

UNIVERSITE JOSEPH KI-ZERBO

(UJKZ)

INSTITUT BURKINABE DES ARTS ET METIERS

(IBAM)



**RAPPORT DE STAGE POUR L'OBTENTION DE LA LICENCE
PROFESSIONNELLE**

OPTION : Méthodes Informatiques Appliquées à la Gestion des Entreprises (MIAGE)

Période de stage : 07 Juillet

THEME :

CONCEPTION ET RÉALISATION D'UNE APPLICATION DE
GESTION DES FAMILLES DES VICTIMES D'ATTAQUES
TERRORISTES : CAS DE LA GENDARMERIE NATIONALE

Présenté par KIEMDE Raogo Gérald

Maître de stage

Adjudant SAWADOGO Moussa

Maitre de rapport :

Dr KONE Lydie

SOMMAIRE

Table des matières

SOMMAIRE	Erreur ! Signet non défini.
DÉDICACE	ii
REMERCIEMENTS	iii
LISTES DES SIGLES ET ABREVIATIONS	iv
LISTE DES FIGURES ET GRAPHIQUES	v
LISTE DES TABLEAUX	vi
INTRODUCTION GENERALE	1
CHAPITRE I	2
PRESENTATION DES STRUCTURES DE FORMATION ET D'ACCUEIL	2
I. Présentation de la structure de formation	2
II. Présentation de la structure d'accueil	5
CHAPITRE II : ANALYSE ET CONCEPTION	10
I. ÉTUDE PRÉALABLE	10
II. Expressions des besoins	22
III. Conception globale	43
IV. RÉALISATION	48
CHAPITRE III : BILAN DU STAGE	65
I. DÉROULEMENT DU STAGE ET ACTIVITÉS MENÉES	65
II. OBSERVATIONS ET SUGGESTIONS	67
CONCLUSION GÉNÉRALE	68
BIBLIOGRAPHIE ET WEBOGRAPHIE	69
TABLE DES MATIERES	70
ANNEXES	i
ANNEXE 1 : LANGAGE UML	i
ANNEXE 2 : LA MÉTHODE COCOMO	ii

DÉDICACE

Je dédie ce travail à Dieu Tout-Puissant, dont la grâce et la lumière m'ont accompagné à chaque étape de ce parcours. Sans Sa force et Son soutien, la réalisation de ce projet n'aurait jamais été possible.

À vous, mes chers parents, je rends un hommage profond pour votre amour, votre patience et vos prières constantes, qui ont toujours été pour moi une source de courage et d'inspiration.

À vous, mes amis, je témoigne toute ma gratitude pour votre soutien fidèle et vos encouragements, particulièrement dans les moments difficiles où votre présence a fait toute la différence.

Enfin, à vous, mes professeurs et mentors, je dis merci pour vos précieux conseils, votre encadrement et votre expertise. Vos enseignements resteront pour moi une référence et un guide durable.

REMERCIEMENTS

Nous ne saurions poursuivre ce travail sans exprimer d'abord nos vifs et sincères remerciements à toutes les personnes qui, de près ou de loin, ont contribué à la réussite de ce stage et à la réalisation de ce projet.

Nous tenons particulièrement à remercier :

- ❖ Dieu Tout-Puissant, dont la miséricorde et la grâce nous ont accordé la santé, la force et la persévérance nécessaires pour mener à bien ce travail dans les délais ;
- ❖ Adjudant Moussa SAWADOGO, notre maître de stage, pour son encadrement, ses conseils éclairés, ses critiques constructives, ses encouragements et son accompagnement tout au long de notre mission ;
- ❖ Pr. Gilbert BAYILI, Directeur de l'IBAM, ainsi que l'ensemble du corps professoral, pour la qualité de la formation reçue et les conseils précieux dispensés durant ces trois années de formation ;
- ❖ Dr. Lydie KONE, notre Directeur de rapport, pour sa disponibilité, son soutien académique et ses conseils pertinents qui ont grandement facilité la rédaction de ce document ;
- ❖ Tout le personnel de la Gendarmerie , pour leur collaboration, leur disponibilité et leur esprit d'équipe, qui ont contribué à créer un cadre de travail agréable et enrichissant ;
- ❖ Nos parents, amis et camarades, dont le soutien moral, matériel et affectif a été d'un apport inestimable tout au long de cette expérience ;
- ❖ Enfin, une pensée particulière à tous les soldats courageux tombés au front. Nous honorons leur bravoure, leur engagement et leur sacrifice suprême pour la patrie.

LISTES DES SIGLES ET ABREVIATIONS

Sigles/ Abréviations	Significations
2TUP	Two (2) Tracks Unified Process
ABF	Assurance Banque Finance
ADB	Assistance de Direction Bilingue
CCA	Comptabilité Contrôle Audit
COCOMO	Constructive Cost Model
CSAF	Chef du Service Administratif et Financier
CSS	Cascading Style Sheets
CU	Cas d'utilisation
HTML	HyperText Markup Language
http	HyperText Transfer Protocol
IHM	Interface Homme-Machine
LMD	Licence Master Doctorat
MAGE	Master en Administration et Gestion des Entreprises
MBF	Master en Banque Finance
MCCA	Master en Comptabilité-Contrôle-Audit
MG	Marketing et Gestion
MID	Marketing et Innovation Digital
MVT	Model Vue Template
RI-MT	Réseau Informatique-Multi média et Télécom
SGBD	Système de Gestion de Base de Données
SQL	Structured Query Language
TIC	Technologie de l'Information et de la Communication
UJKZ	Université Joseph KI-ZERBO
UML	Unified Modeling Language
WWW	World Wide Web

LISTE DES FIGURES ET GRAPHIQUES

Figure 1 : Organigramme de l'IBAM.....	3
Figure 2 :Organigramme de la Gendarmerie Nationale	9
Figure 3 : schéma présentant le processus 2TUP	18
Figure 4 : Planning de réalisation du projet	21
Figure 5 : Diagramme des cas d'utilisation	27
Figure 6 : Diagramme de séquence du cas d'utilisation « S'authentifier ».....	33
Figure 7 : Diagramme de séquence du cas d'utilisation « Créer Fiche Victime »	35
Figure 8 : Diagramme de séquence du cas d'utilisation « Ajouter Famille »	36
Figure 9 : Diagramme d'activité pour le CU « S'authentier »	37
Figure 10 : Diagramme d'activité pour le CU « Créer Fiche Victime »	38
Figure 11 : Schéma représentant l'architecture deux tiers	40
Figure 12 : Schéma représentant l'architecture trois tiers	41
Figure 13 : Diagramme de classes.....	46
Figure 14 : Diagramme de déploiement.....	47
Figure 15 : résultat du test de création d'un utilisateur avec les champs requis	57
Figure 16 : résultat du test de création d'un utilisateur avec un champ manquant	57
Figure 17 : résultat du test de création d'une fiche victime	58
Figure 18 : résultat du test de validation d'une demande d'aide.....	58
Figure 19 : Ecran de connexion.....	61
Figure 20 : tableau de bord Agent	62
Figure 21 : tableau de bord Assistant Social	62
Figure 22 : Écran de suivi des aides	63
Figure 23 : Écran de gestion des victimes.....	64

LISTE DES TABLEAUX

Tableau 1 : Comparaison des méthodologies de développement.....	15
Tableau 2 : Comparaison entre quelques langages de modélisation	19
Tableau 3 : Liste des cas d'utilisation	26
Tableau 4 : Description textuelle du cas d'utilisation “S’authentifier”	28
Tableau 5 : Description textuelle du cas d'utilisation “S’authentifier”	30
Tableau 6 : Description textuelle du cas d'utilisation “Ajouter Famille”	31
Tableau 7 : Comparaison des différentes architectures de développement.....	40
Tableau 8 : Dictionnaire des données.....	43
Tableau 9 : Liste comparative des SGBD	49
Tableau 10 : Tableau comparatif des serveurs d'applications.....	51
Tableau 11 : Comparaison des méthodes d'estimation de coût.....	59
Tableau 12 : Coût total du projet.....	60
Tableau 13 : Formule de calcul COCOMO	ii

INTRODUCTION GENERALE

Depuis quelques années, notre pays traverse une période difficile. Les attaques terroristes, de plus en plus fréquentes, ont bouleversé la vie de nombreuses familles. Certaines ont perdu un proche. D'autres se retrouvent dans le silence, oubliées, sans véritable suivi. Ces drames humains, lourds de conséquences, exigent une réponse adaptée.

Dans ce contexte tendu, la Gendarmerie nationale occupe une place centrale. Elle assure la sécurité, mais elle veille aussi, autant que possible, au soutien des familles touchées. Pourtant, les démarches actuelles restent lentes. La gestion repose encore sur des fichiers papier, des échanges oraux, des déplacements. Résultat : un système lourd, peu réactif, et parfois inefficace.

Or, aujourd'hui, la technologie permet d'aller plus vite, de mieux organiser, de centraliser l'information. Pourquoi ne pas en tirer parti pour alléger le travail des agents et mieux accompagner les familles ? C'est cette réflexion qui a motivé le projet confié durant notre stage.

L'objectif de ce projet est de concevoir et de mettre en place un système informatisé permettant une gestion efficace, fiable et sécurisée des dossiers des familles des victimes d'attaques terroristes. Cette solution vise à répondre aux limites des méthodes traditionnelles de suivi, en offrant aux agents de la Gendarmerie un outil moderne qui facilite le traitement des informations, accélère la prise en charge des familles et garantit une meilleure traçabilité des actions.

Ce rapport a pour but de présenter, avec précision et clarté, l'ensemble du travail réalisé autour de la conception et de la mise en place d'un système de gestion dédié aux familles des victimes d'attaques terroristes. Pour organiser notre réflexion, nous allons structurer ce document en trois grands volets.

Le premier chapitre est consacré à la présentation de notre institut de formation (IBAM) ainsi que de la structure d'accueil (la Gendarmerie nationale). Dans le second chapitre, nous entrons dans le vif du sujet : l'analyse des besoins, les choix techniques, et la démarche suivie pour concevoir le système. Enfin, le troisième chapitre revient sur l'expérience vécue durant le stage. On y évoque les difficultés rencontrées, les solutions apportées, et les leçons tirées au fil du projet. Chaque partie vise à éclairer un aspect du processus. Ensemble, elles offrent une vue d'ensemble cohérente sur l'objectif poursuivi : moderniser et simplifier une mission à la fois sensible et humaine.

CHAPITRE I

PRESENTATION DES STRUCTURES DE FORMATION ET D'ACCUEIL

Ce chapitre sera consacré à la présentation de nos structures de formation et d'accueil. Tout d'abord, nous présenterons notre structure de formation qui est l'Institut Burkinabé des Arts et Métiers, puis par la suite nous ferons une présentation de notre structure d'accueil qui est la Gendarmerie Nationale

I. Présentation de la structure de formation

L'Institut Burkinabè des Arts et Métiers (IBAM) est un établissement d'enseignement professionnel implanté à Ouagadougou, dans la zone de Somgandé. Créé en janvier 2000, il incarne la volonté de l'Université Joseph KI-ZERBO de renforcer l'orientation professionnelle des filières de formation. Doté d'une organisation interne rigoureusement structurée, l'IBAM poursuit plusieurs objectifs clairement définis. Dans cette partie, nous présenterons d'abord ses objectifs, ensuite son organisation interne, et enfin les différentes filières de formation qu'il propose.

1. Objectifs de l'IBAM

Créé pour former autrement, l'IBAM porte une mission claire : offrir un enseignement de qualité, en phase avec les besoins réels du marché de l'emploi. Il mise sur un capital humain compétent, capable de fournir des cadres moyens et supérieurs dans plusieurs domaines d'activité. Cette orientation, volontaire et réfléchie, s'inscrit dans une dynamique : améliorer l'insertion professionnelle et prolonger la vision de l'Université Joseph KI-ZERBO.

2. Organisation

L'organisation de l'IBAM repose sur deux principales structures : l'organe statutaire et l'organe d'exécution. Le fonctionnement de l'institut, notamment en ce qui concerne l'encadrement pédagogique et l'évaluation des compétences, est régi par des arrêtés ministériels. La direction de l'établissement est assurée par un directeur assisté d'un directeur adjoint, tous deux élus pour un mandat de deux (2) ans renouvelables. La Figure 1 ci-dessous présente de façon claire et détaillé l'organisation de l'IBAM.

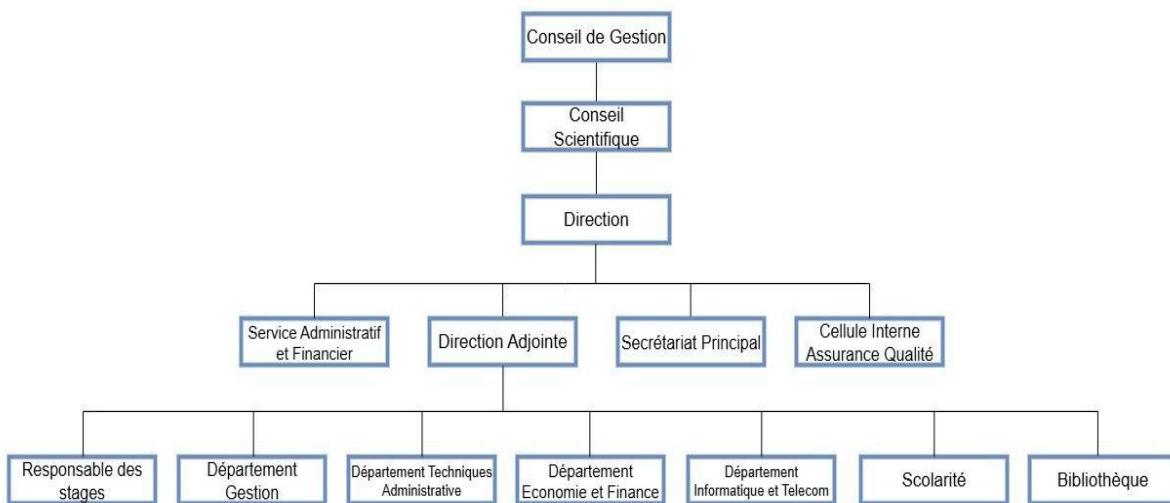


Figure 1 : Organigramme de l'IBAM

a. Organe statutaire

L'organe statutaire de l'Institut Burkinabè des Arts et Métiers (IBAM) comprend deux instances principales : le Conseil de gestion et le Conseil scientifique.

Le **Conseil de gestion** représente l'organe décisionnel majeur de l'institut. Il est constitué des membres suivants : le Directeur de l'IBAM (président du Conseil), le Directeur adjoint, les Coordonnateurs de sections, le Secrétaire principal, le Chef du Service Administratif et Financier (CSAF), les enseignants permanents, deux (02) représentants syndicaux des étudiants un représentant du personnel administratif, technique, ouvrier et de service (ATOS). Ce conseil se réunit tous les trois (03) mois, sauf en cas de session extraordinaire convoquée par son président. Ses principales missions sont :

- ❖ définir le règlement intérieur de l'institut,
- ❖ proposer le budget de fonctionnement,
- ❖ prendre des décisions d'ordre administratif et financier en conformité avec les textes en vigueur
- ❖ proposer la modification des statuts de l'IBAM.

En complément du Conseil de gestion, qui traite principalement des aspects administratifs et financiers, l'IBAM dispose également d'un **Conseil scientifique**, chargé des orientations pédagogiques et académiques de l'institut.

Le Conseil scientifique regroupe : le Directeur de l'IBAM (président du Conseil), le Directeur adjoint, les Coordonnateurs de section, les enseignants de rang A et B de l'institut. Comme le Conseil de gestion, il se réunit selon la même périodicité et ses séances sont organisées dans les mêmes conditions. Toutefois, son secrétariat est assuré par l'un des Coordonnateurs de section. Ses missions sont les suivantes :

- ❖ proposer une organisation générale des enseignements de l'institut ;
- ❖ proposer l'ouverture, la fusion ou la fermeture de filières ;
- ❖ créer des titres, diplômes et d'examiner les équivalences pédagogiques ;
- ❖ proposer la création de nouveaux postes et d'apprécier les orientations de la recherche au sein de l'institut.

b. Organe d'exécution

L'organe exécutif a pour mission principale de mettre en application les décisions prises par l'organe statutaire, et il est principalement composé de la Direction de l'IBAM. La Direction, placée sous la direction du Directeur de l'IBAM, est chargée de superviser les opérations internes et de définir la politique globale de l'institut. Elle joue un rôle central en coordonnant toutes les activités des services de l'institut, en maintenant la réputation de l'institut, et en concrétisant les décisions, projets et programmes tout en respectant les règlements internes. Le Directeur de l'IBAM reçoit l'assistance d'une Secrétaire de direction dans l'accomplissement de ses tâches.

3. Filières de formations

En vue d'atteindre les objectifs cités précédemment, l'IBAM offre des formations dans plusieurs filières. Ces filières sont reparties en deux (2) groupes selon les diplômes, à savoir le groupe des Licences professionnelles et le groupe des Masters.

❖ Licences professionnelles

Les filières en formations initiales en licences professionnelles sont :

- ✓ Assurance Banque Finance (ABF) ;
- ✓ Assistant de Direction/Bilingue (ADB) ;
- ✓ Assistant de Direction Comptable (ADC) ;
- ✓ Comptabilité-Contrôle-Audit (CCA) ;
- ✓ Informatique, option Méthodes Informatiques Appliquées à la Gestion (MIAGE) ;
- ✓ Informatique, option Réseaux et Télécommunications (RI-MT) ;

- ✓ Marketing et Innovation Digitale (MID).

❖ Masters

En master, l'IBAM offre cinq (05) filières de formation qui sont :

- ✓ Master en Administration et Gestion des Entreprises (MAGE) ;
- ✓ Master en Ingénierie Bancaire et Financière (IBF) ;
- ✓ Master en Comptabilité-Contrôle-Audit (MCCA) ;
- ✓ Master en Informatique, option Ingénierie des Systèmes d'Information en Entreprise (M2ISIE) ;
- ✓ Master en Informatique, option Sécurité Informatique (MISI).

La filière MIAGE a été créée dans l'optique de répondre aux besoins croissants des entreprises en cadres compétents dans le domaine des Technologies de l'Information et de la Communication (TIC). Cette filière accueille en première année les bacheliers des séries C, D et E ayant passé avec succès le test de sélection ou ayant été acceptés en tant qu'auditeurs libres.

Les étudiants en fin de cycle en MIAGE doivent mettre en pratique les connaissances acquises en classe en effectuant un stage d'une durée d'au moins trois (03) mois suivi d'une soutenance publique. C'est pour répondre à cette exigence académique que nous avons effectué notre stage pratique au sein de la Gendarmerie Nationale que nous présenterons dans les lignes qui suivent.

II. Présentation de la structure d'accueil

1.1. Historique

La Gendarmerie Nationale du Burkina Faso est héritière de la Gendarmerie française dont les origines sont très anciennes. Le premier poste de Gendarmerie a été implanté le 02 juin 1939 à Ouagadougou et régularisé le 29 août 1946. Bobo-Dioulasso, ville de la colonie ivoirienne bénéficie de son poste de gendarmerie le 23 février 1942. Après la reconstitution de l'ancienne colonie de la Haute Volta en septembre 1947, L'arrêté n°4991/CM/GEND a créé une section de Gendarmerie le 05 décembre 1947 avec pour siège Ouagadougou relevant du détachement central de l'Afrique Occidentale Française (AOF). La gendarmerie nationale de la Haute Volta venait de voir le jour. Puis ont suivi les différentes créations de postes et brigades sur toute l'étendue du territoire. Après la proclamation de l'indépendance le 05 août 1960, la Gendarmerie Nationale de la Haute Volta a été créée par la loi n°74/60/AN du 03 août 1960. Après avoir été commandée par les officiers de l'ancienne métropole jusqu'au 28 octobre 1961,

le Capitaine Yemdaogo Michel DEME sera le premier voltaïque à prendre le commandement de la compagnie de Ouagadougou.

1.2. Les Missions

La Gendarmerie est une force créée pour garantir la sécurité des personnes et des biens, assurer le maintien de l'ordre public et l'exécution des lois. Elle participe aussi à la défense militaire de la nation et assure la défense opérationnelle du territoire (DOT) en cas d'état de siège ou de guerre. Par ailleurs, elle participe également au maintien de l'ordre dans les Etats étrangers dans le cadre des missions de la paix militaire de l'ONU.

La Gendarmerie mène ses actions à travers :

la Police administrative : surveillance du territoire, protection des axes routiers, sécurité publique, maintien de l'ordre ;

la Police judiciaire : enquêtes judiciaires, exécution des réquisitions de justice ;

la Police militaire : suivi des militaires en permission, recherches des déserteurs ;

la police de la route ou de la circulation : La surveillance des axes routiers et des pistes rurales à l'occasion des patrouilles, sont des actes majeurs de la Gendarmerie dans le cadre de la prévention des infractions.

1.3. Organisation de la Gendarmerie Nationale

L'organisation de la Gendarmerie au Burkina Faso, peut être présentée ainsi qu'il suit :

- l'Etat-major ;
- les légions de gendarmerie ;
- les groupements de gendarmerie départementale ;
- les groupements de gendarmerie mobile.

1.3.1. L'Etat-major

L'Etat-major est le centre du commandement de l'institution avec à sa tête le Chef d'Etat-major de la Gendarmerie Nationale (CEMGN). Hormis le cabinet, il est composé de douze (12) directions dont :

- ✓ la Direction de la Logistique (DL) ;
- ✓ la Direction de l'Organisation et de l'Emploi (DOE) ;

- ✓ la Direction des Transmissions (DT) ;
- ✓ la Direction des Ressources Humaines (DRH) ;
- ✓ la Direction de la Police Judiciaire (DPJ) ;
- ✓ la Direction des Services de Santé (DSS) ;
- ✓ la Direction du Centre des Recherches et du Traitement de l'Information (DCRTI) ;
- ✓ la Direction de la Communication et des Relations Publiques (DCRP) ;
- ✓ la Direction des Sports (DS) ;
- ✓ la Direction des Services Informatiques (DSI) ;
- ✓ la Direction de l'Intendance Militaire (DIM) ;

En plus des directions on retrouve également au même niveau de la hiérarchie :

- ✓ le Commandement des Ecoles et Centres de Perfectionnement de la Gendarmerie Nationale (CECPGN) ;
- ✓ le Centre Opérationnel de la Gendarmerie Nationale (COGN).

1.3.2. Les légions de gendarmerie nationale (RGN)

Il existe six (06) légions de Gendarmerie créées en fonction des régions militaires et la légion spéciale. Ces légions sont :

- la 1ère légion implantée à Kaya
- la 2e légion implantée à Bobo-Dioulasso
- la 3e légion implantée à Ouagadougou
- la 4e légion implantée à Dori
- la 5e légion implantée à Déoudougou
- la 6e légion implantée à Fada N'Gourma
- la légion spéciale implantée à Ouagadougou

Les légions ont autorité sur les groupements mobiles et départementaux installés en leurs seins.

1.3.3. Les groupements de gendarmerie départementale (GD)

La Gendarmerie Départementale assure essentiellement les missions de police judiciaire et de police administrative. Elle assure également une surveillance continue des zones rurales. Elle a sous son commandement les compagnies et les brigades de gendarmerie. On peut dénombrer dix (10) groupements départementaux répartis comme suit :

- ✓ 1e légion : les groupements départementaux de Kaya et de Ouahigouya ;
- ✓ 2e légion : les groupements départementaux de Bobo-Dioulasso et de Gaoua ;
- ✓ 3e légion : les groupements départementaux de Ouagadougou et de Koudougou ;
- ✓ 4e légion : le groupement départemental de Dori ;
- ✓ 5e légion : le groupement départemental de Déodougou ;
- ✓ 6e légion : les groupements départementaux de Fada N'Gourma et de Tenkodogo.

Et quatre (04) groupements spécifiques à savoir :

- ✓ l'Unité Spéciale d'Intervention de la Gendarmerie Nationale (USIGN) ;
- ✓ le Groupement d'Escorte et d'Honneur de la Gendarmerie Nationale (GEHGN) ;
- ✓ le Groupement de Sécurité et d'Intervention de la Gendarmerie Nationale (GSIGN) ;
- ✓ le Groupement des Transports Aéroportuaires (GTA).

1.4. Présentation du service informatique de la Gendarmerie

La Direction des Services Informatiques de la Gendarmerie Nationale (DSIGN) est l'une des directions de l'EMGN, et se compose de :

- un Secrétariat ;
- un Service Exploitation et formation ;
- un Service Réseau et Maintenance ;

Le Service Exploitation et Formation qui est le service au sein duquel nous effectuons notre stage. Il a pour missions essentielles :

1. L'exploitation de toutes les applications fonctionnelles : administration des systèmes d'information et des serveurs, appui technique aux services qui les utilisent, sauvegarde et restauration des données ;
2. La formation des utilisateurs sur les différents outils de bureautique et les applications métiers ;
3. La vulgarisation de l'outil informatique et des bonnes pratiques dans le domaine des TIC (Technologies de l'information et de la communication).

1.5. Organigramme de la Gendarmerie Nationale

L'organisation de la GN se présente comme l'indique la figure ci-dessous :

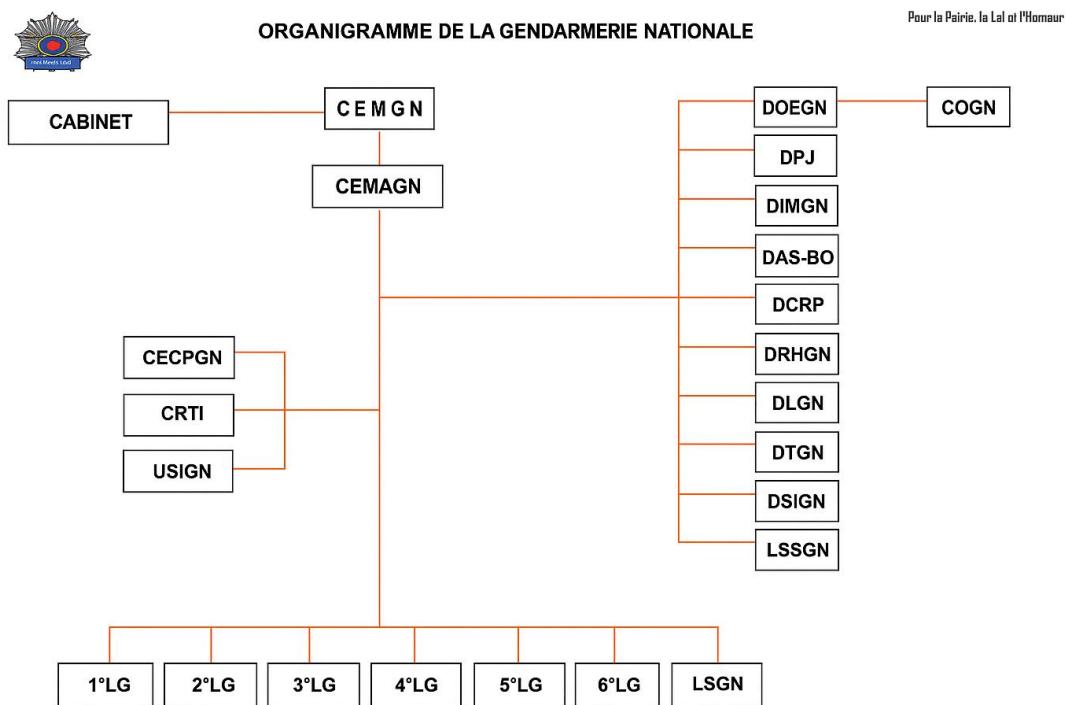


Figure 2 :Organigramme de la Gendarmerie Nationale

CHAPITRE II : ANALYSE ET CONCEPTION

Après avoir présenté nos structures de formation et d'accueil, nous abordons à présent le cœur de notre travail. Ce chapitre expose d'abord le thème retenu, puis la méthode d'analyse adoptée pour mener la conception du projet. Nous y décrivons également les acteurs impliqués et le planning prévisionnel qui a guidé notre démarche. Ensuite, une attention particulière sera accordée à l'expression des besoins, à travers une étude technique approfondie et une conception détaillée. Enfin, nous présenterons l'ensemble des éléments nécessaires à la réalisation effective de l'application destinée à la gestion des familles des victimes d'attaques terroristes.

I. ÉTUDE PRÉALABLE

Cette partie du chapitre est destinée, à une description détaillée du thème, à l'ensemble des éléments d'analyse et de conception entrant dans la réalisation du projet ainsi qu'au planning prévisionnel de réalisation

1. Présentation du Thème

La présentation du thème suivra un chemin clair. Nous commencerons par une partie préliminaire afin d'expliquer certains points essentiels liés à l'intitulé. Puis, nous mettrons en évidence la problématique qui justifie la pertinence du sujet. Avant d'aborder les résultats attendus, nous présenterons les objectifs qui guident la réalisation du projet.

a. Préliminaire

Dans le contexte actuel marqué par la montée des défis sécuritaires et la transformation numérique des institutions, la nécessité d'optimiser les processus internes s'impose avec force. Les organisations, qu'elles soient publiques ou privées, cherchent à gagner en efficacité, en transparence et en réactivité afin de mieux répondre aux attentes de leurs usagers.

À la Gendarmerie nationale, la gestion des familles des victimes d'attaques terroristes constitue un processus à la fois sensible et essentiel. Pourtant, ce processus repose encore largement sur des méthodes traditionnelles, souvent lentes et exposées à des erreurs. Ce fonctionnement entraîne des retards dans le traitement des dossiers, complique la coordination entre les services et peut, par ricochet, affecter l'accompagnement des familles déjà fragilisées

par la perte de leurs proches. C'est dans cette optique que s'inscrit notre projet. Notre objectif est de concevoir et de réaliser une application informatique permettant d'automatiser et de centraliser la gestion des familles des victimes d'attaques terroristes. Une telle solution vise non seulement à réduire les délais de traitement et à limiter les erreurs humaines, mais aussi à offrir un suivi plus rigoureux et une meilleure visibilité sur les informations sensibles liées à chaque famille.

Plus concrètement, cette application permettra :

- aux services compétents d'enregistrer et de consulter, de manière structurée, les informations relatives aux familles des victimes ;
- aux responsables de la gendarmerie d'assurer un suivi précis des prises en charge et des besoins exprimés ;
- à l'administration de disposer d'un tableau de bord centralisé pour coordonner les actions et prendre des décisions plus éclairées.

Par cette initiative, la Gendarmerie nationale renforcera sa capacité de gestion et son efficacité opérationnelle en s'appuyant sur un outil moderne, adapté au contexte et conforme aux exigences organisationnelles actuelles.

b. Problématique

Dans un contexte sécuritaire marqué par la persistance des attaques terroristes, la Gendarmerie nationale doit relever un défi majeur : assurer une prise en charge efficace des familles des victimes tout en garantissant une gestion rigoureuse et sécurisée des informations sensibles. Or, les méthodes actuelles, basées sur des procédures essentiellement manuelles et fragmentées, génèrent des lenteurs, des doublons et des risques de pertes de données. Ces limites compromettent non seulement l'efficacité administrative, mais affectent également la qualité du soutien apporté aux familles déjà fragilisées.

Dès lors, la problématique centrale qui se pose est la suivante :

Comment concevoir un système informatique capable d'assurer une gestion fiable, sécurisée et rapide des dossiers des familles des victimes d'attaques terroristes, afin d'améliorer à la fois l'efficacité organisationnelle et la qualité de l'accompagnement social ?

c. Objectifs

Les objectifs envisagés pour la conception et la réalisation d'une application de gestion des familles des victimes d'attaques terroristes au sein de la Gendarmerie nationale se déclinent comme suit :

- analyser le système actuel de gestion et de suivi des familles concernées, afin d'identifier ses forces, ses limites et ses insuffisances ;
- concevoir et mettre en place une solution numérique automatisée, adaptée aux besoins spécifiques de la Gendarmerie nationale pour un suivi des dossiers des familles des victimes d'attaques terroristes plus rigoureux, rapide et fiable ;
- rédiger un rapport de stage académique, constituant un support scientifique et technique requis pour l'obtention de notre licence professionnelle.

Dans la suite de ce travail, nous mettrons également en lumière les résultats attendus, en lien direct avec les objectifs fixés.

d. Résultats attendus

Au regard des besoins exprimés par la Gendarmerie nationale et des objectifs fixés, les résultats attendus à l'issue de ce projet sont :

- une application opérationnelle dédiée à la gestion et au suivi des familles des victimes d'attaques terroristes, conçue pour répondre aux exigences réelles du terrain ;
- la mise en pratique effective des connaissances théoriques et techniques acquises tout au long de la formation, dans un cadre professionnel concret ;
- la production d'un rapport de stage rédigé et présenté devant un jury académique, en vue de l'obtention de la licence professionnelle.

2. Méthode d'analyse et conception

Une méthode d'analyse et de conception constitue un procédé structuré visant à formaliser les étapes préliminaires du développement d'un système d'information. Son objectif

est de garantir que le futur système soit conforme aux besoins réels exprimés par l'organisation. Dans le cadre de notre projet, cette démarche se déploie en deux étapes :

➤ Phase d'analyse

Cette étape consiste à partir de l'énoncé informel du besoin, tel qu'exprimé par la Gendarmerie nationale, et à le compléter par des informations recueillies auprès des acteurs concernés (direction des ressources humaines, cellules sociales, agents administratifs, etc.). Elle inclut également une étude de l'existant, c'est-à-dire l'examen des pratiques actuelles de gestion des familles des victimes d'attaques terroristes, lesquelles reposent souvent sur des documents papier ou des traitements manuels. L'analyse vise à identifier clairement :

- les résultats attendus en termes de fonctionnalités (enregistrement des familles, suivi des aides, génération de rapports, etc.) ;
- les critères de performance (rapidité, fiabilité, disponibilité) ;
- la robustesse et la maintenabilité du futur système
- les exigences de sécurité et de confidentialité des données ;
- les besoins d'évolutivité et d'interopérabilité avec d'autres outils administratifs.

➤ Phase de conception

Une fois les besoins formalisés, la conception consiste à décrire de manière claire et non ambiguë le fonctionnement futur de l'application. Pour cela, des langages et outils de modélisation (par exemple UML : diagrammes de cas d'utilisation, de séquence, de classes, etc.) sont utilisés afin de représenter :

- les interactions entre les acteurs et le système ;
- les traitements internes à effectuer ;
- la structure de la base de données ;
- les échanges d'informations nécessaires.

Cette étape fournit donc une vision structurée du futur logiciel, facilitant son développement et son implémentation. Ainsi, l'approche adoptée garantit que la solution proposée réponde de

manière efficace aux attentes de la Gendarmerie nationale, tout en tenant compte des contraintes organisationnelles et techniques.

a. Cycle de développement

Les méthodologies de développement regroupent l'ensemble des activités structurées qui encadrent la réalisation d'un projet de système d'information (SI). Elles définissent les étapes, les rôles, les outils et les pratiques à adopter afin de garantir la qualité, la fiabilité et la conformité du produit final avec les besoins exprimés.

Au fil du temps, plusieurs approches se sont imposées, chacune proposant une organisation et un degré de flexibilité différents. Parmi elles, cinq méthodologies se distinguent par leur popularité et leur large adoption dans les projets informatiques :

- **Cascade (Waterfall)** : approche séquentielle où chaque étape doit être achevée avant de passer à la suivante.
- **Scrum** : méthode agile basée sur des itérations courtes appelées *sprints*, favorisant l'adaptation rapide aux changements.
- **2TUP (Two-Track Unified Process)** : variante unifiée qui sépare le développement en deux cycles parallèles (cycle système et cycle logiciel).
- **RUP (Rational Unified Process)** : méthodologie itérative et incrémentale, basée sur UML, orientée vers la gestion des risques.
- **XP (Extreme Programming)** : méthode agile mettant l'accent sur la qualité du code, la collaboration étroite avec le client et la simplicité des solutions.

Afin de mieux comparer ces approches, le **Tableau 1** présente une synthèse comparative de leurs caractéristiques principales.

Tableau 1 : Comparaison des méthodologies de développement

Méthodologies	Description	Atouts	Insuffisances
RUP (Rational Unified Process)	<p>Le RUP est à la fois une méthodologie et un outil prêt à l'emploi.</p> <p>Cible des projets de plus de 10 personnes.</p>	<ul style="list-style-type: none"> -Itératif. -Spécifie le dialogue entre les différents intervenants du projet. 	<ul style="list-style-type: none"> - Assez flou dans sa mise en œuvre. - Ne couvre pas les phases en amont et en aval au développement.
Cascade	Les phases sont déroulées d'une manière séquentielle.	Distingue clairement les phases du projet.	<ul style="list-style-type: none"> - non itératif. - Pas de modèles pour les documents
2TUP (Two Truck Unified Process)	<ul style="list-style-type: none"> - Il s'articule autour de l'architecture. - Propose un cycle de développement en Y. - Cible des projets de toutes tailles. 	<ul style="list-style-type: none"> - Itératif et laisse une large place à la technologie et à la gestion des risques. - Définit les profils des intervenants, les plannings, les prototypes 	<ul style="list-style-type: none"> - Plutôt superficiel sur les phases situées en amont et en aval du développement. - Ne propose pas de documents types.
XP(eXtreme Programming)	<ul style="list-style-type: none"> - Ensemble des bonnes pratiques de développement. - Cible : Moins de 10 personnes 	<ul style="list-style-type: none"> - Itératif et donne une importance aux aspects techniques. -Innovant: programmation en duo. 	<ul style="list-style-type: none"> - Assez flou dans sa mise en œuvre. - Ne couvre pas les phases en amont et en aval au développement.
Scrum	- Se base sur des itérations dites sprints de développement	<ul style="list-style-type: none"> - Donne toute confiance aux développeurs et les laisser effectuer leur travail. - Chaque itération a un objectif bien précis et fournit une fonctionnalité testée. 	<ul style="list-style-type: none"> - La mise en œuvre du développement n'est pas précisée. - Développement rapide et répétitif

L'étude comparative menée sur les processus de développement **RUP**, **Cascade**, **2TUP**, **XP** et **Scrum**, présentés dans le tableau 1, nous conduit à plusieurs constats.

- **RUP** néglige les contraintes techniques, pourtant essentielles à notre projet : nous l'avons donc écarté.
- **Cascade** reste strictement séquentiel, sans ouverture vers l'itératif.
- **2TUP** se distingue par sa capacité à séparer les contraintes fonctionnelles et techniques, sous forme de deux branches exploitables en parallèle.
- **XP** privilégie fortement le développement, mais délaisse la capture des besoins et la conception : il ne convient donc pas
- **Scrum**, enfin, fragmente le projet en sprints courts — en théorie limités à 30 jours.

À l'issue de cette comparaison, et pour mieux contrôler les risques tout en assurant la réussite du projet, nous avons choisi **2TUP**. Deux raisons principales guident ce choix. D'abord, 2TUP valorise la **technologie**, dimension cruciale pour notre application. Ensuite, il adopte une organisation en « **Y** », composée d'une branche fonctionnelle, d'une branche technique et d'une branche réalisation. Les deux premières peuvent évoluer en parallèle.

Ainsi, une évolution technologique ou une modification d'un besoin technique peut être intégrée facilement via la branche correspondante, sans perturber le reste du projet. De même, une nouvelle fonctionnalité peut être ajoutée en ne touchant qu'à la branche fonctionnelle.

En résumé, le **2TUP** est un processus issu du processus unifié. Il propose un cycle en « **Y** » qui dissocie clairement technique et fonctionnel. Son point de départ est une étude préliminaire : identification des acteurs du système, recensement des interactions avec celui-ci, rédaction du cahier des charges et modélisation du contexte.

Le processus s'articule autour de trois (3) phases essentielles :

❖ **La branche fonctionnelle** comprend deux volets essentiels :

- **La capture des besoins fonctionnels** : elle vise à modéliser les attentes des utilisateurs en se concentrant sur leur métier. Cette étape permet d'anticiper le risque de concevoir

un système inadapté. En parallèle, la maîtrise d'œuvre consolide les spécifications et s'assure de leur cohérence ainsi que de leur exhaustivité.

- **L'analyse** : elle consiste à examiner en détail la spécification fonctionnelle afin de déterminer clairement ce que le système devra accomplir sur le plan métier. Les résultats obtenus restent indépendants de toute technologie particulière.

❖ **La branche technique** s'articule autour de deux étapes majeures :

- **La capture des besoins techniques** : elle consiste à identifier l'ensemble des contraintes ainsi que les choix structurants nécessaires à la conception du système. Les outils, les matériels sélectionnés et les exigences liées à l'intégration avec les solutions déjà existantes définissent en grande partie les prérequis de l'architecture technique.
- **La conception générique** : elle précise ensuite les composants indispensables à la mise en place de l'architecture technique. Entièrement indépendante des aspects fonctionnels, cette étape vise à uniformiser et à favoriser la réutilisation des mécanismes communs à l'ensemble du système. L'architecture technique constitue ainsi l'ossature du système informatique et permet de réduire considérablement les risques techniques. Compte tenu de son importance, la réalisation d'un prototype est souvent recommandée pour valider sa pertinence.

❖ **La phase de réalisation** comprend plusieurs étapes successives :

- **La conception préliminaire** : elle constitue une étape sensible puisqu'elle intègre le modèle issu de l'analyse dans l'architecture technique, afin d'établir la cartographie des différents composants du système à développer.
- **La conception détaillée** : elle précise ensuite la manière dont chaque composant sera construit et décrit de façon plus fine leurs interactions.
- **Le codage** : cette étape produit concrètement les composants définis, avec des tests unitaires effectués au fur et à mesure, garantissant la qualité et la fiabilité du code développé.
- **L'étape de recette**, qui consiste enfin à valider les fonctions du système développé.

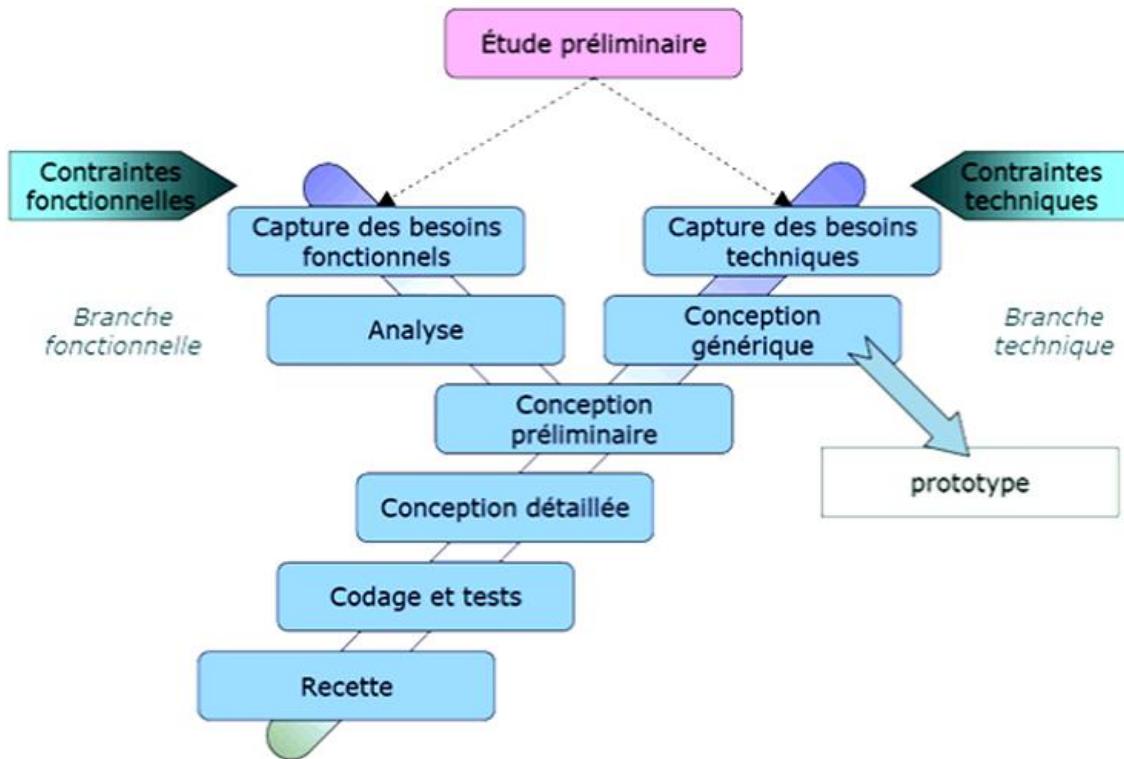


Figure 3 : schéma présentant le processus 2TUP

Source : https://www.memoireonline.com/05/13/7195/m_Mise-en-place-d'une-application-webmapping-de-geolocalisation-des-points-d'intert-de-la-vill6.html

Le processus 2TUP s'appuie sur UML tout au long du cycle de développement, car les différents diagrammes de ce dernier permettent en plus de leur facilité et de leur clarté, de bien modéliser le système à chaque étape.

b. Langage de modélisation

Un langage de modélisation est un langage informatique élaboré pour représenter de manière abstraite des systèmes, des processus, des données ou encore des concepts. Ces modèles servent à comprendre, analyser, concevoir, communiquer et documenter des aspects souvent complexes liés à l'information ou à la réalité. Parmi les principaux langages de modélisation utilisés, on peut citer : UML¹, SysML², BPMN³ et MBT⁴.

¹ UML: <https://www.lucidchart.com/pages/fr/langage-uml>

² SysML : <https://si.blaisepascal.fr/1t-sysml/>

³ BPMN : <https://www.elemente.co/fr/bpmn/>

⁴ MBT : <https://matelo-testing-software.com/fr/blog/model-based-testing-fr/le-model-based-testing/>

Le Tableau 2 présente une étude comparative entre les langages de modélisation cités précédemment.

Tableau 2 : Comparaison entre quelques langages de modélisation

Langages de modélisation	Caractéristiques
UML (Unified Modeling Langage)	<ul style="list-style-type: none"> • Langage de représentation d'un système d'information ; • Système de notation orienté objet ; • Plus orientée vers la conception ; • International.
SysML (Système Modeling Langage)	<ul style="list-style-type: none"> • Langage de modélisation spécifique au domaine de l'ingénierie système ; • Sémantique de SysML est plus riche et flexible • Langage plus réduit que UML.
BPMN(Business Process Model and Notation)	<ul style="list-style-type: none"> • Modèle de processus pour représenter le déroulement des processus internes ou publique d'une organisation ; • Vise l'analyse et la conception des processus métiers qui font intervenir et interagir des systèmes ; • Son but principal est de fournir une notation qui soit facilement compréhensible par tous les utilisateurs de l'entreprise.
MBT (Model Based Testing)	<ul style="list-style-type: none"> • Déclinaison des solutions d'ingénierie des modèles appliquées au test ; • Permet de produire des cas de tests à partir d'un modèle ; • Diminue les efforts de maintenance des jeux de tests ; • Renforce la qualité de la documentation des exigences.

À la suite de l'analyse comparative présentée dans le Tableau 2, nous avons opté pour le langage UML pour plusieurs raisons essentielles :

- ❖ sa reconnaissance universelle en tant que standard de modélisation objet, largement accepté et utilisé dans l'industrie ;
- ❖ sa capacité à exprimer et à développer des modèles de manière indépendante de tout langage de programmation spécifique ;
- ❖ sa flexibilité pour représenter un système sous différentes perspectives complémentaires grâce à l'utilisation de différents types de diagrammes.

Ainsi, dans le cadre de notre approche basée sur le processus 2TUP et l'utilisation d'UML en tant que langage de modélisation, nous devons former une équipe composée de divers acteurs qui jouent un rôle essentiel dans la réalisation réussie du projet.

3. Groupe de travail

Le groupe de travail représente l'ensemble des personnes nécessaires à la réalisation du projet. Ainsi, nous avons trois groupes de travail, à savoir le groupe projet, le groupe de pilotage et le groupe utilisateur.

a. Groupe de pilotage

C'est le groupe dirigeant chargé de veiller au bon déroulement du projet. Il planifie les dates clés du projet, examine les propositions du groupe de projet, et décide des orientations stratégiques. Ce groupe est constitué de :

- ❖ Adjudant SAWADOGO Moussa Ingénieur en MSI notre maître de stage
- ❖ Dr KONE Lydie Maître Assistant en informatique à l'IBAM, notre directeur de rapport ;

b. Groupe de projet

Le groupe de projet est l'ensemble des personnes chargées de réaliser le projet. Il est l'intermédiaire entre le groupe de pilotage et le groupe des utilisateurs. Ce groupe est composé principalement de nous, Gérald Raogo KIEMDE, étudiant en troisième (3ème) année en Méthodes Informatiques Appliquées à la Gestion (MIAGE) à l'IBAM.

c. Groupe des utilisateurs

Ce groupe regroupe l'ensemble des personnes appelées à utiliser le futur système. Il intervient activement dans la capture des besoins fonctionnels, en exprimant ses attentes et en précisant les contraintes liées à son activité. Dans le cadre de notre projet, ce groupe est constitué principalement des agents de la gendarmerie nationale chargés de la gestion et du suivi des familles des victimes.

Dans l'optique d'assurer une bonne conduite de notre projet, nous avons élaboré une planification structurée. La section suivante présentera le planning prévisionnel établi pour la réalisation du projet, en mettant en évidence les différentes étapes et leur chronologie.

d. Planning prévisionnel

Afin de garantir le bon déroulement de notre étude tout en respectant les délais impartis, le projet a été découpé en plusieurs tâches distinctes. Pour organiser et suivre efficacement ces étapes, nous avons recours au diagramme de Gantt, un outil reconnu de planification et de gestion de projet. Celui-ci offre une représentation graphique permettant de visualiser dans le temps l'enchaînement des différentes activités. Le planning prévisionnel ainsi établi est présenté de manière synthétique dans la figure 4 ci-dessous :

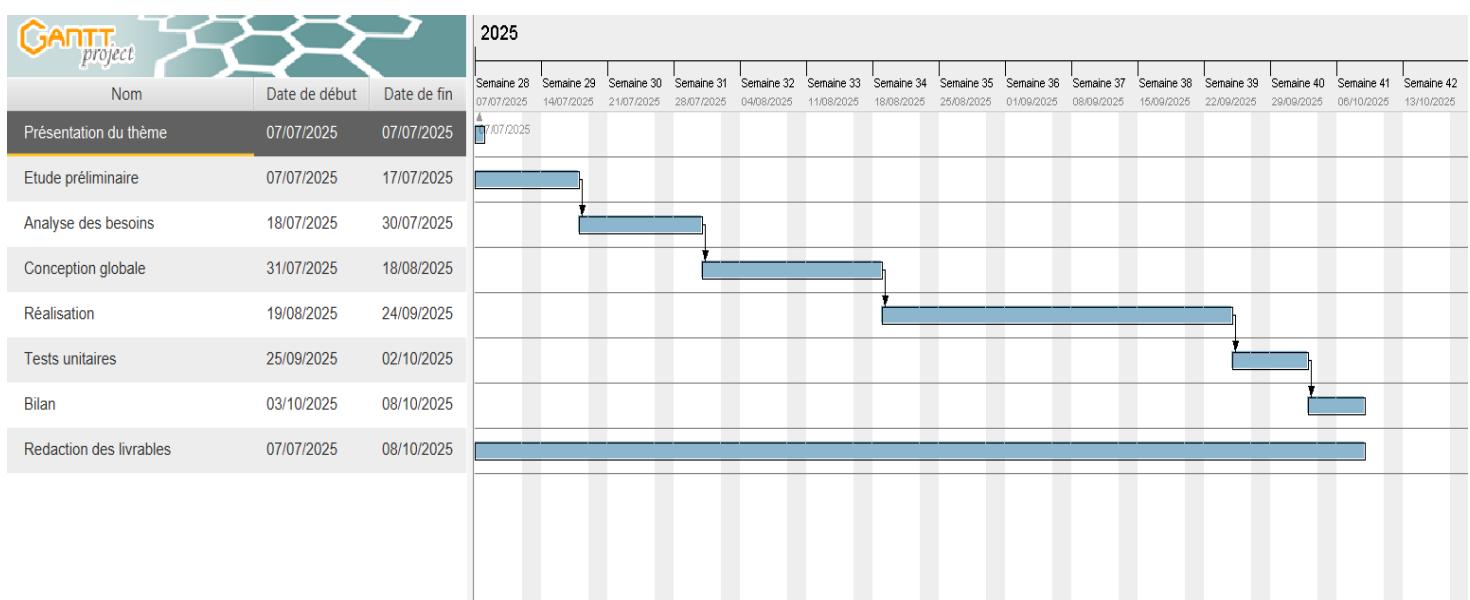


Figure 4 : Planning de réalisation du projet

Après avoir exposé le planning prévisionnel de réalisation, nous abordons à présent l'expression des besoins. Cette étape vise à identifier les acteurs impliqués ainsi que les interactions qu'ils entretiennent avec l'application de gestion des familles des victimes d'attaques terroristes. Elle constitue une phase clé, car elle permet de clarifier les rôles de chaque intervenant et de préciser la nature des échanges entre les utilisateurs et le système à concevoir.

II. Expressions des besoins

Cette partie a pour objectif, d'une part, de présenter le fonctionnement des modèles existants et les améliorations apportées au système de gestion des familles des victimes d'attaques terroristes au sein de la Gendarmerie Nationale. D'autre part, elle vise à mettre en lumière les différents acteurs qui interagissent avec l'application, ainsi que la nature des informations et messages échangés entre eux et le système.

1. Étude de l'existant

L'étude de l'existant constitue une démarche méthodique qui cherche à analyser en profondeur le fonctionnement actuel d'un système ou d'un processus. Dans notre cas, elle consiste à examiner les mécanismes déjà en place pour la gestion des familles des victimes d'attaques terroristes au sein de la Gendarmerie Nationale, afin d'en comprendre les forces, mais aussi les limites.

a. Présentation du système utilisé à la Gendarmerie Nationale

Le but de la présentation du système actuellement utilisé à la Gendarmerie Nationale est de mettre en évidence ses points forts et ses limites, afin de mieux cerner les besoins réels et de les prendre en compte dans la conception de notre solution. Dans cette partie, nous décrivons le fonctionnement existant concernant la gestion des familles des victimes d'attaques terroristes.

Aujourd'hui, la Gendarmerie Nationale ne dispose pas d'une plateforme spécifique dédiée à ce suivi. La collecte et l'organisation des informations se font principalement à travers des documents papier et, parfois, des fichiers Excel. Ces données sont ensuite transmises manuellement entre les services concernés, ce qui rend la communication lente et parfois

incomplète. Le suivi de chaque famille repose alors sur des démarches répétitives et souvent fastidieuses, qui mobilisent beaucoup de temps et de ressources.

Une fois les informations recensées, les responsables en charge établissent des rapports périodiques sur la situation des familles. Ces rapports servent de base pour les décisions administratives, mais leur élaboration prend du temps et complique la réactivité dans la prise en charge. À l'issue de cette observation du système existant à la Gendarmerie Nationale, nous relevons des forces mais aussi des faiblesses :

❖ Forces du système

- conservation des données sous forme écrite, ce qui garantit une certaine traçabilité ;
- contact direct et humain entre les familles et les responsables, favorisant une relation de proximité.

❖ Faiblesses du système

- lenteur dans le traitement des informations et risque de perte de temps ;
- difficulté à centraliser et partager efficacement les données ;
- manque de visibilité sur le suivi global des familles ;
- risques accrus d'erreurs liés aux saisies manuelles.

Ces constats mettent en évidence la nécessité d'une solution numérique personnalisée, capable de centraliser, sécuriser et automatiser la gestion des familles des victimes.

b. Exigences fonctionnelles

Après avoir mis en lumière les limites du système actuel, nous présentons maintenant les exigences fonctionnelles que doit satisfaire l'application de gestion des familles des victimes d'attaques terroristes. Ces exigences traduisent les besoins essentiels exprimés par les utilisateurs et orientent la conception de la solution. Ces exigences sont entre autres :

- ✓ permettre l'enregistrement numérique des familles et des victimes avec toutes les informations administratives nécessaires (identité, liens familiaux, coordonnées, etc.) ;

- ✓ offrir la possibilité de suivre le parcours et la prise en charge de chaque famille (aide financière, soutien psychologique, démarches administratives, etc.) ;
- ✓ centraliser toutes les données dans une base unique afin de limiter les redondances et d'assurer un accès rapide et fiable à l'information ;
- ✓ mettre en place un système de recherche et de filtrage pour retrouver facilement les dossiers ;
- ✓ générer automatiquement des rapports statistiques pour appuyer les prises de décision et améliorer la planification ;
- ✓ assurer la traçabilité des actions grâce à un journal d'activité recensant les opérations effectuées par les différents utilisateurs ;
- ✓ intégrer un système de gestion des rôles (administrateur, responsable, agent) pour sécuriser l'accès aux informations sensibles.

Ces fonctionnalités visent à rendre le processus plus fluide, plus transparent et surtout mieux adapté aux réalités de la Gendarmerie Nationale dans son rôle d'accompagnement des familles touchées par les attaques terroristes.

2. Présentation du processus de fonctionnement amélioré

Le système amélioré permet aux différents acteurs de la Gendarmerie nationale de collaborer efficacement sur une même plateforme de gestion des familles des victimes d'attaques terroristes. Cette application , pensée pour centraliser les données et fluidifier la coordination, facilite à la fois l'enregistrement, le suivi et la prise en charge des familles. Elle comporte deux interfaces principales : l'interface administrateur et l'interface utilisateurs (agents de saisie, assistants sociaux, responsables).

Depuis l'interface administrateur, le responsable peut créer les comptes des agents, définir leurs permissions et organiser le traitement des dossiers. Il enregistre les familles, ajoute les informations des victimes et assigne les suivis nécessaires aux assistants sociaux.

Les agents , une fois connectés, peuvent remplir les fiches familles, téléverser des documents officiels (actes de décès, pièces d'identité, etc.), mettre à jour les informations . Les assistants sociaux, de leur côté, accèdent aux dossiers qui leur sont confiés. Ils proposent des aides adaptées (soutien financier, logement, aide scolaire), ajoutent des remarques et soumettent de nouvelles demandes si la situation évolue.

Les responsables, assurent la validation des aides. Ils suivent l'ensemble des familles, génèrent des rapports statistiques et veillent au bon déroulement du processus.

Ce modèle amélioré offre une vision claire et partagée du suivi, réduit les lenteurs administratives et renforce la transparence des décisions

3. Spécification fonctionnelle

La spécification fonctionnelle est la description des fonctions d'un logiciel en vue de sa réalisation. Dans cette partie, nous décrirons dans les détails les exigences du futur système à travers l'identification des acteurs, des cas d'utilisation et les différents diagrammes.

a. Identification des acteurs

Un acteur correspond à un rôle précis qu'un utilisateur ou une entité peut assumer lors de son interaction avec le système. Dans le cadre de notre projet, les principaux acteurs identifiés sont :

- l'administrateur : il gère les comptes utilisateurs, supervise l'ensemble du système et valide les décisions importantes ;
- l'agent de gendarmerie : il enregistre les victimes et leurs familles, saisit les informations et assure le suivi administratif initial ;
- l'assistant social : il analyse les dossiers, propose des aides adaptées (financières, scolaires, logement) et relance si nécessaire ;
- le responsable : il valide ou rejette les propositions d'aide et génère des rapports de suivi globaux.

b. Identification des cas d'utilisation

Les cas d'utilisation représentent l'ensemble des besoins validés par les utilisateurs. Ils permettent de structurer les besoins des utilisateurs ainsi que les objectifs correspondants d'un système. Le tableau 3 suivant présente l'ensemble des cas d'utilisation de notre système de gestion du stock matériel.

Tableau 3 : Liste des cas d'utilisation

Numéro	Cas d'utilisation	Description	Acteurs
CU 01	S'authentifier	L'utilisateur doit au préalable se connecter avant d'accéder au tableau de bord.	Tous les acteurs
CU 02	Ajouter Famille	Permet d'enregistrer une nouvelle famille de victime dans le système.	Agent Gendarmerie
CU 03	Créer Fiche Victime	Permet de créer une fiche détaillée sur une victime d'attaque terroriste.	Agent Gendarmerie
CU 04	Consulter Fiche Victime	Permet de visualiser les informations liées à une fiche victime.	Assistant Social Responsable
CU 05	Formuler Demande	Permet de saisir une demande d'assistance ou de soutien pour une famille.	Assistant Social
CU 06	Soumettre Demande	Permet de soumettre officiellement une demande enregistrée dans le système.	Assistant Social
CU 07	Valider Demande	Permet de vérifier et d'approuver une demande soumise.	Responsable
CU 08	Générer Rapport	Permet de produire un rapport récapitulatif des activités ou demandes traitées.	Responsable
CU 09	Consulter Historique	Permet de consulter l'historique des demandes et interventions effectuées.	Responsable Administrateur
CU10	Gérer Utilisateurs	Permet d'ajouter, modifier ou supprimer des utilisateurs du système.	Administrateur

Pour une meilleure visualisation globale de l'ensemble des cas d'utilisation du système, nous présentons le diagramme des cas d'utilisation du système dans la suite de notre travail.

c. Diagramme des cas d'utilisation

Le diagramme de cas d'utilisation, en UML, offre une vue d'ensemble du comportement fonctionnel d'un système logiciel. D'une part, il sert à représenter, clarifier et organiser les besoins des utilisateurs. D'autre part, il met en évidence les acteurs ainsi que les différentes fonctionnalités du système, comme l'illustre la Figure 5.

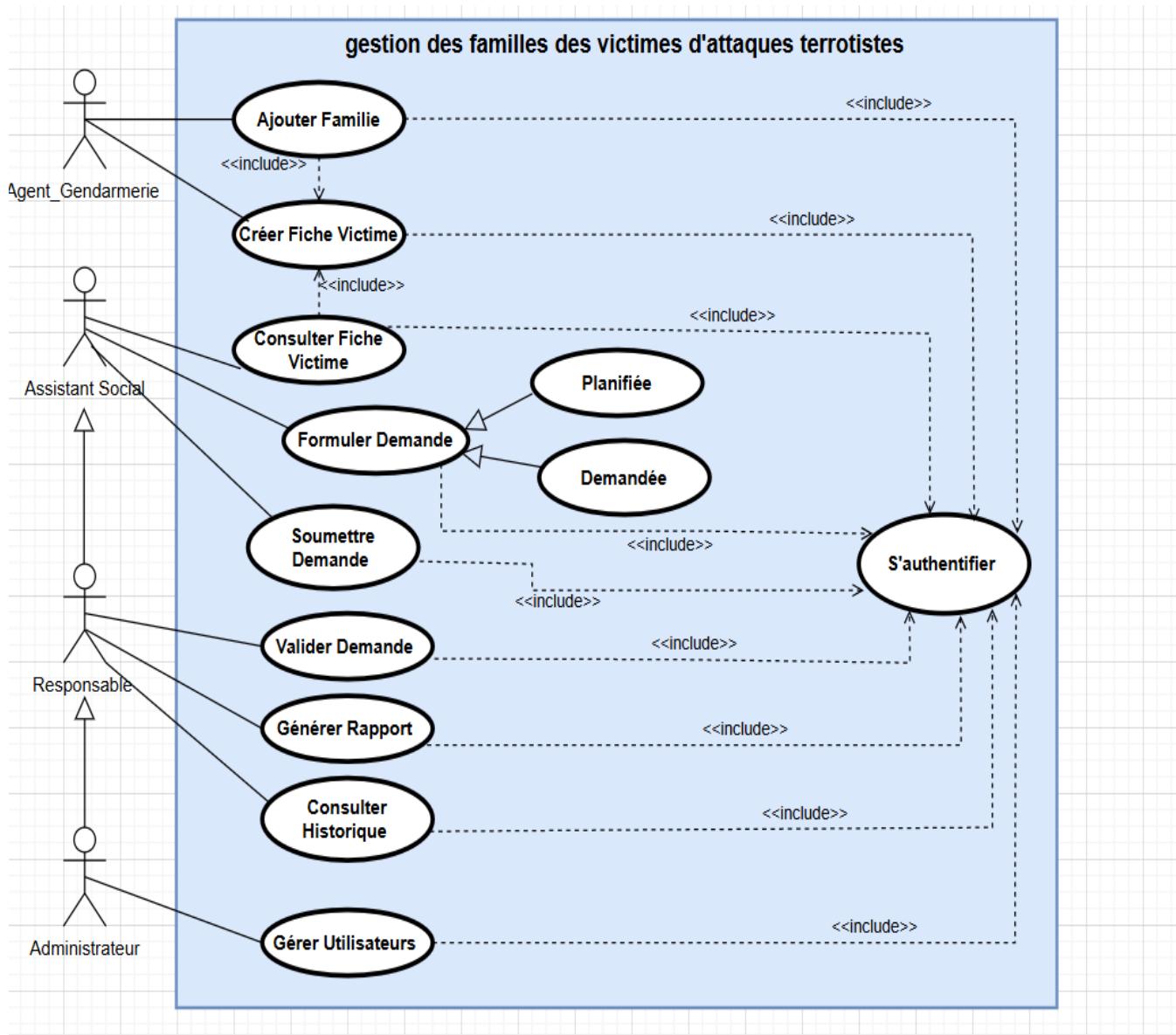


Figure 5 : Diagramme des cas d'utilisation

d. Description textuelle de quelques cas d'utilisation

La description textuelle sert à détailler le dialogue entre les acteurs et le système. Elle montre comment le système interagit avec ses utilisateurs, ou avec d'autres systèmes, pour accomplir une fonction donnée.

Elle se présente sous forme de tableau. Chaque partie a un rôle précis :

- La section Acteur liste ceux qui peuvent utiliser le cas d'utilisation.
- La section Précondition précise l'état du système avant le déclenchement.
- La section Scénario nominal décrit le déroulement normal, sans erreur.
- La section Scénario alternatif présente une variante du scénario nominal.
- La section Postcondition indique l'état final du système, après exécution.

Pour plus de clarté, nous avons choisi de décrire uniquement trois cas d'utilisation : « S'authentifier », « Créer Fiche Victime » et « Ajouter Famille ». Ils sont détaillés respectivement dans le tableau 4 , le tableau 5, le tableau 6

❖ CU1 : S'authentifier

Le Tableau 4 illustre le cas d'utilisation « S'authentifier » et décrit les échanges entre l'acteur et le système. Dans un premier temps, l'utilisateur initie une demande de connexion. Le système lui transmet alors un formulaire de connexion. L'utilisateur saisit son nom d'utilisateur et son mot de passe, puis soumet le formulaire.

À cette étape, le système procède à la vérification des informations. Si elles sont incorrectes, il renvoie le formulaire afin que l'utilisateur puisse réessayer. Si elles sont correctes, l'utilisateur est automatiquement redirigé vers sa page d'accueil personnalisée.

Cette procédure s'applique à l'ensemble des acteurs du système.

Tableau 4 : Description textuelle du cas d'utilisation “S'authentifier”

CU01 : S'authentifier
Nom : S'authentifier
Résumé : Ce cas d'utilisation permet à l'acteur de s'authentifier sur le système
Acteur : tous les utilisateurs
Préconditions : Plateforme disponible ; L'utilisateur possède un compte enregistré.

Scénario nominal:

1. L'utilisateur ouvre la plateforme
2. Le système lui présente la page de connexion
3. L'utilisateur renseigne ses informations de connexion
4. Le système vérifie les informations de connexion [A1]
5. Le système affiche le tableau de bord correspondant

Scénario alternatif :**A1 : Si les informations de connexions sont erronées**

Le système affiche un message d'erreur et invite l'acteur à corriger les informations ;

Le processus reprend au point 3 du scénario nominal.

6. Le système invite l'utilisateur à entrer ses informations de connexion à nouveau

Post conditions : L'utilisateur est connecté à la plateforme**❖ CU3 : Créer Fiche Victime**

Le Tableau 5 illustre le cas d'utilisation « Créer une fiche victime » et décrit les interactions entre l'acteur et le système.

Dans un premier temps, l'acteur (administrateur ou agent autorisé) sélectionne l'option « Créer une fiche victime ».

Le système lui présente alors un formulaire vide destiné à la saisie des informations personnelles et administratives de la victime.

L'acteur renseigne les différents champs requis, puis soumet le formulaire.

À cette étape, le système procède à la vérification des données saisies. Si certaines informations sont invalides ou incomplètes, il renvoie le formulaire accompagné d'un message d'erreur afin que l'acteur puisse corriger et soumettre à nouveau.

Si les informations sont correctes et complètes, le système enregistre la fiche victime dans la base de données, génère un identifiant unique et confirme la création en affichant la nouvelle fiche.

Cette procédure garantit une centralisation et une traçabilité efficace des informations relatives aux victimes.

Tableau 5 : Description textuelle du cas d'utilisation "S'authentifier"

CU03 : Créer Fiche Victime
Nom : Créer une fiche victime
Résumé : Ce cas d'utilisation permet à l'administrateur ou à un agent autorisé d'enregistrer une nouvelle victime d'attentat dans le système à travers une fiche comportant ses informations personnelles et administratives.
Acteur : Administrateur, agent
Préconditions : L'acteur est authentifié sur la plateforme ; Le module de gestion des victimes est accessible.
Scénario nominal: <ol style="list-style-type: none"> 1. L'acteur sélectionne l'option « Créer une fiche victime » depuis le tableau de bord. 2. Le système affiche un formulaire vide destiné à la saisie des informations de la victime. 3. L'acteur renseigne les informations personnelles et administratives (nom, prénoms, date de naissance, adresse, grade, etc.). 4. Le système vérifie la validité des données saisies [A1]. 5. L'acteur valide la saisie. 6. Le système enregistre la fiche victime dans la base de données. 7. Le système confirme la création et affiche la fiche avec un identifiant unique.
Scénario alternatif : A1 : Si des champs obligatoires sont manquants ou invalides : Le système affiche un message d'erreur et invite l'acteur à corriger les informations ; Le processus reprend au point 3 du scénario nominal. 8. Le système invite l'utilisateur à entrer les informations de la victime à nouveau
Post conditions : La fiche victime est correctement créée et stockée dans le système ; L'acteur peut consulter, modifier ou supprimer la fiche ultérieurement selon ses droits.

❖ CU2 : Ajouter famille

Le Tableau 6 illustre le cas d'utilisation « Ajouter une famille » et décrit les échanges entre l'acteur et le système.

Après avoir créé une fiche victime, l'utilisateur (administrateur ou agent habilité) sélectionne l'option « Ajouter famille ».

Le système lui propose alors un formulaire destiné à enregistrer les informations des membres de la famille (nom, prénom, lien de parenté, coordonnées, etc.).

L'utilisateur complète les champs requis et valide le formulaire.

À ce stade, le système vérifie la cohérence des données.

Si des informations sont manquantes ou invalides, le formulaire est renvoyé pour correction.

Si toutes les données sont conformes, le système enregistre la famille dans la base, l'associe automatiquement à la fiche de la victime concernée, et confirme l'opération à l'utilisateur.

Ce processus permet d'assurer que chaque victime a une famille correctement reliée à sa fiche, facilitant ainsi le suivi et la gestion des ayants droit.

Tableau 6 : Description textuelle du cas d'utilisation “Ajouter Famille”

CU2: Ajouter Famille
Nom : Ajouter une famille
Résumé : Ce cas d'utilisation permet à l'acteur d'ajouter une famille à une fiche victime déjà créée.
Acteur : Administrateur, Agent
Préconditions : La fiche victime existe déjà dans le système. L'acteur est connecté à la plateforme.
Scénario nominal: <ol style="list-style-type: none"> 1. L'utilisateur sélectionne la fiche victime concernée. 2. Le système affiche les détails de la victime avec l'option « Ajouter famille ». 3. L'utilisateur clique sur le bouton « Ajouter famille ».

4. Le système présente un formulaire d'ajout
5. L'utilisateur saisit les informations de la famille.
6. Le système vérifie la cohérence et la validité des données [A1].
7. Le système enregistre la famille et l'associe à la fiche victime.
8. Le système confirme l'opération et affiche la famille ajoutée.

Scénario alternatif :

A1 : Si les informations saisies sont incomplètes ou incorrectes

- i. Le système refuse l'enregistrement.
- ii. Le système renvoie le formulaire pour correction.
- iii. Le scénario reprend à l'étape 5 du scénario nominal.

L'enchaînement commence au point 3 du scénario nominal

7. Le système invite l'employé à entrer ses informations de connexion à nouveau

Post conditions : La famille est enregistrée dans le système et liée à la fiche de la victime correspondante.

La description textuelle met en évidence les interactions entre les différents acteurs du système de façon statique. Cependant, il est tout aussi essentiel de représenter ces échanges sous un angle dynamique. C'est précisément le rôle des diagrammes de séquence, qui illustrent la chronologie des actions et des messages échangés entre le système et les acteurs. Dans la suite, nous présentons les diagrammes de séquence représentés respectivement dans les Figures 4, 5 et 6, qui détaillent le déroulement des cas d'utilisation « **S'authentifier** », « **Créer une fiche victime** » et « **Ajouter une famille** ».

e. Diagramme de séquence

Les diagrammes de séquence mettent en lumière les échanges entre le système et les utilisateurs, en illustrant sous forme de scénarios la succession chronologique des messages liés à un cas d'utilisation donné.

Dans le cadre de notre projet, les diagrammes de séquence présentés concernent principalement les cas d'utilisation « **S'authentifier** », « **Créer une fiche victime** » et « **Ajouter une famille** ».

La lecture de ces diagrammes obéit à une logique précise :

- les flèches orientées vers la droite traduisent les messages ou actions initiés par les acteurs ;
- les flèches orientées vers la gauche correspondent aux réponses émises par le système ;
- enfin, les lignes verticales représentent les temps d'activité des acteurs, la lecture se faisant du haut vers le bas.

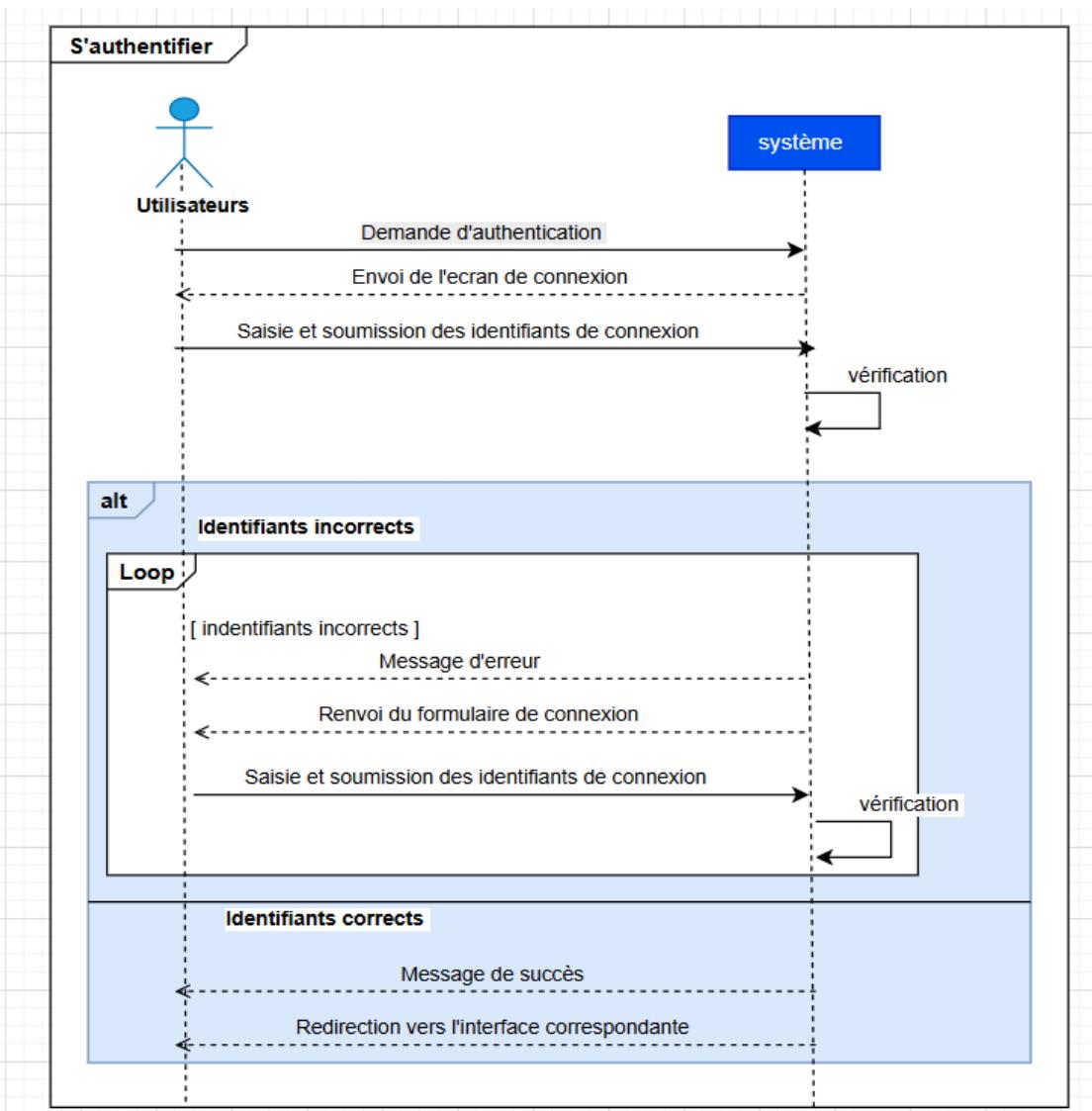


Figure 6 : Diagramme de séquence du cas d'utilisation « S'authentifier »

La Figure 6 illustre les échanges de messages relatifs au cas d'utilisation « S'authentifier ». Dans un premier temps, l'utilisateur initie une demande de connexion. Le système lui renvoie alors un formulaire destiné à la saisie de son username et de son mot de passe.

Une double vérification est ensuite effectuée par le système :

- Vérification syntaxique : le username doit respecter le format attendu et le mot de passe contenir au minimum quatre (04) caractères. Si cette règle n'est pas respectée, l'utilisateur reçoit une notification d'erreur indiquant « Syntaxe non valide ».
- Vérification de compatibilité : si les informations sont syntaxiquement correctes, le système vérifie leur correspondance avec les données enregistrées. En cas de non-concordance, un message d'erreur « Identifiants incorrects » est retourné.

Lorsque les deux contrôles sont validés, l'utilisateur est authentifié avec succès et accède à sa page d'accueil personnalisée.

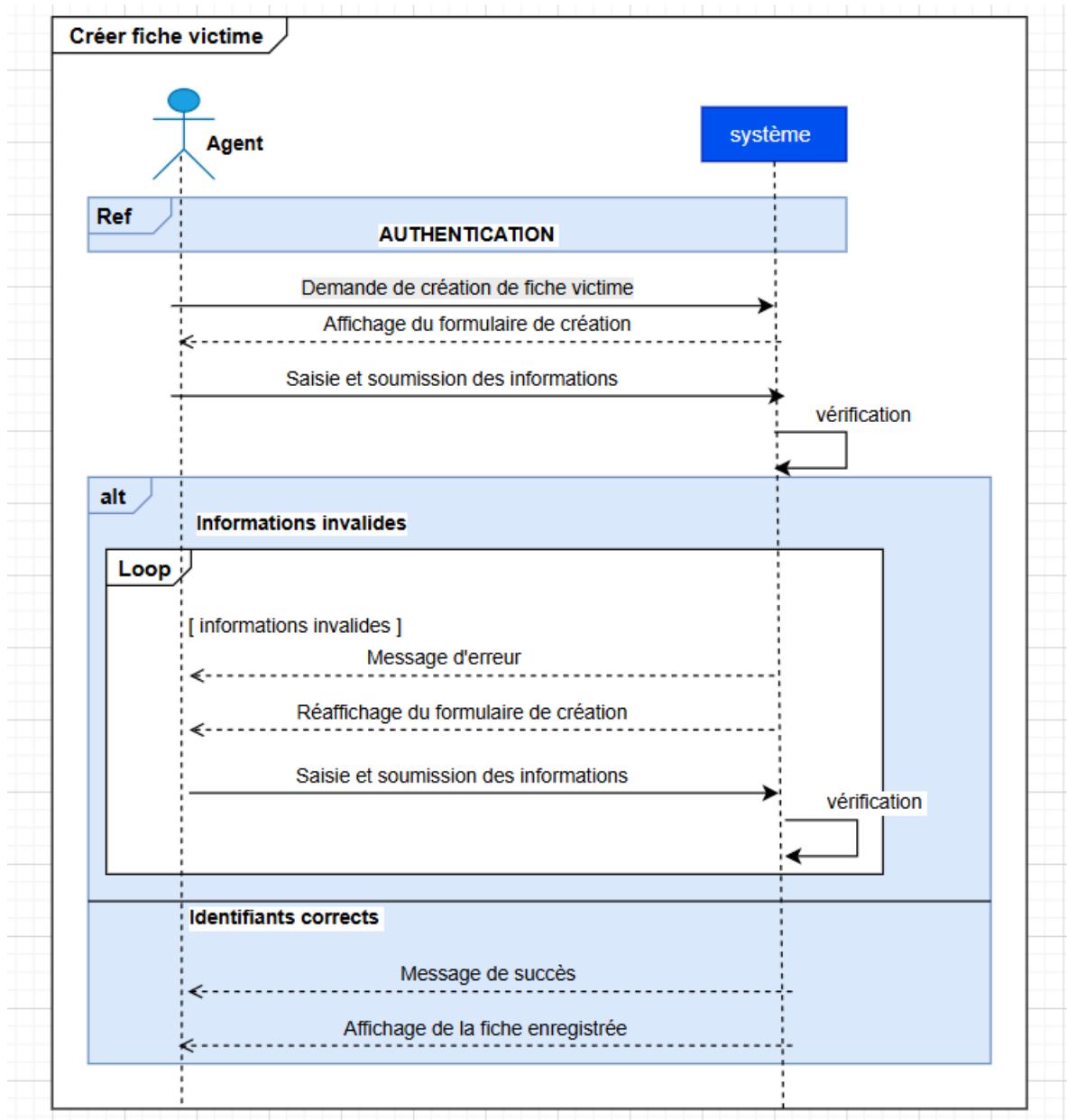


Figure 7 : Diagramme de séquence du cas d'utilisation « Crée Fiche Victime »

La Figure 7 illustre les interactions intervenant dans le cas d'utilisation « Crée Fiche Victime ». L'utilisateur déjà authentifié initie la demande de création d'une nouvelle fiche. Le système lui retourne alors un formulaire destiné à la saisie des informations personnelles de la victime (nom, prénom, date de naissance, grade , etc.).

Une vérification est réalisée par le système pour s'assurer que tous les champs obligatoires sont correctement renseignés. Si certains champs sont manquants ou mal saisis, l'utilisateur reçoit une notification d'erreur indiquant « Informations incomplètes ou non valides ».

Si aucune erreur n'est détectée, la fiche victime est enregistrée avec succès dans la base de données et un accusé de confirmation est renvoyé à l'utilisateur.

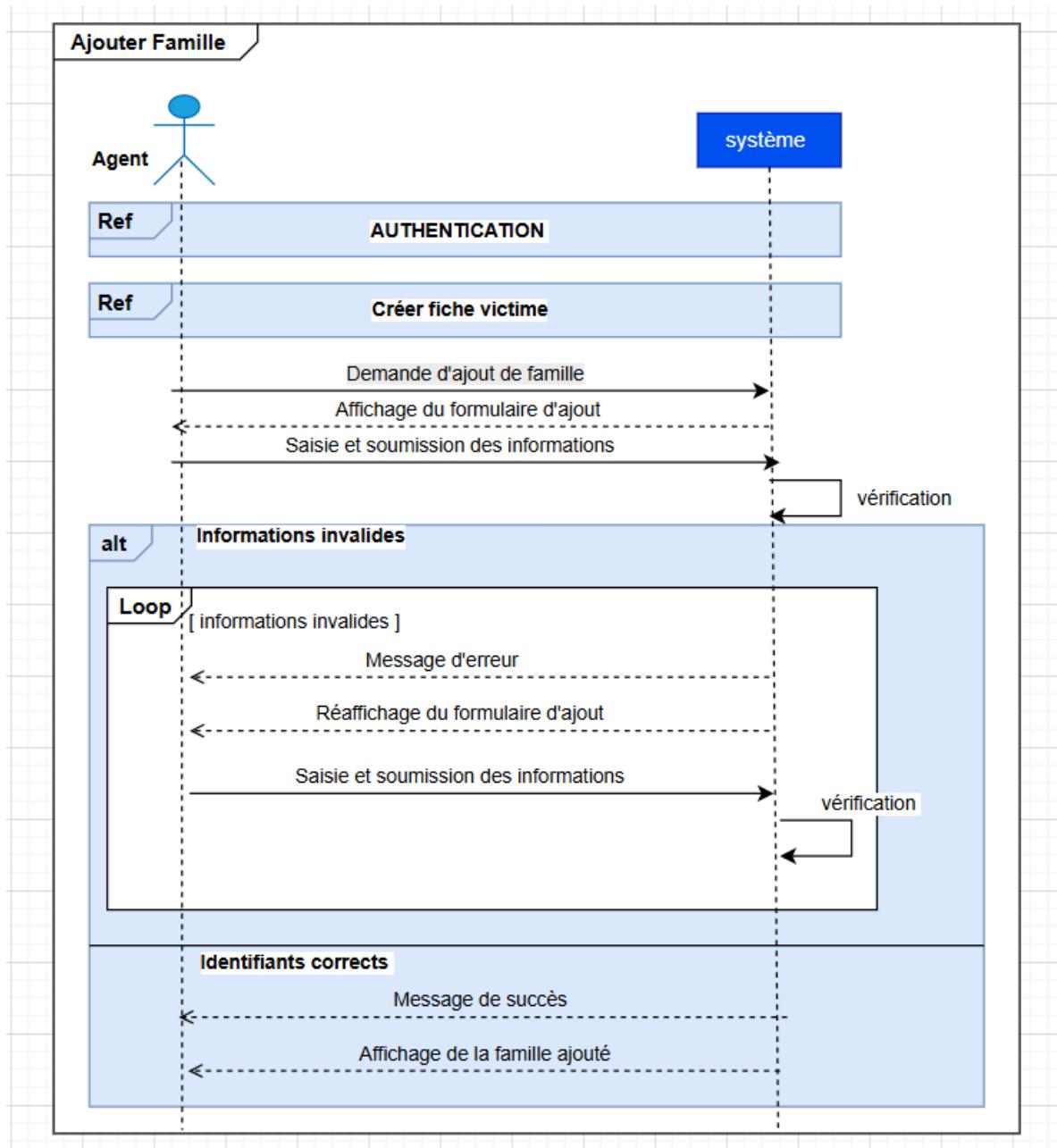


Figure 8 : Diagramme de séquence du cas d'utilisation « Ajouter Famille »

La Figure 8 illustre le fonctionnement du cas d'utilisation « Ajouter Famille ». L'utilisateur, après avoir créé une fiche victime, déclenche la commande d'ajout d'une famille. Le système lui propose alors un formulaire permettant de renseigner les informations requise

Une vérification est effectuée par le système afin de s'assurer que tous les champs obligatoires sont correctement complétés. Si certaines données sont absentes ou incorrectes, l'utilisateur reçoit un message d'erreur signalant « Informations incomplètes ou non valides ».

Lorsque les informations saisies sont valides , le système enregistre les données de la famille dans la base et confirme l'opération en affichant un message de validation.

f. Diagramme d'activité

Le diagramme d'activité offre une représentation graphique du déroulement d'un cas d'utilisation. Chaque activité, correspondant à l'exécution d'une étape précise du processus et est symbolisée par un rectangle aux coins arrondis.

Dans cette section, nous présentons les différents diagrammes d'activités élaborés pour illustrer le fonctionnement des cas d'utilisation retenus

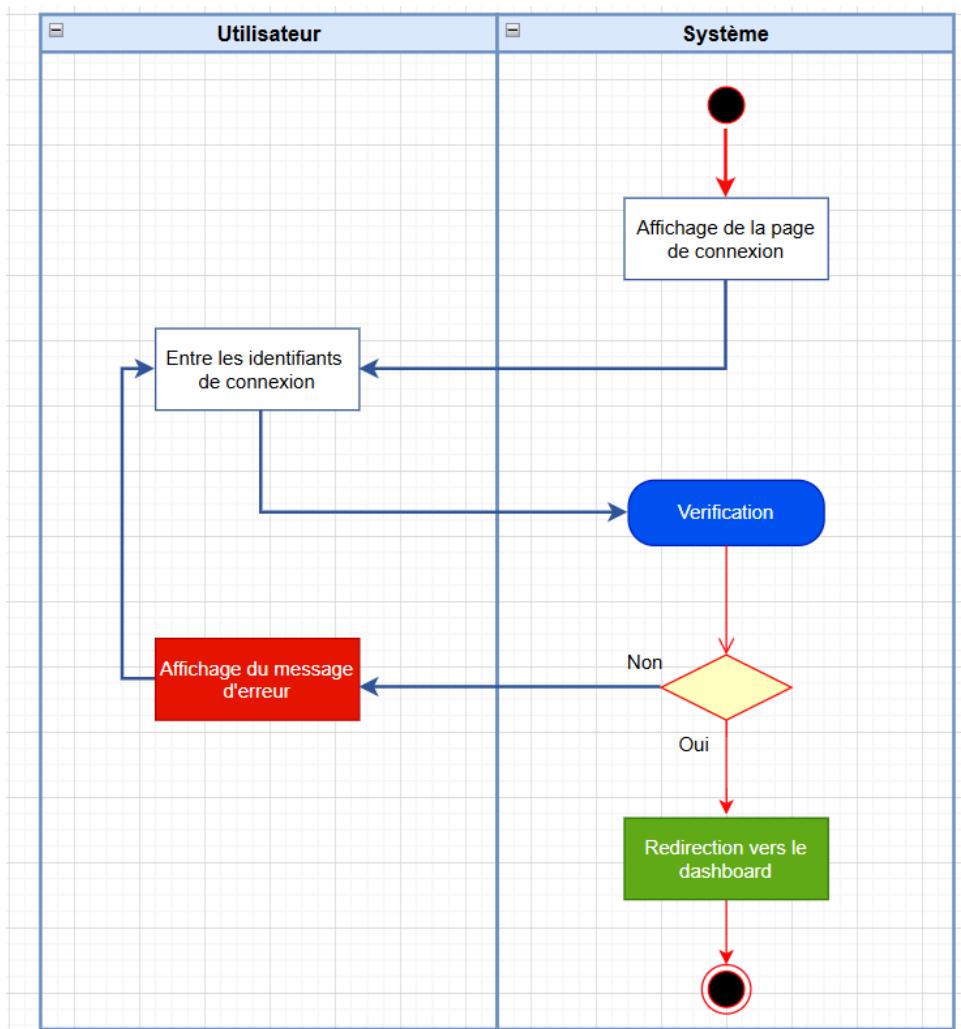


Figure 9 : Diagramme d'activité pour le CU « S'authentier »

La figure 9 met en évidence le déroulement complet du cas d'utilisation « S'authentifier » et décrit les différentes étapes du processus.

Dans un premier temps, l'utilisateur saisit ses identifiants de connexion. Ces informations sont ensuite soumises au système pour validation. Lorsque la vérification confirme que les identifiants sont corrects, l'authentification est considérée comme réussie. L'utilisateur obtient alors l'autorisation d'accès et est redirigé automatiquement vers son tableau de bord.

À l'inverse, si la validation échoue ou si l'authentification n'aboutit pas, le système affiche un message d'erreur approprié afin d'informer l'utilisateur. Dans ce cas, le processus s'interrompt. Ainsi, le diagramme illustre clairement l'ensemble du flux : depuis la saisie initiale des identifiants jusqu'à l'accès accordé ou l'affichage d'un message d'erreur.

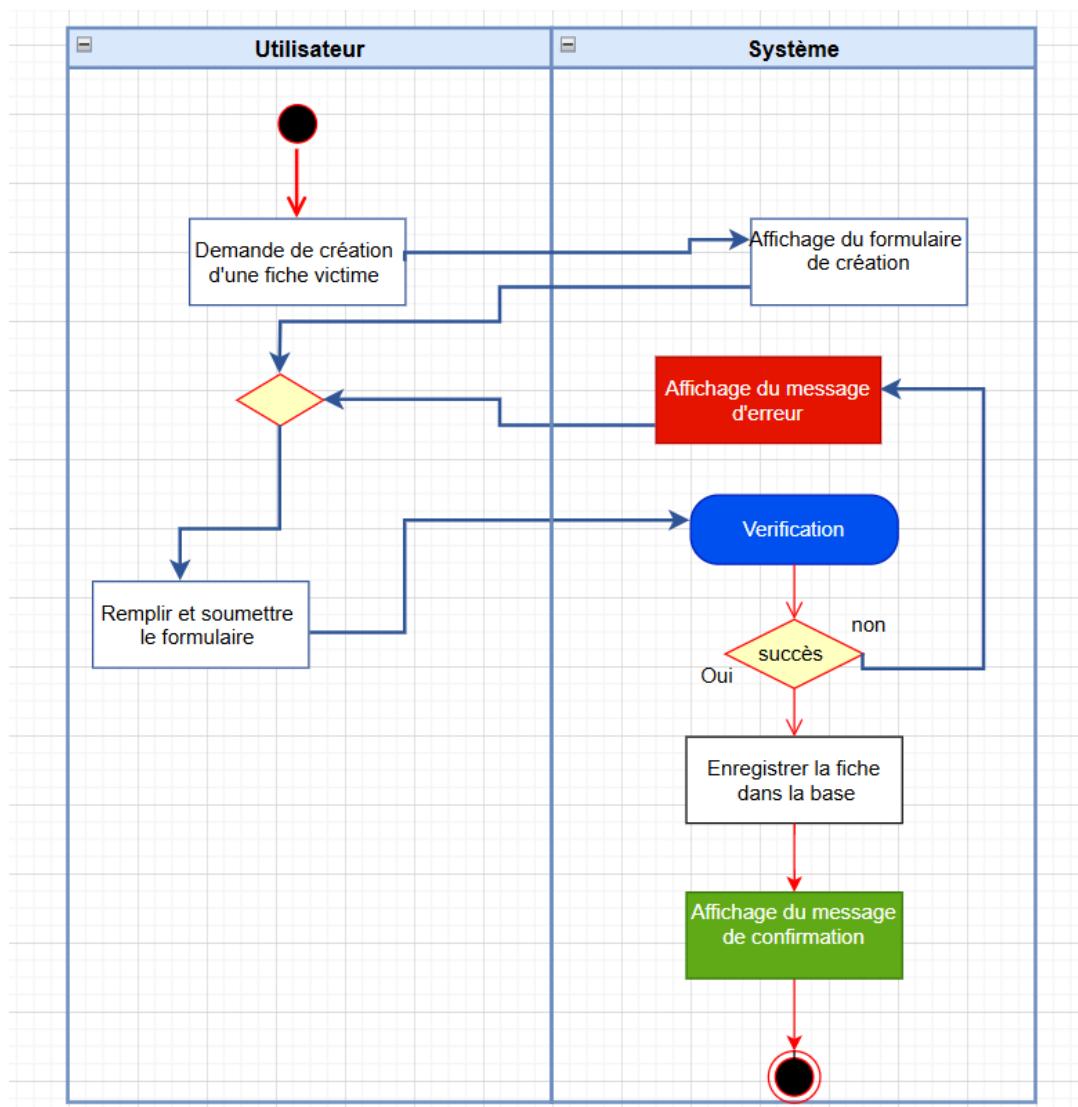


Figure 10 : Diagramme d'activité pour le CU « Créer Fiche Victime »

La figure 10 met en évidence le déroulement complet du cas d'utilisation « Créer fiche victime » et décrit les différentes étapes du processus.

Tout commence lorsque l'utilisateur, après s'être authentifié, accède à la fonctionnalité de création d'une fiche victime. Il saisit alors les informations nécessaires concernant la victime, telles que les données personnelles et les détails liés à l'incident.

Une fois ces informations fournies, le système procède à une première validation pour vérifier la complétude et la conformité des données saisies. Si des erreurs ou des champs manquants sont détectés, un message d'alerte est affiché, invitant l'utilisateur à corriger les informations.

Lorsque la saisie est jugée correcte, le système enregistre la fiche victime dans la base de données. L'opération est alors confirmée par un message de succès indiquant que la fiche a bien été créée.

Ainsi, le diagramme illustre clairement l'ensemble du flux : depuis la saisie des informations par l'utilisateur jusqu'à l'enregistrement effectif de la fiche victime ou l'affichage d'un message d'erreur.

4. Spécifications techniques

De la même manière que les spécifications fonctionnelles s'attachent à représenter les processus métier à travers différents modèles, les spécifications techniques, quant à elles, portent davantage sur la modélisation de l'architecture du système. Elles incluent également le choix des technologies à utiliser ainsi que du matériel requis afin de garantir la couverture des besoins fonctionnels identifiés.

a. Mise à disposition des conditions de travail

La gendarmerie nationale, a mis à disposition une salle de travail adaptée, une connexion internet fiable et un accompagnement permanent. Cette assistance a concerné aussi bien la fourniture d'informations en lien direct avec le projet que des explications détaillées chaque fois que nous rencontrions des difficultés.

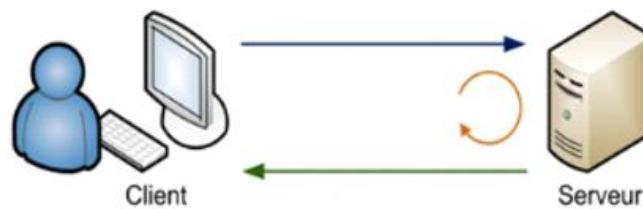
b. Architecture de développement

En informatique, une architecture désigne l'organisation et le mode de communication établis entre différents éléments physiques et/ou logiques au sein d'un réseau. Il en existe plusieurs formes appliquées au développement des systèmes. Dans le cadre de ce projet, une étude comparative de quelques types d'architectures est présentée dans le Tableau 7.

Tableau 7 : Comparaison des différentes architectures de développement

Architecture	Avantages	Inconvénients
Architecture un tiers	Les tables des données sont directement accessibles.	<ul style="list-style-type: none"> Maintenance difficile du système Surcharge la machine hôte
Architecture deux tiers	<ul style="list-style-type: none"> Applications multi-utilisateurs Charges réparties entre le client et le serveur Bonne sécurité des données 	<ul style="list-style-type: none"> Coût de déploiement élevé Évolution difficile du système
Architecture trois tiers	<ul style="list-style-type: none"> Évolution facile Déploiement aisément Une sécurité accrue 	<ul style="list-style-type: none"> nécessite un personnel informatique initié pour sa mise en œuvre

La Figure 11 illustre parfaitement les informations données dans le tableau 7 sur l'architecture deux tiers.

*Figure 11 : Schéma représentant l'architecture deux tiers*

Source : <https://images.app.goo.gl/JctssrzMR6EVX4s38>

L'architecture à deux niveaux repose sur deux entités principales : le client et le serveur de base de données.

Dans ce modèle, le client (ou utilisateur final) envoie directement une requête au serveur de base de données via une application installée sur son poste (par exemple, une application lourde). Le serveur de base de données reçoit alors la requête, l'exécute et renvoie immédiatement les résultats au client.

Cette approche est plus simple à mettre en œuvre que l'architecture à trois niveaux, car elle réduit les intermédiaires. Toutefois, elle présente certaines limites : la sécurité est plus faible (le client communique directement avec la base), la charge est plus lourde sur le serveur de base de données, et la scalabilité (capacité à supporter un grand nombre d'utilisateurs) est limitée.

L'architecture à deux niveaux est généralement utilisée dans des environnements où le nombre d'utilisateurs est restreint et où les besoins en performances et en sécurité sont moins critiques.

La Figure 12 ci-dessous illustre parfaitement les informations données dans le tableau 7 sur l'architecture trois tiers.

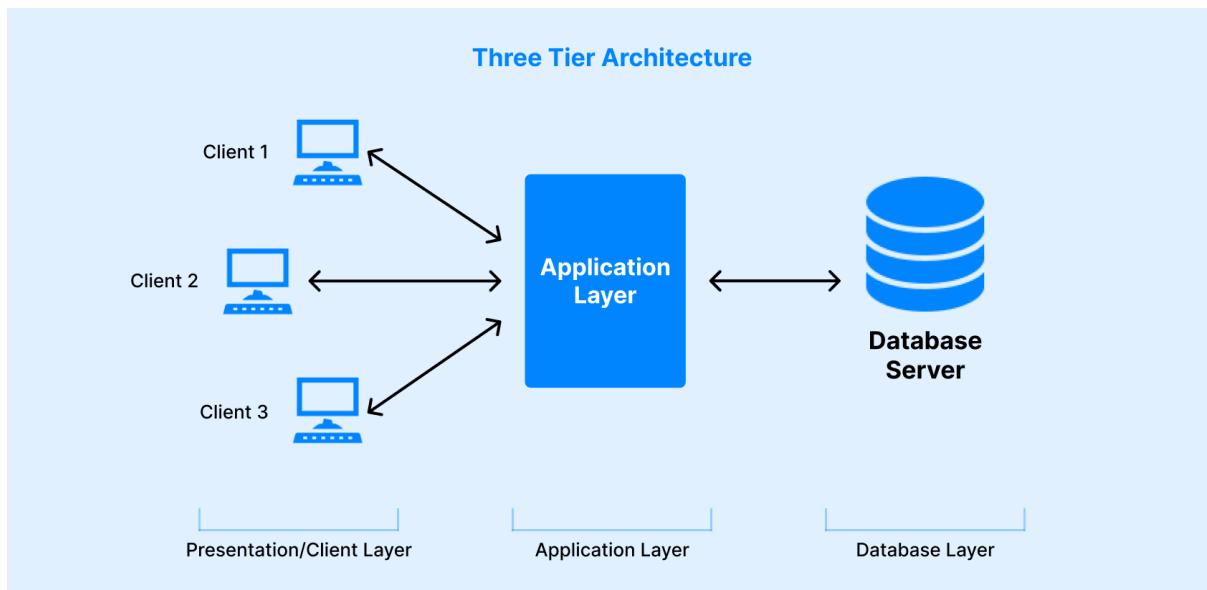


Figure 12 : Schéma représentant l'architecture trois tiers

Source :<https://cdn.hashnode.com/res/hashnode/image/upload/v1666349814710/xVeln9qtG.png?auto=compress,format&format=webp>

L'architecture à trois niveaux (ou 3-tiers) repose sur une séparation des responsabilités entre trois entités principales : le client, le serveur d'application et le serveur de base de données.

Le client, également appelé utilisateur final, initie le processus en envoyant une requête via l'interface utilisateur (par exemple, un formulaire ou une page web). Cette requête est ensuite transmise au serveur d'application, qui joue le rôle d'intermédiaire. Le serveur d'application reçoit la requête, l'analyse et exécute la logique métier nécessaire pour traiter la demande. Lorsque des données sont nécessaires, il interagit avec le serveur de base de données afin d'accéder, de modifier ou de récupérer les informations stockées.

Le serveur de base de données répond alors au serveur d'application en renvoyant les données demandées. À son tour, le serveur d'application reformule ces résultats sous une forme exploitable et compréhensible, puis les renvoie au client.

Ce modèle d'architecture garantit une meilleure sécurité, une modularité accrue et une facilité de maintenance, car chaque couche (client, application, base de données) est indépendante et peut évoluer sans perturber les autres.

Au regard de l'analyse comparative réalisée, notre choix s'est orienté vers l'architecture à trois niveaux (3-tiers). En effet, au-delà de ses avantages en matière de scalabilité, de performances, de disponibilité, de déploiement, de répartition de charge, d'évolutivité et de sécurité, elle demeure la plus conforme aux standards d'architectures modernes. Elle constitue ainsi le modèle le mieux adapté pour garantir la fiabilité et la pérennité de notre application destinée à la gendarmerie nationale.

III. Conception globale

Dans cette section, nous présenterons successivement trois éléments essentiels à la modélisation de notre application. Nous commencerons par le dictionnaire de données, qui recense et décrit de manière précise l'ensemble des données manipulées par le système. Ensuite, nous exposerons le diagramme de classes, destiné à représenter la structure statique du système ainsi que les relations entre les différentes entités. Enfin, nous terminerons avec le diagramme de déploiement, qui illustre la répartition physique des composants logiciels sur l'infrastructure matérielle prévue.

1. Diagramme de classes

Le diagramme de classes est un diagramme UML qui sert à modéliser la structure interne d'un système. Il met en évidence les classes, chacune définissant les caractéristiques (attributs) et les comportements (méthodes) associés à un ensemble d'objets.

Visuellement, une classe se représente par un rectangle divisé en trois parties : la partie supérieure indique le nom de la classe, la partie centrale regroupe ses attributs, tandis que la partie inférieure liste ses méthodes ou opérations possibles.

Dans la suite, nous présenterons le dictionnaire de données, base de référence qui nous a permis d'établir le diagramme de classes de notre système.

a. Dictionnaire des données

Un dictionnaire de données regroupe de manière centralisée l'ensemble des descriptions détaillées de toutes les données utilisées dans un système d'information. Il présente principalement les attributs des différentes données, leur description, ainsi que leur type. Le Tableau 8 ci-dessous expose ainsi le dictionnaire des données du système de gestion des familles des victimes d'attaques terroristes mis en place pour la Gendarmerie Nationale.

Tableau 8 : Dictionnaire des données

Attribut	Description	Type de donnée
id	Identifiant unique	int
grade	Grade de la victime	string
date_deces	Date du décès	date
lieu_deces	Lieu du décès	string
acte_deces	Document officiel du décès	document

adresse	Adresse de la famille	string
ville	Ville de résidence de la famille	string
date_naissance	Date de naissance du membre de la famille	date
lien_parente	Lien de parenté avec la victime	string
matricule	Matricule de l'agent de gendarmerie	string
grade	Grade de l'agent	string
unite	Unité de rattachement de l'agent	string
specialite	Spécialité de l'assistant social	string
type_demande	Type de demande ou d'aide sollicitée	string
description	Description de la demande	string
statut	Statut actuel de la demande	string
departement	Département du responsable	string
titre	Titre du rapport	string
date_generation	Date de génération du rapport	date
contenu	Contenu détaillé du rapport	string
message	Message contenu dans la notification	string
date	Date d'envoi de la notification	date
nom	Nom de l'utilisateur	string
prenom	Prénom de l'utilisateur	string
username	Nom d'utilisateur	string
motDePasse	Mot de passe de l'utilisateur	string

Le dictionnaire des données recense l'ensemble des attributs exploités lors du développement du système. Toutefois, il ne montre pas comment ces informations sont structurées dans la base de données. C'est pourquoi, dans la section suivante, nous présentons le diagramme de classes, lequel met en évidence l'organisation logique des données et les relations entre les différentes entités de notre système.

b. Diagramme de classes

Le diagramme de classes exprime la structure statique du système en mettant en évidence les classes et les relations entre elles. Nous présentons notre diagramme de classes à la figure 13. Il comporte douze (12) classes, dans lesquelles sont représentés les attributs issus du dictionnaire de données. Les classes sont représentées sous forme de rectangles identifiés par leurs noms et reliées entre elles par des associations accompagnées de cardinalités. Ces cardinalités expriment les relations possibles entre les instances, telles que : « 1..* » pour un ou plusieurs, « 0..* » pour zéro ou plusieurs et « 1..1 » pour un seul.

À titre d'exemple, considérons les classes « Fiche_Victime » et « Famille ». La relation entre ces deux classes présente la cardinalité « 1 » du côté de Famille et « 0..* » du côté de Fiche_Victime. Cela signifie qu'une famille est associée à une seule fiche victime, tandis qu'une fiche victime peut être reliée à zéro, une ou plusieurs familles. De la même manière, la classe « Famille » est reliée à la classe « Membre_Famille » avec la cardinalité « 1..* », traduisant le fait qu'une famille doit obligatoirement comporter au moins un membre et peut en compter plusieurs.

Ainsi, le diagramme de classes offre une vue claire sur l'organisation logique des entités de notre système et les liens qui les unissent, garantissant une meilleure compréhension de la structure interne de l'application.

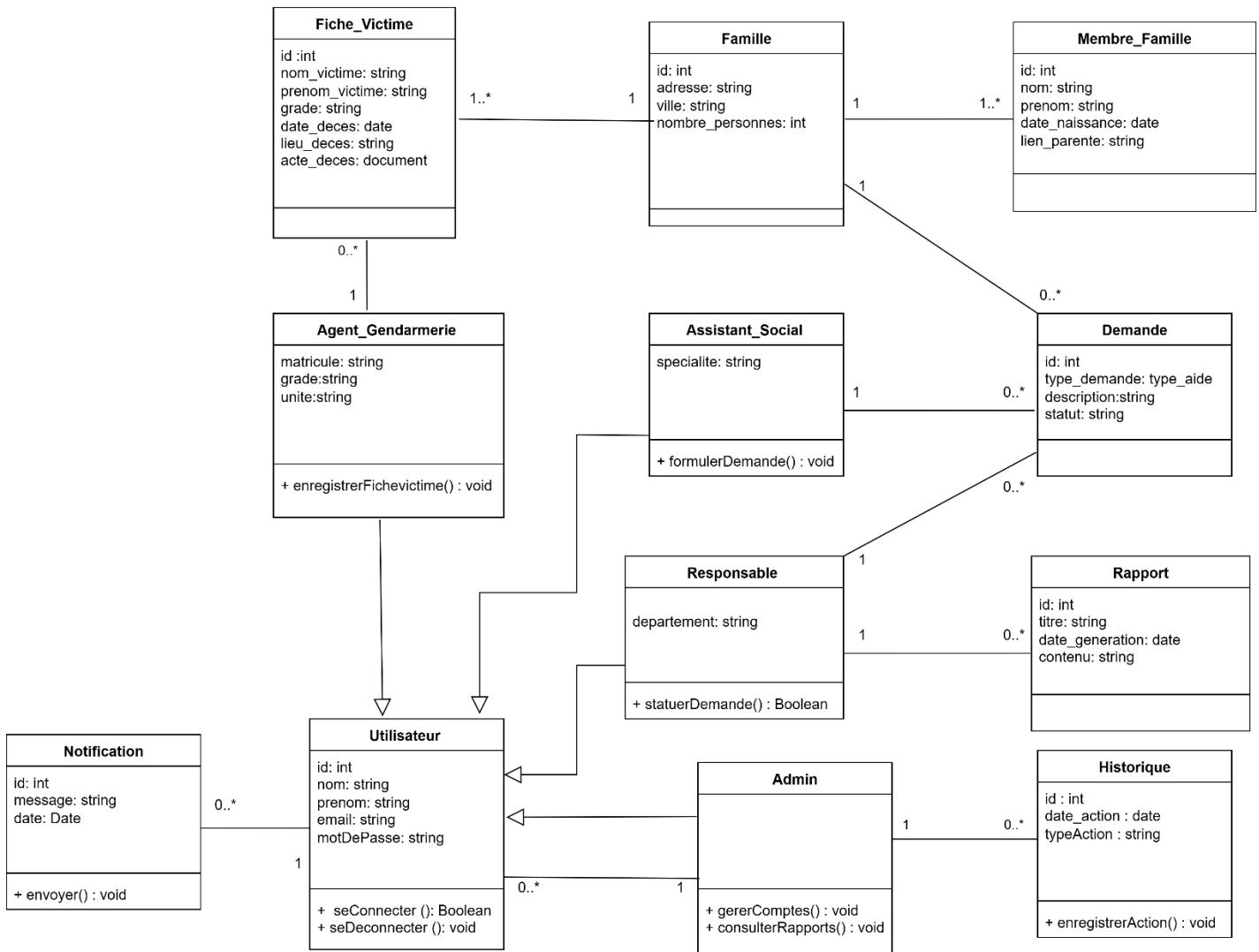


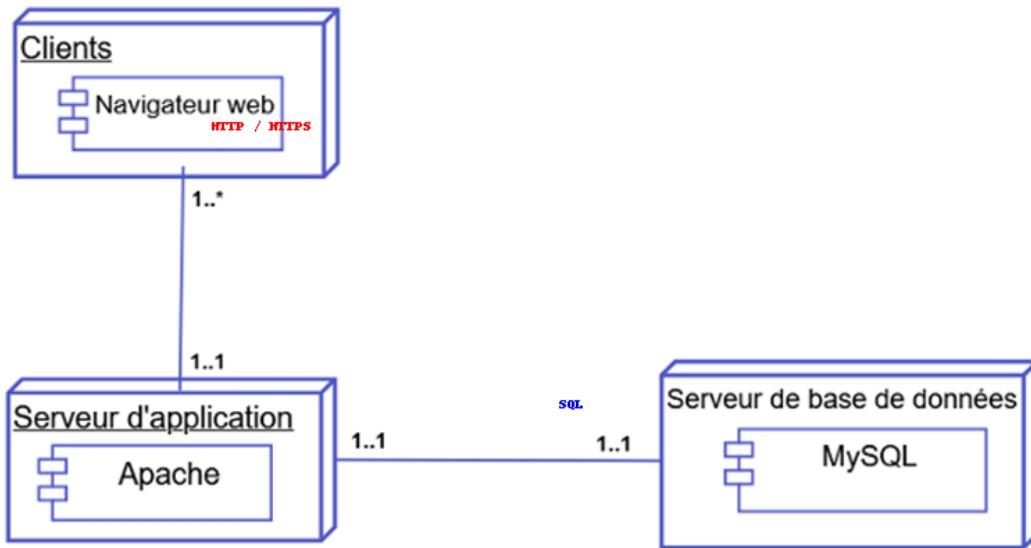
Figure 13 : Diagramme de classes

2. Diagramme de déploiement

Un diagramme de déploiement UML illustre l'architecture physique d'un système en représentant la manière dont les composants logiciels sont installés et exécutés sur les différents nœuds matériels.

Dans le cadre de notre projet de gestion des familles des victimes d'attaques terroristes, ce diagramme met en évidence la répartition des éléments logiciels (application, base de données, interface utilisateur) sur l'infrastructure technique mise à disposition.

Notre diagramme de déploiement se présente comme suit :

*Figure 14 : Diagramme de déploiement*

Notre diagramme de déploiement repose sur trois (03) nœuds principaux : le serveur de données (MySQL), le serveur d'application (Apache Server) et le client (navigateur web). Ce schéma illustre clairement l'architecture 3-tiers adoptée pour notre système.

- Le client représente l'utilisateur final, qui interagit avec l'application via son navigateur web.
- Le serveur d'application exécute la logique métier, traite les requêtes reçues des clients et assure la communication avec la base de données.
- Le serveur de base de données (MySQL) est responsable du stockage, de la gestion et de la mise à disposition des données du système.

La ligne de communication reliant le client au serveur d'application traduit l'envoi des requêtes par l'utilisateur. De même, la ligne entre le serveur d'application et le serveur de base de données illustre la dépendance de la couche applicative vis-à-vis des données stockées.

Ainsi, ce diagramme de déploiement offre une vision claire de l'architecture 3-tiers de notre solution.

Après cette étape de conception globale, nous abordons désormais la phase de réalisation du système.

IV. RÉALISATION

Cette partie est consacrée, dans un premier temps, à la présentation des outils de développement utilisés pour la réalisation de notre système. Dans un second temps, nous aborderons les tests effectués ainsi que les mesures de sécurité mises en place pour garantir la fiabilité du système. Enfin, nous terminerons par une estimation du coût de réalisation de la solution.

1. Présentation des outils de réalisation

Les outils de réalisation désignent l'ensemble des moyens mis à la disposition du développeur pour faciliter et optimiser les différentes étapes du processus de développement. Dans le cadre de notre projet, l'accent sera mis particulièrement sur les langages de programmation employés ainsi que sur les outils de conception mobilisés pour structurer et mettre en œuvre le système.

a. Choix du Système de Gestion de Base de Données

Nous distinguons plusieurs SGBD dont les plus connus sont :

- ❖ Microsoft Office Access⁵;
- ❖ MySQL⁶ ;
- ❖ Microsoft SQL Server⁷;
- ❖ Oracle Database⁸;
- ❖ PostgreSQL⁹.

⁵ https://www.microsoft.com/en-us/microsoft-365/access?ocid=ORSEARCH_Bing

⁶ <https://www.mysql.com/>

⁷ <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

⁸ <https://www.oracle.com/database/>

⁹ <https://aws.amazon.com/fr/compare/the-difference-between-mysql-vs-postgresql/>

Tableau 9 : Liste comparative des SGBD

SGBD	Avantages	Inconvénients
Microsoft Office Access	<ul style="list-style-type: none"> - Simple d'utilisation pour de petites bases - Intégration facile avec les produits Microsoft - Idéal pour des solutions locales simples 	<ul style="list-style-type: none"> - Limite de taille (~2 Go) - Peu adapté aux grandes bases - Peu performant en multi-utilisateur
MySQL	<ul style="list-style-type: none"> - Open source et gratuit (version communautaire) - Haute performance- Large communauté de support 	<ul style="list-style-type: none"> - Fonctionnalités avancées parfois limitées - Moins robuste pour les très grandes applications critiques
Microsoft SQL Server	<ul style="list-style-type: none"> - Puissant pour les applications d'entreprise - Excellente intégration Windows - Outils d'administration avancés 	<ul style="list-style-type: none"> - Coût élevé pour les éditions complètes- Majoritairement lié à l'écosystème Microsoft
Oracle Database	<ul style="list-style-type: none"> - Très robuste pour les grandes applications critiques - Fonctionnalités très avancées - Haute sécurité 	<ul style="list-style-type: none"> - Coût très élevé - Administration complexe - Peut être surdimensionné pour des projets modestes
PostgreSQL	<ul style="list-style-type: none"> - Open source- Très robuste et conforme aux standards SQL - Puissant pour des applications complexes 	<ul style="list-style-type: none"> - Administration parfois plus technique - Moins de support commercial direct que des solutions payantes

Nous avons retenu MySQL comme SGBDR car il présente plusieurs atouts majeurs pour notre projet. Tout d'abord, il s'agit d'une solution open-source et gratuite, ce qui réduit considérablement les coûts. Ensuite, il offre une administration simple et efficace, permettant une prise en main rapide. De plus, c'est le SGBDR que nous maîtrisons le mieux, ce qui garantit une utilisation optimale et adaptée aux besoins spécifiques de notre application.

Après avoir présenté les SGBD, nous allons à présent nous intéresser à un élément essentiel : les langages de programmation.

b. Langages de programmation

Les langages de programmation constituent des outils fondamentaux permettant aux développeurs de transformer des idées en solutions concrètes, de concevoir des applications performantes et de répondre efficacement à divers besoins informatiques.

Dans le cadre du développement de notre application de gestion des familles des victimes d'attaques terroristes, plusieurs langages ont été mobilisés, chacun apportant une contribution spécifique selon son rôle :

- **Python¹⁰** est un langage de programmation interprété, polyvalent et de haut niveau. Il est apprécié pour sa simplicité et sa flexibilité, ce qui le rend largement utilisé aussi bien dans le développement d'applications web que dans l'analyse de données ou encore l'automatisation de processus.
- **JavaScript¹¹** est un langage de script indispensable pour rendre les pages web interactives. Il constitue ainsi un élément central dans la création d'applications web dynamiques.
- **HTML5¹²** (*HyperText Markup Language*) est un langage de balisage qui permet de structurer le contenu des pages web. Il est la base de toute conception d'interface web.
- **CSS3¹³** (*Cascading Style Sheets*) est utilisé pour assurer la mise en forme et la présentation des pages web, permettant ainsi de séparer le contenu de l'apparence et d'améliorer l'expérience utilisateur.

¹⁰ <https://www.python.org/>

¹¹ <https://www.w3schools.com/html/>

¹² <https://www.w3schools.com/html/>

¹³ <https://www.w3schools.com/Css/>

c. Serveur d'application

Un serveur d'application a pour rôle principal d'héberger l'application et d'assurer sa disponibilité pour les utilisateurs. Il sert d'intermédiaire entre le client et la base de données en gérant la logique métier et les traitements nécessaires. L'un de ses objectifs majeurs est de permettre l'exécution de ces traitements sur une machine distante, tout en offrant au client une interface simple et légère. L'accès se fait généralement à travers un navigateur web, qui communique directement avec le serveur d'application.

- Nginx ;
- LiteSpeed ;
- Microsoft IIS ;
- Apache.

Pour faire notre choix, nous procédons par une comparaison de ces serveurs d'applications dans le tableau 14 ci-dessous.

Tableau 10 : Tableau comparatif des serveurs d'applications

Critère	Nginx	LiteSpeed	Microsoft IIS	Apache
Support HTTPS	Oui (Let's Encrypt, SSL natif)	Oui (Let's Encrypt, SSL natif)	Oui (Intégré via Windows)	Oui (SSL via modules)
Communauté	Très large et active	Plus restreinte (propriétaire)	Moyenne, concentrée sur Windows	Très large et historique
Prix	Gratuit (Open source)	Payant (version gratuite limitée)	Inclus avec Windows Server	Gratuit (Open source)
Plateformes supportées	Linux, Windows, macOS	Linux, Windows	Windows uniquement	Linux, Windows, macOS

À l'issue de cette étude comparative, notre choix s'est porté sur Apache. En effet, au-delà du fait qu'il figure parmi les serveurs web les plus utilisés au monde, il présente plusieurs atouts majeurs : il est gratuit, open source, il garantit de bonnes performances et surtout, il s'avère être l'outil que nous maîtrisons le mieux. Ces éléments font de lui la solution la plus adaptée pour la mise en œuvre de notre projet.

Dans la continuité de la présentation des outils utilisés pour la réalisation, nous abordons à présent la section consacrée aux plateformes de développement, qui ont servi de support essentiel pour la conception et la mise en œuvre de notre système.

d. Framework

Un framework, également appelé infrastructure logicielle, se définit comme un ensemble structuré et cohérent de composants logiciels. Il constitue une base solide qui facilite la création des fondations et l'organisation générale d'un logiciel, qu'il s'agisse de l'ensemble ou d'une partie de celui-ci.

Dans le cadre de notre projet, afin d'atteindre les objectifs fixés, nous avons eu recours à plusieurs frameworks, parmi lesquels :

- Django¹⁴ : Django est un framework web open-source écrit en Python qui permet de développer rapidement des applications web robustes, sécurisées et évolutives.
- Bootstrap¹⁵ 4 : Bootstrap est un Framework front-end gratuit pour un développement Web plus rapide et plus facile. Il comprend des modèles de conception basés sur HTML et CSS.

e. Autres outils

Les outils de conception regroupent l'ensemble des solutions logicielles utilisées depuis la phase de conception jusqu'à la réalisation du projet. Dans notre cas, nous avons mobilisé les outils suivants :

- Draw.io¹⁶: outil de conception qui nous a permis d'élaborer les différents diagrammes UML.

¹⁴ <https://www.djangoproject.com/>

¹⁵ <https://getbootstrap.com/>

¹⁶ <https://app.diagrams.net/>

- GanttProject¹⁷ : utilisé pour la mise en place de notre diagramme de GANTT.
- Figma¹⁸ : un outil de prototypage de sites web et d'applications, grâce auquel nous avons conçu les maquettes des différents écrans de notre application.
- PhpMyAdmin¹⁹ : employé pour l'administration de notre base de données.
- Git²⁰ : logiciel de gestion de versions décentralisé, qui nous a permis de gérer et d'archiver le code source de la plateforme.
- Visual Studio Code²¹: éditeur de code extensible, développé par Microsoft, que nous avons utilisé pour écrire et organiser notre code.
- Microsoft Office Word et PowerPoint 2019²²: des outils efficaces de traitement et de présentation de texte, qui nous ont servi pour la rédaction et la présentation de la documentation liée à notre application.

En somme, l'ensemble de ces outils de conception et de réalisation s'est révélé indispensable pour mener à bien notre projet. Chacun a joué un rôle spécifique et complémentaire, depuis la modélisation et le prototypage jusqu'au développement, à la gestion du code et à la documentation. Leur intégration harmonieuse a permis d'assurer une conception structurée et une mise en œuvre efficace du système.

Dans la partie suivante, nous aborderons un aspect tout aussi essentiel : la politique de sécurité mise en place pour garantir la fiabilité, l'intégrité et la confidentialité des données de notre application.

¹⁷ <https://www.ganttproject.biz/>

¹⁸ <https://www.phpmyadmin.net/downloads/>

¹⁹ <https://www.phpmyadmin.net/downloads/>

²⁰ <https://github.com/>

²¹ <https://code.visualstudio.com/Download>

²² https://www.microsoft.com/en-us/microsoft-365/word?ocid=ORSEARCH_Bing

2. Politique de sécurité

Une politique de sécurité informatique a pour objectif de définir clairement les règles et procédures nécessaires pour protéger les actifs informatiques d'une organisation, en particulier ses données et ses systèmes, contre les différentes menaces de sécurité.

Elle repose principalement sur trois principes fondamentaux :

- ❖ La confidentialité des données, qui vise à s'assurer que seules les personnes autorisées puissent accéder aux informations sensibles.
- ❖ L'intégrité des données, qui garantit que les informations ne soient ni altérées ni modifiées de manière non autorisée.
- ❖ La disponibilité des données, qui permet de s'assurer que les ressources et services informatiques soient accessibles en permanence aux utilisateurs légitimes.

Dans le cadre de notre projet, compte tenu de la très forte sensibilité des données traitées (informations relatives aux familles des victimes d'attaques terroristes), la politique de sécurité repose sur trois axes principaux :

- Les mesures de sécurité mises en œuvre durant le développement du système.
- Les mesures de sécurité prévues lors de la mise en production du système.
- Les mesures de sécurité appliquées après la mise en production, afin d'assurer une protection continue.

a. Mesures de sécurité prises pendant le développement

Le service de sécurité implémenté dans notre application repose principalement sur un mécanisme d'authentification, illustré dans la figure 6 par le diagramme de séquence du cas d'utilisation « *S'authentifier* ». L'accès à l'application est protégé par un nom d'utilisateur et un mot de passe. Ainsi, chaque utilisateur qui se connecte ne peut exécuter que les fonctionnalités du système correspondant aux droits associés à son profil.

L'authentification des utilisateurs est gérée grâce au système d'authentification de Django, qui offre une gestion centralisée et sécurisée des comptes. En complément, des restrictions d'accès sont appliquées sur certaines pages et informations sensibles. Par conséquent, un utilisateur

authentifié ne peut consulter ou manipuler que les données pour lesquelles il dispose des autorisations nécessaires.

b. Mesures de sécurité à prendre pour l'exploitation

Le système de gestion des familles des victimes sera hébergé chez un hébergeur externe. Ainsi, les mesures de sécurité appliquées au niveau du serveur relèveront directement des dispositifs mis en place par ce prestataire. Cette approche garantit une infrastructure robuste et sécurisée, bénéficiant des normes de sécurité de pointe adoptées par l'hébergeur.

De ce fait, les données sensibles relatives aux victimes, à leurs familles ainsi qu'aux différentes interventions de la gendarmerie seront protégées par des protocoles de sécurité rigoureux, assurant la confidentialité, l'intégrité et la disponibilité des informations. Cette configuration offre ainsi un environnement fiable et sécurisé pour l'exploitation du système, tout en renforçant la confiance des utilisateurs et des acteurs institutionnels concernés.

c. Autres mesures de sécurité

Le facteur humain joue un rôle déterminant dans toute démarche de sécurité. Il est donc essentiel de sensibiliser les utilisateurs de la plateforme à de bonnes pratiques afin de garantir un usage sûr et optimal du service. À cet effet, plusieurs consignes sont recommandées : ne jamais partager ses identifiants avec un tiers, éviter de laisser son poste de travail sans surveillance, installer et mettre à jour régulièrement un antivirus fiable, et s'abstenir autant que possible de naviguer sur des sites frauduleux ou dépourvus de certificat SSL.

Après avoir présenté la politique de sécurité, nous abordons à présent les tests logiciels réalisés dans le cadre de notre projet.

3. Tests logiciels

Le test logiciel constitue une procédure méthodique et rigoureuse qui permet d'évaluer et de vérifier les fonctionnalités, la performance ainsi que la fiabilité d'une application. Il vise à s'assurer que le système développé répond avec exactitude et conformité aux spécifications fonctionnelles définies.

Dans le cadre de notre projet, nous avons effectué des tests unitaires et des tests fonctionnels portant sur certaines fonctionnalités clés de l'application, que nous présentons en détail ci-après.

- Environnement de test

Dans notre projet, nous avons utilisé la classe TestCase de Django pour réaliser les tests. Cet outil permet de simuler des requêtes utilisateurs et de vérifier le bon fonctionnement des vues et formulaires. Il met en place une base de données temporaire dédiée aux tests, isolée de la base réelle. Cela nous a permis de tester la création d'utilisateurs et de contrôler les erreurs liées aux champs obligatoires.

Grâce aux assertions intégrées, nous avons validé que les résultats attendus correspondent aux résultats obtenus. TestCase offre ainsi un environnement de test sûr, automatisé et reproductible. Il contribue à garantir la fiabilité et la stabilité de l'application.

a. Cas de test

- Tests unitaires

Dans le cadre des tests unitaires de notre application, nous avons procédé à la vérification du processus de création d'un utilisateur. Deux scénarios principaux ont été envisagés :

- **Cas nominal** : nous avons tenté de créer un utilisateur en renseignant l'ensemble des champs obligatoires (nom d'utilisateur, mot de passe). Le système a accepté la requête et l'utilisateur a été correctement enregistré dans la base de données. Ce scénario valide le bon fonctionnement du processus de création lorsqu'il est utilisé conformément aux règles définies.
- **Cas d'erreur** : nous avons essayé de créer un utilisateur en laissant le champ *username* vide. Dans ce cas, le système a refusé la création et a renvoyé un message d'erreur indiquant l'impossibilité de poursuivre sans renseigner ce champ. Ce scénario confirme que les règles de validation des données sont bien appliquées et que le système empêche l'enregistrement d'informations incomplètes.

```
(ecomme1) PS C:\Users\user\Desktop\soutenance\gendarmerie> python manage.py test victimes.test_models
Found 1 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
Test : Création d'un utilisateur avec tous les champs obligatoires
Succès : L'utilisateur a bien été créé avec tous les champs obligatoires
OK
.
-----
Ran 1 test in 1.331s
```

Figure 15 : résultat du test de création d'un utilisateur avec les champs requis

```
• (ecomme1) PS C:\Users\user\Desktop\soutenance\gendarmerie> python manage.py test victimes.test_models
Found 1 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
Test : Création d'un utilisateur avec username vide (doit échouer)
OK : erreur Le champ username est vide - remplissez tout les champs obligatoires
.
-----
Ran 1 test in 0.002s
```

Figure 16 : résultat du test de création d'un utilisateur avec un champ manquant

- Tests fonctionnels

Dans le cadre de notre projet, nous avons réalisé des tests fonctionnels afin de vérifier le comportement global du système. Ces tests ont porté notamment sur la création d'une fiche victime et sur la validation d'une demande.

L'objectif était de s'assurer que chaque fonctionnalité réponde correctement aux besoins métiers définis. Pour la création d'une fiche victime, nous avons vérifié que toutes les informations saisies sont correctement enregistrées et affichées.

Concernant la validation d'une demande, nous avons contrôlé que le système applique bien les règles prévues selon le profil utilisateur.

Ces tests permettent de confirmer que l'application fonctionne conformément aux scénarios réels d'utilisation. Ils garantissent ainsi la qualité et la fiabilité du service rendu par la plateforme.

```
(ecomme1) PS C:\Users\user\Desktop\soutenance\gendarmerie> python manage.py test victimes.test_models
Found 1 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
Test fonctionnel : Création d'une fiche victime via la vue
Succès : Fiche victime créée avec succès
OK
.
-----
Ran 1 test in 1.345s
```

Figure 17 : résultat du test de création d'une fiche victime

```
(ecomme1) PS C:\Users\user\Desktop\soutenance\gendarmerie> python manage.py test victimes.test_models
Found 1 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
Test fonctionnel : Validation d'une demande d'aide par un responsable
Succès : Demande d'aide validée avec succès
OK
.
-----
Ran 1 test in 1.352s
```

Figure 18 : résultat du test de validation d'une demande d'aide

Après avoir présenté et décrit les différents tests effectués dans le cadre de notre projet, il convient à présent d'évaluer le coût global de réalisation du système de gestion. Cette estimation prend en compte l'ensemble des ressources mobilisées, qu'elles soient matérielles, logicielles ou humaines, afin de fournir une vision claire de l'investissement nécessaire pour la mise en place effective de la solution.

4. Estimation du cout de développement

La conception d'un logiciel s'accompagne généralement d'une étape essentielle : l'estimation du coût de réalisation. Cette démarche permet d'identifier et d'évaluer avec précision les ressources humaines, matérielles et financières nécessaires à la bonne exécution du projet. Pour ce faire, plusieurs approches existent, parmi lesquelles on retrouve notamment la méthode d'estimation par analogie, le jugement d'experts ainsi que la méthode COCOMO. Le tableau 11 ci-après présente une comparaison des principales caractéristiques de ces méthodes d'estimation de coût de développement logiciel.

Tableau 11 : Comparaison des méthodes d'estimation de coût

Méthodes d'estimations	Principes	Avantages	Inconvénients
Estimation par analogie	Le coût du projet est calculé en le comparant à d'autres projets similaires dans le même domaine d'application.	Estimations précises si les données concernant des projets semblables sont disponibles.	Impossible si des projets similaires n'étaient pas réalisés
Jugement d'experts	Un ou plusieurs experts à la fois utilisent leur expertise pour faire des prévisions de coût.	Estimations précises si les experts ont de l'expérience dans des projets similaires.	Estimations non précises s'il y a peu d'expertise.
méthode COCOMO	Il s'appuie uniquement sur la taille estimée du logiciel et sur le type de logiciel à développer.	Estimations précises, très populaires et faciles à mettre en œuvre avec divers facteurs.	Ignore les contraintes matérielles

Nous avons opté pour l'utilisation de la méthode COCOMO simplifiée (Constructive Cost Model). En effet, cette approche offre non seulement une bonne précision mais également une mise en œuvre relativement simple, ce qui correspond le mieux à nos compétences. Les explications détaillées de cette méthode figurent en Annexe 2 : La méthode COCOMO. Dans notre cas, nous avons évalué le nombre de lignes de code de l'application à environ huit mille sept cent (8700). En considérant qu'un ingénieur informaticien perçoit un salaire mensuel de 250 000 F CFA, nous pouvons alors procéder à l'estimation du coût de développement :

- Effort = $2,4 * (8700/1000)^{1.05} = 23,26 \text{ Hommes/Mois}$
- TDEV = $2,5 * \text{Effort}^{0.38} = 2,5 * (23,26)^{0,38} = 8,25 \text{ mois}$
- CDEV = $23,26 * X \text{ FCFA} = 23,26 * 250 000 \text{ FCFA} = 5 815 000 \text{ FCFA}$

Nous avons retenu la méthode COCOMO simplifiée car elle offre une estimation plus précise et objective que les approches basées uniquement sur l'intuition ou l'analogie. Contrairement au jugement d'experts, qui reste subjectif, ou à l'analogie, qui nécessite des projets de référence très proches, COCOMO repose sur un modèle mathématique validé. Elle permet d'obtenir des valeurs chiffrées en termes d'effort, de temps et de coût, facilitant ainsi la planification. De plus, sa simplicité d'utilisation en fait un outil pratique et adapté à la taille de notre projet. Enfin, elle est reconnue comme une méthode fiable dans l'estimation des coûts logiciels.

En prenant en considération le coût du développement ainsi que celui du besoin en matériel informatique et logiciel, on obtient le coût total du projet tel que présenté dans le Tableau 12.

Tableau 12 : Coût total du projet

Désignation	Quantité	Coût unitaire	Coût total
Ordinateur	1	405000 FCFA	405000 FCFA
GanttProject	1		Licence gratuite
Visual studio code	1		Licence gratuite
MySQL	1		Licence gratuite
Draw.io	1		Licence gratuite
Développement	1		5 815 000 FCFA
Coût total			6 220 000 FCFA

5. Présentation de quelques interfaces graphiques de l'application

❖ Écran de connexion

L'accès au logiciel est réservé aux utilisateurs disposant d'un compte préalablement créé par l'administrateur. Pour s'authentifier, chaque utilisateur doit renseigner son username ainsi que son mot de passe. La figure 19 ci-dessous illustre l'interface de la page de connexion.

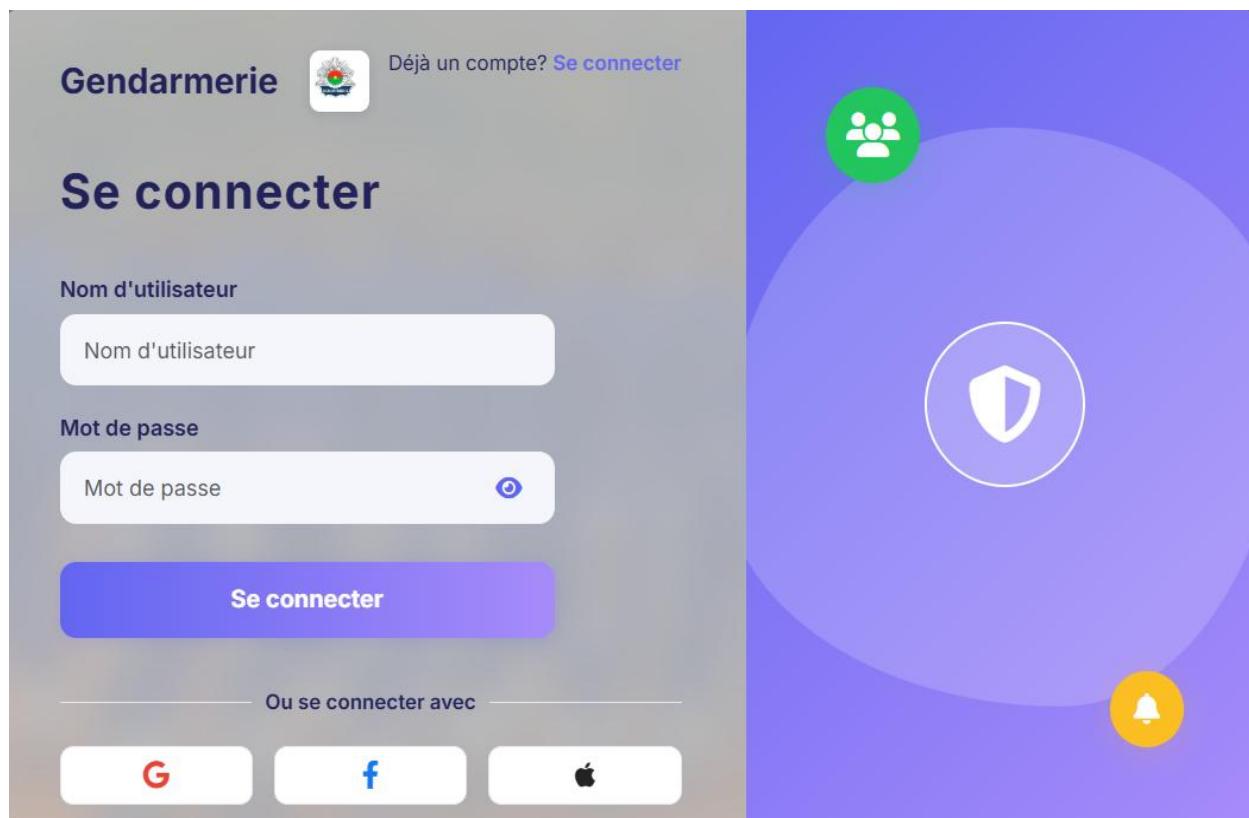


Figure 19 : Ecran de connexion

❖ Tableau de bord – Agent

Après authentification réussie, l'agent accède à son tableau de bord personnalisé. Cette interface regroupe l'ensemble des fonctionnalités qui lui sont accessibles en fonction de son profil, telles que la création d'une fiche victime, l'ajout d'une famille, ou encore le suivi des informations enregistrées. La figure 20 ci-dessous présente l'écran du tableau de bord dédié à l'agent.

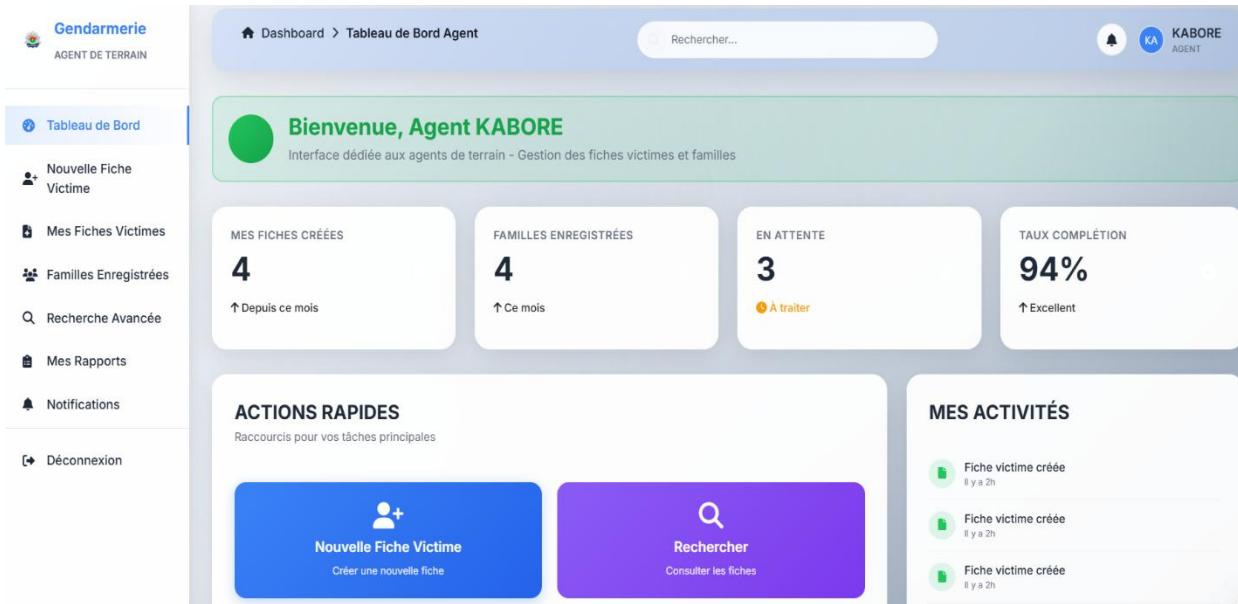


Figure 20 : tableau de bord Agent

❖ Tableau de bord – Assistant Social

Une fois connecté, l'assistant social accède à un tableau de bord conçu pour faciliter son travail. À travers cette interface, il peut soumettre et suivre les demandes d'aide, consulter les dossiers des victimes et de leurs familles, ainsi qu'intervenir dans le processus de validation des demandes. La figure 21 ci-dessous illustre l'écran du tableau de bord réservé à l'assistant social.

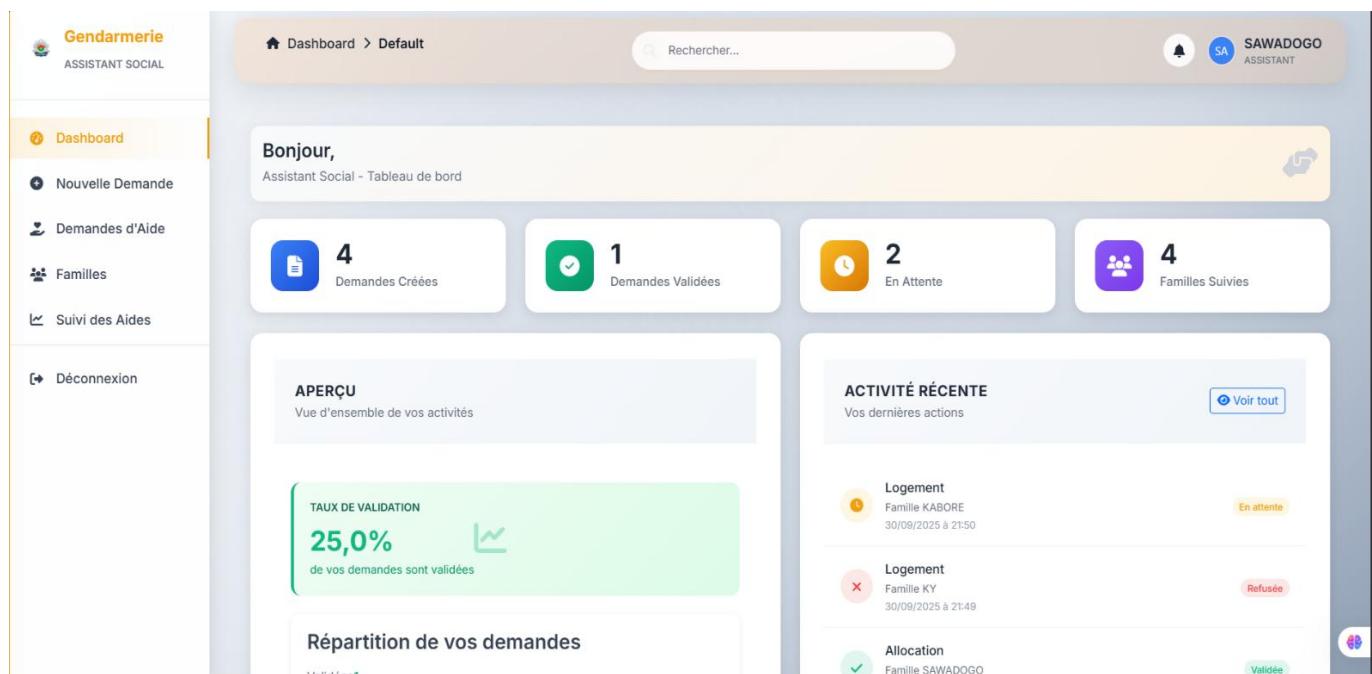


Figure 21 : tableau de bord Assistant Social

❖ Page de suivi des aides – Responsable

Le responsable dispose d'une page dédiée au suivi des aides accordées aux victimes et à leurs familles. À partir de cette interface, il peut visualiser l'état d'avancement des différentes demandes, vérifier les aides validées ou en attente et superviser l'ensemble des interventions sociales. Cette page assure une meilleure traçabilité et un contrôle rigoureux du processus d'assistance. La figure 22 ci-dessous illustre l'écran de suivi des aides réservé au responsable.

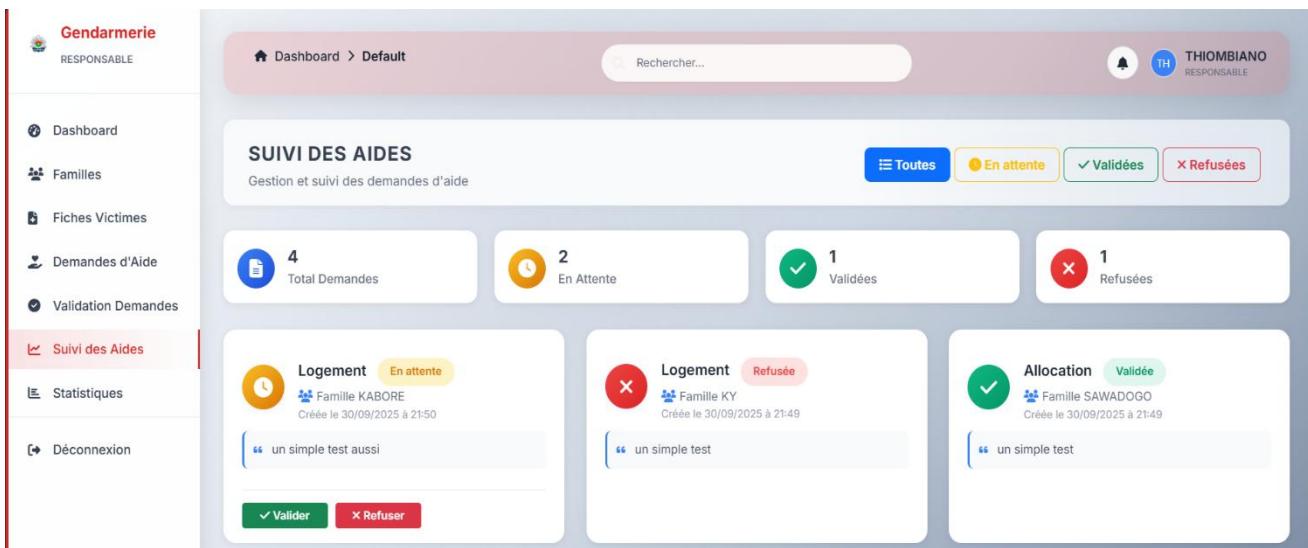


Figure 22 : Écran de suivi des aides

❖ Page de gestion des victimes – Administrateur

L'administrateur dispose d'une page spécifique qui lui permet d'accéder à la liste complète des victimes enregistrées dans le système. Depuis cette interface, il peut consulter les informations détaillées de chaque victime, effectuer des mises à jour, supprimer des enregistrements ou encore ajouter de nouvelles fiches en cas de besoin. Cette page constitue un outil central pour assurer la cohérence et la fiabilité des données dans le système. La figure 23 ci-dessous illustre la page de gestion des victimes accessible par l'administrateur.

The screenshot shows a user interface for managing victim records. The left sidebar has a dark theme with white icons and text. The main area has a light background with a header bar at the top.

Header:

- Logo: Gendarmerie
- User: gerald
- Navigation: Accueil, Utilisateurs, Victimes, Rechercher Utilisateurs..
- Language: English (US)

Breadcrumbs: Fiche victimes > Accueil > Victimes > Fiche victimes

Search and Filter:

- Grade dropdown: Grade
- Date of death dropdown: Date de décès
- Search input field: Rechercher
- Text input field: -----
- Send button: Envoyer
- Count: 0 sur 4 sélectionné

Table Headers:

Prénom	Nom	Grade	Date de décès	Famille	Cree par
--------	-----	-------	---------------	---------	----------

Table Data:

<input type="checkbox"/> sylvain	kabore	Sous-Lieutenant	5 septembre 2025	Famille KABORE -	KABORE
<input type="checkbox"/> Faysa	ky	Adjudant	11 septembre 2025	Famille KY -	KABORE
<input type="checkbox"/> BENÔT	sawadogo	Sergent	23 septembre 2025	Famille SAWADOGO -	KABORE
<input type="checkbox"/> Mahamadi	OUEDRAOGO	Caporal	29 septembre 2025	Famille ouedraogo -	KABORE

Footer:

- 4 fiche victimes
- Groups

Figure 23 : Écran de gestion des victimes

CHAPITRE III : BILAN DU STAGE

Après avoir présenté les différentes activités réalisées dans le cadre de la conception et de la mise en place du système de gestion des familles des victimes d'attaques terroristes à la Gendarmerie Nationale, nous consacrons ce dernier chapitre au bilan de notre stage. Pour ce faire, nous allons d'abord présenter le déroulement du stage ainsi que les principales tâches accomplies au cours de son exécution. Ensuite, nous formulerons nos critiques et proposerons quelques suggestions pour améliorer le projet et son environnement de travail.

I. DÉROULEMENT DU STAGE ET ACTIVITÉS MENÉES

Dans cette partie, nous présentons la manière dont s'est déroulé notre stage au sein de la Gendarmerie Nationale. Dès le début, nous avons été accueillis dans un environnement de travail favorable qui nous a permis de bien nous imprégner des objectifs assignés. Le stage s'est déroulé en plusieurs étapes allant de la prise de contact avec le milieu professionnel, à la compréhension des besoins spécifiques de la Gendarmerie en matière de gestion des familles des victimes d'attaques terroristes, jusqu'à la conception et la réalisation de l'application. Tout au long de cette période, nous avons bénéficié d'un suivi constant et de conseils avisés qui ont facilité notre intégration et l'atteinte progressive des objectifs fixés.

1. Activités menées

Au cours de notre stage au sein de la Gendarmerie Nationale, nous avons participé à plusieurs activités qui ont constitué des étapes clés dans la réalisation de notre projet. Ces activités peuvent être regroupées comme suit :

- Prise de contact et compréhension du contexte

Dès le début du stage, nous avons échangé avec les responsables afin de mieux comprendre le cadre dans lequel s'inscrit le projet. Nous avons pris connaissance des réalités de la Gendarmerie Nationale et des difficultés rencontrées dans la gestion des familles des victimes d'attaques terroristes.

- Analyse des besoins

Nous avons procédé à la collecte et à l'analyse des besoins fonctionnels et techniques. Cette étape nous a permis de définir les fonctionnalités essentielles du système telles que la gestion des victimes, le suivi des aides, la validation des demandes et la génération de rapports.

- Conception de l'application

Sur la base des besoins exprimés, nous avons réalisé la conception de l'application en élaborant les diagrammes UML (cas d'utilisation, séquence, classes). Cette phase a également permis de définir l'architecture du système à mettre en place.

- Développement du système

Nous avons procédé au codage des différentes fonctionnalités en utilisant le framework Django. Cette étape a été marquée par la mise en place des modules de gestion des utilisateurs, de création de fiches victimes, de validation des demandes et de suivi des aides.

- Tests et validation

Une fois le développement effectué, nous avons réalisé des tests unitaires et fonctionnels afin de vérifier la conformité du système par rapport aux besoins initiaux. Ces tests ont permis d'assurer la fiabilité et la qualité de l'application.

- Rédaction du rapport

Enfin, nous avons consigné l'ensemble du travail réalisé dans ce rapport, en mettant en avant les différentes étapes du projet, les outils utilisés et les résultats obtenus.

2. Connaissances acquises

Ce stage au sein de la Gendarmerie Nationale nous a permis d'acquérir un ensemble de connaissances aussi bien techniques que professionnelles. Sur le plan technique, nous avons approfondi nos compétences en développement web grâce à l'utilisation du framework Django, en particulier sur la gestion des bases de données, la création de formulaires sécurisés et la mise en œuvre des tests unitaires et fonctionnels. Nous avons également renforcé nos connaissances dans la modélisation UML et l'estimation des coûts de développement logiciel à travers l'application de la méthode COCOMO.

Sur le plan méthodologique, nous avons appris à conduire un projet de bout en bout, depuis l'analyse des besoins jusqu'aux tests et à la validation. Cela nous a permis de mieux comprendre l'importance d'une planification rigoureuse et d'une organisation structurée dans la réussite d'un projet informatique.

Enfin, sur le plan professionnel, nous avons découvert le fonctionnement interne d'une institution telle que la Gendarmerie Nationale et compris les exigences liées à la sécurité, à la

confidentialité et à la fiabilité des systèmes d'information utilisés dans ce contexte sensible. Cette expérience nous a également permis de développer des qualités comme l'adaptabilité, le sens de l'écoute et la rigueur dans le travail.

3. Les Perspectives

Dans la continuité de ce travail, plusieurs pistes d'amélioration et d'évolution peuvent être envisagées pour le système de gestion des familles des victimes d'attaques terroristes.

- Interopérabilité : connecter le système avec d'autres bases de données nationales pour faciliter le partage et la centralisation des informations relatives aux victimes.
- Automatisation des aides : développer un module de suivi automatique permettant de générer des rappels et des notifications concernant l'attribution ou le renouvellement des aides.
- Accessibilité mobile : créer une application mobile permettant aux agents et assistants sociaux de consulter et mettre à jour les informations en temps réel, même en déplacement.

II. OBSERVATIONS ET SUGGESTIONS

Nous adressons nos vifs remerciements à la Gendarmerie Nationale pour nous avoir accordé ce stage qui fut d'un grand apport aussi bien sur le plan académique que professionnel. Nous avons particulièrement apprécié l'accueil chaleureux du personnel ainsi que la disponibilité des encadreurs qui nous ont guidés et accompagnés tout au long de notre apprentissage. Les conditions de travail mises à notre disposition ont favorisé une immersion réussie dans le domaine de la gestion et de la sécurisation des systèmes d'information.

Au regard des résultats atteints, nous suggérons à la Gendarmerie Nationale de poursuivre ses efforts dans la modernisation de ses outils numériques afin d'optimiser davantage la gestion des données sensibles, notamment celles concernant les familles de victimes d'attaques terroristes. Nous encourageons également l'institution à renforcer la formation continue de son personnel dans le domaine des technologies de l'information, afin de tirer le meilleur parti des solutions innovantes mises en place.

CONCLUSION GÉNÉRALE

En conclusion, ces trois (03) mois de stage au sein de la Gendarmerie Nationale ont constitué une expérience profondément enrichissante qui nous a permis de relever un défi majeur : la mise en place d'un système numérique dédié à la gestion des familles des victimes d'attaques terroristes. Nous avons conçu et réalisé une application web qui facilite la centralisation, le suivi et le traitement des informations relatives aux victimes et à leurs familles, contribuant ainsi à une meilleure organisation et à une plus grande efficacité dans la prise en charge.

Sur le plan technique, ce stage a été l'occasion de renforcer nos compétences en conception et développement d'applications web en mobilisant différents outils et technologies. Sur le plan professionnel, il nous a offert une véritable immersion dans un environnement sensible et stratégique, en nous permettant d'appliquer concrètement les connaissances théoriques acquises à l'IBAM.

Les objectifs fixés au départ ont été atteints, et le système mis en place apporte une réelle valeur ajoutée à la gestion interne.

En perspective, nous envisageons le développement d'une application mobile pour rendre le système encore plus accessible sur le terrain, ainsi que l'intégration de fonctionnalités avancées telles que des statistiques d'aide et un suivi automatisé des demandes. Ces évolutions permettront de garantir la pérennité et l'adaptabilité de la solution face aux besoins croissants de l'institution.

Nous tenons enfin à saluer la qualité de la formation dispensée par l'IBAM, qui nous a préparés efficacement à affronter les réalités du monde professionnel. Nous exprimons également notre profonde gratitude à la Gendarmerie Nationale pour la confiance placée en nous et pour cette opportunité, espérant que notre contribution participe à l'amélioration de la gestion et du soutien aux familles des victimes.

BIBLIOGRAPHIE ET WEBOGRAPHIE

1. Bibliographie

[1]. **GUIRE Ismaël Faïssal (2023)** « Automatisation des processus d'approbation dans l'administration des systèmes informatiques au sein de TICANALYSE » Rapport de fin de cycle de Licence : MIAGE. Ouagadougou : Institut Burkinabè des Arts et Métiers (IBAM), 84p.

[2]. **SAWADOGO Rimalguedo Rahima (2023)** « Conception et réalisation d'une plateforme de gestion intégrée et de suivi des startups incubées à SIRA LABS » Rapport de fin de cycle de Licence : MIAGE. Ouagadougou : Institut Burkinabè des Arts et Métiers (IBAM),64p.

2. Webographie

[3] MORONVAL,Christine. « Architecture logicielle et modèles de conception ». Softfluent.
<https://www.softfluent.fr/blog/architecture-logicielle-et-modeles-de-conception/>

(consulté le 30/07/2025).

[4] « Présentation du langage de modélisation unifié (UML) ». Cybermédia (blog),
<https://www.cybermedian.com/fr/unified-modeling-language-uml-introduction/>

consulté le 9/08/2025

[5] Ambrosy, Mathieu. « Architecture client-serveur ». Consulté le (20/08/2025).
<https://www.geonov.fr/architecture-client-serveur/>.

[6] Les SGBD. « Comparaison générale des SGBD », consulté le (26/08/2025).
<https://sgbdslam5.wordpress.com/comparaison-generale-des-sgbd/>

[7] appvizer.fr. « Les 11 meilleurs logiciels de gestion de projets gratuits 2023 ». Consulté le (10/09/2025).
<https://www.appvizer.fr/magazine/operations/gestion-de-projet/meilleurs-logiciels-gestion-de-projet-gratuits-2021>

[8] « outil open-source qui utilise des descriptions textuelles simples pour dessiner des diagrammes UML. » Consulté le 22/09/2025 .
<https://plantuml.com/fr/>

[9] OpenAI. GPT- 4.1 - modèle d'IA développé par OpenAI.
<https://chat.openai.com/%5d%20.>
 (Page consulté plusieurs fois pendant le développement).

[10] « Stack Overflow - où les développeurs apprennent, partagent et construisent des carrières ». Consulté plusieurs fois pendant le développement.
<https://stackoverflow.com/>.

TABLE DES MATIERES

SOMMAIRE	Erreurs ! Signet non défini.
DÉDICACE.....	ii
REMERCIEMENTS	iii
LISTES DES SIGLES ET ABREVIATIONS	iv
LISTE DES FIGURES ET GRAPHIQUES.....	v
LISTE DES TABLEAUX	vi
INTRODUCTION GENERALE.....	1
CHAPITRE I.....	2
PRESENTATION DES STRUCTURES DE FORMATION ET D'ACCUEIL.....	2
I. Présentation de la structure de formation	2
1. Objectifs de l'IBAM.....	2
2. Organisation	2
a. Organe statutaire.....	3
b. Organe d'exécution	4
3. Filières de formations.....	4
II. Présentation de la structure d'accueil.....	5
1.1. Historique	5
1.2. Les Missions.....	6
1.3. Organisation de la Gendarmerie Nationale	6
1.3.1. L'Etat-major	6
1.3.2. Les légions de gendarmerie nationale (RGN)	7
1.3.3. Les groupements de gendarmerie départementale (GD).....	7
1.4. Présentation du service informatique de la Gendarmerie.....	8
1.5. Organigramme de la Gendarmerie Nationale.....	9
CHAPITRE II : ANALYSE ET CONCEPTION	10
I. ÉTUDE PRÉALABLE.....	10
1. Présentation du Thème	10
a. Préliminaire	10
b. Problématique.....	11
c. Objectifs	12
d. Résultats attendus.....	12

2.	Méthode d'analyse et conception	12
a.	Cycle de développement	14
b.	Langage de modélisation.....	18
3.	Groupe de travail	20
a.	Groupe de pilotage	20
b.	Groupe de projet.....	20
c.	Groupe des utilisateurs	21
d.	Planning prévisionnel.....	21
II.	Expressions des besoins	22
1.	Étude de l'existant.....	22
a.	Présentation du système utilisé à la Gendarmerie Nationale	22
b.	Exigences fonctionnelles.....	23
2.	Présentation du processus de fonctionnement amélioré.....	24
3.	Spécification fonctionnelle.....	25
a.	Identification des acteurs.....	25
b.	Identification des cas d'utilisation.....	25
c.	Diagramme des cas d'utilisation	27
d.	Description textuelle de quelques cas d'utilisation	28
e.	Diagramme de séquence.....	32
f.	Diagramme d'activité.....	37
4.	Spécifications techniques	39
a.	Mise à disposition des conditions de travail.....	39
b.	Architecture de développement.....	39
III.	Conception globale.....	43
1.	Diagramme de classes	43
a.	Dictionnaire des données	43
b.	Diagramme de classes	45
2.	Diagramme de déploiement	46
IV.	RÉALISATION	48
1.	Présentation des outils de réalisation	48
a.	Choix du Système de Gestion de Base de Données	48
b.	Langages de programmation	50
c.	Serveur d'application	51

d. Framework	52
e. Autres outils	52
2. Politique de sécurité	54
a. Mesures de sécurité prises pendant le développement.....	54
b. Mesures de sécurité à prendre pour l'exploitation	55
c. Autres mesures de sécurité.....	55
3. Tests logiciels	55
4. Estimation du cout de développement	58
5. Présentation de quelques interfaces graphiques de l'application	60
CHAPITRE III : BILAN DU STAGE.....	65
I. DÉROULEMENT DU STAGE ET ACTIVITÉS MENÉES	65
1. Activités menées	65
2. Connaissances acquises.....	66
3. Les Perspectives	67
II. OBSERVATIONS ET SUGGESTIONS.....	67
CONCLUSION GÉNÉRALE	68
BIBLIOGRAPHIE ET WEBOGRAPHIE	69
1. Bibliographie.....	69
2. Webographie.....	69
TABLE DES MATIERES	70
ANNEXES	i
ANNEXE 1 : LANGAGE UML	i
ANNEXE 2 : LA MÉTHODE COCOMO	ii

ANNEXES

ANNEXE 1 : LANGAGE UML

UML est l'acronyme anglais pour « Unified Modeling Language ». Il se traduit par « Langage de modélisation unifié ». C'est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée permettant de visualiser la conception d'un système. Ce langage est utilisé pour la spécification, la visualisation, la modification et la construction des documents nécessaires au bon développement d'un logiciel orienté objet. Il offre un standard de modélisation, pour représenter l'architecture logicielle. Grâce à UML, il est possible de générer tout ou une partie du code d'un logiciel à partir des divers documents réalisés. Ce langage de modélisation nous offre principalement 13 diagrammes (depuis sa deuxième version) pour modéliser un système. Ces diagrammes peuvent être utilisés selon la phase du développement d'un logiciel. En analyse, nous pouvons utiliser des :

- ❖ Diagrammes de cas d'utilisation : modélisent les besoins des utilisateurs ;
- ❖ Diagrammes de séquences vus de l'extérieur : présentent les scénarios entre les utilisateurs ;
- ❖ Diagrammes d'activités : c'est un enchaînement d'actions représentant un comportement du logiciel. En phase de conception, le développeur peut utiliser des :
- ❖ Diagrammes de classes : pour représenter la structure interne du logiciel ;
- ❖ Diagrammes d'objets : pour présenter l'état interne du logiciel à un instant donné ;
- ❖ Diagrammes d'états-transitions : pour présenter l'évolution de l'état d'un objet ;
- ❖ Diagrammes de séquence vue de l'intérieur : pour montrer les scénarios d'interactions avec les utilisateurs au sein du logiciel ;
- ❖ Diagrammes de composants : pour présenter les composants physiques du logiciel ;
- ❖ Diagrammes de déploiement : pour l'organisation matérielle du logiciel.

Ces diagrammes sont rarement tous implémentés dans le cadre du même projet. Le choix des diagrammes à mettre en œuvre dans la modélisation est généralement fonction de la nature du projet et de sa taille.

ANNEXE 2 : LA MÉTHODE COCOMO

Un grand nombre de méthodes est mis à la disposition des développeurs pour estimer le coût de leurs plateformes. Nous utiliserons la méthode Constructive Cost Model (COCOMO) pour l'estimation du cout total du développement (CTDEV) de notre application du fait de sa fiabilité. De plus, cette méthode permet également d'estimer le temps de développement (TDEV) du système correspondant au temps requis pour terminer le projet avec toutes les ressources disponibles. La méthode COCOMO se base principalement sur la complexité de l'application à développer qui correspond à l'un des trois (03) types suivants :

- ❖ S : ce sont des applications simples, n'ayant que peu de cas particuliers et de contraintes. Elles sont parfaitement déterministes.
- ❖ P : ce sont des applications intermédiaires, plus complexes que les applications de type S. Elles restent tout de même déterministes bien que le nombre de leurs cas particuliers et de tests soit plus important que pour les applications de type S.
- ❖ E : ce sont des applications très complexes, que ce soit au niveau de leurs contraintes, comme un système temps réel, où au niveau des données saisies, comme certaines interfaces graphiques ou on ne peut envisager toutes les possibilités de saisies qu'un utilisateur pourrait effectuer. Elles ne sont pas déterministes.

La formule de calcul COCOMO est présentée dans le Tableau 13 .

Tableau 13 : Formule de calcul COCOMO

Complexité	Effort (en Homme Mois)	Temps de développement (TDEV en mois)
S	Effort = 2,4 * KLS ^{1,05}	TDev = 2,5 * Effort ^{0,38}
P	Effort = 3 * KLS ^{1,12}	TDev = 2,5 * Effort ^{0,35}
E	Effort = 3,6 * KLS ^{1,2}	TDev = 2,5 * Effort ^{0,32}

NB : HM est le nombre d'« homme mois » nécessaire à la réalisation du projet, et KLS est le nombre de Kilo Lignes Sources. Un homme mois correspond à 152 heures de travail effectif. Le nombre de personnes requis pour réaliser le projet dans cet intervalle de temps est donc : N = HM/TDEV. Étant donné que le salaire moyen d'un informaticien est de X FCFA, le coût total de développement pour ce projet est : CTDEV = HM*X.