



Université de Tunis El Manar.
Ecole Nationale d'Ingénieurs de Tunis.
Département de Technologies d'Informations
et de Communications.



Projet de stage ouvrier

Développement d'une application de gestion de parc informatique

Ecadré par :

Mr Ben Mansour Ramzi

Realisé par :

Barhoumi Dhia Eddine

Realisé au sein de l'entreprise
TUNISIE TELECOM district du Kef



Durée du stage : du 01/06/2025 au 30/06/2025

Année universitaire 2024-2025

Remerciements

Je tiens à exprimer mes sincères gratitude à monsieur Mr ben Mansour Ramzi mon encadreur au sein de l'entreprise TUNISIE TELECOM district du Kef pour ses précieuses conseils et sa compréhensibilité. Je voudrais remercier aussi tous ceux qui n'ont épargné aucun effort et qui ont veillé par leur contribution à me fournir des informations utiles et l'ai nécessaire pour accomplir ce travail.

Table des matières

0.1	Cadre de stage :	5
0.2	Objectifs du stage	5
1	Présentation de l'entreprise	6
1.1	Tunisie Télécom	6
1.2	Organisation generale de L'entreprise :	7
1.3	Organisation regional de L'entreprise :	7
2	Cahier de charge	8
2.1	Objectifs du projet	8
2.1.1	Objectif général	8
2.1.2	Objectifs spécifiques	8
2.2	Fonctionnalités principales	8
2.3	Architecture technique	8
2.4	Exigences non fonctionnelles	9
2.5	Interface utilisateur	9
2.5.1	Page d'accueil	9
2.5.2	Template d'administration	9
2.6	Livrables	9
2.7	Planning estimatif	9
2.8	Contraintes	9
2.9	Annexes	9
3	Implémentation de l'application	10
3.1	Outils de developpement :	10
3.1.1	Frontend - Angular :	10
3.1.2	Backend – Spring Boot	11
3.1.3	Base de données – H2	11
3.2	Diagramme de classe :	12
3.3	Relations entre les classes	12
3.3.1	Classe Machine	12
3.3.2	Classe Utilise	13
3.3.3	Classe Installer	13
3.3.4	Classe Ecran	13
3.3.5	Autres relations	13
3.4	Description de l'application	13
3.4.1	Page d'accueil	13
3.4.2	Interface d'administration (admin-template)	14
3.4.3	Avantages	14

3.4.4	Diagramme de cas d'utilisation :	15
3.4.5	Liste des API REST :	16
3.5	Exemples de fonctionnalités :	16
3.5.1	Gestion des Logiciels	16
4	Conclusion générale	19

Table des figures

1.1	organosation de Telecome	7
1.2	organosation regionnale de Telecome	7
3.1	/h2-console	11
3.2	Digramme de classes	12
3.3	Vue d'accueil	14
3.4	diagramme de cas d'utilisation	15
3.5	Exemple de liste des API REST des machines	16
3.6	Liste des Logiciels	16
3.7	Interface ajouter Logiciel	17

Introduction

0.1 Cadre de stage :

Ce stage présente ma première occasion par laquelle j'ai effectué un contact avec une entreprise qui est la société national des télécommunications et dont le nom commercial est Tunisie Télécoms. C'est un stage d'ouvrier dont le but est de découvrir en premier lieu le mécanisme général du fonctionnement des entreprises, puis de s'approcher du monde du travail, de contacter en proximité les ouvriers pendant leurs tâche journalières et de voir les sections de base et les fonctions essentielles de cette entreprise .

0.2 Objectifs du stage

Ce stage m'a permis d'approfondir mes connaissances en développement full-stack, en particulier avec le framework backend Spring Boot et le framework frontend Angular. Les objectifs principaux étaient :

- Développer une interface web intuitive pour la gestion du matériel informatique
- Mettre en œuvre une architecture RESTful sécurisée
- Permettre l'ajout et la suppression et la consultation de machines
- Fournir une base de données fiable avec des données persistantes

Chapitre 1

Présentation de l'entreprise

1.1 Tunisie Télécom

Depuis sa création, Tunisie Télécom œuvre à consolider l'infrastructure des télécoms en Tunisie, à améliorer le taux de couverture et à renforcer sa compétitivité. Elle contribue également activement à la promotion de l'usage des TIC et au développement des sociétés innovantes dans le domaine des télécoms. Tunisie Télécom compte dans ses rangs plus de 6 millions abonnés dans la téléphonie fixe et mobile, en Tunisie et à l'étranger. Elle joue en outre un rôle important dans l'amélioration du taux de pénétration de l'Internet en Tunisie, ce qui lui permet d'atteindre le nombre de 140 mille abonnés à la fin du mois d'avril 2008. Tunisie Télécom se compose de 24 directions régionales, de 80 Actels et points de vente et de plus de 13 mille points de vente privés. Elle emploie plus de 8500 agents. Elle est chargée de :

- L'installation, l'entretien et l'exploitation des réseaux publics de télécommunications.
- L'offre de tous les services publics ou privés de télécommunications correspondant aux divers besoins à caractère social et économique.
- promotion des nouveaux services de télécommunication.
- La contribution au développement des études et recherches scientifiques liées au secteur de télécommunication.
- La participation à l'effort national d'enseignement supérieur en matière de télécommunication.
- L'application des conventions et des traités des organisations internationales et régionales spécialisées dans ce domaine.
- La promotion de la coopération à tous les niveaux dans le domaine des télécommunications.

1.2 Organisation generale de L'entreprise :

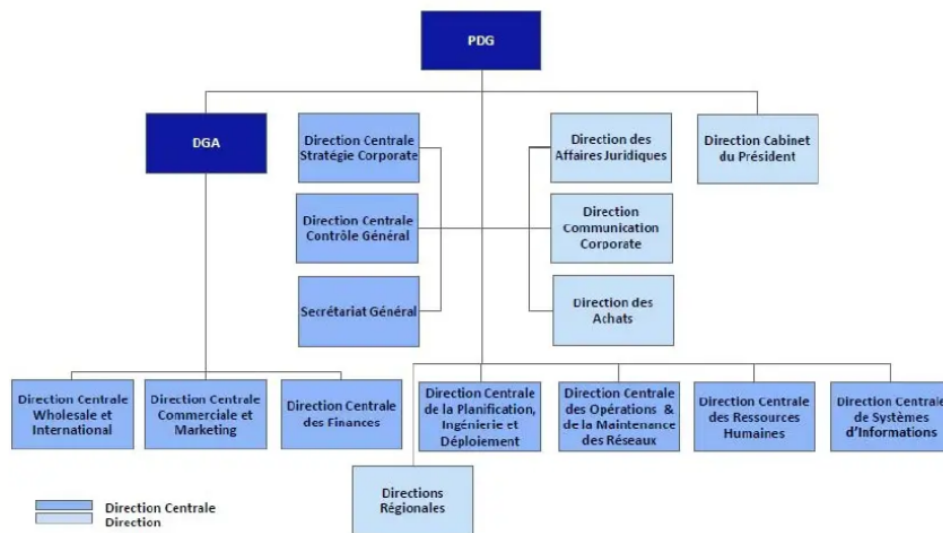


FIGURE 1.1 – organosation de Telecom

1.3 Organisation regional de L'entreprise :

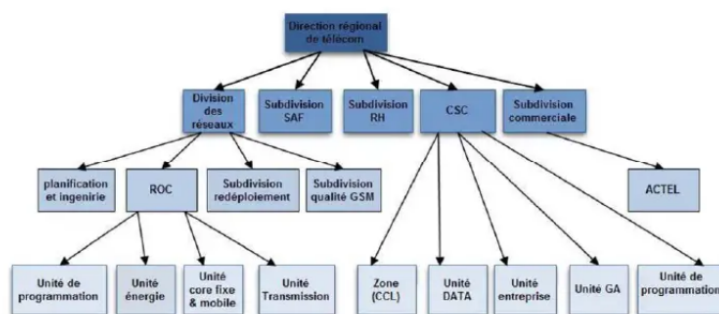


FIGURE 2: ORGANIGRAMME DE LA DIRECTION GENERAL

SAF : service administratif et financier
 RH : ressource humaine
 CSC : centre service clientèle.
 ROC : centre opérationnel régional
 ACTEL : Agence commerciale de télécommunication.
 CCL : centre de construction des lignes téléphoniques.
 GA : gestion d'abonné

FIGURE 1.2 – organosation regionnale de Telecom

Chapitre 2

Cahier de charge

2.1 Objectifs du projet

2.1.1 Objectif général

Concevoir une application web permettant la gestion complète des équipements informatiques d'une organisation.

2.1.2 Objectifs spécifiques

- Gestion des machines : ajout, suppression et suivi.
- Suivi des logiciels installés et de leurs licences.
- Gestion des utilisateurs internes et affectation du matériel.
- Localisation physique des équipements.
- Référencement des fournisseurs de matériel et de services.

2.2 Fonctionnalités principales

- **Machines** : suivi complet du matériel informatique.
- **Logiciels** : gestion des logiciels et des licences.
- **Fournisseurs** : base de données des fournisseurs.
- **Utilisateurs** : association du matériel aux employés.
- **Emplacements** : cartographie des ressources physiques.

2.3 Architecture technique

- **Front-end** : Angular 20, Angular Material.
- **Back-end** : Spring Boot 3.5.3.
- **Base de données** : H2 en développement, MySQL/PostgreSQL en production.
- **Langages** : TypeScript, Java, html, css.
- **Sécurité** : sessions sécurisées.

2.4 Exigences non fonctionnelles

- Interface responsive compatible mobile.
- Sécurité des données et des accès.
- Navigation fluide, chargement rapide.

2.5 Interface utilisateur

2.5.1 Page d'accueil

Affiche un message de bienvenue et des boutons vers les modules : Machines, Logiciels, Fournisseurs, Utilisateurs, Emplacements.

2.5.2 Template d'administration

- Barre de navigation supérieure avec identification de l'utilisateur.
- Menu latéral fixe pour la navigation.
- Contenu dynamique via `router-outlet`.

2.6 Livrables

- Code source (frontend et backend).
- Documentation technique.
- Scripts de base de données.
- Cahier de tests fonctionnels.
- Présentation finale du projet.

2.7 Planning estimatif

Étape	Durée
Analyse des besoins et Conception technique	1 semaine
Développement Angular et Spring Boot	2 à 3 semaines
Tests, intégration et livraison	1 semaine

2.8 Contraintes

- Respect des technologies imposées : Angular + Spring Boot.
- Déploiement sur serveur local ou cloud.
- Contrainte de temps liée à un encadrement académique ou professionnel.

2.9 Annexes

- Diagrammes UML (cas d'utilisation et classes).
- Captures d'écran de l'interface.
- Liste des API REST disponibles.

Chapitre 3

Implémentation de l'application

Cette partie détaille les outils et frameworks utilisés, ainsi que les choix techniques ayant guidé l'implémentation.

Le code source complet de l'application est disponible sur le dépôt GitHub suivant : [gestion de parc informatique](#)

3.1 Outils de developpement :

Cette application est construite avec les outils suivants :

3.1.1 Frontend - Angular :

- **Angular v20**
Framework SPA basé sur TypeScript, utilisé pour sa structure modulaire, son CLI et ses performances.
- **TypeScript**
Langage de programmation principal avec typage statique, facilitant la lisibilité et la maintenabilité.
- **HTML / SCSS**
Utilisés pour le markup et le style des composants.
- **Angular Material** (@angular/material)
Bibliothèque de composants UI suivant les normes Material Design.
- **Angular CDK** (@angular/cdk)
Utilisé pour des fonctionnalités bas niveau comme les overlays, le drag drop, et l'accessibilité.
- **RxJS v7.8**
Bibliothèque de programmation réactive utilisée pour gérer les flux de données (Observables) entre composants, services et APIs.
- **Reactive Forms** (@angular/forms)
Utilisé pour la gestion réactive des formulaires avec validations synchrones et asynchrones.
- **Angular CLI**
Outil de ligne de commande pour créer, développer, tester et builder l'application.
- **npm / tslib**
Gestion des dépendances et bibliothèque d'exécution pour TypeScript.

3.1.2 Backend – Spring Boot

Le backend de cette application est développé avec les technologies suivantes :

— Spring Boot 3.5.3

Framework Java moderne pour le développement rapide d'applications web et de services RESTful.

— Java 21

Langage de programmation principal utilisé, version à jour et performante.

— Maven

Outil de gestion de projet et de dépendances, utilisé ici pour la compilation, les tests et le packaging.

- `spring-boot-starter-web`

Fournit les composants nécessaires à la création d'applications web, incluant Tomcat embarqué, Spring MVC, etc.

- `spring-boot-starter-data-jpa`

Fournit une intégration simplifiée avec JPA (Hibernate), facilitant les opérations sur les bases de données relationnelles.

— Lombok

Bibliothèque utilisée pour réduire le code boilerplate Java (getters, setters, constructeurs, etc.).

- `spring-boot-devtools`

Fournit un rechargement automatique à chaud lors du développement, accélérant la productivité.

- `spring-boot-starter-test`

Fournit les dépendances nécessaires pour les tests unitaires et d'intégration (JUnit, Mockito, etc.).

- `spring-boot-maven-plugin`

Plugin Maven pour simplifier le lancement, le packaging et le déploiement des applications Spring Boot.

3.1.3 Base de données – H2

- H2 est une base de données embarquée idéale pour les environnements de test ou développement léger
- Supporte une console web accessible sur ‘/h2-console’ :

```

1  #!/usr/bin/perl
2
3  use strict;
4  use warnings;
5
6  my $url = "http://www.example.com";
7  my $user_agent = "Mozilla/5.0 (Windows NT 6.0; rv:2.0.1) Gecko/20100101 Firefox/4.0.1";
8
9  my $response = LWP::Simple::get($url);
10
11  if ($response) {
12      print "Success: $response\n";
13  } else {
14      print "Error: $!\n";
15  }
16
17  exit 0;

```

FIGURE 3.1 – /h2-console

3.2 Diagramme de classe :

Après l'élaboration du modèle conceptuel des données, on a formé le modèle logique correspondant qu'on retrouve ci-dessous. On voit bien l'utilisation de deux tables intermédiaires (installer, utilise) qui correspondent aux associations n-aires de notre modèle conceptuel.

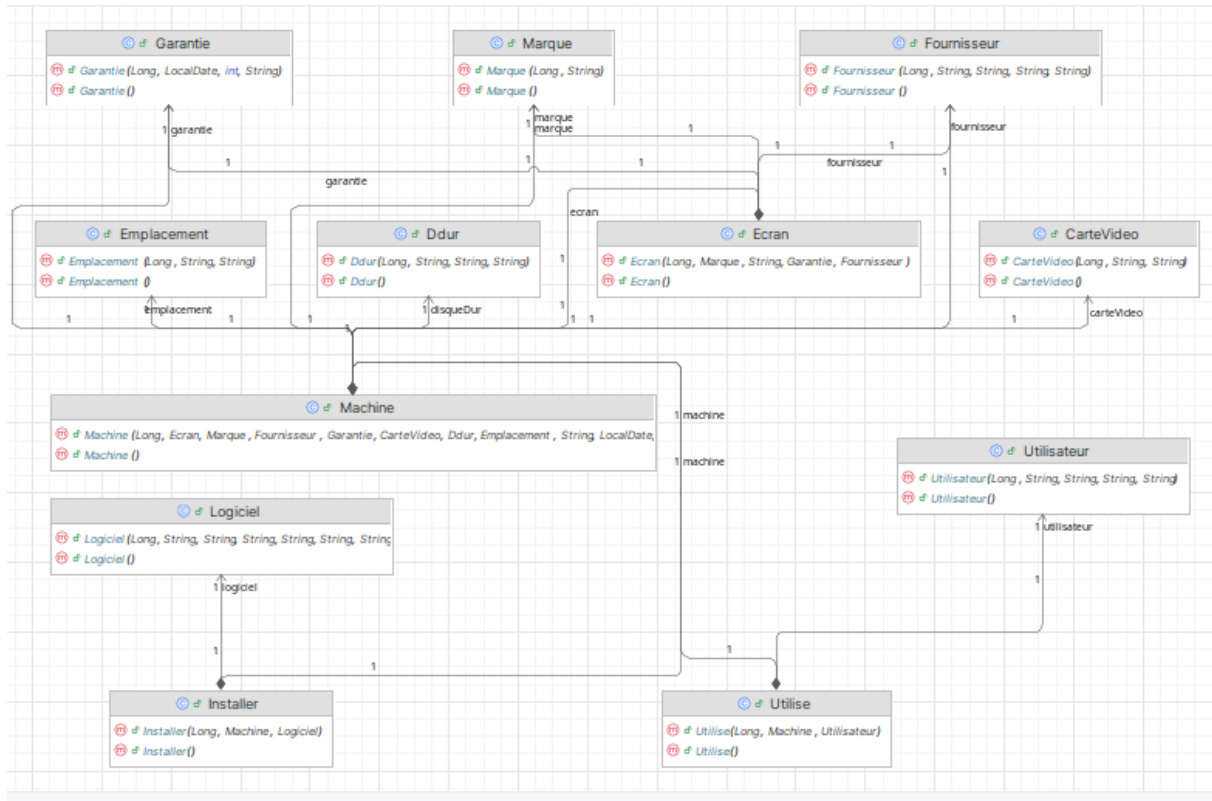


FIGURE 3.2 – Diagramme de classes

3.3 Relations entre les classes

Ce projet de gestion de parc informatique utilise des entités interconnectées à l'aide des annotations de persistance JPA, telles que `@ManyToOne` et `@OneToMany`, afin de refléter les relations existantes entre les objets métiers.

3.3.1 Classe Machine

- `@ManyToOne` avec :
 - Marque
 - Fournisseur
 - Garantie
 - CarteVideo
 - Ddur
 - Ecran
 - Emplacement

3.3.2 Classe Utilise

- @ManyToOne vers :
 - Machine
 - Utilisateur

3.3.3 Classe Installer

- @ManyToOne vers :
 - Machine
 - Logiciel

3.3.4 Classe Ecran

- @ManyToOne avec :
 - Marque
 - Garantie
 - Fournisseur

3.3.5 Autres relations

Les entités suivantes sont référencées par des relations @ManyToOne :

- Garantie
- Fournisseur
- CarteVideo
- Marque
- Utilisateur
- Emplacement
- Logiciel

Chaque relation est modélisée de manière à refléter les dépendances réelles entre les équipements, les logiciels, les utilisateurs et les fournisseurs dans un système de gestion de parc informatique.

3.4 Description de l'application

Cette application web a pour objectif de faciliter la **gestion centralisée d'un parc informatique**. Elle permet aux administrateurs de suivre, organiser et contrôler efficacement les ressources matérielles et logicielles d'une organisation.

3.4.1 Page d'accueil

La page d'accueil offre une **vue de bienvenue** intuitive. Elle présente un message d'introduction accompagné de plusieurs boutons de navigation permettant d'accéder rapidement aux fonctionnalités principales :

- **Machines** : gestion des postes informatiques (ordinateurs, serveurs, etc.).
- **Logiciels** : suivi des logiciels installés et de leurs licences.
- **Fournisseurs** : enregistrement et suivi des prestataires.
- **Utilisateurs** : association des équipements aux utilisateurs internes.

— **Emplacements** : localisation physique du matériel au sein de l'entreprise.

Cette page facilite l'orientation rapide de l'utilisateur dès sa connexion, avec une interface claire et fonctionnelle.

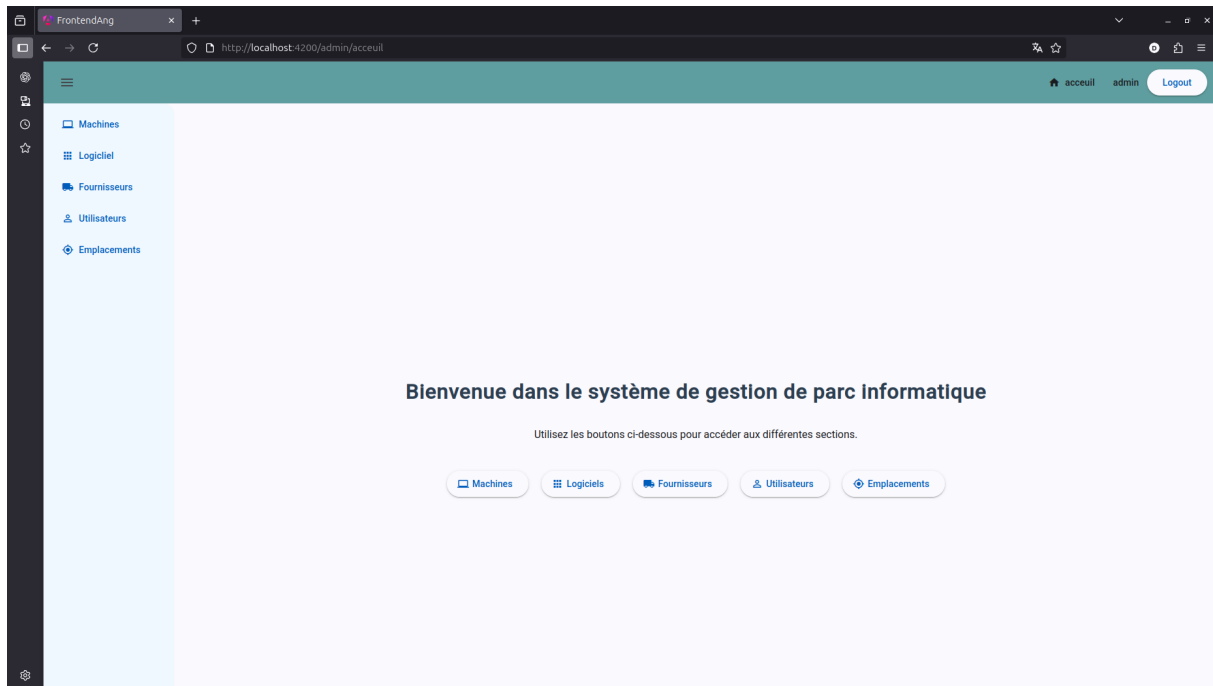


FIGURE 3.3 – Vue d'accueil

3.4.2 Interface d'administration (admin-template)

Le *template* d'administration repose sur une structure moderne utilisant **Angular Material**, composée de :

- Une **barre de navigation supérieure** (`mat-toolbar`) qui affiche l'identité de l'utilisateur connecté ainsi qu'un bouton de déconnexion.
- Un **menu latéral** (`mat-drawer`) accessible en permanence, permettant de naviguer facilement entre les différentes sections.
- Une zone de contenu dynamique intégrée via `router-outlet`, dans laquelle les composants métiers sont affichés selon la route active.

Ce modèle garantit une expérience utilisateur cohérente, rapide et organisée.

3.4.3 Avantages

L'application est conçue pour être :

- **Intuitive** : grâce à une interface utilisateur claire et interactive.
- **Modulaire** : architecture basée sur des composants Angular réutilisables.
- **Réactive** : performance optimisée pour une navigation fluide.
- **Sécurisée** : avec une gestion des sessions utilisateurs.
- **Extensible** : adaptable à d'autres besoins tels que la gestion de tickets ou l'inventaire réseau.

3.4.4 Diagramme de cas d'utilisation :

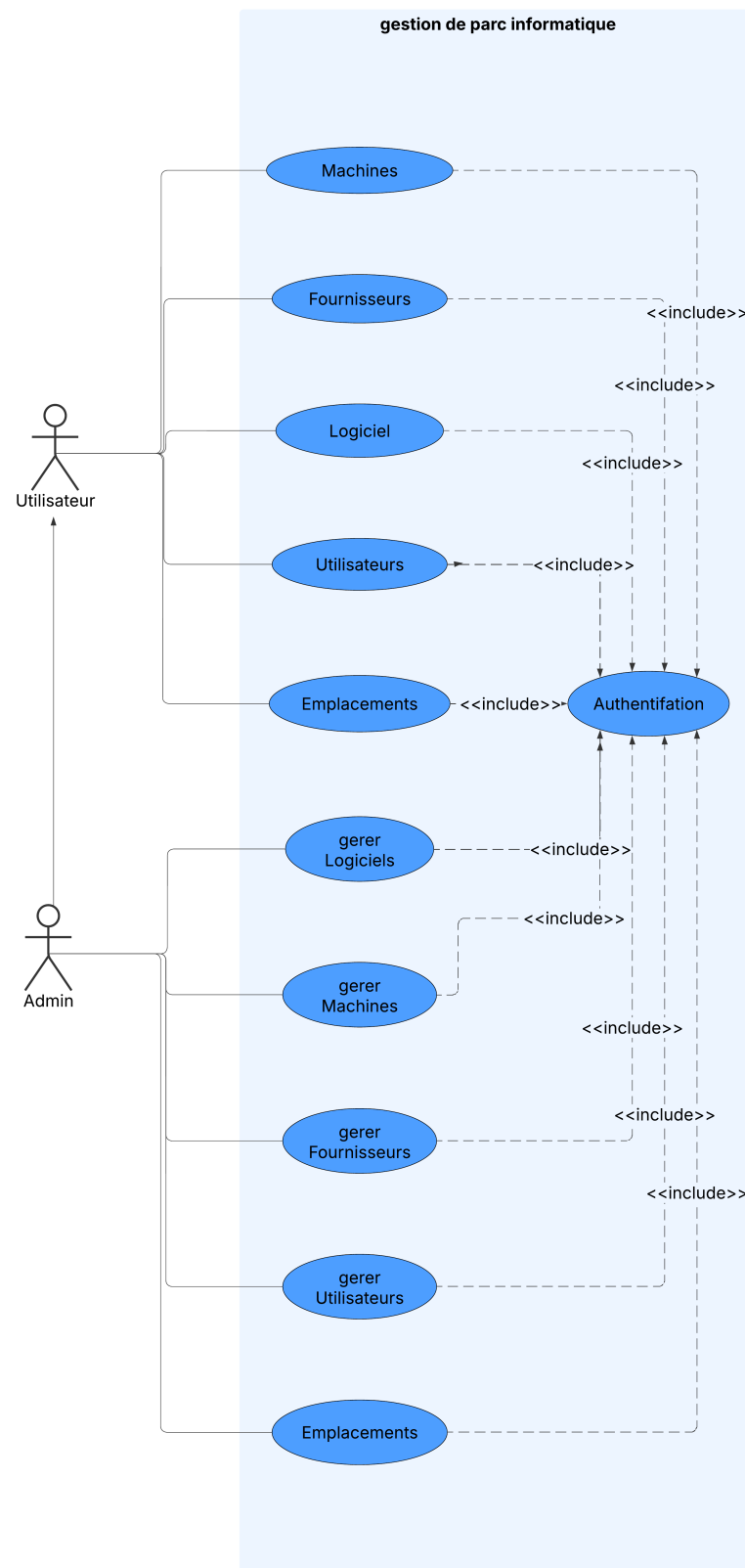


FIGURE 3.4 – diagramme de cas d'utilisation

3.4.5 Liste des API REST :

Méthode HTTP	URL	Description
GET	/api/machines	Récupérer toutes les machines
POST	/api/machines	Ajouter une nouvelle machine
GET	/api/machines/{id}	Récupérer une machine spécifique
PUT	/api/machines/{id}	Modifier une machine
DELETE	/api/machines/{id}	Supprimer une machine

FIGURE 3.5 – Exemple de liste des API REST des machines

3.5 Exemples de fonctionnalités :

3.5.1 Gestion des Logiciels

L'interface est encapsulée dans une carte `<mat-card>` composée de :

- un en-tête `<mat-card-header>` contenant le titre : **Ajouter un logiciel** ;
- une section principale `<mat-card-content>` contenant le formulaire.

I. Interface : Liste des logiciels

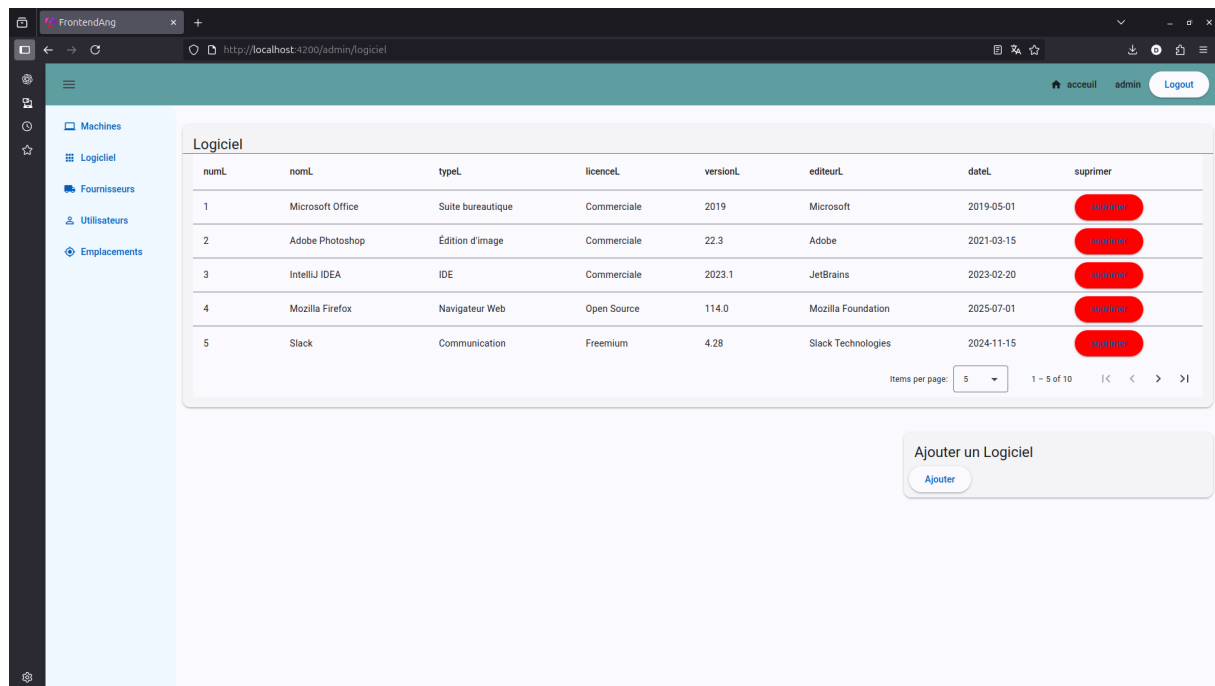


FIGURE 3.6 – Liste des Logiciels

1. Structure générale

La page est divisée en deux sections :

- Une carte affichant la liste des logiciels dans un tableau.
- Une carte optionnelle pour l'ajout, visible uniquement pour les administrateurs.

2. Affichage du tableau

Un tableau `<table mat-table>` présente les logiciels via une `dataSource` dynamique, avec tri (`matSort`) et pagination.

- Colonnes : `numL`, `nomL`, `typeL`, `licenceL`, `versionL`, `editeurL`, `dateL`, `supprimer`.
- `mat-paginator` permet de naviguer entre les pages.

3. Suppression conditionnelle

Une colonne "supprimer" contient un bouton rouge visible uniquement si l'utilisateur est administrateur. Ce bouton appelle la fonction `onDeleteLogiciel(numL)`.

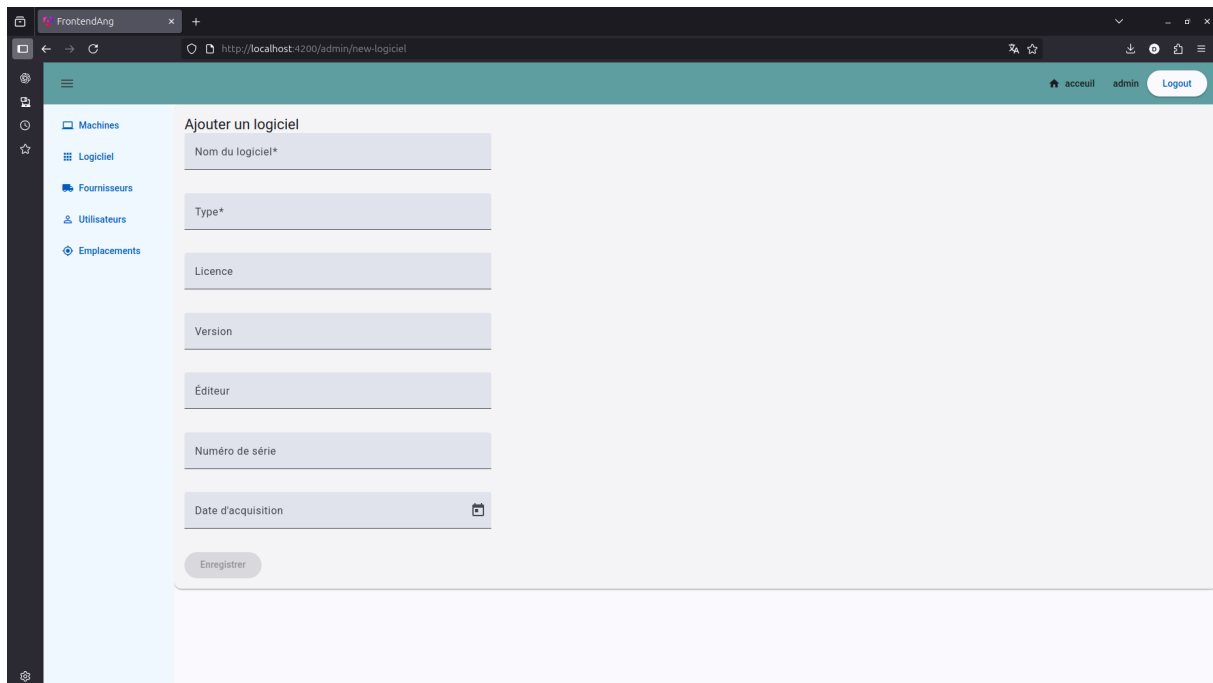
4. Ajout rapide (carte admin)

Une seconde carte s'affiche uniquement pour les administrateurs avec un bouton **Ajouter**, qui appelle `goToNewLogiciel()`.

5. Conclusion

Cette interface permet la visualisation et la gestion efficace des logiciels, avec des actions conditionnées au rôle de l'utilisateur. Elle utilise Angular Material pour le style et l'interactivité.

II. Interface : Ajout d'un logiciel



The screenshot shows a web browser window with the URL `http://localhost:4200/admin/new-logiciel`. The page features a dark sidebar on the left with navigation links: Machines, Logiciel, Fournisseurs, Utilisateurs, and Emplacements. The main content area is titled 'Ajouter un logiciel' and contains a form with the following fields: 'Nom du logiciel*' (required), 'Type*' (required), 'Licence', 'Version', 'Éditeur', 'Numéro de série', and 'Date d'acquisition' (with a calendar icon). A grey 'Enregistrer' button is at the bottom of the form. The top right of the page has links for 'accueil', 'admin', and a 'Logout' button.

FIGURE 3.7 – Interface ajouter Logiciel

Cette interface permet à un utilisateur (privilege d'administrateur) d'ajouter un logiciel via un formulaire. Elle est développée avec Angular et Angular Material.

(a) **Structure générale**

La page utilise un composant `<mat-card>` contenant :

- Un en-tête avec le titre **Ajouter un logiciel** ;
- Un contenu principal qui inclut un formulaire réactif.

(b) **Formulaire de saisie**

Le formulaire repose sur un `FormGroup` Angular. Les champs sont :

- **Nom du logiciel** : requis.
- **Type** : requis.
- **Licence, Version, Éditeur, Numéro de série** : facultatifs.
- **Date d'acquisition** : champ date avec `mat-datepicker`.

(c) **Soumission**

Le bouton **Enregistrer** est un `<button mat-raised-button>` désactivé tant que le formulaire est invalide. Lors du clic, la méthode `onSubmit()` est déclenchée.

(d) **Conclusion**

Le formulaire est structuré, sécurisé et utilise Angular Material pour une expérience utilisateur moderne.

Chapitre 4

Conclusion générale

Ce stage au sein de Tunisie Télécom m'a offert une première immersion concrète dans le monde professionnel. Il m'a permis non seulement de découvrir le fonctionnement global d'une grande entreprise nationale dans le secteur des télécommunications, mais également de consolider mes compétences techniques à travers la réalisation d'un projet de développement full-stack.

Sur le plan technique, j'ai pu approfondir mes connaissances en développement web en utilisant les frameworks Angular et Spring Boot, tout en respectant les contraintes réelles d'un cahier des charges. Le projet de gestion des équipements informatiques m'a permis de mettre en œuvre une architecture logicielle complète, intégrant une interface utilisateur ergonomique, une base de données relationnelle, et des services REST sécurisés.

D'un point de vue personnel, ce stage a renforcé ma rigueur, ma capacité d'adaptation, ainsi que mes compétences en travail d'équipe et en communication. L'échange avec les encadrants et les professionnels m'a aidé à mieux comprendre les exigences du métier et les attentes du monde de l'entreprise.

En somme, cette expérience a constitué une étape importante dans ma formation, en me préparant aussi bien sur le plan technique que professionnel à mes futures responsabilités d'ingénieur en informatique.