

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ ABDELHAMID IBN BADIS – MOSTAGANEM



Faculté des Sciences Exactes et d'Informatique
Département de Mathématiques et informatique
Filière : Informatique

RAPPORT DE PROJET DE MASTER EN INFORMATIQUE

Option : **Ingénierie des Systèmes d'Information**

THEME :

REALISATION D'UNE APPLICATION DE GESTION DU
COURRIER

Etudiantes : FATAH MAROUA
FERHI ASMAA

Encadrants : HABIB ZAHMANI Mohamed
BENAMMEUR Abdelkader

Projet soutenu dans le cadre de l'arrêté 1275
Année Universitaire 2022-2023

Remerciements

Nous tenons à exprimer notre profonde gratitude envers Dieu pour toutes les bénédictions et les opportunités qu'Il nous a accordées. Nous sommes également reconnaissants envers nous-mêmes pour notre engagement, notre persévérance et notre détermination tout au long de notre parcours académique. Nous avons fait preuve de résilience et nous sommes fiers de nos réalisations.

Nous souhaitons exprimer notre profonde gratitude envers nos familles, en particulier envers nos parents, pour leur dévouement, leurs encouragements constants, leur confiance et leur amour inconditionnel. Ils ont été les piliers essentiels de notre parcours académique, et leur soutien financier, moral et émotionnel a joué un rôle crucial dans notre réussite. Nous leur sommes profondément reconnaissants.

Nous tenons également à remercier notre encadrant, M. Habib Zahmani Mohamed, pour son soutien précieux, ses conseils avisés et sa guidance tout au long de notre travail. Son expertise et son accompagnement ont été d'une grande importance dans notre cheminement académique, et nous lui sommes sincèrement reconnaissants.

De plus, nous voulons exprimer notre reconnaissance envers toutes les personnes qui nous ont soutenus de près ou de loin. Qu'il s'agisse de nos enseignants, de nos collègues, de nos proches, de nos amis ou de toute autre personne qui a contribué à notre réussite, nous vous sommes reconnaissants pour votre présence, vos encouragements et votre aide précieuse. Votre soutien inestimable a été un moteur essentiel de notre parcours et nous vous en sommes profondément reconnaissants.

Enfin, nous adressons nos pensées les plus tendres à l'âme du petit ange au paradis, Amjed Abderahmen Ouled Youcef. Ton amour continuera à briller dans nos cœurs et nous te garderons toujours dans nos souvenirs les plus chers.

Merci à tous pour votre soutien inestimable tout au long de notre parcours. Votre présence, vos encouragements et votre soutien indéfectible ont été des éléments essentiels dans notre réussite. Que Dieu vous bénisse tous.

Résumé

Ce document est une partie du mémoire de projet de fin d'études portant sur la réalisation d'une application web de gestion de courrier au sein d'une entreprise. L'objectif est de créer un système qui gère les courriers entrants et sortants, ainsi que les courriers internes entre les différents services et départements de manière assez facile, simple et efficace.

Le mémoire présente tout d'abord l'environnement et l'étude de l'existant en matière de gestion de courrier manuelle, en mettant en évidence les difficultés rencontrées et les limites des solutions existantes.

Ensuite, le projet de développement de l'application est présenté, en détaillant les différentes caractéristiques et exigences par la rédaction de documents de spécifications des exigences logicielles.

Mots-clés

**Courrier - Application web - Gestion de courrier - numérisation - OPGI -
Automatisation - Suivi de courrier - Spécifications des exigences logicielles -
Développement logiciel - Processus de développement - RUP - Phase - Diagrammes -
Vue**

Abstract

This document is a part of a final project for the development of a web application for mail management within a company. The objective of this application is to create a system that manages incoming and outgoing mails, as well as internal mails between different departments and services, in an easy, simple, and efficient manner.

The document first presents the existing environment and study of manual mail management, highlighting the difficulties and limitations of current solutions.

Then, the project for developing the application is presented, detailing the different features and requirements through the writing of Software Requirements Specifications documents.

Keywords

Mail - Web Application - Mail Management - Scanning - OPGI - Automation - Mail Tracking - Software Requirements Specification - Software Development - Development Process - RUP - Phase - Diagrams – View.

Liste des abréviations

Abréviation	Expression Complète
OPGI	Office de Promotion et de Gestion Immobilière
D.P.I.F	Le Département de la Promotion Immobilière et Foncière
D.G.M.P	Le Département Gestion et Maintenance du Patrimoine
D.M.O	Le Département Maîtrise D'ouvrage
D.F.C	Le Département Finance et Comptabilité
D.R.H.M.G	Le Département Ressources Humaines et Moyens Généraux
BPMN	Business Process Model and Notation
EPIC	Établissement public à caractère industriel et commercial
SGBD	Système de gestion de base de données
IDE	Integrated développement environnement
SEL	Spécifications des Exigences Logicielles (SRS en anglais)
UML	Langage de Modélisation Unifié
RUP	Processus unifié rationnel
sd	Diagramme de séquence

Liste des figures ¹

Figure N°	Titre de la figure	Page
Figure 01	L'Office de Promotion et de Gestion Immobilière Mostaganem	14
Figure 02	L'Organigramme de la Direction Générale	17
Figure 03	BPMN de la gestion des courriers externes (O.P.G.I Mostaganem)	18
Figure 04	BPMN de la gestion des courriers interne	20
Figure 05	BPMN de la gestion des courriers de départ	22
Figure 06	Le diagramme de chorégraphie du courrier	23
Figure 07	Le diagramme de chorégraphie du courrier de départ	23
Figure 08	Le diagramme de conversation	24
Figure 09	Le logo d'OpenCourrier	26
Figure 10	Le logo d'Egesco	27
Figure 11	Le logo de MENARA-GBO	28
Figure 12	L'interface de MENARA-GBO	28
Figure 13	Les phases de la méthode RUP	32
Figure 14	Modèle 4+1 vues	47
Figure 15	Diagramme de contexte statique	48
Figure 16	Diagramme des cas d'utilisation de GEKOUR	50

¹ Pour visualiser les figures plus clairement, veuillez cliquer sur le titre correspondant.

Figure 17	Diagramme de classe	51
Figure 18	Diagramme de déploiement	53
Figure 19	Diagramme de séquence Première authentification	55
Figure 20	Diagramme de séquence S'authentifier	56
Figure 21	Diagramme de séquence Voir les courriers reçus	57
Figure 22	Fragment Ajouter une note à un courrier	58
Figure 23	Fragment Archiver un courrier	58
Figure 24	Fragment Rédiger une réponse	59
Figure 25	Fragment Envoyer un courrier	59
Figure 26	Diagramme de séquence Rédiger un nouveau courrier	60
Figure 27	Diagramme de séquence Rédiger un courrier à l'aide d'un modèle	61
Figure 28	Diagramme de séquence Ajouter un courrier	62
Figure 29	Diagramme de séquence Écrire une note	63
Figure 30	Diagramme de séquence Voir les notes envoyées	63
Figure 31	Diagramme de séquence Voir les notes reçues	64
Figure 32	Diagramme de séquence Voir l'état des traitements	64
Figure 33	Diagramme de séquence Voir l'archive	65
Figure 34	Diagramme de séquence Ajouter un modèle	65
Figure 35	Diagramme de séquence Ajouter un employé dans un sous-groupe	66

Figure 36	Diagramme de séquence Supprimer un employé d'un sous-groupe	67
Figure 37	Diagramme de séquence Ajouter un nouvel employé	68
Figure 38	Diagramme de séquence Supprimer un employé	69
Figure 39	Diagramme de séquence Voir la liste des employés	69
Figure 40	Diagramme de séquence Modifier les informations d'un employé	70
Figure 41	Diagramme de séquence Créer un groupe	71
Figure 42	Diagramme de séquence Supprimer un groupe	72
Figure 43	Diagramme de séquence Voir les statistiques	72
Figure 44	Interface de connexion	77
Figure 45	La structure générale de GEKOUR	78
Figure 46	Le menu déroulant	78
Figure 47	L'interface d'accueil	79
Figure 48	L'interface Ajouter un courrier	80
Figure 49	L'interface de l'éditeur de texte	81
Figure 50	L'interface Ajouter un nouvel employé	82
Figure 51	L'interface Statistiques	83
Figure 52	Statistiques 2	83
Figure 53	Statistiques 3	83

Sommaire

INTRODUCTION GENERALE.....	11
CHAPITRE 1 (ENVIRONNEMENT ET CONTEXTE GENERAL DU PROJET).....	13
1.1 INTRODUCTION	13
1.2 ENVIRONNEMENT ET ORGANISME D'ACCUEIL	13
1.2.1 Présentation de l'O.P.G. I Mostaganem.....	13
1.2.2 Fonctionnement et activité de l'O.P.G. I Mostaganem.....	14
1.2.3 Organisation et structure de l'O.P.G. I Mostaganem	16
1.3 LE CONTEXTE GENERAL	18
1.3.1 Modélisation des tâches étudiées	18
1.3.2 Critique du système.....	24
1.3.3 Problématique.....	25
1.3.4 Étude de l'existant.....	25
1.3.5 Solution proposée.....	29
1.3.6 Processus de développement.....	30
1.3.6.1 Définition :.....	30
1.3.6.2 Objectif du processus de développement :.....	30
1.3.6.3 Types de cycles de développement :.....	30
1.3.6.4 Comparaison entre le processus prédictif et le processus adaptatif :.....	31
1.3.6.5 La Méthode RUP :	31
1.3.6.6 Justification de choix de la méthode RUP	33
1.4 CONCLUSION.....	33
CHAPITRE 2 (SPECIFICATIONS DES EXIGENCES DE NOTRE LOGICIEL).....	34
2.1 INTRODUCTION	34
2.2 LA SEL.....	34
2.2.1 Introduction (Chapitre 1 de la SEL).....	34
2.2.1.1 Objet (section 1.1 de la SEL).....	34
2.2.1.2 Portée (section 1.2 de la SEL).....	34
2.2.1.3 Définitions (section 1.3 de la SEL).....	35
2.2.1.4 Références (section 1.4 de la SEL).....	36
2.2.1.5 Vue d'ensemble (section 1.5 de la SEL).....	36
2.2.2 Description générale (Chapitre 2 de la SEL) :	37
2.2.2.1 Environnement.....	37

2.2.2.2	Fonctions.....	37
2.2.2.3	Caractéristiques des utilisateurs.....	37
2.2.2.4	Contraintes.....	39
2.2.2.5	Hypothèses et dépendances.....	39
2.2.3	Exigences spécifiques (Chapitre 3 de la SEL).....	40
2.2.3.1	Exigences des interfaces externes.....	40
2.2.3.2	Exigences fonctionnelles.....	40
2.2.3.2.1	Directeur/Administrateur.....	40
2.2.3.2.2	Responsable.....	41
2.2.3.2.3	Employé.....	43
2.2.3.3	Exigences de performance.....	43
2.2.3.4	Contraintes de conception.....	44
2.2.3.5	Attributs.....	44
2.2.3.6	Index.....	45
2.3	CONCLUSION.....	45
CHAPITRE 3 (CONCEPTION)		46
3.1	INTRODUCTION.....	46
3.2	MODELE DE KRUCHTEN.....	46
3.2.1	Vue des cas d'utilisation.....	47
3.2.2	Vue logique.....	51
3.2.3	Vue de réalisation.....	52
3.2.4	Vue de déploiement.....	53
3.2.5	Vue des processus.....	53
3.3	CONCLUSION.....	73
CHAPITRE 4 (TRAITEMENT)		74
4.1	INTRODUCTION.....	74
4.2	OUTILS UTILISES POUR L'IMPLEMENTATION.....	74
4.2.1	De développement.....	74
4.2.2	De collaboration.....	75
4.3	CAPTURES D'ECRAN DES INTERFACES PRINCIPALES.....	77
4.4	CONCLUSION.....	84
CONCLUSION GENERALE.....		85
BIBLIOGRAPHIE		86

Introduction Générale

Le courrier est un élément clé de toute organisation. La gestion de ce dernier est un aspect crucial au sein des entreprises. Cependant, le traitement manuel du courrier peut être fastidieux et chronophage, en particulier lorsque l'organisation traite de grandes quantités de courrier chaque jour.

Pour remédier à cela, de nombreuses entreprises ont besoin de solutions efficaces pour gérer l'ensemble du flux de courrier et cherchent à améliorer ce processus en utilisant des outils numériques [1].

Dans ce contexte, la mise en place d'une application de gestion du courrier peut s'avérer être une solution idéale.

Sachant que l'informatique devient indispensable et sa présence de plus en plus répandue en raison de son efficacité ainsi que de son utilité. En effet, l'informatique désigne l'automatisation du traitement de l'information par un système et une machine. On entend également par l'informatique l'ensemble des sciences et techniques en rapport avec le traitement de l'information. Elle est de plus en plus utilisée dans tous les domaines d'activités, y compris celui de la gestion du courrier qui est le thème de notre étude.

L'objectif de ce projet est donc de développer une application web de gestion du courrier qui permettra aux entreprises de traiter rapidement et efficacement leur courrier entrant et sortant. L'application sera conçue pour être facile à utiliser et intuitive pour les utilisateurs. Elle offrira des fonctionnalités qui permettront aux utilisateurs de gagner du temps et d'améliorer leur productivité, en réduisant les coûts liés à la gestion manuelle du courrier et en fournissant une solution efficace pour une gestion rapide et efficiente du courrier.

Pour atteindre cet objectif, nous avons utilisé la méthode RUP qui divise le processus de développement en quatre phases distinctes, dont chacune sera détaillée dans un chapitre.

Ce mémoire comporte quatre chapitres :

Chapitre 1 : Ce chapitre contient la présentation de l'organisme d'accueil et la présentation générale du problème et de la solution proposée.

Chapitre 2 : Ce chapitre représente la première phase du méthode RUP, il contient une spécification des exigences logicielles.

Chapitre 3 : Ce chapitre représente la deuxième phase du méthode RUP, il contient la modélisation conceptuelle de l'application.

Chapitre 4 : Ce chapitre contient une présentation des outils utilisés dans la troisième phase du RUP : l'implémenter des premières fonctionnalités de l'application, ainsi que des captures d'écran des interfaces.

Chapitre 1

Environnement et contexte général du projet

1.1 Introduction

Ce chapitre est composé de trois parties principales dont la première contient la présentation et la structure de l'Office de Promotion et de Gestion immobilière de Mostaganem.

Puis dans la seconde partie nous parlerons de la problématique et présenterons quelques applications sur le marché et ensuite nous parlerons de la solution proposée et des objectifs à atteindre.

Enfin, dans la dernière partie de ce chapitre, nous définirons la méthode de développement adoptée pour implémenter notre application.

1.2 Environnement et organisme d'accueil

1.2.1 Présentation de l'O.P.G. I Mostaganem

L'Office de Promotion et de Gestion Immobilière est un établissement public à caractère industriel et commercial (EPIC) détenu par d'État et est responsable de la promotion, de la restauration et de la réhabilitation des biens immobiliers à l'échelle de la wilaya de Mostaganem [2].



Figure 01 : L'Office de Promotion et de Gestion Immobilière Mostaganem

L'organisation a été créée par le décret n° 147/76 du 23/10/1976 fixant les conditions de création des OPGI, avec le statut d'une entreprise de qualité de fonctionnement EPIC.

L'établissement public est géré - sous la tutelle du Directeur Général - par un Conseil d'Administration composé du Directeur Central du ministère de l'Habitat et de l'Urbanisme, du Directeur du Logement de Mostaganem, du Directeur du Budget de la wilaya de Mostaganem, du Trésorier de la Wilaya de Mostaganem, du Directeur de l'Administration Locale de la wilaya de Mostaganem et des Représentants des Travailleurs [3].

1.2.2 Fonctionnement et activité de l'O.P.G. I Mostaganem

À travers ses 14 unités réparties sur la wilaya et ses plus de 427 employés, l'Office est investi de plusieurs missions, parmi lesquelles :

1. La promotion et la gestion immobilière et foncière.
2. La réalisation des opérations d'accession à la propriété de l'habitation.
3. La location et/ou la cession de logements et de locaux à usage professionnel, commercial et artisanal.
4. La promotion et la réalisation de programmes d'habitat public.
5. La réhabilitation et la restructuration d'immeubles.

L'Office s'engage à respecter cinq principes fondamentaux dans l'accomplissement de ses missions :

1. Définir et délimiter le but poursuivi par l'Office à travers :
 - Acquisition d'une vue d'ensemble.
 - Choix des priorités d'action.
 - Analyse du produit.
 - Analyse de déroulement.
 - Analyse du processus.
 - Statistiques.
2. Étudier et prévoir les moyens à mettre en œuvre à travers :
 - Mise en évidence des problèmes du bon choix des solutions.
 - Audit opérationnel des différentes fonctions de l'office.
3. Préparer et planifier les moyens rendus nécessaires à travers :
 - Analyse générale du système commercial et productif.
 - Procédure Budgétaire et évaluation des coûts.
 - Plan de recouvrement des loyers, et de réalisation de logements.
4. Réaliser et harmoniser à travers :
 - Mise en place des solutions choisies.
 - ERP.
 - L'assurance et la vérification de la qualité totale et le flux tendu.
5. Gérer et contrôler à travers :
 - Assurance de retour d'informations.
 - Mesure des résultats.
 - Modifications si nécessaires.
 - Budget et Plan.
 - Analytique et tableaux de bord.
 - Procédure budgétaire.

1.2.3 Organisation et structure de l'O.P.G. I Mostaganem

L'O.P.G. I est une organisation assez importante, puisqu'elle regroupe environ 427 personnes, séparées en plusieurs équipes qui à leur tour sont répartis en cinq départements.

Par exemple, le département de Ressources Humaines et Moyens Généraux est le département chargé de la gestion des moyens humains et matériels nécessaires pour assurer le bon fonctionnement de l'ensemble des structures de l'entreprise, ainsi que la coordination et le contrôle des activités liées à la gestion administrative et professionnelle du personnel de l'Office. Ce rôle est exercé par quatre services distincts chargés chacun des missions bien précisées :

Service de Gestion des Ressources Humaines : qui est chargé de la gestion de l'ensemble du personnel de l'Office.

Service des Moyens Généraux : qui est chargé de La gestion des moyens matériels, l'approvisionnement et la gestion des stocks.

Service Contentieux : qui est chargé de la mise en œuvre de toutes les procédures prévues par la législation pour la protection et de défense des intérêts de l'Office.

Service informatique : La mission du service informatique au titre de conception et d'informatisation des systèmes d'information porte principalement sur les tâches suivantes :

- La conception de la politique générale d'information au sein de l'Office et l'élaboration du plan directeur informatique.
- La conception des programmes et de traitement informatique des opérations et activités de l'entreprise.
- Le suivi des évolutions technologiques dans les domaines informatiques.
- Assurer l'exploitation rationnelle des moyens informatiques disponibles et leur maintenance.
- Installation, gestion et maintenance des réseaux.
- Développement de la formation des employés en informatique.
- Acquisition des équipements, matériels et logiciels dans les conditions économiques requises (qualité, prix et délais) et le respect des procédures mises en place.
- Ordonnancement des factures d'achats du matériel et des consommables informatiques.
- La maintenance du parc informatique au niveau de l'entreprise (maintenance

préventive, maintenance des installations systèmes et équipements sur site, etc.).

- Proposition de réparation des matériels et équipements gérés.

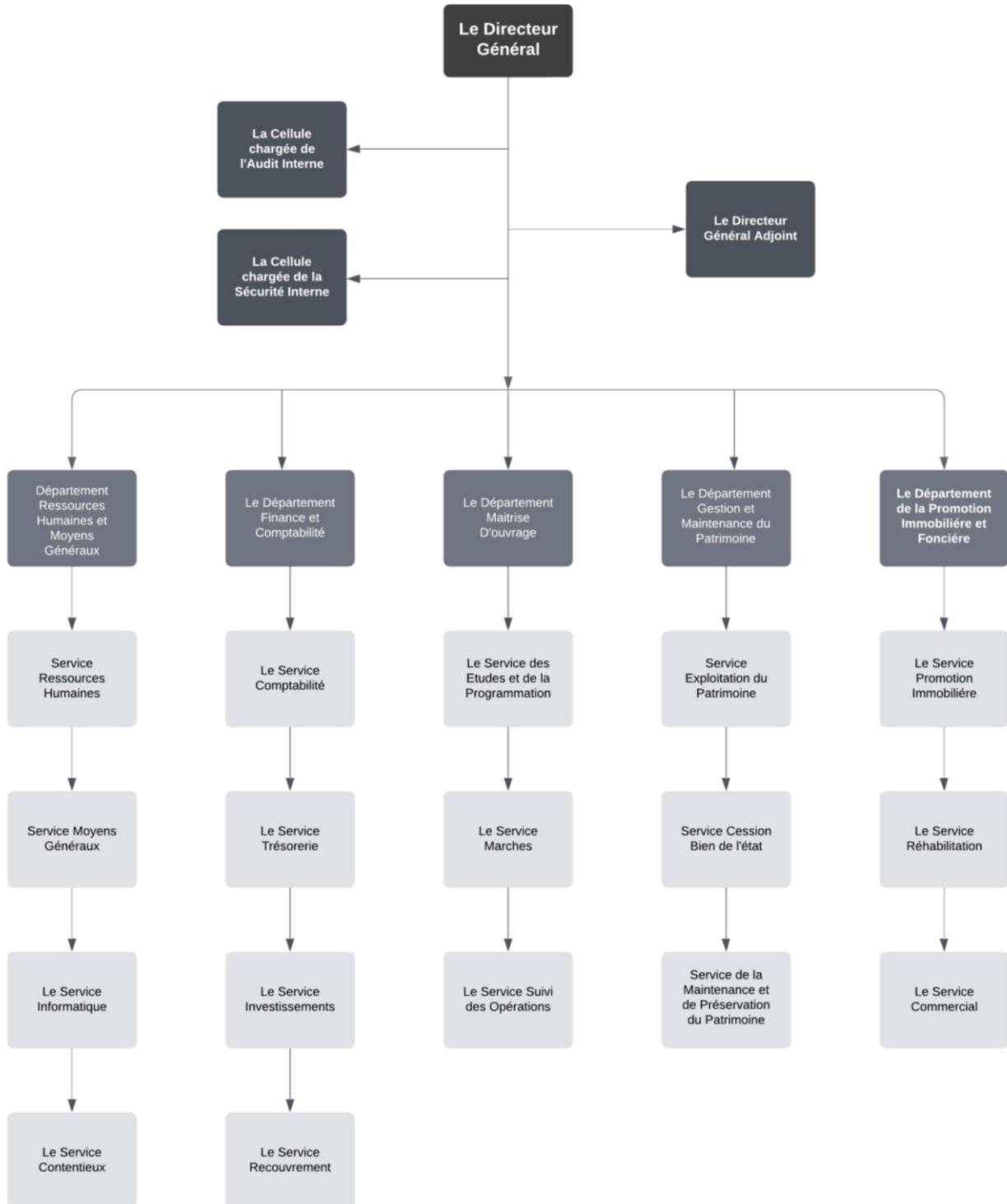


Figure 02 : L'Organigramme de la Direction Générale [4]

1.3 Le contexte général

1.3.1 Modélisation des tâches étudiées

BPMN (Business Process Model and Notation) c'est-à-dire modèle de procédé d'affaire et notation, est un langage graphique standardisé utilisé pour modéliser les processus d'affaires, il permet de représenter visuellement les processus métier sous forme de diagrammes de flux de processus, afin de mieux comprendre, communiquer et gérer les processus métier. [5]

- Processus métier du traitement des courriers externes : Le courrier externe : les courriers reçus par une entreprise ou une organisation à partir d'une source extérieure

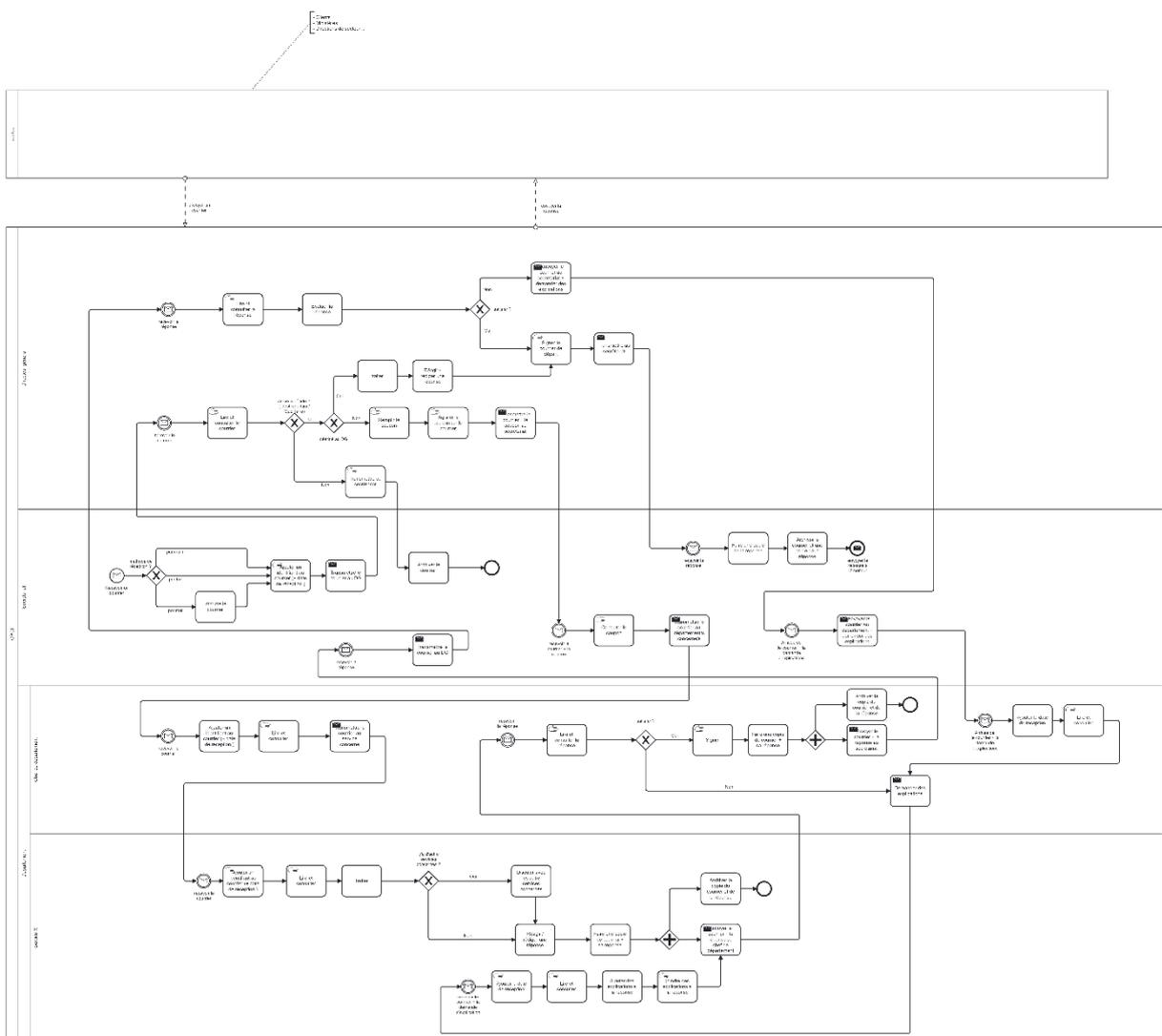


Figure 03 : Processus métier du traitement des courriers externes (O.P.G.I Mostaganem)

Le courrier externe peut être reçu de trois façons. Le client peut l'envoyer par fax ou le déposer manuellement à la réception de l'entreprise. Il peut également l'envoyer par voie électronique, auquel cas le courrier sera imprimé pour pouvoir être géré manuellement.

Le processus de gestion du courrier commence au niveau de la secrétaire, qui est chargée d'attribuer un identifiant et d'indiquer la date de réception du courrier avant de le transmettre au directeur général.

Le directeur traite le courrier s'il lui est destiné et signe la réponse qui sera envoyée à l'expéditeur. Sinon, en fonction du sujet du courrier, il le transmet au service chargé de le traiter.

Le courrier doit ensuite passer par le chef de département avant d'arriver au service concerné. Ensuite, le chef de service le traite et rédige une réponse, qui sera évaluée par le chef de département avant de la transmettre au directeur général. Ce dernier évaluera la réponse et la signera pour qu'elle soit prête à être envoyée à l'expéditeur, par la secrétaire.

- Processus métier du traitement des courriers internes :

Les courriers internes : les courriers échangés entre les différentes entités ou services d'une entreprise ou d'une organisation

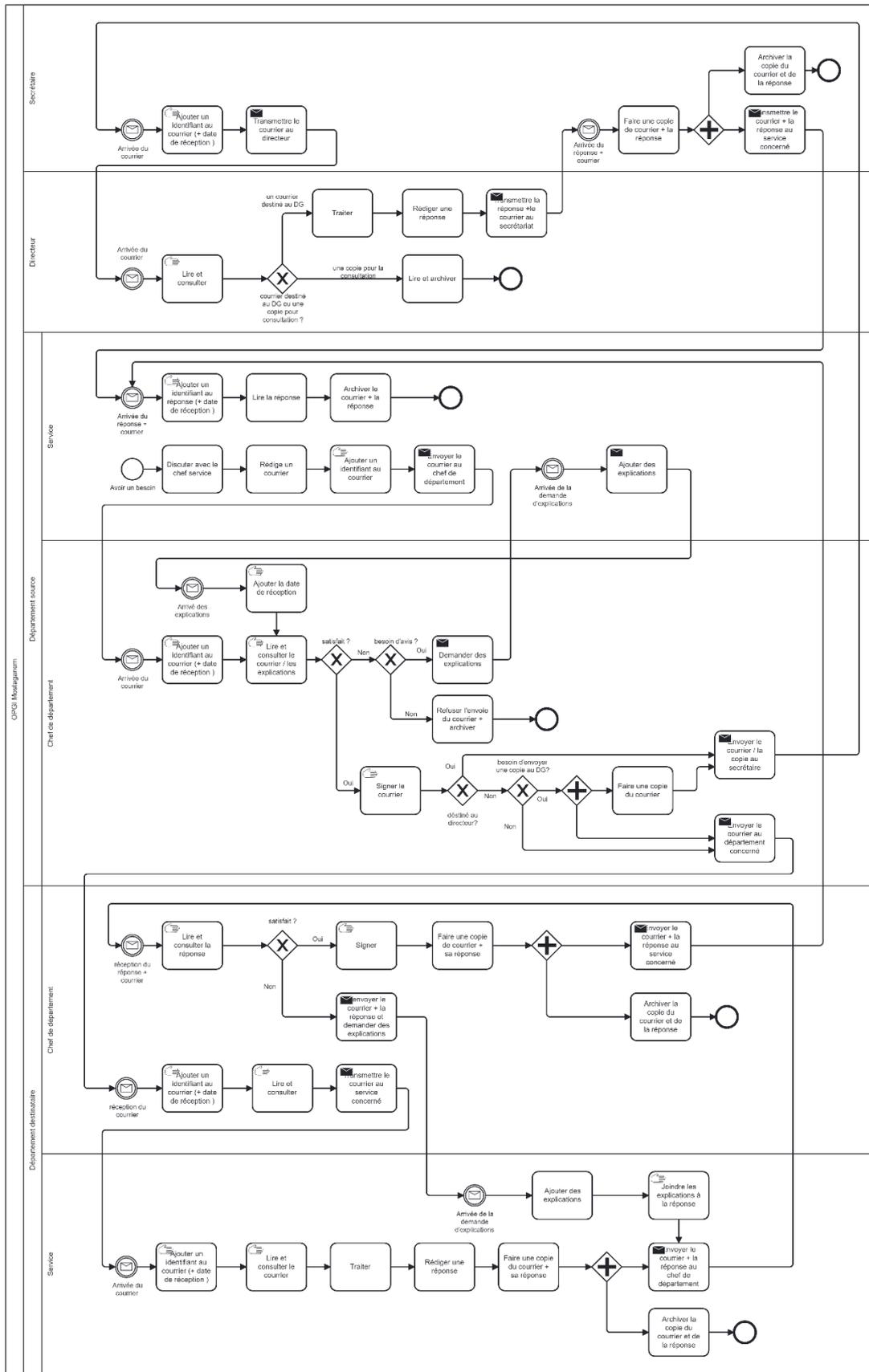


Figure 04 : Processus métier du traitement des courriers internes

En cas de besoin de communication entre deux services, le service rédige et identifie un courrier interne, puis le transmet au chef de département pour l'évaluer et le signer avant de l'orienter vers le service concerné. Dans certains cas, une copie sera envoyée au directeur général pour l'informer.

Au niveau du département adressé, le courrier arrivera au service concerné après avoir été examiné par le chef de département. Au service concerné, le courrier sera consulté et traité, et sa réponse sera rédigée et prête à être envoyée au service expéditeur, passant par le chef de département qui doit l'évaluer.

La réponse au retour doit être consultée par le chef de département expéditeur, qui va la transmettre par suite au chef de service source.

- Processus métier du traitement des courriers de départ :

Le courrier de départ : les courriers envoyés par une entreprise ou une organisation à une source extérieure.

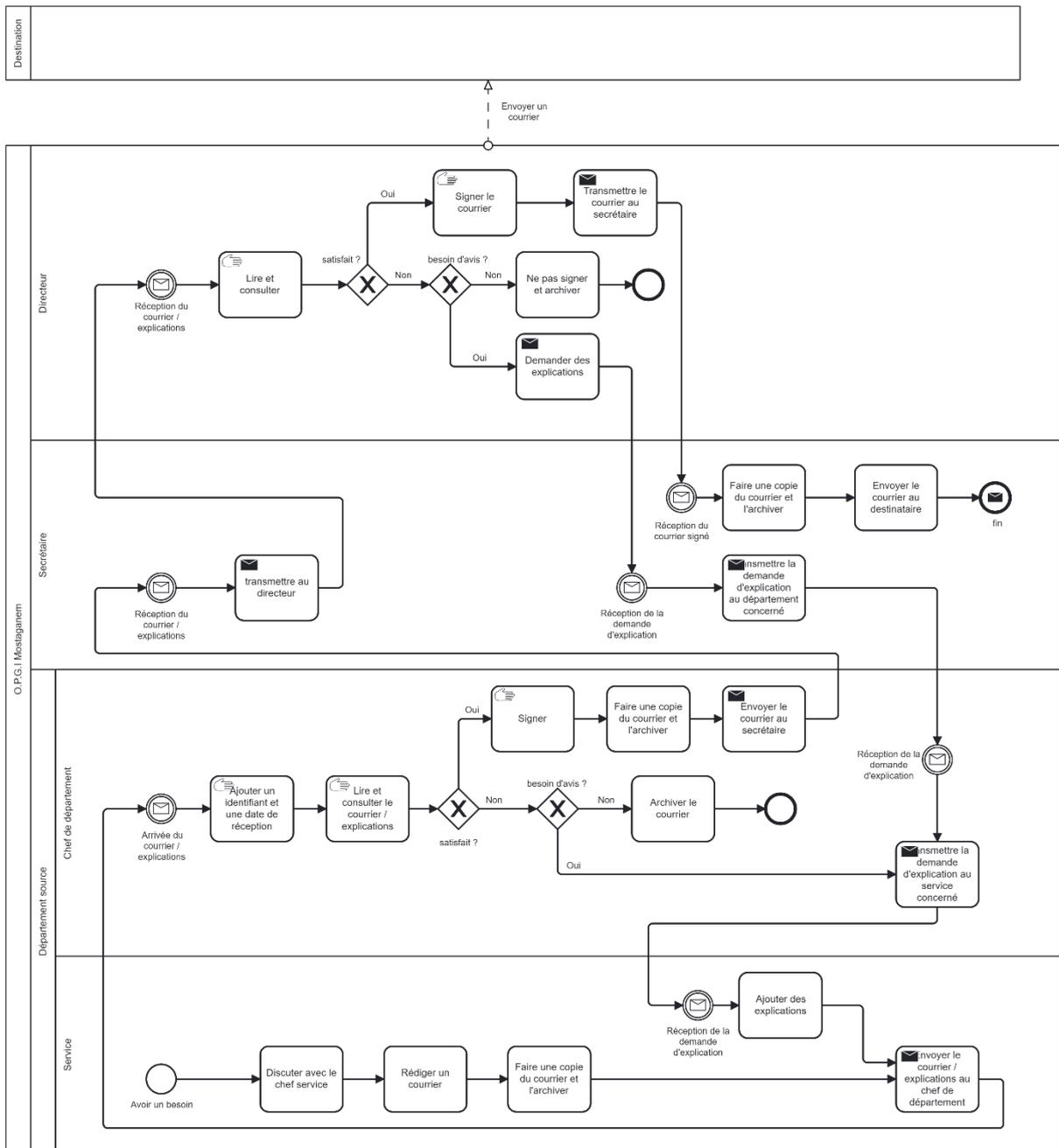


Figure 05 : Processus métier du traitement des courriers de départ

En cas de besoin de communication avec une certaine autorité ou une entité, le service rédige et identifie un courrier, puis le transmet au chef de département qui va l'examiner avant de le transmettre au directeur général.

Après le passage par la secrétaire, le courrier arrivera au directeur général qui l'évalue et le signera pour qu'il soit prêt à être envoyé par la secrétaire.

- Diagrammes de chorégraphie :

Une chorégraphie est la modélisation d'un comportement attendu entre des participants qui interagissent les uns avec les autres et qui veulent coordonner leurs activités ou leurs tâches à l'aide de messages. [6]

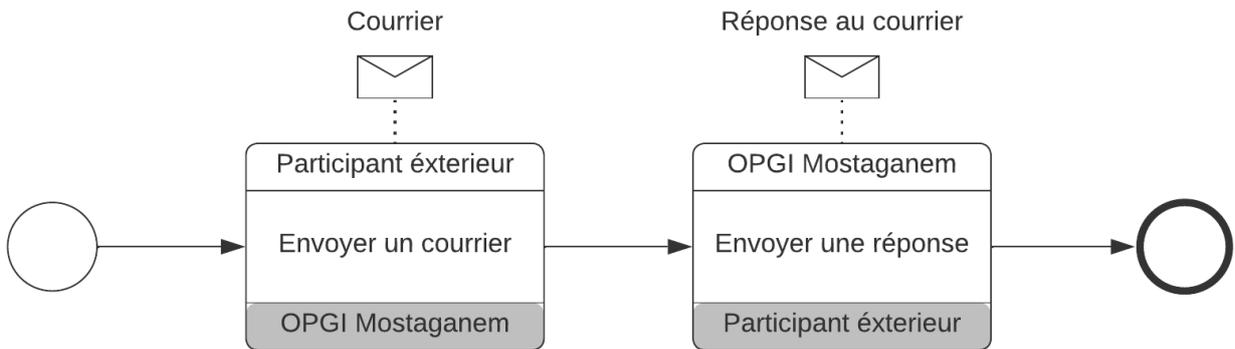


Figure 06 : Diagramme de chorégraphie d'échange de courrier

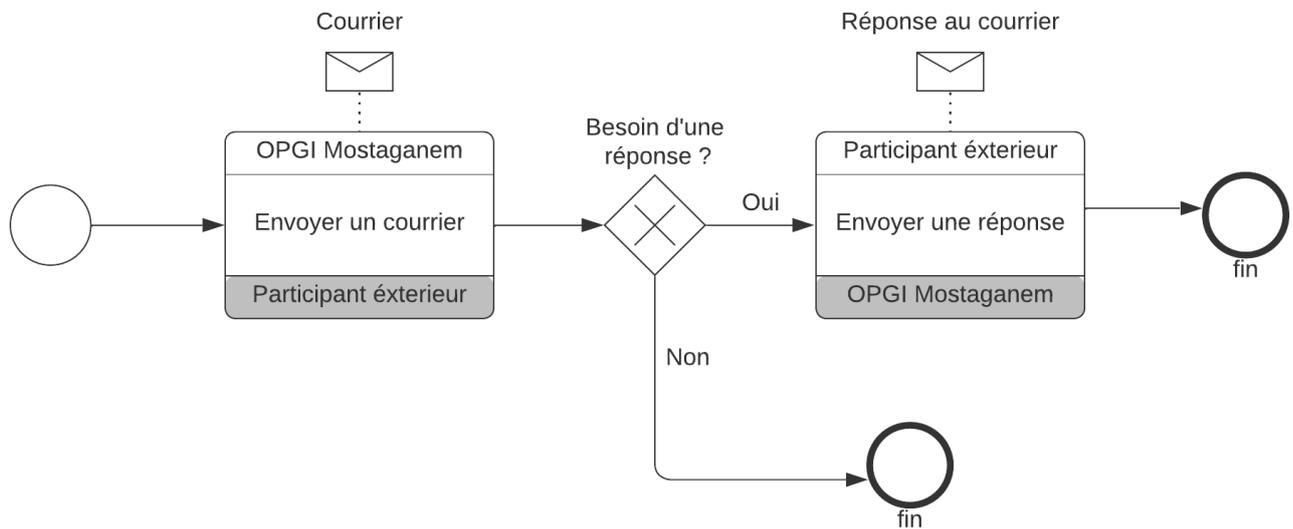


Figure 07 : Diagramme de chorégraphie du courrier de départ

- Diagramme de conversation :

Un diagramme de conversation est la description informelle d'un diagramme de collaboration.

Il représente des échanges de messages (présentés sous la forme d'un regroupement de flux de messages), logiquement reliés entre les participants et qui concernent un objet d'affaires d'intérêt. Il représente un ensemble de flux de messages qui est regroupé ensemble [7].

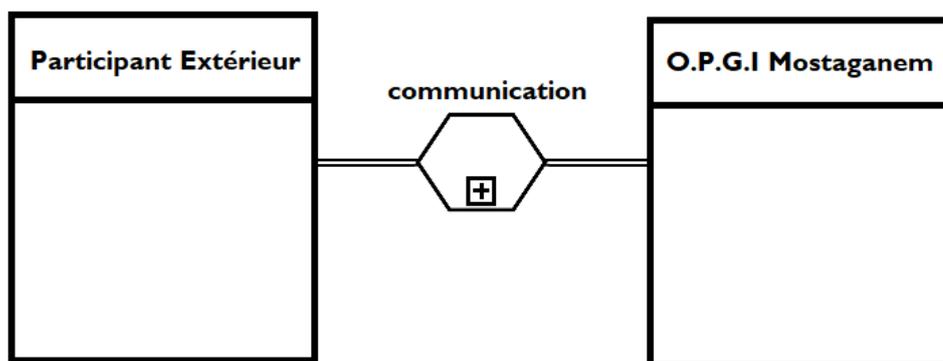


Figure 08 : Diagramme de conversation d'échange de courrier

1.3.2 Critique du système

Le système de gestion du courrier en place à l'O.P.G. I Mostaganem est complexe et comprend de nombreuses tâches supplémentaires et répétitives, sous la supervision du directeur général et des chefs de départements.

Il comporte également des boucles qui peuvent être répétées à l'infini dans certains cas, afin que le courrier puisse sortir de cette boucle répétitive, il doit être évalué et approuvé par le chef de service, le chef de département et le directeur général. Cette procédure peut prolonger davantage le délai de traitement du courrier, ce qui peut nuire aux intérêts de l'entreprise et retarder les réponses aux clients.

Le système d'archivage des courriers pose également plusieurs problèmes, car chaque courrier peut avoir plusieurs copies et identifiants différents, un pour chaque service et département qu'il traverse. Cela peut poser des problèmes de suivi et d'archivage, en particulier lorsqu'un même courrier a plusieurs identifiants.

De plus, le délai de traitement du courrier, depuis son arrivée jusqu'à l'envoi de la réponse, est souvent très long, ce qui peut entraîner des retards préjudiciables pour l'entreprise et ses clients. Également, le délai de transmission entre les différents services et départements est encore très long, ce qui peut avoir un impact négatif sur l'efficacité et la rapidité de traitement des dossiers, ainsi que sur la satisfaction des clients et partenaires de l'entreprise.

1.3.3 Problématique

Quotidiennement, des millions de courriers sont envoyés à travers le monde, sous forme papier ou électronique.

Le secteur de l'immobilier à lui seul génère des milliers de correspondances par mois afin de communiquer régulièrement avec les clients, les ministères, les différentes directions et administrations du secteur.

La gestion de ces courriers est une tâche importante qui exige rigueur et organisation, comprenant la concertation, l'acheminement vers les services/personnes concernés et l'archivage.

Toutefois, avec un grand nombre de courriers reçus, cette tâche peut rapidement devenir complexe si elle est mal gérée, augmentant ainsi le risque de perdre des informations importantes et utiles.

Au sein de l'entreprise O.P.G.I Mostaganem, un grand nombre de courriers et de correspondances externes sont échangés et gérés manuellement chaque jour avec le Ministère du Logement, de l'Urbanisme et de la Ville, les clients de diverses classes et catégories (locataires et propriétaires), ainsi que les administrations du secteur. De plus, un volume important de courriers circule en interne entre les différents départements et services.

1.3.4 Étude de l'existant

Dans ce chapitre et dans le cadre de la collecte des informations nécessaires à la conception et à la réalisation de notre application de gestion de courrier, nous avons procédé à

l'étude de différentes applications et logiciels existants sur le marché ayant des objectifs similaires à ceux de notre application.

→ Critère de choix : les plus connus et les plus téléchargés

OpenCourrier



Figure 09 : Le logo d'OpenCourrier

OpenCourrier est un logiciel de gestion de courrier dédié pour les organisations administratives tel que les mairies [8].

Il répond au besoin de gestion centralisée du courrier avec une gestion des tâches au sein d'une entreprise avec un organigramme de traitement du courrier.

Les tâches d'OpenCourrier :

- Gérer le courrier entrant/sortant.
- Affecter un courrier au service concerné.
- Numériser le courrier arrivé.
- Notifier les services après l'affectation des tâches.
- Les tâches peuvent ainsi être suivies jusqu'à ce qu'elles soient traitées par les services.

Les points positifs d'OpenCourrier :

- Rapidité : l'enregistrement d'un courrier se fait en 10 secondes.
- Rappel automatique des courriers en attente de traitement.
- Mise en place de workflows de validation (simples ou avancés).
- Mise en place des procédures d'archivage.
- Un module de collecte statistique.

Les points négatifs d'OpenCourrier :

- Complexité d'utilisation.
- Beaucoup de mises-à-jour.
- Traitement des courriers documentaire seulement.

Egesco



Figure 10 : Le logo d'Egesco

Egesco est une application de gestion de courrier développée par L'AGETIC (Mali).

Créée pour concevoir, développer et entretenir les infrastructures TIC des services publics d'une façon bien sécurisée [9].

Les tâches d'Egesco :

- Gérer le courrier administratif.
- Numériser le document ou fichier reçu.
- L'enregistrement d'un courrier et de ses pièces jointes.
- La transmission, l'annotation, le traitement et le classement des courriers.
- La gestion des erreurs et de non-traitement par les alertes.

Les points positifs d'Egesco :

- La sécurité et la confidentialité des informations.
- Le système de gestion des erreurs.
- La génération des alertes visuelles à la hiérarchie émettrice.
- L'archivage et la sauvegarde des données.

Les points négatifs d'Egesco :

- Utilisable seulement dans le réseau local.
- Interface non intuitive et compliquée à utiliser.
- Utilisation limitée aux documents papier uniquement.

MENARA-GBO



Figure 11 : Le logo de MENARA-GBO



The screenshot displays the MENARA-GBO web application interface. At the top, there is a header with the SIAT logo and the text 'MENARA-GBO'. Below the header, a navigation bar contains links for 'Déconnexion', 'Accueil', 'Nouveau', 'Enregistrer', 'Supprimer', and 'Imprimer'. A sidebar on the left lists menu items: 'Administration', 'Codification', 'Courier Entrant' (with sub-items 'Nouveau Courier' and 'Consulter Les Courries'), and 'Courier Sortant'. The main content area is titled 'Fiche Courrier' and contains several input fields and dropdown menus. The fields include 'Code Courier', 'Date', 'Objet', 'Caractéristique', 'Mode Transmission', 'Etat Courier' (set to 'Arrivé'), 'Emis Le', 'Reçu Le', 'Date de Butée', 'Date de Prémption', 'Référence', and 'Information'. A bottom navigation bar includes tabs for 'Détails Courier', 'Le correspondant', 'Interlocuteurs interne/externes', 'Indexation', 'Fichier attaché', and 'Suivi des courriers'. The footer of the page reads 'SIAT © 2011 Tous droits réservés'.

Figure 12 : L'interface de MENARA-GBO

L'entreprise tunisienne SIAT propose le logiciel MENARA-GBO pour un pilotage collaboratif et transversal du courrier entrant, sortant et interne.

Il assure le traitement des courriers et permet d'entretenir des relations avec l'ensemble des contacts [10].

Les tâches de MENARA-GBO :

- L'enregistrement et le traitement des courriers entrants, sortants et internes.

- Le suivi des courriers.
- L'indexation par code / mots clés.
- L'édition des statistiques.
- La diffusion des courriers vers les services.

Les points positifs de MENARA-GBO :

- Interface intuitive et facile à utiliser.
- La simplicité du Workflow.
- Le contrôle et le suivi efficace des courriers.
- L'archivage numérique.

Les points négatifs de MENARA-GBO :

- La nécessité des mise-à-jour (l'utilisation des anciennes technologies).
- La saisie manuelle des métadonnées.

1.3.5 Solution proposée

Suivant les avancées technologiques, il est envisageable de concevoir une application de gestion des courriers automatisée qui facilite la gestion des correspondances pour les structures concernées. Cette application permettrait d'enregistrer chaque courrier reçu, son traitement, la réponse envoyée avec une option d'archivage, en garantissant la sécurité des informations traitées.

De plus, l'application facilitera la communication entre les utilisateurs de l'application en envoyant et en recevant des notes.

L'application fournira aussi des statistiques utiles pour la prise de décision de l'entreprise cliente.

En supplément, une fonctionnalité de reconnaissance automatique de texte basée sur l'intelligence artificielle pourrait être intégrée à l'application pour numériser le courrier et extraire des informations et des métadonnées, ainsi que pour déterminer le service ou la personne destinataire.

Pour réaliser notre projet, il est essentiel de suivre un processus de développement logiciel bien défini. Ce processus nous guidera tout au long du cycle de vie de développement,

en nous fournissant une structure claire pour mener à bien chaque étape du projet. En respectant ce processus, nous pourrions efficacement planifier, collecter les exigences, concevoir, programmer, tester, déployer et maintenir notre logiciel.

1.3.6 Processus de développement

1.3.6.1 Définition :

Un processus de développement logiciel est une série d'étapes organisées et structurées visant à concevoir, créer, tester et livrer des logiciels de manière efficace et de haute qualité. Il fournit un cadre pour gérer le cycle de vie du développement logiciel, depuis la phase de spécification des exigences jusqu'à la maintenance continue du produit fini.

1.3.6.2 Objectif du processus de développement :

La réalisation d'une application sans faute (bug / erreur) qui satisfait les besoins du client tout en respectant le budget (les coûts) et le délai prévu à l'avance.

1.3.6.3 Types de cycles de développement :

Parmi les cycles de vie de développement logiciel existants on trouve :

1. **Processus prédictif** : il vise à planifier et à exécuter un projet de développement logiciel en se basant sur des estimations et des prévisions détaillées. Ce type de processus repose sur l'idée que les exigences du projet peuvent être entièrement comprises et définies dès le départ, et que la planification peut être effectuée de manière précise pour l'ensemble du cycle de vie du projet.

On trouve comme exemple :

- a. Modèle en v
 - b. Modèle en cascade
2. **Processus adaptatif** : est un cycle de vie itératif et incrémental qui met l'accent sur la flexibilité, la réactivité et l'adaptabilité aux besoins changeants et aux nouvelles informations qui émergent tout au long du projet.

Il privilégie la livraison par tranche en commençant par les fonctionnalités utiles au client.

On trouve comme exemple :

- a. Méthode Agile
 - i. Scrum
 - ii. XP, ...
- b. Processus Unifié
 - i. RUP
 - ii. 2TUP

1.3.6.4 Comparaison entre le processus prédictif et le processus adaptatif :

	Processus Adaptatif	Processus Prédictif
Avantages	<ul style="list-style-type: none">- Adaptabilité aux changements des besoins- Livraison rapide de fonctionnalités incrémentales- Collaboration étroite avec les utilisateurs et les parties prenantes- Possibilité d'ajuster les fonctionnalités en fonction des retours d'expérience	<ul style="list-style-type: none">- Planification détaillée des étapes du projet- Visibilité des délais et des coûts- Gestion des risques anticipée- Documentation détaillée
Inconvénients	<ul style="list-style-type: none">- Difficulté à prédire précisément les délais et les coûts- Besoin d'un engagement continu des parties prenantes- Moindre visibilité des délais et des coûts- Gestion des exigences en évolution	<ul style="list-style-type: none">- Rigidité face aux changements- Moindre réactivité aux nouvelles informations- Risque de non-conformité aux besoins réels- Difficulté à intégrer de nouvelles fonctionnalités en cours de projet

Table 1 : Comparaison entre le processus adaptatif et le processus prédictif

→ En comparant les deux processus, et selon le contexte spécifique de notre projet, le processus de développement le plus approprié pour la mise en œuvre de GEKOUR est **le processus adaptatif**, nous choisirons donc **la méthode RUP**.

1.3.6.5 La Méthode RUP :

La méthode de Processus Unifié (UP) s'intéresse au cycle de vie d'un logiciel et le gère. En ce qui concerne les logiciels orientés objet, elle se concentrera sur son développement.

La méthode RUP (Rational Unified Process), quant à elle, est une des émanations de la méthode PU, qui s'attache à donner un cadre précis au développement du logiciel. C'est une méthode générique, itérative et incrémentale assez lourde mais qui s'adapte très facilement aux processus et aux besoins du développement [11].

Ce processus se compose de quatre phases principales, chacune se termine par un jalon majeur :

- **Pré-étude** : Cette phase est axée sur la compréhension des objectifs commerciaux du projet. Elle implique l'identification des parties prenantes, la définition des fonctionnalités de base du système, et la planification globale du projet.
- **Élaboration** : Durant cette phase, une analyse approfondie des exigences du système est effectuée, et une architecture logicielle est élaborée.
- **Construction** : Cette phase se concentre sur le développement du système logiciel. Il s'agit de réaliser l'implémentation des premières fonctionnalités et effectuer les tests unitaires.
- **Transition** : Au terme de cette phase finale, le système est préparé pour le déploiement. Les tests d'intégration et de système sont réalisés.



Figure 13 : Les phases de la méthode RUP

1.3.6.6 Justification de choix de la méthode RUP

Nous avons opté pour l'adoption du processus de développement RUP par rapport à d'autres processus adaptatifs en raison de :

- La complétude : la méthode RUP fournit des lignes directrices détaillées pour chaque étape du processus, y compris l'analyse des besoins, la conception, le développement, les tests et la maintenance.
- L'accent sur la qualité : la méthode RUP étant itérative et incrémentale, des tests sont effectués de manière itérative tout au long du processus de développement ce qui permet de détecter et de corriger les problèmes et les bogues au plus tôt, et donc de livrer un produit de meilleure qualité.
- Le support et ressources : la méthode RUP est une méthodologie solidement établie et très répandue, il y a donc une abondance de ressources disponibles.

1.4 Conclusion

Dans ce chapitre, nous avons présenté l'organisme d'accueil, ainsi que le système qu'il utilise pour gérer son courrier. Cette présentation comprenait une modélisation du système et une mise en évidence de ses limites

Nous avons analysé l'existant pour identifier les points faibles, puis nous avons proposé notre solution et nous avons précisé la méthodologie de développement.

Cette étape nous a permis d'entamer la phase de pré-étude qui sera détaillée dans le chapitre suivant.

Chapitre 2

Spécifications des exigences de notre logiciel

2.1 Introduction

Dans ce chapitre nous allons définir et présenter la spécification des exigences logicielles pour notre application web, et c'est la première phase de la méthode RUP.

Cette phase est essentielle dans le processus de développement, car elle permet de comprendre les besoins et de déterminer les parties prenantes et les utilisateurs, déterminer leurs besoins, besoins fonctionnels et non fonctionnels, et les contraintes de conception.

2.2 La SEL

2.2.1 Introduction (Chapitre 1 de la SEL)

2.2.1.1 Objet (section 1.1 de la SEL)

Ce document sert à présenter les spécifications d'une application de gestion de courrier au sein d'une entreprise, et la description de ses exigences et ses processus, en respectant la norme IEEE 830-1993 [12].

Public-cible : Le document s'adresse aux développeurs qui devront bien appréhender les exigences du projet pendant la réalisation de celui-ci.

Il servira aussi aux clients qui auront une première vue du système.

2.2.1.2 Portée (section 1.2 de la SEL)

GEKOUR est le nom du nouveau système à développer pour la gestion de courrier. Il réalisera des fonctions de base qui permettront au directeur et aux employés d'échanger facilement et rapidement des courriers.

Via l'interface principale de l'application, un agent peut scanner et ajouter des courriers en leur attribuant un identifiant et une date de réception.

L'application à créer prend également en charge la transmission automatique des courriers aux services concernés et la consultation de l'avancement du traitement et également la rédaction des réponses.

2.2.1.3 Définitions (section 1.3 de la SEL)

La plupart des termes utilisés dans cette SEL sont explicites et sont communs aux applications de gestions de courriers et aux logiciels de reconnaissance optique de caractères.

Définitions

Terme	Définition
Courrier	Ensemble des lettres, imprimés, paquets qui sont acheminés et distribués vers une destination précisée.
Application Web	L'application web est une application manipulable directement en ligne grâce à un navigateur web [13].
Numérisation	La numérisation consiste à créer, à partir d'un document original au format papier, une copie numérique qui peut être stockée et utilisée plus tard.
Utilisateurs	Les personnes qui utilisent le système
Le courrier interne	Les courriers échangés entre les différentes entités ou services d'une entreprise ou d'une organisation
Le courrier externe	Les courriers reçus par une entreprise ou une organisation à partir d'une source extérieure
Le courrier de départ	Les courriers envoyés par une entreprise ou une organisation à une source extérieure.

Table 1 : Définitions des termes utilisés dans la SEL

Acronymes et abréviations

Acronymes et abréviations	Descriptions
SEL	Spécifications des exigences d'un logiciel
IEEE	Institute of Electrical and Electronics Engineering
IA	Intelligence artificielle
OCR	Reconnaissance Optique de Caractères.
SGBD	Système de gestion de base de données
IDE	Integrated development environment

Table 2 : Descriptions des acronymes et abréviations utilisés dans la SEL

2.2.1.4 Références (section 1.4 de la SEL)

IEEE. IEEE Std. 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

2.2.1.5 Vue d'ensemble (section 1.5 de la SEL)

Ce document est préparé pour expliquer tous les détails concernant la description globale de l'application à réaliser, les exigences fonctionnelles, non fonctionnelles et spécifiques, les données et la description du modèle comportemental du système.

Ce document est composé de trois parties :

- La deuxième partie contient les exigences spécifiques, les données et la description du modèle de comportement du système.
- La troisième et dernière section donne la planification du système, les conclusions et des informations supplémentaires.

2.2.2 Description générale (Chapitre 2 de la SEL) :

2.2.2.1 Environnement

Le système à réaliser est indépendant et parfaitement autonome.

2.2.2.2 Fonctions

Les besoins fonctionnels :

L'application doit assurer les fonctionnalités suivantes :

- Numériser les courriers.
- Archiver les courriers.
- Fournir des statistiques utiles à la prise de décision.
- Faire la distinction entre les courriers internes et externes.

L'application doit permettre au directeur ou toute autre personne chargée du suivi des courriers de :

- Consulter tous les courriers de l'entreprise.
- Suivre les états de traitement des courriers.
- Vérifier les réponses rédigées.
- Distribuer les courriers automatiquement.
- Rédiger des réponses facilement à l'aide des modèles prédéfinis.
- Filtrer les courriers.

L'application doit permettre au chef service de :

- Consulter les courriers : chaque chef de service ne peut consulter que les courriers qui sont affectés à son service.
- Rédiger des réponses facilement à l'aide des modèles prédéfinis.
- Modifiez le statut du courrier après l'avoir traité.

2.2.2.3 Caractéristiques des utilisateurs

Pour cette application, il existe trois classes d'utilisateurs, chacune ayant un niveau de privilège.

Utilisateur	Niveau d'instruction	Description du rôle et objectifs
Directeur	Moyen	<ul style="list-style-type: none"> - Ajouter des courriers - Envoyer des courriers - Consulter des courriers - Vérifier l'état du traitement de chaque courrier - Consulter les réponses aux courriers - Rédiger des courriers / réponses - Créer des modèles de réponses - Imprimer des courriers - Archiver des courriers - Envoyer des notes - Créer des comptes - Supprimer des comptes - Créer des groupes - Supprimer des groupes - Ajoutés des employés aux groupes - Voir les statistiques
Responsable	Moyen	<ul style="list-style-type: none"> - Ajouter des courriers - Envoyer des courriers - Consulter des courriers - Vérifier l'état du traitement de chaque courrier - Consulter les réponses aux courriers - Rédiger des courriers / réponses - Imprimer des courriers - Archiver des courriers - Envoyer des notes - Créer des modèles de réponses - Créer des groupes - Supprimer des groupes - Ajoutés des employés aux groupes

Employé	Faible Moyen	<ul style="list-style-type: none"> - Ajouter des courriers - Envoyer des courriers - Consulter des courriers - Vérifier l'état du traitement de chaque courrier - Rédiger des courriers / réponses - Imprimer des courriers - Archiver des courriers - Envoyer des notes
---------	-----------------	--

Table 3 : Liste des utilisateurs de l'application et leurs rôles

2.2.2.4 Contraintes

Limites imposées par le matériel :

- Le système nécessite d'avoir accès à une imprimante et à un scanner pour pouvoir numériser le courrier entrant et imprimer le courrier sortant.
- Pour accéder à l'application, une connexion internet est indispensable.

Considérations relatives à la sécurité et à la sûreté :

- Un compte doit être absolument créé par l'utilisateur « Directeur » du système.
- Pour avoir accès au système, l'utilisateur doit avoir une adresse mail valide, et un mot de passe valide.
- L'accès aux fonctionnalités se fera par les utilisateurs dont les droits d'accès de son type de compte.

2.2.2.5 Hypothèses et dépendances

La solution proposée repose sur la disponibilité permanente d'une connexion internet ainsi que sur le bon fonctionnement du serveur de données. L'impossibilité de garantir l'un des deux auraient comme conséquence l'obligation de la modification de la SEL donnée dans ce document en vue de s'adapter aux nouvelles exigences/contraintes.

2.2.3 Exigences spécifiques (Chapitre 3 de la SEL)

2.2.3.1 Exigences des interfaces externes

Cette partie devrait compléter la description des interfaces mentionnée dans la partie 2.2 (description générale) et vu que notre application est indépendante et parfaitement autonome nous n'avons aucune exigence sur les interfaces externes.

2.2.3.2 Exigences fonctionnelles

Cette section décrit les exigences fonctionnelles de l'application GEKOUR. Cette application web permet aux utilisateurs de :

2.2.3.2.1 Directeur/Administrateur

- Ajouter des courriers

L'application doit permettre au directeur d'ajouter les courriers à l'application, soit des courriers postaux ou bien des courriers électroniques reçus par mail.

- Envoyer des courriers

L'application doit permettre au directeur d'envoyer les courriers aux autres utilisateurs de l'application.

- Consulter les courriers

L'application doit permettre au directeur de voir les courriers qu'il a reçus ainsi que tous les courriers.

- Vérifier l'état du traitement de chaque courrier

L'application doit permettre au directeur de vérifier l'état du traitement de chaque courrier (ajouté / reçu / vu / en cours de traitement / réponse envoyée / archivé).

- Consulter les réponses aux courriers

L'application doit permettre au directeur de consulter les réponses rédigées par les responsables/ employés.

- Rédiger des réponses

L'application doit permettre au directeur de répondre aux courriers, soit en utilisant des modèles de réponse enregistrés, soit en écrivant une nouvelle réponse.

- Créer des modèles de réponses

L'application doit permettre au directeur de créer des modèles pour répondre aux courriers et de les enregistrer pour une utilisation ultérieure.

- Imprimer des réponses

L'application doit permettre au directeur d'imprimer des courriers.

- Archiver les courriers

L'application doit permettre au directeur d'archiver tous les courriers.

- Envoyer des notes aux employés

L'application doit permettre au directeur d'envoyer des notes/petits messages aux autres utilisateurs de l'application.

- Créer des comptes

L'application doit permettre au directeur de créer les comptes pour les utilisateurs en choisissant le type de compte (employé / responsable / directeur).

- Supprimer des comptes

L'application doit permettre au directeur de supprimer un compte d'un utilisateur.

L'application doit afficher une petite fenêtre de confirmation de suppression quand le directeur supprime un compte.

- Créer un groupe

L'application doit permettre au directeur de créer des groupes et des sous-groupes.

- Supprimer un groupe

L'application doit permettre au directeur de supprimer un groupe ou un sous-groupe.

L'application doit afficher une petite fenêtre de confirmation de suppression quand le directeur supprime un groupe.

- Ajouter des employés aux groupes

L'application doit permettre au directeur d'ajouter un employé ou un responsable à un groupe ou un sous-groupe.

- Voir les statistiques

L'application doit permettre au directeur de voir les statistiques :

- Nombre de courriers reçus par jour.
- Nombre de courriers traités par jour (ainsi par semaine et par mois)
- La durée du traitement en moyenne des courriers selon chaque groupe ((par jour, par semaine, et par mois)
- Nombre de courriers reçus par chaque service

2.2.3.2.2 Responsable

- Ajouter des courriers

L'application doit permettre au responsable d'ajouter les courriers à l'application, soit des courriers postaux ou bien des courriers électroniques reçus par mail.

- Envoyer des courriers

L'application doit permettre au responsable d'envoyer les courriers aux autres utilisateurs de l'application.

- Consulter les courriers

L'application doit permettre au responsable de voir les courriers qu'il a reçus ainsi que tous les courriers de son groupe.

- Vérifier l'état du traitement de chaque courrier

L'application doit permettre au responsable de vérifier l'état du traitement de chaque courrier de son groupe (ajouté / reçu / vu / en cours de traitement / réponse envoyée / archivé).

- Consulter les réponses aux courriers

L'application doit permettre au responsable de consulter les réponses rédigées par les employés de son groupe.

- Rédiger des réponses

L'application doit permettre au responsable de répondre aux courriers, soit en utilisant des modèles de réponse enregistrés, soit en écrivant une nouvelle réponse.

- Créer des modèles de réponses

L'application doit permettre au responsable de créer des modèles pour répondre aux courriers et de les enregistrer pour une utilisation ultérieure.

- Imprimer des réponses

L'application doit permettre au responsable d'imprimer des courriers.

- Archiver les courriers

L'application doit permettre au responsable d'archiver des courriers.

- Envoyer des notes aux employés

L'application doit permettre au responsable d'envoyer des notes/petits messages aux autres utilisateurs de l'application.

- Créer un groupe

L'application doit permettre au responsable de créer des sous-groupes dans son groupe.

- Supprimer un groupe

L'application doit permettre au responsable de supprimer un sous-groupe de son groupe.

L'application doit afficher une petite fenêtre de confirmation de suppression quand le responsable supprime un sous-groupe.

- Ajouter des employés aux groupes

L'application doit permettre au responsable d'ajouter un employé ou un responsable à un groupe ou un sous-groupe.

2.2.3.2.3 Employé

- Ajouter des courriers

L'application doit permettre à l'employé d'ajouter les courriers à l'application, soit des courriers postaux ou bien des courriers électroniques reçus par mail.

- Envoyer des courriers

L'application doit permettre à l'employé d'envoyer les courriers aux autres utilisateurs de l'application.

- Consulter les courriers

L'application doit permettre à l'employé de voir les courriers qu'il a reçus.

- Vérifier l'état du traitement de chaque courrier

L'application doit permettre à l'employé de vérifier l'état du traitement de chaque courrier qu'il a envoyé à un autre utilisateur de l'application (reçu / vu / en cours de traitement / réponse envoyée / archivé).

- Rédiger des réponses

L'application doit permettre à l'employé de répondre aux courriers, soit en utilisant des modèles de réponse enregistrés, soit en écrivant une nouvelle réponse.

- Imprimer des réponses

L'application doit permettre à l'employé d'imprimer des courriers.

- Archiver les courriers

L'application doit permettre à l'employé d'archiver des courriers.

- Envoyer des notes aux employés

L'application doit permettre à l'employé d'envoyer des notes/petits messages aux autres utilisateurs de l'application.

2.2.3.3 Exigences de performance

- Le temps de réponse pour effectuer une tâche donnée doit être faible.
- La capacité de supporter plusieurs utilisateurs concurrents sans dégrader la performance d'application.
- L'utilisation des ressources informatiques (mémoire, stockage) doit être minimisée.

2.2.3.4 Contraintes de conception

- Base de données :

La base de données doit être sécurisée pour empêcher tout accès par des personnes non autorisées aux données enregistrées.

L'application doit être compatible avec les navigateurs web les plus couramment utilisés.

- Interfaces graphiques :
 - L'application doit s'adapter aux différentes tailles d'écran (ordinateurs de bureau, tablettes, smartphones, etc.).
 - L'interface doit être simple et intuitive, afin que les utilisateurs puissent facilement l'utiliser, et la navigation doit être claire.

2.2.3.5 Attributs

- Sécurité :

L'application doit être développée conformément aux normes de sécurité en vigueur et mettre en place des mesures de sécurité telles que l'authentification des utilisateurs, et la gestion des autorisations et des privilèges ainsi que le cryptage des données.

- Performance :

L'application doit être rapide et réactive pour fournir une expérience utilisateur satisfaisante.

Les requêtes doivent être traitées rapidement.

- Accessibilité :

L'application web doit être accessible aux utilisateurs, quel que soit leur localisation ou leur dispositif de connexion.

- Évolutivité :

Le site doit être conçu pour être évolutif, capable de gérer un grand nombre d'utilisateurs et de données sans ralentir ou échouer.

2.2.3.6 Index

Spécifications des exigences de notre logiciel

- 2.1 Introduction (Chapitre 1 de la SEL)
 - 2.1.1 Objet (section 1.1 de la SEL)
 - 2.1.2 Portée (section 1.2 de la SEL)
 - 2.1.3 Définitions (section 1.3 de la SEL)
 - 2.1.4 Références (section 1.4 de la SEL)
 - 2.1.5 Vue d'ensemble (section 1.5 de la SEL)

- 2.2 Description générale (Chapitre 2 de la SEL)
 - 2.2.1 Environnement
 - 2.2.2 Fonctions
 - 2.2.3 Caractéristiques des utilisateurs
 - 2.2.4 Contraintes

- 2.3 Exigences spécifiques (Chapitre 3 de la SEL)
 - 2.3.2 Exigences fonctionnelles
 - 2.3.1.1 Agent
 - 2.3.1.2 Directeur
 - 2.3.1.3 Responsable
 - 2.3.1.4 Employé
 - 2.3.3 Exigences de performance
 - 2.3.4 Contraintes de conception

2.3 Conclusion

Ce chapitre était la phase de pré-étude qui est la première phase de la méthode RUP, nous avons fourni une spécification des exigences pour notre application où nous avons identifié les parties prenantes et les utilisateurs et déterminé leurs besoins, ainsi que les contraintes de conception.

En conclusion, la phase de pré-étude nous a permis d'obtenir une vision claire et détaillée du projet et nous a fourni une base solide pour les phases ultérieures de développement, et grâce à cette analyse approfondie, nous sommes maintenant prêts à passer à la prochaine phase : la phase d'élaboration, qui sera détaillé dans le chapitre suivant.

Chapitre 3

Conception

3.1 Introduction

Dans ce chapitre nous allons mettre en évidence le côté conceptuel de notre application qui constitue une étape fondamentale qui précède l'implémentation, et c'est la deuxième phase de la méthode RUP : la phase d'Élaboration.

Ce chapitre servira de guide un peu détaillé pour la mise en œuvre de la conception de notre application, en fournissant des informations essentielles sur les choix conceptuels, les choix architecturaux, et les modèles de données. Et pour ce faire nous avons choisi d'utiliser le modèle de Kruchten qui, avec ses perspectives architecturales, offre une approche structurée pour analyser et concevoir l'architecture logicielle de notre application. Il permet de décomposer les différents aspects de l'architecture et de les voir sous différents angles.

3.2 Modèle de Kruchten

Le modèle de Kruchten ou le modèle 4+1 vues, également connu sous le nom de modèle architectural 4+1, est une approche de modélisation logicielle utilisée dans le domaine de l'ingénierie logicielle pour la conception et le développement de systèmes logiciels complexes. [y]

Le modèle 4+1 vues a été proposé par Philippe Kruchten dans les années 1990 et est basé sur le modèle d'architecture logicielle en couches. Il vise à fournir une représentation complète et holistique d'un système logiciel en utilisant cinq perspectives ou "vues" différentes :

- La vue logique.
- La vue de réalisation.
- La vue des processus.
- La vue de déploiement.
- La vue des cas d'utilisation.

Chacune de ces vues représente un aspect spécifique du système et permet aux parties prenantes de comprendre et de communiquer différents aspects de la conception logicielle.

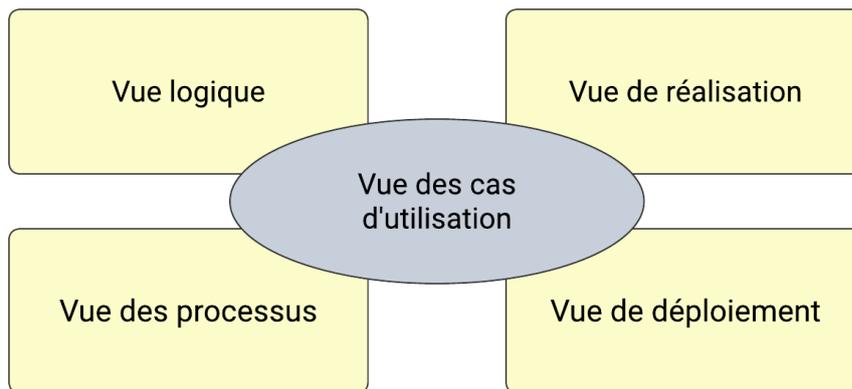


Figure 14 : modèle 4+1 vues

L'utilisation de vues permet de traiter séparément les intérêts des divers groupes d'intervenants et ainsi de mieux séparer les préoccupations fonctionnelles et les préoccupations extra fonctionnelles.

3.2.1 Vue des cas d'utilisation

Cette vue se concentre sur la cohérence en présentant des scénarios d'utilisation qui mettent en œuvre les éléments des quatre autres vues. Les scénarios sont une abstraction des exigences fonctionnelles de l'application. Cette vue valide en quelque sorte les autres vues et assure la cohérence globale. C'est aussi cette vue qui est considérée comme la dernière vue qui est construite en premier, juste après l'établissement du cahier des charges, pour fixer les contours du système à réaliser avec ses fonctionnalités appelées, dans la terminologie UML, des cas d'utilisation. [15]

a. Les acteurs

Un acteur représente un rôle joué par une entité externe, telle qu'un utilisateur humain, un dispositif matériel ou un autre système, qui interagit directement avec le système logiciel. Ces acteurs ont la capacité de consulter et/ou de modifier l'état du système.

Identification des acteurs qui interagissent avec GEKOUR :

- Utilisateurs humains directs :
 - Employé
 - Responsable (est une sorte de "Employé")
 - Directeur (est une sorte de "Responsable")
- Systèmes connexes qui interagissent directement avec le système :
 - Système d'information de l'entreprise cliente

b. Diagramme de contexte statique

Le diagramme de contexte est l'un des premiers diagrammes pouvant être réalisés afin de décrire à haut niveau le futur système en mode boîte noire. Il permet de se concentrer sur l'identification des différents acteurs interagissant avec lui.

La réalisation de ce diagramme permet de s'assurer que le contexte du système a bien été saisi avant d'aller plus de l'avant. [16]

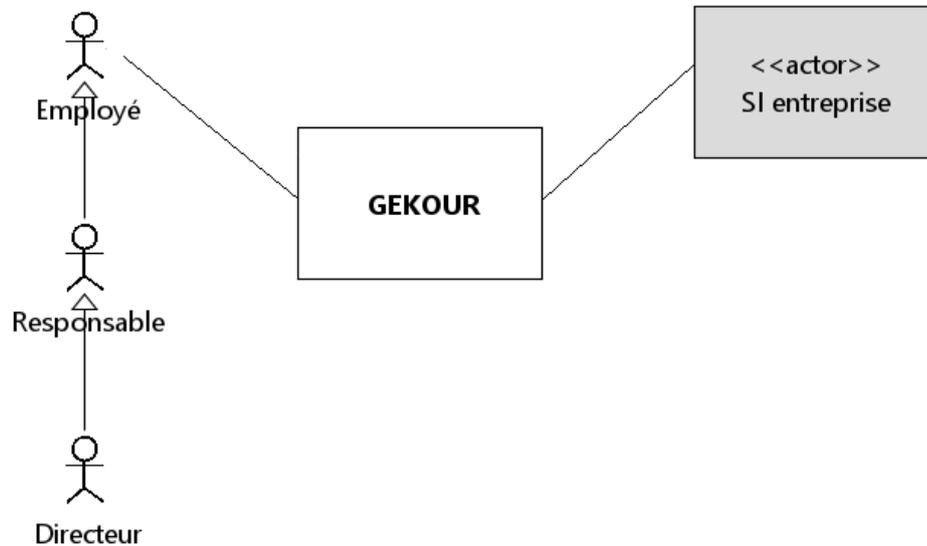


Figure 15 : Diagramme de contexte statique

c. Cas d'utilisation

i. Les cas d'utilisation de GEKOUR, par acteur :

- Employé :
 - S'authentifier
 - Voir ses informations (Changer ses informations)
 - Se déconnecter
 - Rédiger un courrier (Envoyer un courrier, Imprimer un courrier)
 - Voir les courriers reçus (Envoyer un courrier, Archiver un courrier, Rédiger un courrier)
 - Ajouter un courrier
 - Voir l'archive
 - Envoyer une note
 - Voir les notes reçues
 - Voir les notes envoyées
 - Voir l'état de traitement des courriers
- Responsable :
 - Créer un groupe
 - Supprimer un groupe
 - Voir la liste des employés

- Ajouter un employé a un groupe
- Ajouter un modèle d'un courrier
- Directeur :
 - Créer un nouveau compte
 - Modifier les informations d'un compte
 - Supprimer un compte
 - Créer un groupe
 - Supprimer un groupe
 - Voir les statistiques
- Système d'information d'entreprise (SI entreprise) :
 - Néant

ii. Diagramme des cas d'utilisation :

Un diagramme de cas d'utilisation est un type de diagramme utilisé en UML. Il décrit l'interaction des acteurs externes avec le système en cours de conception. Il ne montre pas beaucoup de détails, mais résume seulement certaines des relations entre les cas d'utilisation, les acteurs et les systèmes.

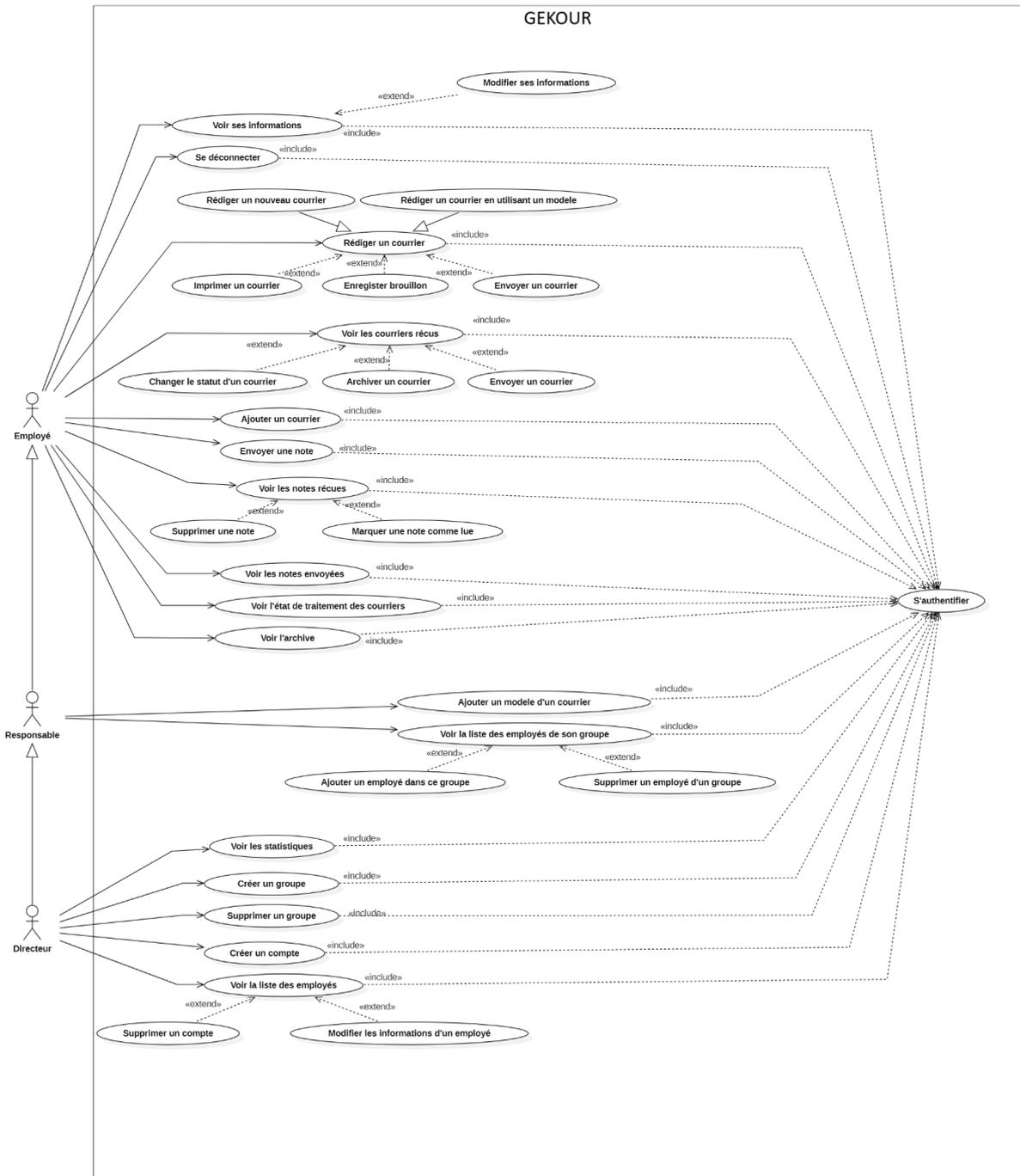


Figure 16 : Diagramme des cas d'utilisation de GEKOUR

b. Diagramme de packages :

Un diagramme de packages est un type de diagramme utilisé en UML, il représente la structure modulaire d'un système logiciel. Il est utilisé pour organiser et regrouper les éléments logiciels connexes, tels que les classes, les interfaces, les packages ou les composants, afin de montrer comment ils interagissent.

→ Étant donné que le système est relativement simple et n'a pas de structure modulaire complexe, nous n'avons pas eu besoin de concevoir un diagramme de package et de l'inclure dans cette partie.

3.2.3 Vue de réalisation

Cette vue se concentre sur les aspects techniques de l'implémentation du système. Elle identifie les modules, les composants, les bibliothèques et les technologies utilisés dans la construction du système logiciel.

Dans le cadre de notre étude, nous avons pris la décision de négliger la vue de réalisation pour notre application. Cette décision est motivée par deux facteurs principaux : un haut niveau d'abstraction et des contraintes de ressources.

En adoptant un niveau d'abstraction élevé, notre objectif principal est de fournir une vision globale de l'architecture de l'application en mettant l'accent sur les fonctionnalités principales et les besoins des utilisateurs. En négligeant la vue de réalisation, nous sommes en mesure de présenter une description claire et concise des fonctions essentielles du système, sans nous encombrer des détails spécifiques de mise en œuvre.

Il est important de souligner que cette décision de négliger la vue de réalisation se justifie dans le contexte spécifique de notre étude, où le système est conçu comme un prototype en évolution avec l'ajout ultérieur de fonctionnalités et de composants supplémentaires.

3.2.4 Vue de déploiement

Cette vue se concentre sur la configuration matérielle et les contraintes du déploiement du système. Elle identifie les serveurs, les machines, les réseaux et les protocoles nécessaires pour déployer et exécuter le système logiciel.

Diagramme de déploiement :

Un diagramme de déploiement est un type de diagramme utilisé en UML qui détermine comment le logiciel est déployé sur le matériel sous-jacent ainsi que la spécification du matériel physique sur lequel le système logiciel s'exécutera. En termes simples, ce diagramme aide à spécifier les détails de la construction par la visualisation d'une caractéristique statique des nœuds physiques du logiciel et de leurs relations.

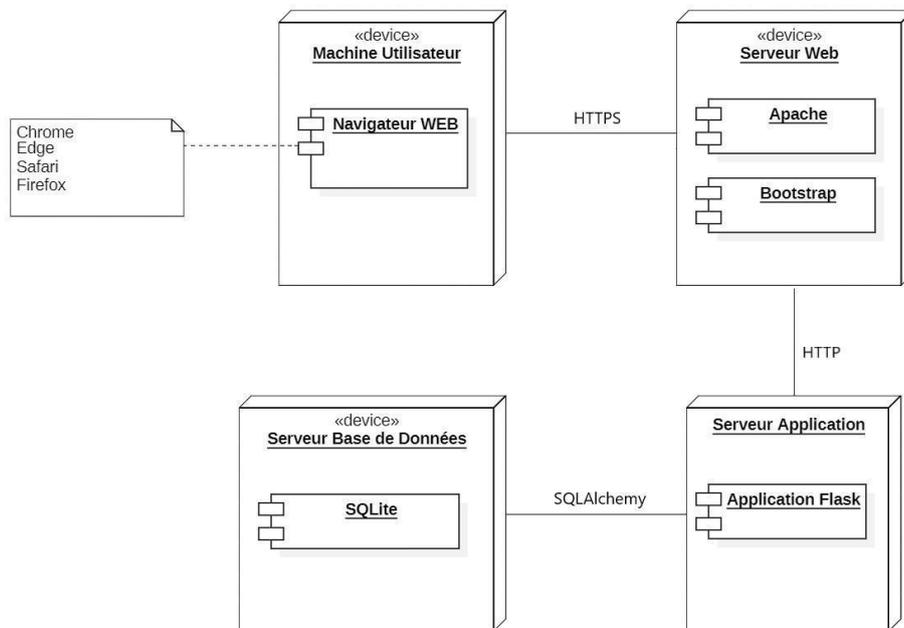


Figure 18 : Diagramme de déploiement

3.2.5 Vue des processus

Cette vue se concentre sur les aspects dynamiques du système logiciel. Elle représente les interactions et les collaborations entre les composants du système, les flux de données, les performances, la concurrence et d'autres aspects liés à l'exécution du système.

Les types de diagrammes qui peuvent être utilisés dans cette vue :

a. Diagramme d'activité :

Un diagramme d'activité est un type de diagramme utilisé en UML, il décrit le flux procédural des actions dans le cadre d'une activité. Il est utilisé pour modéliser la façon dont les activités sont coordonnées pour fournir un service, pour montrer les événements nécessaires à la réalisation d'une opération et pour illustrer les relations entre les événements dans un cas d'utilisation unique.

→ Nous avons privilégié l'utilisation des diagrammes de séquence plutôt que des diagrammes d'activité dans cette vue pour représenter l'aspect dynamique du système, car les diagrammes de séquence offrent un niveau de détail plus fin. Ils permettent de représenter spécifiquement les échanges de messages, les actions et les réponses entre les objets, et ils peuvent capturer des informations cruciales comme les paramètres des messages, les itérations et les conditions. En utilisant les diagrammes de séquence, nous pourrions mieux représenter et documenter les interactions dynamiques du système.

b. Diagramme de séquence :

Un diagramme de séquence est un type de diagramme d'interaction utilisé en UML, il détaille la manière dont les opérations sont effectuées et représente la séquence chronologique des messages échangés entre les objets ou les acteurs d'un système. Il permet de décrire l'interaction entre les différents composants d'un système logiciel ou matériel.

i. Les interactions de "Employé" avec le système

- S'authentifier pour la première fois : lors de la création d'un compte d'un employé, celui-ci sera informé par email qu'il peut se connecter avec son email et avec un mot de passe par défaut qu'il devra changer lors de sa première connexion.

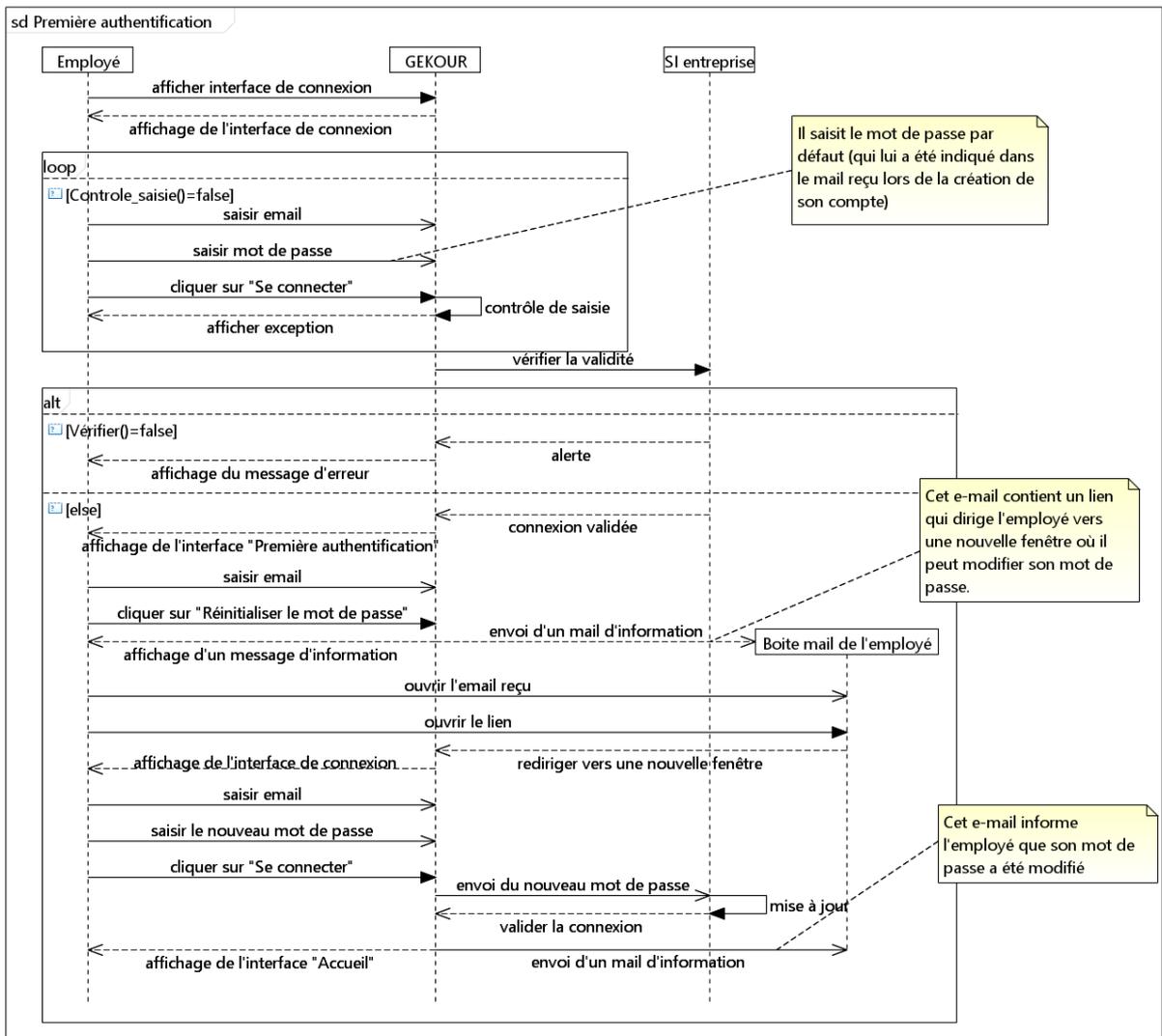


Figure 19 : Diagramme de séquence Première authentification

- S'authentifier : chaque fois que l'employé veut se connecter au système, il doit s'authentifier avec un email et un mot de passe valides.

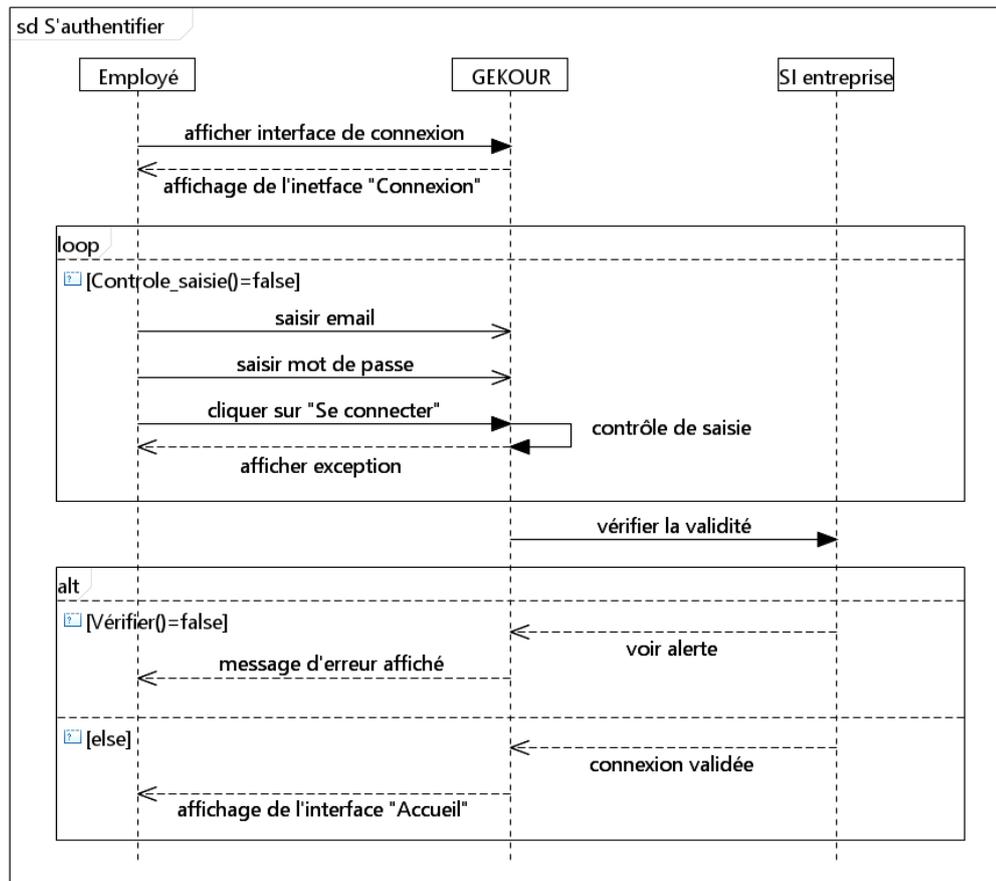


Figure 20 : Diagramme de séquence S'authentifier

L'identification du fragment s'authentifier permet d'alléger les diagrammes de séquence système en évitant la duplication et en mettant l'accent sur les aspects essentiels du système, ce qui facilite leur compréhension.

- Voir les courriers reçus : pour qu'un employé puisse voir les mails reçus, il doit s'authentifier avec un email et un mot de passe valides, après la validation de la connexion, l'interface d'accueil s'affiche directement, il pourra donc voir tout ce qu'il a reçu dans la table "Courrier reçu".

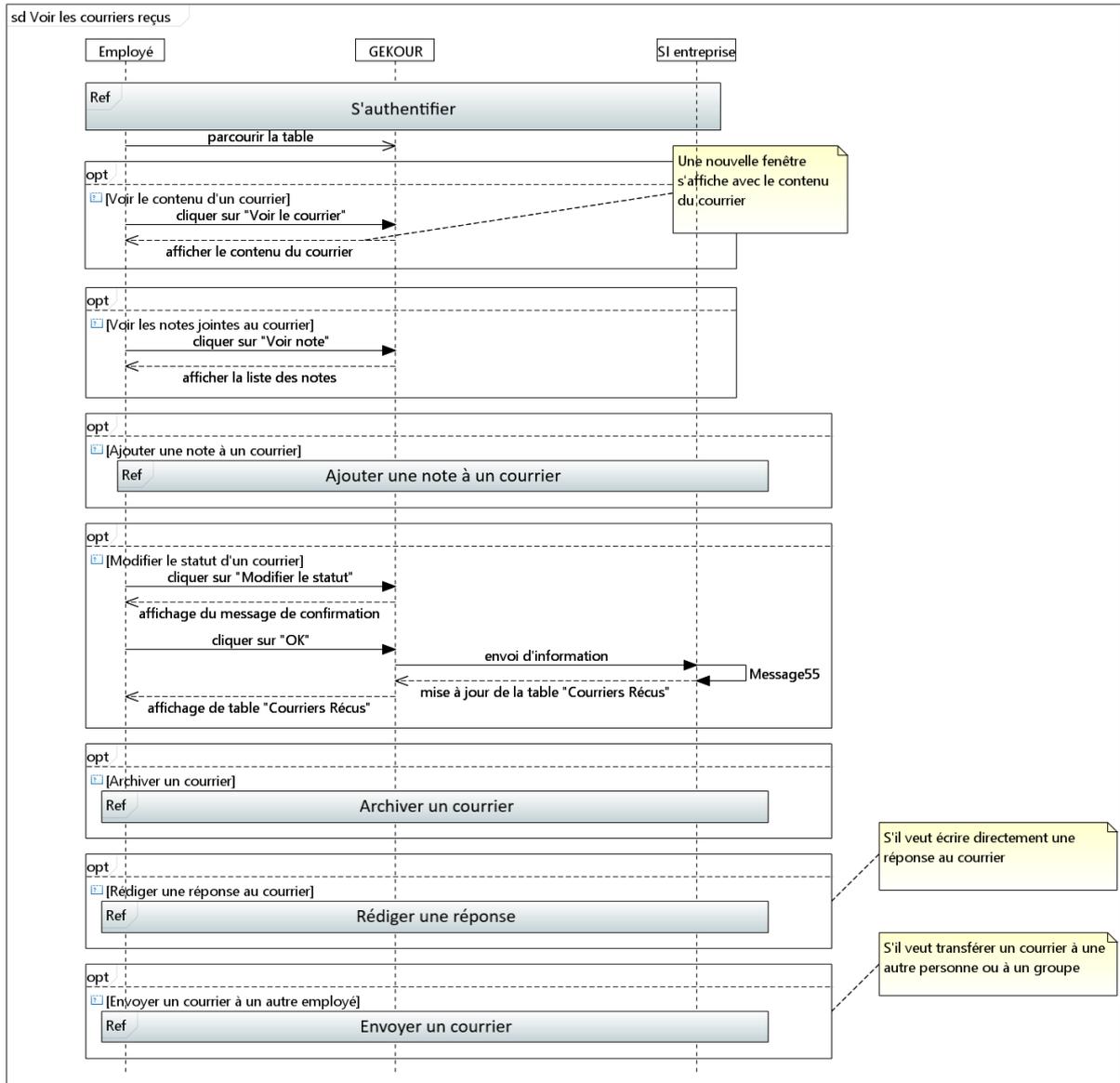


Figure 21 : Diagramme de séquence Voir les courriers reçus

Nous avons identifié les fragments “Ajouter une note à un courrier”, “Rédiger une réponse”, et “Archiver un courrier” pour alléger le diagramme et faciliter sa compréhension, et le fragment “Envoyer un courrier” pour le réutiliser dans les prochains diagrammes.

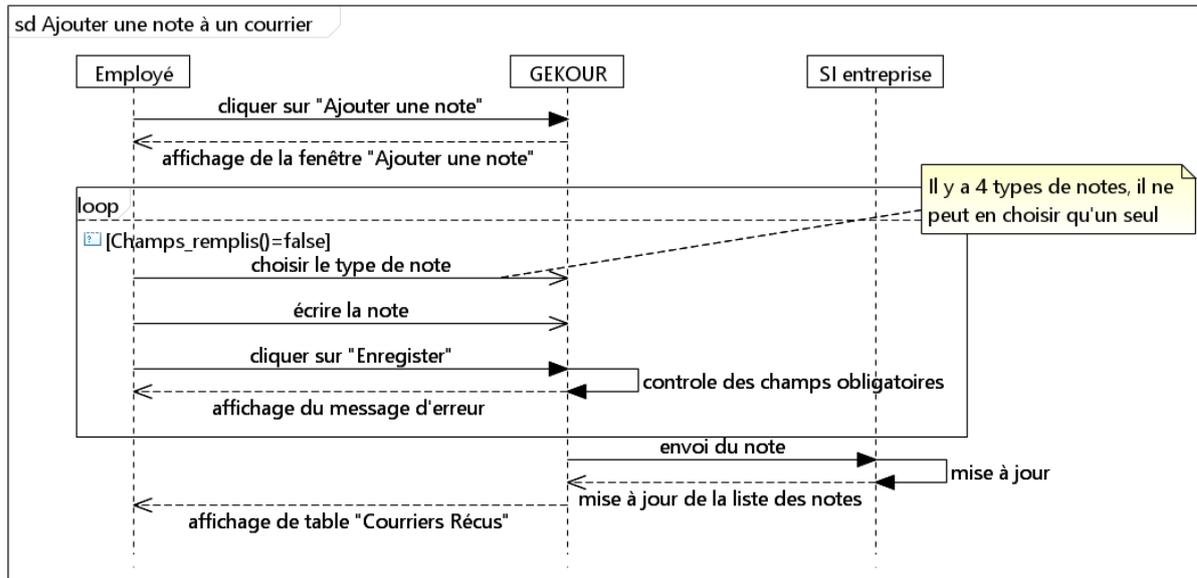


Figure 22 : Fragment Ajouter une note à un courrier

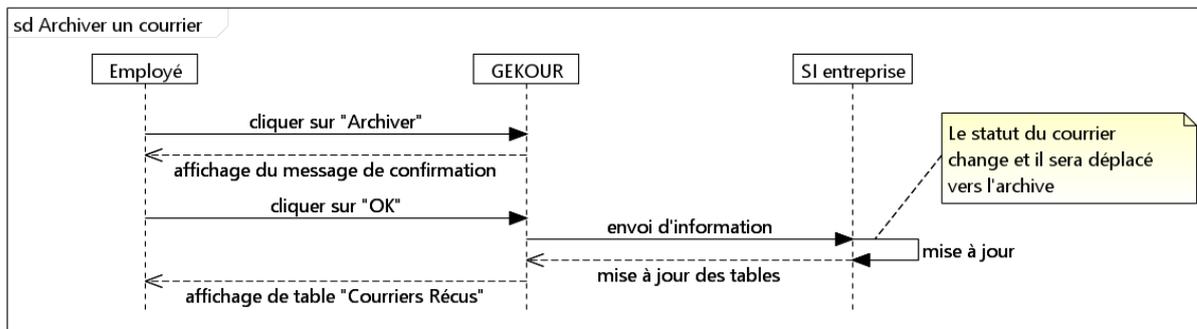


Figure 23 : Fragment Archiver un courrier

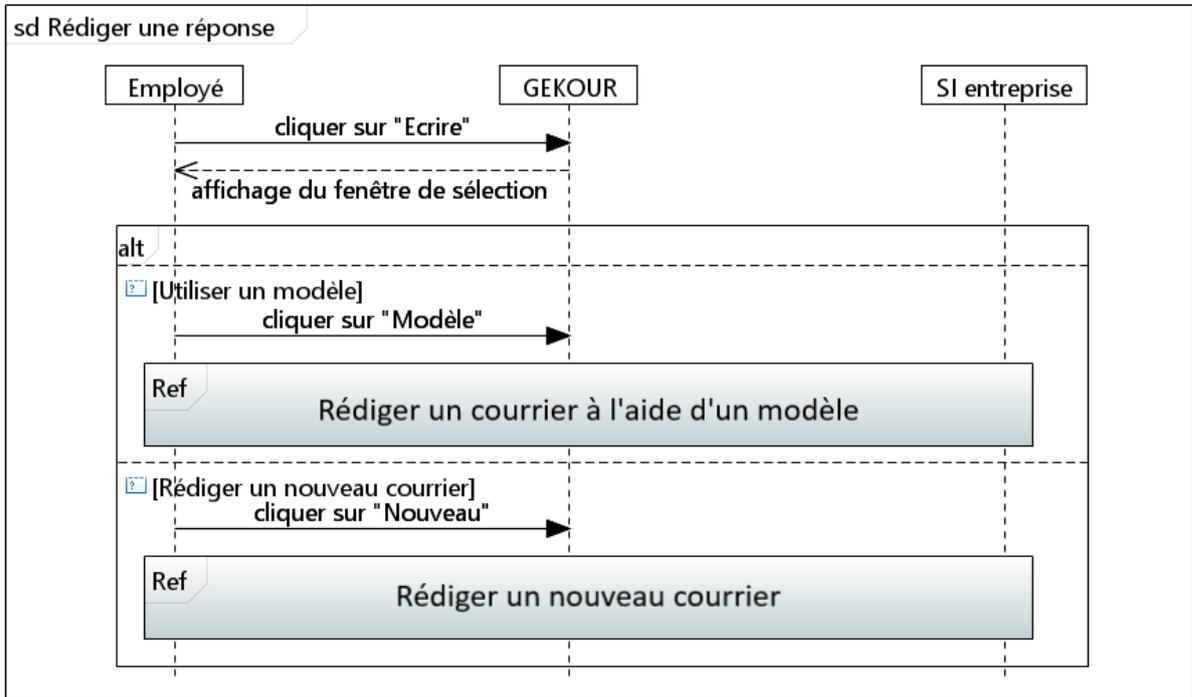


Figure 24 : Fragment Rédiger une réponse

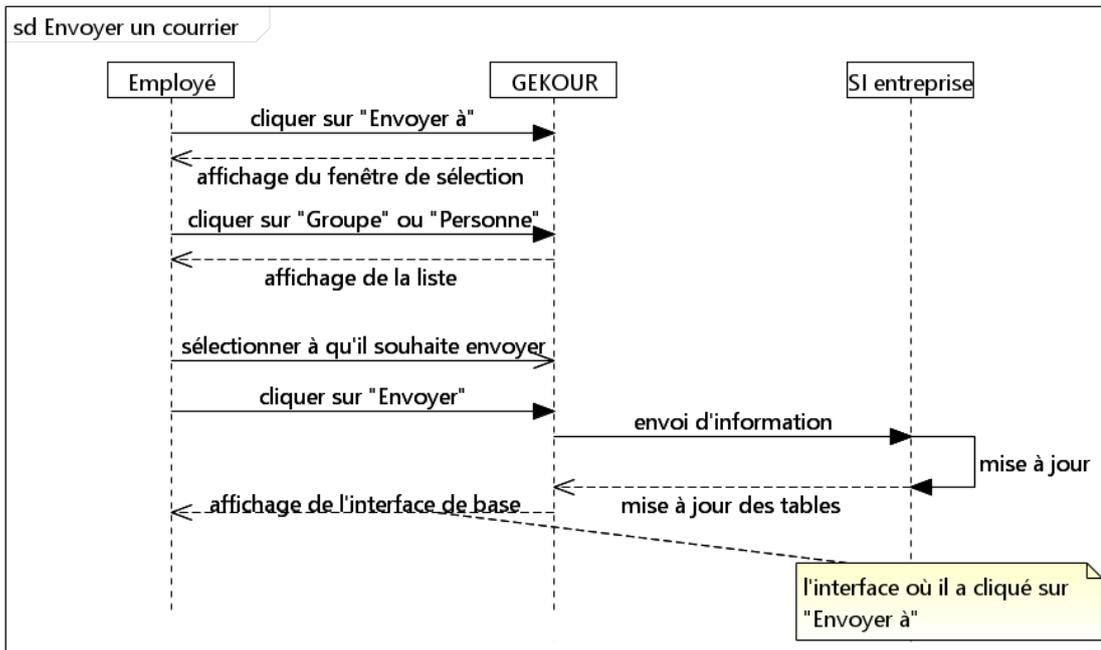


Figure 25 : Fragment Envoyer un courrier

- Rédiger un courrier : si un employé veut rédiger un courrier, il aura deux possibilités :
 - Rédiger un nouveau courrier.
 - Utiliser un modèle.

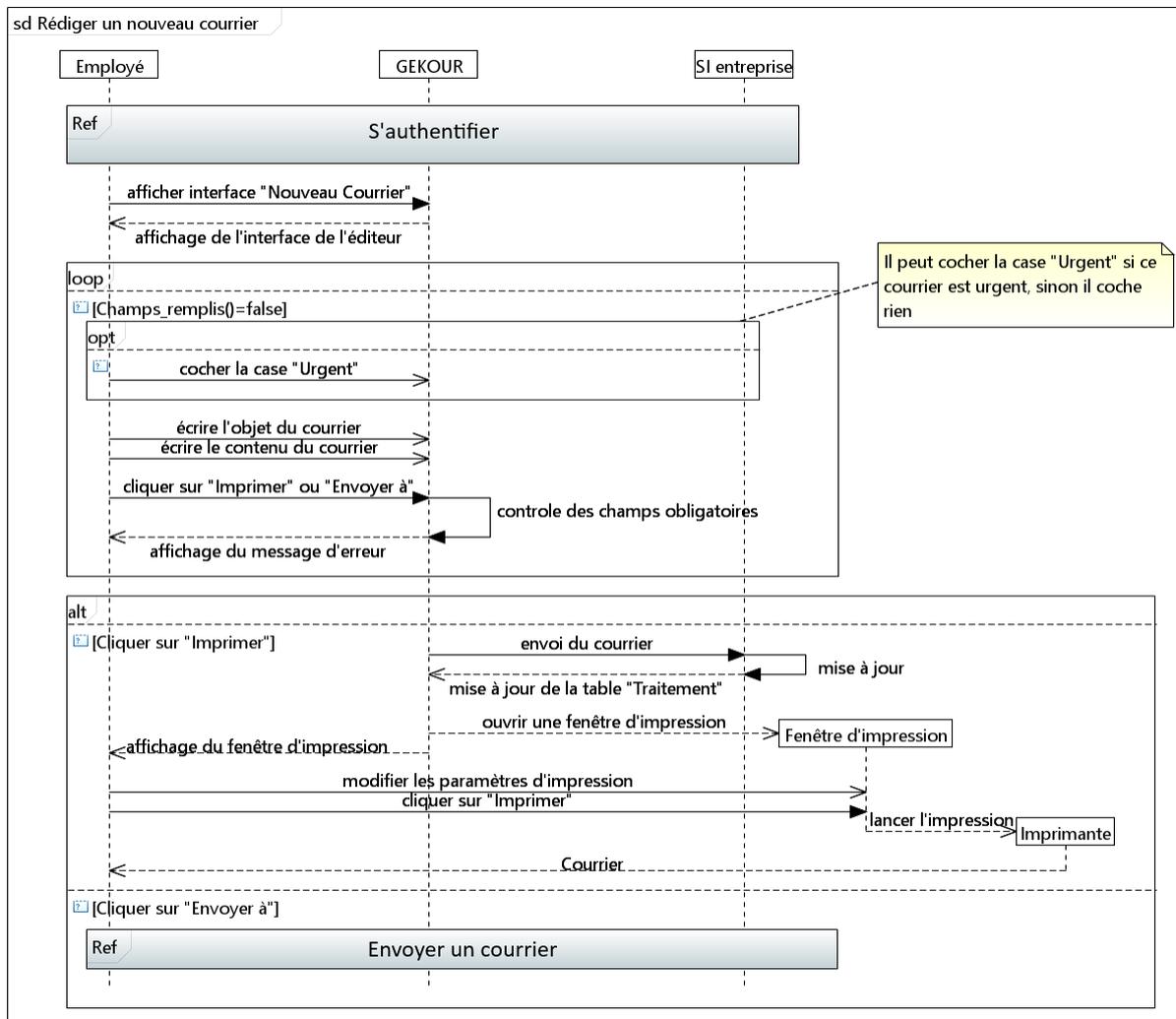


Figure 26 : Diagramme de séquence Rédiger un nouveau courrier

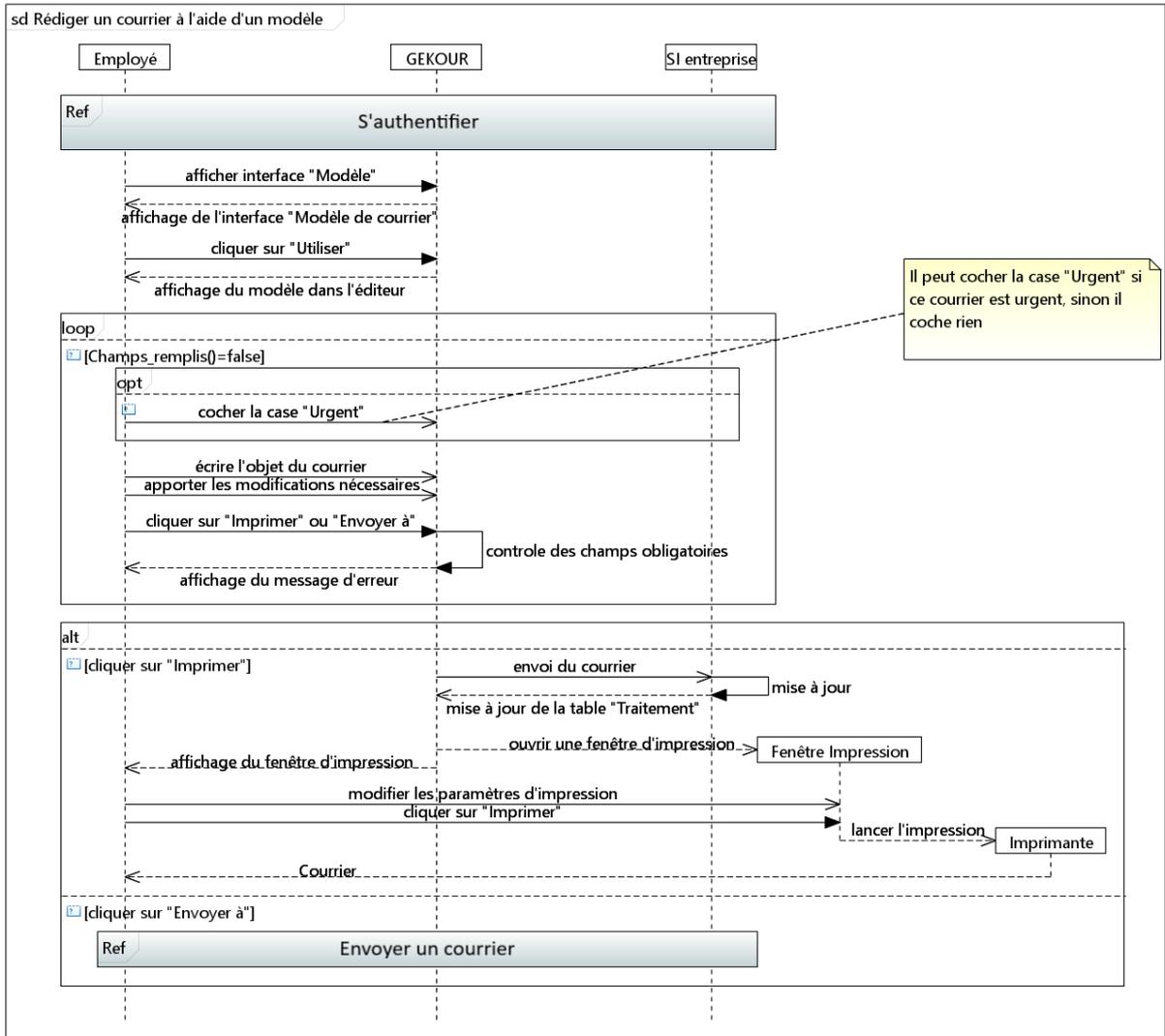


Figure 27 : Diagramme de séquence Rédiger un courrier à l'aide d'un modèle

- Ajouter un courrier : un employé peut ajouter un courrier à l'application, il peut ajouter des courriers postaux (après les avoir scannés) ou des courriers électroniques (si ces courriers comportent des pièces jointes il doit les télécharger au préalable).

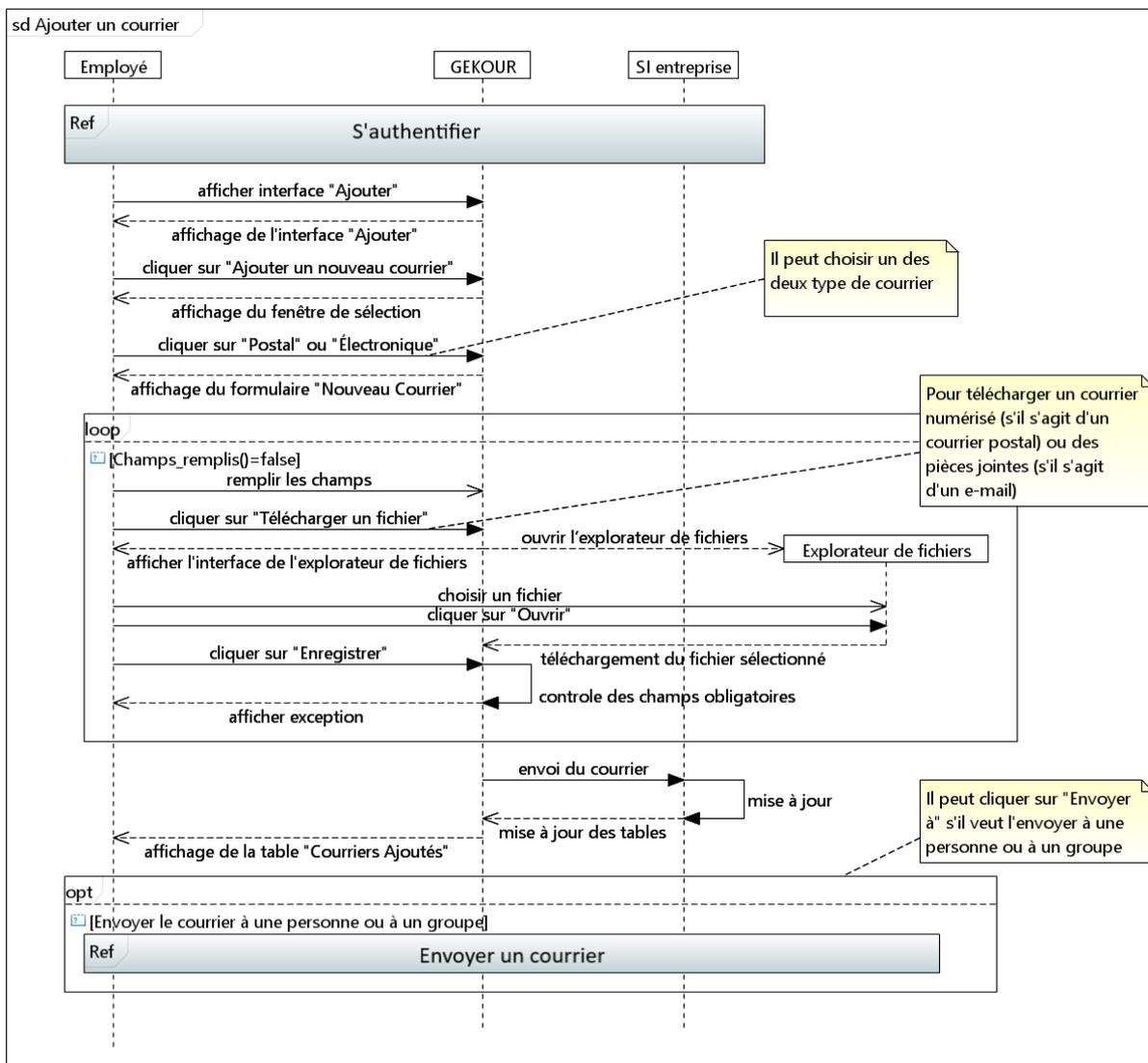


Figure 28 : Diagramme de séquence Ajouter un courrier

- Écrire une note : un employé a la possibilité d'écrire et d'envoyer des notes aux autres utilisateurs de l'application quel que soit leur type de compte (employé, responsable, ou directeur). Il existe quatre types de notes.

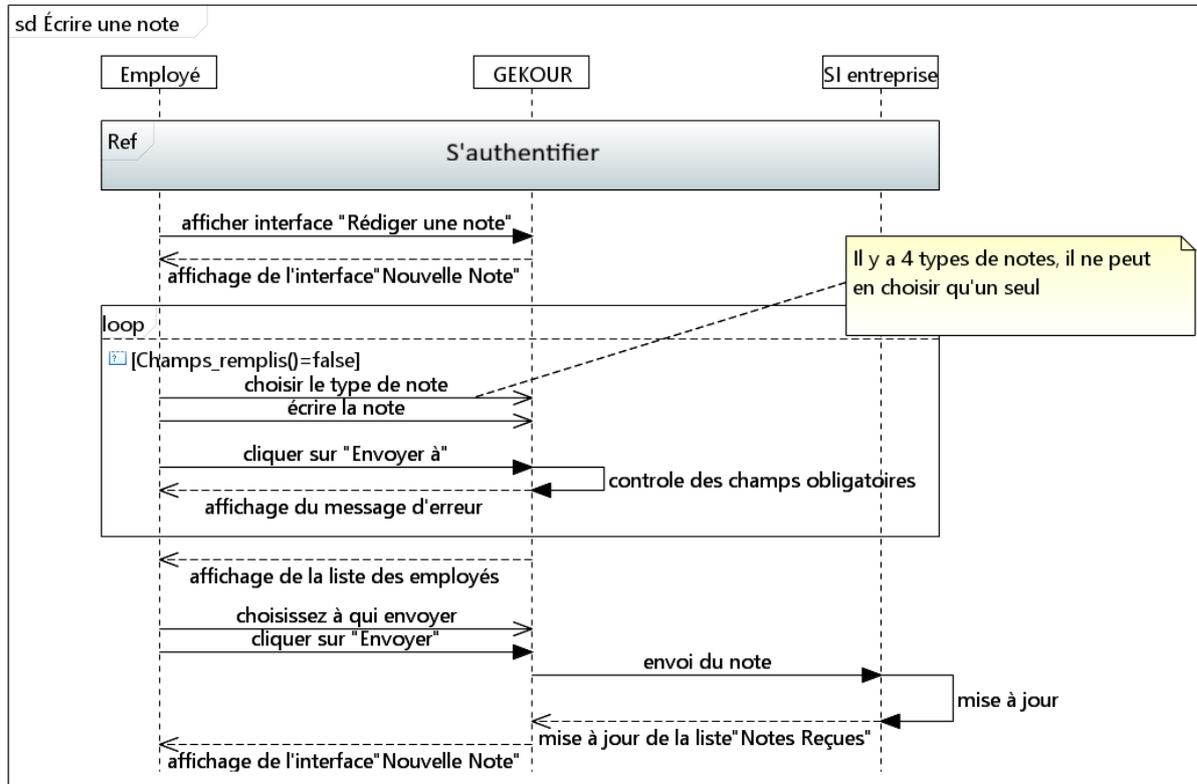


Figure 29 : Diagramme de séquence Écrire une note

- Voir les notes envoyées : un employé peut voir les notes qu'il a envoyées à d'autres employés et savoir si elles ont été lues.

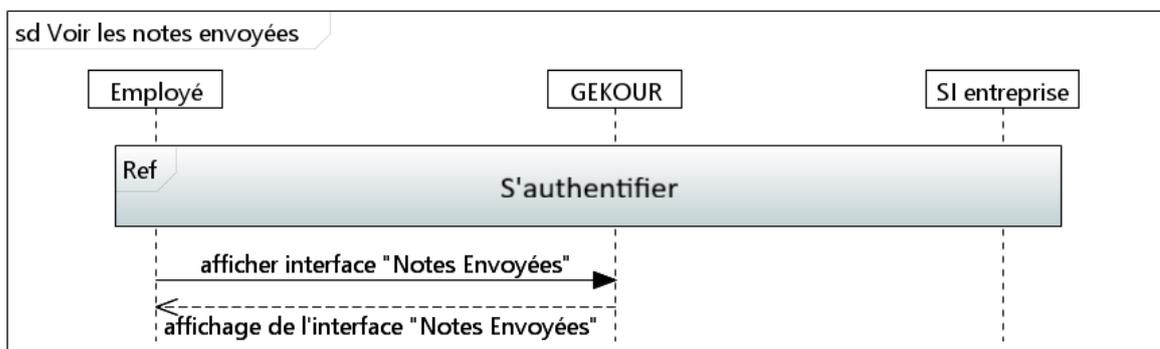


Figure 30 : Diagramme de séquence Voir les notes envoyées

- Voir les notes reçues : un employé peut voir les notes qu'il a reçues, il peut les marquer comme lues ou les supprimer.

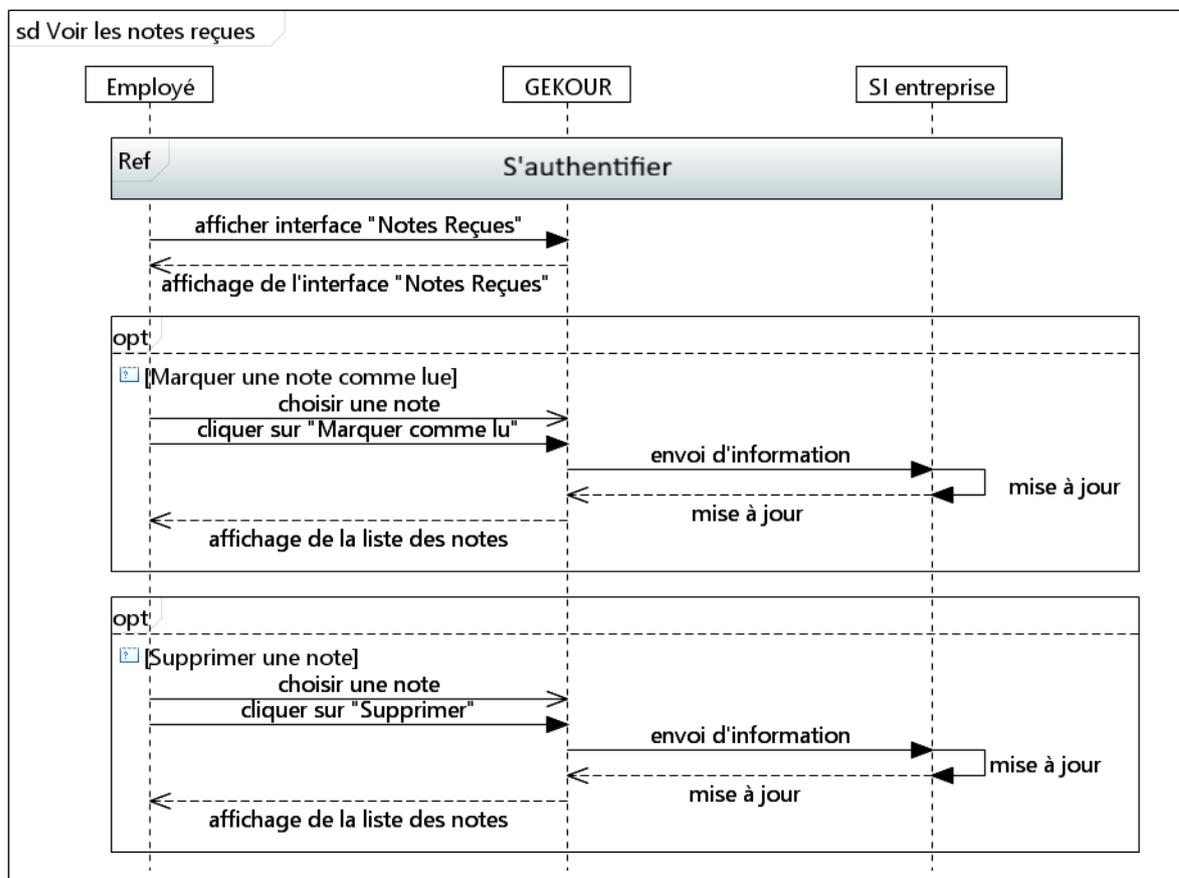


Figure 31 : Diagramme de séquence Voir les notes reçues

- Voir l'état de traitement : l'employé peut voir l'état de traitement des courrier qu'il a envoyé aux autres employés

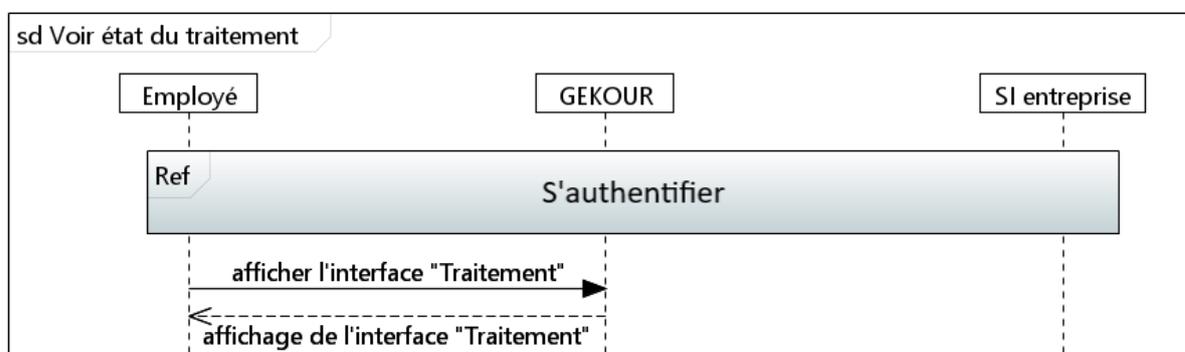


Figure 32 : Diagramme de séquence Voir l'état des traitements

- Voir l'archive : l'employé peut voir l'archive de ses courriers (courriers qu'il a écrits ou reçus).

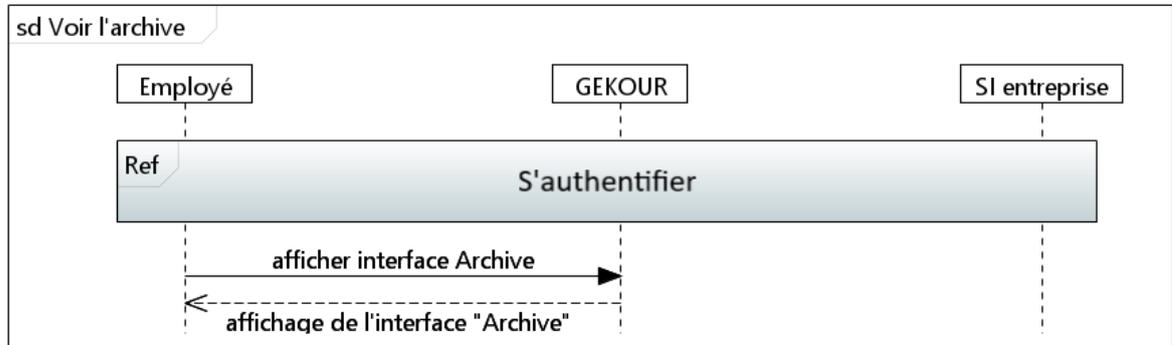


Figure 33 : Diagramme de séquence Voir l'archive

ii. Les interactions de “Responsable” avec le système

Puisque le responsable est aussi un employé, il peut faire tout ce qu'un employé peut faire, en plus d'effectuer d'autres actions et interactions.

- Ajouter un modèle : un responsable peut ajouter un modèle de courrier.

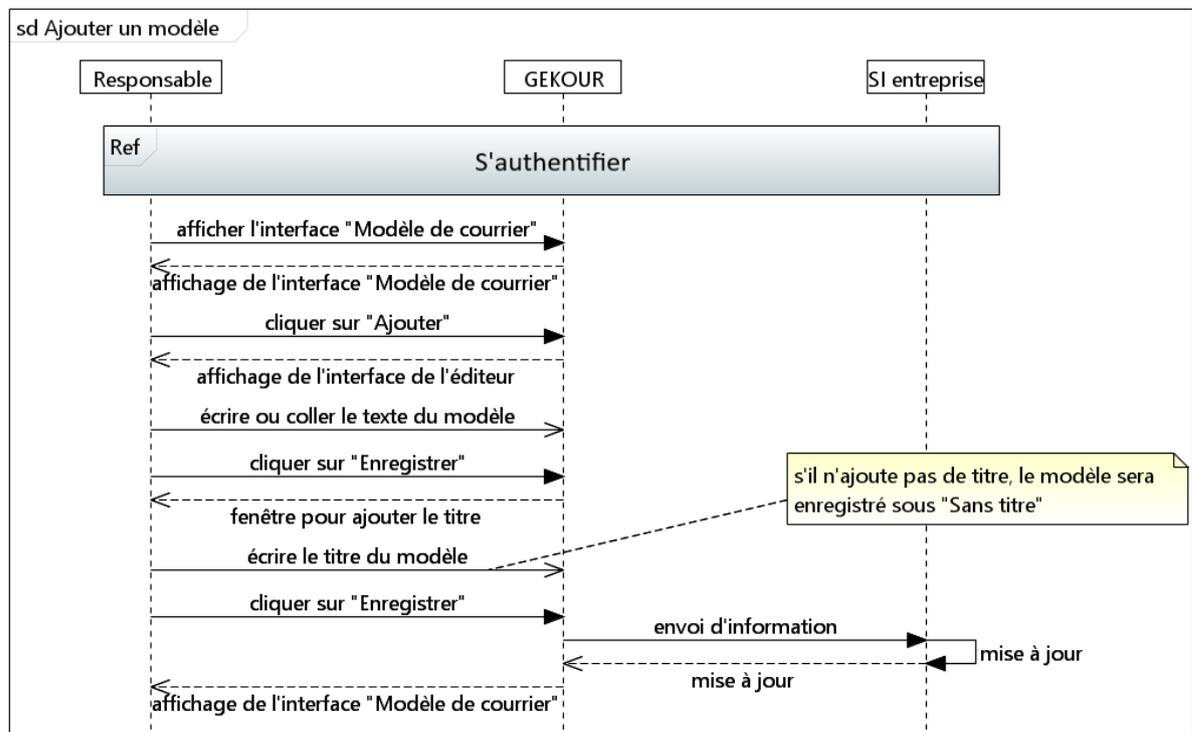


Figure 34 : Diagramme de séquence Ajouter un modèle

- Ajouter un employé dans un sous-groupe : un responsable peut ajouter un employé à un sous-groupe de son groupe.

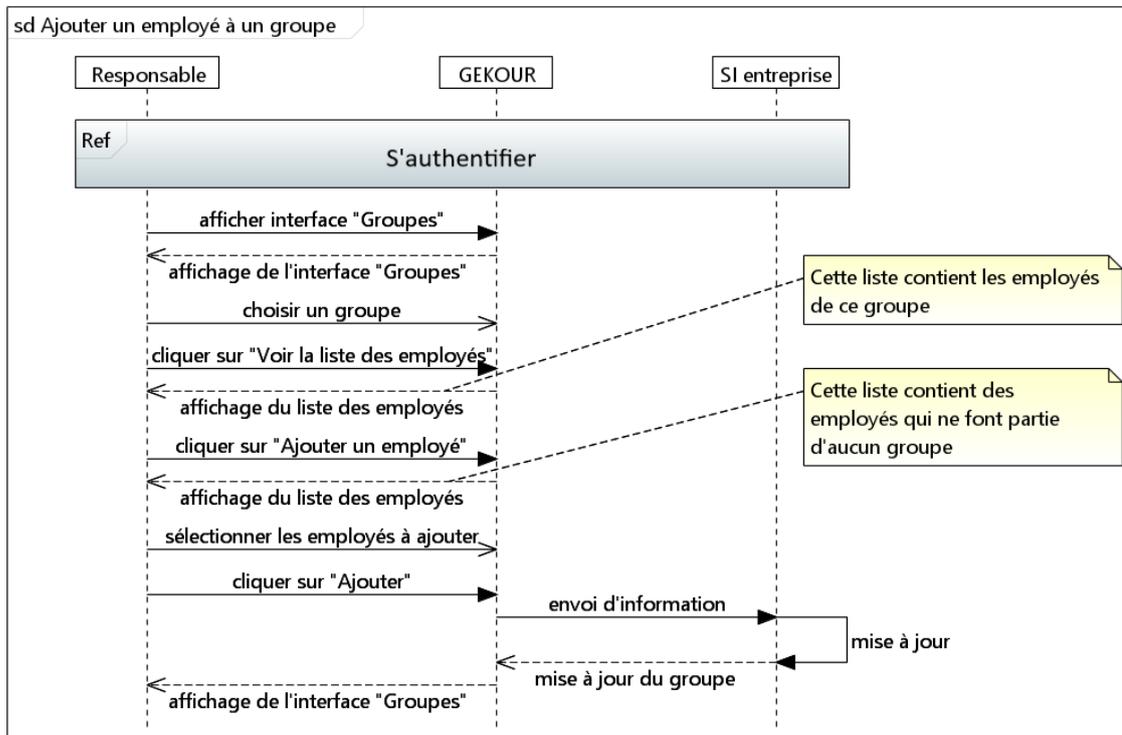


Figure 35 : Diagramme de séquence Ajouter un employé dans un sous-groupe

- Supprimer un employé d'un groupe : un responsable peut retirer un employé d'un sous-groupe de son groupe.

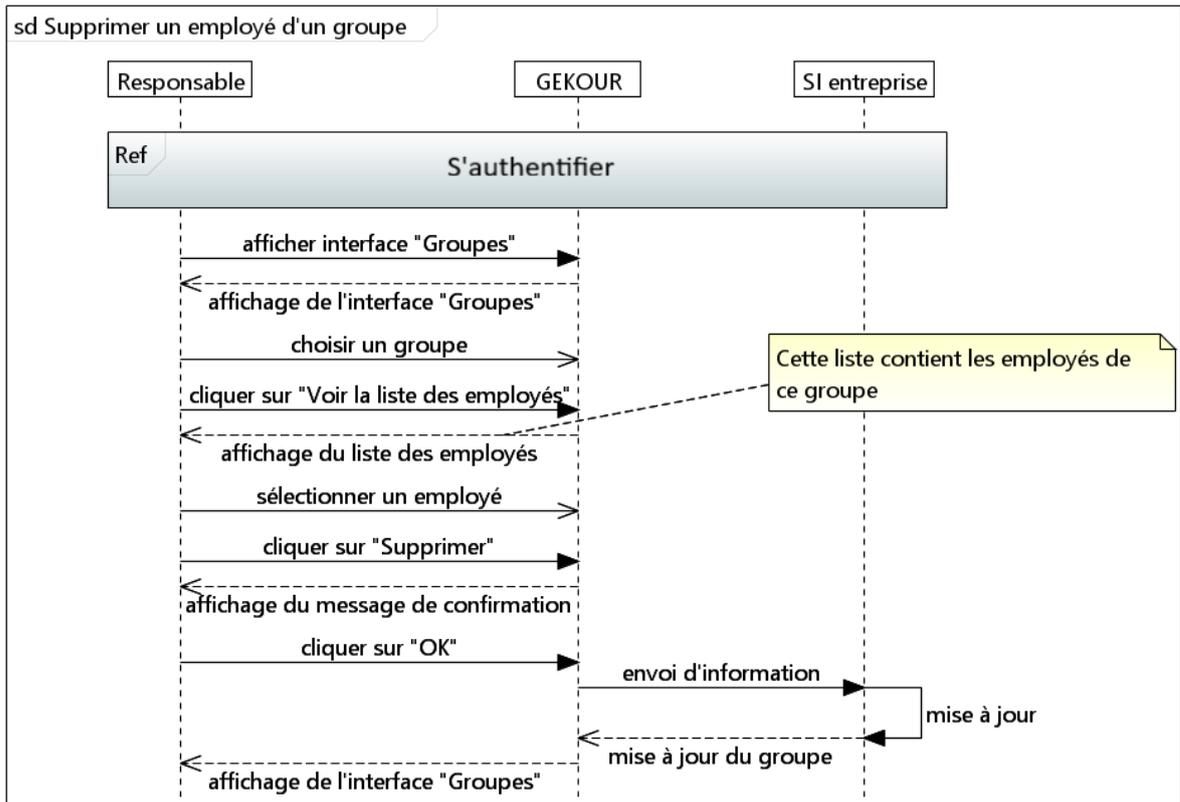


Figure 36 : Diagramme de séquence Supprimer un employé d'un sous-groupe

iii. Les interactions de "Directeur" avec le système

Puisque le directeur est aussi un responsable, il peut faire tout ce qu'un responsable peut faire, en plus d'effectuer d'autres actions et interactions.

- Créer un compte : un directeur peut créer des comptes pour tous les employés et responsables, et même pour les autres directeurs s'ils existent (il existe trois types de comptes, chacun avec des droits d'accès limités).

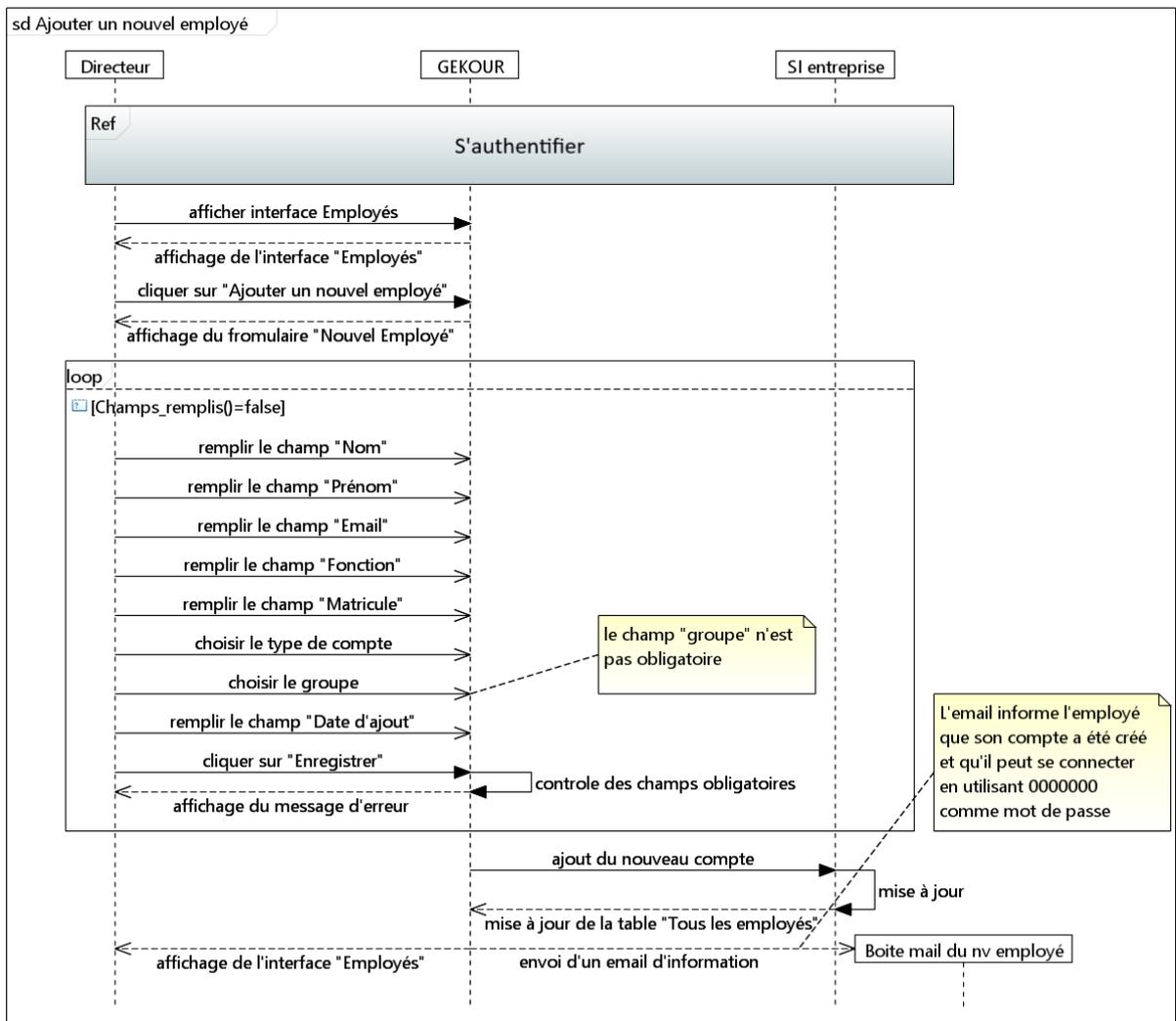


Figure 37 : Diagramme de séquence Ajouter un nouvel employé

Après la création d'un compte, un email sera envoyé à l'adresse email de l'employé l'informant que son compte vient d'être créé et lui attribuant un mot de passe afin qu'il puisse s'authentifier.

- Supprimer un compte : un directeur peut supprimer un compte d'un employé, d'un responsable, ou d'un autre directeur.

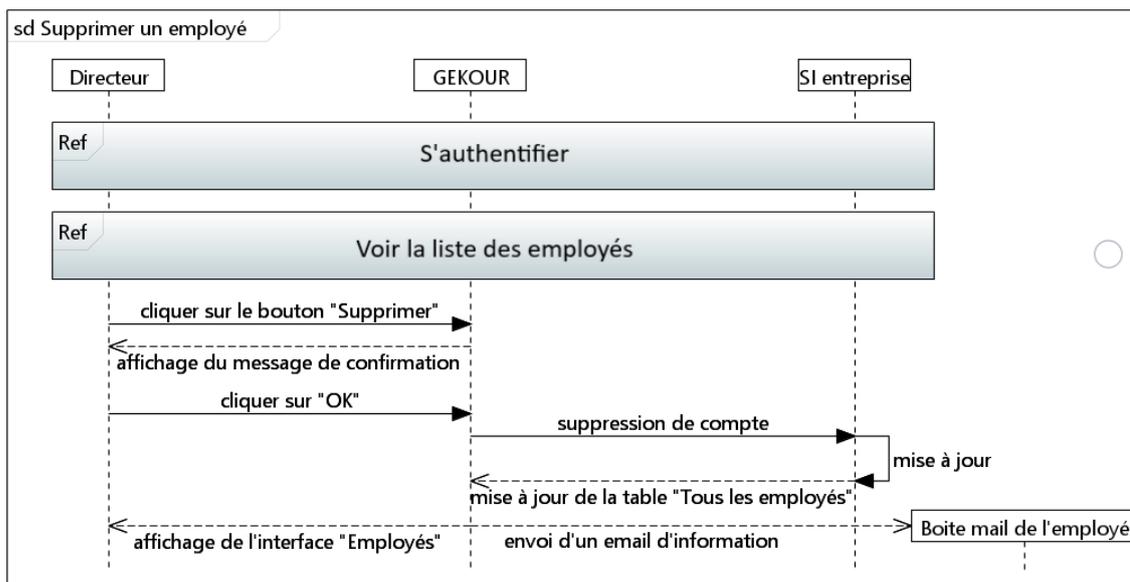


Figure 38 : Diagramme de séquence Supprimer un employé

Après la suppression d'un compte, un email sera envoyé à l'adresse email de l'employé l'informant que son compte vient d'être supprimé.

- Voir la liste des employés : un directeur peut voir la liste des employés.

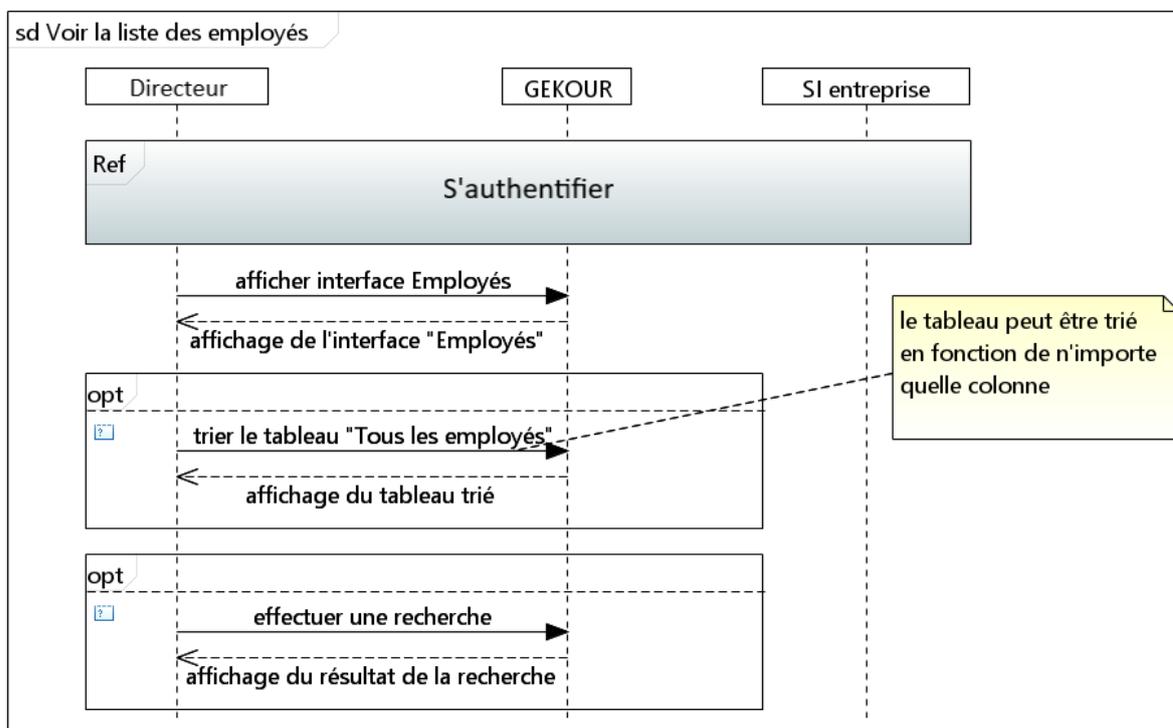


Figure 39 : Diagramme de séquence Voir la liste des employés

- Modifier les informations d'un compte : un directeur peut modifier les informations d'un compte comme : la fonction, le groupe, et le type de compte.

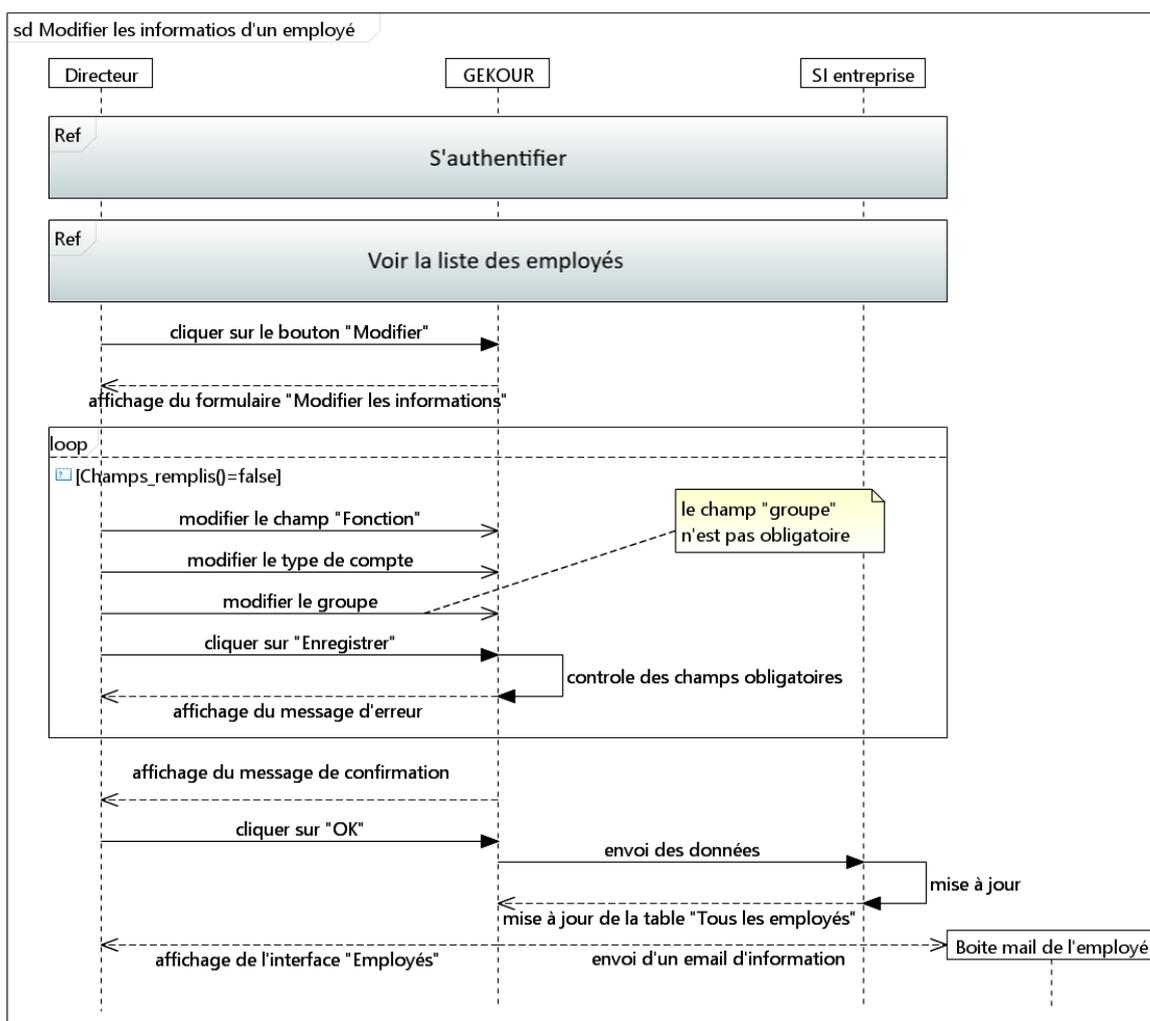


Figure 40 : Diagramme de séquence Modifier les informations d'un employé

Après la modification des informations d'un compte, un email sera envoyé à l'adresse email de l'employé l'informant que les informations de son compte viennent d'être modifiées.

- Créer un groupe : un directeur peut créer des groupes et des sous-groupes

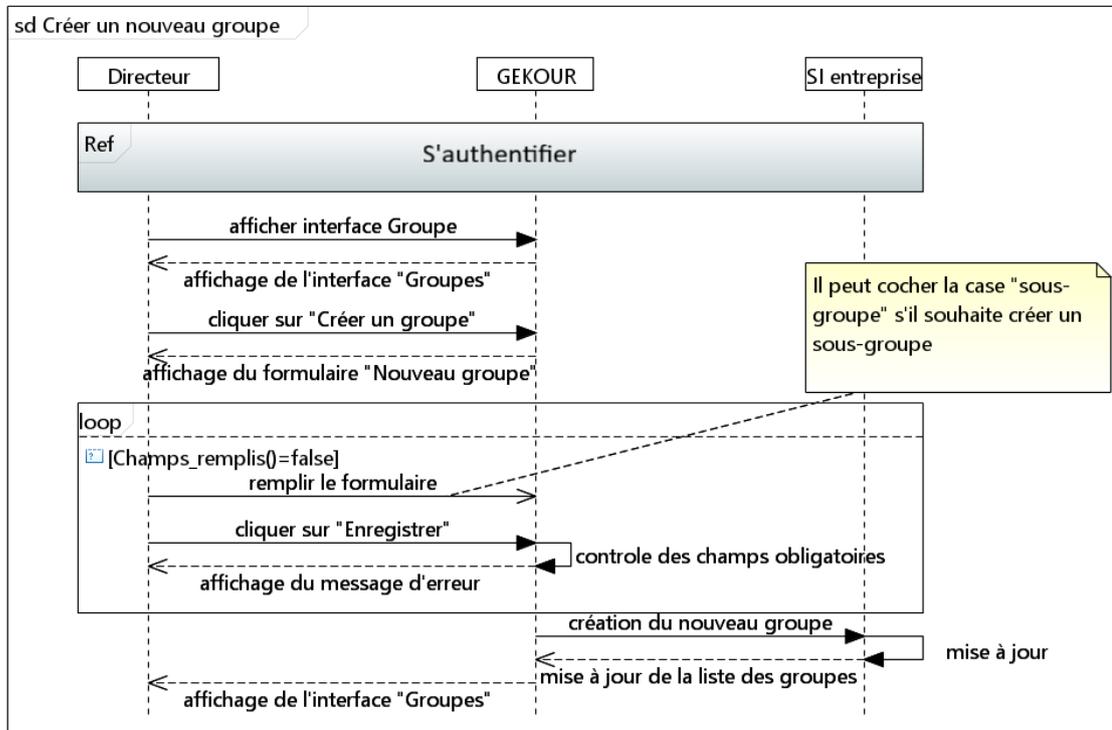


Figure 41 : Diagramme de séquence Créer un groupe

- Supprimer un groupe : un directeur peut supprimer des groupes et des sous-groupes

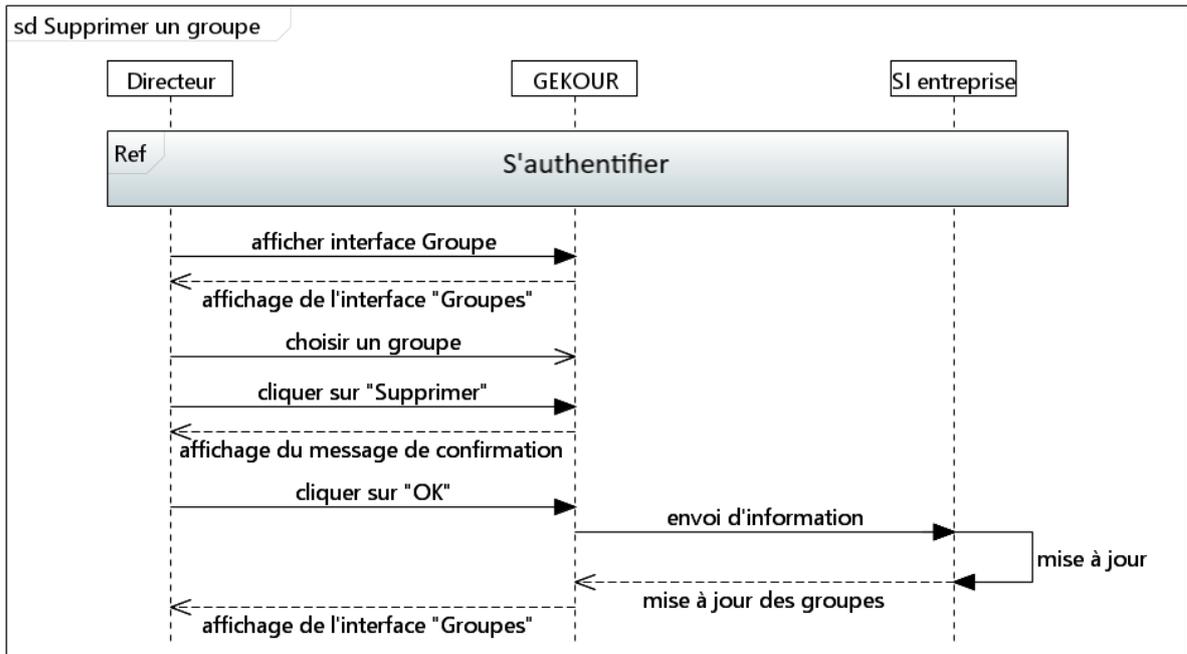


Figure 42 : Diagramme de séquence Supprimer un groupe

Si un groupe est supprimé, tous ses sous-groupes seront supprimés, mais si un sous-groupe est supprimé, cela n'affectera pas son groupe parent.

La suppression d'un groupe ne supprime pas les comptes qui lui appartiennent.

- Voir les statistiques : un directeur peut consulter toutes les statistiques, telles que le nombre de courriers reçus et envoyés, le nombre de courriers dans chaque groupe, le temps de traitement moyen, etc.

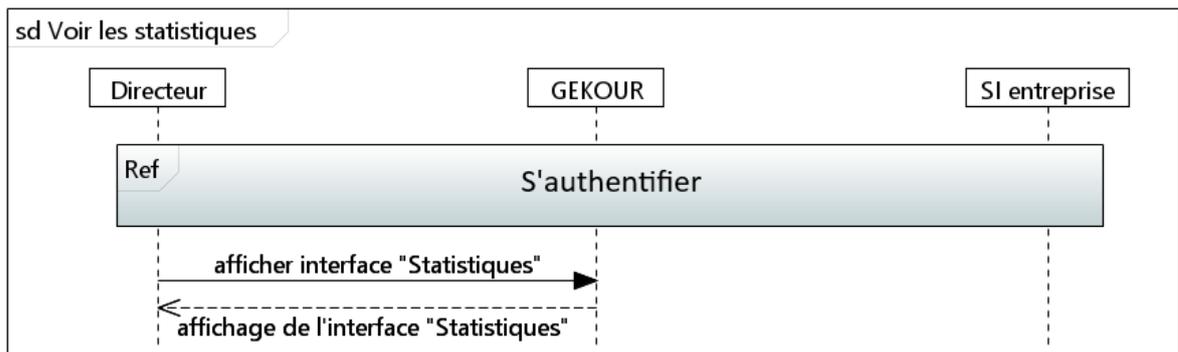


Figure 43 : Diagramme de séquence Voir les statistiques

3.3 Conclusion

Dans ce chapitre, nous avons exploré les aspects clés pour créer une architecture solide et fonctionnelle. Nous avons examiné différents éléments, tels que l'identification des acteurs, la définition des cas d'utilisation et l'utilisation des diagrammes UML pour structurer l'application. [17]

Ainsi, en utilisant le modèle de Kruchten, nous avons bénéficié d'une vision holistique de l'application, en identifiant et en gérant les différents points de vue des parties prenantes. Cela nous a aidés à prendre des décisions éclairées et à garantir que l'application réponde aux besoins de tous les utilisateurs.

La prochaine étape consistera à passer à la phase de construction, qui est la troisième phase du cycle RUP. Nous mettrons en œuvre notre conception et continuerons à itérer pour réaliser les premières fonctionnalités de l'application. Les outils utilisés seront définis dans le chapitre suivant.

Chapitre 4

Implémentation

4.1 Introduction

Ce chapitre est dédié à la présentation des outils essentiels utilisés pour l'implémentation de notre application, ainsi qu'à la mise en avant des captures d'écran illustrant les interfaces principales développées. Divisé en deux parties distinctes, ce chapitre offre un aperçu complet des éléments techniques et visuels qui ont contribué à la réalisation de notre projet.

La première partie mettra en évidence les outils et les technologies adoptés lors de la phase de construction pour la mise en œuvre des premières fonctionnalités de l'application.

La seconde partie mettra l'accent sur les captures d'écran des interfaces principales développées.

4.2 Outils utilisés pour l'implémentation

La construction d'une application web nécessite une sélection minutieuse des outils et des technologies afin d'assurer une implémentation efficace et robuste. Nous avons utilisé une combinaison d'outils de développement et de collaboration pour faciliter notre travail et optimiser notre productivité.

→ Nous avons utilisé comme outils :

4.2.1 De développement

Flask : Flask est un micro framework open-source de développement web en Python. Flask a pour objectif de garder un noyau simple mais extensible.

Jinja : Jinja est un moteur de templating (ou de gabarits) utilisé dans le développement web avec le langage Python. Il est souvent associé aux frameworks web tels que Flask et Django. L'utilisation de Jinja permet de créer des modèles HTML dynamiques et réutilisables, en séparant le code Python du code HTML. Cela facilite la maintenance, la lisibilité et la modularité du code.

Bootstrap : Bootstrap est un framework de développement web open source utilisé pour la création d'interfaces utilisateur réactives et esthétiquement attrayantes. Il fournit une

collection de composants prédéfinis, de styles CSS et de scripts JavaScript qui facilitent la conception et la mise en page des sites web.

Visual Studio Code [18] : Visual Studio Code est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l'autocomplétion, la coloration syntaxique, le débogage, et les commandes git.[19]

DB Browser for SQLite [20] : DB Browser pour SQLite est un outil visuel et open source de haute qualité pour créer, concevoir et éditer des fichiers de base de données compatibles avec SQLite.[21]

4.2.2 De collaboration

Github : Github est une plateforme de développement logiciel basée sur Git, qui facilite la collaboration entre les développeurs, le suivi des versions du code source et la gestion des problèmes.

Google Drive : Un service de stockage en ligne de Google qui permet de créer, partager et collaborer sur des documents, des feuilles de calcul et des présentations en temps réel.

Trello : Un outil de gestion de projets basé sur des tableaux Kanban, qui permet aux équipes de suivre les tâches, de collaborer et de visualiser l'avancement des projets.

→ Nous avons utilisé comme langages de programmation :

Html : L'HyperText Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques.

CSS : CSS (Cascading Style Sheets) est un langage de feuilles de style utilisé pour décrire l'apparence et la mise en forme des documents HTML et XML. Il définit comment les éléments HTML doivent être affichés sur les pages web, y compris les couleurs, les polices, les marges, les espacements, les bordures et autres aspects visuels.

JavaScript : est un langage de script orienté objet principalement utilisé dans les pages HTML. À l'opposé des langages serveur (qui s'exécutent sur le site), JavaScript est exécuté sur l'ordinateur de l'internaute par le navigateur lui-même. Ainsi, ce langage permet une interaction avec l'utilisateur en fonction de ses actions (lors du passage de la souris au-dessus d'un élément, du redimensionnement de la page...) [22].

Python : est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet [23].

Les bibliothèques utilisées avec Python :

- json : json est une bibliothèque standard de Python qui permet de travailler avec le format de données JSON (JavaScript Object Notation). Elle propose des fonctions pour la sérialisation (conversion d'objets Python en chaînes

JSON) et la désérialisation (conversion de chaînes JSON en objets Python) de données.

- `base64` : `base64` est une bibliothèque standard de Python qui fournit des fonctions pour l'encodage et le décodage de données en utilisant l'algorithme Base64.
- `os` : `os` est une bibliothèque standard de Python qui offre des fonctionnalités pour interagir avec le système d'exploitation. Elle permet de manipuler des fichiers et des répertoires, d'exécuter des commandes système, de gérer des variables d'environnement, etc.
- `datetime` : `datetime` est une bibliothèque standard de Python qui fournit des classes pour la manipulation de dates, d'heures et de combinaisons de celles-ci. Elle permet de travailler avec des objets de date et de temps, d'effectuer des calculs temporels et de formater les dates et les heures selon les besoins.
- `sqlalchemy` : `sqlalchemy` est une bibliothèque populaire qui fournit une interface pour interagir avec des bases de données relationnelles en utilisant Python. Elle facilite la gestion des opérations de base de données en offrant une couche d'abstraction au-dessus du langage SQL.
- `flask_sqlalchemy` : `flask_sqlalchemy` est une extension pour Flask qui intègre la bibliothèque SQLAlchemy dans une application Flask. Elle facilite l'utilisation de SQLAlchemy pour interagir avec une base de données relationnelle dans une application Flask.
- `flask_migrate` : `flask_migrate` est une extension pour Flask qui facilite la gestion des migrations de bases de données. Elle offre des fonctionnalités pour créer, appliquer et annuler les migrations de schéma de base de données lorsque vous apportez des modifications à la structure de votre base de données.
- `flask_login` : `flask_login` est une extension pour Flask qui simplifie la gestion de l'authentification utilisateur et des sessions. Elle fournit des fonctionnalités pour la gestion des utilisateurs, la gestion des sessions d'authentification et la protection des routes nécessitant une authentification.
- `flask_jwt_extended` : `flask_jwt_extended` est une extension pour Flask qui permet d'implémenter JSON Web Tokens (JWT) pour l'authentification et l'autorisation des utilisateurs. Elle offre des fonctionnalités pour la génération, la validation et la gestion des JWT dans une application Flask.
- `flask_mail` : `flask_mail` est une extension pour le framework Flask qui facilite l'envoi d'e-mails à partir d'une application Flask. Elle offre des fonctionnalités pour la configuration des paramètres SMTP, la création de messages électroniques et l'envoi d'e-mails.

- `werkzeug` : `werkzeug` est une bibliothèque utilisée par Flask pour gérer les requêtes et les réponses HTTP. Elle fournit des outils pour le routage, la gestion des sessions, la gestion des cookies et d'autres fonctionnalités liées à la manipulation des demandes et des réponses HTTP.
- `werkzeug.security` : `werkzeug.security` est un module de la bibliothèque `werkzeug` qui propose des fonctions pour la gestion des mots de passe sécurisés. Il permet de générer des hachages de mots de passe, de vérifier les mots de passe hachés et de générer des jetons de réinitialisation de mot de passe.

4.3 Captures d'écran des interfaces principales

Dans cette deuxième partie du chapitre, nous présenterons des captures d'écran des interfaces principales de notre application. Ces captures d'écran fourniront un aperçu visuel des fonctionnalités et de l'aspect général de notre application.

L'interface de connexion :

C'est la première interface affichée lorsqu'un employé accède à l'application, il doit entrer un email et un mot de passe valides pour pouvoir se connecter.

Si un employé oublie son mot de passe, il peut cliquer sur "Mot de passe oublié" et un email lui sera envoyé afin qu'il puisse changer son mot de passe.

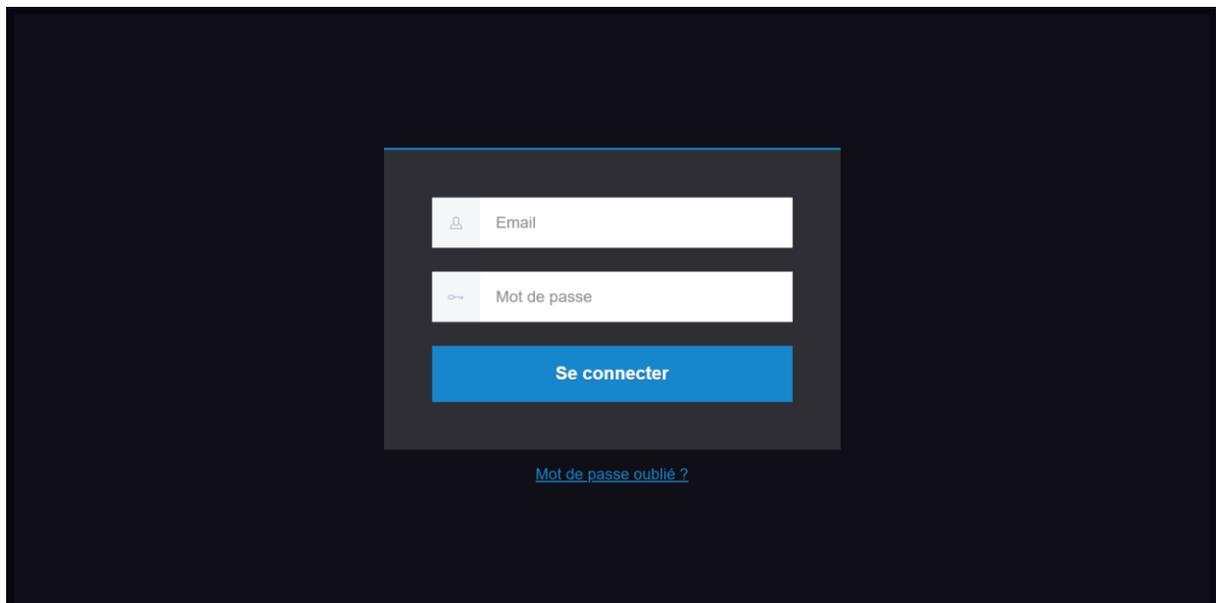


Figure 44 : Interface de connexion

Après l'authentification, l'interface d'accueil apparaîtra automatiquement.

La structure générale des interfaces de l'application :

Dans toutes les interfaces d'application, il existe une barre de navigation située en haut. Elle contient 5 boutons + le logo GEKOUR qui est lui-même un bouton.

Il existe aussi une barre latérale située sur le côté gauche des pages, elle se rétracte lorsqu'on clique sur le bouton "☰" dans la barre de navigation, cette barre contient des liens rapides vers d'autres sections du site.

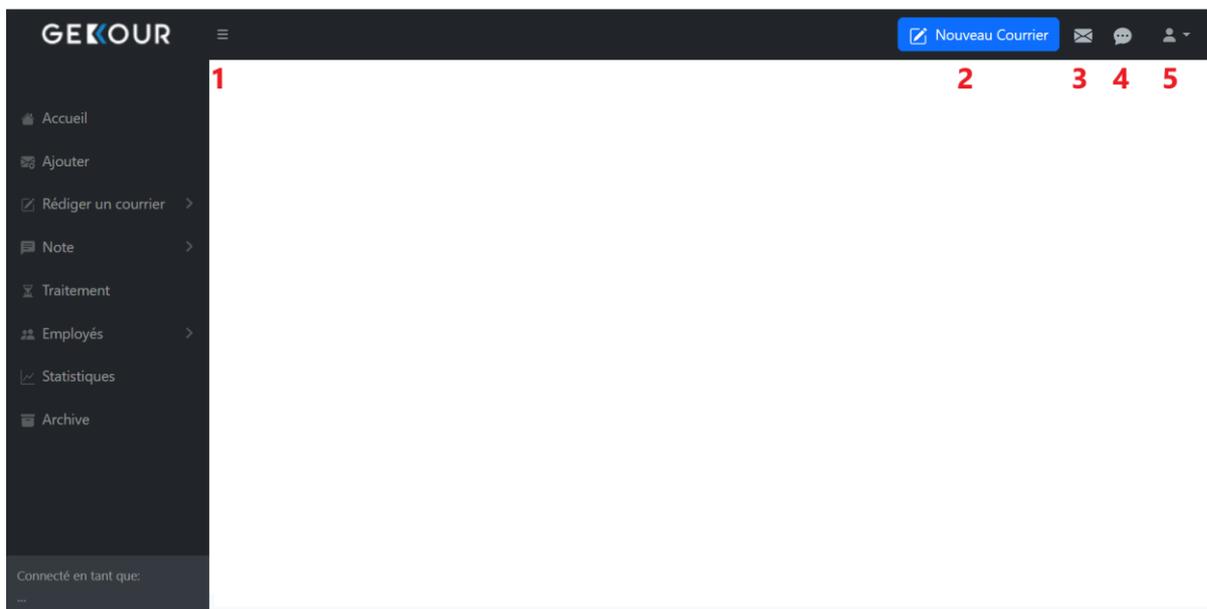


Figure 45 : La structure générale de GEKOUR

2. Le deuxième bouton ouvre l'interface de l'éditeur de texte pour rédiger un nouveau courrier.
3. Le troisième bouton ouvre l'interface d'accueil pour consulter les courriers reçus.
4. Le quatrième bouton ouvre l'interface des notes reçues.
5. Le cinquième bouton est un menu déroulant contenant deux éléments, comme le montre la figure suivante :

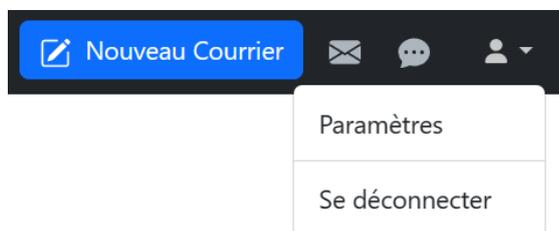


Figure 46 : Le menu déroulant

- " Paramètres " : lorsque l'employé clique dessus, l'interface "Paramètres" s'affiche. Dans cette interface, l'employé peut modifier son mot de passe et ses informations personnelles telles que son nom, son prénom, etc.
- " Se déconnecter " : lorsque l'employé clique dessus, il se déconnecte.

L'interface Accueil :

Il s'agit de l'interface principale de l'application où l'employé peut voir les courriers qu'il a reçus.

Dans le tableau "Courriers reçus", l'employé peut voir les courriers qu'il a reçus, il peut voir les notes jointes à un courrier comme il peut ajouter une ou plusieurs notes à un courrier, il peut changer le statut d'un courrier, l'archiver, rédiger une réponse directement ou l'envoyer à une autre personne ou à un autre groupe.

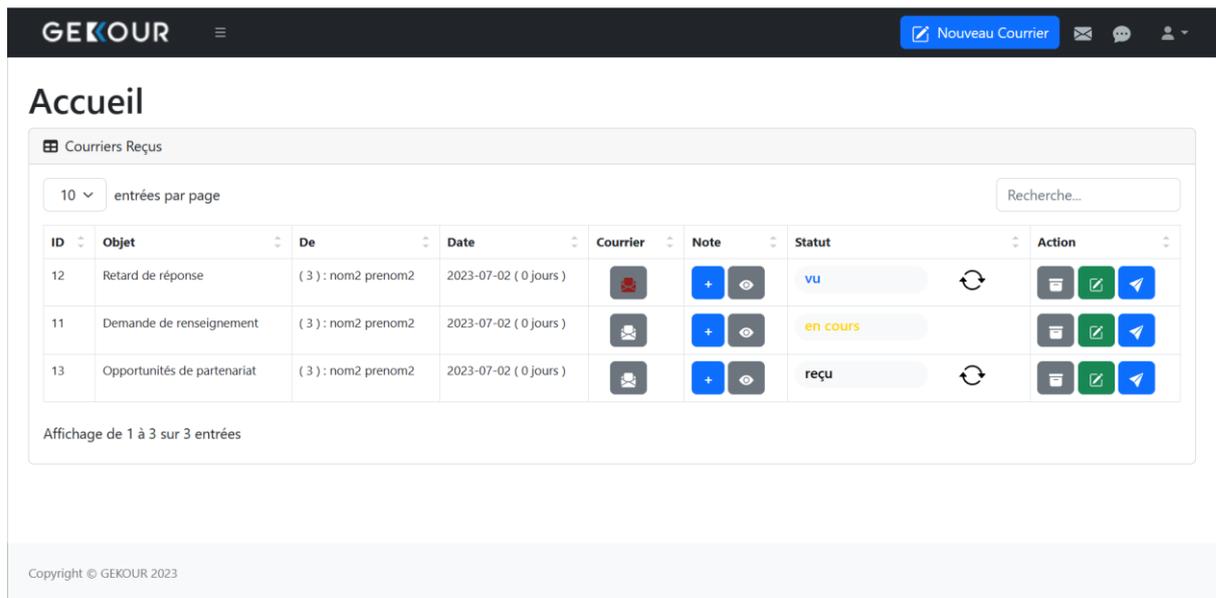


Figure 47 : L'interface d'accueil

Pour voir le contenu d'un courrier, l'employé clique sur l'icône située dans la colonne 'Courrier'. Certaines icônes sont affichées en blanc tandis que d'autres apparaissent en rouge. La couleur rouge indique que le courrier est urgent.

Pour changer le statut d'un courrier, l'employé clique sur le bouton "" dans la colonne "Statut" à droite. Le statut se modifie progressivement.

Lorsque l'employé clique sur "Archiver", le courrier est déplacé vers l'archive.

Lorsque l'employé clique sur "Écrire", une petite fenêtre apparaît, lui permettant de choisir d'utiliser un modèle préétabli ou d'écrire un courrier.

Lorsque l'employé clique sur "Envoyer", une petite fenêtre apparaît, lui permettant de choisir à qui envoyer le courrier : une personne ou un groupe.

L'interface Ajouter:

Lorsque l'employé clique sur "Ajouter" dans la barre latérale, cette interface s'affiche.

Dans le tableau "Courriers Ajoutés", il peut voir la liste des courriers qui ont été ajoutés mais pas encore envoyés, s'il veut envoyer un courrier à une personne ou à un groupe, il suffit de cliquer sur le bouton "Envoyer" et choisir à qui l'envoyer comme indiqué en vert.

S'il souhaite ajouter un courrier, il doit appuyer sur le bouton 'Ajouter un nouveau courrier'. Une petite fenêtre apparaîtra alors, lui permettant de choisir le type de courrier à ajouter, comme indiqué en rouge. Il peut ensuite sélectionner l'un des deux types et remplir les champs requis.



Figure 48 : L'interface Ajouter un courrier

L'interface Nouveau Courrier:

Lorsque un employé souhaite rédiger un nouveau courrier, il clique soit sur le bouton "Nouveau courrier" dans la barre de navigation, soit sur "Rédiger un courrier" puis "Nouveau" dans la barre latérale.

Après avoir cliqué sur "Nouveau", un simple éditeur de texte apparaîtra où l'employé pourra écrire un courrier en lui donnant un objet, il pourra cocher la case urgent si ce courrier est urgent.

Après avoir écrit le courrier, il peut l'imprimer, l'envoyer à une personne ou à un groupe, ou il peut l'enregistrer en tant que brouillon où il peut revenir et le modifier.

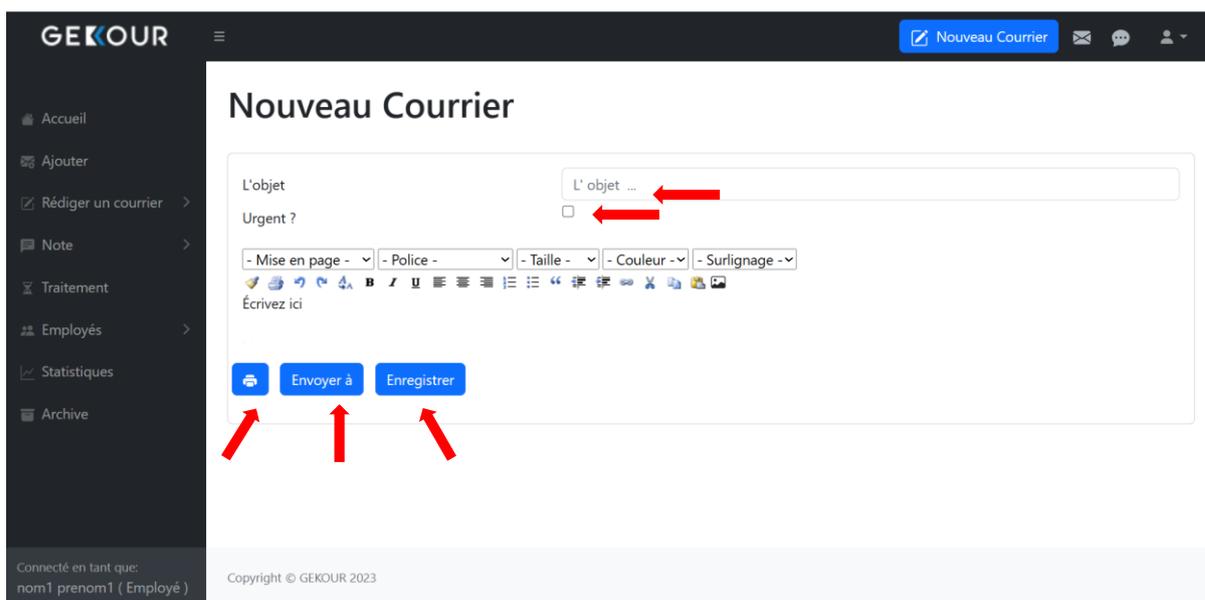


Figure 49 : L'interface de l'éditeur de texte

L'interface Employés:

Si un directeur (administrateur) souhaite consulter la liste des employés ayant déjà des comptes, il doit cliquer sur "Employés" puis sur "Liste des Employés" dans la barre latérale. L'interface "Employés" s'affiche alors, lui permettant de visualiser tous les employés dans le tableau "Tous les employés". Il peut modifier un compte ou le supprimer complètement, comme indiqué en vert.

Et si un directeur veut créer un nouveau compte, il doit cliquer sur "Ajouter un nouvel employé" et remplir les données et choisir le type de compte qui lui convient, comme indiqué en rouge.

The screenshot displays the 'Employés' interface in the GEKOUR system. The left sidebar contains navigation options: Accueil, Ajouter, Rédiger un courrier, Note, Traitement, Employés, Statistiques, and Archive. The main content area is titled 'Employés' and features a blue button 'Ajouter un nouvel employé' circled in red. Below this is a table titled 'Tous les employés' with columns for ID, Nom, Prénom, Matricule, Compte, Groupe, Fonction, and Modifier. A red arrow points to the 'Ajouter un nouvel employé' button, and a green circle highlights the 'Modifier' column's edit and delete icons. A modal window titled 'Nouvel Employé' is open, containing a form with the following fields: Nom *, Prenom *, Email *, Matricule *, Fonction, Type de Compte *, Groupe, and Date d'ajout (with a date picker). An 'Enregistrer' button is at the bottom of the modal.

ID	Nom	Prénom	Matricule	Compte	Groupe	Fonction	Modifier
6	Fatah	Maroua		Responsable	DRHMG : Service Informatique	ingénieur en système d'information	 
5	Ferhi	Ass		Responsable	DMO	chef département	 
2	nom1	prenom1	123506a12lk2	Employé	DRHMG : Service Contentieux	juriste	 
3	nom2	prenom2	123506a12lk2	Directeur	DRHMG : Service Contentieux	directeur général	 

Nouvel Employé

Nom *

Prenom *

Email *

Matricule *

Fonction

Type de Compte *

Groupe

Date d'ajout 

Figure 50 : L'interface Ajouter un nouvel employé

L'interface Statistiques:

Si un directeur souhaite voir les statistiques, il doit cliquer sur "Statistiques" dans la barre latérale. L'interface "Statistiques" s'affiche alors, lui permettant de visualiser les statistiques.

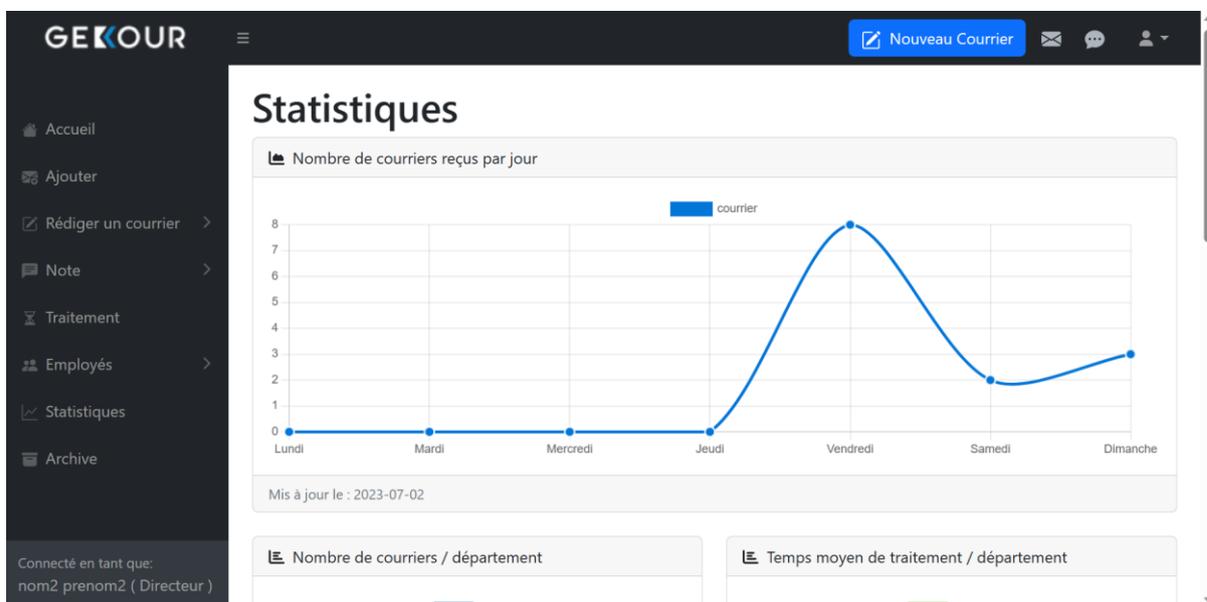


Figure 51 : L'interface Statistiques

La durée de la période de mesure peut être modifiée en jours, semaines ou mois selon les besoins, il suffit de cliquer sur les boutons en bas comme indiqué en rouge.



Figure 52 : Statistiques 2

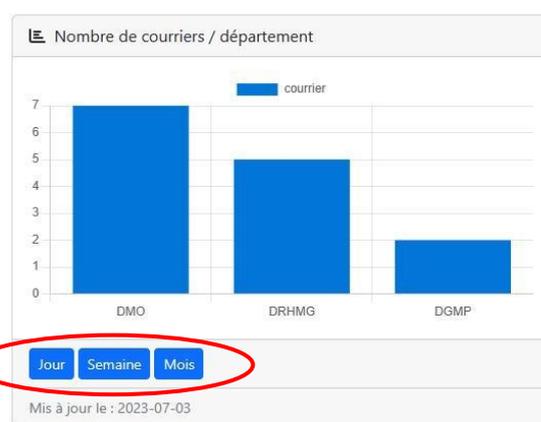


Figure 53 : Statistiques 3

4.4 Conclusion

En récapitulant, ce chapitre a fourni des informations détaillées sur les outils essentiels qui ont été judicieusement sélectionnés pour l'implémentation de notre application. Nous avons étudié en détail les langages de programmation, les frameworks et les bibliothèques utilisés.

Grâce aux choix stratégiques réfléchis, nous avons pu créer une application fonctionnelle et conviviale. L'association de langages de programmation efficaces et de frameworks adaptés a été déterminante dans la création d'un code robuste et performant.

Parallèlement, les interfaces principales développées ont été soigneusement conçues pour répondre aux besoins et aux attentes des utilisateurs. Une attention particulière a été accordée à l'ergonomie et à l'expérience utilisateur, en veillant à ce que l'application soit intuitive, agréable à utiliser et réponde de manière optimale aux objectifs des utilisateurs.

Enfin, nous avons réussi à créer une application qui combine efficacité et convivialité, offrant ainsi une expérience utilisateur satisfaisante. En veillant à ce que notre application soit fonctionnelle et intuitive, nous sommes en mesure de répondre aux besoins et aux attentes des utilisateurs, tout en favorisant leur engagement et leur satisfaction optimale.

Conclusion Générale

La gestion efficace des flux de courriers est un élément basique et important pour le bon fonctionnement d'une entreprise [24].

Dans le cadre de notre projet de fin d'études, nous avons proposé une solution visant à améliorer la gestion des courriers au sein des entreprises en développant une application web qui permettra de traiter de manière rapide et fiable les courriers entrants et sortants tout en simplifiant les tâches. Elle sera facile à utiliser et intuitive pour les utilisateurs et offrira des fonctionnalités innovantes permettant d'améliorer leur productivité.

En somme, notre projet permettra de moderniser la gestion des courriers de l'entreprise tout en assurant un gain de temps et d'efficacité pour les utilisateurs, une solution qui répondra aux besoins des entreprises modernes en matière de gestion de courrier, afin d'assurer ses intérêts.

Ce mémoire a ainsi offert une vision globale et approfondie de notre travail de développement, mettant en exergue la cohérence entre les objectifs fixés et les résultats obtenus. Dans la continuité de ce projet, nous espérons que notre application suscitera un impact positif et apportera une réelle valeur ajoutée à ses utilisateurs.

Bibliographie

Livre

- [14] Clements, Paul, et al , Documenting Software Architectures : Views and Beyond, 2nd édition, Addison-Wesley, 2010.
- [17] Pascal Roques, UML 2 par la pratique, Etudes de cas et exercices corrigés, 5eme édition, 14/09/2006
- [24] Trabichet , La Gestion électronique de documents et de courrier. Éditions d'Organisation., J. (2016).

Articles de revue

- [1] Bach Tobji, M. A. (2011). Gestion électronique des courriers au sein d'une administration. Université virtuelle de Tunis.

Documents web

- [2] OPGI Mostaganem : présentation, <https://opgimostaganem.dz/presentation.php>
- [3] OPGI Mostaganem : Historique, <https://opgimostaganem.dz/Historique.php>
- [4] OPGI Mostaganem : organigramme, <https://opgimostaganem.dz/organigramme.php>
- [5] lucidchart : BPMN : définition, utilisation et exemples, <https://www.lucidchart.com/pages/fr/bpmn>
- [6] EduTechWiki : Les diagrammes de chorégraphie, [/https://edutechwiki.unige.ch/fr/Bpmn 2.0#Les diagrammes de chor%C3%A9graphie](https://edutechwiki.unige.ch/fr/Bpmn 2.0#Les diagrammes de chor%C3%A9graphie)
- [7] EduTechWiki: Les diagrammes de conversation, <https://edutechwiki.unige.ch/fr/Bpmn 2.0#Les diagrammes de conversation>
- [8] openMairie : openCourrier , <http://www.openmairie.org/catalogue/opencourrier>

- [9] Journal Scientifique et Technique du Mali : Gestion électronique des courriers administratifs : l'Agetic développe eGESCO , <https://www.jstm.org/gestion-electronique-des-courriers-administratifs-lagetic-developpe-egesco>
- [10] SIAT : MENARA-GBO : Gestion du courrier (Bureau d'Ordre), <http://www.siat.com.tn/gbo.html>
- [11] Nutcache : Méthode RUP, <https://www.nutcache.com/fr/blog/methode-rup>
- [12] Ptidej.net : Pratique recommandée par IEEE pour la spécification des exigences de logiciels, 2011, https://www.ptidej.net/courses/log3410/fall11/Lectures/Pratique_Recommandee_Par_III_III_III_pour_la%20Specification.pdf
- [13] Wikipedia : Application web, https://fr.wikipedia.org/wiki/Application_web
- [15] Complex systems and AI : Analyse logicielle , <https://complex-systems-ai.com/analyse-logicielle/>
- [16] IPPON positive technology : Diagramme de contexte , <https://blog.ippon.fr/2020/10/16/diagramme-de-contexte-vs-hexagone/#:~:text=Le%20diagramme%20de%20contexte%20est%20une%20mod%C3%A9lisation%20permettant%20de%20mettre,et%20les%20POs%20du%20projet.>
- [18] Wikipédia contributors : Visual Studio Code, https://fr.wikipedia.org/wiki/Visual_Studio_Code
- [19] Framalibre : Visual Studio Code, <https://framalibre.org/content/visual-studio-code>
- [20] DB Browser for SQLite : DB Browser for SQLite, <https://sqlitebrowser.org/>
- [21] Blaise Pascal University : CPGE-DB Browser for SQLite , <https://info.blaisepascal.fr/cpge-db-browser-for-sqlite/>
- [22] Wikipédia contributors : JavaScript, [JavaScript - Wikipedia](https://fr.wikipedia.org/wiki/JavaScript)
- [23] Python Software Foundation : Python 3.9.7 Documentation, <https://docs.python.org/fr/3/tutorial/>