République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université M'hamed Bougara - Boumerdès



Faculté des Sciences
Département d'Informatique

Domaine : Mathématiques Informatique
Filière : Informatique
Spécialité : Systèmes d'informatique

N° de l'Arrêté d'habilitation de la spécialité : arrêté n °872 du 26/07/2016

Mémoire de fin d'études en vue de l'obtention du

Diplôme de Licence Académique

Thème

Conception et réalisation d'un système d'information pour la gestion des rapports d'activité avec extraction automatique de mots-clés

Présenté par :

Hafsa Nesrine — Kerchou Manel
Tadjouimat Feriel — Drouazi Kaouthar

Stage Pratique réalisé au : SONATRACH.

Soutenu le ...../...../ ..... Devant le jury composé de

Herbadji Abderrahmane : Examinateur
Hadjadji Imene : Encadreur
Ouragh Nawel : Promoteur

Remerciements

Avant tout, nous remercions Allah, le Miséricordieux, de nous avoir accordé la force, le courage et la patience nécessaires pour mener à bien ce travail.

Nous tenons à remercier chaleureusement nos encadrantes, au sein de l'université et de l'entreprise, pour leurs conseils, leurs orientations et l'intérêt qu'elles ont porté à notre projet tout au long de sa réalisation.

Nous remercions également les membres du jury pour l'honneur qu'ils nous font en acceptant d'examiner et d'évaluer ce mémoire.

Nous exprimons notre gratitude à tous nos enseignants qui ont contribué à notre formation et nous ont transmis leurs connaissances durant notre parcours universitaire.

Enfin, nous remercions nos familles pour leur soutien, leurs encouragements et leur présence tout au long de l'élaboration de ce travail.

Table des matières

Remerciement **II**

Liste des figures **VI**

Liste des tableaux **VII**

Introduction générale **1**

1 Présentation de l'organisme d'accueil et cadre du projet **2**

1.1 Introduction 2
1.2 Présentation de SONATRACH 2
1.3 Présentation de la structure d'accueil 3
1.3.1 Division Exploration 3
1.3.2 Direction Data Management 3
1.3.3 Département Technologie de l'Information 4
1.4 Cadre du projet 4
1.5 Problématique 4
1.6 Solution proposée 5
1.7 Conclusion 5

2 Analyse des besoins et Conception **6**

2.1 Introduction 6
2.2 Analyse des besoins 7
2.2.1 Identification des acteurs : 7
2.2.2 Besoins fonctionnels : 7
2.2.3 Besoins non fonctionnels : 7
2.3 Diagramme des cas d'utilisation : 8
2.3.1 Diagramme de cas d'utilisation général : 8
2.3.2 Diagramme de cas d'utilisation de l'administrateur applicatif : . . . 9
2.3.3 Diagramme de cas d'utilisation du chef de groupe : 10
2.3.4 Diagramme de cas d'utilisation du responsable d'activité : 11
2.3.5 Diagramme de cas d'utilisation de chef de département : 12
2.4 Diagramme de séquence : 13

2.4.1 Diagramme de séquence authentification 13
2.4.2 Diagramme de séquence Création d'une Activité / Sous-activité . . 14
2.4.3 Diagramme de séquence – Assignation d'un responsable d'activité . 14
2.4.4 Diagramme de séquence – Description d'une activité 15
2.4.5 Diagramme de séquence – Enregistrement du nombre d'intervention- 16
2.4.6 Diagramme de séquence validation des activités 17
2.4.7 Diagramme de séquence –Validation des interventions - 18
2.4.8 Diagramme de séquence validation de rapport global : 19
2.4.9 Diagramme de séquence –Assistant IA 19
2.5 Diagramme de classe : 20
2.5.1 Dictionnaire de données 20
2.5.2 Diagramme de classe : 23
2.6 Diagramme d'activité : 24
2.6.1 Diagramme d'activité général : 24
2.7 Conclusion 26

3 Implémentation **27**
3.1 Introduction 27
3.2 Environnement de développement 27
3.2.1 Environnement logiciel 27
3.2.2 Langages de développement 28
3.2.3 Frameworks et bibliothèques 30
3.3 Modèle d'intelligence artificielle 31
3.3.1 Objectif du modèle 31
3.3.2 Données utilisées 31
3.3.3 Méthode utilisée 31
3.3.4 Justification du choix méthodologique 31
3.3.5 Évaluation du modèle 32
3.4 Interfaces de l'application 33
3.4.1 Interface de Connexion : 33
3.4.2 Interface de Dashboard : 33
3.4.3 Interface des interventions : 33
3.4.4 Interface des activités : 34
3.4.5 Interface de l'espace éditeur : 34
3.4.6 Interface d'administration : 35
3.4.7 Interface d'assignation : 35
3.4.8 Interface du rapport global : 35
3.4.9 Les tests fonctionnels : 36

3.5 Conclusion : 37

Bibliographie **39**

Table des figures

1.1 Logo de SONATRACH [1] 2
1.2 Organigramme de SONATRACH [1] 3
1.3 Organigramme de la Direction Data Management [2] 3

2.1 Diagramme de cas d'utilisation général 9
2.2 Diagramme de cas d'utilisation de l'admin applicatif 10
2.3 Diagramme de cas d'utilisation de chef de groupe 11
2.4 Diagramme de cas d'utilisation de responsable d'activité 12
2.5 Diagramme de cas d'utilisation Chef de département 12
2.6 Diagramme de séquence authentification 13
2.7 Diagramme de séquence création d'une Activité / Sous-activité 14
2.8 Diagramme de séquence assignation d'un responsable d'activité 15
2.9 Diagramme de séquence description d'une activité 16
2.10 Diagramme de séquence Enregistrement du nombre d'intervention 17
2.11 Diagramme de séquence Validation des activités 18
2.12 Diagramme de séquence Validation des interventions 18
2.13 Diagramme de séquence Validation du rapport global 19
2.14 Diagramme de séquence Assistant IA 20
2.15 Diagramme de classe 24
2.16 Diagramme d'activité général 25

3.1 logo de wampserver [3] 27
3.2 Logo de visual studio code[4] 28
3.3 Logo phpmyadmin [5] 28
3.4 Logo de postman [4] 28
3.5 Logo google colab [6] 28
3.6 Logo de draw.io[7] 28
3.7 logo html[8] 29
3.8 Logo css[8] 29
3.9 Logo java script 29
3.10 Logo PHP [9] 29
3.11 logo Python [8] 29
3.12 Logo laravel [10] 30

3.13 Logo bootstrap[11] 30
3.14 Répartition des données utilisées pour le modèle IA 31
3.15 Rapport de classification du modèle de type de demande 32
3.16 Matrice de confusion du modèle de type de demande 32
3.17 Interface de connexion 33
3.18 interface Dashboard 33
3.19 interface des interventions 33
3.20 interface des activités 34
3.21 Interface de l'espace éditeur 34
3.22 interface administration 35
3.23 Interface assignation 35
3.24 Interface du rapport global 35
3.25 Nouvelle activité 36
3.26 Liste des activités 36
3.27 Interface de l'Assistant IA 37

Liste des tableaux

2.1 Identification des acteurs 7
2.2 Besoins non fonctionnels du système 8
2.3 Dictionnaire de données 23

Introduction générale

La gestion de l'information occupe aujourd'hui une place importante dans le fonctionnement des organisations. Dans un contexte professionnel où les équipes produisent régulièrement des documents de suivi et des rapports d'activité, disposer d'outils adaptés pour centraliser, organiser et exploiter ces informations devient une nécessité.

C'est dans ce cadre que s'inscrit notre projet de fin d'études, réalisé au sein du Département Technologie de l'Information de SONATRACH, à la Division Exploration de Boumerdès. Ce département regroupe plusieurs groupes de travail qui produisent chaque mois des rapports d'activité décrivant les tâches réalisées et les interventions techniques effectuées. Actuellement, ces rapports sont rédigés séparément sous forme de fichiers Word, puis regroupés et consolidés manuellement par les responsables. Cette méthode engendre plusieurs difficultés : les fichiers sont dispersés, la centralisation est absente, les erreurs de saisie sont fréquentes et la visibilité globale sur l'avancement des travaux reste limitée.

Face à ces constats, l'objectif de notre projet est de concevoir et de développer une plateforme web centralisée qui permet aux différents acteurs du département de saisir leurs activités et interventions dans un espace commun, de suivre l'état des rapports en temps réel, de valider les données à chaque niveau hiérarchique et de générer automatiquement un rapport global consolidé. Cette solution vise à réduire les erreurs, à améliorer la traçabilité et à simplifier le travail de consolidation mensuelle.

La plateforme inclut également un module d'analyse des demandes textuelles, basé sur des techniques de traitement du langage naturel. Ce module permet d'extraire automatiquement les termes significatifs des demandes saisies en texte libre, puis de suggérer une catégorie correspondante, afin d'assister les utilisateurs dans leur travail de saisie et de classement. Afin de mener ce travail de façon structurée, ce mémoire est organisé en trois chapitres. Le premier présente l'organisme d'accueil, le cadre du projet et la problématique identifiée. Le deuxième chapitre est consacré à l'analyse des besoins et à la conception du système à travers les diagrammes UML. Le troisième chapitre décrit l'environnement de développement, les technologies utilisées, le module d'analyse textuelle et les principales interfaces réalisées.

Chapitre 1

Présentation de l'organisme d'accueil et cadre du projet

1.1 Introduction

Ce chapitre présente l'organisme au sein duquel nous avons effectué notre stage, à savoir SONATRACH et plus précisément son Département Technologie de l'Information, rattaché à la Direction Data Management de la Division Exploration de Boumerdès. Après avoir décrit cette structure d'accueil, nous exposons le cadre du projet, la problématique observée sur le terrain et la solution que nous proposons .



FIGURE 1.1 – Logo de SONATRACH [1]

1.2 Présentation de SONATRACH

SONATRACH est la compagnie nationale algérienne des hydrocarbures, créée en 1963. Elle couvre toute la chaîne pétrolière et gazière : exploration, production, transport, trans-

formation et commercialisation. Son organisation comprend une Direction Générale, des directions fonctionnelles (finance, RH, digitalisation, etc.), ainsi que des pôles opérationnels dont le pôle Exploration-Production [1].

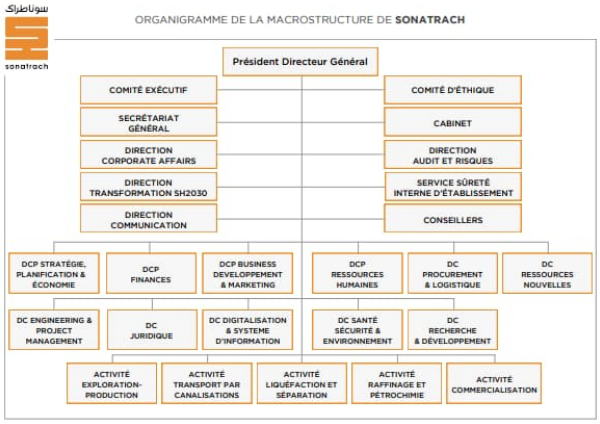


FIGURE 1.2 – Organigramme de SONATRACH [1]

1.3 Présentation de la structure d'accueil

1.3.1 Division Exploration

La Division Exploration est chargée de la recherche de nouveaux gisements d'hydrocarbures. Elle regroupe plusieurs directions techniques, dont la Direction Data Management [2].

1.3.2 Direction Data Management

Cette direction assure la gestion, l'organisation et la valorisation des données au sein de la Division Exploration. Elle contient plusieurs départements, dont le Département Technologie de l'Information [2].

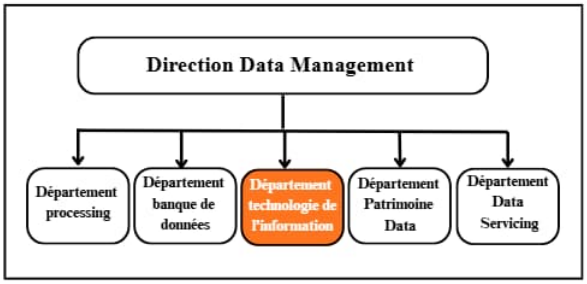


FIGURE 1.3 – Organigramme de la Direction Data Management [2]

1.3.3 Département Technologie de l'Information

Le Département TI est responsable des systèmes informatiques, du développement des applications et du support aux utilisateurs

Ses missions principales sont :
— gestion et maintenance du parc informatique ;
— développement et maintenance des applications ;
— sécurité des systèmes d'information ;
— disponibilité et performance des outils numériques.

Pour mener à bien ces missions, le département s'organise autour de quatre groupes : HelpDesk, Internet et Intranet ; Réseaux ; Support des applications cœurs de métiers ; et Système. Chaque groupe assure des tâches techniques régulières et établit, en fin de mois, un rapport d'activité retraçant les interventions réalisées. C'est précisément cette production mensuelle de rapports qui est à l'origine de notre projet de centralisation [12].

1.4 Cadre du projet

Le projet concerne la gestion des rapports d'activité du Département TI. Ces rapports décrivent les tâches réalisées et les interventions techniques.

Actuellement, chaque groupe crée ses rapports sous forme de fichiers Word. Ces fichiers sont ensuite regroupés et consolidés manuellement par les responsables.

Cette méthode pose plusieurs problèmes :
— dispersion des fichiers ;
— absence de centralisation ;
— consolidation manuelle lente et sujette aux erreurs ;
— manque de visibilité globale ;
— absence d'exploitation automatique du contenu.

1.5 Problématique

L'analyse de l'existant montre plusieurs limites :

— absence d'espace centralisé ;
— manque de visibilité entre les groupes ;
— consolidation manuelle complexe ;
— faible traçabilité des actions ;
— absence d'analyse automatique des rapports.

La problématique est donc :

Comment concevoir un système d'information centralisé permettant de faciliter la saisie, le suivi et la consolidation des rapports d'activité du département IT, tout en intégrant des mécanismes d'extraction automatique des mots-clés ?

1.6 Solution proposée

Nous proposons une plateforme web centralisée pour la gestion des rapports d'activité.

Cette solution permet :

— centralisation des données ;
— amélioration de la visibilité ;
— réduction du temps de consolidation ;
— meilleure traçabilité ;
— recherche facilitée des rapports.

Elle inclut aussi un module d'extraction automatique des mots-clés basé sur le traitement automatique du langage naturel.

1.7 Conclusion

Ce chapitre a présenté l'organisme d'accueil SONATRACH et la structure du projet. Nous avons analysé le système existant et identifié ses limites. Le chapitre suivant sera consacré à l'analyse des besoins et à la conception du système proposé.

Chapitre 2

Analyse des besoins et Conception

2.1 Introduction

Dans ce chapitre, nous présentons l'analyse des besoins de notre système ainsi que sa conception. Nous commençons par identifier les acteurs et les besoins fonctionnels et non fonctionnels, puis nous utilisons le langage UML pour modéliser le système à travers différents diagrammes. Cette phase de conception est essentielle avant de passer à la réalisation.

2.2 Analyse des besoins

2.2.1 Identification des acteurs :

Acteur	**Rôle**
Admin applicatif	Celui qui gère l'application avec des privilèges d'administration. Il est responsable de la création des utilisateurs, de l'attribution des rôles et de la configuration générale du système.
Chef de groupe	Celui qui supervise les activités de son groupe. Il ajoute les membres, crée et organise les activités et sous-activités, assigne les responsables et valide les interventions avant verrouillage du rapport.
Responsable d'activité	Celui qui prend en charge une activité ou une sous-activité. Il renseigne la description dans l'espace éditeur et saisit le nombre d'interventions réalisées selon les types définis.
Chef de département	Celui qui supervise l'ensemble des groupes. Il effectue la consolidation globale et valide le rapport mensuel final.

TABLE 2.1 – Identification des acteurs

2.2.2 Besoins fonctionnels :

Les besoins fonctionnels décrivent les principales fonctionnalités offertes par le système de gestion des rapports d'activité, notamment la gestion des activités, des interventions et des rapports mensuels. Le système permet l'authentification des utilisateurs, l'assignation des responsabilités, la saisie des données ainsi que la génération et la validation du rapport global.

2.2.3 Besoins non fonctionnels :

Le tableau suivant présente les besoins non fonctionnels, qui définissent les contraintes de qualité que doit respecter le système afin d'assurer son bon fonctionnement :

Critère	**Exigence**	**Indicateur / Seuil**
Sécurité	Authentification par identifiant et mot de passe	Mots de passe hachés avec bcrypt (coût ≥ 10)
Performance	Temps de réponse des pages	$\leq$ 3 secondes pour 95 % des requêtes
Disponibilité	Accessibilité du système	Disponibilité $\geq$ 99 % durant les heures ouvrables
Charge	Nombre d'utilisateurs simultanés	Support de 50 utilisateurs concurrents minimum
Traçabilité	Journal d'audit des actions critiques	Rétention des logs $\geq$ 12 mois
Ergonomie	Interface utilisable sans formation	Prise en main en moins de 30 minutes
Fiabilité	Intégrité des données saisies	Validation côté serveur sur tous les formulaires

TABLE 2.2 – Besoins non fonctionnels du système

2.3 Diagramme des cas d'utilisation :

Le diagramme de cas d'utilisation est un diagramme UML qui représente les interactions entre les acteurs et le système. Il décrit les fonctionnalités offertes par le système du point de vue de l'utilisateur [13].

2.3.1 Diagramme de cas d'utilisation général :

Cette figure présente le diagramme général des cas d'utilisation du système, permettant d'identifier les différentes interactions possibles entre les acteurs et le système.

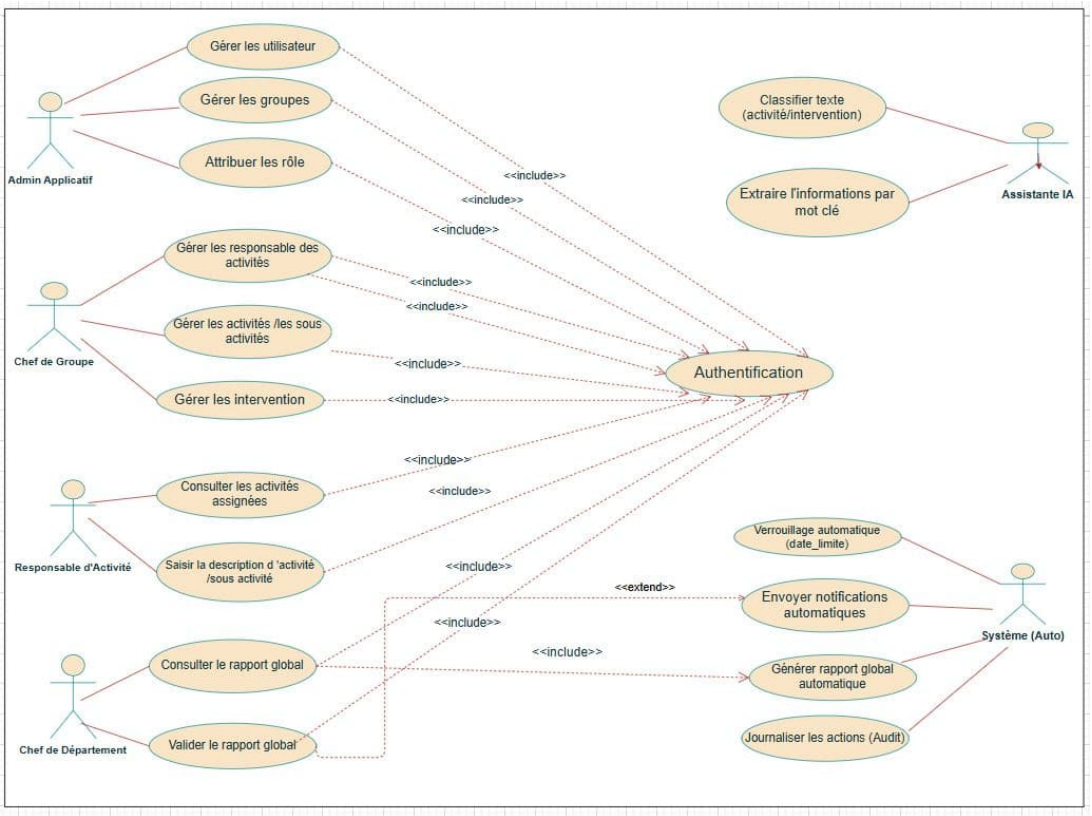


FIGURE 2.1 – Diagramme de cas d'utilisation général

Description du diagramme de cas d'utilisation général :

Le système repose sur quatre acteurs principaux : l'administrateur applicatif, le chef de groupe, le responsable d'activité et le chef de département. Chacun accède au système après authentification et dispose de fonctionnalités adaptées à son rôle. Le système prend aussi en charge certaines actions automatiques, comme le verrouillage des rapports, l'envoi des notifications et la génération du rapport global.

2.3.2 Diagramme de cas d'utilisation de l'administrateur applicatif :

Cette figure représente le diagramme de cas d'utilisation de l'admin applicatif ainsi que les différentes fonctionnalités que l'administrateur applicatif peut réaliser :

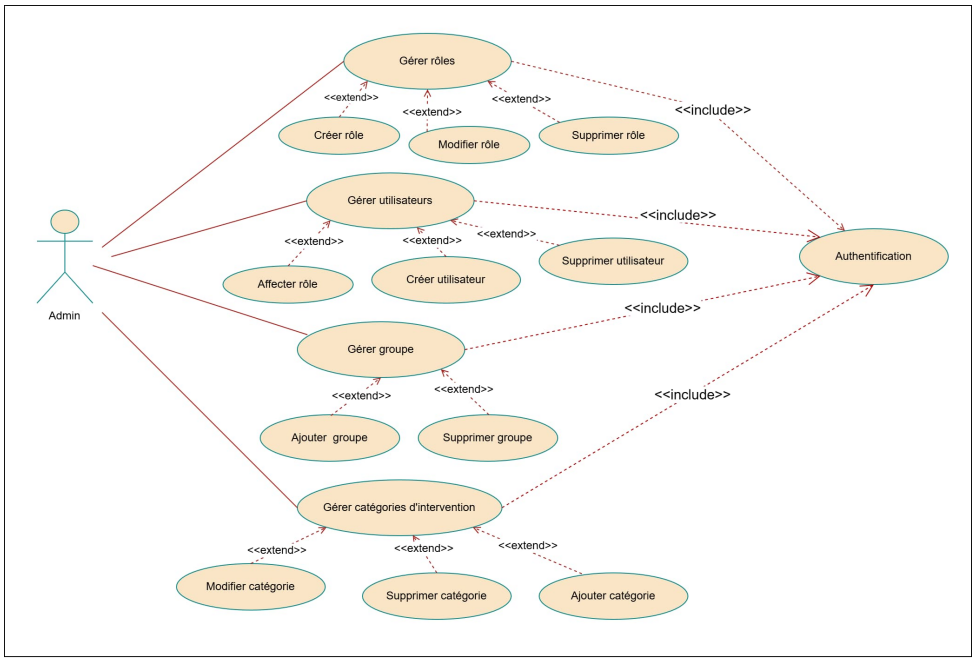


FIGURE 2.2 – Diagramme de cas d'utilisation de l'admin applicatif

Description du diagramme de cas d'utilisation "ADMIN APPLICATIF " :

L'administrateur est responsable de la configuration générale du système. Il gère les utilisateurs, leur attribue des rôles et organise les groupes de travail. Il s'occupe aussi des catégories d'intervention pour faciliter la classification des interventions. Son rôle est essentiel pour assurer un accès sécurisé et bien structuré à l'application.

2.3.3 Diagramme de cas d'utilisation du chef de groupe :

Cette figure représente le diagramme de cas d'utilisation de chef de groupe ainsi que les différentes fonctionnalités que le chef de groupe peut réaliser :

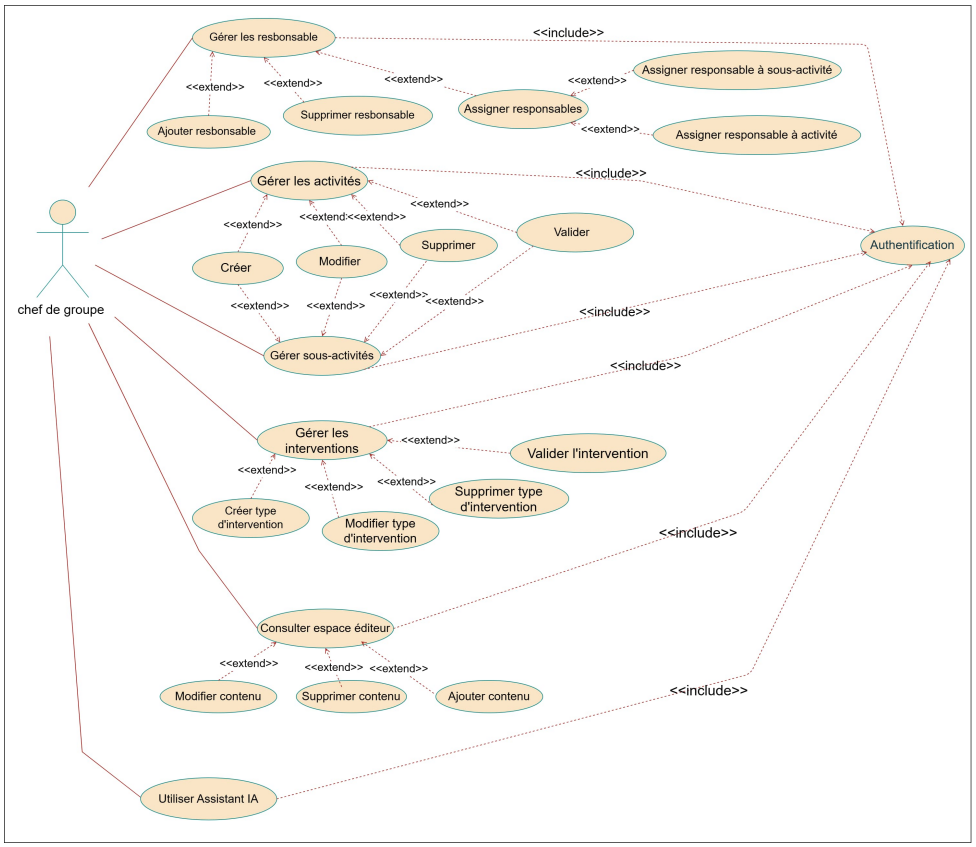


FIGURE 2.3 – Diagramme de cas d'utilisation de chef de groupe

Description du diagramme de cas d'utilisation "chef de groupe " :

Le chef de groupe joue un rôle central dans l'organisation du travail de son équipe. Il crée les activités et sous-activités, désigne les responsables et gère les types d'intervention. Il peut également modifier le contenu de l'espace éditeur et valider les activités et les interventions.

2.3.4 Diagramme de cas d'utilisation du responsable d'activité :

Cette figure représente le diagramme de cas d'utilisation du responsable d'activité ainsi que les différentes fonctionnalités que le responsable d'activité peut réaliser :

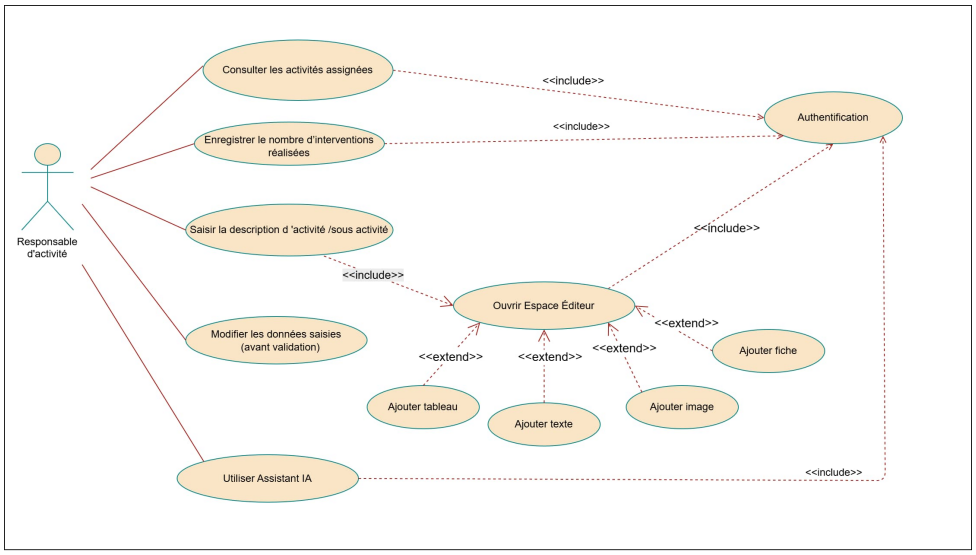


FIGURE 2.4 – Diagramme de cas d'utilisation de responsable d'activité

Description du diagramme de cas d'utilisation "responsable d'activité " :

Le responsable d'activité prend en charge les tâches qui lui sont confiées par le chef de groupe. Il renseigne la description de chaque activité via l'espace éditeur, en y intégrant du texte, des tableaux, des images ou des fiches. Il saisit aussi le nombre d'interventions réalisées et peut corriger ses données avant validation.

2.3.5 Diagramme de cas d'utilisation de chef de département :

Cette figure représente le diagramme de cas d'utilisation de chef de département ainsi que les différentes fonctionnalités que Chef de Département peut réaliser :

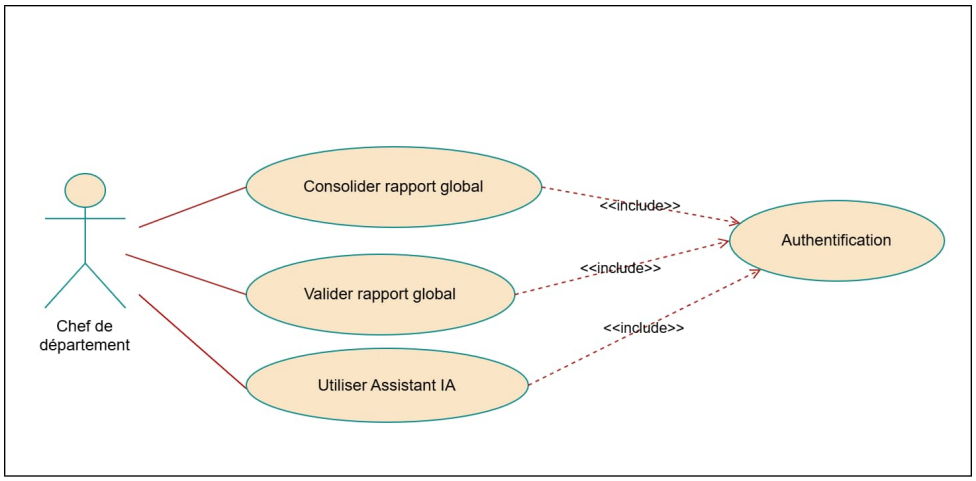


FIGURE 2.5 – Diagramme de cas d'utilisation Chef de département

Description du diagramme de cas d'utilisation "chef de département" :

Le chef de département supervise l'ensemble du processus de reporting au niveau du département. Il accède au rapport global consolidé, examine les données validées des différents groupes, puis décide de le valider ou de le refuser.

2.4 Diagramme de séquence :

Le diagramme de séquence est un diagramme UML qui illustre l'enchaînement des messages entre les objets dans le temps. Il permet de visualiser le déroulement d'un scénario précis d'un système [13].

2.4.1 Diagramme de séquence authentification

Cette figure représente le diagramme de séquence pour l'authentification :

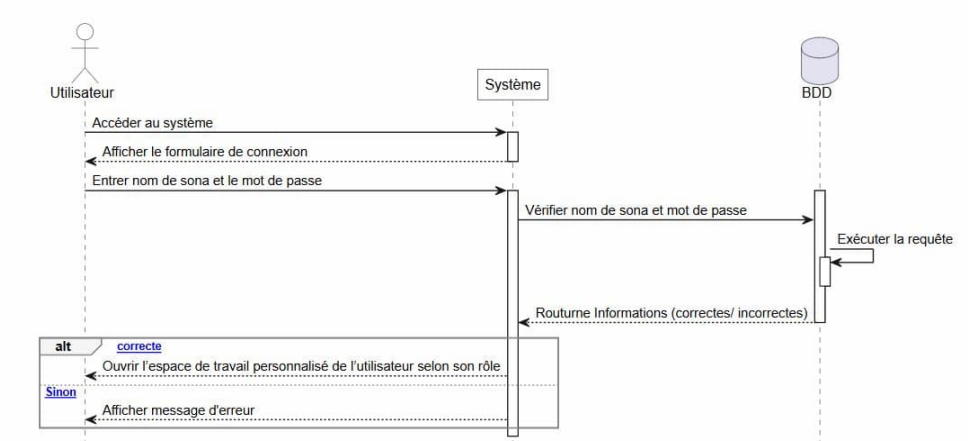


FIGURE 2.6 – Diagramme de séquence authentification

Description du diagramme de séquence " Authentification " :

L'utilisateur saisit son identifiant et son mot de passe. Le système vérifie ces informations dans la base de données : si elles sont correctes, il est redirigé vers son espace personnalisé selon son rôle. Sinon, un message d'erreur s'affiche et l'accès est refusé.

2.4.2 Diagramme de séquence Création d'une Activité / Sous-activité

Cette figure représente le diagramme de séquence création d'une Activité / Sous-activité :

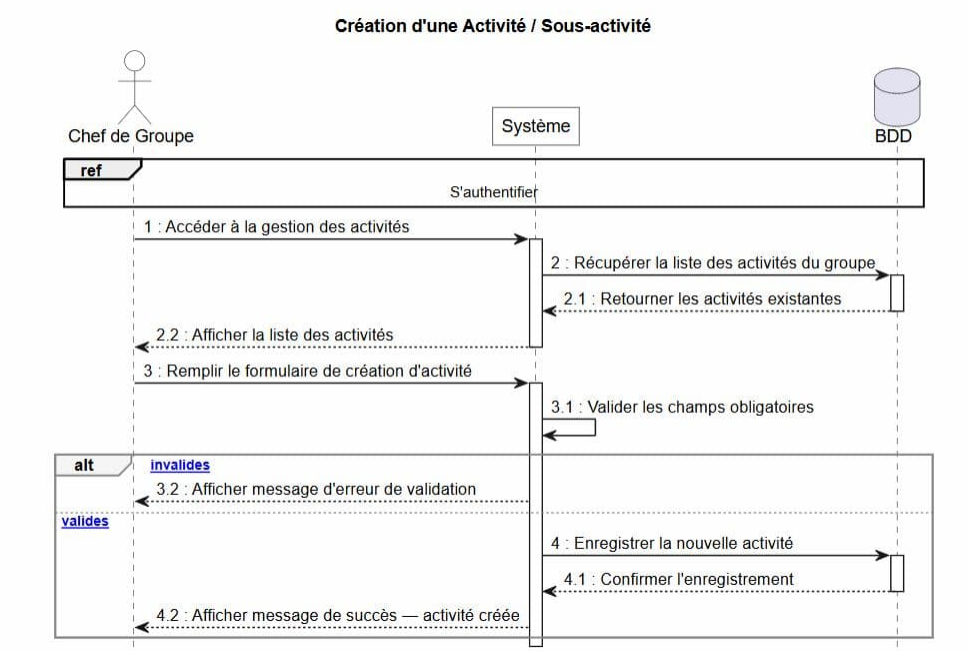


FIGURE 2.7 – Diagramme de séquence création d'une Activité / Sous-activité

Description du diagramme de séquence - création d'une activité / Sous-activité :

Le chef de groupe remplit le formulaire de création après avoir accédé à la gestion des activités. Le système vérifie les champs obligatoires et affiche un message d'erreur si des informations manquent. Si tout est correct, l'activité est enregistrée et un message de succès est retourné.

2.4.3 Diagramme de séquence – Assignation d'un responsable d'activité

Cette figure représente le diagramme de séquence assignation d'un responsable d'activité :

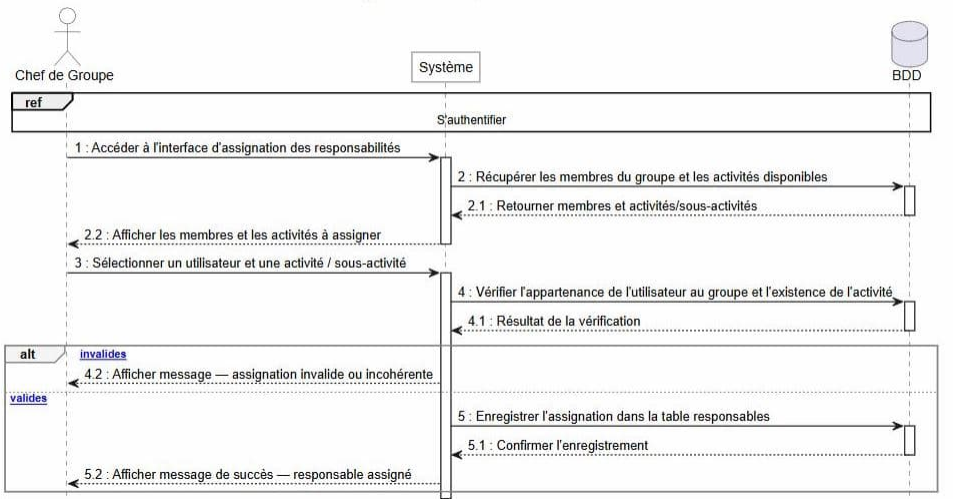


FIGURE 2.8 – Diagramme de séquence assignation d'un responsable d'activité

Description du diagramme de séquence assignation d'un responsable d'activité " :

Le chef de groupe sélectionne un membre de son équipe et l'activité à lui confier. Avant d'enregistrer, le système s'assure que l'utilisateur appartient bien au groupe et que l'activité existe. L'assignation est ensuite confirmée ou rejetée selon le résultat de cette vérification.

2.4.4 Diagramme de séquence – Description d'une activité

Cette figure représente le diagramme de séquence description d'une activité :

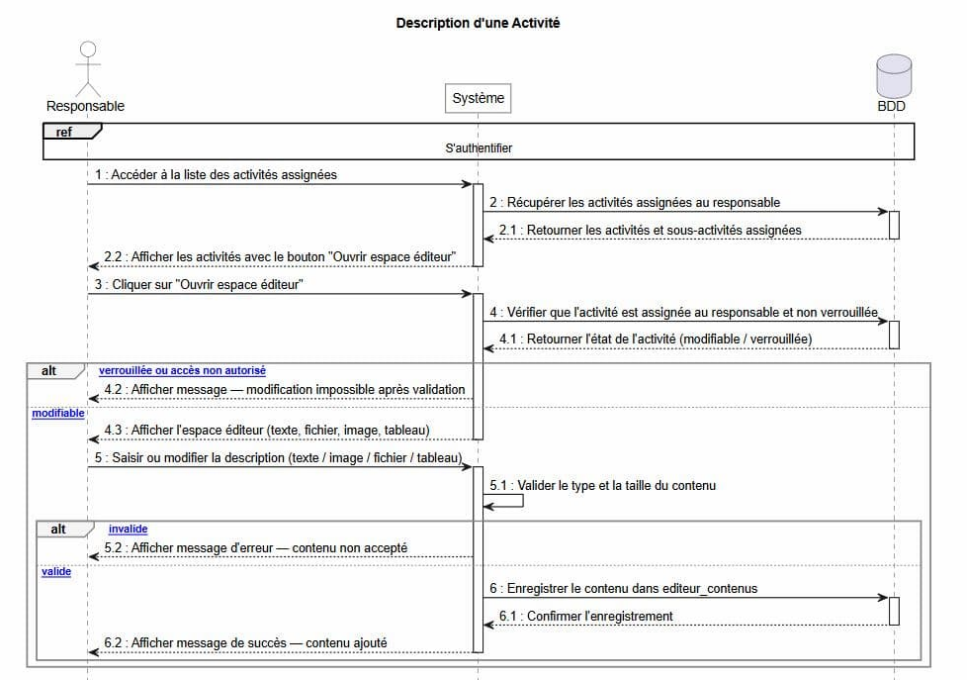


FIGURE 2.9 – Diagramme de séquence description d'une activité

Description du diagramme de séquence description d'une activité :

Le responsable ouvre l'espace éditeur de son activité. Le système vérifie qu'elle lui est bien assignée et qu'elle n'est pas encore verrouillée. Si les conditions sont réunies, il peut y saisir du texte, des images, des tableaux ou des fiches. Le contenu est validé puis enregistré dans la base de données.

2.4.5 Diagramme de séquence – Enregistrement du nombre d'intervention-

Cette figure représente le diagramme de séquence Enregistrement du nombre d'intervention :

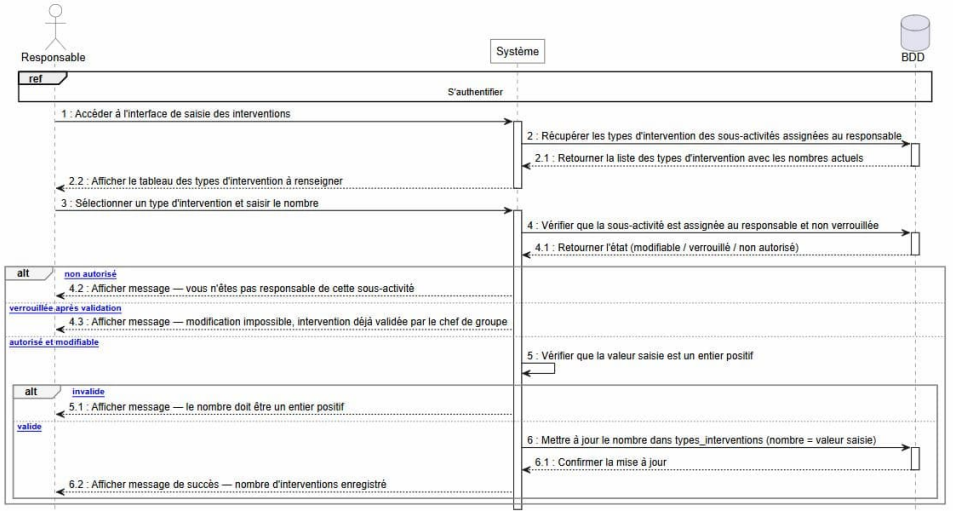


FIGURE 2.10 – Diagramme de séquence Enregistrement du nombre d'intervention

Description du diagramme de séquence enregistrement du nombre d'intervention " :

Le responsable sélectionne le type d'intervention et saisit le nombre correspondant. Le système vérifie que la sous-activité lui est assignée, qu'elle est encore modifiable, et que la valeur saisie est un entier positif. Si tout est valide, le nombre est mis à jour et un message de succès s'affiche.

2.4.6 Diagramme de séquence validation des activités

Cette figure représente le diagramme de séquence Validation des activités :

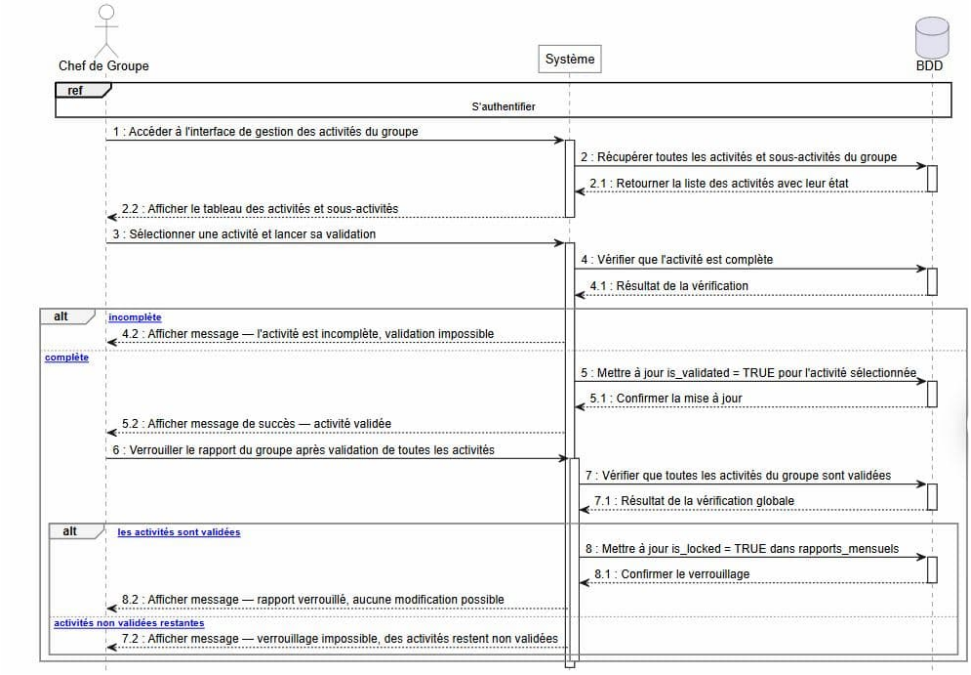


FIGURE 2.11 – Diagramme de séquence Validation des activités

Description du diagramme de séquence validation des activités " :

Le chef de groupe sélectionne une activité et lance sa validation. Le système vérifie qu'elle est complète avant de mettre son statut à jour. Lorsque toutes les activités du groupe sont validées, le rapport mensuel est automatiquement verrouillé et ne peut plus être modifié.

2.4.7 Diagramme de séquence –Validation des interventions -

Cette figure représente le diagramme de séquence validation des interventions :

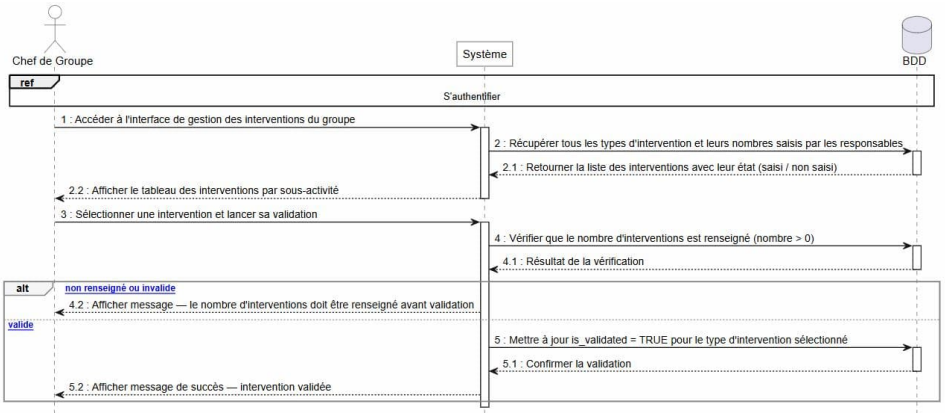


FIGURE 2.12 – Diagramme de séquence Validation des interventions

Description du diagramme de séquence validation des interventions" :

Le chef de groupe consulte le tableau des interventions par sous-activité et choisit celle à valider. Le système vérifie que le nombre d'interventions a bien été renseigné. Si c'est le cas, la validation est enregistrée ; sinon, un message d'erreur invite à compléter la saisie.

2.4.8 Diagramme de séquence validation de rapport global :

Cette figure représente le diagramme de séquence Validation de rapport global :

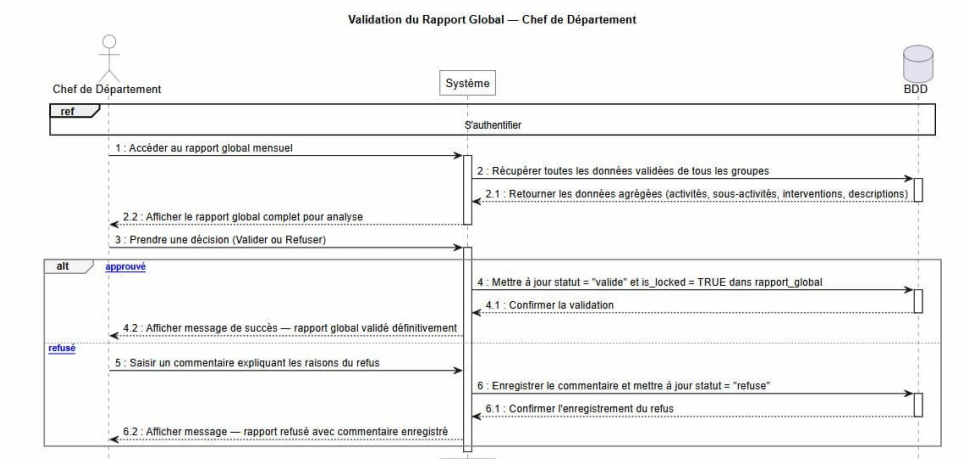


FIGURE 2.13 – Diagramme de séquence Validation du rapport global

Description du diagramme de séquence - Validation du rapport global- " :

Le chef de département accède au rapport global consolidé, qui regroupe les données validées des différents groupes. Il peut ensuite valider le rapport afin de le verrouiller définitivement, ou le refuser en ajoutant un commentaire explicatif. Dans les deux cas, la décision est enregistrée dans la base de données.

2.4.9 Diagramme de séquence –Assistant IA

Cette figure représente le diagramme de l'assistant IA :

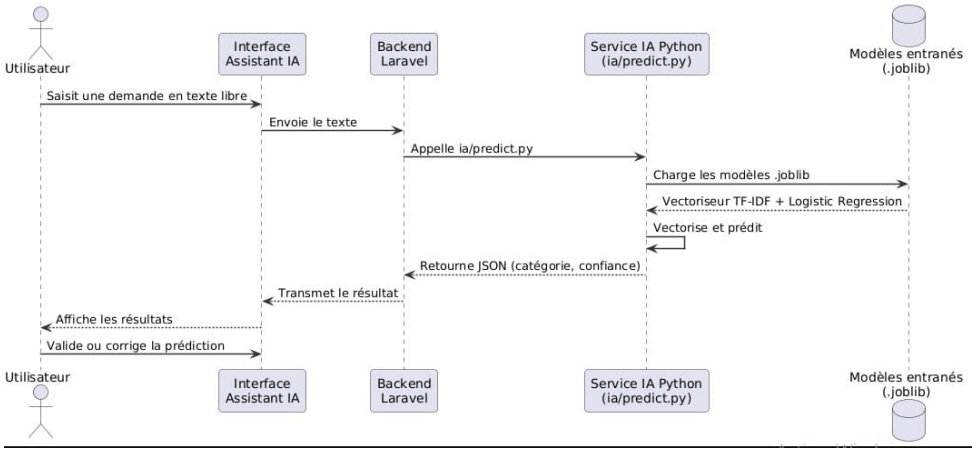


FIGURE 2.14 – Diagramme de séquence Assistant IA

Description du diagramme de séquence Assistant IA " :

L'utilisateur saisit une demande en texte libre depuis l'interface de l'assistant IA. Le backend Laravel reçoit ce texte et appelle le script Python chargé du traitement. Ce script vectorise le texte à l'aide de la méthode TF-IDF, qui attribue à chaque terme un poids selon son importance dans la demande. Ces termes pondérés, représentatifs du contenu saisi, servent ensuite de base au classifieur pour déterminer la catégorie de la demande. La classification est effectuée par régression logistique selon une architecture hiérarchique. Le résultat, contenant la catégorie prédite et le score de confiance, est retourné au backend sous forme JSON, puis affiché à l'utilisateur. Ce dernier peut alors valider la prédiction ou la corriger si nécessaire. Ce module reste une fonctionnalité complémentaire destinée à assister l'utilisateur dans la saisie et le classement des demandes textuelles.

2.5 Diagramme de classe :

Le diagramme de classes est un diagramme UML qui décrit la structure statique d'un système. Il représente les classes, leurs attributs, leurs méthodes ainsi que les relations entre elles [13].

2.5.1 Dictionnaire de données

Le tableau suivant présente le dictionnaire de données associé à notre système :

Légende des types

— **N** : Numérique

— **AN** : Alphanumérique
— **D** : Décimal
— **DT** : Date / Horodatage
— **B** : Booléen (0/1)
— **CF** : Clé étrangère
— – : Taille variable ou non applicable

Classe	Signification	Code	Type	Taille
Groupe				
Groupe	Identifiant du groupe	id	N	–
	Nom du groupe	nom	AN	100
	Code du groupe	code	AN	50
	Description	description	AN	–
Rôle				
Rôle	Identifiant du rôle	id	N	–
	Nom du rôle	name	AN	100
	Identifiant unique (slug)	slug	AN	100
Utilisateur				
Utilisateur	Identifiant de l'utilisateur	id	N	–
	Nom de famille	nom	AN	100
	Prénom	prenom	AN	100
	Nom SONA (identifiant interne)	nom_sona	AN	100
	Adresse e-mail	email	AN	191
	Statut d'activation	is_active	B	1
	Dernière connexion	last_login_at	DT	–
	Groupe d'appartenance (CF)	groupe_id	N	–
Affectation des rôles				
Affectation rôle	Utilisateur concerné (CF)	user_id	N	–
	Rôle attribué (CF)	role_id	N	–
Activité				
Activité	Identifiant de l'activité	id	N	–
	Code de l'activité	code	AN	100

Suite page suivante...

Classe	Signification	Code	Type	Taille
	Libellé de l'activité	libelle	AN	255
	Type (activité/sous-activité)	type	AN	–
	Activité parente (CF)	parent_id	N	–
	Groupe associé (CF)	groupe_id	N	–
	Rapport mensuel lié (CF)	rapport_id	N	–
	Statut de validation	statut	AN	–
Responsable				
Responsable	Identifiant du responsable	id	N	–
	Utilisateur responsable (CF)	user_id	N	–
	Activité concernée (CF)	activite_id	N	–
	Statut de la responsabilité	status	AN	–
	Date d'affectation	assigned_at	DT	–
Catégorie				
Catégorie	Identifiant de la catégorie	id	N	–
	Nom de la catégorie	nom	AN	150
	Groupe associé (CF)	groupe_id	N	–
Type d'intervention				
Type d'intervention	Identifiant du type	id	N	–
	Libellé du type	libelle	AN	255
	Code du type	code	AN	100
	Catégorie associée (CF)	categorie_id	N	–
	Sous-activité liée (CF)	sous_activite_id	N	–
Intervention				
Intervention	Identifiant de l'intervention	id	N	–
	Titre de l'intervention	titre	AN	255
	Description	description	AN	–
	Statut de l'intervention	statut	AN	–
	Niveau de priorité	priorite	AN	–
	Nombre d'occurrences	nombre	N	–
	Date de l'intervention	date_intervention	DT	–

Suite page suivante...

Classe	Signification	Code	Type	Taille
	Rapport mensuel associé (CF)	rapport_id	N	–
	Activité associée (CF)	activite_id	N	–
	Type d'intervention (CF)	type_intervention_id	N	–
	Agent assigné (CF)	assigned_to	N	–
Rapport mensuel				
Rapport mensuel	Identifiant du rapport	id	N	–
	Groupe concerné (CF)	groupe_id	N	–
	Mois du rapport	mois	N	–
	Année du rapport	annee	N	–
	Statut du rapport	status	AN	–
	Date limite de soumission	date_limite	D	–
	Validé par (CF)	valide_par	N	–
Rapport global				
Rapport global	Identifiant du rapport global	id	N	–
	Mois concerné	mois	N	–
	Année concernée	annee	N	–
	Statut du rapport	status	AN	–
	Lien vers le fichier PDF	pdf_url	AN	255
	Validé par (CF)	valide_par	N	–
Détails du rapport global				
Détails rapport global	Rapport global associé (CF)	rapport_global_id	N	–
	Rapport mensuel associé (CF)	rapport_mensuel_id	N	–
	Groupe concerné (CF)	groupe_id	N	–
	Nombre total d'interventions	total_interventions	N	–
	Taux de satisfaction (%)	taux_satisfaction	D	5,2

TABLE 2.3 – Dictionnaire de données

2.5.2 Diagramme de classe :

La figure suivante présente Diagramme de classe de notre système :

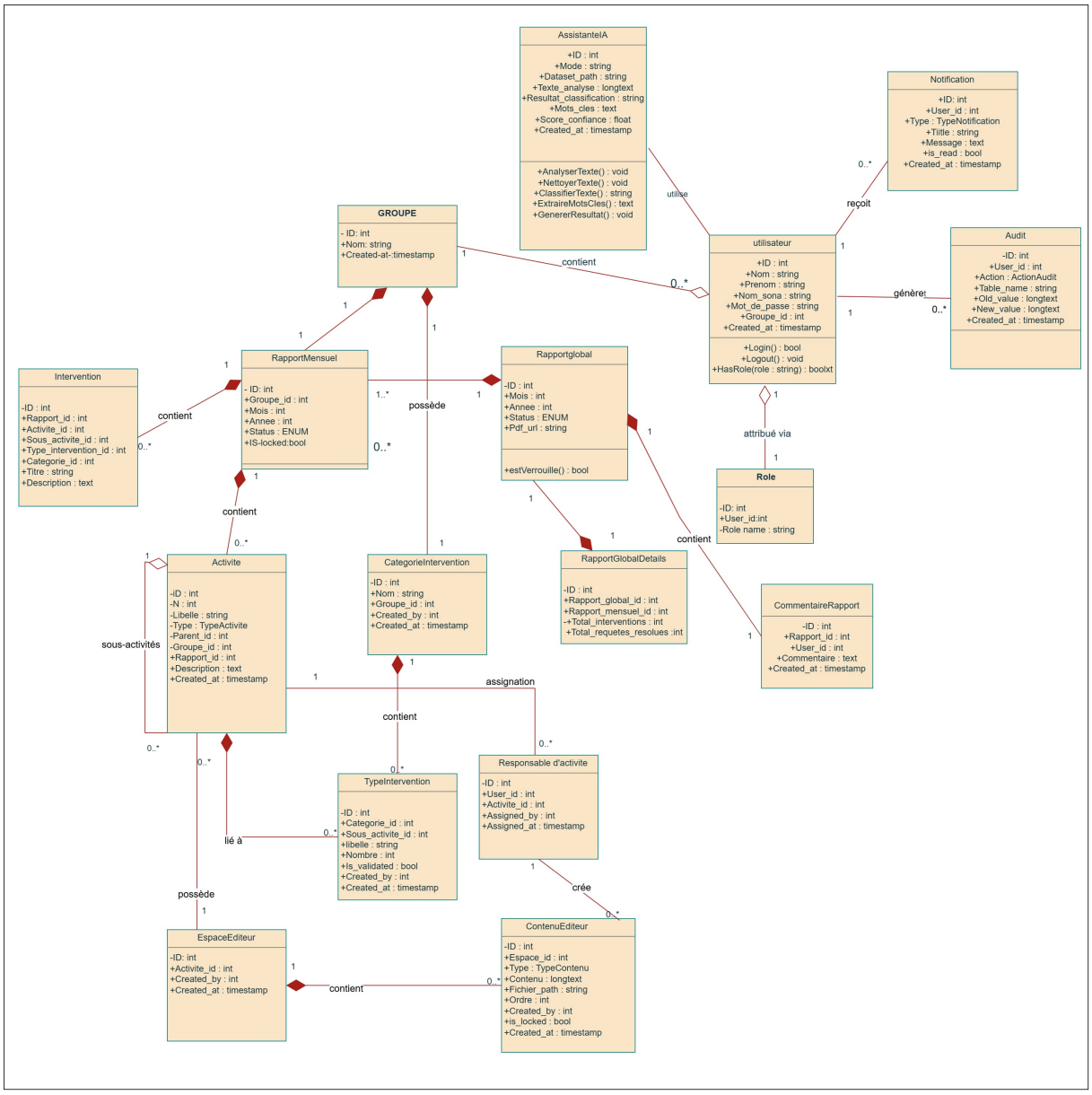


FIGURE 2.15 – Diagramme de classe

2.6 Diagramme d'activité :

Le diagramme d'activité est un diagramme UML qui permet de représenter le déroulement d'un processus ou d'un flux de travail dans un système. Il décrit les différentes activités, les décisions et l'enchaînement des actions jusqu'à la fin du processus [13].

2.6.1 Diagramme d'activité général :

Cette figure représente le diagramme d'activité général de notre système :

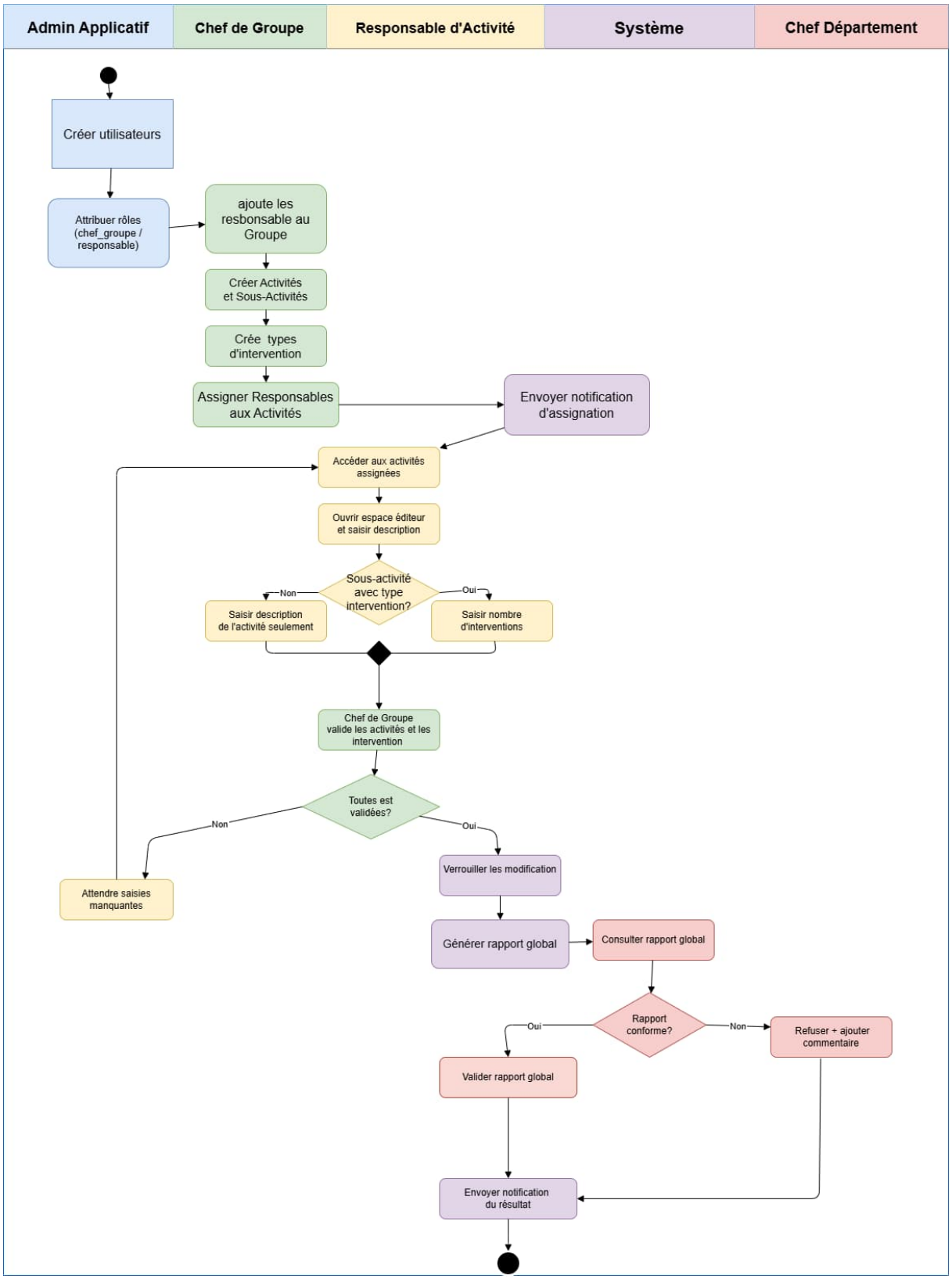


FIGURE 2.16 – Diagramme d'activité général

Description du diagramme d'activité général :

Le processus débute par l'administrateur, qui crée les comptes utilisateurs et leur attribue les rôles appropriés. Le chef de groupe prend ensuite le relais : il organise son équipe, crée les activités et sous-activités, définit les types d'intervention, puis assigne les responsables. Ces derniers reçoivent une notification et accèdent à leurs tâches via l'espace éditeur, où ils saisissent les descriptions et, si nécessaire, le nombre d'interventions réalisées. Le chef de groupe valide ensuite les saisies ; tant que des données manquent, le processus reste en attente. Une fois tout validé, le système verrouille les modifications et génère le rapport global. Le chef de département le consulte et décide de le valider ou de le refuser avec un commentaire. Dans les deux cas, une notification est envoyée.

2.7 Conclusion

Ce chapitre nous a permis de définir les besoins du système et de poser les bases de sa conception. À travers les diagrammes de cas d'utilisation, nous avons identifié les interactions entre les différents acteurs et le système. Les diagrammes de séquence ont détaillé le déroulement des principales opérations, tandis que les diagrammes d'activité ont modélisé le flux des traitements et la logique des processus métier pour chaque fonctionnalité clé. Le diagramme de classes a décrit la structure des données, et le dictionnaire de données a complété cette conception en précisant les attributs de chaque entité. L'ensemble de ces éléments constitue une base solide pour aborder le chapitre suivant, consacré à l'implémentation de l'application.

Chapitre 3

Implémentation

3.1 Introduction

Dans ce dernier chapitre, nous présentons la phase de réalisation de notre système . Nous commençons par présenter l'environnement de développement ainsi que les différents outils et technologies utilisés pour l'implémentation de l'application. Ensuite, nous décrivons les principales fonctionnalités développées à travers les différentes interfaces du système.

3.2 Environnement de développement

3.2.1 Environnement logiciel

WampServer Environnement de développement local sous Windows regroupant Apache, MySQL et PHP. Il a été utilisé pour exécuter l'application en local durant toute la phase de développement [3].



FIGURE 3.1 – logo de wampserver [3]

Visual Studio Code Éditeur de code open source développé par Microsoft. Il a été utilisé pour écrire le code du projet grâce à ses nombreuses extensions dédiées à PHP, JavaScript et Laravel [4].



FIGURE 3.2 – Logo de visual studio code[4]

phpMyAdmin Interface web permettant d'administrer les bases de données MySQL. Elle facilite la création, la modification et la visualisation des tables directement depuis le navigateur [5]



FIGURE 3.3 – Logo phpmyadmin [5]

Postman Outil de test d'API utilisé pour tester les endpoints du back-end et vérifier les réponses avant leur intégration avec le front-end [4].



FIGURE 3.4 – Logo de postman [4]

Google Colab Environnement Python en ligne fourni par Google. Il a été utilisé pour développer et entraîner le modèle d'analyse et de classification automatique des demandes textuelles sans installation locale [6].



FIGURE 3.5 – Logo google colab [6]

Draw.io Outil de modélisation utilisé pour réaliser les diagrammes UML présentés dans le chapitre précédent [7]



FIGURE 3.6 – Logo de draw.io[7]

3.2.2 Langages de développement

HTML5 Langage de balisage utilisé pour structurer les pages de l'application web [8].



FIGURE 3.7 – logo html[8]

CSS3 Langage de style utilisé pour définir la mise en forme et l'apparence des interfaces utilisateur [8].



FIGURE 3.8 – Logo css[8]

JavaScript (Vanilla + Fetch API) Langage de programmation utilisé côté client afin de gérer les interactions avec l'utilisateur. La Fetch API permet d'envoyer des requêtes asynchrones vers le back-end sans recharger la page [8].



FIGURE 3.9 – Logo java script

PHP 8.3 Langage principal du back-end utilisé à travers le framework Laravel [9].



FIGURE 3.10 – Logo PHP [9]

Python Langage utilisé pour développer le module d'intelligence artificielle chargé de l'analyse et de la classification automatique des demandes textuelles [8].



FIGURE 3.11 – logo Python [8]

3.2.3 Frameworks et bibliothèques

Laravel 13 : Framework PHP basé sur le modèle MVC permettant d'organiser le back-end de manière structurée à travers le routage, les contrôleurs, les modèles Eloquent et les migrations de base de données [10]



FIGURE 3.12 – Logo laravel [10]

Laravel Sanctum : Package d'authentification intégré à Laravel utilisant des tokens pour sécuriser les routes de l'API et contrôler l'accès au système [14].

Bootstrap : Framework CSS utilisé pour créer des interfaces cohérentes, responsives et adaptées aux différentes tailles d'écran.



FIGURE 3.13 – Logo bootstrap[11]

Composer : Gestionnaire de dépendances PHP utilisé pour installer et mettre à jour les packages Laravel du projet [9]

npm / Vite : npm est utilisé pour gérer les dépendances JavaScript du front-end, tandis que Vite permet de compiler et d'optimiser les fichiers CSS et JavaScript durant le développement [15].

pandas / numpy / openpyxl : Bibliothèques Python utilisées pour préparer et manipuler les données servant à l'entraînement du modèle d'intelligence artificielle. La bibliothèque openpyxl permet de lire les fichiers Excel contenant les rapports sources [16].

scikit-learn – TF-IDF et Régression Logistique : Bibliothèque de machine learning utilisée pour la vectorisation des textes grâce à la technique TF-IDF ainsi que pour l'entraînement du modèle de régression logistique destiné à la classification automatique des demandes textuelles[17].

joblib : Bibliothèque Python utilisée pour sauvegarder le modèle entraîné et le vectoriseur afin de les réutiliser sans effectuer un nouvel entraînement [18]

3.3 Modèle d'intelligence artificielle

Dans le cadre de ce projet, nous avons intégré, à titre complémentaire, un modèle d'intelligence artificielle chargé d'aider à la classification des demandes textuelles. Il ne constitue pas l'objectif principal du projet et ne remplace pas la décision de l'utilisateur.

3.3.1 Objectif du modèle

Lorsqu'un utilisateur saisit une demande en texte libre, le modèle analyse ce texte et propose automatiquement une catégorie, accompagnée d'un score de confiance. L'utilisateur peut valider ou corriger la prédiction selon sa propre appréciation.

3.3.2 Données utilisées

Le modèle a été entraîné à partir d'un dataset construit dans le cadre de ce projet. Il contient 1 500 demandes textuelles inspirées du contexte du Département Technologie de l'Information de SONATRACH, réparties en 714 activités et 786 interventions.

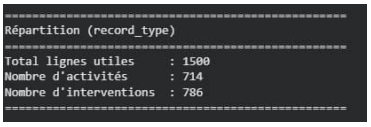


FIGURE 3.14 – Répartition des données utilisées pour le modèle IA

3.3.3 Méthode utilisée

Le traitement des données suit les étapes suivantes :

- **nettoyage du texte :** normalisation Unicode, suppression des accents, conversion en minuscules ;
- **transformation** des textes en vecteurs numériques par la méthode TF-IDF [19, 20] ;
- **classification** par régression logistique selon une architecture hiérarchique [21] ;
- **sauvegarde** des modèles au format `.joblib` ;
- **intégration** à l'application via un script Python appelé par le backend Laravel.

3.3.4 Justification du choix méthodologique

La combinaison TF-IDF et régression logistique a été retenue parce qu'elle est simple, rapide à entraîner, interprétable et adaptée à un dataset de taille limitée [22]. Elle est

suffisante pour un projet de Licence et facile à intégrer dans une application web. Les modèles plus avancés, comme les transformers, nécessitent davantage de données et de ressources de calcul.

3.3.5 Évaluation du modèle

Le modèle a été évalué sur un jeu de test séparé, obtenu par un découpage 80 % / 20 % (1 200 exemples pour l'apprentissage et 300 exemples pour le test). Les métriques utilisées sont l'accuracy, la precision, le recall et le F1-score [21, 22].

Le modèle évalué est le modèle `record_type`, chargé de distinguer les activités et les interventions. Il a obtenu une accuracy de 1,00 sur le jeu de test. Ce score concerne seulement le premier niveau de classification (activité / intervention). Il s'explique par la séparation claire entre ces deux classes dans le dataset et ne signifie pas que l'ensemble du modèle est parfait.

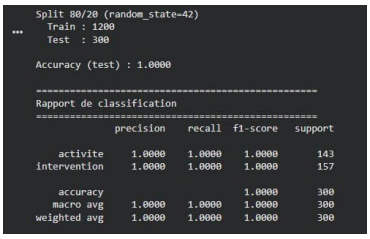


FIGURE 3.15 – Rapport de classification du modèle de type de demande

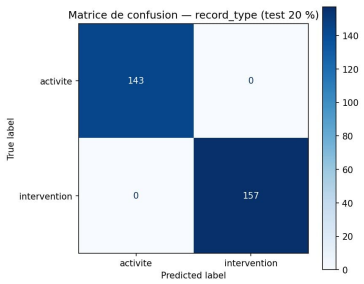


FIGURE 3.16 – Matrice de confusion du modèle de type de demande

Ces résultats montrent le bon fonctionnement du modèle sur le jeu de test utilisé, tout en gardant la validation finale à l'utilisateur.

3.4 Interfaces de l'application

3.4.1 Interface de Connexion :

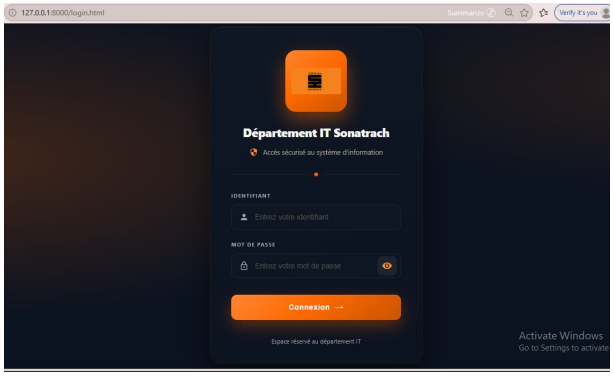


FIGURE 3.17 – Interface de connexion

3.4.2 Interface de Dashboard :



FIGURE 3.18 – interface Dashboard

3.4.3 Interface des interventions :

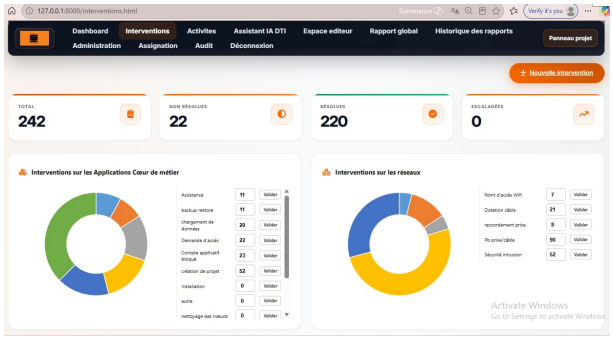


FIGURE 3.19 – interface des interventions

3.4.4 Interface des activités :

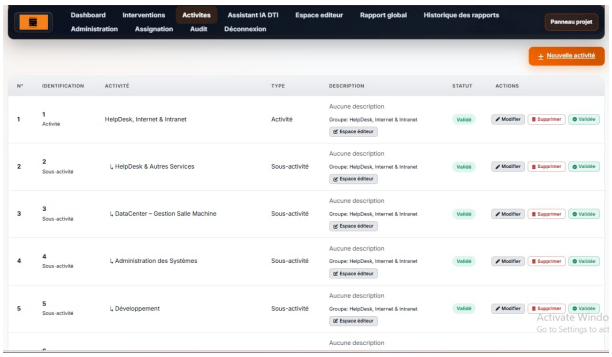


FIGURE 3.20 – interface des activités

3.4.5 Interface de l'espace éditeur :

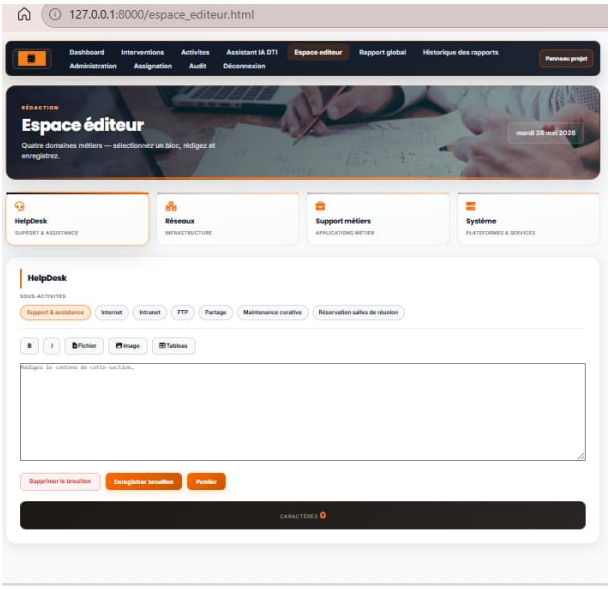


FIGURE 3.21 – Interface de l'espace éditeur

3.4.6 Interface d'administration :

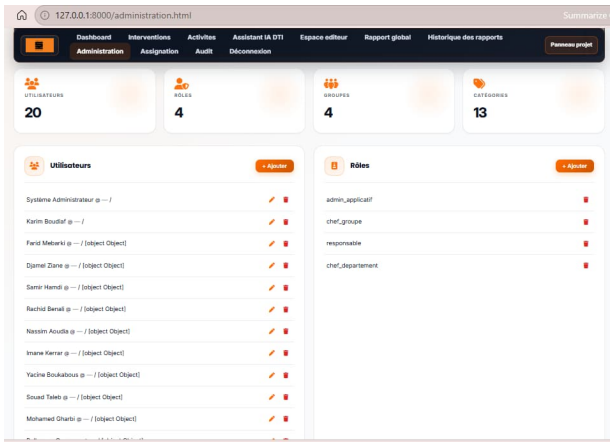


FIGURE 3.22 – interface administration

3.4.7 Interface d'assignation :

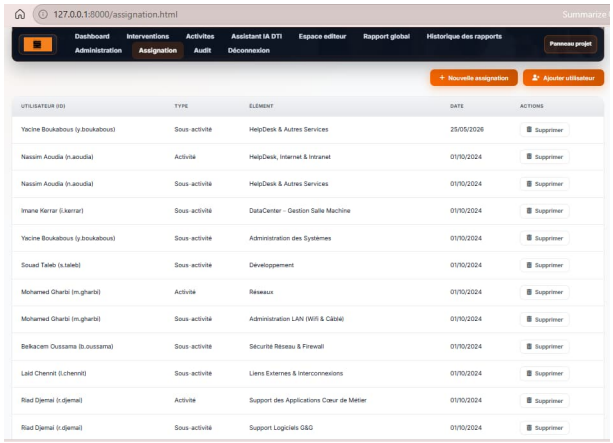


FIGURE 3.23 – Interface assignation

3.4.8 Interface du rapport global :

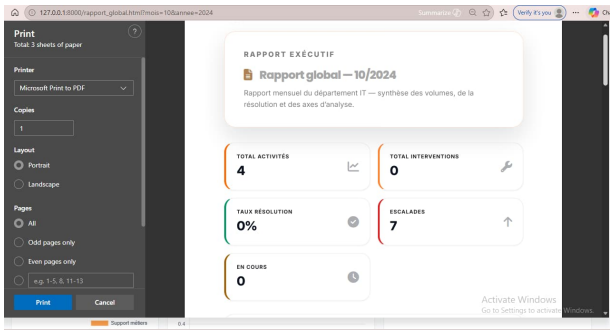


FIGURE 3.24 – Interface du rapport global

3.4.9 Les tests fonctionnels :

Les tests fonctionnels visent à vérifier que fonctionnalité répond correctement aux exigences exprimées Dans ce qui suit, nous présentons quelques tests effectués sur l'application par chef de groupe :

Cas de test – Gestion des activités

Création d'une nouvelle activité par le chef de groupe activité ajoutée et visible dans la liste :

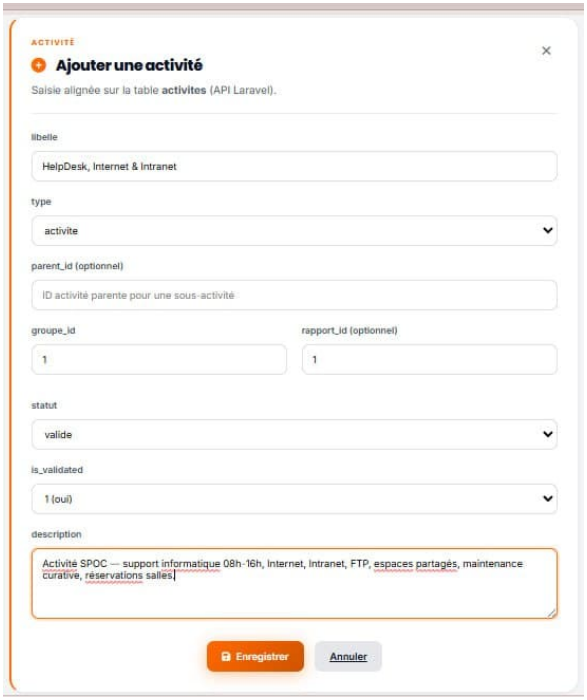


FIGURE 3.25 – Nouvelle activité

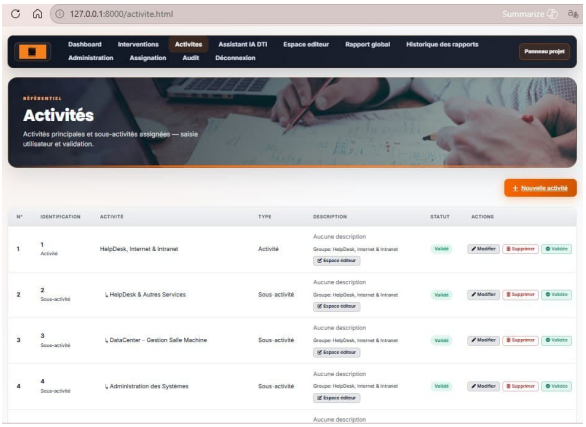


FIGURE 3.26 – Liste des activités

Cas de test-l'assistant IA :

Le chef de groupe saisie d'une demande textuelle liée au HelpDesk Catégorie « HelpDesk » prédite avec score de confiance

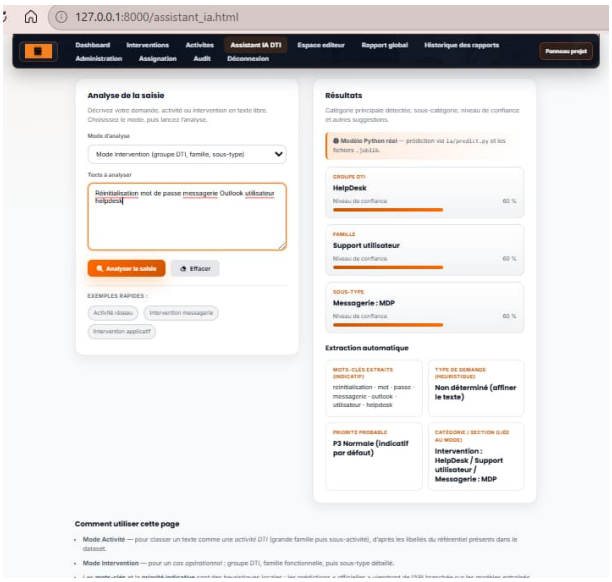


FIGURE 3.27 – Interface de l'Assistant IA

3.5 Conclusion :

Ce chapitre a présenté la phase de réalisation de notre application, en décrivant les outils et technologies choisis ainsi que les principales interfaces développées. Le système réalisé concrétise la conception présentée dans le chapitre précédent et répond aux besoins identifiés lors de l'analyse des besoins.

Conclusion générale :

Ce projet avait pour objectif de répondre à un problème concret observé au sein du Département Technologie de l'Information de SONATRACH : la gestion manuelle et dispersée des rapports d'activité. Cette méthode entraînait des erreurs, un manque de visibilité et une consolidation longue et fastidieuse.

Pour y remédier, nous avons conçu et développé une plateforme web centralisée qui permet aux différents acteurs de saisir leurs activités et interventions dans un espace commun, de suivre l'état des rapports en temps réel et de générer automatiquement un rapport global consolidé. La solution améliore la centralisation des données, réduit les erreurs de saisie, assure une meilleure traçabilité et facilite l'exploitation du contenu textuel grâce à un module d'analyse textuelle et d'extraction automatique de mots-clés basé sur des techniques de traitement automatique du langage naturel.

Ce travail nous a également permis de renforcer nos compétences en développement web avec Laravel, en modélisation UML et en traitement automatique du langage naturel avec Python.

Des améliorations restent envisageables, notamment le renforcement du module d'analyse textuelle avec des modèles plus avancés, l'ajout de tableaux de bord statistiques, l'enrichissement des fonctionnalités selon les retours des utilisateurs et le déploiement de l'application sur l'infrastructure interne de SONATRACH.

Bibliographie

[1] SONATRACH. Documentation officielle sonatrach. https://www.sonatrach.com/, 2026. Consulté le 10/05/2026.

[2] SONATRACH. Rapport annuel 2023. https://sonatrach.com/wp-content/uploads/2024/12/Rapport_Annuel_2023_FR.pdf, 2024. Alger : Groupe SONATRACH, Consulté le 10/05/2026.

[3] Romain Bourdon. Documentation officielle wampserver. https://www.wampserver.com/. Consulté le 21/05/2026.

[4] Microsoft Corporation. Documentation officielle visual studio code. https://code.visualstudio.com/docs. Consulté le 21/05/2026.

[5] phpMyAdmin Project. Documentation officielle phpmyadmin. https://www.phpmyadmin.net/docs/. Consulté le 17/05/2026.

[6] Google LLC. Documentation officielle google colab. https://colab.research.google.com/. Consulté le 21/05/2026.

[7] JGraph Ltd. Documentation officielle draw.io. https://www.drawio.com. Consulté le 17/05/2026.

[8] W3Schools. Documentation officielle html, css, javascript. https://www.w3schools.com/. Consulté le 20/05/2026.

[9] The PHP Group. Documentation officielle php. https://www.php.net/. Consulté le 17/05/2026.

[10] Laravel LLC. Documentation officielle laravel. https://laravel.com/docs. Consulté le 17/05/2026.

[11] Bootstrap Team. Documentation officielle bootstrap. https://getbootstrap.com/. Consulté le 20/05/2026.

[12] SONATRACH. Modèle de rapport d'activité mensuel du département technologies de l'information. Technical report, Division Exploration, Direction Data Management, SONATRACH, 2024. Document interne.

[13] Object Management Group (OMG). Unified modeling language (uml) specification. https://www.omg.org/spec/UML, 2017. Consulté le 10/05/2026.

[14] Laravel LLC. Documentation officielle laravel sanctum. https://laravel.com/docs/sanctum. Consulté le 17/05/2026.

[15] Vite Team. Documentation officielle vite. https://vitejs.dev/. Consulté le 21/05/2026.

[16] pandas Development Team. Documentation officielle pandas. https://pandas.pydata.org/docs/. Consulté le 21/05/2026.

[17] scikit-learn developers. Documentation officielle scikit-learn. https://scikit-learn.org/. Consulté le 21/05/2026.

[18] joblib Contributors. Documentation officielle joblib. https://joblib.readthedocs.io/. Consulté le 21/05/2026.

[19] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1) :11–21, 1972. Consulté en 2026.

[20] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5) :513–523, 1988. Consulté en 2026.

[21] Alexander Genkin, David D. Lewis, and David Madigan. Large-scale Bayesian logistic regression for text categorization. *Technometrics*, 49(3) :291–304, 2007. Consulté en 2026.

[22] Ramesh Nallapati. Discriminative models for information retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 64–71. ACM, 2004. Consulté en 2026.