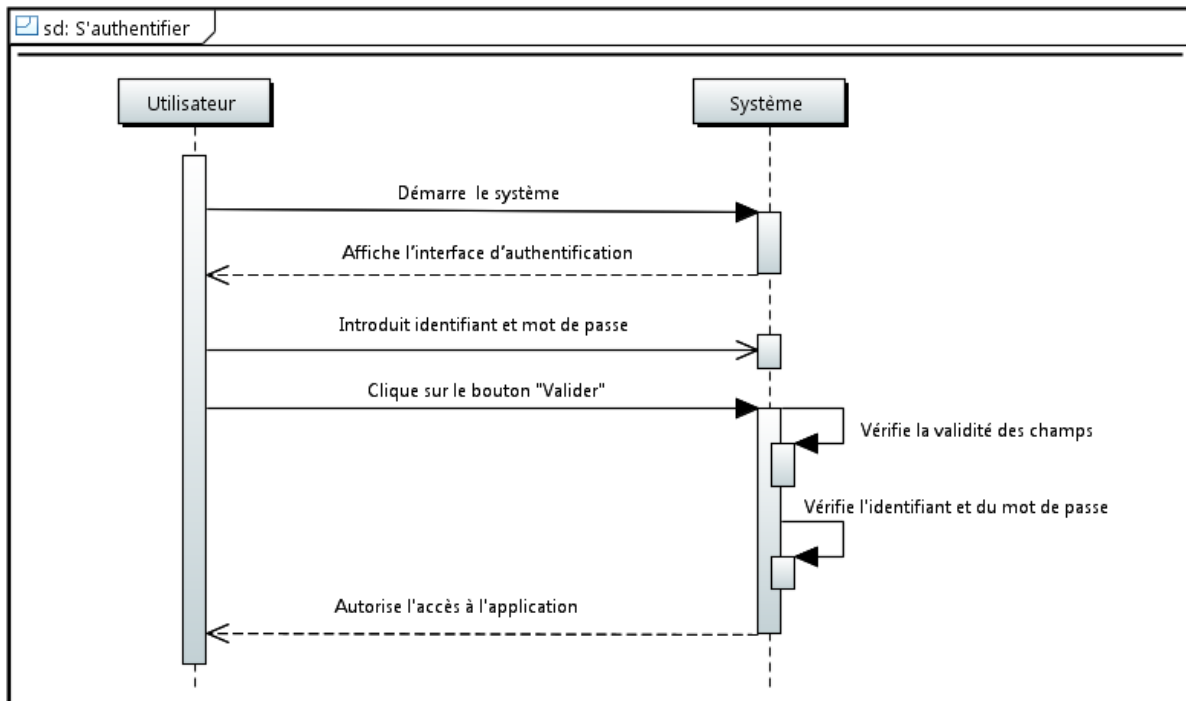


Figure n° 13 : Diagramme de séquence système du cas « S'authentifier ».

IV.4.2.2. Diagramme de séquence système du cas « Introduire les informations »

Le diagramme de séquence du fragment « Introduire les informations » référencé dans les autres diagrammes, représente le cas où l'utilisateur introduit les informations via un formulaire et le système vérifie qu'il n'y a pas d'erreurs et passe à l'étape suivante. Ce cas de figure est commun à plusieurs autres diagrammes c'est pourquoi nous l'avons représenté sous forme d'un fragment référencé dans les autres diagrammes.

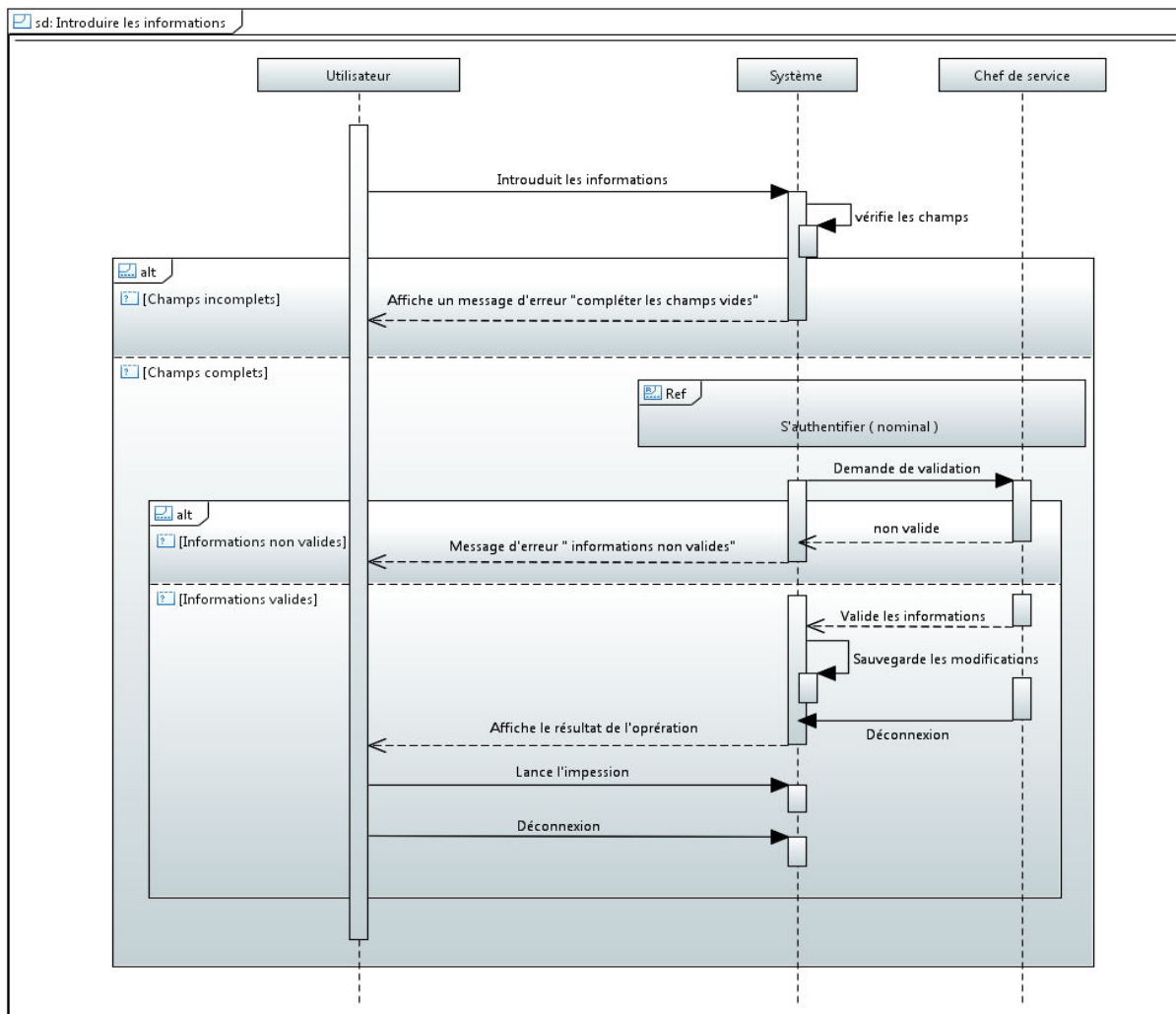


Figure n° 14: Diagramme de séquence système du cas « Introduire les informations ».

IV.4.2.3. Diagramme de séquence système du cas « Valider les documents »

Le chef de service a pour rôle de valider et signer les documents établis par les gestionnaires administratifs et les cadres de gestion administrative, de ce fait tout document doit être validé. La figure ci-après interprète le scénario qui se déroule entre le système et le chef de service lorsque ce dernier valide un document. Dans ce cas, nous avons pris en considération l'enchaînement nominal puisque c'est ce cas qui va être référencé dans les diagrammes de séquence.

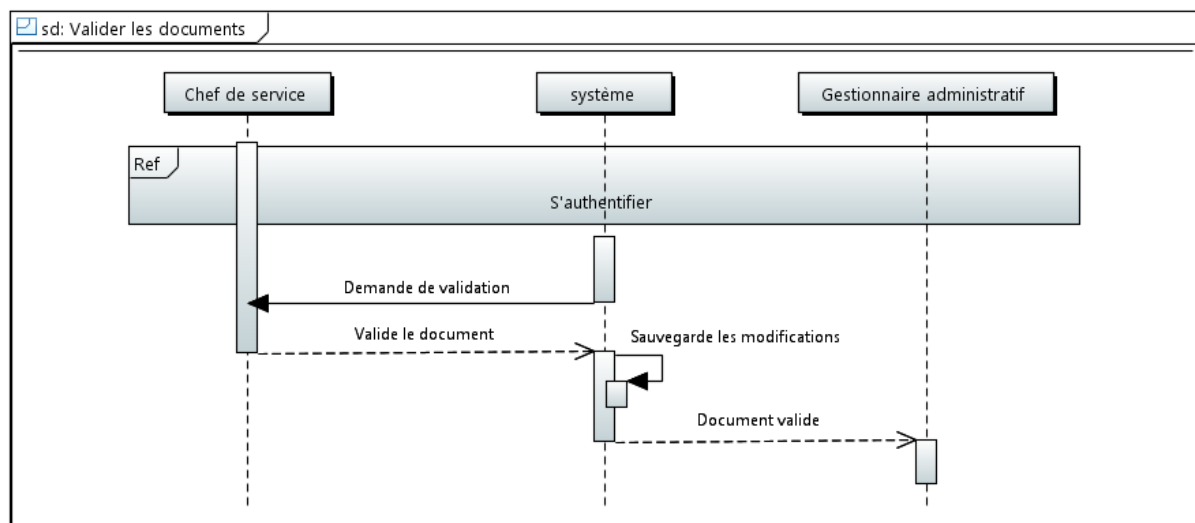


Figure n° 15: Diagramme de séquence système du cas « Valider les documents ».

IV.4.2.4. Diagramme de séquence système du cas « Établir une décision »

L'établissement d'une ou plusieurs décisions est l'une des tâches régulières que le cadre de gestion administrative effectue au cours de son travail, il suit tout un processus constitué de plusieurs étapes.

Le diagramme suivant (dans la page suivante) représente les différentes étapes de ce processus.

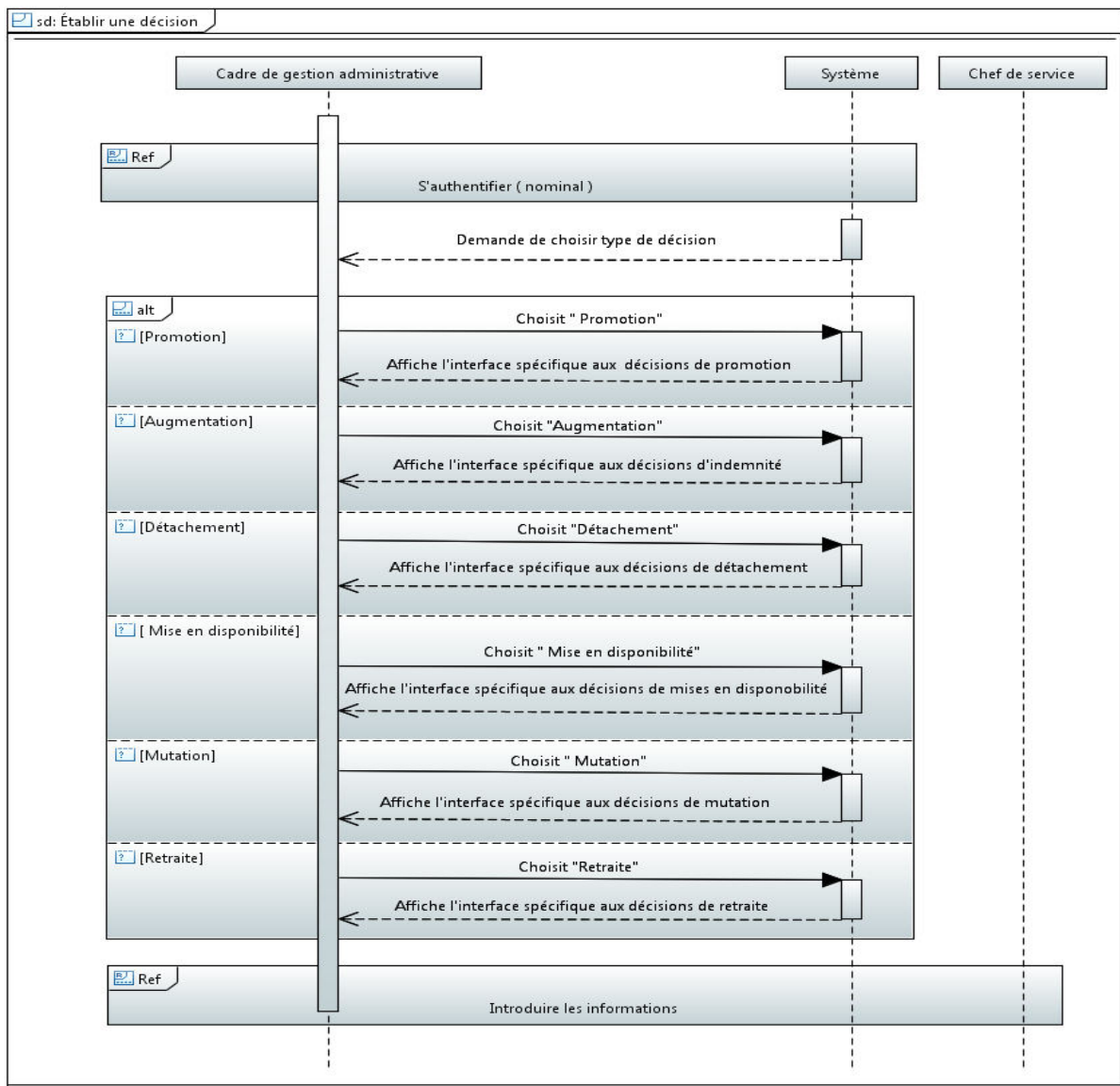


Figure n° 16: Diagramme de séquence système du cas « Établir une décision ».

Voici un exemple de décision qui est « la décision de mutation », la figure suivante représente le diagramme d'activité qui décrit explicitement ce cas.

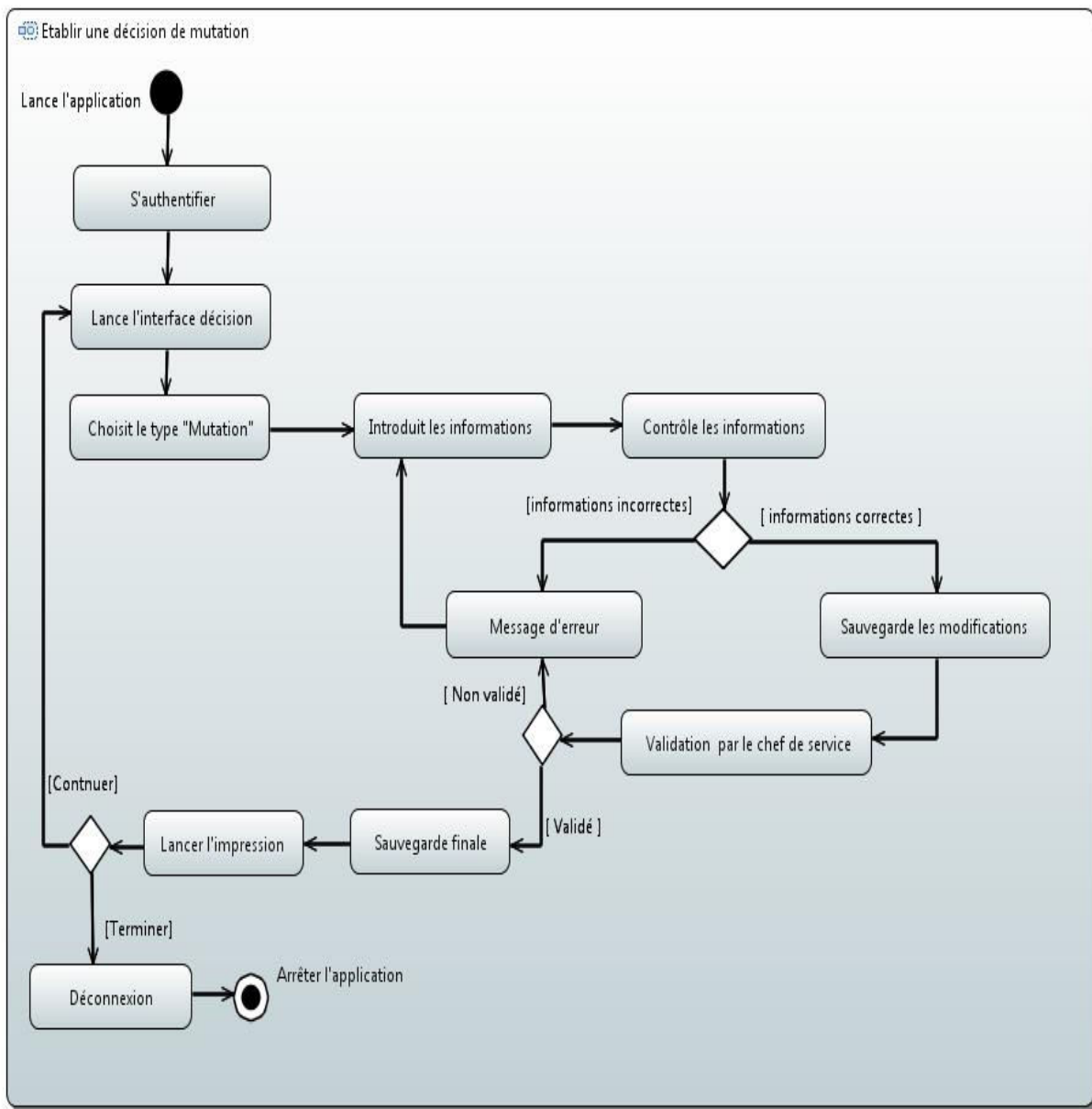


Figure n° 17: Diagramme d'activité « Établir une décision de mutation ».

IV.5. Conception

La conception est une étape qui consiste à définir l'architecture du système. Dans cette section, nous allons réaliser les diagrammes de séquence système qui correspondent à chaque cas d'utilisation, suivi du diagramme de classe du sprint 1.

I.5.1. Diagrammes de séquences détaillés

Les diagrammes ci-dessous est une représentation graphique détaillée des interactions selon leur ordre chronologique. Dans ces diagrammes nous avons noté: IU : interface utilisateur V : Vue, C : Contrôleur, M : Modèle et BD pour Base de données.

Tout d'abord le diagramme de séquence détaillé du cas d'utilisation « S'authentifier » puis celui de « Établir une décision ».

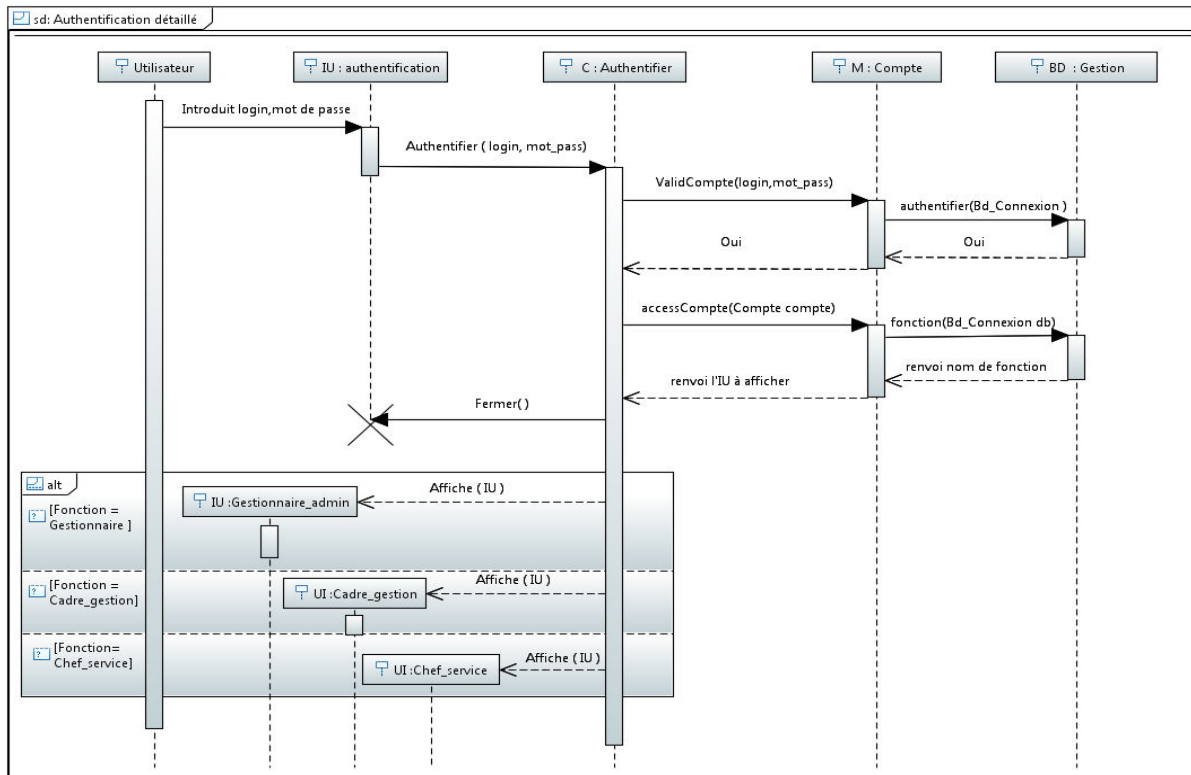


Figure n° 18: Diagramme de séquence détaillé du cas d'utilisation « S'authentifier ».

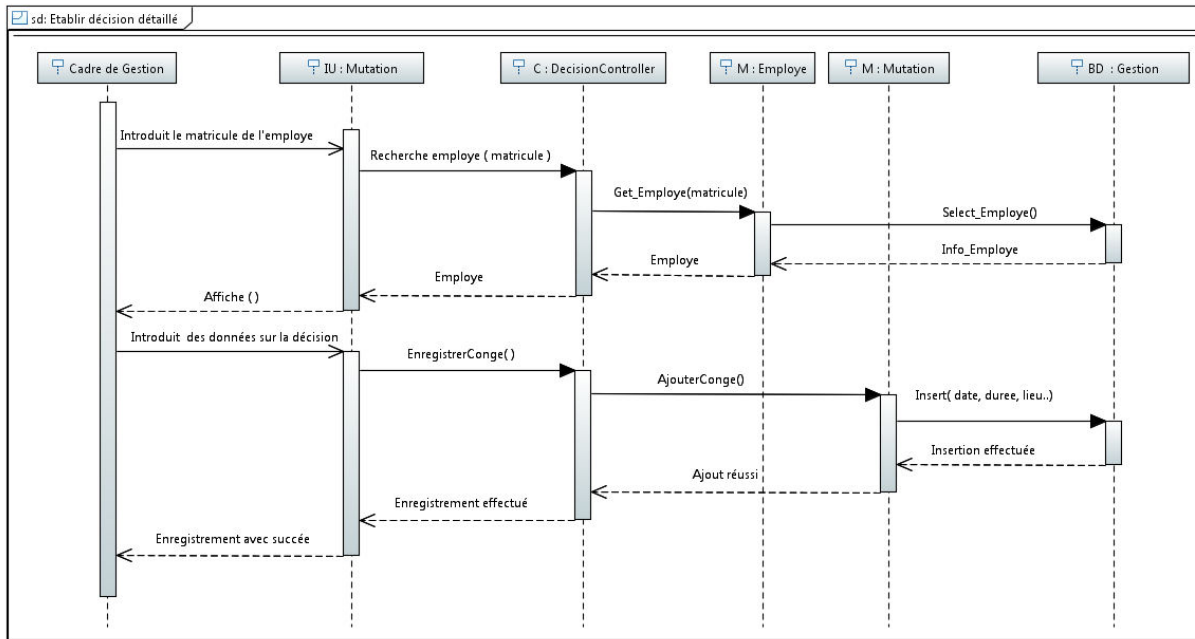


Figure n° 19: Diagramme de séquence détaillé du cas d'utilisation « Établir une décision ».

IV.4.4. Diagrammes de classes global du premier sprint

Voici les règles de gestion qui nous ont permis de mettre en œuvre notre diagramme de classes global :

- ✚ Un employé exerce une et une seule fonction à un moment, et un service donné;
- ✚ Une fonction peut être exercée par plusieurs employés dans des services différents;
- ✚ Un employé peut faire l'objet d'une décision;
- ✚ Une décision concerne un seul employé;
- ✚ Une décision est basée sur d'autres décisions;
- ✚ Une décision peut avoir plusieurs types (retraite, mutation, mise en disponibilité, détachement, augmentation, promotion);
- ✚ Certaines décisions peuvent avoir une demande;
- ✚ Une demande concerne une seule décision;
- ✚ Une décision contient des lignes;

- ✚ Des lignes constituent une décision;
- ✚ Un employé de service de gestion possède un compte;
- ✚ Un compte concerne un seul employé;
- ✚ Un employé est affecté à un seul service;
- ✚ Un service comprend plusieurs employés;
- ✚ Un service appartient à un seul département;
- ✚ Un service possède un seul chef de service ;
- ✚ Un département est composé de plusieurs services.
- ✚ Chaque employé est identifié par son matricule;
- ✚ Un service a au moins un employé et un seul chef de service.

La figure suivante représente le diagramme de classes global du premier sprint.

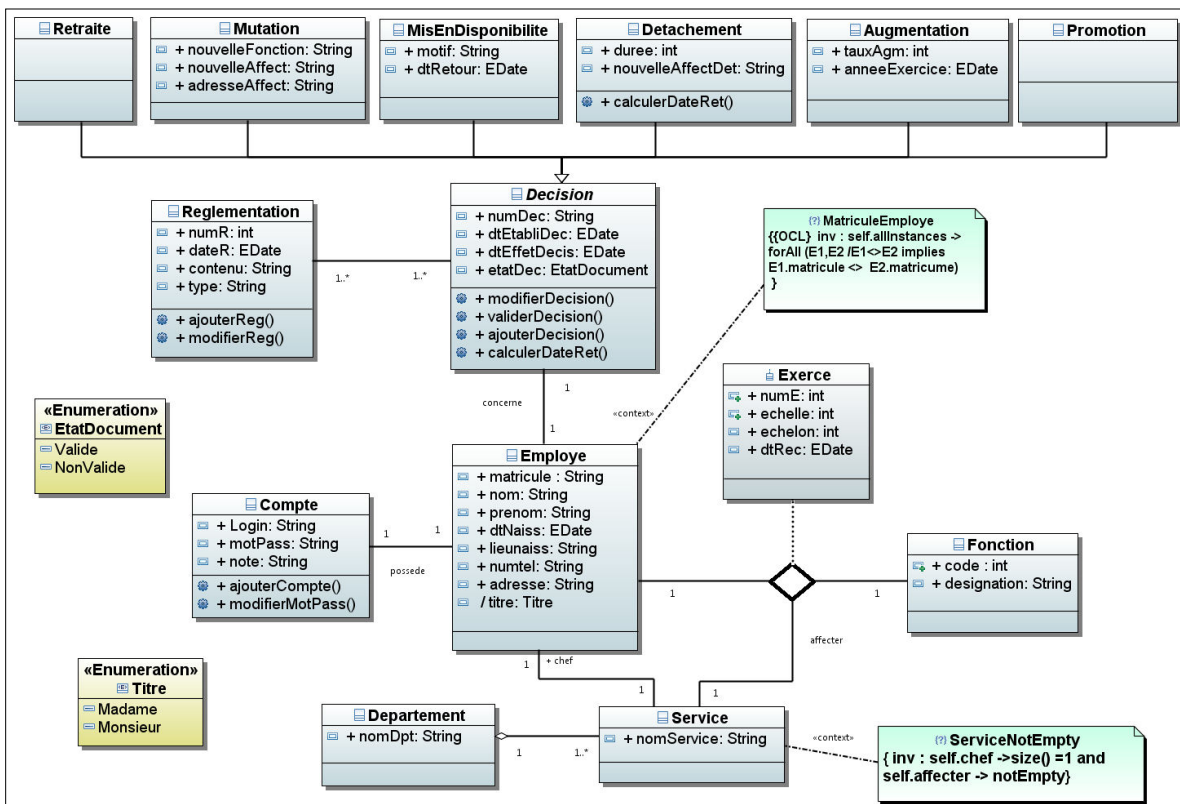


Figure n° 20 : Diagramme de classes de conception du sprint 1.

IV.6. Développement

Le développement consiste à implémenter les users stories analysées et conçues lors des deux étapes précédentes. Mais avant de passer à l'étape suivante, nous allons d'abord présenter la structure de la base de données pour ce premier sprint. Ce schéma de la BDD est obtenu en appliquant les règles de passage d'un diagramme de classes UML en un modèle relationnel-objet. Nous avons aussi utilisé Papyrus pour la génération partiel du code, cette fonctionnalité nous a permis d'avoir un code de classes Java à partir de classes UML.

IV.6.1. Schéma de la Base de Données

Afin d'implémenter la base de données, nous avons créé des stéréotypes propre à la BDD objet-relationnelle permettant de faire le lien entre la classe de conception et les tables de la BDD. La figure suivante représente le schéma des tables de la base de données du premier sprint.

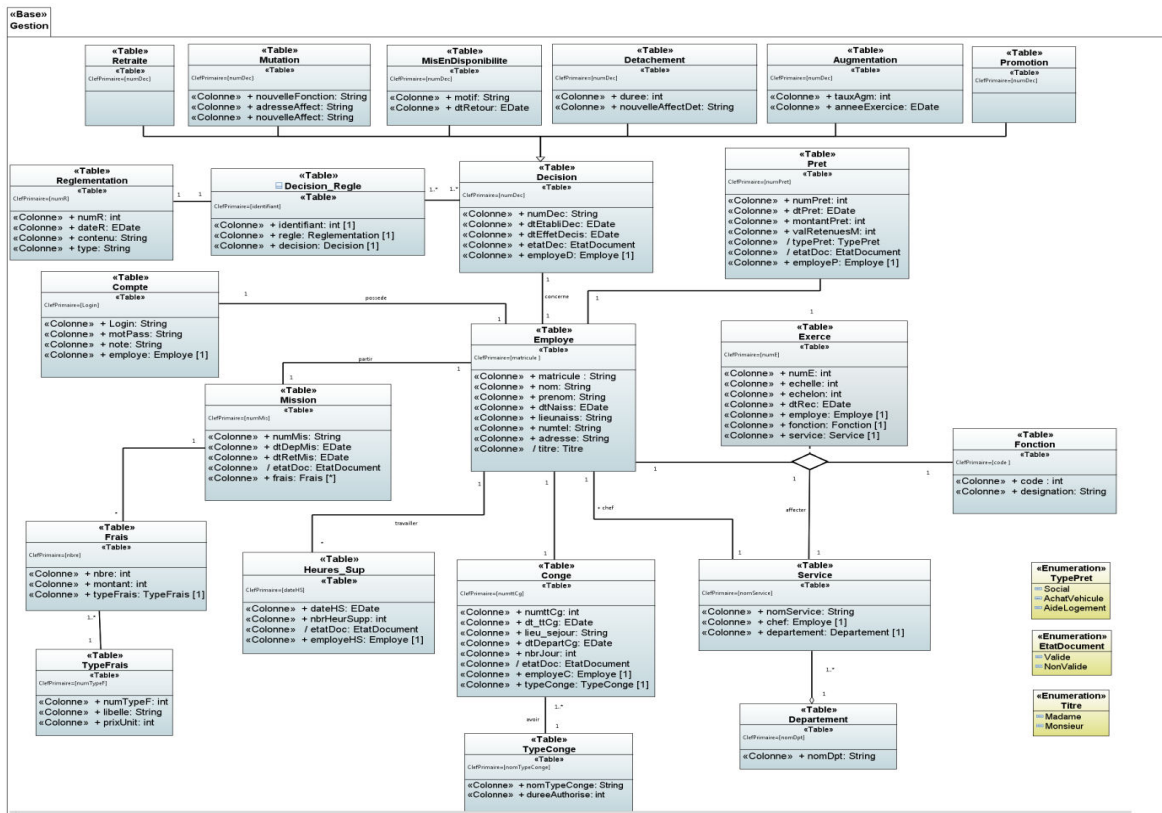


Figure n° 21: Schéma de la base de données du sprint 1.

IV.6. Test

Cette partie représente la dernière phase du cycle de développement d'un sprint. Elle permet de vérifier le bon fonctionnement de l'application et les résultats obtenus lors du développement et de garantir une version livrable de bonne qualité.

Les figures ci-dessous représentent des captures d'écrans de quelques interfaces de notre application développée afin de montrer que les résultats de ces tests sont corrects et fonctionnels.

IV.6.1. L'interface d'authentification

L'interface authentification est la première interface qui s'affiche dès que l'utilisateur lance l'application.

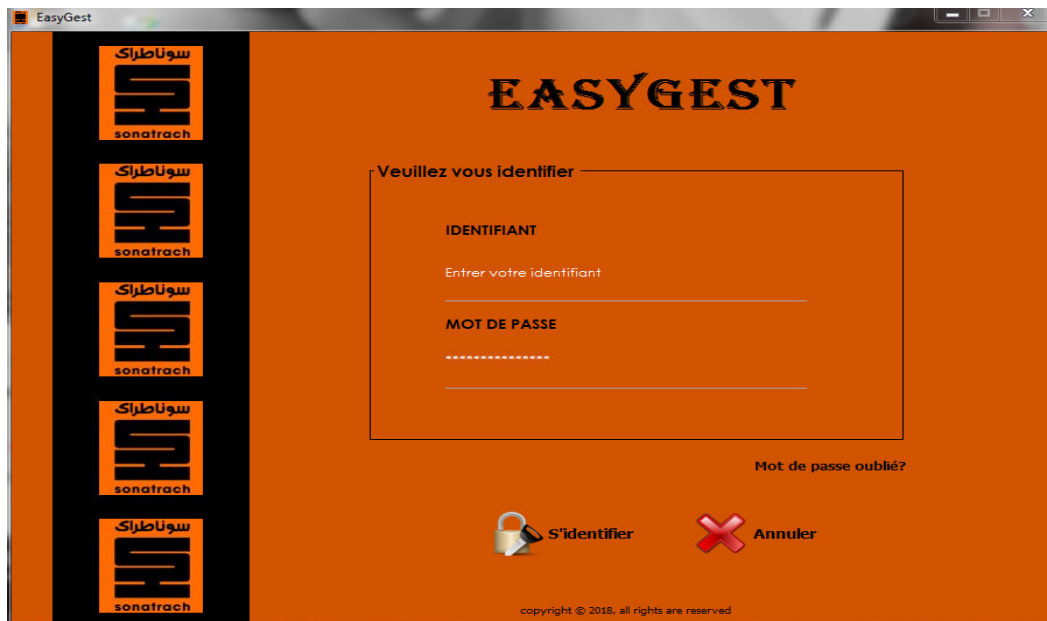


Figure n° 22 : Interface d'authentification.

IV.6.2. Interface de décision mutation

Cette interface est dédiée au cadre de gestion administrative, c'est le seul qui peut y accéder une fois qu'il s'authentifie. C'est l'une des fonctionnalités qui lui est réservées.

Figure n° 23 : Interface de décision de mutation.

IV.6.3. Interface Attestation de travail

Le gestionnaire administratif et le cadre de gestion ont tous les deux la possibilité d'établir une attestation de travail.

Figure n° 24 : Aperçu d'avant génération en PDF d'une attestation de travail.

IV.7. Conclusion

À ce niveau, nous avons pu réaliser et développer notre premier sprint. Nous disposons à présent de la première version de notre application qui est dédiée aux cadres de gestion. Elle va rendre facile et rapide la tâche d'établir toutes sortes de décisions. La partie suivante sera donc consacrée à la réalisation de notre deuxième sprint.

Chapitre V : Sprint 2

" Calcul des éléments de la paie"

V.1. Introduction

Au cours du chapitre précédant nous avons présenté le premier sprint. À la fin de ce dernier nous avons pu aboutir et obtenir la première version de notre application. Désormais nous allons passer au deuxième sprint durant lequel nous traiterons une autre fonctionnalité de l'application qui est « le calcul des éléments de la paie ». Ce sprint va suivre les quatre phases d'UP suivantes : la spécification fonctionnelle, la conception, le développement et les tests.

V.2. Backlog du sprint 2

Au début de ce sprint nous allons approfondir notre étude au près des gestionnaires administratifs. À l'issue de cette dernière, nous avons donc obtenu le tableau ci-dessous. Ce tableau regroupe les users stories concernant ce sprint dont chacune est répartie en tâches et affectée à un membre de l'équipe

Table n° 8 : Les Users Story du sprint 2.

Id U. S	User Story	Id Tâches	Tâches	Affectation	Durée
4.1	En tant que gestionnaire administratif, je souhaite établir un titre de congé	4.1. A	Réaliser les diagrammes de cas d'utilisation, séquence, de class de conception de la fonctionnalité «établir un titre de congé »	A. Lamia	8h
		4.1. B	Développer le cas «établir un titre de congé »	A.Narimane	24h

		4.1. C	Tester le cas «établir un titre de congé »	A.Lamia	2h
4.2	En tant que gestionnaire administratif, je souhaite établir un contrat de prêt	4.2. A	Réaliser les diagrammes de cas d'utilisation, séquence, de class de conception de la fonctionnalité «établir un contrat de prêt»	A.Narimane	8h
		4.2. B	Développer le cas «établir un contrat de prêt»	A.Lamia	4 j
		4.2. C	Tester le cas «établir un contrat de prêt»	A.Narimane	2h
4.3	En tant que gestionnaire administratif, je souhaite calculer les frais de mission	4.3. A	Réaliser les diagrammes de cas d'utilisation, séquence, de class de conception de la fonctionnalité «calculer les frais de mission»	A.Lamia	8h
		3.4. B	Développer le cas «calculer les frais de mission »	A.Narimane	24 h
		3.4. C	Tester le cas «calculer les frais de mission»	A.Lamia	3h
4.4	En tant que	4.4. A	Réaliser les diagrammes	A.Narimane	8h

	gestionnaire administratif, je souhaite calculer les heures supplémentaires		de cas d'utilisation, séquence, de class de conception de la fonctionnalité «calculer les heures supplémentaires»		
		4.4. B	Développer le cas «calculer les heures supplémentaires»	A.Lamia	5h
		4.4. C	Tester le cas «calculer les heures supplémentaires »	A.Narimane	3h

V.3. Spécifications fonctionnelles

Nous commençons ce sprint par une étape importante qui est la spécification fonctionnelle. Durant cette étape nous allons identifier les acteurs intervenant dans les cas d'utilisation de ce sprint, puis passer à la réalisation d'un diagramme de cas d'utilisation suivi de sa description textuelle. Ce qui va nous permettre à la fin d'enchaîner avec la conception.

V.3.1. Classification des cas d'utilisation par acteur

Après avoir détecté et identifié les acteurs et les cas d'utilisation on peut maintenant les classer de la manière suivante :

Table n° 9 : Classification des cas d'utilisation par acteur.

Gestionnaires administratif	Chef de service
<ul style="list-style-type: none"> ✓ Enregistrer les jours de tout type de congés ; ✓ Établir et imprimer plusieurs sortes de documents (attestation de travail, titre de congé, le contrat de 	<ul style="list-style-type: none"> ✓ Validation des modifications et des sauvegardes effectuées pas les gestionnaires administratifs.

prêt, ordre de paiement...etc)	
✓ Calculer les frais de mission après avoir calculer le montant par type de frais et le montant total.	
✓ Calculer le nombre d'heures supplémentaires.	

V.3.2. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation présenté dans la figure n° 28 représente les relations entre les cas d'utilisation cités dans la section précédente. Ces relations ont été déduites comme suit :

- ✓ Le calcul des éléments de la paie comprend le calcul des heures supplémentaires, le calcul des frais de mission, établir les titres de congé et des contrats de travail. Ce qui fait qu'il existe une relation d'extension entre le cas d'utilisation « calculer les éléments de la paie » et les autres cas d'utilisation.
- ✓ Il y'a plusieurs types de congé d'où l'existence de plusieurs cas d'utilisation qui héritent du cas d'utilisation « Établir un titre de congé ».
- ✓ Le calcul des frais de mission comprend le calcul du montant par frais et le montant total, d'où la relation d'inclusion entre le cas d'utilisation « Calculer les frais de mission » et les deux cas à savoir « Calculer le montant total des frais de mission » et « Calculer le montant de chaque frais ».
- ✓ Un employé a droit à un prêt dont le type peut être soit social, d'aide au logement ou bien d'achat véhicule donc il y'a une relation de généralisation entre le cas d'utilisation « établir un prêt » et tout les types de prêt
- ✓ L'établissement d'un prêt est finalisé par l'établissement d'un contrat de prêt donc le cas d'utilisation « Établir un contrat de prêt » inclut le cas d'utilisation « imprimer un contrat de prêt ».
- ✓ Le calcul des heures supplémentaires est finalisé par l'établissement d'un relevé des heures supplémentaires chaque mois.

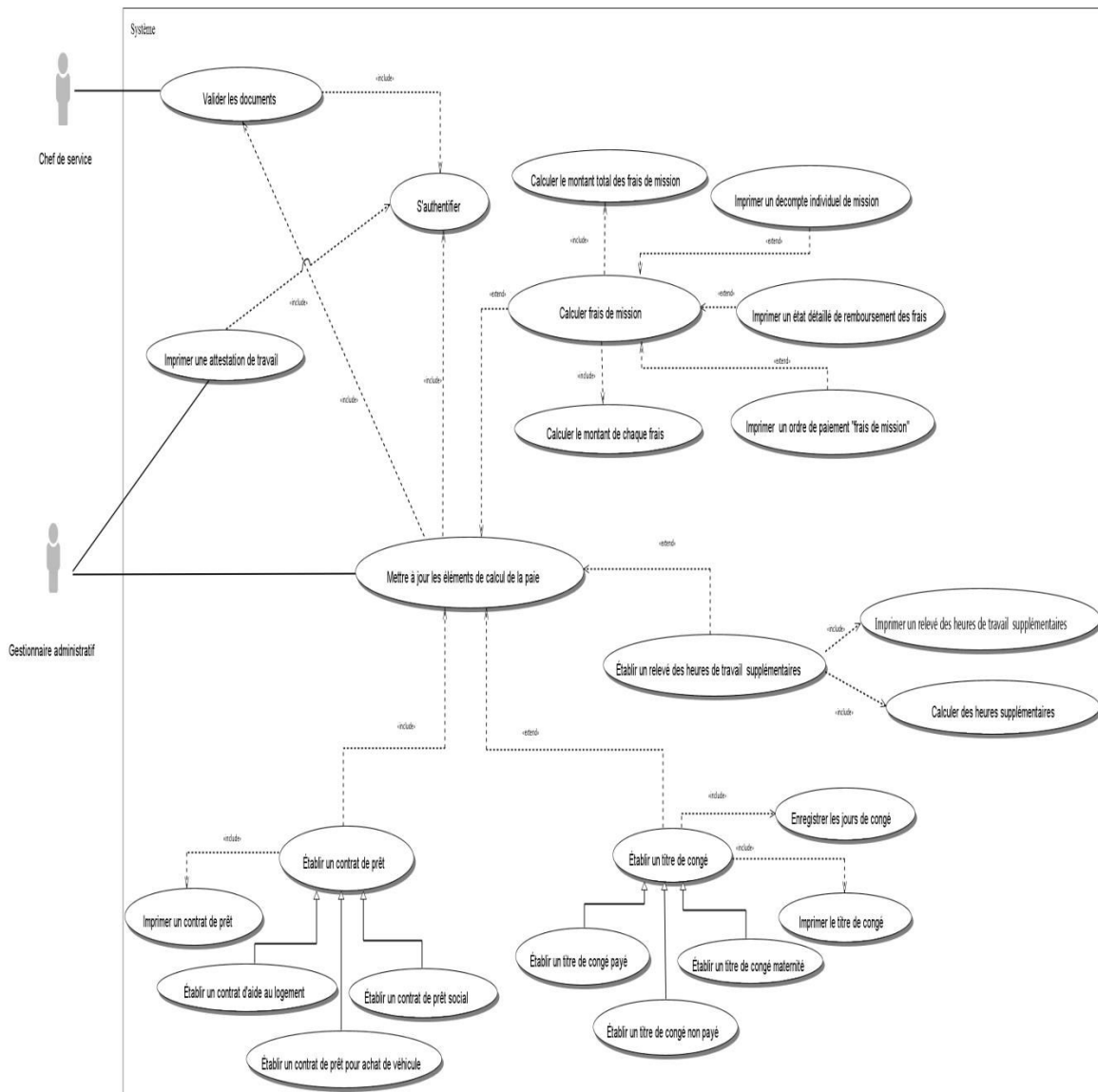


Figure n° 25 : Diagramme de cas d'utilisation « Calculer les éléments de la paie».

V.3.3. Diagramme d'activités

Le diagramme d'activités suivant est une représentation graphique qui permet d'illustrer d'une manière plus facile et de mieux consolider les différentes tâches menées et faites par le gestionnaire administratif dès le lancement de l'application.

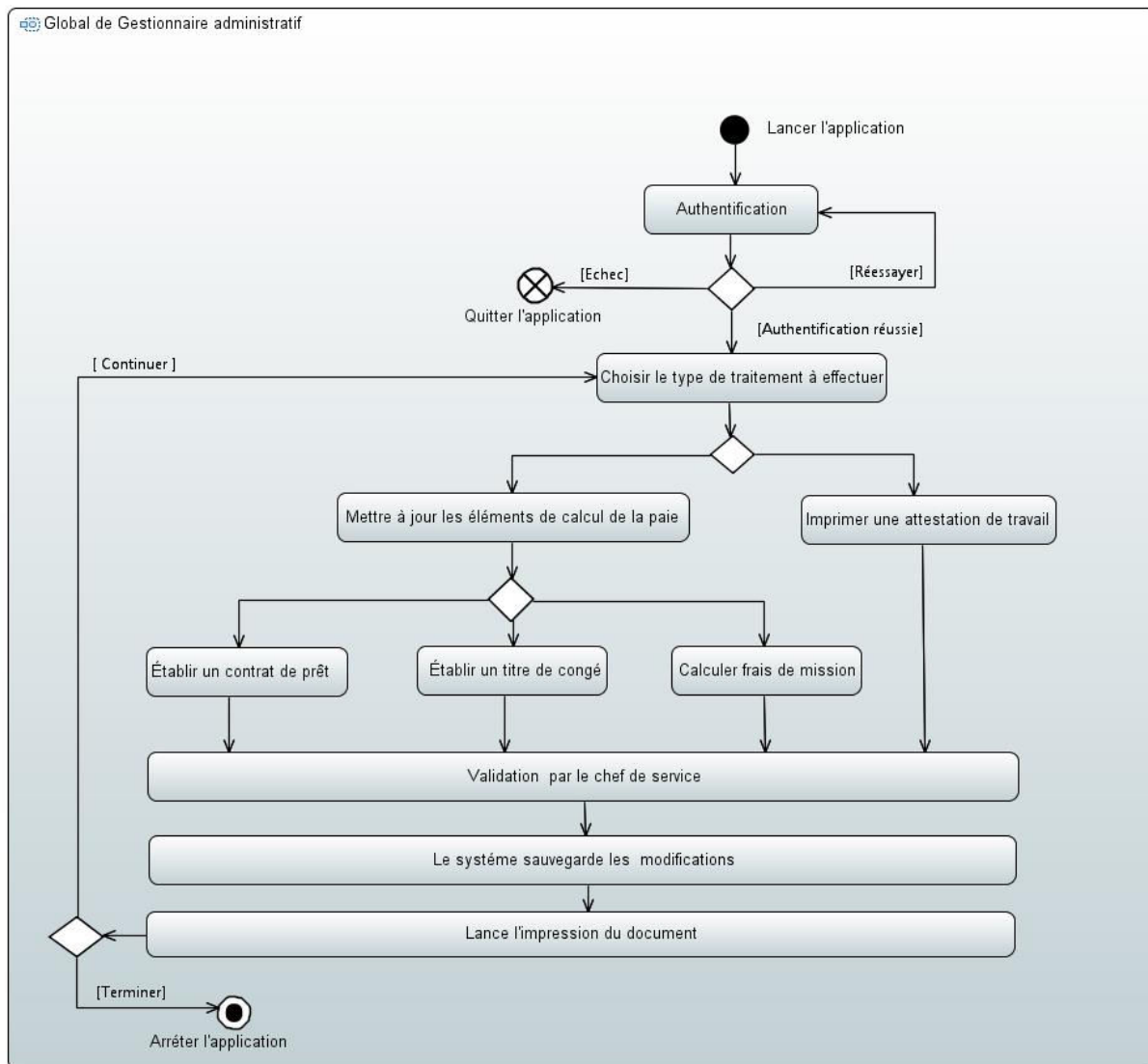


Figure n° 26: Vue d'ensemble des tâches effectuée par le gestionnaire administratif.

V.3.4. Description textuelle des cas d'utilisation

Les cas d'utilisation sont définis par une description textuelle, décrivant les objectifs et interactions entre le système et ses acteurs. Voici en ce qui suit la description textuelle des cas d'utilisation identifiés.

V.3.4.1. Cas d'utilisation « Mettre à jour les éléments du calcul de la paie »

Table n° 10: Description textuelle du cas d'utilisation « Mettre à jour les éléments du calcul de la paie ».

Cas d'utilisation	Mettre à jour les éléments du calcul de la paie.
But et objectif	<ul style="list-style-type: none"> • Calculer les heures supplémentaires ; • Établir les titres de congés ; • Calculer les frais de missions ; • Gérer les prêts en établissant des contrats de prêt.
Acteur	Gestionnaire administratif.
Pré-condition	L'utilisateur doit s'authentifier.
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur choisit l'interface adéquate ; 2. Le système renvoi l'interface demandée ; 3. L'utilisateur sélectionne le type du calcul à faire; 4. Le système affiche l'interface demandée qui se charge du calcul choisit pas l'utilisateur; 5. L'utilisateur introduit des informations liées au calcul et à l'employé concerné par celle-ci et valide le tout. [A1] [A2] 6. Le système fait les calculs et les modifications demandés, sauvegarde et affiche le résultat. 7. L'utilisateur a la possibilité d'imprimer le résultat une fois que le chef de service valide les données.
Enchaînement Alternatif [A]	<p>A1 : champ d'information incomplet</p> <p>6. Le système affiche un message d'erreur, le scénario reprend au point 3 du scénario nominal.</p> <p>A2 : informations introduites sont erronées avant validation</p> <p>L'utilisateur a la possibilité de rectifier les erreurs avant de valider ses choix, le scénario reprend au point 4 du scénario nominal.</p>
Post-condition	Impression des documents une fois validés par le chef de service.
Besoin d'IHM	L'interface de validation

	<p>Les interfaces du calcul des éléments de la paie à savoir l'interface de</p> <ul style="list-style-type: none"> • Calcule des heures supplémentaires ; • Établissement des titres de congés ; • Calcule des frais de missions ; • Gestion des prêts en établissant des contrats de prêt.
--	---

V.4. Analyse

Nous allons passer maintenant à l'étape d'analyse qui nous permettra d'étudier et d'écrire les besoins du client sous une forme qui nous faciliterait et permettrait de mieux implémenter notre application. Durant cette phase nous allons commencer par réaliser le modèle du domaine et quelques diagrammes de séquence système

V.4.1. Modèle du domaine

La figure ci-dessous représente la version du diagramme de classes du sprint 2 de notre application. C'est un diagramme de classes d'analyse. Il modélise de manière globale les différentes entités manipulées par les utilisateurs. Cette version sera encore plus détaillée ultérieurement.

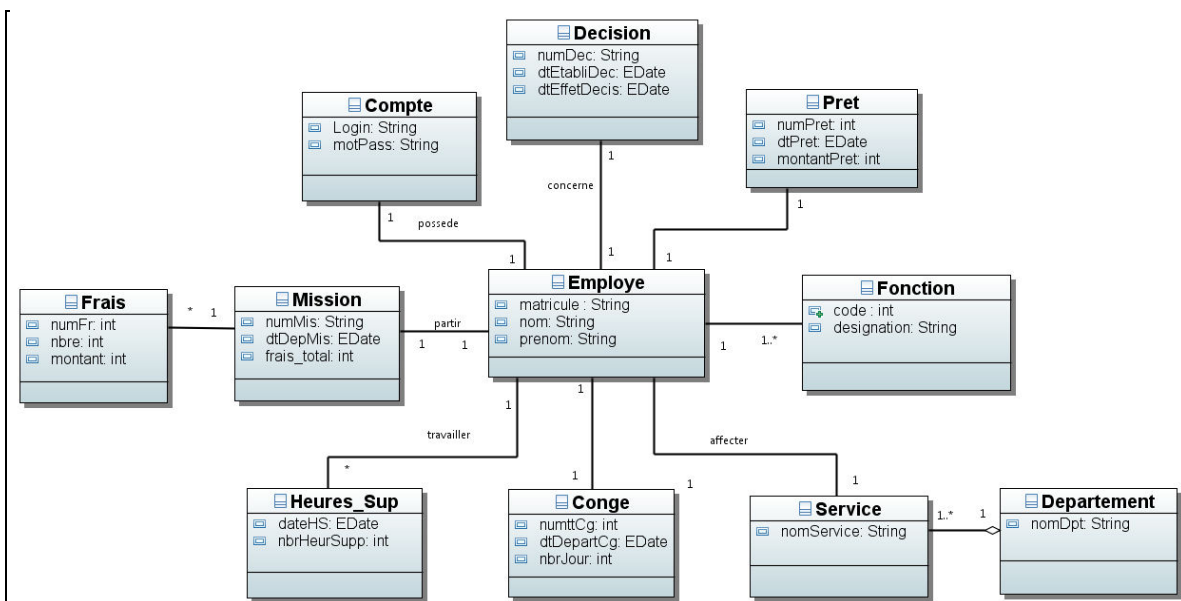


Figure n° 27 : Modèle du domaine du sprint 2.

V.4.2. Diagrammes de séquence système

Cette partie est consacrée à la présentation des diagrammes de séquence système du deuxième sprint. Ce sprint est basé sur le calcul des éléments de la paie.

V.4.2.1. Diagramme de séquence système du cas « Calculer les frais de mission »

Dans le cas où l'utilisateur, qui est dans ce cas le gestionnaire administratif, souhaite calculer les frais de mission, celui-ci demande l'interface réservée à ce traitement. Une fois l'interface affichée l'utilisateur saisit le matricule de l'employé concerné puis renseigne les éléments de la mission tout en choisissant les types de frais. L'utilisateur pourra ensuite imprimer une fois que le chef de service valide le document ainsi que ses données.

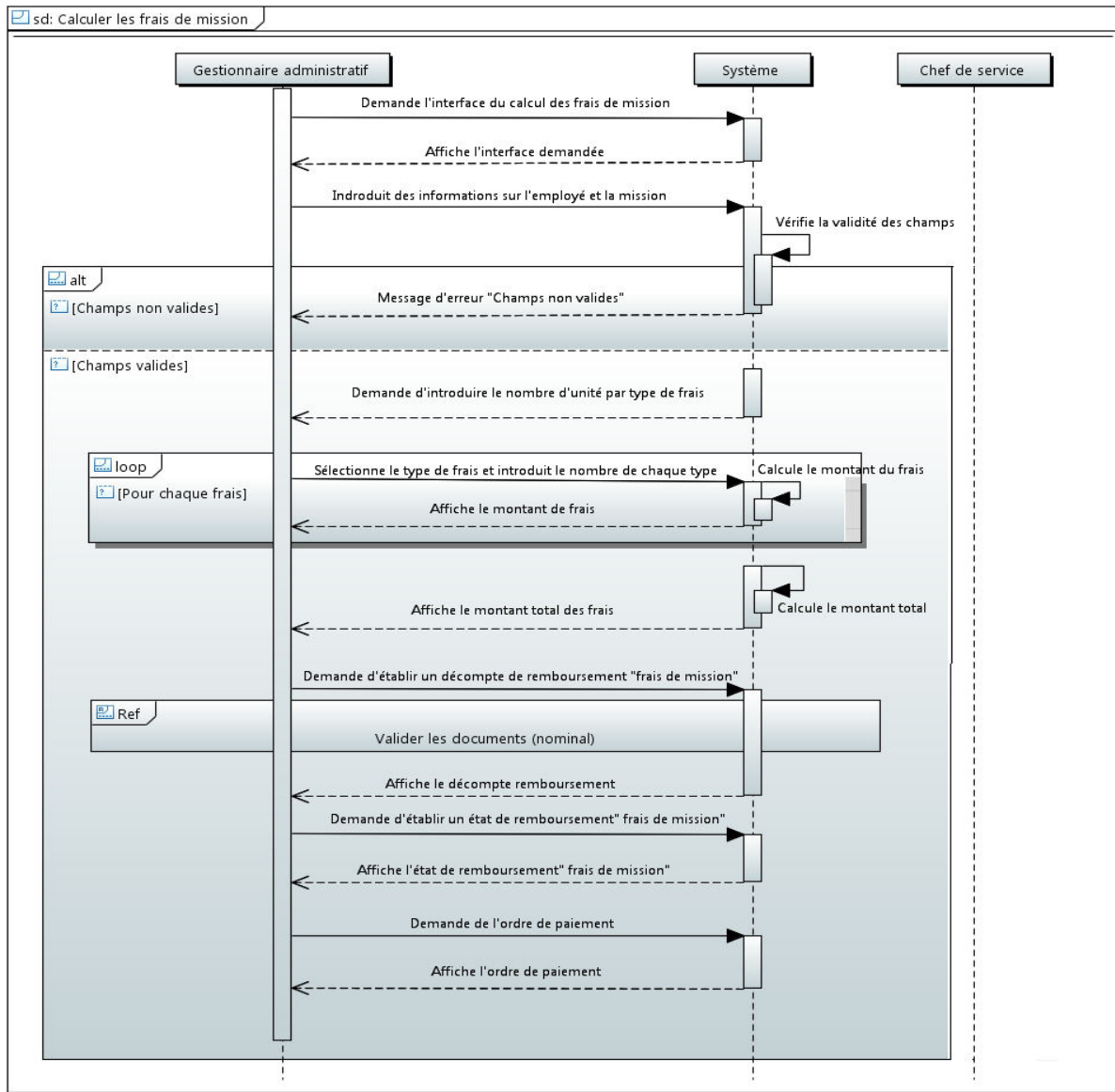


Figure n° 28 : Diagramme de séquence système du cas « Calculer les frais de mission ».

V.4.2.2. Diagramme de séquence système du cas « Établir un titre de congé »

Le gestionnaire administratif a la possibilité d'établir un titre de congé selon la nature de ce congé. Il doit choisir un type parmi les types de congé, ensuite il introduit les informations concernant ce congé tel que la date de départ, le nombre de jours, le lieu de séjour, etc, puis il valide le tout. Quant au système de son côté vérifie la validité des champs

et des données saisies et sauvegarde après validation des informations par le chef de service. Le diagramme de séquence suivant résume toutes les étapes du scénario décrit.

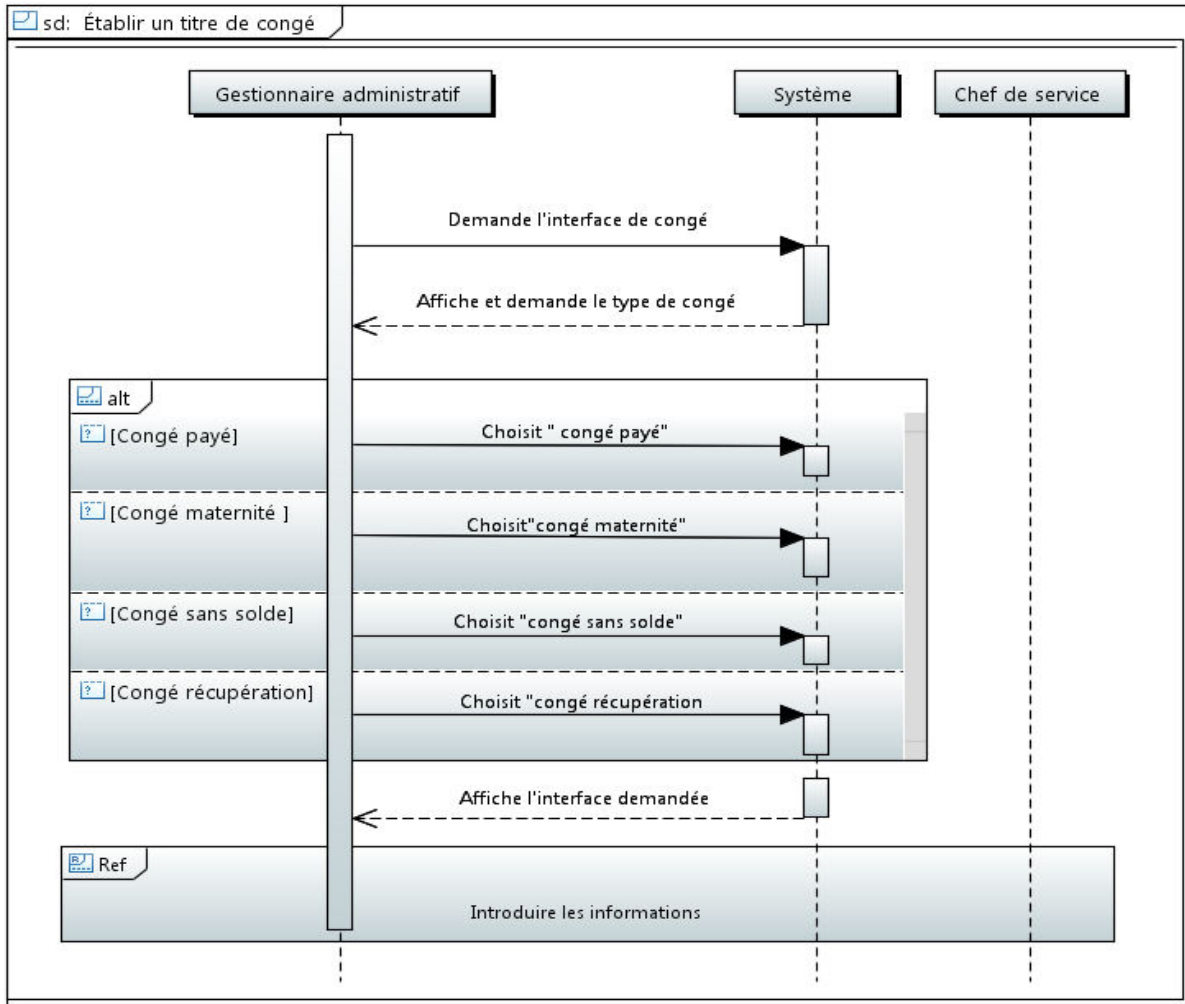


Figure n° 29: Diagramme de séquence système du cas « Établir un titre de congé ».

V.4.2.3. Diagramme de séquence système du cas « Établir un contrat de prêt »

Le diagramme de séquence suivant exprime de manière plus simple le cas d'utilisation « Établir un contrat de prêt ». Le scénario exprimé dans cette représentation graphique se déroule lorsque le gestionnaire administratif souhaite établir un contrat de prêt.

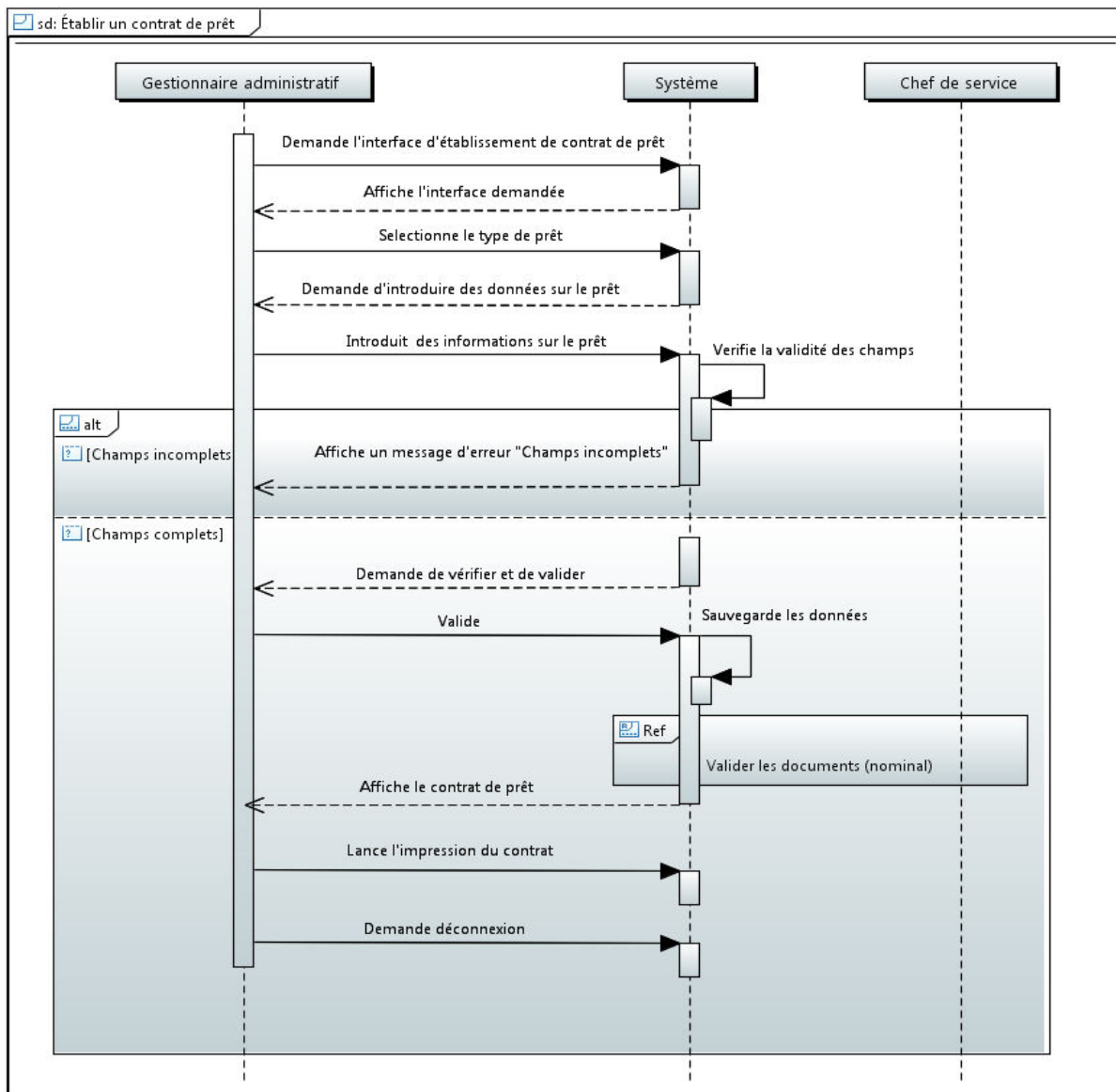


Figure n° 30 : Diagramme de séquence système du cas « Établir un contrat de prêt ».

V.4.2.4. Diagramme de séquence système du cas « Calculer les heures supplémentaires »

La figure suivante illustre le diagramme de séquence du deuxième sprint qui est dans notre cas le dernier sprint. Ce diagramme permet de décrire en détails et d'une manière graphique simple les différents scénarios du cas d'utilisation « Calculer les heures supplémentaires » fait par le gestionnaire administratif tout en respectant la chronologie des opérations.

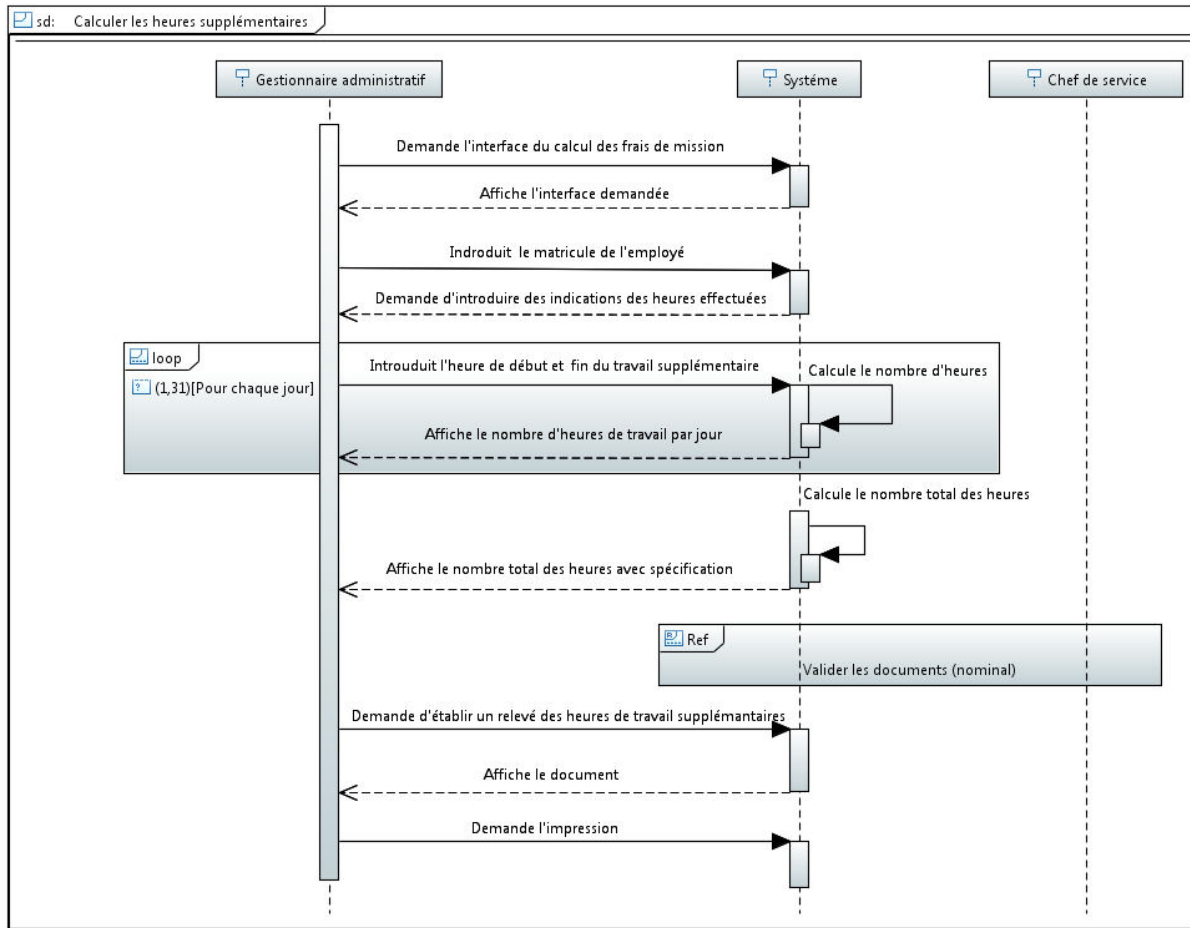


Figure n° 31: Diagramme de séquence système du cas « Calculer les heures supplémentaires ».

V.4.2.5. Diagramme de séquence système du cas « calcul des éléments de la paie »

L'utilisateur (plus exactement le gestionnaire administratif) est chargé du calcul de tout ce qui est en relation avec le calcul de la paie à savoir : le calcul des heures supplémentaires, le calcul des frais de mission mais aussi il est chargé d'établir des titres de congés et des contrats de prêts. Ces cas d'utilisations sont présentés dans le diagramme de séquence suivant, celui-ci fait référence aux autres diagrammes qui le suivent. La représentation

graphique est faite ainsi dans l'objectif de mieux les décrire et les expliquer.

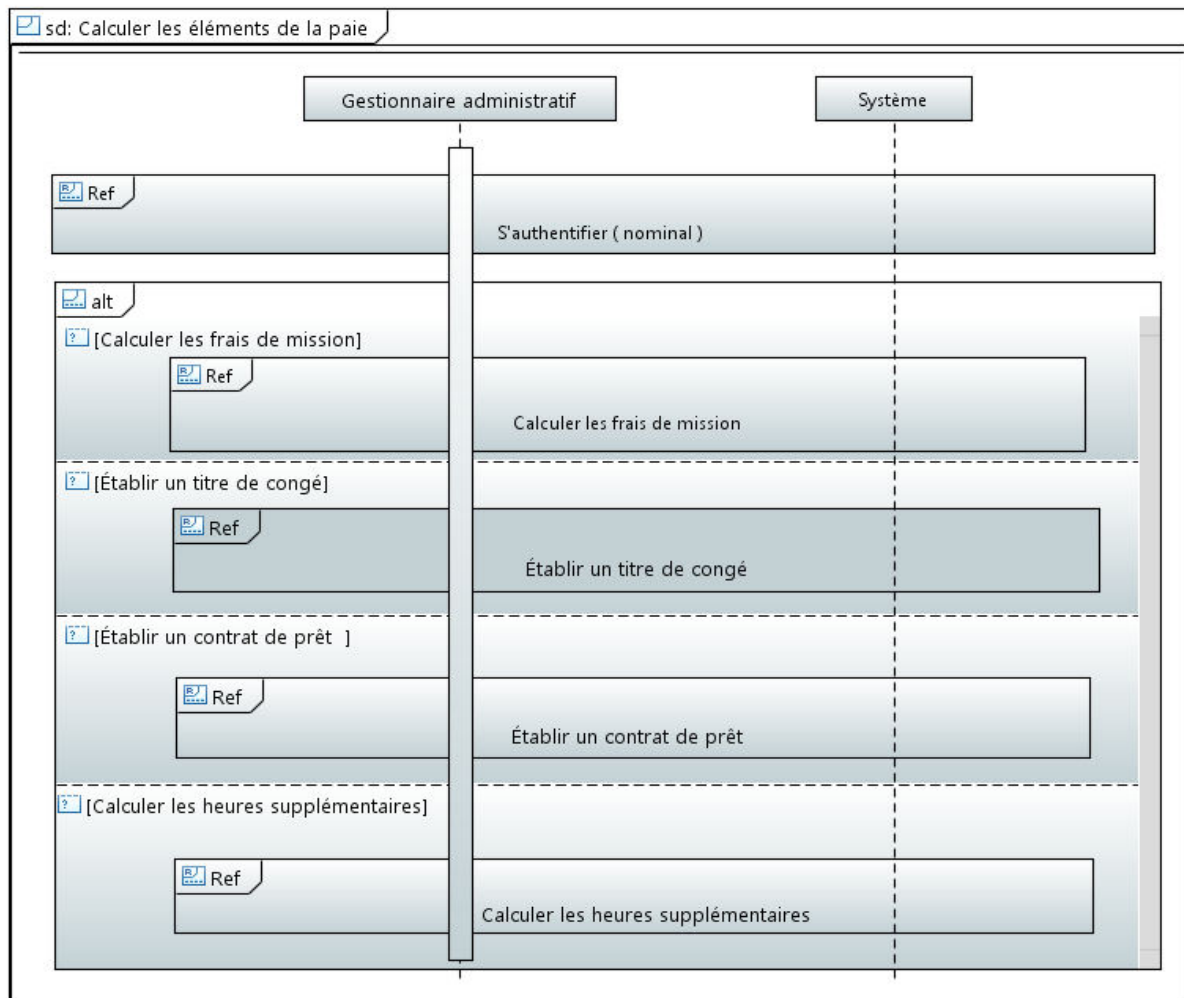


Figure n° 32: Diagramme de séquence système du cas « Calcul des éléments de la paie ».

V.5. Conception

Une fois les besoins du futur système sont recueillis et analysés, nous passons désormais à la phase de conception. Durant cette phase nous allons réaliser quelques diagrammes de séquence détaillés, et nous terminerons par un diagramme de classe de conception. Cette phase nous permettra de définir l'architecture et le comportement de l'application

V.5.1. Diagrammes de séquences détaillés

Le diagramme de séquence détaillé suivant concerne le cas d'utilisation d'établir un titre de congé.

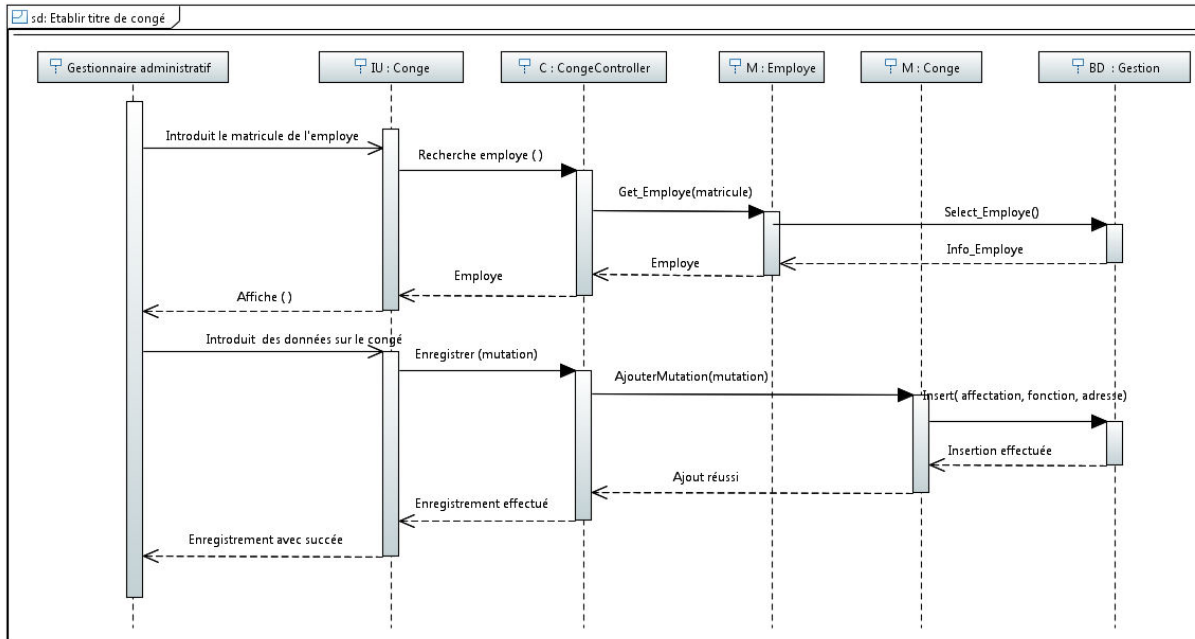





Figure n° 33: Diagramme de séquence système du cas « Calcul des éléments de la paie ».

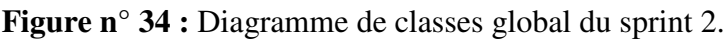
V.5.2. Diagramme de classes de conception

Le diagramme de classes de conception est considéré comme le document le plus important et le plus obligatoire lors de la modélisation orientée objet. Il représente la vue de conception statique d'un système. Ce diagramme a été réalisé en prenant en considérations les règles de gestions suivantes (en plus de celle citée dans le premier sprint) :

-  L'employé part en congé;
-  Un titre de congé concerne un seul employé;
-  Un congé a un seul type ;

- ✚ La durée du congé ne doit pas dépasser le nombre de jours autorisé pour chaque type ;
- ✚ Il peut y avoir plusieurs titres de congé d'un même type ;
- ✚ L'employé bénéficie d'un prêt ;
- ✚ Le contrat de prêt concerne un seul employé;
- ✚ L'employé travaille des heures supplémentaires;
- ✚ Le nombre d'heures total ne doit pas dépasser 12 heures par jours;
- ✚ L'employé part en mission;
- ✚ Un ordre de mission concerne un seul employé;
- ✚ Une mission peut avoir plusieurs frais;
- ✚ Un frais a un type de frais ;

Voici donc le diagramme de classes de conception de notre système étudié dans la page suivante:



À présent nous allons passer à la phase développement ou codage durant laquelle nous mettons en œuvre les fonctionnalités analysées et conçues précédemment, tout en prenant en considération la première version développée au premier sprint. Nous allons tout d'abord présenter le schéma de la base de données suivi du diagramme de composants.

V.6.1. Schéma de la Base de Données

Pour implémenter notre base de données nous avons utilisées des stéréotypes qui nous permettront d'ajouter de la sémantique à la modélisation des classes et de définir des types de classe, afin de regrouper conceptuellement un ensemble de classes. Voici la figure qui décrit le schéma des tables de la base de données de notre deuxième sprint.

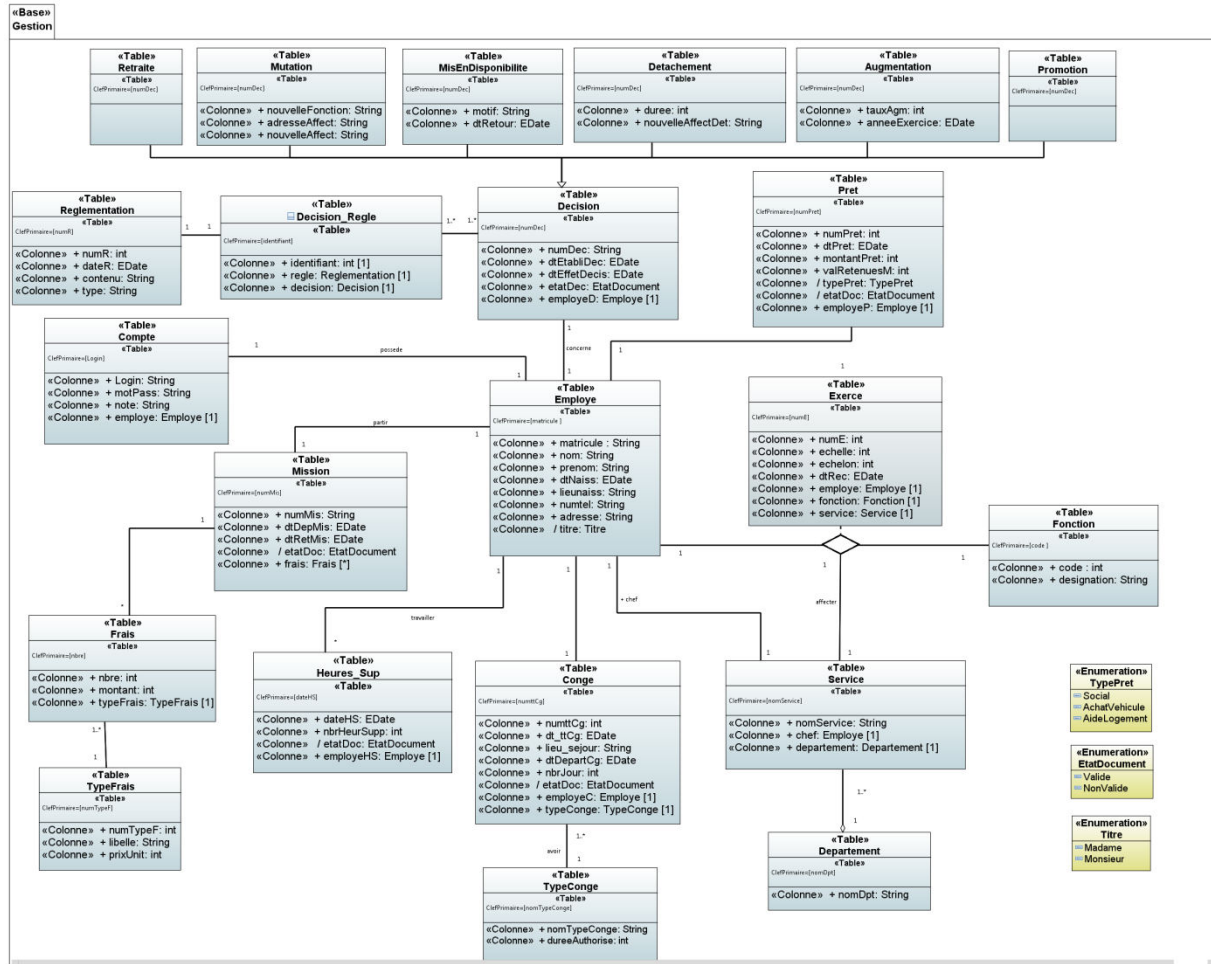


Figure n° 35 : Schéma de la BDD du sprint 2.

V.7. Test

La phase de test permet de s'assurer que la réalisation de l'application correspond aux besoins et aux attentes de l'utilisateur. Au cours de la phase de développement nous avons effectué des tests afin de mettre les différentes fonctionnalités de notre application.

Voici ci-après quelques interfaces de notre application qui illustrent ce qui se présente à cette phase de test. Ces interfaces nous permettent de faciliter le dialogue IHM ainsi que d'améliorer les performances de l'application.

V.7.1. Interface du gestionnaire administratif

Voici l'interface qui s'affiche une fois que le gestionnaire administratif accède à l'application et s'authentifie. Plusieurs fonctionnalités lui sont affichées et qui lui sont autorisées.



Figure n° 36 : Interface du gestionnaire administratif.

V.7.2. Interface titre de congé

L'interface suivante est l'interface dédiée au gestionnaire administratif, elle lui permet d'établir un titre de congé.

EasyGest

سوناطراكي sonatrach

EASYGEST
TITRE DE CONGÉ

سوناطراكي sonatrach

Accueil

Décisions

- Frais mission
- Congés
- Prêts

Attestation

- Attestation de travail

Heures supplémentaires

- Heures supplémentaires

Annuler

Enregistrer

Quitter

Veillez saisir le matricule OU le nom et prénom de l'employé

Matricule : CH0001

Annuler OK

Ou

Nom :

Prénom :

Annuler OK

Mme / Mr : Mr

Nom : Kaci

Prénom : Nadir

Fonction : Chef de service

Département : administration et social

Exercice : 1999-02-12

Nombre de jour * :

Lieu de séjour * :

Date départ * :

Date de retour * :

Annuler

copyright © 2018, all rights are reserved 22/06/2018 10:50:40 PM

Figure n° 37 : Interface de titre de congé.

V.7.2. Interface des heures supplémentaires

L'interface suivante est celle qui permet au gestionnaire administratif de calculer les heures supplémentaires de l'employé.

EASYGEST
HEURES SUPPLÉMENTAIRES

سونا تراش sonatrach

Accueil

Décisions

- Frais mission
- Congés
- Prêts

Attestation

- Attestation de travail

Heures supplémentaires

- Heures supplémentaires

Annuler

Enregistrer

Convertir en PDF

Quitter

Veuillez saisir le matricule OU le nom et prénom de l'employé

Matricule * : CH0001

Annuler OK

Mme / Mr : Mr

Nom : Kaci

Prénom : Nadir

Fonction : Chef de département

Jours ouvrables * : 05/06/2018

De * : 19

à * : 22

Nombres d'heures * : 3

Ajouter

Modifier

Supprimer

Jours ouvrables	de	jusqu'à	Nombre d'heures
01/06/2018	9	16	7
02/06/2018	12	20	8
05/06/2018	19	22	3

Total des heures 18

copyright © 2018, all rights are reserved 2018/06/24 04:14:24 PM

Figure n°38 : Interface des heures supplémentaires.

V.8. Conclusion

À la fin de ce chapitre, nous avons réussi à produire notre deuxième et dernier sprint. Nous disposons ainsi d'une version finale de notre application et d'un produit livrable sur laquelle des tests de validation ont été concluants.

L'application répond donc aux besoins du client. Elle permet une bonne gestion d'une manière rapide et facile et la génération automatique des documents en format PDF pour pouvoir les signer directement une fois imprimés.



Conclusion générale

Dans le cadre de notre projet de fin de cycle master, nous avons conçu et développer une application de gestion administrative. Cette application est dédiée aux employés du service gestion du département administration et social (ASL) de la région de transport centre (RTC) de la SONATRACH. C'est là où nous avons effectué notre stage pratique.

Pour commencer nous avons fait une étude détaillée du contexte général du service gestion et ceci en allant tous les jours pendant notre période de stage au siège de la RTC. Nous nous en sommes chargés d'établir un questionnaire destiné aux employés de ce service, afin de récolter le maximum d'informations utiles qui vont nous aider à mieux concevoir notre application, ces employés ont été très collaboratifs et coopératifs. Ce questionnaire comprenait plusieurs questions dont : le mode de travail des employés du service gestion et les tâches effectuées par chaque employé, la politique adoptée vis-à-vis des autres services et aussi les données des documents qui circulent dans ce service. Cette étude nous a permis de détecter les difficultés et les problèmes que rencontrent quotidiennement les employés du service de gestion.

Ensuite nous sommes passés au choix de la méthodologie de développement à adopter pour assurer une gestion adéquate et idéale de notre projet. Notre choix s'est porté sur la combinaison de SCRUM avec UP.

Par la suite nous avons mis en pratique les concepts des méthodes adoptées, en commençant par une planification de notre travail. Nous avons énuméré toutes les tâches à réaliser, avec l'attribution des degrés d'importance, ce qui nous a permis de découper les fonctionnalités en deux sprints. Durant chaque sprint nous avons appliqué le même enchainement en commençant par la phase spécification des besoins, passant par la conception, ensuite le développement, couronnant le tout par le test de chaque itération.

Pour conclure, nous avons pu réaliser une application de gestion administrative qui sera dorénavant l'outil indispensable de l'employé du service gestion. Les réunions que nous

avons eus au sein de l'entreprise avec l'ingénieur en informatique, qui nous a accueillis convivialement, nous ont permis de nous rectifier à chaque fois et d'essayer de faire un sans fautes. Cette application permettra au final une meilleure et bonne gestion et surtout un gain de temps.

Ce projet a été formateur et bénéfique pour nous, il a fait l'objet d'une expérience professionnelle très intéressante et enrichissante. Cette expérience nous a permis de découvrir la rigueur du monde professionnel et de nous familiariser avec les notions de la gestion administrative d'une part, et d'améliorer nos connaissances et nos compétences dans le domaine du logiciel, de la modélisation et la programmation, de renforcer notre esprit de travail en groupe d'une autre part. En outre, notre projet est une opportunité qui nous permis de nous adapter à la gestion des projets qui à son tour nous a permis une insertion et une meilleure intégration dans le domaine professionnel.

Et comme perspectives, nous envisageons de faire évoluer l'application de façon à donner la possibilité aux employés de demander des documents administratifs par envoi de messages (chat, email, ...) et d'améliorer la gestion des réglementations liées aux différentes décisions.



Bibliographie

- [1]: Sonatrach, 2018, <https://fr.wikipedia.org/wiki/Sonatrach>.
- [2] : V. MESSENGER ROTA, Gestion de projet vers les méthodes agiles, Eyrolles, 2008.
- [3] : Scrum pour les nuls, 2018, <https://www.youtube.com/watch?v=kZTLIWkxFN4>.
- [4]: Scrum en moins de 10 minutes.2018. <https://www.thierry-pigot.fr/scrum-en-moins-de-10-minutes>.
- [5] : Guide de démarrage Scrum, <https://www.agiliste.fr/guide-de-demarrage-scrum>.
- [6]: J.GUIOCHET, Le Processus Unifié : Une Démarche Orientée Modèle, IUP NTIE - Master 1. 2009.
- [7]: F.DI GALLO, Méthodologie des systèmes d'information - UML Cours du Cycle Probatoire. 28/11/01.
- [8]: EXTREME PROGRAMMING - méthodes agiles : l'état des lieux, Business Interactif, 2001.
- [9]: UP et une dose de scrum, <http://www.qualitystreet.fr/2007/02/08/up-et-une-dose-de-scrum>.
- [10]: D. BRUNO. Génie logiciel, Object Constraint Language (OCL). Université de Montréal.
- [11] : Java, 2018, http://www.mcours.net/cours/pdf/info/presentation_de_java.pdf.
- [12]: P.ROQUES, UML par la pratique. Paris, Eyrolles, 2009.
- [13]: P.ROQUES, F.VALLÉE, UML 2 en action, De l'analyse des besoins à la conception, Eyrolles.
- [14]: Papyrus : vos diagrammes UML 2, <https://www.urbanisation-si.com/tutoriel-papyrus-vos-diagrammes-uml-2-comme-a-l-epoque-des-manuscrits-de-l-antiquite>.
- [15]: NetBeans, <https://fr.wikipedia.org/wiki/NetBeans>.
- [16]: Oracle Database, https://fr.wikipedia.org/wiki/Oracle_Database.
- [17]: Mise en œuvre d'UML, <https://laurent-audibert.developpez.com/Cours-UML/?page=mise-en-oeuvre-uml>.



Annexe

I. Historique de la SONATRACH

L'exploitation des hydrocarbures en Algérie a existé bien avant la naissance de la SONATRACH. C'était la mission de la SOciété PETrolière de Gérance (SOPEG).

Cette société dont le siège était à Paris a été fondée le 12 mars 1956 par la Compagnie Française des Pétroles Algérie (C F P A) et la Société Nationale de Recherche et Exploitation des Pétroles en Algérie (S N R E P AL).

Après l'indépendance, et grâce au décret n° 36/491 de la nationalisation des hydrocarbures, la SOPEG est devenue SONATRACH le 31 Décembre 1963. Elle se chargeait du transport et la commercialisation des hydrocarbures, et à partir de 1966, son champ d'action s'élargit et englobe la recherche et la transformation des hydrocarbures par le décret n° 66/292. Puis l'état algérien nationalise tout le secteur des hydrocarbures en 24/02/1971, pour une meilleure efficacité économique, ce qui a conduit ensuite à la restructuration de la SONATRACH en 1981.

La stratégie d'internationalisation a permis à SONATRACH de renforcer sa position sur la scène internationale de l'industrie pétrolière et gazière, par une diversification de ses activités sur tous les maillons de la chaîne des hydrocarbures : l'exploration-Production ; le transport par canalisation, la liquéfaction, raffinage et pétrochimie, la commercialisation des produits pétroliers, le transport maritime et autres services.

Après la réorganisation de l'économie nationale, la restauration de la SONATRACH à été engagée et a abouti en 1984 à la création de 17 entreprises, telles que :

- **NAFTAL** : Entreprise nationale de distribution de produits pétroliers.
- **NAFTEC** : Entreprise nationale de raffinage.
- **ASMIDAL** : Entreprise nationale d'énergie et des produits pétroliers
- **ENIP** : Entreprise nationale de pétrochimie.

- **ENPC** : Entreprise nationale des plastiques et caoutchouc.

II. Activités de base de la SONATRACH

Les activités de base de SONATRACH ont été fixées en 1992 afin d'atteindre ses objectifs en :

- L'exploitation et la recherche ;
- L'exploitation des gisements d'hydrocarbures ;
- La liquéfaction et la transformation du gaz ;
- Le transport par canalisation ;
- La commercialisation.

III. Direction Régionale de l'activité Transport

L'activité de transport se répartie en cinq (05) régions de transport localisées différemment. Ces régions sont les suivantes :

- **RTO** : Région Transport Ouest (Arzew);
- **RTH** : Région Transport Haoud El Hamra (Centre distribution);
- **RTE** : Région Transport Est (Skikda);
- **RTC** : Région Transport Centre (Bejaia);
- **RTI** : Région Transport In Amenas.



Résumé

La gestion administrative est une tâche très importante au sein de toute entreprise désirant avoir un bon suivi administratif de ses employés. Notre travail consiste donc à concevoir et à implémenter une application de gestion des employés du service Gestion du département ASL de la RTC-SONATRACH de Bejaia. L'objectif est de permettre aux utilisateurs une bonne gestion régulière et automatique des tâches qui leur sont confiées ainsi qu'une facilité et une rapidité de traitement de l'information. Pour ce faire, nous avons opté pour un processus de développement combiné SCRUM – UP tirant profit des avantages des deux et pour la plateforme de développement Java NetBeans pour la réalisation.

Mots-clés : Gestion administrative, SONATRACH, SCRUM, UP, UML2, OCL, Papyrus, Java, NetBeans, Oracle Database.



Abstract

Administrative management is a very important task in any company wishing to have a good administrative follow-up of its employees. Our job is to design and implement an employee management application of the ASL department management service of the RTC-SONATRACH of Bejaia. The aim is to enable users to manage regularly and automatically the tasks assigned to them as well as to facilitate and speed up the processing of information. To do this, we opted for a combined SCRUM - UP development process leveraging the benefits of both and for the Java NetBeans development platform for implementation.

Keywords: Administrative Management, SONATRACH, SCRUM, UP, UML2, OCL, Papyrus, Java, NetBeans, Oracle Database.