RÉPUBLIQUE DU BÉNIN
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ D'ABOMEY-CALAVI

INSTITUT DE FORMATION ET DE RECHERCHE EN INFORMATIQUE

BP 526 Cotonou Tel : +229 21 14 19 88
http://www.ifri-uac.net Courriel : contact@ifri.uac.bj

MÉMOIRE

pour l'obtention du

Diplôme de Licence en Informatique

Option : Génie Logiciel

Présenté par :
Rivaldo Clarin Zindo **ASSOKPE**

Optimisation des Services Financiers de la Direction Centrale du Service du Commissariat des Armées à travers les e-Services : Étude et Réalisation d'une Application dédiée

Sous la supervision :
M. Babatoundé Aristide **YAGBO**

Membres du jury :

Dr Ratheil HOUNDJI	Maître Assistant	UAC	Président
M. Lémec AHOUANDJINOU	Ingénieur	UAC	Examinateur
M. Babatoundé Aristide YAGBO	Ingénieur	UAC	Rapporteur

Année Académique : **2022 - 2023**

Sommaire

Dédicace ii

Remerciements iii

Résumé iv

Abstract v

Table des figures vi

Liste des acronymes vii

Introduction 1

1 Revue de littérature 3

2 Solutions, choix techniques et implémentation 9

3 Résultats et analyse critique 25

Conclusion 34

Webographie 35

Table des matières 36

Dédicace

A

Mes parents bien-aimés, en reconnaissance de leurs nombreux sacrifices, de leur amour constant, de leur soutien indéfectible, et de leurs prières tout au long de mon parcours éducatif.
Mes estimés enseignants qui ont su me transmettre les connaissances indispensables pour réussir mes études.
Monsieur François BEKPON, Directeur Informatique de la Direction Générale du Budget pour m'avoir encadré durant mes trois (03) mois de stage et aidé à finir mon projet.
L'administration de l'Institut de Formation et de Recherche en Informatique (IFRI) qui a su me fournir les conditions appropriées pour suivre ma formation sur une période de trois (03) ans.

Remerciements

Je tiens à remercier :

- Dieu le Père Tout-Puissant, pour m'avoir accordé la santé, la force, la persévérance tout au long de cette tâche.
- Ma famille, source constante de soutien tout au long de mes trois années de formation.
- Mes parents, qui ont assuré un accompagnement indéfectible sur les plans économique, moral et affectif durant mon parcours éducatif.
- L'IFRI, et particulièrement au Professeur Eugène C. EZIN, Directeur de l'institut, pour son engagement en faveur de la réussite des étudiants.
- Le Directeur Adjoint de l'IFRI, le Professeur Gaston EDAH, pour son travail assidu et son accompagnement tout au long de ma formation.
- Monsieur Babatoundé Aristide YAGBO mon maître mémoire, pour avoir consacré son temps et apporté l'accompagnement nécessaire.
- Monsieur François BEKPON, pour son soutien inestimable et ses conseils précieux tout au long de l'élaboration de ce mémoire.
- L'ensemble du personnel administratif et tous les enseignants de l'IFRI pour leur contribution inestimable à ma formation.
- Le personnel administratif du Ministère de l'Économie et des Finances pour leur soutien dans la réalisation de mon projet.
- Tous mes camarades et amis qui ont partagé ce parcours avec moi.

Résumé

La dématérialisation des processus administratifs revêt une importance capitale pour la Direction Centrale du Service du Commissariat des Armées (DCSCA), visant à moderniser et optimiser la gestion du traitement de la solde. Cela représente une étape significative dans l'engagement de la DCSCA en faveur de l'efficacité, de la transparence et de la satisfaction des administrés. Ce projet s'inscrit dans une démarche d'amélioration continue des services, offrant un accès simplifié, une réduction des délais de traitement et une sécurité accrue des données à caractère personnel. La réussite d'un projet nécessite l'accès à des ressources essentielles telles que des ressources financières, matérielles et humaines. Ainsi, nous avons entrepris le développement d'une plateforme d'e-services destinés aux forces armées béninoises. L'approche adoptée a permis de définir tous les éléments clés du projet, faisant de notre plateforme un espace dédié à la gestion des besoins des agents des forces armées. Pour cela, nous avons utilisé des outils tels que le framework Spring Boot de Java. L'analyse globale de la situation indique que notre solution offre une méthode sécurisée pour que nos utilisateurs formulent et traitent leurs demandes.

Mots clés : DCSCA

Abstract

The dematerialization of administrative processes is of crucial importance to DCSCA, aiming to modernize and optimize payroll processing management. This represents a significant step in DCSCA's commitment to efficiency, transparency, and user satisfaction. This project aligns with a continuous improvement approach to services, providing simplified access, reduced processing times, and increased security of personal data. The success of a project requires access to essential resources, including financial, material, and human resources. Therefore, we embarked on developing an e-services platform for the Beninese armed forces. The adopted approach allowed us to define all key elements of the project, making our platform a dedicated space for managing the needs of armed forces personnel. To achieve this, we used tools such as the Spring Boot framework in Java. The overall analysis of the situation indicates that our solution offers a secure method for users to formulate and process their requests.

Key words: DCSCA

Table des figures

1.1 Capture de la page d'accueil des e-Services de la DGB du Bénin 6

2.1 Diagramme de cas d'utilisation de l'Administré 10
2.2 Diagramme de cas d'utilisation de l'Agent de la DCSCA 11
2.3 Diagramme de Classe 12
2.4 Diagramme de Séquence de Afficher Historique de Paie 13
2.5 Diagramme de Séquence de Traiter un Dossier 14
2.6 Logo draw.io 16
2.7 Logo Adobe illustrator 17
2.8 Logo Eclipse 18
2.9 Logo Gitlab 18
2.10 Logo HTML5 & CSS3 19
2.11 Logo JavaScript 19
2.12 Logo Bootstrap 19
2.13 Logo Java 22
2.14 Logo Spring MVC 22
2.15 Logo PostgreSQL 23

3.1 Accueil 25
3.2 Connexion 26
3.3 Inscription 26
3.4 Profil Administrateur 27
3.5 Profil Relevé de Salaire 27
3.6 Profil Domiciliation des Salaires 28
3.7 Profil Cessation de Paiement 28
3.8 Informations Personnelles 29
3.9 Nature CCP 29
3.10 Ajout Pièces Requises 30
3.11 Enregistrement 30
3.12 Demande en cours 31
3.13 Début du traitement 31
3.14 Traitement 32
3.15 Résultat 32
3.16 Déconnexion 33

Liste des acronymes

CNI :

Carte Nationale d'Identité

CP :

Carte Professionnelle

CSS :

Cascading Style Sheets

DCSCA :

Direction Centrale du Service du Commissariat des Armées

DSF :

Direction des Services Financiers

GKDY-DSF :

Global Kady DSF

HTML :

Hyper Text Markup Language

RIB :

Relevé d'Identité Bancaire

UML :

Unified Modeling Language

WWW :

Word Wide Web

Introduction Générale

Contexte

L'évolution rapide des technologies numériques transforme profondément la manière dont les services publics sont fournis et gérés. Dans un contexte global où l'efficacité et la réactivité sont devenues des impératifs, les administrations publiques cherchent à tirer parti des innovations technologiques pour améliorer leurs prestations. La Direction Centrale des Services Comptables et Administratifs (DCSCA), responsable de la gestion des besoins financiers des forces armées béninoises, n'échappe pas à cette tendance. Face à la croissance du défi lié au traitement des différents besoins financiers des administrés, il devient crucial d'explorer des solutions innovantes qui répondent à ces exigences de manière efficace. Les e-services, ou services électroniques, offrent une opportunité unique d'optimiser les processus financiers en les rendant plus rapides, transparents et accessibles. Dans ce cadre, notre projet se positionne comme une réponse proactive visant à instaurer un système de gestion novateur pour la DCSCA. Nous cherchons à simplifier le traitement des demandes financières, telles que les Certificats de Cessation de Paiement (CCP), les domiciliations et les relevés de solde, par le biais d'une application dédiée. Ce mémoire présente une démarche méthodique qui s'articule autour de plusieurs étapes : une introduction au projet, une revue de la littérature existante, une analyse des outils et de la sécurité, la conception d'un prototype, et enfin, des conclusions significatives.

Contexte

Les besoins des administrés en matière de traitement sont en constante augmentation. Dans le but de surmonter cette difficulté, notre initiative consiste à instaurer un système qui intègre les besoins réels des administrés répartis sur le territoire national, les anticipe afin de travailler à leur satisfaction. Cette solution vise principalement à faciliter le paiement des salaires et à répondre efficacement aux diverses exigences des administrés.

Problématique

Le problème qui se pose est le suivant : **Comment optimiser les services finnaciers de la DCSCA à gtravers des e-services en permettant non seulement le règlement rapide et sécurisé des salaires,**

mais également la prise en charge optimale des divers besoins financiers des agents militaires, tout en tenant compte des enjeux de sécurité et des avancées existantes dans ce domaine ?

Objectifs

Notre objectif principal est d'aider l'armée béninoise dans la gestion des différents besoins de leurs agents.

Objectifs spécifiques

La solution permettra en particulier :

- aux forces armées de consulter et de tirer leurs relevés de solde ;
- aux forces armées de retirer leur certificat de cessation de paiement ;
- aux gestionnaires de comptes de faire la demande de domiciliation des agents des forces armées.

Organisation du mémoire

Le travail en cours se divise en trois (03) chapitres distincts. Dans un premier temps, le chapitre 1 aborde la revue de littératiure . Le chapitre 2, quant à lui, détaille notre proposition en déployant une modélisation UML, en expliquant son fonctionnement, ainsi qu'en exposant les choix techniques inhérents à sa mise en œuvre. Enfin, le chapitre 3 traitera des résultats obtenus dans le cadre de la mise en place de la solution, mettant en évidence les éventuelles insuffisances tout en esquissant les perspectives à venir.

Chapitre 1

Revue de littérature

Pour conduire notre étude de manière approfondie, il est impératif de commencer par définir certaines notions clés avant de procéder à un état des lieux. Cela nous permettra de positionner clairement notre travail par rapport aux solutions existantes poursuivant des objectifs similaires. Dans cette optique, ce chapitre propose d'abord une clarification des concepts essentiels, suivie d'un récapitulatif des éléments existants, se concentrant particulièrement sur une brève présentation du système actuel de gestion des besoins au sein des forces armées.

1.1 Généralités sur les services WEB

Les services web [6] sont des systèmes informatiques conçus pour permettre la communication et l'échange de données entre différentes applications et systèmes informatiques sur un réseau, généralement l'internet. Les plateformes web, quant à elles, sont des environnements logiciels intégrés qui fournissent des services en ligne, des applications et des outils interactifs accessibles via un navigateur web.

- **Qu'est-ce qu'une plateforme WEB ?**
  Une plateforme web [5] est un ensemble de logiciels, d'outils et de services qui permettent aux utilisateurs d'accéder et d'interagir avec des applications via un navigateur web. Elle offre un environnement centralisé pour le développement, le déploiement et la gestion d'applications web.

- **L'intérêt d'une plateforme WEB**
  Les plateformes web [5] offrent de nombreux avantages. Elles permettent un accès universel depuis tout appareil connecté à Internet, favorisant ainsi une accessibilité généralisée. La collaboration est simplifiée grâce au partage d'informations et de ressources entre utilisateurs, indépendamment de leur emplacement géographique. Les mises à jour logicielles sont déployées de manière centralisée, garantissant une uniformité et une mise à jour facile pour tous les utilisateurs. De plus, ces plateformes sont souvent conçues pour être évolutives, s'adaptant ainsi à la croissance des utilisateurs et des besoins.

- **Les caractéristiques d'une plateforme WEB**
  Les caractéristiques des plateformes WEB peuvent varier en fonction de l'objectif, et domaine

d'application et de l'architecture. Cependant, nous pouvons noter un certain nombre de caractéristiques :

- L'**accessibilité en ligne** :
  Une plateforme web est accessible via Internet à partir de n'importe quel appareil connecté, ce qui permet aux utilisateurs d'y accéder depuis n'importe où dans le monde.
- L'**Interface Utilisateur (UI)** :
  Elle propose une interface conviviale et intuitive pour offrir une expérience utilisateur optimale. Cette interface peut s'ajuster dynamiquement aux différentes tailles d'écrans, offrant une expérience cohérente sur des dispositifs variés, tels que les ordinateurs de bureau, les tablettes et les smartphones (réactivité).
- Les **services web** :
  Les plateformes web fournissent souvent des API qui permettent l'intégration avec d'autres systèmes et services. Cela favorise l'interopérabilité et la création d'écosystèmes logiciels interconnectés.
- La **gestion des utilisateurs** :
  Une plateforme web doit offrir des mécanismes d'authentification robustes pour vérifier l'identité des utilisateurs. De plus, la gestion des autorisations garantit que les utilisateurs ont accès uniquement aux fonctionnalités et aux données appropriées.
- La **base de données** :
  La plateforme doit être capable de stocker, organiser et récupérer efficacement les données. Elle peut utiliser différents types de bases de données (relationnelles, NoSQL, etc.) en fonction des besoins spécifiques de l'application. Des mécanismes de sauvegarde réguliers et des plans de récupération des données sont essentiels pour éviter la perte de données en cas de défaillance du système
- L'**interopérabilité** :
  Les plateformes web sont généralement conçues pour être compatibles avec différents systèmes d'exploitation, navigateurs web et dispositifs, assurant ainsi une interopérabilité maximale.
- La **sécurité** :
  La sécurité des données et des utilisateurs est une priorité. Les plateformes web mettent en place des mesures de sécurité pour protéger les informations sensibles et prévenir les menaces en ligne telles que les attaques par injection SQL, les attaques par déni de service, etc.

Ce sont là quelques caractéristiques des plateformes WEB.

1.2 Revue de littérature

1.2.1 Analyse de l'existant

Les forces armées béninoises utilisent actuellement un système manuel qui permet à divers acteurs de recevoir et traiter une gamme d'informations telles que les données personnelles, les décisions de

mise à la retraite, les avis de mutation, les avis d'avancement, les demandes de domiciliation bancaire, les états de prélèvement, les notes de service, les lettres de demande, et plus encore.

1. Le **Directeur des Services Financiers (DSF)** ou le **Directeur Central du Service du Commissariat des Armées (DCSCA)** collectent les informations personnelles des administrés, les décisions de mise à la retraite, et les procès-verbaux du tribunal administratif. Cette collecte vise à fournir des documents tels qu'une attestation de salaire, un relevé de solde, un certificat de cessation de paiement, et à effectuer le changement de numéro de compte. Ils reçoivent également des administrés leurs informations personnelles, une attestation de non-engagement de l'ancienne banque, leur nouveau Relevé d'Identité Bancaire (RIB), une copie de leur Carte Nationale d'Identité ou Carte Professionnelle (CNI/CP), et numérisent les preuves nécessaires pour traiter leurs réclamations.

2. Le **DSF** et le **DCSCA** reçoivent des banques les fichiers de domiciliation en vue de délivrer les attestations de domiciliation ainsi que les courriers liés aux contentieux et autres.

Ce système manuel peut avoir des limites en termes de temps, de précision et d'efficacité. Il pourrait bénéficier d'améliorations ou d'un remplacement par un système informatisé plus moderne pour répondre aux besoins de l'armée béninoise en matière de gestion des ressources humaines et financières.

1.2.2 Insuffisances de l'existant

- **Risques d'erreurs** :
  Un système manuel facilite les erreurs humaines telles que la saisie, le calcul ou le traitement de données. Cela peut entraîner des retards, des incohérences ou des erreurs dans les paiements, les relevés de solde, les certificats de cessation de paiement, etc. Un système automatisé pourrait contribuer à réduire ces risques.
- **Complexité de la gestion des documents** :
  Avec des quantités importantes de documents papier à gérer, il est difficile de les stocker, de les classer, de les rechercher et de les suivre. Cela peut entraîner des pertes de documents, des doublons, des retards dans les traitements, etc.
- **Manque de traçabilité** :
  Un système manuel peut avoir une faible traçabilité des opérations effectuées, ce qui peut rendre difficile la vérification de l'historique des transactions, la résolution des litiges ou des erreurs, et l'audit de la gestion des ressources financières et humaines.
- **Manque de sécurité** :
  Il est plus difficile de garantir la sécurité des informations personnelles, financières et sensibles des administrés, notamment en cas de perte, de vol ou d'utilisation frauduleuse des documents ou des données.
- **Difficulté de mise à jour** :
  Il est plus difficile de mettre à jour les informations et les données relatives aux administrés, ce qui peut entraîner des retards dans les traitements, des incohérences ou des erreurs dans les opérations de gestion des ressources humaines et financières.

En somme, un système manuel peut présenter des limites importantes en termes de fiabilité, d'efficacité et de sécurité, ce qui peut avoir des conséquences négatives sur la qualité de la gestion des ressources humaines et financières dans une organisation.

1.3 Solutions existantes : e-Services de la DGB du Bénin

La plateforme d'e-services de la Direction Générale du Budget (DGB) du Bénin est un outil numérique conçu pour moderniser et optimiser les services administratifs et financiers destinés aux fonctionnaires de l'État, aux pensionnés du Fonds National des Retraites du Bénin (FNRB) et à leurs ayants droit.

1.3.1 Page d'accueil



FIGURE 1.1 : Capture de la page d'accueil des e-Services de la DGB du Bénin

1.3.2 Points clés de la plateforme

La plateforme met l'accent sur la dématérialisation des services financiers et administratifs, visant à améliorer l'efficacité, la transparence et la satisfaction des usagers en leur permettant d'accéder à divers services en ligne.

- **Services financiers en ligne** :
  - **Certificat de vie pour les pensions** : Les pensionnés peuvent obtenir leurs certificats de vie en ligne.
  - **Relevé de pension** : Les retraités peuvent consulter et imprimer leurs relevés de pension.
  - **Relevé de salaire** : Les fonctionnaires peuvent accéder à leurs bulletins de salaire.
  - **Domiciliation bancaire** : Les agents peuvent effectuer des demandes de domiciliation bancaire.
  - **Demande de capital décès** : Les ayants droit peuvent soumettre des demandes de capital décès en ligne.
- **Transparence et suivi** : La DGB encourage la transparence en permettant aux utilisateurs de suivre l'état d'avancement de leurs demandes. Des mises à jour régulières sont fournies pour assurer l'engagement et la satisfaction des utilisateurs.
- **Accompagnement et conseil** : La plateforme offre un accompagnement tout au long du processus. Des outils et des ressources sont disponibles pour aider à la soumission des demandes et à la résolution des problèmes éventuels.

1.3.3 Limites des e-Services de la DGB

Les e-Services de la Direction Générale du Budget (DGB) offrent de nombreux avantages, mais ils présentent également quelques limites que les utilisateurs pourraient remarquer :

- **Temps de réponse :** En fonction de la charge du système et de la qualité de la connexion Internet des utilisateurs, les temps de réponse peuvent varier, affectant l'expérience utilisateur en termes de rapidité et de fluidité de la navigation.
- **Restrictions géographiques :** Certains services ou fonctionnalités peuvent être limités en fonction de la région géographique de l'utilisateur ou de l'accès à Internet, restreignant ainsi l'accès complet à toutes les fonctionnalités pour tous les utilisateurs du Bénin.
- **Absence d'envoi d'e-mails pour notifications :** Actuellement, il manque un système d'envoi d'e-mails pour notifier automatiquement les bénéficiaires des mises à jour ou des états de leurs demandes, ce qui pourrait améliorer la communication et l'engagement des utilisateurs.

1.4 Définition des caractéristiques de notre plateforme

Pour pallier aux insuffisances identifiées dans le système actuel utilisé par les forces armées béninoises et dans la plateforme d'e-services de la DGB, l'implémentation d'une plateforme web moderne est envisagée, dotée de caractéristiques spécifiques visant à résoudre les limitations suivantes :

- La minimisation des risques d'erreurs
- La simplification de la gestion des documents
- L'amélioration de la traçabilité des opérations
- Le renforcement de la sécurité des données
- La facilitation de la mise à jour des informations

Ces points permettent de définir globalement les finalités de la plateforme.

1. **La minimisation des risques d'erreurs** :
   La plateforme inclura des fonctionnalités automatisées de vérification des données pour réduire les erreurs humaines lors de la saisie, du calcul et du traitement des informations.
2. **La simplification de la gestion des documents** :
   L'adoption d'un système électronique permettra de faciliter le stockage, la classification, la recherche et le suivi des documents, minimisant ainsi les pertes, les doublons et les retards dans les traitements.
3. **L'amélioration de la traçabilité des opérations** :
   La plateforme enregistrera de manière détaillée chaque opération effectuée, assurant ainsi une traçabilité complète des transactions pour faciliter la vérification, la résolution des litiges et les audits.

4. **Le renforcement de la sécurité des données** :
   Mise en place de protocoles de sécurité robustes pour garantir la confidentialité et l'intégrité des informations personnelles, financières et sensibles des administrés, ainsi que des mécanismes de contrôle d'accès pour prévenir les utilisations non autorisées.

5. **La facilitation de la mise à jour des informations** :
   La plateforme offrira une interface conviviale permettant une mise à jour facile et rapide des informations relatives aux administrés, évitant ainsi les retards, les incohérences et les erreurs dans les opérations de gestion des ressources humaines et financières.

Conclusion

Il existe un système manuel pour la gestion des besoins liés à la finance des forces armées présentant quelques insuffisances . Dans le but de palier à ces insuffisances , nous avons convenu de réalisation une plateforme web de pour la gestion des différents besoins des forces armées béninoises. Le prochain chapitre sera dédié à la présentation de notre solution, mettant en lumière sa conception ainsi que les outils employés dans sa mise en œuvre .

Chapitre 2

Solutions, choix techniques et implémentation

Introduction

La création d'une plateforme requiert l'organisation et la structuration des concepts associés à sa conception et à son développement. Ce chapitre expose la solution envisagée, détaillant les différentes étapes de sa modélisation. Il offre également un aperçu des outils et technologies employés dans la concrétisation de notre solution, baptisée Global Kady dsf **(GKDY-DSF)**.

2.1 Description de la solution

GKDY-DSF est la plateforme répondant à la problématique de la gestion rapide des différents besoins des agents des forces armées béninoises. A travers une plateforme WEB, GKDY-DSF permet aux administrés de formuler différentes demandes . Il permet aussi daux différentes structures des finances de répondre aux demandes des administrés. Afin d'assurer une expérience utilisateur optimale, des directives sont fournies pour rendre l'utilisation de la plateforme intuitive.

2.2 Modélisation

Pour représenter les fonctionnalités de GKDY-DSF, nous avons opté pour l'utilisation d'UML, qui est l'abréviation anglaise de "Unified Modeling Language" ou en français, "Langage de Modélisation Unifié". UML [8] est langage de modélisation graphique utilisé dans le domaine du génie logiciel pour visualiser, spécifier, concevoir et documenter les systèmes logiciels. UML offre un ensemble de notations graphiques standardisées qui facilitent la communication entre les membres d'une équipe de développement logiciel et permettent de représenter visuellement la structure, le comportement, l'interaction et les fonctionnalités d'un système.

2.2.1 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation [8] fait mention du comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Il scinde les fonctionnalités du système en unités cohérentes appelées cas d'utilisation, ayant un sens pour les acteurs.

1. **Diagramme de cas d'utilisation de l'Administré** (voir figure 2.1)

   - **Formuler une demande** : L'administré peut soumettre une nouvelle demande au système.
   - **S'authentifier** : Avant d'accéder à toute autre fonctionnalité, l'administré doit s'authentifier.
   - **Suivre le statut de sa demande** : Après avoir soumis une demande, l'administré peut suivre son statut.
     - **Consulter et télécharger** : Cette fonctionnalité permet à l'administré de voir et de télécharger les documents relatifs à sa demande.
   - **Afficher l'historique de paie** : L'administré peut voir l'historique de ses paiements.
     - **Rechercher par période** : L'administré peut effectuer une recherche d'historique de paie par période spécifique.
     - **Rechercher par échéance** : L'administré peut également rechercher par échéance.
     - **Imprimer leur fiche de paie** : Cette fonctionnalité permet d'imprimer les fiches de paie.

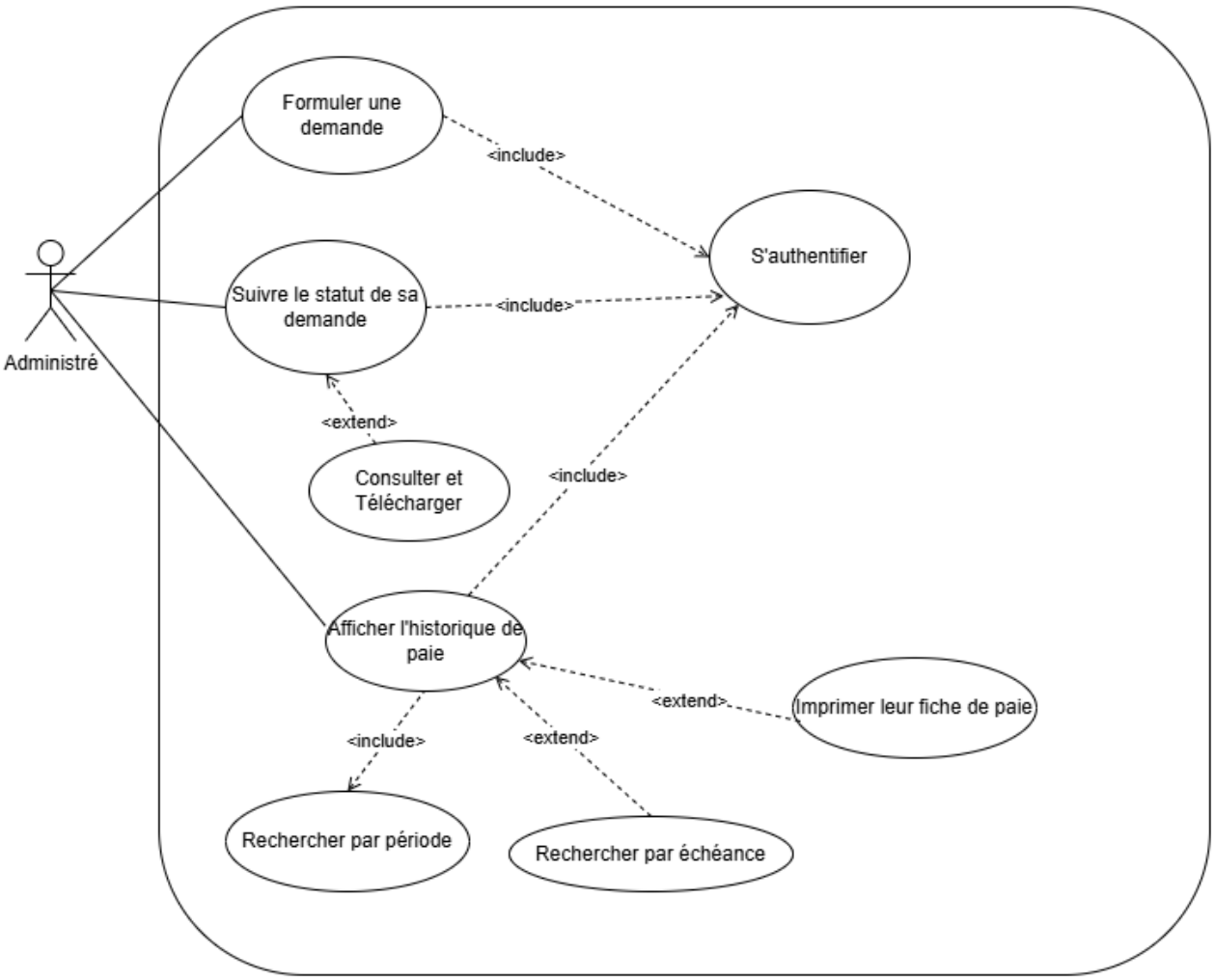


FIGURE 2.1 : Diagramme de cas d'utilisation de l'Administré

2. **Diagramme de cas d'utilisation de l'Agent de la DCSCA** (voir figure 2.2)

   - **Rechercher un dossier** : L'agent peut rechercher des dossiers spécifiques dans le système.
   - **S'authentifier** : Avant d'accéder à toute autre fonctionnalité, l'agent doit s'authentifier.
   - **Consulter et éditer un dossier** : L'agent peut visualiser et modifier les informations contenues dans un dossier.
   - **Traiter le dossier** : L'agent a plusieurs options pour traiter un dossier :
     - **Rejeter le dossier** : L'agent peut rejeter un dossier si nécessaire.
     - **Retourner le dossier pour correction** : Si des modifications sont nécessaires, l'agent peut renvoyer le dossier pour correction.
     - **Valider le dossier** : L'agent peut approuver le dossier une fois les conditions remplies.
       * **Envoyer la domiciliation signée au bénéficiaire** : Après validation, la domiciliation signée est envoyée au bénéficiaire.

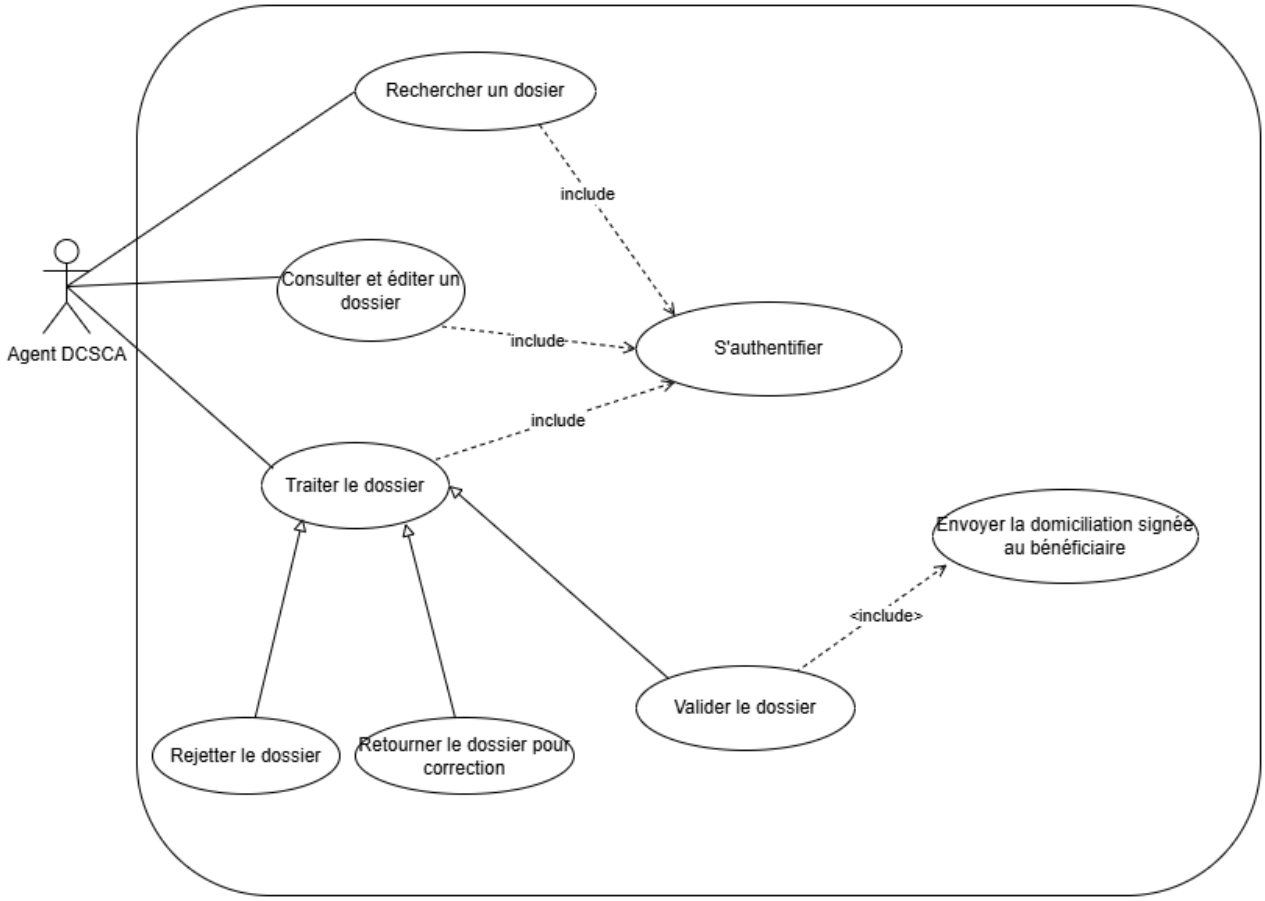


FIGURE 2.2 : Diagramme de cas d'utilisation de l'Agent de la DCSCA

2.2.2 Diagramme de classe

Le diagramme de classes [8] a pour objectif de décrire la structure des entités manipulées par les utilisateurs. Il s'agit d'une vue statique, car on ne tient pas compte du facteur temporel dans le comportement du système. Le diagramme de classes modélise les concepts du domaine d'application ainsi que les concepts internes créés de toutes pièces dans le cadre de l'implémentation d'une application.

1. Diagramme de Classe

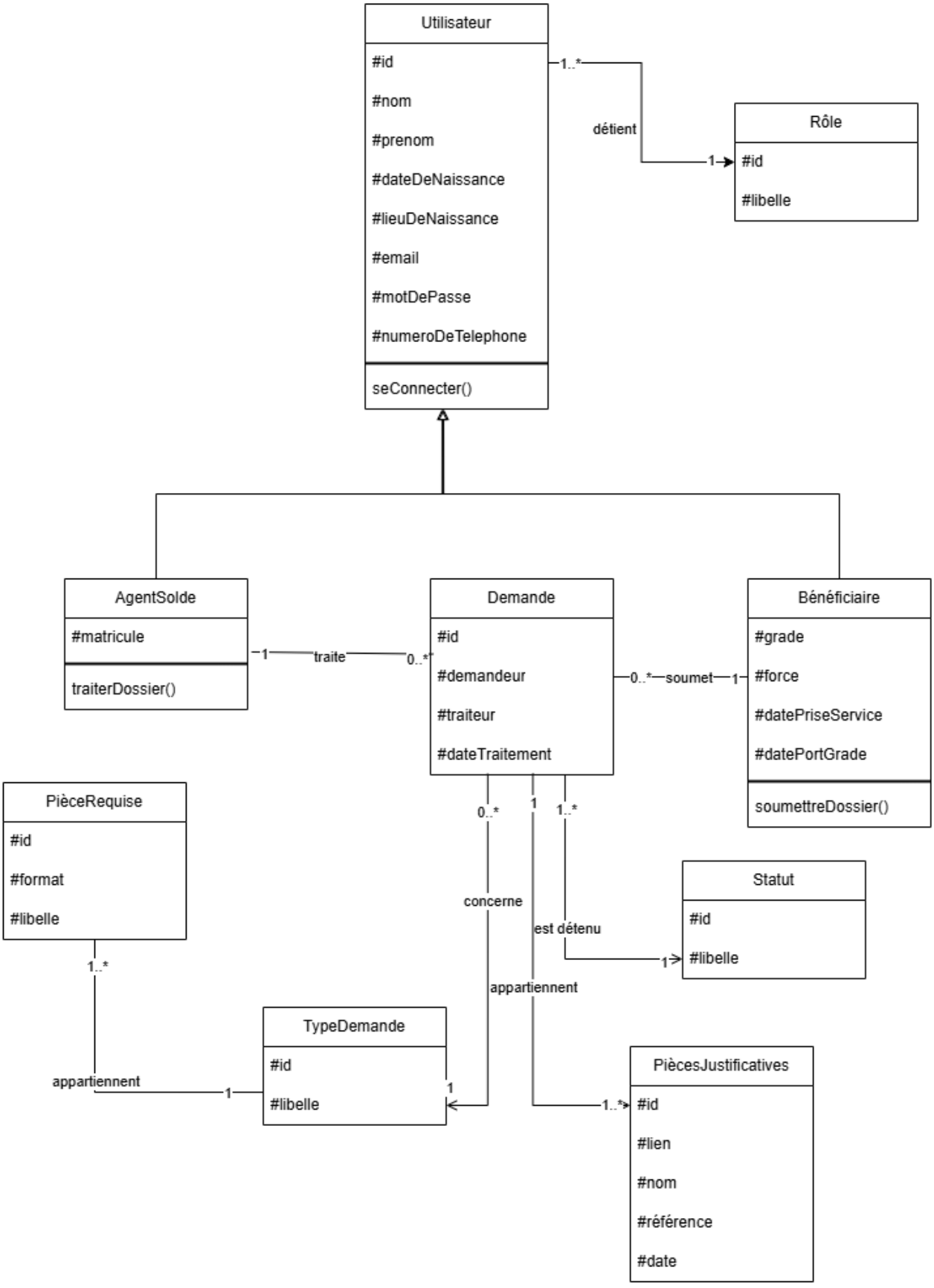


FIGURE 2.3 : Diagramme de Classe

2.2.3 Diagramme de séquences

Le diagramme de séquences [8] représente visuellement l'interaction entre différents objets ou composants dans un système logiciel. Il met en évidence la séquence chronologique des messages échangés entre les objets pendant l'exécution d'un scénario ou d'un cas d'utilisation spécifique. Les diagrammes de séquence sont particulièrement utiles pour comprendre la manière dont les objets interagissent dans le temps et pour modéliser le flux des opérations au sein d'une application ou d'un système.

1. **Diagramme de Séquence "Afficher Historique de Paie"**

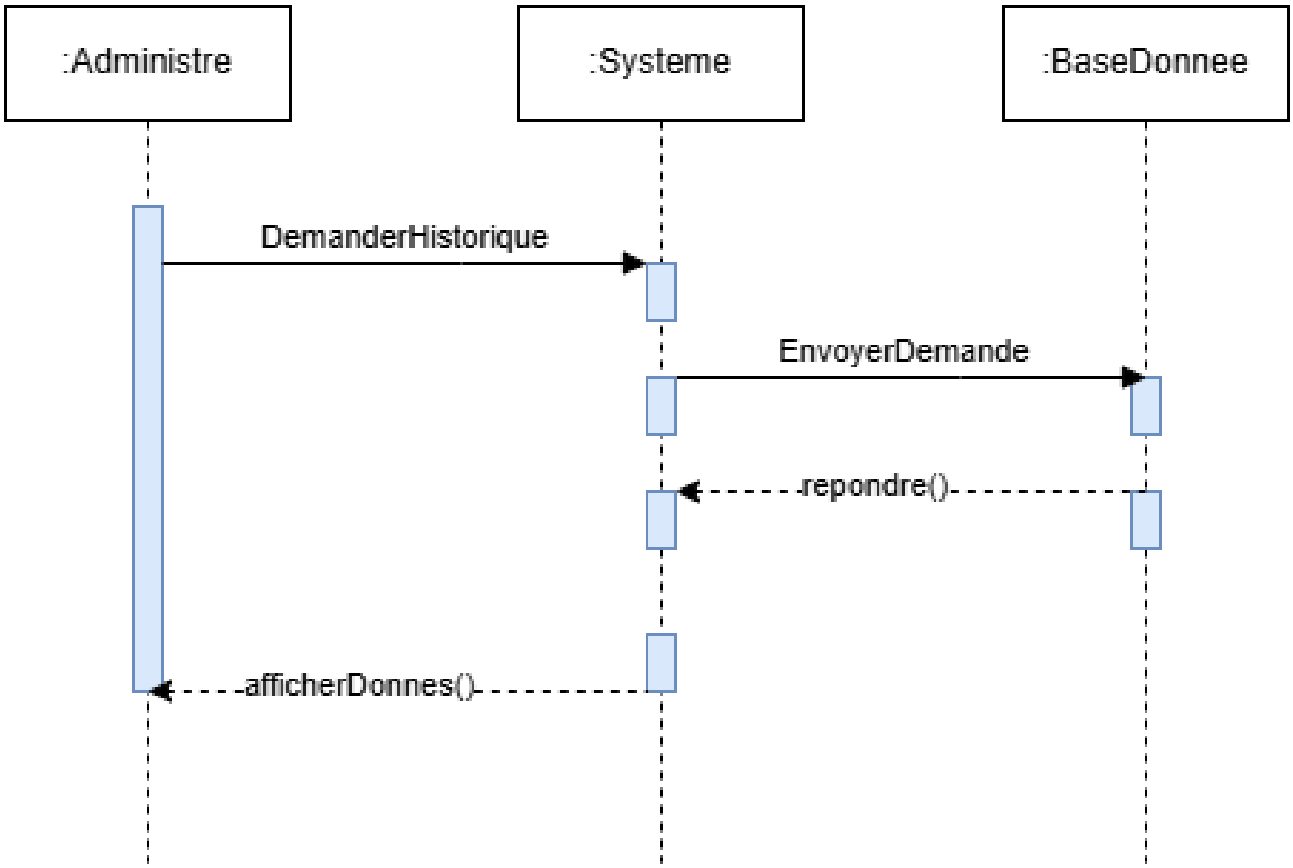


FIGURE 2.4 : Diagramme de Séquence de Afficher Historique de Paie

2. **Diagramme de Séquence "Traiter un Dossier"**

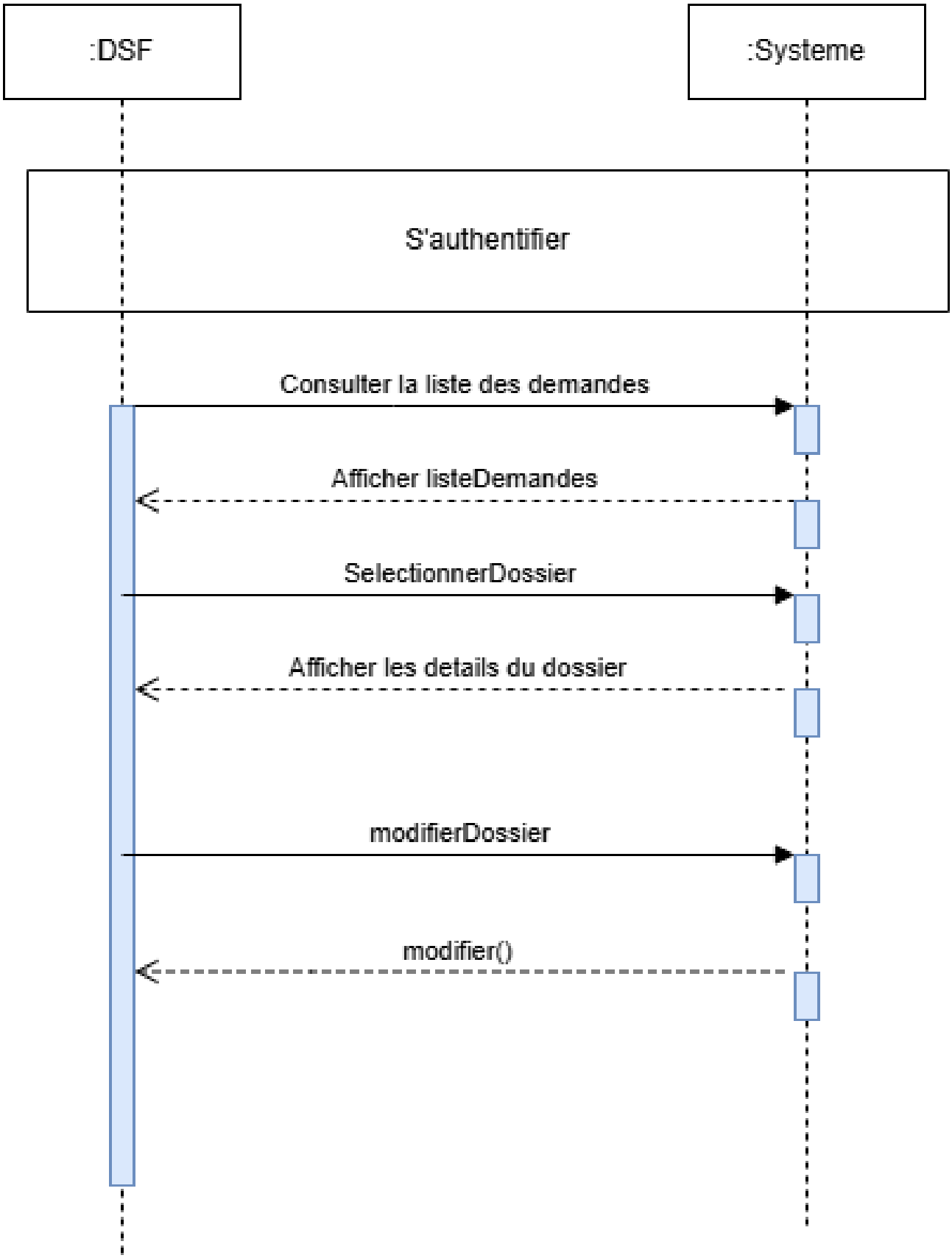


FIGURE 2.5 : Diagramme de Séquence de Traiter un Dossier

2.3 Choix techniques

2.3.1 Méthodologie de développement

Nous avons opté pour la méthodologie Agile pour notre plateforme d'e-services de la DCSCA en raison de ses nombreux avantages. Agile [4] permet une grande flexibilité et une adaptabilité rapide aux changements, ce qui est essentiel dans un environnement en constante évolution. Elle favorise une collaboration étroite entre les membres de l'équipe et les parties prenantes, assurant ainsi que le produit final répond aux besoins des utilisateurs. De plus, Agile utilise des cycles de développement courts appelés sprints, ce qui permet une livraison rapide et régulière de fonctionnalités utilisables et améliore ainsi la qualité et la satisfaction des utilisateurs.
En outre, l'approche Agile [4] encourage les retours rapides et fréquents, permettant de corriger les erreurs précocement et de s'adapter aux nouvelles exigences tout au long du projet. Cette méthodologie réduit également les risques en maintenant une attention constante sur la qualité du produit par des tests continus et une amélioration progressive.

2.3.2 Outils utilisés

La réalisation de notre solution a nécessité l'utilisation d'un certain nombre d'outils. Il s'agit entre autres de logiciels nous ayant permis d'avancer progressivement dans le projet. Nous pouvons les énumérer en donnant ce à quoi ils ont servis. Nous avons :

- **Outils de création de diagrammes**[7] :

Outil	Avantages	Limites
draw.io	Gratuit, facile à utiliser, intégration avec les services cloud	Fonctionnalités limitées par rapport à des outils payants
Microsoft Visio	Puissant, riche en fonctionnalités, support d'entreprise	Coût élevé, courbe d'apprentissage
Lucidchart	Collaboration en ligne, nombreuses intégrations, interface utilisateur intuitive	Abonnement payant pour les fonctionnalités avancées
Gliffy	Simple d'utilisation, intégration avec Atlassian, version cloud	Moins de fonctionnalités que Visio ou Lucidchart

TABLE 2.1 : Outils de création de diagrammes

draw.io est adopté pour sa gratuité, sa simplicité d'utilisation, et ses nombreuses intégrations avec des services de stockage cloud comme Google Drive et Dropbox. De plus, étant open source, il offre une flexibilité et une transparence appréciables.



FIGURE 2.6 : Logo draw.io

- **Outils de création de logos**[7] :

Outil	**Avantages**	**Limites**
Adobe Illustrator 2020	Outil professionnel, riche en fonctionnalités, précision vectorielle, nombreux tutoriels disponibles	Coût élevé, courbe d'apprentissage
CorelDRAW	Interface intuitive, bonnes fonctionnalités de design vectoriel, moins cher qu'Illustrator	Moins utilisé dans l'industrie, compatibilité limitée avec certains formats
Inkscape	Open source, gratuit, bon pour les graphiques vectoriels de base	Moins de fonctionnalités avancées, interface moins polie
Canva	Facile à utiliser, nombreuses templates, options gratuites disponibles	Moins de contrôle sur les détails, limité pour les créations complexes
Gravit Designer	Basé sur le web, interface intuitive, gratuit pour les fonctionnalités de base	Moins puissant que les outils de bureau, fonctionnalités avancées payantes

TABLE 2.2 : Outils de création de logos

Nous avons choisi **Adobe Illustrator 2020** pour sa richesse en fonctionnalités, sa précision vectorielle et son large éventail d'outils professionnels. Bien qu'il soit coûteux et possède une courbe d'apprentissage, il est largement utilisé dans l'industrie du design graphique, ce qui en fait un choix idéal pour les professionnels cherchant à créer des logos de haute qualité avec une grande précision et flexibilité.

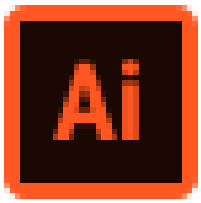


FIGURE 2.7 : Logo Adobe illustrator

- **Environnements de développement**[7] :

Environnement	Avantages	Limites
Eclipse	Gratuit et open source, supporte de nombreux langages, vaste écosystème de plugins, large communauté	Peut être lent et gourmand en ressources, interface parfois complexe
IntelliJ IDEA	Interface utilisateur intuitive, support excellent pour Java, nombreuses fonctionnalités avancées	Version complète payante, peut être gourmand en ressources
Visual Studio	Support étendu pour les langages Microsoft, outils de débogage puissants, intégration avec Azure	Coût élevé pour la version complète, très gourmand en ressources
NetBeans	Gratuit et open source, bon support pour Java et PHP, intégration avec plusieurs serveurs d'application	Moins de plugins que Eclipse, interface moins moderne
PyCharm	Excellent support pour Python, nombreuses fonctionnalités pour le développement web, outils de débogage puissants	Version complète payante, peut être gourmand en ressources

TABLE 2.3 : Environnements de développement (IDE)

Nous avons opté pour **Eclipse**[1] pour sa gratuité et son caractère open source, son support étendu pour de nombreux langages de programmation, et son vaste écosystème de plugins qui permet de l'étendre et de l'adapter à différents besoins. Bien qu'il puisse être lent et gourmand en ressources, sa large communauté et son support continu en font un choix robuste pour les développeurs, en particulier ceux qui travaillent avec Java.



FIGURE 2.8 : Logo Eclipse

- **Plateformes de développement collaboratif**[7] :

Plateforme	**Avantages**	**Limites**
GitLab	Intégration CI/CD, open source, interface utilisateur intuitive	Peut être complexe à installer et configurer pour les grandes équipes
GitHub	Large communauté, nombreuses intégrations, interface intuitive	Fonctionnalités CI/CD limitées par rapport à GitLab
Bitbucket	Intégration avec Atlassian, support CI/CD, prix compétitifs	Moins populaire, moins de plugins que GitHub
Azure DevOps	Outils complets pour DevOps, bonne intégration avec les services Microsoft	Peut être coûteux, courbe d'apprentissage

TABLE 2.4 : Plateformes de développement collaboratif

Nous avons choisi **GitLab** pour son intégration CI/CD, sa nature open source et son interface utilisateur intuitive. Même si son installation et sa configuration peuvent être complexes pour les grandes équipes, ses fonctionnalités complètes et son support open source en font une plateforme précieuse pour le développement collaboratif.



FIGURE 2.9 : Logo Gitlab

Ces technologies nous ont aider à mettre notre plateforme en place. Cette mise en place a été possible au moyen d'un certain nombre de technologies.

2.3.3 Technologies utilisées

La mise en œuvre de notre solution a exigé l'utilisation de diverses technologies, classées en deux catégories distinctes : le Front-end et le Back-end.

2.3.3.1 Front-end

Le Front-end [3] fait référence à la partie d'une application informatique ou d'un site web avec laquelle l'utilisateur interagit directement. Le Front-end englobe tous les éléments visibles et interactifs d'une interface utilisateur, notamment la disposition, le design, les menus déroulants, les formulaires, les boutons, et tout autre élément graphique ou fonctionnel avec lequel l'utilisateur peut interagir.
Nous avons examiné plusieurs langages et frameworks frontend (voir Tableau 2.5). Nous avons choisi les technologies suivantes en fonction de leurs avantages respectifs :

- **HTML (HyperText Markup Language)** : pour sa structure sémantique claire et son statut de standard du web et **CSS (Cascading Style Sheets)** : pour le contrôle précis de l'apparence des éléments HTML, y compris les animations et transitions.



FIGURE 2.10 : Logo HTML5 & CSS3

- **JavaScript** : pour ajouter de l'interactivité aux pages web et créer des applications réactives.



FIGURE 2.11 : Logo JavaScript

- **Bootstrap** : pour ses composants préconstruits et son design responsive, facilitant le développement frontend.



FIGURE 2.12 : Logo Bootstrap

- **Langages & Frameworks**[7] :

Langage/Framework	Avantages	Limites
HTML	Structure sémantique claire, standard du web, facile à apprendre	Pas de fonctionnalités de mise en forme avancées, dépend des CSS pour le style
CSS	Contrôle détaillé de l'apparence, supporte les animations, compatible avec tous les navigateurs	Peut devenir complexe à gérer pour les grands projets, nécessite de bonnes pratiques pour la maintenabilité
JavaScript	Langage de programmation dynamique, interactivité, large écosystème de bibliothèques et frameworks	Peut être difficile à déboguer, problèmes de compatibilité entre navigateurs
Bootstrap	Composants préconstruits, responsive design, accélère le développement	Personnalisation limitée sans réécriture des styles, sites peuvent se ressembler
TypeScript	Superset de JavaScript, typage statique, améliore la maintenabilité et la détection des erreurs	Nécessite une étape de compilation, courbe d'apprentissage pour les débutants
Angular	Framework complet, facilite le développement de SPA, supporté par Google	Courbe d'apprentissage, complexité pour les petits projets
React	Bibliothèque flexible, composant réutilisables, supportée par Facebook	Nécessite souvent des bibliothèques supplémentaires, courbe d'apprentissage
Vue.js	Facile à intégrer, flexible, courbe d'apprentissage moins raide	Moins de support d'entreprise que React ou Angular, communauté plus petite

TABLE 2.5 : Langages et frameworks frontend

2.3.3.2 Back-end

Le Back-end [3] désigne la partie invisible d'une application ou d'un site web. Il est responsable de la gestion des données, de la logique métier, des connexions avec la base de données, et de la gestion des requêtes des utilisateurs. Les technologies utilisées à cet effet sont :

- **Langages & Frameworks**[7] :

Langage/Framework	**Avantages**	**Limites**
Java	Haute performance, portabilité, sécurité, grande communauté et support	Syntaxe verbeuse, courbe d'apprentissage plus raide
Spring MVC (Java)	Flexibilité, modularité, supporte les architectures RESTful, large écosystème	Configuration complexe, nécessite une bonne compréhension de Java
Node.js (JavaScript)	I/O non-bloquant, haute performance, grand écosystème de packages (npm)	Pas adapté pour les tâches CPU-intensives, dépendance sur le moteur V8
Express (Node.js)	Léger, rapide, flexible, middleware extensible	Nécessite des connaissances en JavaScript, moins structuré que d'autres frameworks
Python	Syntaxe simple et claire, nombreuses bibliothèques, bon pour le prototypage rapide	Moins performant que Java ou C++, gestion des threads limitée
Django (Python)	Complet, batteries incluses, suit les meilleures pratiques de sécurité	Peut être surdimensionné pour les petits projets, courbe d'apprentissage
Ruby	Syntaxe élégante, orienté objet, communauté active	Performance moins élevée, moins de bibliothèques que Java ou Python
Ruby on Rails (Ruby)	Productivité élevée, convention over configuration, intégration facile avec les bases de données	Moins performant, courbe d'apprentissage pour les conventions
PHP	Large support pour les CMS, facile à déployer, large communauté	Incohérence dans la conception, sécurité dépend des pratiques de codage
Laravel (PHP)	Syntaxe expressive, nombreuses fonctionnalités intégrées, support de la communauté	Moins performant que d'autres frameworks, courbe d'apprentissage
C	Intégration avec l'écosystème Microsoft, haute performance, riche en fonctionnalités	Dépendance à l'écosystème Microsoft, moins populaire en open source
ASP.NET (C)	Haute performance, sécurisé, bon support pour les applications de grande envergure	Complexité, dépendance à l'écosystème Microsoft

TABLE 2.6 : Langages et frameworks backend

Nous avons opté pour **Java** [2] pour sa haute performance, sa portabilité (write once, run anywhere), et sa sécurité. Il possède une grande communauté et un support continu, ce qui assure sa pérennité et l'accès à de nombreuses ressources. Bien que sa syntaxe puisse être verbeuse et sa courbe d'apprentissage plus raide, ses avantages en termes de performance et de robustesse en font un choix privilégié pour les applications backend.



FIGURE 2.13 : Logo Java

Nous avons choisi **Spring MVC** parce qu'il offre une grande flexibilité et modularité, supporte les architectures RESTful, et dispose d'un large écosystème. Bien que sa configuration puisse être complexe et qu'il nécessite une bonne compréhension de Java, il permet de construire des applications backend robustes et évolutives en suivant les meilleures pratiques.



FIGURE 2.14 : Logo Spring MVC

- **Systèmes de gestion de base de données**[7] :

SGBD	**Avantages**	**Limites**
PostgreSQL	Open source, riche en fonctionnalités, extensible	Peut être complexe à configurer et optimiser
MySQL	Open source, largement utilisé, facile à installer	Moins de fonctionnalités avancées par rapport à PostgreSQL
SQLite	Légèreté, facile à intégrer, zéro configuration	Limité aux petites bases de données, pas de support client-serveur
Oracle Database	Très performant, riche en fonctionnalités, support complet	Coût élevé, complexe à administrer

TABLE 2.7 : Systèmes de gestion de base de données

Nous avons choisi **PostgreSQL**, pour sa richesse en fonctionnalités et son extensibilité. Bien que sa configuration puisse être complexe, ses capacités avancées et sa robustesse en font un

excellent choix pour des projets nécessitant une base de données puissante et fiable.



FIGURE 2.15 : Logo PostgreSQL

2.3.4 Stratégie de sécurisation

La sécurité de l'application est assurée grâce à l'utilisation du framework Spring MVC. Spring MVC propose plusieurs fonctionnalités intégrées pour garantir la sécurité de l'application, notamment :

- **Des mises à jour régulières** :
  Spring MVC encourage l'utilisation de versions récentes pour bénéficier des correctifs de sécurité. Le maintien à jour de la dernière version de Spring MVC assure l'intégration des correctifs de sécurité les plus récents.

- **De la protection contre les injections SQL et CRSF(Cross-Site Request Forgery** :
  Spring MVC intègre étroitement Spring Security, qui est un module de sécurité complet pour les applications Java. Spring Security offre une gestion flexible de l'authentification, de l'autorisation, et de la protection contre les attaques courantes comme les attaques CSRF (Cross-Site Request Forgery) et les injections SQL.

- **De la gestion des sessions** :
  Spring MVC gère la gestion des sessions, permettant de stocker les informations d'authentification de manière sécurisée. Il offre également des mécanismes pour contrôler la durée de vie des sessions.

- **De la sécurité des fichiers statiques et médias** :
  En utilisant les annotations de sécurité, Spring MVC permet de restreindre l'accès aux fichiers statiques en fonction des rôles définis pour les utilisateurs. Par exemple, l'annotation @PreAuthorize peut être utilisée pour spécifier des conditions d'accès basées sur les rôles.

- **Des mots de passe** :
  Spring MVC propose des utilitaires pour la gestion sécurisée des mots de passe, des fonctions de hachage, et d'autres opérations cryptographiques.

2.3.5 Tests et Validation

Avant d'entrer dans les détails des résultats des tests, il convient de souligner l'importance d'un plan de test solide dans tout projet de développement logiciel. Pour la Plateforme d'E-Services de la DCSCA, le plan de test a été minutieusement élaboré en collaboration avec la division d'exploitation de la direction informatique du ministère des Finances. Ce plan comprenait des tests unitaires, des tests d'intégration, des tests de système, des tests de régression, ainsi que d'autres tests spécifiques nécessaires pour assurer la qualité et la fiabilité du logiciel.

Conclusion

Dans ce chapitre, nous avons exposé en détail notre solution, en abordant sa modélisation, son fonctionnement, ainsi que les choix techniques impliqués dans sa mise en œuvre. Le prochain chapitre se concentrera sur la présentation des résultats obtenus, une analyse critique de ces derniers, et enfin, les perspectives envisagées.

Chapitre 3

Résultats et analyse critique

Introduction

Suite à la conception et à la mise en œuvre de GKDY-DSF, il devient essentiel de mettre en avant les résultats obtenus pour procéder à une analyse critique. Ce chapitre propose ainsi une exposition détaillée des résultats issus des tests, une présentation globale de l'application, et enfin une discussion centrée sur les lacunes identifiées au cours de l'étude.

3.1 Présentation du résultat

Nous présentons ici quelques fonctions de GKDY-DSF. Il s'agit de la création de projets, de la validation de projet et de la contribution aux projets qui sont les fonctions principales de la plateforme.

3.1.1 Accueil

Lorsqu'un utilisateur accède à notre plateforme, il a la possibilité de consulter les informations du site et explorer nos projets sans nécessité de se connecter. Ainsi, à son arrivée sur la page d'accueil, il sera confronté à l'interface suivante :



FIGURE 3.1 : Accueil

3.1.2 Connexion

L'utilisateur est encouragé à se connecter s'il dispose déjà d'un compte. Dans le cas contraire, il est incité à créer un compte. Pour la connexion, il doit fournir son nom d'utilisateur et son mot de passe.

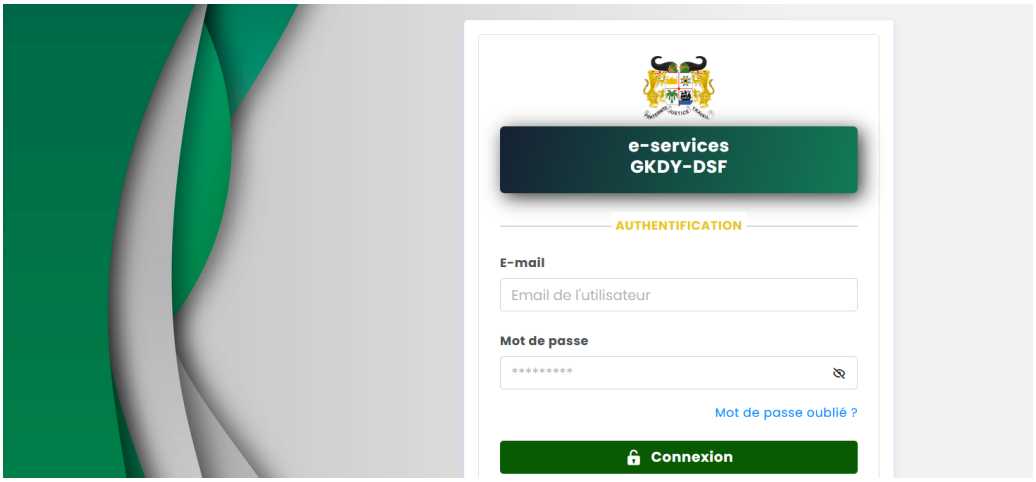


FIGURE 3.2 : Connexion

3.1.3 Inscription

Elle se fait sur 3 étapes. L'utilisateur devra, dans un premier temps donner son rôle et son matricule puis, dans un second temps, son nom de famille et sa date de prise de service et enfin renseigner ses identifiants .

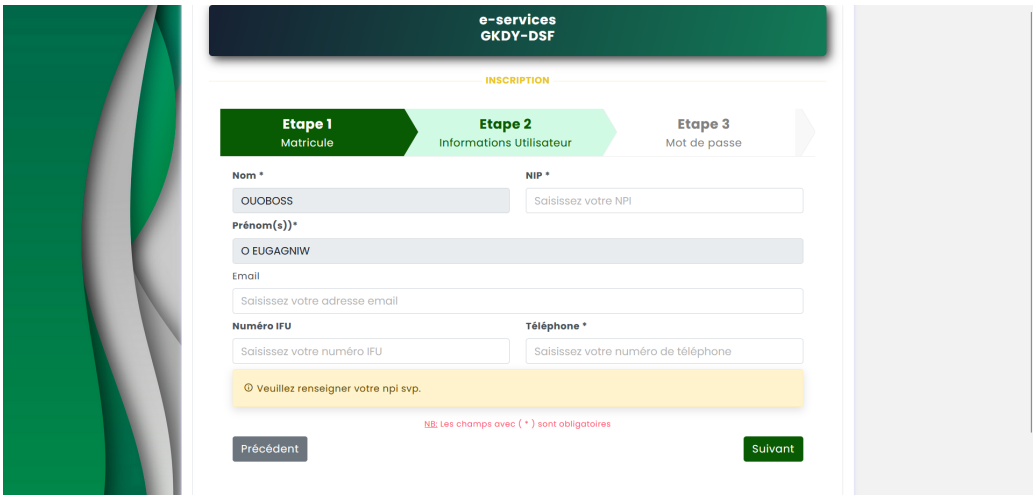


FIGURE 3.3 : Inscription

3.1.4 Profil Administrateur

Une fois connecté, l'administrateur dispose d'un ensemble d'outils et de fonctionnalités lui permettant d'assurer la gestion efficace du système et des utilisateurs : consulter le menu des agents, gérer les utilisateurs, attribuer des profils, gérer les plaintes etc...

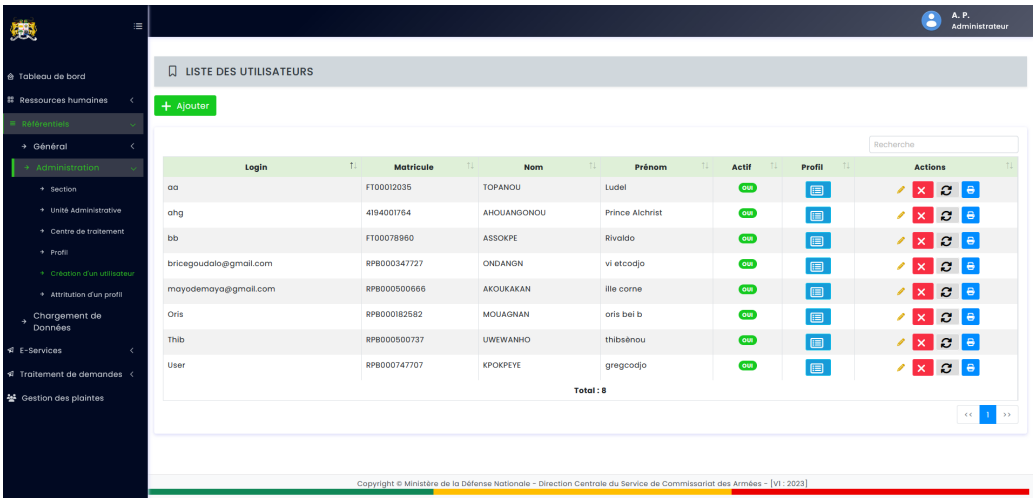


FIGURE 3.4 : Profil Administrateur

3.1.5 Profil Administre

Une fois connecté, l'administré a la possibilité d'entreprendre les actions qui lui sont attribuées. Il peut accéder à son espace, accéder aux différents menus(relevé de salaire, domiciliation, cessation de paiement) pour formuler les demandes qu'il souhaite :

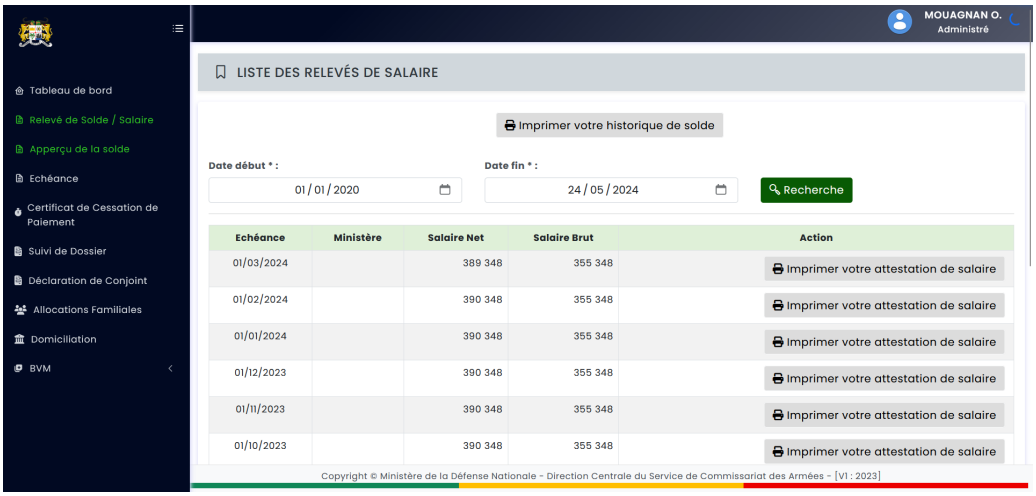


FIGURE 3.5 : Profil Relevé de Salaire

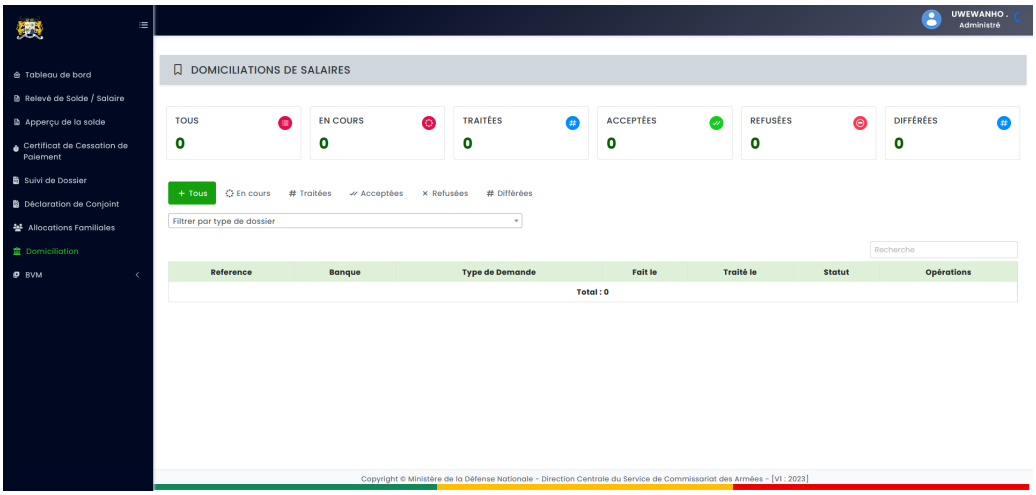


FIGURE 3.6 : Profil Domiciliation des Salaires

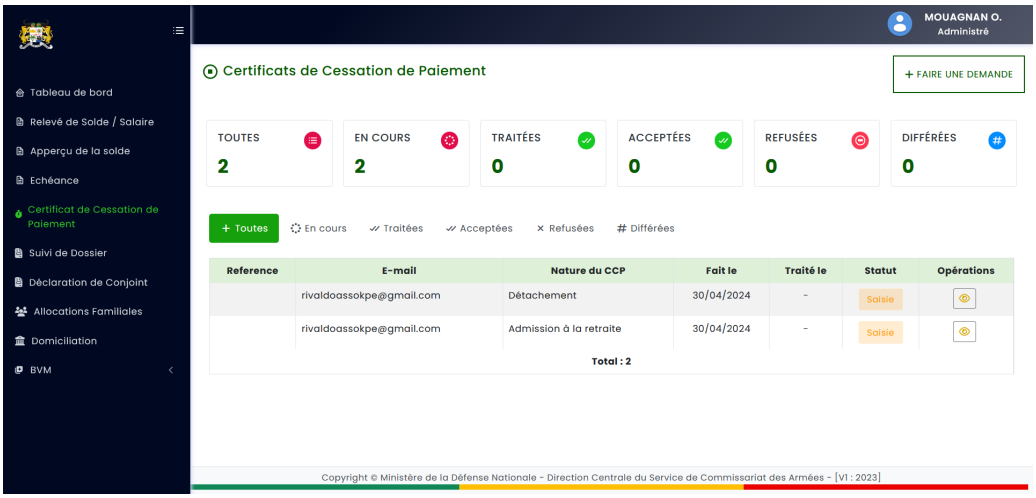


FIGURE 3.7 : Profil Cessation de Paiement

3.1.6 Formulation de Demandes

La formulation de demandes se fait au niveau des trois (03) services. Ici on abordera principalement la demande du certificat de cessation de paiement qui se déroule en quatre(04) étapes

1. **Le renseignement de ses informations** :
   Ici, l'dministré renseigne ses informations personnellles

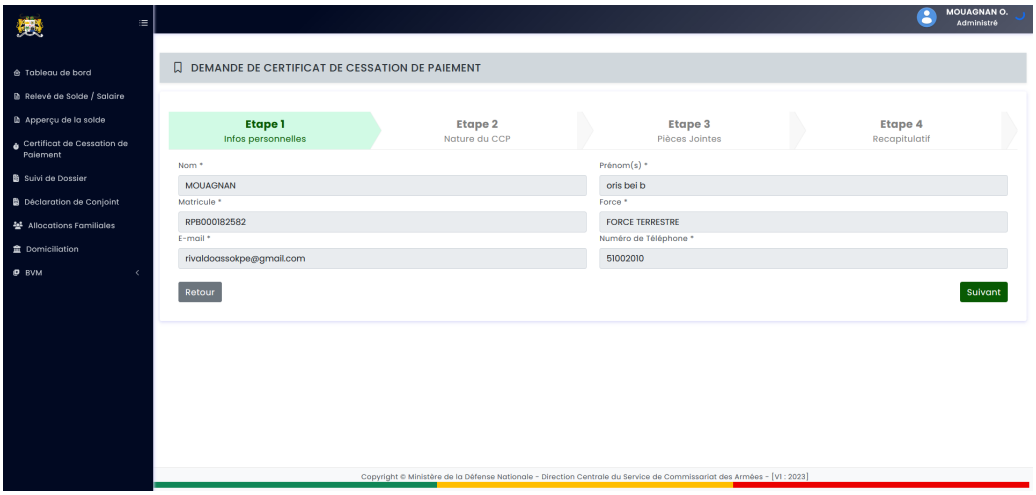


FIGURE 3.8 : Informations Personnelles

2. **La nature du CCP** :
   Après le renseignement de ses informations personnelles, l'interessé est appelé à choisir la nature de son CCP.

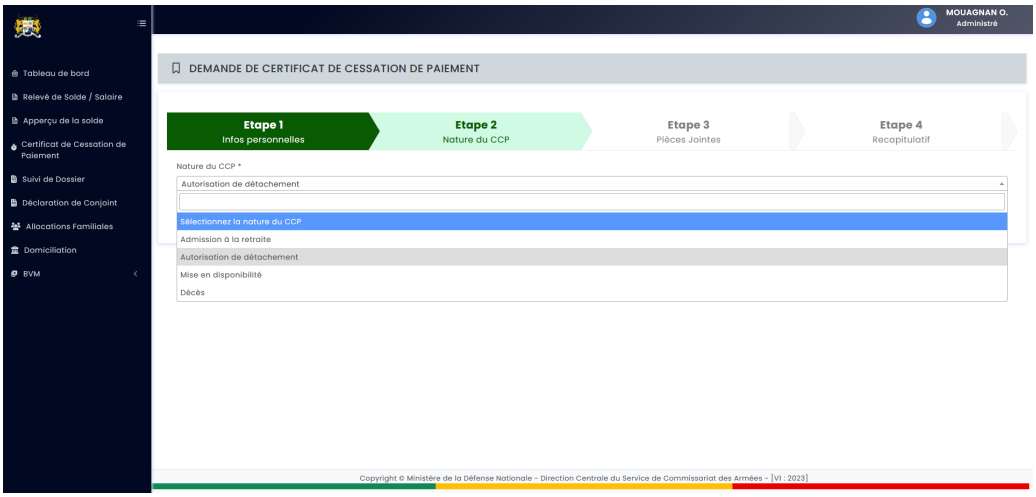


FIGURE 3.9 : Nature CCP

3. **L'ajout des pièces requises** :
   Après le sélectionnement de la nature du CCP, l'interessé est appelé à joindre les pièces requises.

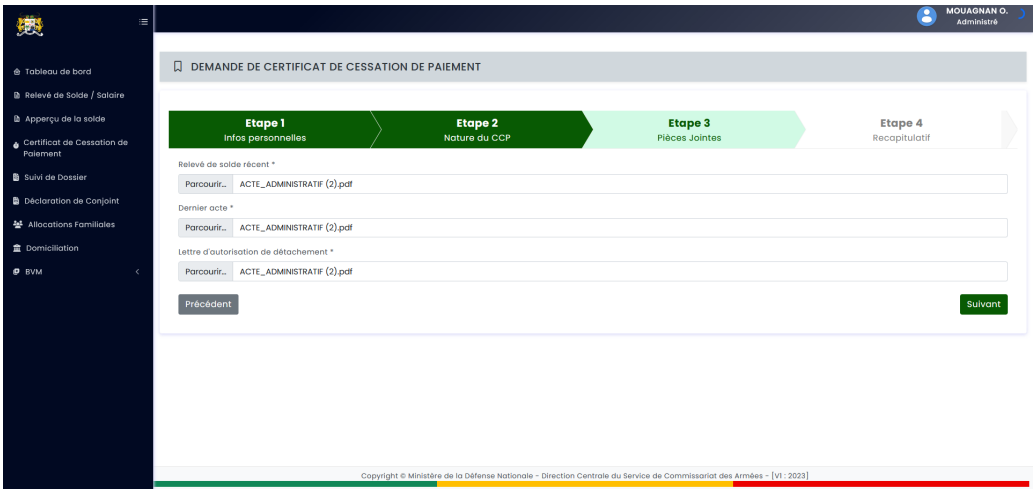


FIGURE 3.10 : Ajout Pièces Requises

4. **L'enregistrement de la demande** :
   Enfin, l'administré enregistre sa demande

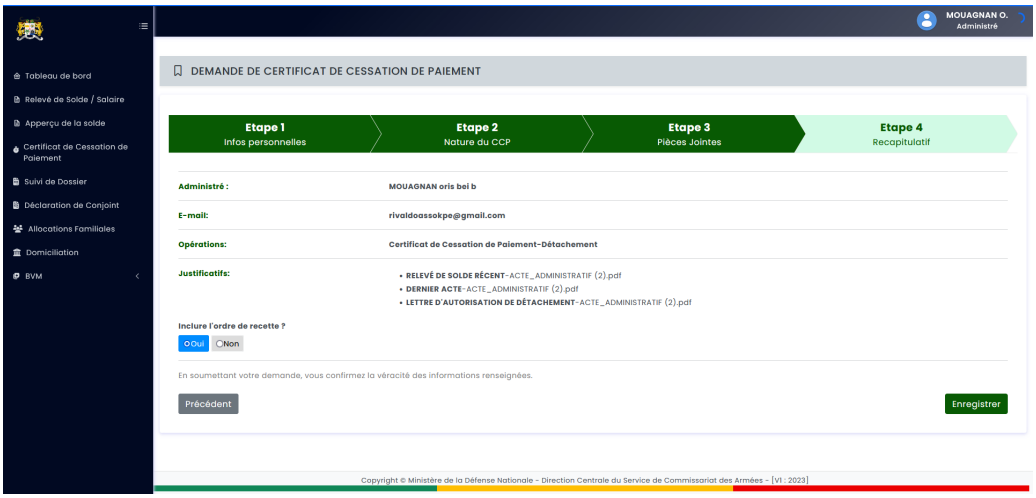


FIGURE 3.11 : Enregistrement

5. **L'ajout de la demande** :
   Après enregistrement, le système ajoute la demande envoyée dans la liste des demandes avec le statut **en cours**.

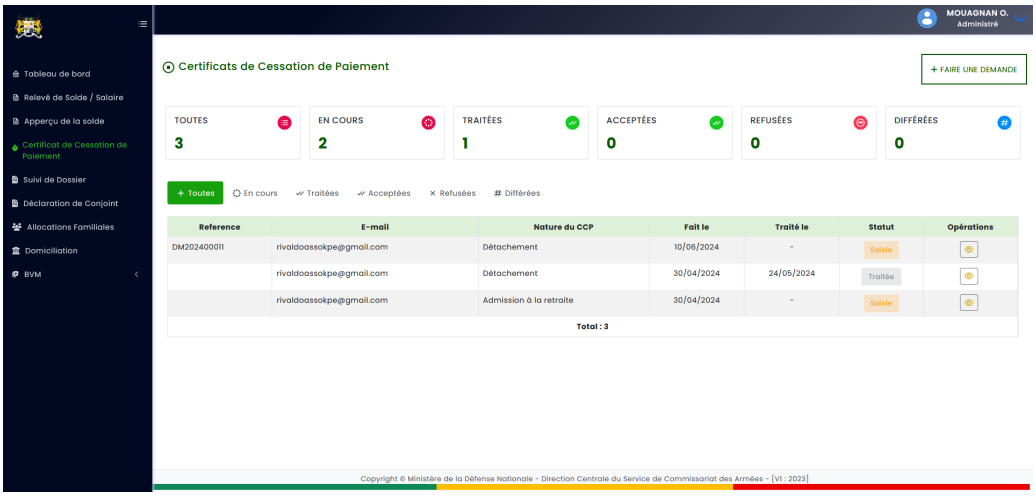


FIGURE 3.12 : Demande en cours

3.1.7 Traitement de demandes

Après l'enregistrement de la demandes, l'agent de la DSF (connecté) peut traiter le dossier

1. **Début du traitement** :
   Afin de notifier à l'administré que son dossier est pris en charge, l'agent de la DSF lance le traitement, ce qui modifie le statut de la demande. La demande est maintenant **en traitement**

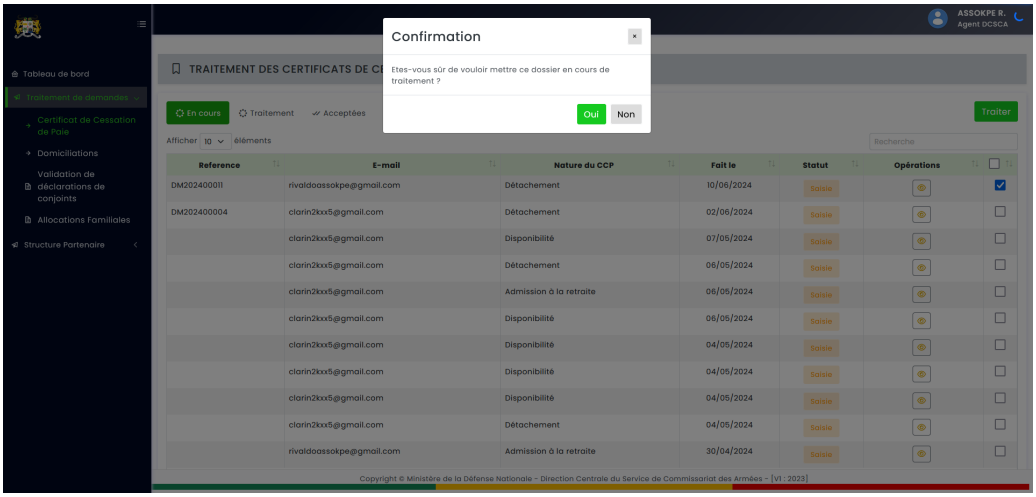


FIGURE 3.13 : Début du traitement

2. **Traitement** :
   Une fois le processus lancé, l'agent de la DSF procède au traitement effectif, qui, dans ce cas, n'est pas dématérialisé. Le résultat de ce traitement peut prendre la forme d'une demande rejetée, différée ou acceptée. Dans ce scénario particulier, la demande est approuvée et le certificat de cessation est inclus dans la réponse.

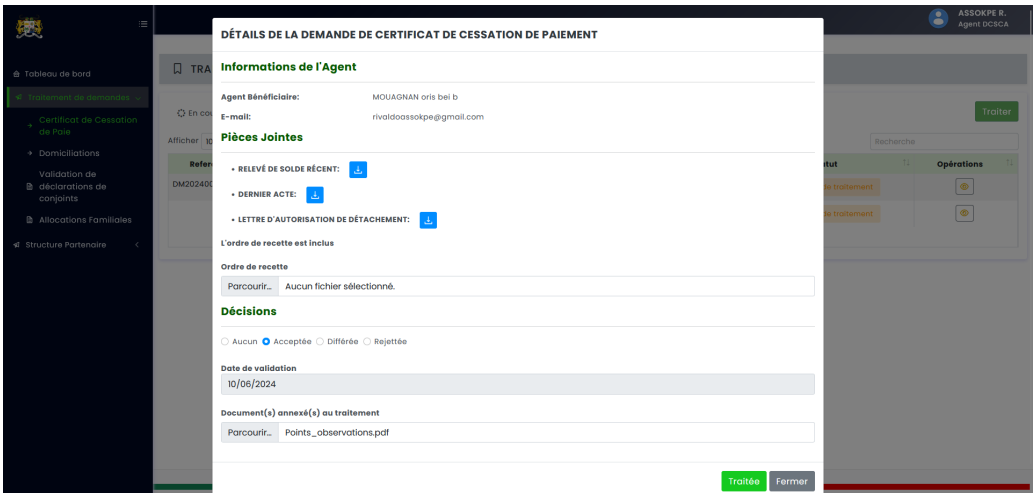


FIGURE 3.14 : Traitement

3. **Résultat de la demande** :
   En conclusion, l'usager a la possibilité de vérifier l'état de sa demande et constatera qu'elle a été acceptée, accompagnée du certificat associé.

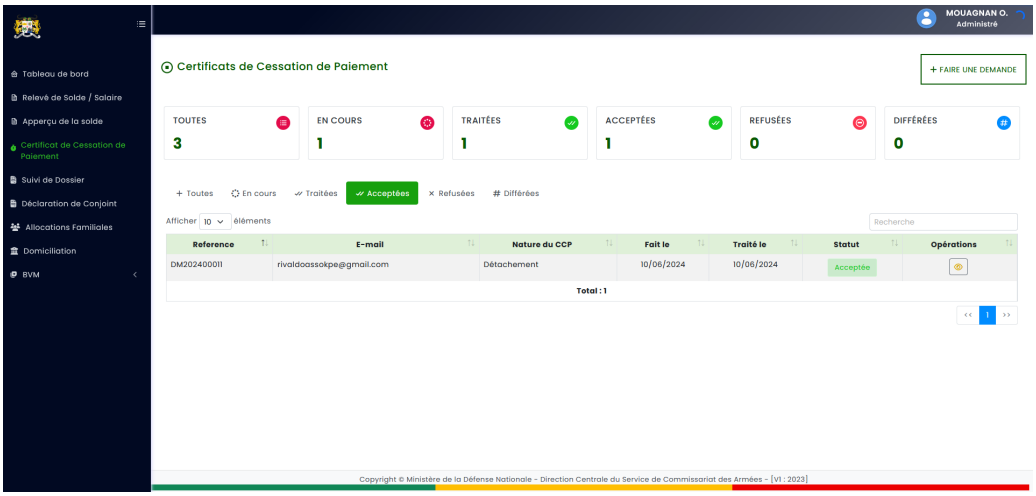


FIGURE 3.15 : Résultat

3.1.8 Déconnexion

L'utilisateur se déconnecte de la plateforme

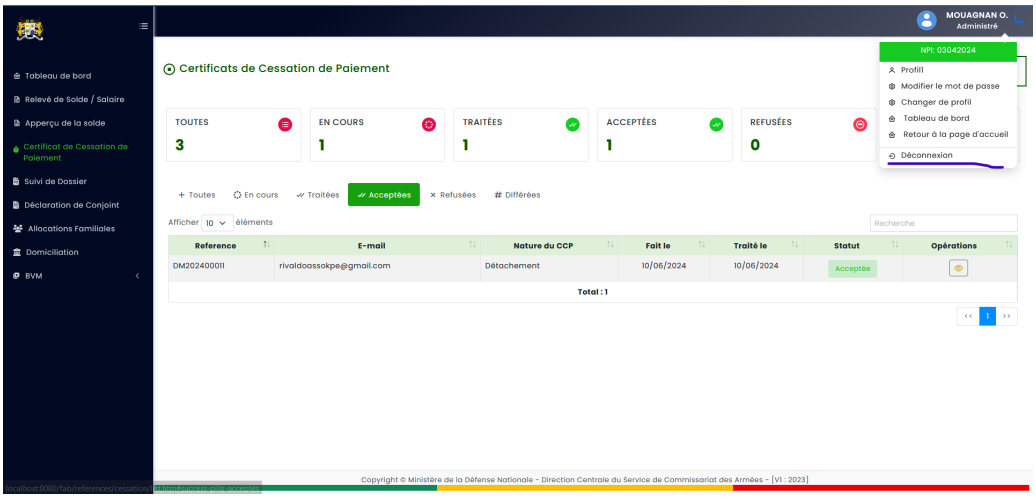


FIGURE 3.16 : Déconnexion

3.2 Analyse critique

GKDY-DSF est un système basé sur la livraison de différents services. Permettre aux administrés de soumettre leurs différents besoins, suivre le traitement de leurs demandes, dans cet but, nous avons construit une plateforme qui répond à ces besoins. Cependant, les traitements des demandes ne sont toujours pas dématérialisés, nous pensons qu'il serait agréable de penser à un moyen de prise de décision automatique. Une intelligence artificielle pourrait alors remplacer ou aider les agents de la solde pour le traitement des demandes afin de diminuer la main d'oeuvre et de réduire les erreurs liées aux traitements.

Conclusion

Ce chapitre a fait l'objet des résultats obtenus à travers la présentation de notre plateforme "**GKDY-DSF**". Les insuffisances liées à GKDY-DSF ont été soulevées et seront utilisées comme base pour améliorer notre application.

Conclusion Générale

La première phase de la dématérialisation des demandes au sein des forces armées béninoises consiste en la mise en œuvre de notre solution, GKDY-DSF. Nous avons élaboré une solution visant à faciliter la formulation des demandes par les administrés et à permettre aux agents des structures de recevoir ces demandes et d'y apporter des résultats. Cependant, pour que cette transition soit réussie, il est impératif que les utilisateurs, en particulier les plus âgés, se familiarisent de plus en plus avec les outils informatiques. Ainsi, la formation du personnel devient une étape cruciale, incluant une section dédiée à la formation du personnel de la DSF, les guidant dans l'utilisation efficace et sécurisée des nouvelles technologies et outils. ;Dans le souci de perfectionner notre application, des perspectives intéressantes sont à envisager.
Ainsi, les perspectives futures sont :

- intégrer un système d'authentification utilisant des empreintes digitales ;
- mettre en place un mécanisme de prise de décision automatisée pour le traitement des demandes des administrés et
- configurer des alertes par courrier électronique pour informer des actions liées à l'application.

Webographie

[1] Documentation eclipse. https://eclipseide.org/, visité en mars 2024.

[2] Documentation java. https://docs.oracle.com/en/java/, visité en mars 2024.

[3] Front-end + back-end. https://www.codecademy.com, visité en mars 2024.

[4] Méthodologie agile. https://www.scrum.org/resources/blog/purpose-being-agile, visité en mars 2024.

[5] plateforme web. https://eloken.com/blog/plateforme-web-un-guide-complet-pour-comprendre-et-creer-votre-propre-plateforme, visité en mars 2024.

[6] service web. https://fr.wikipedia.org/wiki/Service_web, visité en mars 2024.

[7] tableau. https://chatgpt.com/, visité en mars 2024.

[8] Unified modeling language. https://www.tutorialspoint.com/uml/index.htm, visité en mars 2024.

Table des matières

Dédicace ii

Remerciements iii

Résumé iv

. iv

Abstract v

. v

Table des figures vi

Liste des acronymes vii

Introduction 1

1 Revue de littérature 3

Introduction 3

1.1 Généralités sur les services WEB 3

1.2 Revue de littérature 4

1.2.1 Analyse de l'existant 4

1.2.2 Insuffisances de l'existant 5

1.3 Solutions existantes : e-Services de la DGB du Bénin 6

1.3.1 Page d'accueil 6

1.3.2 Points clés de la plateforme 6

1.3.3 Limites des e-Services de la DGB 7

1.4 Définition des caractéristiques de notre plateforme 7

Conclusion 8

2 Solutions, choix techniques et implémentation 9

Introduction 9

2.1 Description de la solution 9

2.2 Modélisation 9

2.2.1 Diagramme de cas d'utilisation 10

2.2.2 Diagramme de classe 11

2.2.3 Diagramme de séquences 13

2.3 Choix techniques 15

2.3.1 Méthodologie de développement 15

2.3.2 Outils utilisés 15

2.3.3 Technologies utilisées . . . 19
2.3.3.1 Front-end . . . 19
2.3.3.2 Back-end . . . 20
2.3.4 Stratégie de sécurisation . . . 23
2.3.5 Tests et Validation . . . 23
Conclusion . . . 23

3 Résultats et analyse critique **25**
Introduction . . . 25
3.1 Présentation du résultat . . . 25
3.1.1 Accueil . . . 25
3.1.2 Connexion . . . 26
3.1.3 Inscription . . . 26
3.1.4 Profil Administrateur . . . 26
3.1.5 Profil Administre . . . 27
3.1.6 Formulation de Demandes . . . 28
3.1.7 Traitement de demandes . . . 31
3.1.8 Déconnexion . . . 32
3.2 Analyse critique . . . 33
Conclusion . . . 33

Conclusion **34**

Webographie **35**

Table des matières **36**