



REPUBLIQUE TUNISIENNE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA
RECHERCHE SCIENTIFIQUE

DIRECTION GENERALE DES ETUDES TECHNOLOGIQUES

INSTITUT SUPERIEUR DES ETUDES TECHNOLOGIQUES DE
CHARGUIA

Département Technologies de l'Informatique



RAPPORT De Stage de Perfectionnement

Parcours : Développement des Systèmes d'Information

Sujet :

CIRCUIT DE LA DEMANDE DE STAGE

Elaboré par
Sofien LAGHOUANEM

Encadré par :

Mr. Ramzi BOUAZIZ

Société d'accueil : Tunisair

Année Universitaire : 2024/2025

Remerciements

À l'issue de ce projet, je tiens à exprimer ma gratitude aux personnes qui ont contribué, de près ou de loin, à sa réalisation.

Je remercie tout particulièrement Monsieur Ramzi BOUAZIZ, mon encadrant, pour ses conseils et son soutien tout au long de ce projet. Ses orientations et ses retours ont été précieux pour structurer mon travail et surmonter les défis techniques rencontrés.

Je tiens également à remercier Madame Sameh HOUIMLI, du service administratif, pour les informations et les clarifications qu'elle a fournies concernant les processus de gestion des demandes de stage. Son aide a été essentielle pour comprendre les besoins métier et les attentes du projet.

Enfin, je suis reconnaissant envers toutes les personnes qui ont contribué, directement ou indirectement, à la réalisation de ce travail. Leur soutien, même à distance, a été apprécié et a permis de mener ce projet à son terme.

Sommaire

INTRODUCTION GENERALE.....	1
CHAPITRE 1 : PRESENTATION DU CADRE DU STAGE.....	2
I. PRESENTATION DE LA SOCIETE	2
II. PRESENTATION DU DOMAINE METIER	3
III. ETUDE DE L'EXISTANT	3
1. Description de l'existant.....	3
2. Critique de l'existant	4
3. Solution proposée	5
IV. CAHIER DES CHARGES	6
1. Besoins fonctionnels.....	6
2. Besoins non fonctionnels	7
CHAPITRE 2 : MODELISATION COMPORTEMENTALE.....	9
I. IDENTIFICATION DES ACTEURS.....	9
II. DIAGRAMME DE CAS D'UTILISATION	9
1. Diagramme du cas d'utilisation général.....	9
2. Description textuelle des cas d'utilisation	10
III. ANALYSE DES CAS D'UTILISATION.....	14
1. Diagramme de séquence système	14
CHAPITRE 3 : MODELISATION STRUCTURELLE	18
I. DIAGRAMME DE CLASSES	18
1. Descriptif du diagramme.....	18
2. Représentation du diagramme.....	19
II. MODELISATION DE LA BASE DE DONNEES	20
III. DIAGRAMME DE DEPLOIEMENT	20
CHAPITRE 4 : REALISATION.....	21
I. ENVIRONNEMENT DE DEVELOPPEMENT	21
1. Environnement matérielle.....	21
2. Environnement logiciel.....	21
3. Présentation de l'environnement logiciel.....	21
II. GESTION DE L'AUTHENTIFICATION AVEC JWT	26
III. PRINCIPALES INTERFACES GRAPHIQUES	28
IV. MAQUETTE D'INTERFACE	32
CONCLUSION GENERALE.....	35
NETOGRAPHIE	36
ANNEXE	37

Listes des figures

Figure 1: Diagramme cas d'utilisation générale	10
Figure 2: Diagramme de Séquence " Postuler Demande"	14
Figure 3: Diagramme de séquence de "Valider une Demande"	15
Figure 4: Diagramme de séquence "Gérer Utilisateur"	16
Figure 5: Diagramme de class	19
Figure 6: Diagramme de Deploiement	20
Figure 7: Log MySQL.....	22
Figure 8: Logo Angular 19	22
Figure 9: Logo SpringBoot	23
Figure 10: Request d'authentification simple	24
Figure 11: Logo StartUML.....	24
Figure 12: Logo VS Code	24
Figure 13: Logo STS	24
Figure 14: Logo IntelliJ.....	24
Figure 15: Logo Figma.....	25
Figure 16: Logo HTML & CSS	25
Figure 17: Logo TS	25
Figure 18: Logo Java.....	25
Figure 19: Logo Postman	25
Figure 20:Schéma de l'authentification	27
Figure 21: Interface d'Authentification	28
Figure 22: Interface du liste des demandes	29
Figure 23: Interface du Tableau de bord du Service Administrative.....	29
Figure 24: Interface d'une demande d'un stagiaire	30
Figure 25:Interface du demande de stage.....	31
Figure 26: Interface du suivie du stagiaire	30
Figure 27: Maquette du formulaire de demande de stage	32
Figure 28: Maquette authentification	33
Figure 29: Maquette Liste des demandes en attente.....	33
Figure 30: Maquette du Table de bord du Stagiaire	34
Figure 31: Diagramme de cas d'utilisation complet	37

Listes des tableaux

Tableau 1: Description Textuelle du soumis de demande de stage	10
Tableau 2: Description Textuelle Du Validation D'une Demande.....	11
Tableau 3: Description Textuelle Du Téléversement d'un document.....	11
Tableau 4: Description Textuelle Du Consultation d'une Demande.....	12
Tableau 5: Description Textuelle du Cas d'utilisation "Gérer Utilisateur"	13
Tableau 6: Description Matérielle	21
Tableau 7: Description Logicielle	21

Introduction générale

Dans un contexte de transformation numérique accélérée, les entreprises de grande envergure comme Tunisair se doivent d'optimiser leurs processus métier par des solutions technologiques innovantes. La gestion des stages académiques, pierre angulaire de la formation pratique des étudiants, constitue un enjeu stratégique tant pour les établissements d'enseignement que pour les entreprises d'accueil. C'est dans cette dynamique que s'inscrit notre projet de rationalisation du circuit des demandes de stage.

Ce rapport présente la conception et la réalisation d'une solution web intégrée visant à digitaliser l'ensemble du parcours de demande de stage. Face aux défis posés par le traitement manuel actuel - caractérisé par des délais prolongés, des risques d'erreurs humaines et une traçabilité limitée - notre système propose une automatisation complète du workflow. L'objectif central est de créer une plateforme sécurisée et collaborative permettant une gestion fluide depuis la soumission initiale jusqu'à l'attestation finale, tout en garantissant une expérience utilisateur optimale pour l'ensemble des parties prenantes.

La structure de ce rapport s'articule autour de quatre chapitres clés. Le premier chapitre contextualise le projet en présentant le cadre organisationnel de Tunisair et l'analyse détaillée de l'existant. Le deuxième chapitre dévoile notre modélisation comportementale à travers des diagrammes UML spécifiques. Le troisième chapitre expose l'architecture technique et les choix de modélisation structurelle. Enfin, le quatrième chapitre détaille la mise en œuvre concrète de la solution, incluant les aspects de sécurité et les interfaces utilisateur développées. Chaque étape de cette étude sera illustrée par des exemples concrets et des schémas explicatifs, reflétant la rigueur méthodologique appliquée tout au long de ce projet.

Chapitre 1 : Présentation du cadre du stage

Le présent chapitre constitue une introduction détaillée au projet, en commençant par une présentation de l'entreprise d'accueil, suivie d'une étude de l'existant dans la gestion des demandes de stage. Cette analyse permettra de mieux comprendre les enjeux du projet et les besoins à satisfaire. Ce chapitre couvre également les modalités de travail en place dans l'entreprise, en identifiant les forces et les faiblesses des solutions existantes.

I. Présentation de la société

La société d'accueil pour ce projet est Tunisair, une entreprise de transport aérien de premier plan en Tunisie. Tunisair est la compagnie nationale qui relie la Tunisie à de nombreuses destinations internationales et régionales. Elle est un acteur clé dans le secteur de l'aviation en Tunisie, contribuant de manière significative à l'économie locale et à l'industrie touristique.

Tunisair offre une gamme de services, incluant le transport de passagers, la maintenance d'avions, et des services de fret aérien. En plus de ses activités, la société s'engage aux maintenances techniques activement dans des projets de digitalisation et d'amélioration des processus internes pour renforcer son efficacité et sa compétitivité sur le marché international.

Le projet d'automatisation des demandes de stage se situe au sein de la **Direction Télécom et Réseaux** de Tunisair. Ce département joue un rôle stratégique dans la gestion des infrastructures réseau et des technologies de communication qui soutiennent l'ensemble des opérations de l'entreprise. Il est chargé de la gestion des systèmes de communication, ainsi que des technologies liées aux systèmes d'information et aux ressources humaines et Service administrative, comme les demandes de stage. [1]

II. Présentation du domaine métier

Le domaine métier en question est la gestion des stages et des demandes de stage au sein des grandes entreprises, en particulier dans le secteur public et privé. La gestion des stages, bien que cruciale pour l'insertion des étudiants dans le monde professionnel, reste souvent un processus long et manuel dans de nombreuses entreprises. Cela inclut la soumission des demandes de stage, la validation par les différents responsables, la gestion des documents et la communication entre les différentes parties prenantes.

L'automatisation de ces processus représente un enjeu majeur pour améliorer l'efficacité administrative, réduire les erreurs humaines et offrir une meilleure expérience aux étudiants et aux responsables. En digitalisant le parcours des demandes de stage, Tunisair peut optimiser leurs ressources tout en garantissant une meilleure traçabilité et transparence du processus.

III. Etude de l'existant

Cette section examine les solutions actuelles utilisées pour gérer les demandes de stage et identifie les points faibles des systèmes existants, tout en proposant une solution innovante et améliorée

1. Description de l'existant

Actuellement, la gestion des demandes de stage chez Tunisair est effectuée de manière manuelle, nécessitant plusieurs étapes administratives et interactions entre différents départements tels que les ressources humaines, le service administratif, et les Encadrants. Les stagiaires, une fois leur demande soumise, doivent attendre une validation de pouvoir poursuivre avec leur processus.

Le processus inclut les étapes suivantes :

- Les stagiaires soumettent leurs demandes de stage sur papier ou par e-mail.
- Les demandes sont ensuite envoyées à l'Encadrant et après au Service Administratif pour validation.
- Une fois validée, l'information est transmise à la Direction des Ressources Humaines pour l'enregistrement.
- Le stagiaire doit se présenter après 2 ou 3 jours au Centre de Formation de Tunisair pour fournir la convocation de stage.

- La convocation de stage doit être remise à l'institut ou à la faculté pour signature, puis à la Municipalité pour la légalisation.
- Le stagiaire doit rendre la convocation signée au Centre de Formation pour obtenir deux documents : Laisser Passer et Prise de Service à remettre au service administratif de préparer le bulletin de mouvement.
- Pour obtenir une attestation de stage, le stagiaire doit se présenter avec le Laisser Passer le rapport de stage signé par l'encadrant

La validation finale du stage se fait manuellement, ce qui génère des retards et des erreurs humaines dans le processus.

2. Critique de l'existant

Bien que le processus actuel permette d'assurer la gestion des demandes de stage, plusieurs faiblesses sont identifiées :

- **Manque d'automatisation** : Le processus est entièrement manuel, ce qui entraîne des retards dans la validation des demandes et augmente les risques d'erreurs humaines, telles que des demandes oubliées ou mal traitées.
- **Manque de visibilité et de traçabilité** : Il est difficile pour les parties prenantes (stagiaires, Encadrants, responsables RH, service Administrative, Centre de Formation) de suivre l'état d'avancement des demandes. La communication entre les différents départements est fragmentée, ce qui rend le processus opaque.
- **Complexité administrative** : Le processus nécessite une série d'interactions humaines (approbations et validations), ce qui peut rendre le processus lent, en particulier durant les périodes de forte demande. Les demandes manuelles sont également sujets à des erreurs de documentation.
- **Technologies obsolètes ou inefficaces** : Bien que des outils de gestion de projet et de collaboration soient utilisés pour certaines parties du processus, ces outils ne sont pas adaptés aux spécificités des demandes de stage et ne permettent pas une gestion fluide et rapide des demandes. L'absence d'un système unifié crée une perte de temps et d'efficacité.

La problématique réside donc dans l'inefficacité du processus actuel, qui ralentit l'ensemble du traitement des demandes et augmente la charge administrative. L'entreprise a besoin d'une solution numérique qui permette de simplifier et automatiser ces processus pour améliorer la réactivité et réduire les erreurs.

3. Solution proposée

Afin de répondre à ces problématiques, nous proposons une solution numérique complète qui permettra d'automatiser le processus de gestion des demandes de stage chez Tunisair. Cette solution sera basée sur un système web moderne utilisant les technologies Spring Boot pour le backend, Angular pour le frontend, et MySQL pour la gestion des données. L'objectif est de créer un processus fluide, transparent et efficace.

Voici les principales caractéristiques de la solution proposée :

- **Automatisation des demandes de stage** : Les stagiaires seront auto-inscrits dès que leur demande sera validée par un Encadrant, ce qui évitera tout traitement manuel supplémentaire et permettra une gestion en temps réel.
- **Visibilité en temps réel** : Chaque partie prenante (stagiaire, Encadrant, Service Administratif, DCRH, Centre de formation) pourra suivre l'avancement des demandes via un tableau de bord simple et intuitif.
- **Gestion centralisée et sécurisée** : Le processus sera entièrement digitalisé, permettant un suivi transparent et sécurisé des données personnelles et des documents associés aux demandes de stage. L'intégration de Spring Security assurera la sécurité des données sensibles.
- **Simplification des étapes administratives** : L'application permettra une gestion automatisée des approbations, ainsi que le transfert des documents à chaque étape du processus.

En résumé, cette solution vise à rendre la gestion des demandes de stage plus rapide, plus transparente et moins sujette à erreurs, tout en améliorant l'expérience des utilisateurs internes et des stagiaires. Elle apportera des gains d'efficacité significatifs en réduisant les délais de traitement et en simplifiant les tâches administratives.

IV. Cahier des charges

1. Besoins fonctionnels

Les besoins fonctionnels de l'application se divisent en plusieurs sous-besoins, organisés par catégories spécifiques selon le processus de demande de stage.

Gestion des demandes de stage

- Soumission des demandes de stage

Le stagiaire peut soumettre une demande de stage via un formulaire en ligne, en incluant ses informations personnelles et les documents nécessaires.

- Suivi du statut de la demande

Le stagiaire peut suivre l'évolution de la demande si accepter à travers le tableau de bord, avec une visualisation du statut actuel.

Notifications automatiques sont envoyées pour informer le stagiaire de tout changement de statut par email.

- Consultation et validation des demandes par l'Encadrant

L'encadrant peut consulter les demandes soumises par les stagiaires, télécharger les documents et valider ou rejeter la demande.

En cas de validation, un compte stagiaire est créé automatiquement avec les informations nécessaires.

En cas de rejet, l'encadrant peut ajouter des commentaires et un email est envoyé au stagiaire avec les raisons du rejet.

- Gestion des documents associés

Les documents nécessaires sont téléchargeables et téléversables par le stagiaire, l'encadrant et les autres intervenants (Service Administratif, DCRH, Centre de formation).

Chaque acteur a des droits spécifiques de gestion et de validation de ces documents.

Gestion des utilisateurs et des rôles

- Création d'utilisateur

Lorsqu'une demande est validée, un compte stagiaire est automatiquement créé.

D'autres utilisateurs (Service Administratif, DCRH, Encadrants, Centre de Formation) sont créés manuellement par le Service Administratif.

- Gestion des rôles

Chaque utilisateur a des rôles et des permissions spécifiques selon son type : Stagiaire, Encadrant, Service Administratif, DCRH, Centre de Formation.

Sécurité et gestion des accès

- Authentification des utilisateurs

Tous les utilisateurs doivent se connecter à la plateforme avant d'accéder à leurs fonctionnalités spécifiques.

En cas d'oubli de mot de passe, une fonction de récupération est disponible via l'email associé.

2. Besoins non fonctionnels

Les besoins non fonctionnels visent à améliorer l'expérience utilisateur, optimiser les performances du système, et garantir la qualité et la sécurité de l'application.

- **Convivialité et ergonomie**

Concevoir une interface utilisateur intuitive et conviviale pour une expérience utilisateur optimale,

Assurer une navigation fluide à travers les différentes fonctionnalités de l'application,

Veiller à ce que le formulaire et les interactions utilisateur soient clairs et faciles à comprendre,

- **Performance et réactivité**

Optimiser le temps de réponse de l'application pour garantir une expérience utilisateur fluide et efficace.

Minimiser le temps de chargements des pages et des fonctionnalités pour une utilisation plus agréable.

- **Sécurité et confidentialité**

Mettre en œuvre des mesures de sécurité robustes pour protéger les données utilisateur et les transactions.

Utiliser des protocoles de cryptage et des techniques d'authentification pour garantir la confidentialité des informations.

- **Contraintes techniques**

Utiliser des technologies modernes et évolutives telles que Angular et Spring boot pour assurer la flexibilité et la maintenabilité de l'application.

L'analyse de l'existant a révélé les lacunes du système actuel, notamment son manque d'automatisation et de traçabilité. La solution proposée, basée sur une plateforme web moderne, vise à simplifier et accélérer le processus tout en améliorant l'expérience des utilisateurs.

Chapitre 2 : Modélisation Comportementale

Ce chapitre aborde la modélisation comportementale du système, en identifiant les acteurs clés et leurs interactions avec l'application. À travers des diagrammes de cas d'utilisation et de séquence, nous détaillons les processus métier et les flux de données essentiels au bon fonctionnement du système.

I. Identification des Acteurs

L'objectif de cette section est d'identifier les acteurs principaux qui interagiront avec l'application et leurs rôles spécifiques.

- **Stagiaire** : L'étudiant qui soumet une demande de stage et interagit avec l'application pour suivre l'état de sa demande.
- **Encadrant** : Le superviseur du stagiaire qui valide ou rejette la demande de stage, les rapports des demandes, et gère le processus.
- **Service Administratif** : L'équipe responsable de la gestion administrative des demandes de stage et document et de gérer des comptes.
- **Centre de Formation** : Interagir avec l'application pour vérifier la convocation des stagiaires et garantir la conformité des demandes.
- **DCRH** (Direction des Ressources Humaines) : Impliqués dans la validation des documents et dans le suivi des stagiaires.

II. Diagramme de cas d'utilisation

Cette section présente les cas d'utilisation globaux de l'application à travers un diagramme.

1. Diagramme du cas d'utilisation général

Le diagramme ci-dessous montre les générales interactions des acteurs avec le système, pour le diagramme de cas d'utilisateur complet disponible dans les annexes.

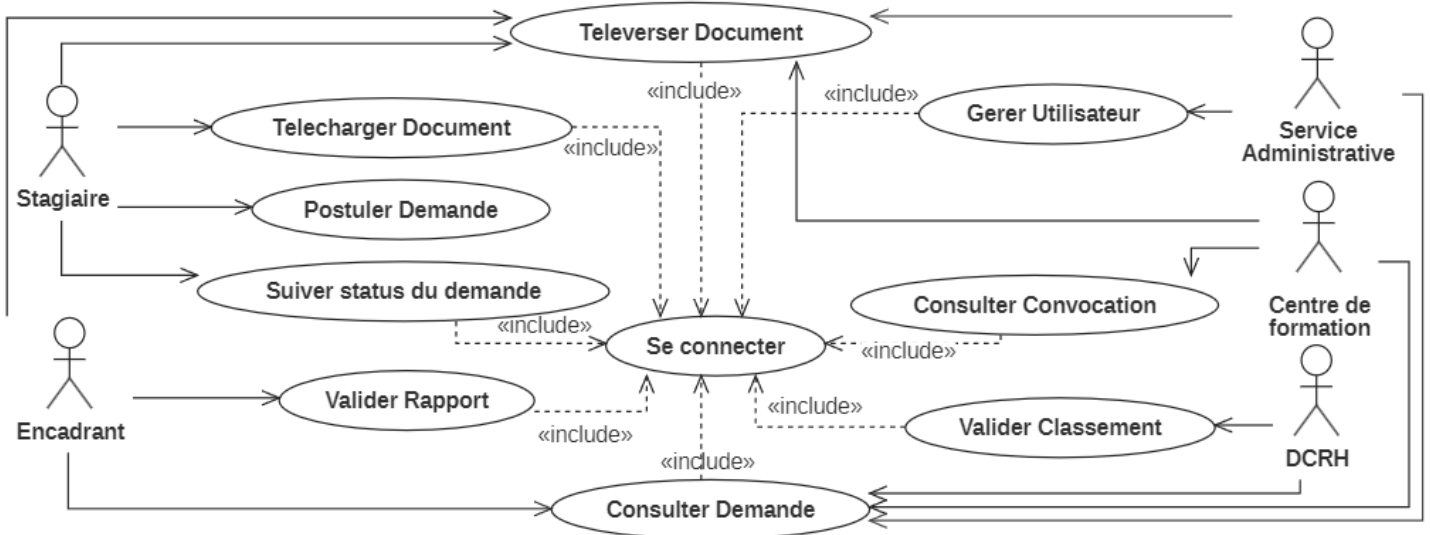


FIGURE 1: DIAGRAMME CAS D'UTILISATION GENERALE

2. Description textuelle des cas d'utilisation

Les descriptions textuelles permettent de fournir une explication détaillée et narrative des interactions entre les acteurs et le système informatique. Nous complétons certains cas d'utilisations par des descriptions textuelles afin de mieux comprendre les fonctionnalités du système.

TABLEAU 1: DESCRIPTION TEXTUELLE DU SOUMIS DE DEMANDE DE STAGE

Sommaire d'identification	
Titre	Postuler Demande
Résumé	Permet au stagiaire de soumettre une demande de stage via un formulaire en ligne avec les documents requis.
Acteurs	Stagiaire (principal)
Description des scénarios	
Préconditions	<ul style="list-style-type: none"> • Le stagiaire dispose des documents nécessaires. • L'application est accessible et fonctionnelle.
Scénario Nominal	<ol style="list-style-type: none"> 1. Le stagiaire accède au formulaire de demande de stage. 2. Le système affiche les champs obligatoires : informations personnelles (nom, prénom, CIN, email, Tel, Institue), détails du stage (type, dates, spécialité, Niveau, Année). 3. Le stagiaire remplit le formulaire et téléverse les documents requis. 4. Le stagiaire clique sur « Soumettre ». 5. Le système enregistre la demande pour l'Encadrant.

Scénario Alternatif	A1. Demande de Même Type Soumis : <ol style="list-style-type: none"> 1. Le système affiche un message d'erreur : « Une demande de ce type de stage existe déjà. ». 2. Le stagiaire remplit et soumet un autre type de stage.
Postcondition	<ul style="list-style-type: none"> • La demande est en attente de validation par l'Encadrant.

TABLEAU 2: DESCRIPTION TEXTUELLE DU VALIDATION D'UNE DEMANDE

Sommaire d'identification	
Titre	Valider une Demande
Résumé	Permet à l'Encadrant de valider ou rejeter une demande de stage.
Acteurs	Encadrant (principal)
Description des scénarios	
Préconditions	<ul style="list-style-type: none"> • L'Encadrant est authentifié. • Une demande est en attente de validation.
Scénario Nominal	<ol style="list-style-type: none"> 1. L'Encadrant consulte la liste des demandes en attente. 2. Le système affiche les détails de la demande (documents, informations du stagiaire). 3. L'Encadrant Télécharge la demande, la signe et Téléverse la demande Signer. 4. L'Encadrant clique sur « Valider ». 5. Le système génère un compte stagiaire et envoie les identifiants par email. 6. La demande passe au Service Administratif.
Scénario Alternatif	A1. Rejet de la demande : <ol style="list-style-type: none"> 1. L'Encadrant écrit la raison de rejet. 2. L'Encadrant clique sur « Rejeter ». 3. Le système supprime la demande après 1 heure et notifie le stagiaire.
Postcondition	<ul style="list-style-type: none"> • La demande est validée ou rejetée.

TABLEAU 3: DESCRIPTION TEXTUELLE DU TELEVERSEMENT D'UN DOCUMENT

Sommaire d'identification	
Titre	Téléverser un Document
Résumé	Permet à un acteur de téléverser un document associé à une demande.
Acteurs	Stagiaire, Encadrant, Service Administratif, Centre de Formation

Description des scénarios	
Préconditions	<ul style="list-style-type: none"> • L'acteur est authentifié. • La demande existe dans le système.
Scénario Nominal (Stagiaire)	<ol style="list-style-type: none"> 1. Le stagiaire consulte sa demande validée. 2. Le système affiche l'option « Téléverser la convocation signée ». 3. Le stagiaire sélectionne le fichier et clique sur « Envoyer ». 4. Le système met à jour la demande pour le Centre de Formation.
Scénario Alternatif (Centre de Formation)	A1. Document non conforme : <ol style="list-style-type: none"> 1. Le Centre de Formation rejette le document. 2. Le système envoie un email au stagiaire avec les corrections nécessaires.
Postcondition	<ul style="list-style-type: none"> • Le document est disponible pour l'étape suivante du workflow.

TABLEAU 4: DESCRIPTION TEXTUELLE DU CONSULTATION D'UNE DEMANDE

Sommaire d'identification	
Titre	Consulter une Demande
Résumé	Permet à tous les acteurs de consulter l'état et documents d'une demande.
Acteurs	Encadrant, Service Administratif, DCRH, Centre de Formation.
Description des scénarios	
Préconditions	<ul style="list-style-type: none"> • L'acteur est authentifié. • La demande existe dans le système.
Scénario Nominal	<ol style="list-style-type: none"> 1. L'acteur accède à son tableau de bord. 2. Le système affiche la liste des demandes. 3. L'acteur clique sur une demande pour voir les détails : statut, information, documents (à télécharger).
Scénario Alternatif	A1. Demande introuvable : <ol style="list-style-type: none"> 1. Le système affiche un message : « Aucun stagiaire trouvé ».
Postcondition	<ul style="list-style-type: none"> • L'acteur a accès aux informations complètes de la demande.

TABLEAU 5: DESCRIPTION TEXTUELLE DU CAS D'UTILISATION "GERER UTILISATEUR"

Sommaire d'identification	
Titre	Gérer les Utilisateurs
Résumé	Permet au Service Administratif de créer, modifier ou supprimer des comptes utilisateurs.
Acteurs	Service Administratif (principal).
Description des scénarios	
Préconditions	<ul style="list-style-type: none"> • Le Service Administratif est authentifié. • Les droits d'administration sont activés.
Scénario Nominal	<ol style="list-style-type: none"> 4. Le Service Administratif accède à l'interface « Gestion des Utilisateurs ». 5. Le système affiche la liste des utilisateurs existants. 6. Le Service Administratif clique sur « Ajouter un Utilisateur ». 7. Le système affiche un formulaire (nom, prénom, email, mot de passe, rôle). 8. Après validation, le système génère un mot de passe temporaire et envoie un email.
Scénario Alternatif	<p>A1. Email déjà existant :</p> <ol style="list-style-type: none"> 1. Le système affiche un message : « Cet email déjà existe ».
Postcondition	<ul style="list-style-type: none"> • Le compte utilisateur est créé/modifié/supprimé.

III. Analyse des cas d'utilisation

1. Diagramme de séquence système

- Le diagramme de séquence de postuler demande

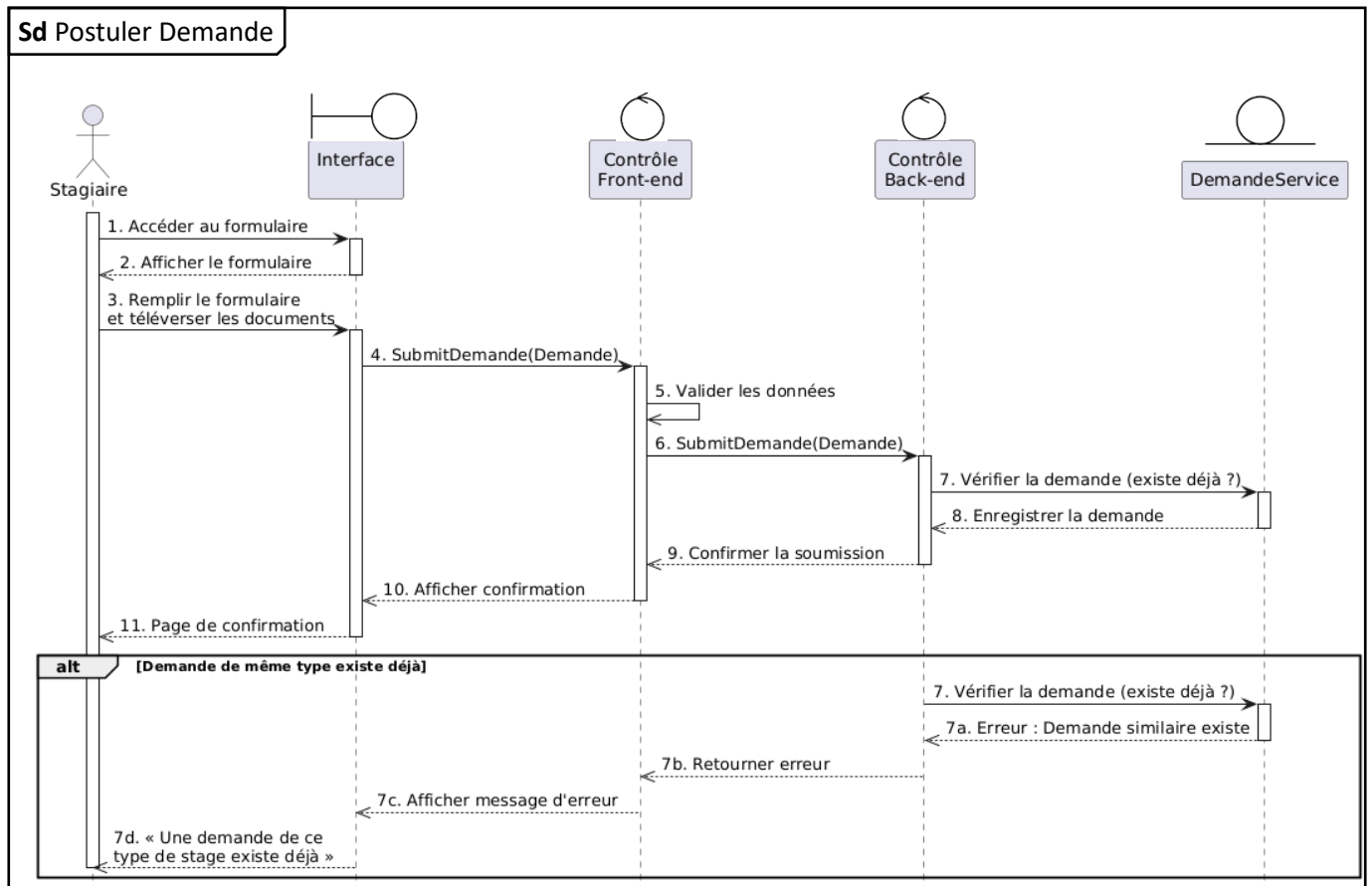


FIGURE 2: DIAGRAMME DE SEQUENCE " POSTULER DEMANDE"

Ce diagramme illustre le processus de soumission d'une demande de stage par un stagiaire. Il montre les interactions entre l'interface utilisateur, le contrôleur front-end, le contrôleur back-end, et le service de gestion des demandes. Le diagramme inclut également la validation des données et la gestion des erreurs en cas de demande dupliquée.

- Le diagramme de séquence de Valider une Demande

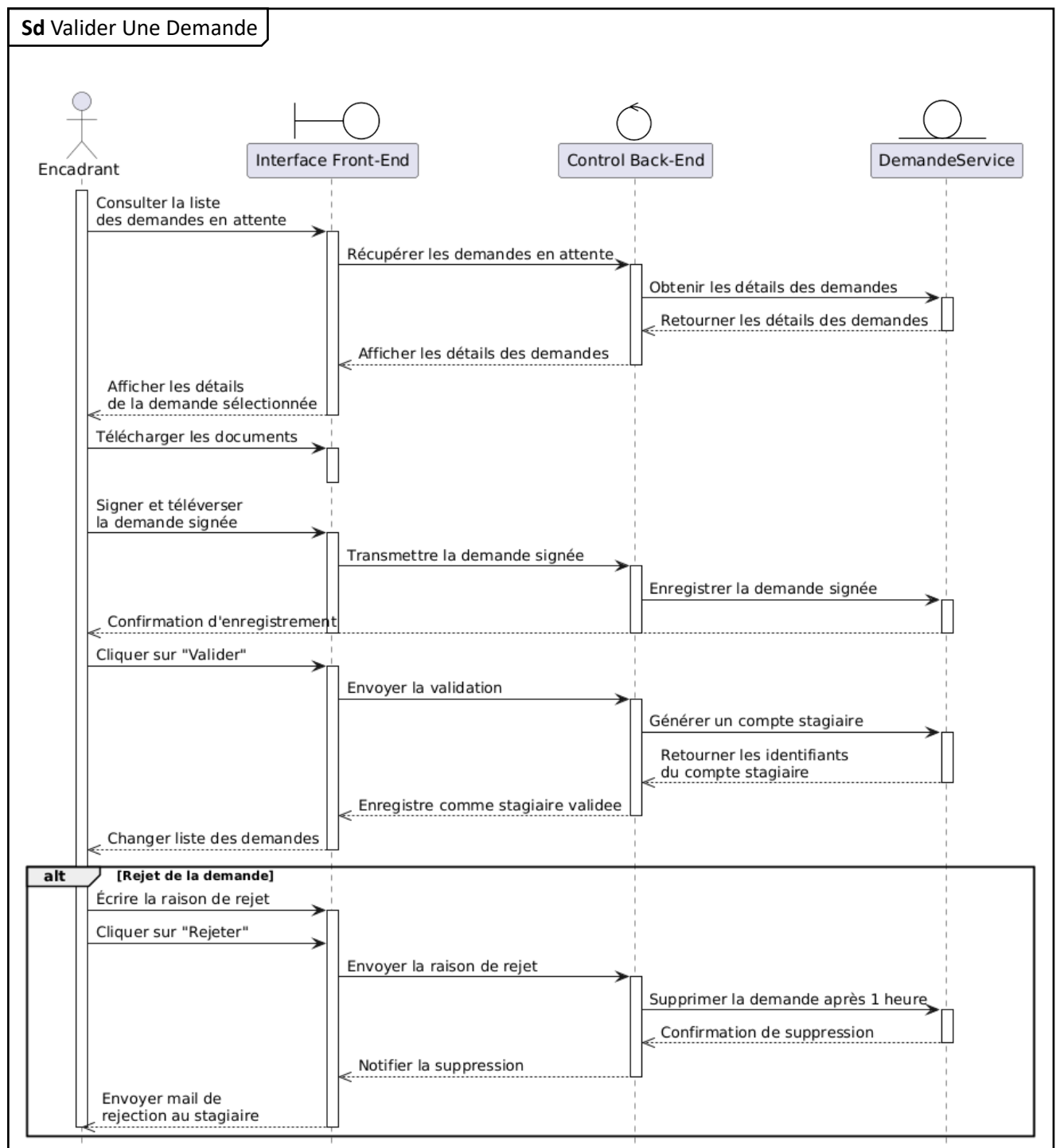


FIGURE 3: DIAGRAMME DE SEQUENCE DE "VALIDER UNE DEMANDE"

Ce diagramme décrit le processus de validation d'une demande de stage par un encadrant. Il inclut la consultation des détails de la demande, le téléchargement des documents, la signature, et la validation ou le rejet de la demande. En cas de validation, un compte stagiaire est automatiquement créé.

- Le diagramme de séquence de Gérer Utilisateur.

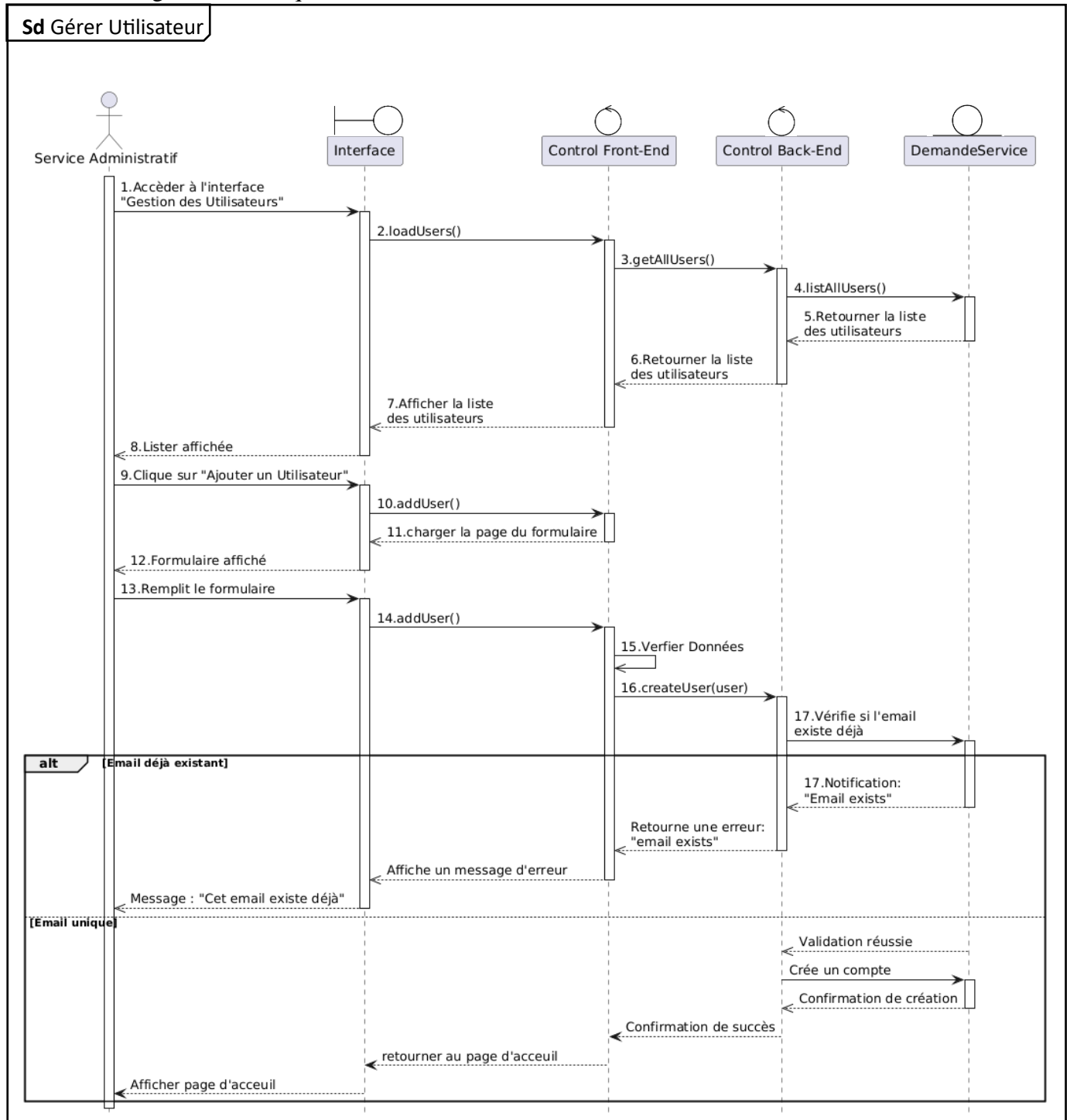


FIGURE 4: DIAGRAMME DE SEQUENCE "GERER UTILISATEUR"

Ce diagramme montre le processus de gestion des utilisateurs par le service administratif. Il inclut la consultation de la liste des utilisateurs, l'ajout d'un nouvel utilisateur, la validation des données, et la gestion des erreurs en cas d'email déjà existant.

La modélisation comportementale a permis de formaliser les interactions entre les acteurs et le système, en mettant en évidence les processus clés tels que la soumission, la validation et le suivi des demandes de stage. Ces diagrammes constituent une base solide pour la conception technique du système.

Chapitre 3 : Modélisation Structurelle

Dans ce chapitre, nous allons aborder la question du comment faire en présentant la conception détaillée de la solution sous forme de diagramme de classes et de base de données. Cette approche permet de décrire les entités, leurs relations et la façon dont les données seront structurées dans l'application, et finalisons par quelques interfaces et maquettes.

I. Diagramme de classes

1. Descriptif du diagramme

Le diagramme de classes de l'application inclut plusieurs classes principales :

Classe Document : Représente les documents liés à une demande. Chaque document possède un identifiant (id), un type (type), un nom (nom), un fichier (fichier), et un statut (statuts). Cette classe contient également des méthodes pour mettre à jour le type et statut du document, le télécharger, le téléverser, et valider ou rejeter un document.

Classe Utilisateur : Représente les utilisateurs de l'application, qui peuvent être des stagiaires, des encadrants, des membres du service administratif, etc. Elle contient des informations personnelles comme l'identifiant (id), le nom (nom), le prénom (prenom), l'email (email), et le mot de passe (passe). Cette classe permet également la gestion des utilisateurs (création, modification, suppression, consultation).

Classe Demande : Représente une demande de stage. Elle est liée à un stagiaire et contient des informations telles que l'identifiant de la demande (id), le type de stage (type), la période du stage (debutStage et finStage), ainsi que le statut de la demande (statuts). Des méthodes associées permettent de soumettre, valider, ou rejeter la demande, ainsi que de récupérer des informations sur le stagiaire et les types de documents associés.

Classe Stagiaire : Représente un stagiaire dans le système. Cette classe hérite de la classe Utilisateur et contient des informations supplémentaires comme l'email personnel (emailPerso), le numéro de CIN (cin), le numéro de téléphone (tel), l'institut d'origine (institut), et le niveau d'étude (niveau). Elle dispose de méthodes permettant de s'enregistrer et de désactiver un compte.

Rôles : Des classes telles que ServiceAdministrative, Encadrant, DCRH, et CentreDeFormation représentent les différents rôles dans l'application. Ces rôles n'ont pas de méthodes spécifiques, mais leur logique est centralisée dans la classe Utilisateur et contrôlée par l'API.

Les relations entre les classes sont les suivantes :

Un stagiaire peut créer plusieurs demandes.

Une demande associe plusieurs documents et peut être validée ou rejetée par un encadrant.

Un service administratif gère plusieurs utilisateurs.

L'utilisateur peut consulter un ou plusieurs demandes.

Le centre de formation peut valider ou rejeter un ou plusieurs documents.

Le DCRH peut valider un ou plusieurs documents.

2. Représentation du diagramme

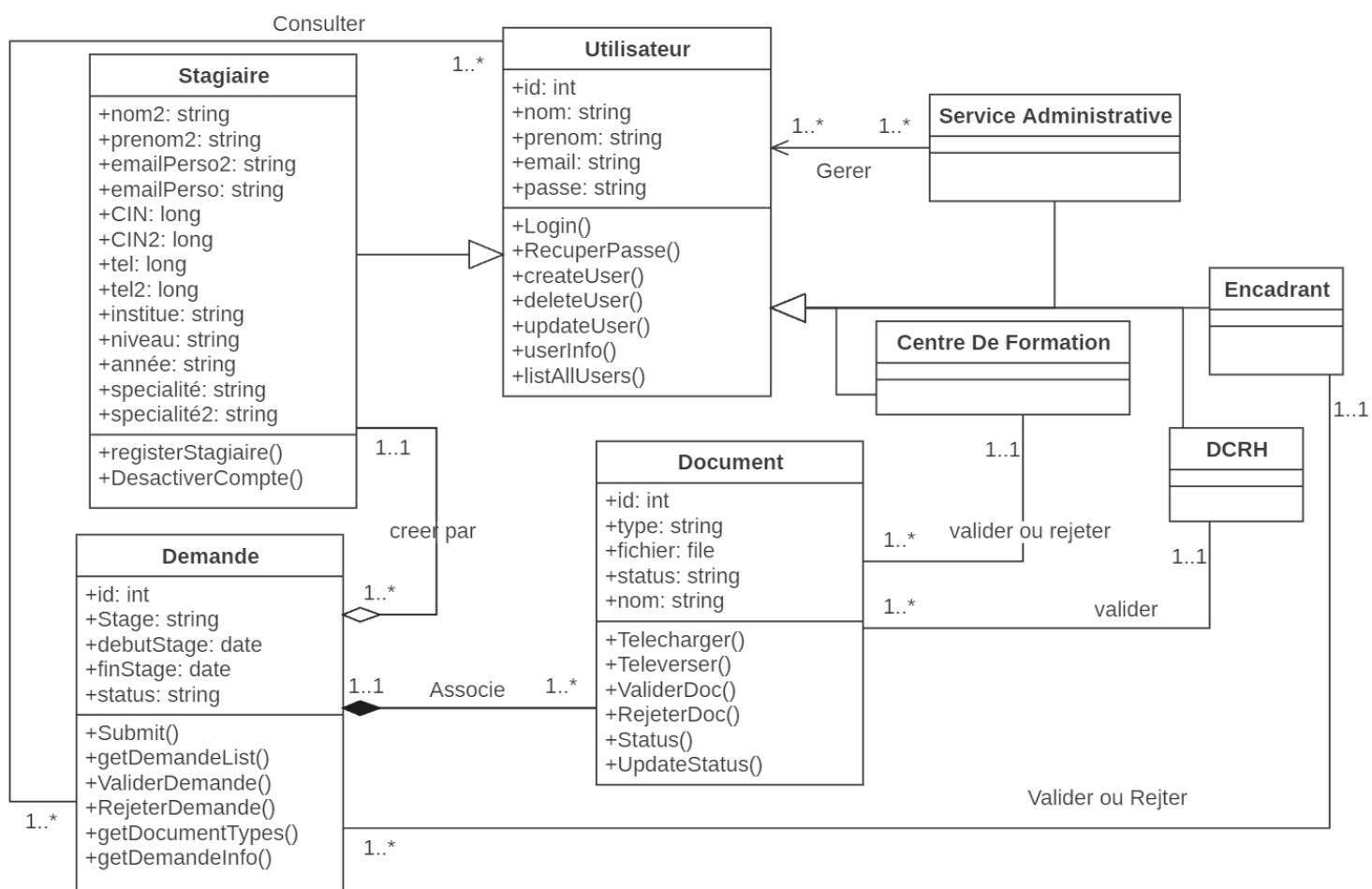


FIGURE 5: DIAGRAMME DE CLASS

II. Modélisation de la base de données

Demande (id, debut_stage, fin_stage, stage, status, encadrant_id, #stagiaire_id).

Document (id, created_at, fichier, name, status, type, centre_formation_id, demande_id)

Stagiaire (#user_id, annee, cin, cin2, email_perso, email_perso2, institut, niveau, nom2, prenom2, specialite, specialite2, tel, tel2)

Utilisateur (id, email, nom, passe, prenom, type)

User_Document_Seen (id, role, seen, #document_id, #utilisateur_id)

III. Diagramme de déploiement

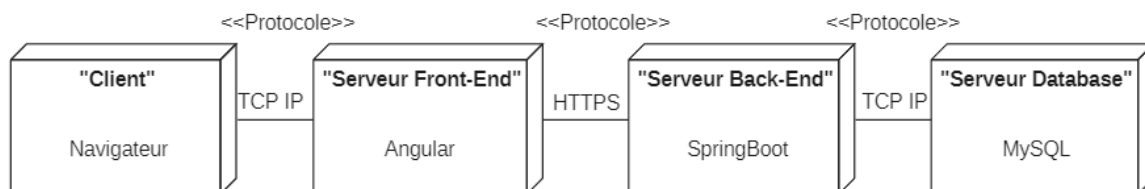


FIGURE 6: DIAGRAMME DE DEPLOIEMENT

Ce diagramme montre l'architecture du système et les interactions entre ses composants.

1. Client (Navigateur) : Envoie des requêtes HTTP/HTTPS au serveur front-end.
2. Serveur Front-End (Angular) : Gère l'interface et communique avec le back-end via HTTPS.
3. Serveur Back-End (Spring Boot) : Traite la logique métier et interagit avec la base de données MySQL via TCP/IP.
4. Base de Données (MySQL) : Stocke les données et exécute les requêtes SQL.
5. Flux de Données :
 - Client → Front-End (TCP/IP).
 - Front-End → Back-End (HTTPS).
 - Back-End ↔ Base de Données (TCP/IP).
 - Résultats renvoyés au client pour affichage.

La modélisation structurelle a permis de définir les entités principales et leurs relations, en assurant une gestion cohérente des données. Cette étape a été cruciale pour la conception de la base de données et la mise en œuvre des fonctionnalités du système.

Chapitre 4 : Réalisation

Ce chapitre décrit la mise en œuvre technique du projet, en présentant l'environnement de développement, les technologies utilisées et les principales interfaces graphiques. Il met en lumière les choix technologiques et les défis rencontrés lors de la réalisation.

I. Environnement de développement

1. Environnement matérielle

TABLEAU 6: DESCRIPTION MATÉRIELLE

Description matérielle	
Marque	MSI
Processeur	11th Gen Intel(R) Corei5-11400F
Disque dur	1536 GB
Mémoire vive	32,0 GB
Carte graphique	NVIDIA GeForce RTX 4060

2. Environnement logiciel

TABLEAU 7: DESCRIPTION LOGICIELLE

Description logicielle	
Système d'exploitation	Windows 11
Serveur base de données	MySQL
Outils de modélisation	StarUML
Outils de développement	Visual Studio code, Spring Tool Suite 4, IntelliJ
Maquettage	Figma
Langages	Java, HTML, CSS, typeScript
Technologies	Angular 19, Spring boot 3.4.2, Spring Security
Outils de Test	Postman

3. Présentation de l'environnement logiciel

MySQL

- **Définition** : MySQL est un système de gestion de base de données relationnelles basé sur SQL, développé par Oracle. Ce logiciel open source est écrit en C et C++ et a été testé avec une large gamme de compilateurs différents. MySQL stocke, gère et affiche des données dans des tableaux en fonctionnant comme un système client-serveur. [4]

- **Utilisation dans le projet** : Dans ce projet, MySQL est utilisé comme serveur de base de données pour gérer les informations de manière efficace et sécurisée.



FIGURE 7: LOG MYSQL

Angular 19

- **Définition** : Angular est un Framework de développement front-end open-source, basé sur TypeScript, et développé par Google. Il est utilisé pour créer des applications web dynamiques et réactives, notamment des applications monopages (SPA). [3]
- **Fonctionnalités clés utilisées** :
 - **Architecture basée sur les composants** : Chaque composant représente une unité autonome et réutilisable de l'interface utilisateur.
 - **Binding des données bidirectionnel** : Synchronisation automatique entre la vue et le modèle.
 - **Système de routage** : Gestion fluide de la navigation entre les différentes vues.
 - **Formulaires réactifs** : Création de formulaires interactifs et dynamiques.



FIGURE 8: LOGO ANGULAR 19

Spring Boot 3.4.2

- **Définition** : Spring Boot est un framework open-source moderne et puissant pour le développement d'applications Java. Il simplifie le processus de création, de configuration et de déploiement d'applications Java. [2]
- **Avantages** :
 - **Facilité de démarrage** : Configuration automatique rapide et facile.
 - **Embarquement de serveur** : Intègre un serveur d'application (Tomcat) directement dans l'application.

- **Architecture :**
 - **Modèle-Vue-Contrôleur (MVC)** : Structure claire et séparation des responsabilités.
 - **Repositories** : Interfaces fournissant des méthodes prédéfinies pour les opérations CRUD.
 - **Services** : Encapsulation de la logique métier complexe.
 - **Contrôleurs** : Gestion des requêtes HTTP et des réponses.



FIGURE 9: LOGO SPRINGBOOT

Spring Security

- **Définition** : Spring Security est un Framework puissant pour la sécurisation des applications web et des services REST. Il fournit des fonctionnalités robustes pour gérer l'authentification et l'autorisation . [2]
- **Caractéristiques :**
 - **Authentification flexible** : Supporte divers mécanismes, y compris JWT.
 - **Autorisation granulaire** : Définition de règles au niveau des méthodes, des URL et des objets métiers.
 - **Protection contre les attaques** : Implémentation de mesures de sécurité avancées.

JSON Web Token (JWT)

- **Définition** : Un JSON Web Token (JWT) est un Access token qui permet un échange sécurisé de données entre deux parties. Il contient toutes les informations importantes sur une entité, rendant la consultation d'une base de données superflue. [5]
- **Structure :**
 - **En-tête** : Informations essentielles sur le type de Token et l'algorithme de signature.
 - **Charge utile** : Données transmises à l'application, définissant les standards nécessaires.
 - **Signature** : Vérification de l'intégrité et authentification de l'expéditeur.

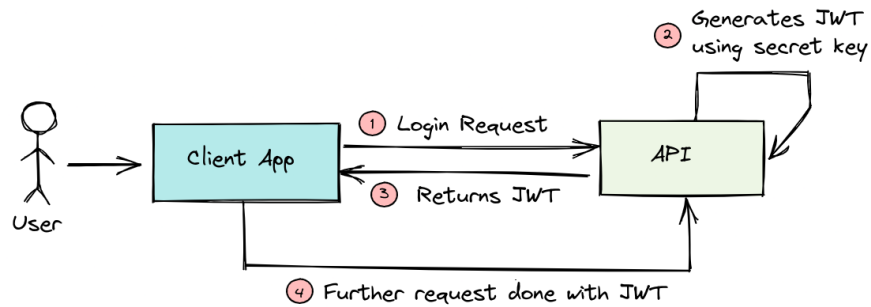


FIGURE 10: REQUEST D'AUTHENTIFICATION SIMPLE

StarUML

- Description** :
 StarUML est un outil de modélisation logicielle basé sur UML (Unified Modeling Language). Il permet de créer rapidement et facilement des modèles logiciels précis conformes aux normes UML. Cet outil a été utilisé pour concevoir et visualiser l'architecture du projet.



FIGURE 11: LOGO STARTUML

Visual Studio Code, Spring Tool Suite 4, IntelliJ

- Visual Studio Code** : Éditeur de code léger mais puissant, offrant une prise en charge étendue des extensions pour les langages tels que HTML, CSS, TypeScript, et Java.
- Spring Tool Suite 4 (STS)** : IDE spécialement conçu pour le développement avec Spring, offrant des fonctionnalités avancées comme le code completion intelligent pour les éléments Spring. Utilisé ici pour le développement backend avec Spring Boot.
- IntelliJ IDEA** : Environnement de développement intégré (IDE) populaire pour Java, offrant des fonctionnalités avancées telles que la gestion des dépendances Maven et la résolution des erreurs en temps réel.



FIGURE 12: LOGO VS CODE



FIGURE 13: LOGO STS



FIGURE 14: LOGO INTELIJ

Figma

- **Description** :
Figma est un outil de conception d'interface utilisateur collaboratif basé sur le cloud. Il a été utilisé pour créer des maquettes interactives de l'application avant le développement, permettant de visualiser et tester l'expérience utilisateur finale.



FIGURE 15: LOGO FIGMA

Java, HTML, CSS, TypeScript

- **Java** : Utilisé principalement pour le développement backend avec Spring Boot, Java assure une performance élevée et une grande flexibilité.
- **HTML & CSS** : Ces langages sont essentiels pour structurer et styliser l'interface utilisateur côté frontend.
- **TypeScript** : Langage principal pour le développement frontend avec Angular, offrant des fonctionnalités modernes comme le typage statique pour améliorer la qualité du code



FIGURE 16: LOGO HTML & CSS



FIGURE 17: LOGO TS



FIGURE 18: LOGO JAVA

Postman

- **Description** :
Postman est un outil puissant pour tester les API REST. Il a été utilisé pour vérifier le bon fonctionnement des endpoints de l'application backend, en envoyant des requêtes HTTP et en validant les réponses reçues.



FIGURE 19: LOGO POSTMAN

II. Gestion de l'Authentification avec JWT

Dans le cadre de notre projet, la gestion de l'authentification et la génération de JWT (JSON Web Token) sont centrales pour sécuriser les accès aux différentes ressources de l'application. Voici une explication détaillée de ce processus dans le contexte de notre projet.

1. Création du JWT à partir d'une Authentication

Lorsqu'un utilisateur se connecte à l'application via la page de login, Spring Security utilise les informations fournies (email et mot de passe) pour créer un objet de type Authentication. Cet objet :

- **Est stocké dans le Security Context** jusqu'à la fin de la requête.
- **Est utilisé pour générer un JWT** qui sera renvoyé à l'utilisateur pour les requêtes suivantes.

Dans notre projet, le JWT est généré à partir de l'email de l'utilisateur. Ce JWT contient les informations suivantes :

- **Sujet (Subject)** : L'email de l'utilisateur.
- **Date d'émission (Issued At)** : La date à laquelle le JWT a été créé.
- **Date d'expiration (Expiration)** : La date à laquelle le JWT expire (dans notre cas, 24 heures après sa création).
- **Signature** : Le JWT est signé avec l'algorithme HS512 et une clé secrète, qui est récupérée à partir de la configuration de l'application (jwt.secret).

Le JWT est ensuite renvoyé dans la réponse à la requête de connexion, permettant à l'utilisateur de l'utiliser pour les requêtes suivantes.

2. Envoi du JWT avec chaque requête

Une fois le JWT généré et récupéré par l'utilisateur, il doit être transmis à Spring Security avec chaque nouvelle requête pour prouver l'identité de l'utilisateur. Typiquement, le JWT est ajouté aux requêtes dans un header nommé Authorization. Par convention, sa valeur contient la chaîne Bearer suivie du JWT crypté.

3. Schéma de l'authentification et génération du JWT

Le processus de connexion, de création du JWT et de validation des requêtes peut être résumé dans le schéma suivant :

1. **Connexion** : L'utilisateur envoie ses identifiants (email et mot de passe) via une requête POST à l'endpoint /api/auth/login.
2. **Vérification des identifiants** : Le serveur vérifie les identifiants dans la base de données.
3. **Génération du JWT** : Si les identifiants sont valides, un JWT est généré et renvoyé à l'utilisateur.
4. **Validation du JWT** : Pour chaque requête suivante, le JWT est validé par le serveur pour s'assurer que l'utilisateur est authentifié.

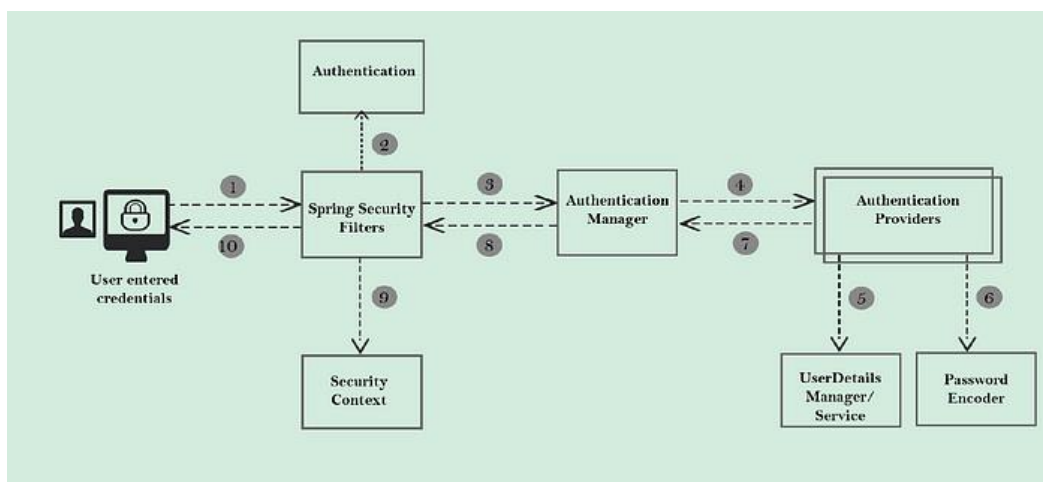


FIGURE 20:SCHEMA DE L'AUTHENTIFICATION

4. Intégration avec Spring Security

Spring Security s'intègre harmonieusement avec cette structure en permettant la configuration de l'authentification et de l'autorisation basées sur les rôles des utilisateurs. Voici comment cela fonctionne dans notre projet :

- **Authentification** : Lors de la connexion, Spring Security vérifie les identifiants de l'utilisateur ou du stagiaire et génère un JWT si les identifiants sont valides.
- **Autorisation** : Pour chaque requête, Spring Security valide le JWT et vérifie que l'utilisateur a les droits nécessaires pour accéder à la ressource demandée.

6. Exemple de code pour la génération du JWT

Voici un extrait de code de votre projet qui illustre la génération du JWT :

```
public String generateToken(String utilisateur) {  
    return Jwts.builder()  
        .setSubject(utilisateur)  
        .setIssuedAt(new Date())  
        .setExpiration(new Date(System.currentTimeMillis() + 86400000))  
        .signWith(secretKey)  
        .compact();  
}
```

Ce code montre comment un JWT est généré avec l'email de l'utilisateur comme sujet, la date d'émission, la date d'expiration, et signé avec la clé secrète.

III. Principales interfaces graphiques

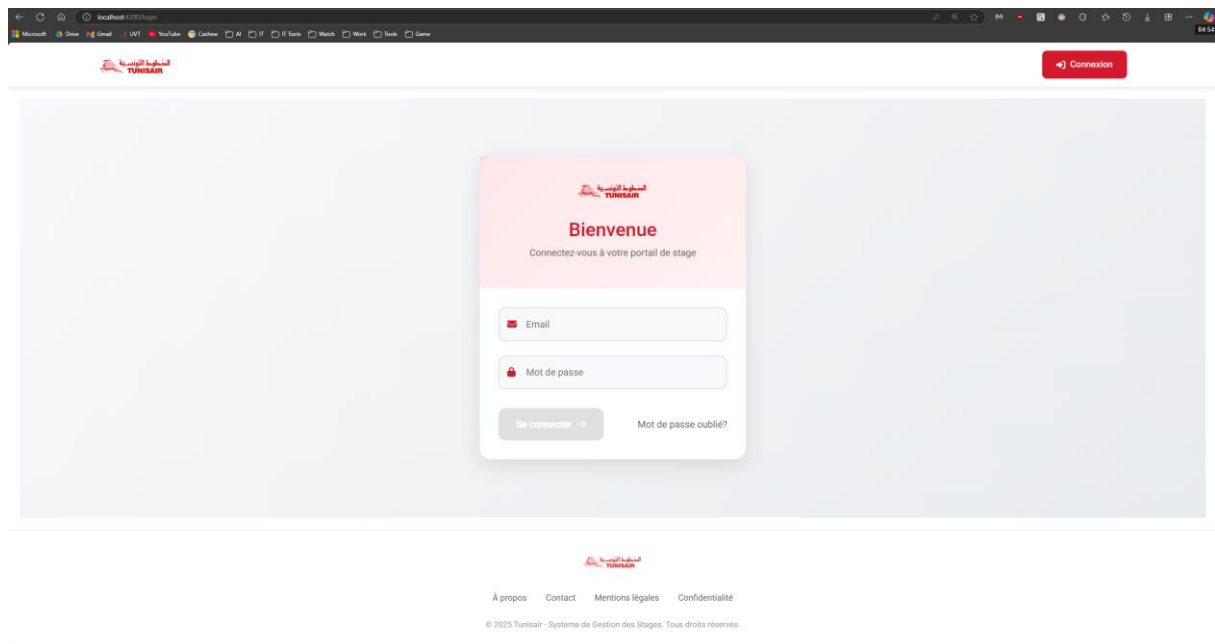


FIGURE 21: INTERFACE D'AUTHENTIFICATION

Cette interface permet aux utilisateurs de se connecter à la plateforme. Une option de récupération de mot de passe est également disponible pour les utilisateurs ayant oublié leurs identifiants.

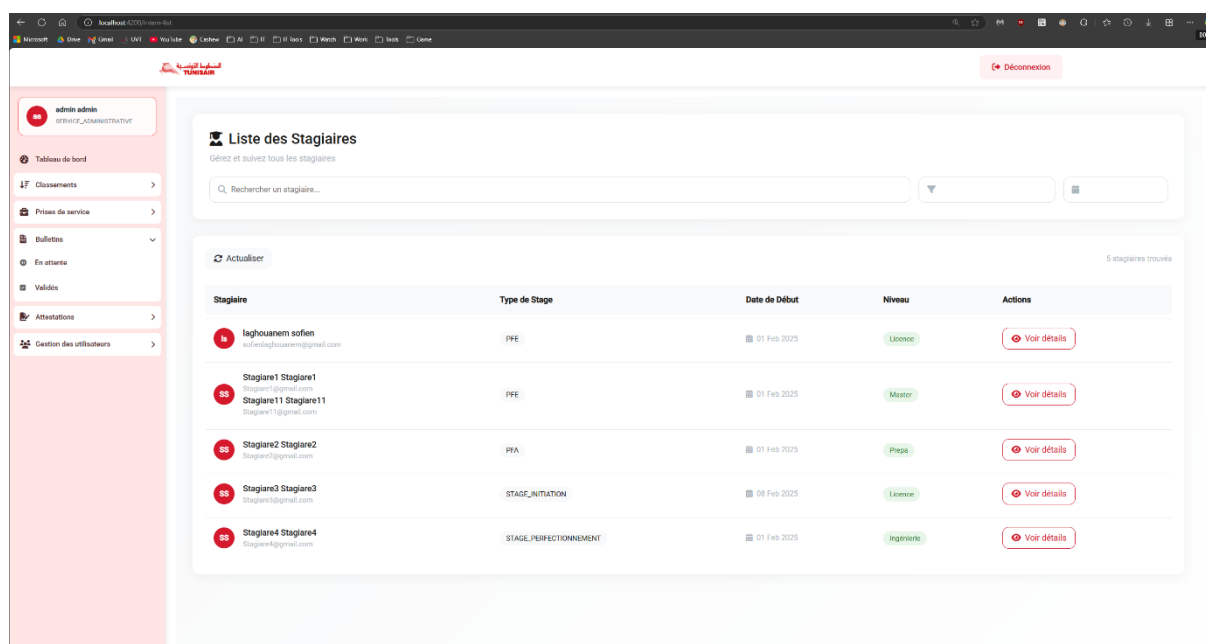


FIGURE 22: INTERFACE DU LISTE DES DEMANDES

Cette interface est destinée aux administrateurs pour visualiser les demandes de stage en attente de validation. Elle affiche une liste des demandes avec des informations clés (nom du stagiaire, type de stage, date de soumission) et permet de consulter les détails de chaque demande.

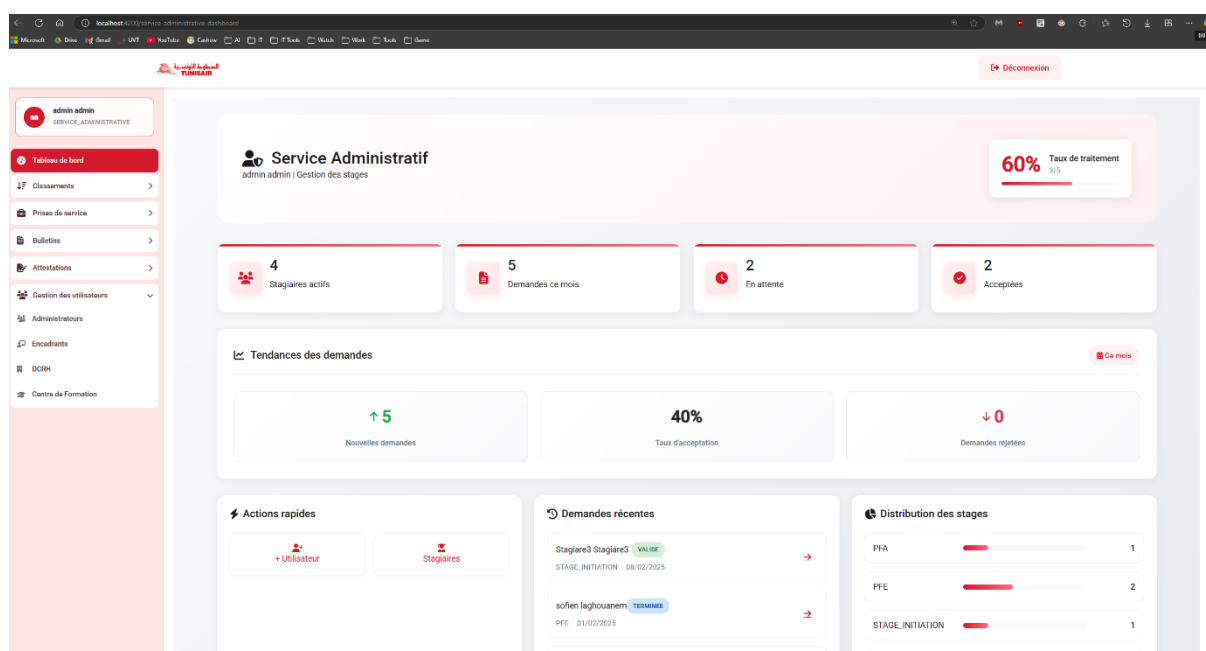


FIGURE 23: INTERFACE DU TABLEAU DE BORD DU SERVICE ADMINISTRATIVE

Cette interface offre une vue d'ensemble des demandes de stage validées et en attente de traitement, ... par le service administratif. Elle permet de télécharger les documents de suivre l'état d'avancement des dossiers.

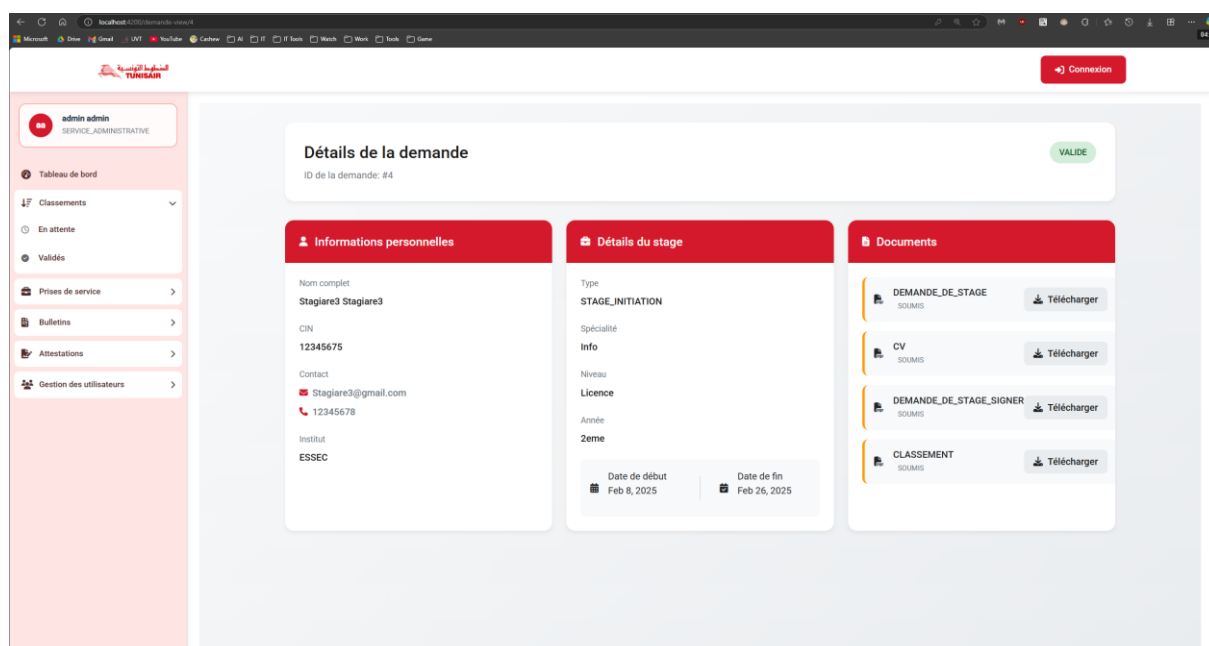


FIGURE 24: INTERFACE D'UNE DEMANDE D'UN STAGIAIRE

Cette interface permet à l'utilisateur de consulter les détails de sa demande de stage, y compris les informations personnelles, les détails du stage, les documents associés (téléchargeables ou à téléverser), et le statut actuel de la demande. Elle offre une vue claire et intuitive pour suivre l'évolution de la demande.

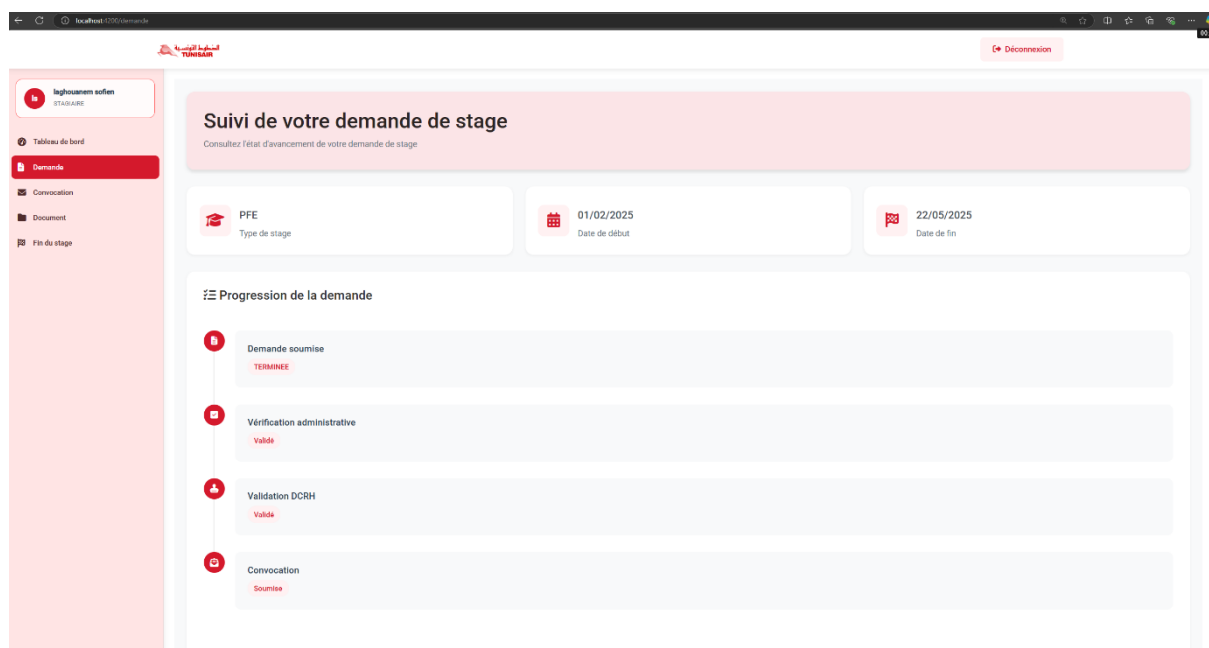


FIGURE 25: INTERFACE DU SUIVIE DU STAGIAIRE

Cette interface permet aux stagiaires de suivre l'état de leur demande de stage. Elle affiche les étapes du processus (soumission, validation, convocation, etc.) et permet de télécharger les documents requis ou de téléverser les documents signés.

localhost:4200/intern-form

الخطوط التونسية
TUNISAIR

Connexion

Demande de Stage

Bienvenue sur le portail de demande de stage Tunisair

1 Informations Personnelles — 2 Informations sur l'Institut — 3 Détails du Stage — 4 Documents

☐ Demande en Binôme

Prénom*

Nom*

CIN*

Email Personnel*

Numéro de Téléphone*

Suivant

الخطوط التونسية
TUNISAIR

[À propos](#) [Contact](#) [Mentions légales](#) [Confidentialité](#)

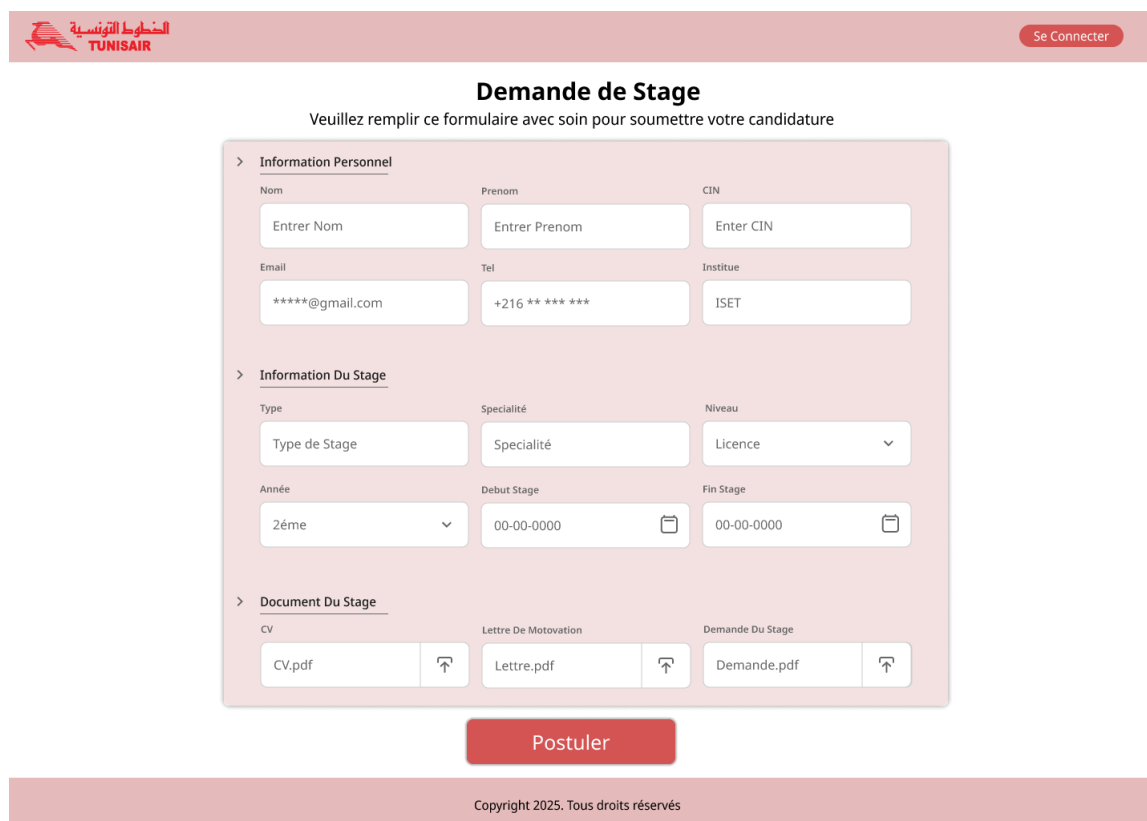
© 2025 Tunisair - Système de Gestion des Stages. Tous droits réservés.

FIGURE 26: INTERFACE DU DEMANDE DE STAGE

Cette interface permet aux stagiaires de soumettre une demande de stage en. Elle comprend un formulaire avec des champs pour les informations personnelles (nom, prénom, CIN, email, etc.), les détails du stage et institue (type, dates, spécialité), et une section pour téléverser les documents requis (CV, lettre de motivation, Demande de stage.).

IV. Maquette d'interface

Voici les maquettes d'interfaces conçues pour notre projet, offrant un aperçu visuel complet et précis de l'expérience utilisateur. Ces maquettes ont été réalisées avec Figma, un outil de conception collaboratif qui permet une création fluide, un accès facile depuis n'importe quel appareil, et la gestion des commentaires et des versions.



الخطوط التونسية
TUNISAIR

Se Connecter

Demande de Stage

Veuillez remplir ce formulaire avec soin pour soumettre votre candidature

> Information Personnel

Nom: Entrer Nom

Prenom: Entrer Prenom

CIN: Enter CIN

Email: *****@gmail.com

Tel: +216 ** * ** *

Institute: ISET

> Information Du Stage

Type: Type de Stage

Specialité: Spécialité

Niveau: Licence

Année: 2ème

Debut Stage: 00-00-0000

Fin Stage: 00-00-0000

> Document Du Stage

CV: CV.pdf

Lettre De Motivation: Lettre.pdf

Demande Du Stage: Demande.pdf

Postuler

Copyright 2025. Tous droits réservés

FIGURE 27: MAQUETTE DU FORMULAIRE DE DEMANDE DE STAGE

Cette interface permet au stagiaire de soumettre une demande de stage en ligne. Le formulaire est conçu pour être simple et intuitif.

FIGURE 28: MAQUETTE AUTHENTIFICATION

Cette interface permet aux utilisateurs (stagiaires, encadrants, service administratif, etc.) de se connecter à la plateforme.

Stagiaire	Stage	Date Debut	Niveau	Action
Stagiaire X	Initition	00-00-000	Licence	Info
Stagiaire X	Initition	00-00-000	Licence	Info
Stagiaire X	Initition	00-00-000	Licence	Info
Stagiaire X	Perfectionnement	00-00-000	Licence	Info
Stagiaire X	Perfectionnement	00-00-000	Licence	Info
Stagiaire X	PFA	00-00-000	Prepa	Info
Stagiaire X	PFE	00-00-000	Licence	Info
Stagiaire X	PFE	00-00-000	Master	Info
Stagiaire X	PFE	00-00-000	Master	Info

FIGURE 29: MAQUETTE LISTE DES DEMANDES EN ATTENTE

Cette interface est destinée à l'Encadrant pour visualiser et gérer les demandes de stage en attente de validation.

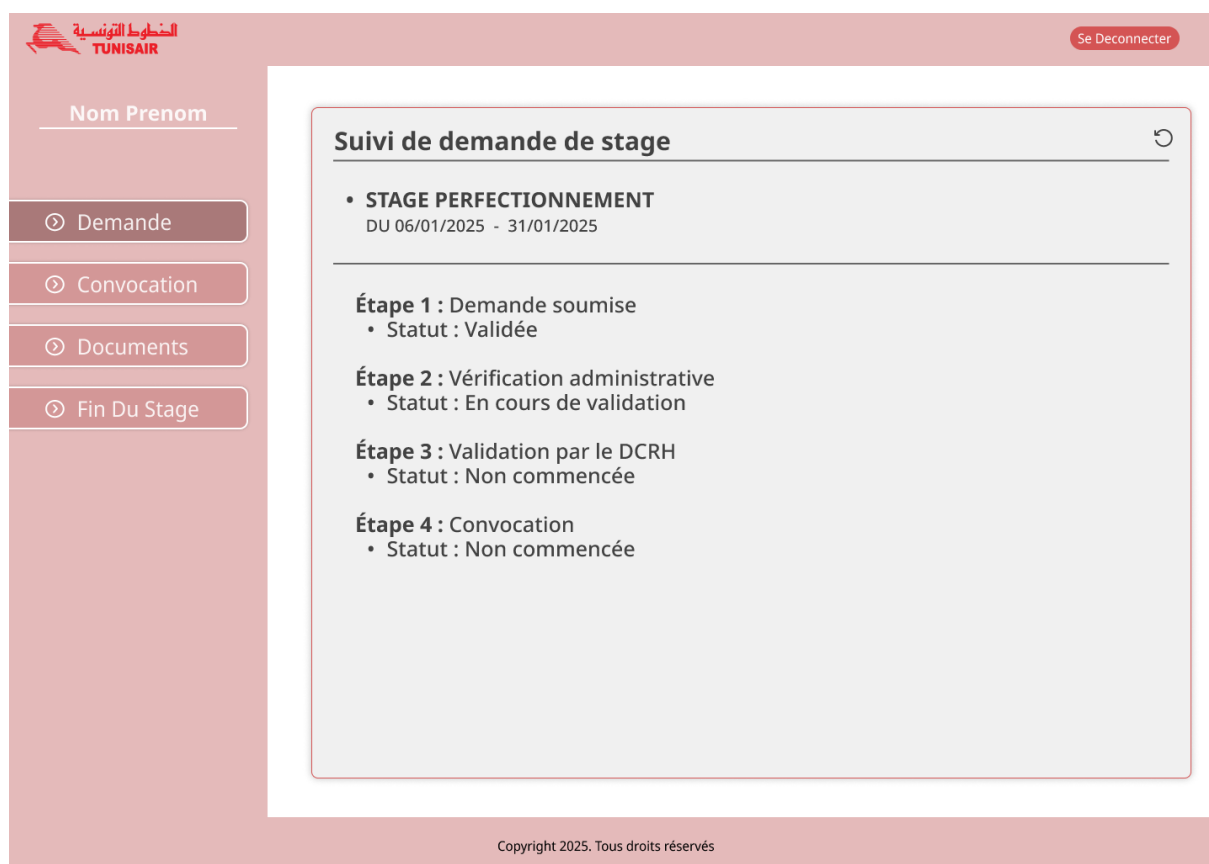


FIGURE 30: MAQUETTE DU TABLEAU DE BORD DU STAGIAIRE

Cette interface est le tableau de bord principal du stagiaire, lui permettant de suivre l'état de sa demande de stage, de consulter les documents requis, et d'interagir avec les différentes parties prenantes

La réalisation du projet a permis de concrétiser les modèles conceptuels en une application fonctionnelle. Les technologies choisies, telles que Spring Boot et Angular, ont offert une base robuste pour le développement, tout en garantissant une expérience utilisateur optimale.

Conclusion générale

En conclusion, ce rapport a détaillé le processus de développement du projet d'automatisation du processus de demande de stage. Le principal objectif était de concevoir une solution qui résout les problèmes liés aux retards, aux inefficacités et à la lourdeur administrative du système actuel. À travers les différentes sections du rapport, nous avons présenté l'analyse des besoins, la conception de la base de données, ainsi que la structure et les fonctionnalités du système développé.

Le modèle de base de données a été soigneusement conçu pour refléter les différentes entités impliquées dans le processus de demande de stage et de gestion des utilisateurs. Les choix technologiques et les outils utilisés ont permis d'assurer la cohérence et la fiabilité de l'application. La solution backend, basée sur Spring Boot, a été construite pour offrir une gestion optimale des demandes et des utilisateurs, tandis que le frontend Angular permet une expérience utilisateur fluide et intuitive.

Bien que le projet ait permis de répondre aux objectifs fixés, certaines améliorations sont possibles, notamment en termes d'optimisation de l'interface utilisateur pour la rendre encore plus conviviale. De plus, la gestion des erreurs et la validation des données pourraient être renforcées afin d'assurer une plus grande résilience du système.

Ce projet représente une avancée importante dans la digitalisation des processus internes et dans l'optimisation des interactions entre les stagiaires, les encadrants et les services administratifs. En poursuivant son développement, il est possible d'élargir ses fonctionnalités et de l'adapter à d'autres processus internes, offrant ainsi une meilleure efficacité et une gestion plus fluide des demandes de stage.

Nétographie

- [1] Tunisair Accès à l'information (tunisair.com)
- [2] Spring Boot Documentation (spring.io/projects)
- [3] Angular Documentation (angular.io/docs)
- [4] MySQL Documentation (dev.mysql.com/doc)
- [5] JSON Web Tokens (JWT) Introduction (jwt.io/introduction)

Annexe

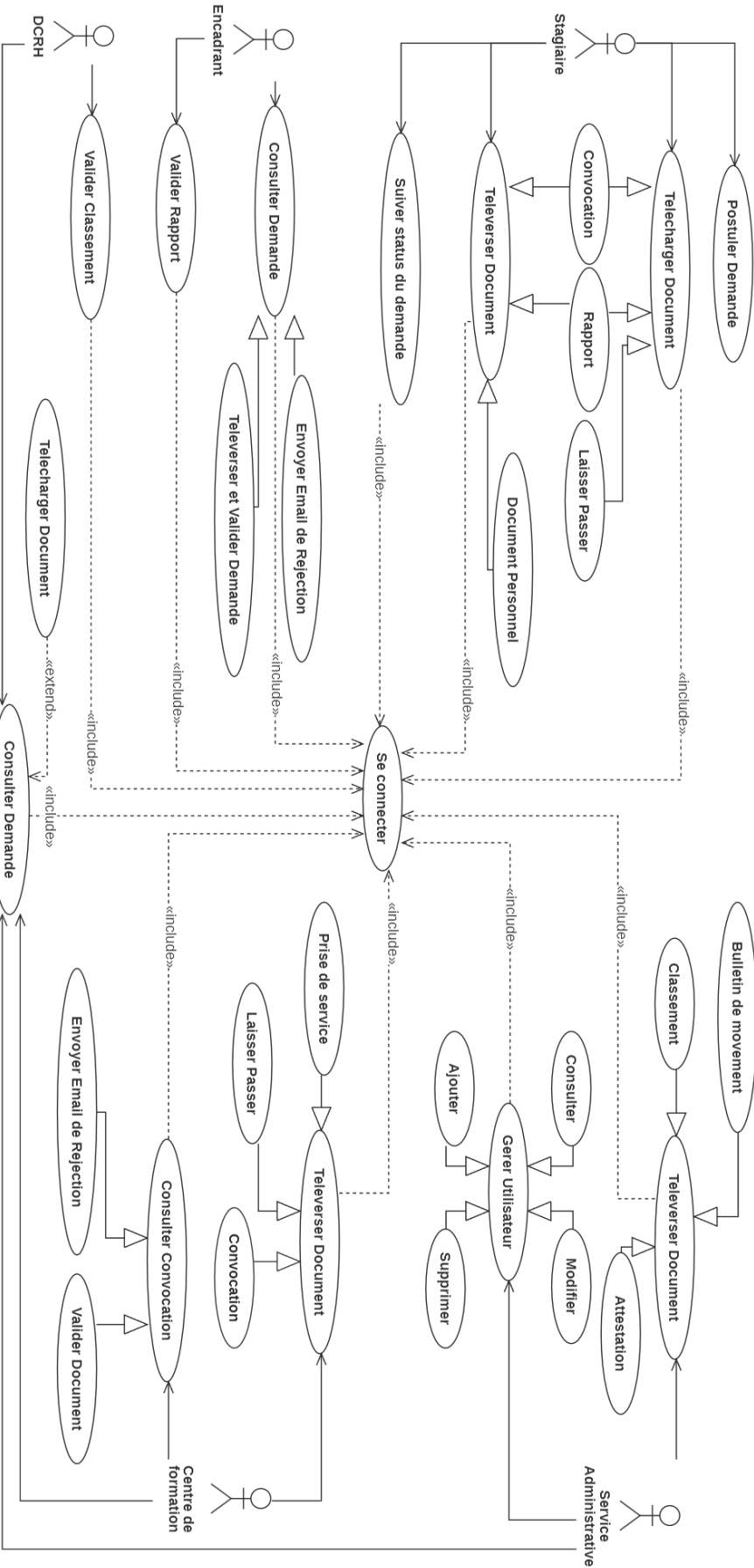


FIGURE 31: DIAGRAMME DE CAS D'UTILISATION COMPLET