|  |  |  |
| --- | --- | --- |
|  | **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** | Institut Supérieur des Etudes Technologiques de Charguia ... |

**RAPPORT**

**De**

**Stage de Perfectionnement**

**Parcours : Développement des Systèmes d’Information**

**Sujet :**

|  |
| --- |
| **CIRCUIT DE LA DEMADE DE STAGE** |

Elaboré par

**Sofien LAGHOUANEM**

**Encadré par :**

Mr. Ramzi BOUAZIZ

**Société d’accueil :** Tunisair

Remerciements

[Introduction générale 1](#_Toc188199997)

[Chapitre 1 : Présentation du cadre du stage 2](#_Toc188199998)

[I. Présentation de la société 2](#_Toc188199999)

[II. Présentation du domaine métier 2](#_Toc188200000)

[III. Etude de l’existant 2](#_Toc188200001)

[1. Description de l’existant 2](#_Toc188200002)

[2. Critique de l’existant 2](#_Toc188200003)

[3. Solution proposée 2](#_Toc188200004)

[IV. Cahier des charges 2](#_Toc188200005)

[4. Besoins fonctionnels 2](#_Toc188200006)

[5. Besoins non fonctionnels 2](#_Toc188200007)

[Chapitre 2 : Modélisation Comportementale 3](#_Toc188200008)

[I. Identification des Acteur 3](#_Toc188200009)

[II. Diagramme de cas d’utilisation 3](#_Toc188200010)

[1. Diagramme du cas d’utilisation général 3](#_Toc188200011)

[2. Description textuelle des cas d’utilisation 3](#_Toc188200012)

[III. Analyse des cas d’utilisation 3](#_Toc188200013)

[1. Diagramme de séquence objet 3](#_Toc188200014)

[2. Diagramme de séquence conception 3](#_Toc188200015)

[3. Diagramme de séquence système 3](#_Toc188200016)

[4. Diagramme d'activité 3](#_Toc188200017)

[Chapitre 3 : Modélisation Structurelle 4](#_Toc188200018)

[I. Diagramme de classes 4](#_Toc188200019)

[1. Descriptif du diagramme 4](#_Toc188200020)

[2. Représentation du diagramme 4](#_Toc188200021)

[II. Modélisation de la base de données 4](#_Toc188200022)

[III. Maquette d’interface 4](#_Toc188200023)

[Chapitre 4 : Réalisation 5](#_Toc188200024)

[I. Environnement de développement 5](#_Toc188200025)

[1. Environnement matérielle 5](#_Toc188200026)

[2. Environnement logiciel 5](#_Toc188200027)

[3. Présentation de l’environnement logiciel 5](#_Toc188200028)

[4. Diagramme de deploiement 5](#_Toc188200029)

[II. Gestion de l'Authentification avec JWT 5](#_Toc188200030)

[III. Principales interfaces graphiques 5](#_Toc188200031)

[Conclusion générale 6](#_Toc188200032)

[Nétographie 7](#_Toc188200033)

# 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.

## Présentation de la société

La société d’accueil pour ce projet est Tunisair, une entreprise de transport aérien de premier plan en Tunisie. Fondée en 1948, 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 techniquesactivement 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.

## 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, les entreprises peuvent optimiser leurs ressources tout en garantissant une meilleure traçabilité et transparence du processus.

## 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

### 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 avant de pouvoir poursuivre avec leur processus administratif.

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 la pour 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.

### 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.

### 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, RH, 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 et des notifications, ainsi que la génération et l’envoi automatique 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.

## Cahier des charges

### 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.

### 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 financières.

Utiliser des protocoles de cryptage et des techniques d’authentification avancées pour garantir la confidentialité des informations sensibles.

* **Contraintes techniques**

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

Veiller à ce que l’architecture de l’application soit extensible et évolutive pour répondre aux besoins futurs.

# Chapitre 2 : Modélisation Comportementale

## 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.

## Diagramme de cas d’utilisation

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

### Diagramme du cas d’utilisation général

A diagram of a document

Description automatically generatedLe 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 générale

### Description textuelle des cas d’utilisation

Les descriptions textuelles permettent de fournir une explication détaillée et narrative des interactions entre les acteurs (utilisateurs) 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** | * Le stagiaire dispose des documents nécessaires. * L’application est accessible et fonctionnelle. |
| **Scénario Nominal** | 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 et notifie l’Encadrant par email. |
| **Scénario Alternatif** | A1. Demande de Même Type Soumis :   1. Le système affiche un message d’erreur : « Une demande de ce type de stage existe déjà. ». 2. Le stagiaire remplie et soumet un autre type de stage. |
| **Postcondition** | * 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** | * L’Encadrant est authentifié. * Une demande est en attente de validation. |
| **Scénario Nominal** | 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 :   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** | * La demande est validée ou rejetée. |

Tableau 3: Description Textuelle Du Téléversement 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** | * L’acteur est authentifié. * La demande existe dans le système. |
| **Scénario Nominal**  (Stagiaire) | 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 :   1. Le Centre de Formation rejette le document. 2. Le système envoie un email au stagiaire avec les corrections nécessaires. |
| **Postcondition** | * 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** | * L’acteur est authentifié. * La demande existe dans le système. |
| **Scénario Nominal** | 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, documents (à télécharger). |
| **Scénario Alternatif** | A1. Demande introuvable :   1. Le système affiche un message : « Aucun stagiaire trouvé ». |
| **Postcondition** | * L’acteur a accès aux informations complètes de la demande. |

Tableau 5: Description Textuelle du Cas d'utilisation "Gérer 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** | * Le Service Administratif est authentifié. * Les droits d’administration sont activés. |
| **Scénario Nominal** | 1. Le Service Administratif accède à l’interface « Gestion des Utilisateurs ». 2. Le système affiche la liste des utilisateurs existants. 3. Le Service Administratif clique sur « Ajouter un Utilisateur ». 4. Le système affiche un formulaire (nom, prénom, email, mot de passe, rôle). 5. Après validation, le système génère un mot de passe temporaire et envoie un email. |
| **Scénario Alternatif** | A1. Email déjà existant :   1. Le système affiche un message : « Cet email déjà existe ». |
| **Postcondition** | * Le compte utilisateur est créé/modifié/supprimé. |

## Analyse des cas d’utilisation

### Diagramme de séquence objet

Ce diagramme détaille la séquence des interactions entre les objets du système pour un cas d’utilisation spécifique, tel que la soumission de demande de stage ou l’approbation d’une demande.

Exemple pour la soumission de la demande de stage :

* Le Stagiaire envoie la demande à l’application.
* Le Service Administratif vérifie et assigne l'encadrant approprié.
* L'Encadrant reçoit la demande et prend la décision (accepter ou rejeter).

### Diagramme de séquence système

Ce diagramme montre les interactions globales du système entre les différents acteurs et les composants du backend lors de l’exécution de l’application. Par exemple, lors de l'enregistrement d’un stagiaire ou de la validation d’une demande, il illustre comment les données sont transférées entre les différentes couches 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 maquettes.

## Diagramme de classes

### 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 (status). 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 (status). 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.

### A diagram of a computer program Description automatically generatedReprésentation du diagramme

Figure 2: Diagramme de class

## 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)

## Diagramme de déploiement

## Chapitre 4 : Réalisation

## Environnement de développement

### Environnement 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 |

### Environnement logiciel

|  |  |
| --- | --- |
| 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 |

### 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 **[2]**. MySQL stocke, gère et affiche des données dans des tableaux en fonctionnant comme un système client-serveur.
* **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.

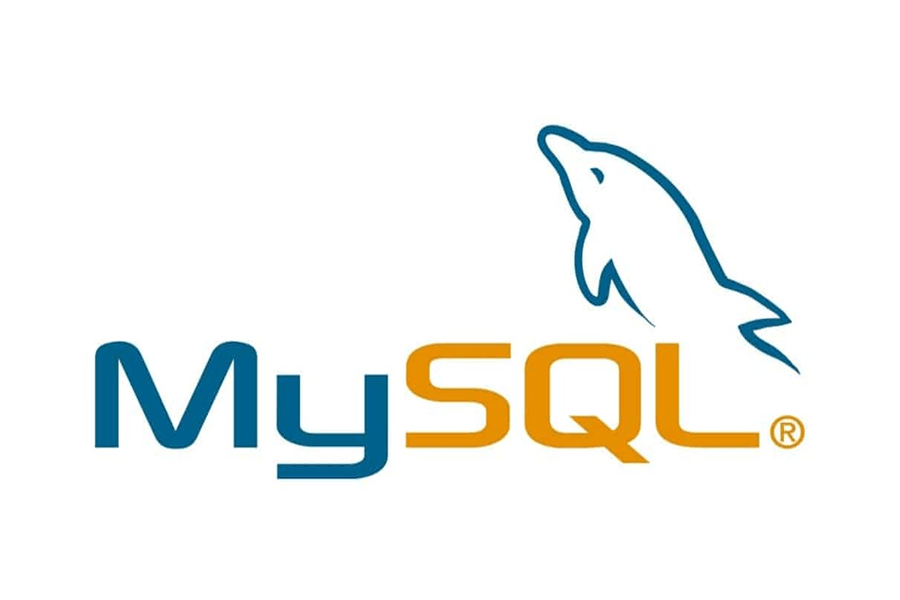


Figure 3: Log MySQL

**Angular 19**

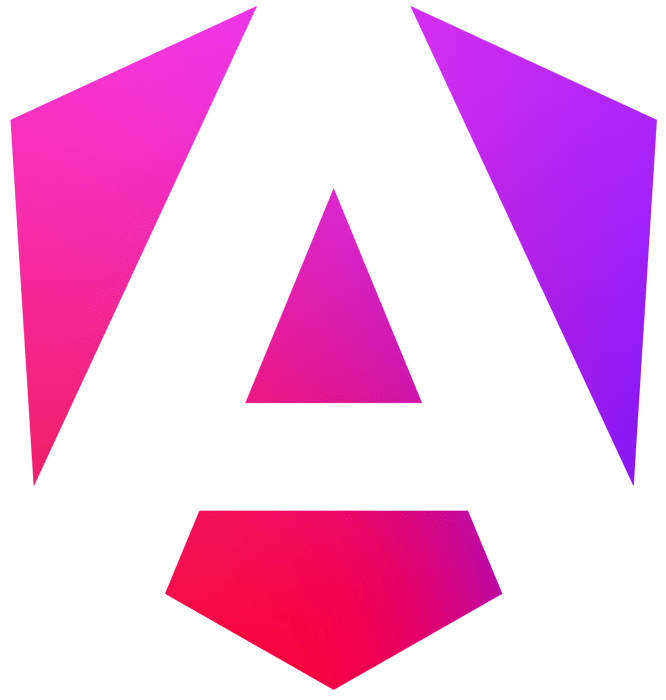
* **Définition** : Angular est un Framework de développement front-end open-source, basé sur TypeScript, et développé par Google **10**. Il est utilisé pour créer des applications web dynamiques et réactives, notamment des applications monopages (SPA).
* **Fonctionnalités clés utilisées** :
  + **Architecture 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 4: 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 **3**.
* **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 5: 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 **4**.
* **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.
* **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.

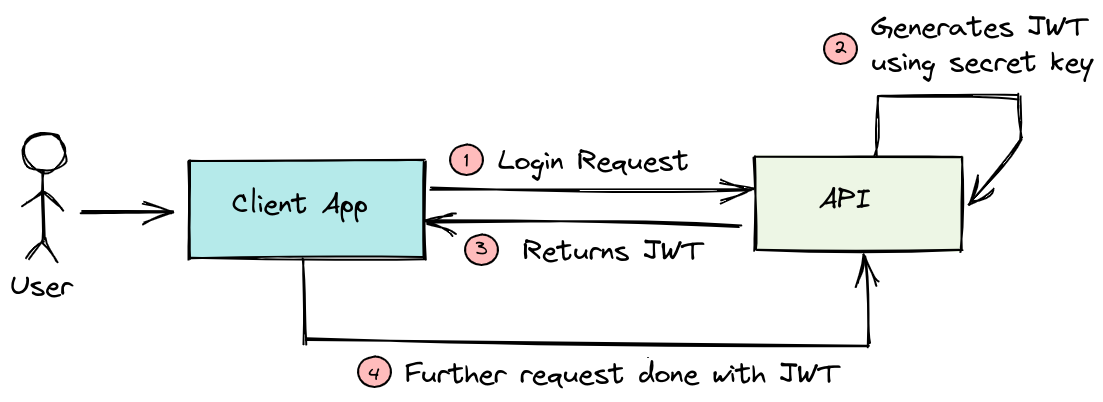


Figure 6: Request d'authentification simple

**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, facilitant la communication entre les membres de l'équipe.

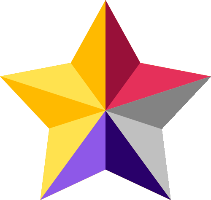


Figure 7: Logo StartUML

**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.

|  |  |  |
| --- | --- | --- |
| A blue logo with a black background  Description automatically generated  Figure 8: Logo VS Code | Spring | Tools  Figure 9: Logo STS | A logo with colorful triangles  Description automatically generated with medium confidence  Figure 10: Logo IntelliJ |

**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 11: 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 [[Exemple]].

**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 [[Exemple]].

## Gestion de l'Authentification avec JWT

## Principales interfaces graphiques

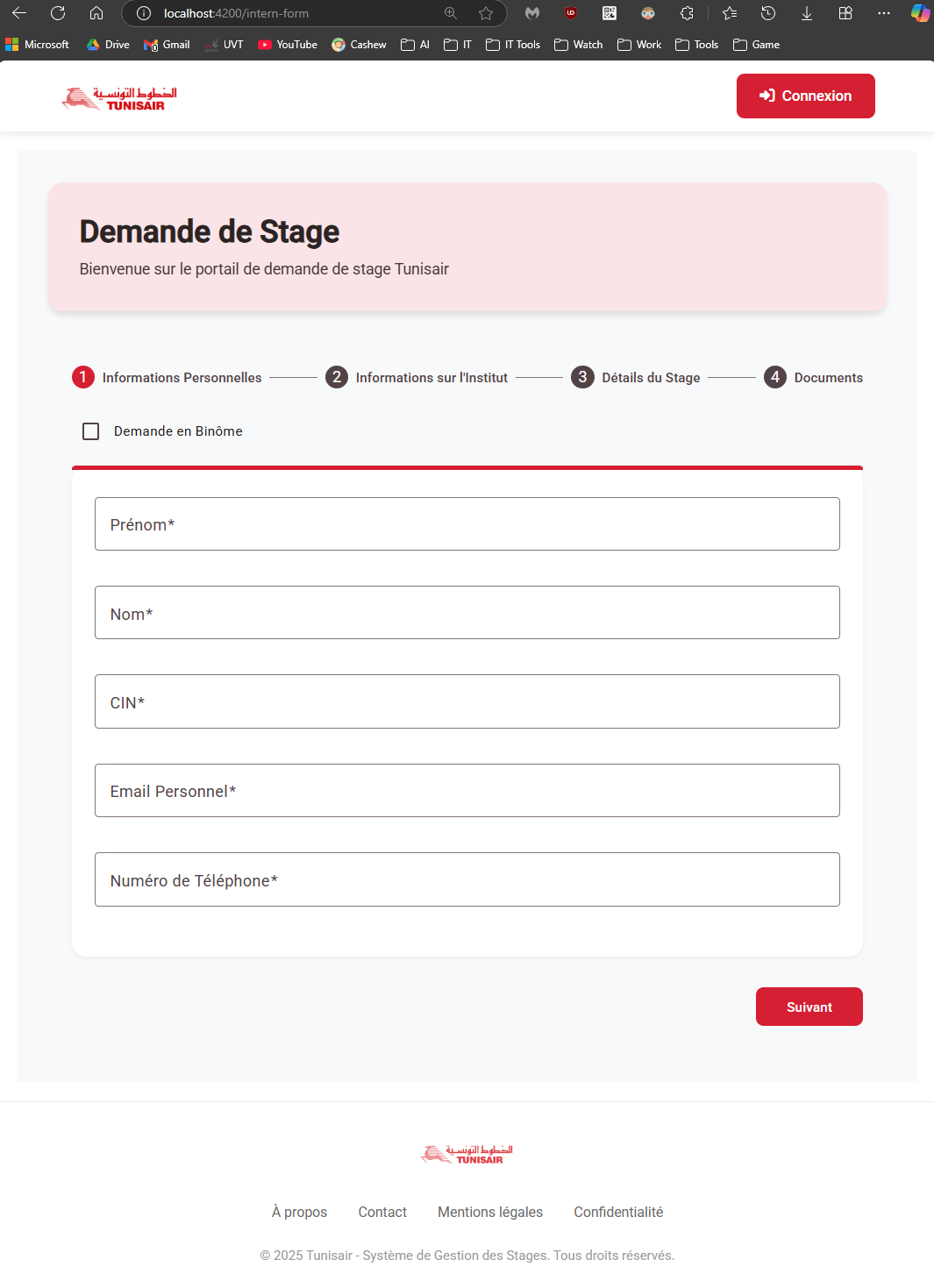


Figure 12: Interface du demande de stage

A screenshot of a computer

Description automatically generated

Figure 13: Interface d'Authentification

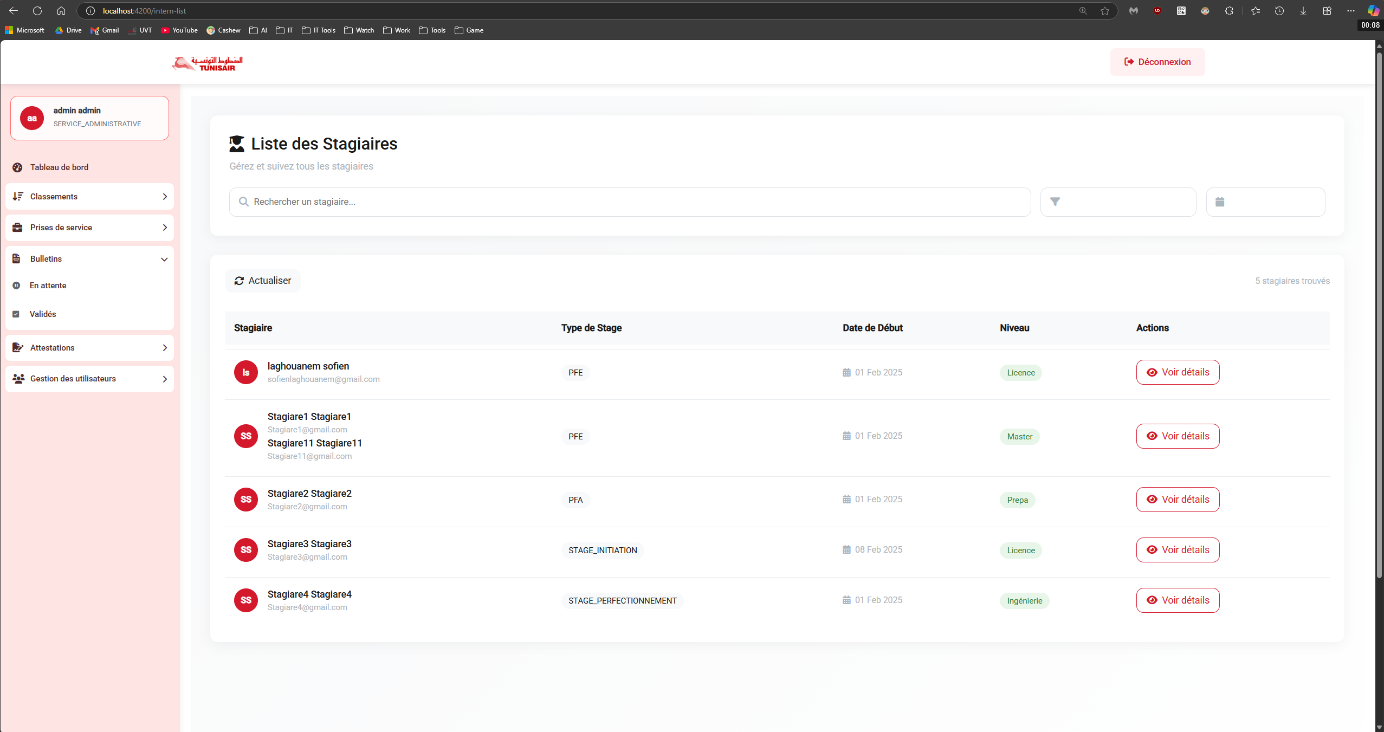


Figure 14: Interface du liste des demandes

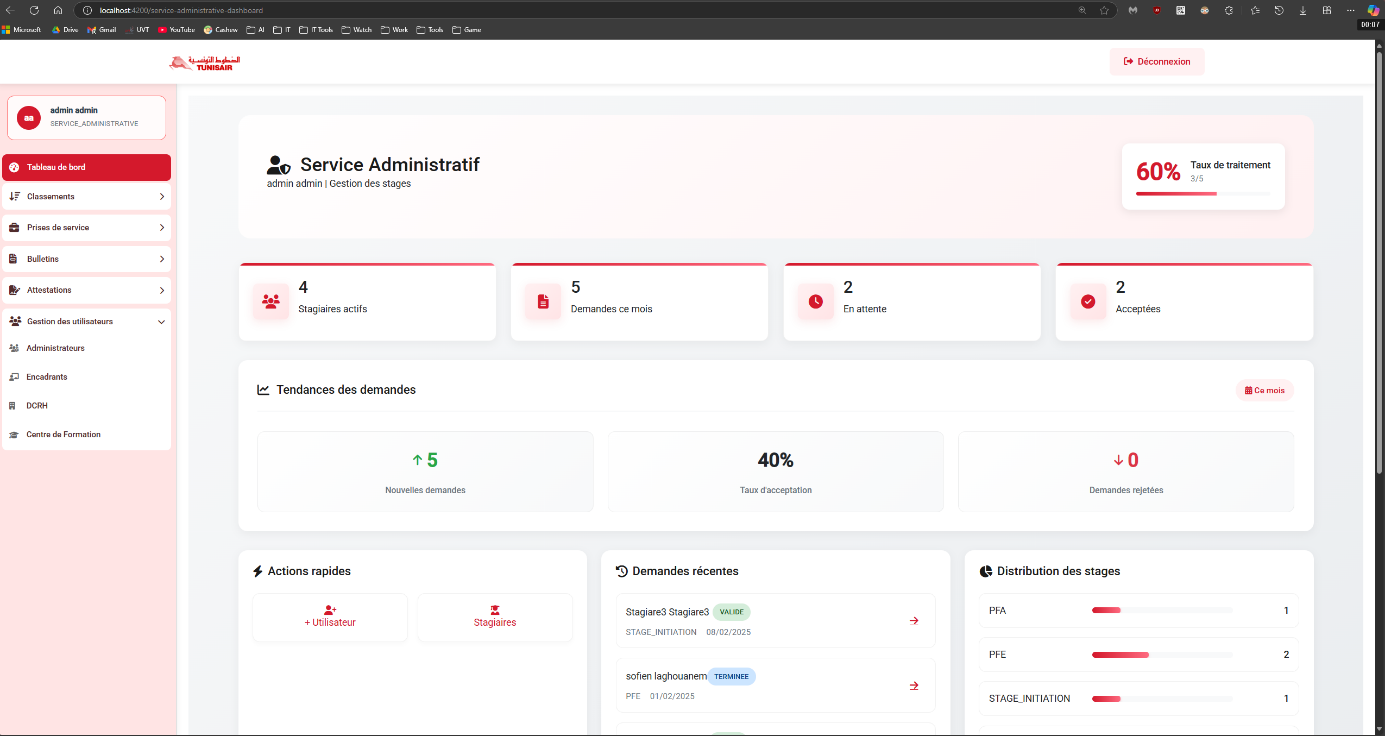


Figure 15: Interface du Tableau de bord du Service Administrative

A screenshot of a computer

Description automatically generated

Figure 16: Interface d'une demande d'un stagiaire

A screenshot of a computer

Description automatically generated

Figure 17: Interface du suivie du stagiaire

## 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.

A screenshot of a computer

Description automatically generated

Figure 18: 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.

A screenshot of a computer

Description automatically generated

Figure 19: Maquette authentication

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

A screenshot of a computer

Description automatically generated

Figure 20: 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.

A screenshot of a computer

Description automatically generated

Figure 21: Maquette du Table 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.

# 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