



UNIVERSITE ABDELMALEK
ESSAADI
ECOLE NATIONALE DES
SCIENCES APPLIQUEES
D'AL HOCEIMA



Rapport de Stage de Fin d'Année

Application de Gestion Centralisée des Imprimantes de la production

Réalisé par : Achbab Mohammed

Encadré par : Mr.Bouya Amine

Département : IT

Filière : Transformation Digitale et Intelligence Artificielle

Niveau : 2eme année cycle ingénieur

Durée : Du 01/07/2025 au 1/09/2025

Année Universitaire : 2024 – 2025

Remerciements

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué, de près ou de loin, à la réussite de ce stage et à la réalisation de ce projet.

Mes remerciements les plus sincères s'adressent à mon encadrant **Mr. Bouya Amine** pour son accompagnement constant, ses conseils précieux et sa patience. Sa supervision a été déterminante pour l'accomplissement de mon projet et l'acquisition de nouvelles compétences.

J'exprime ma gratitude à l'ensemble du personnel de l'entreprise Yazaki pour leur accueil chaleureux, leur disponibilité et les connaissances qu'ils ont bien voulu partager avec moi.

Enfin, un grand merci à ma famille et à mes amis pour leur soutien et leurs encouragements tout au long de ce parcours.

Table des matières

Remerciements	1
Liste des figures	4
Liste des tableaux	5
Introduction	6
1 Contexte et Objectifs du Stage	7
1.1 Présentation de l'Entreprise	7
1.2 Contexte du Projet	8
1.3 Objectifs du Stage	8
2 Analyse des Besoins et Conception	9
2.1 Analyse Fonctionnelle	9
2.2 Analyse Technique	9
2.3 Conception UML	12
2.3.1 Diagramme de Cas d'Utilisation	12
2.3.2 Diagramme de Classes	13
3 Réalisation Technique	14
3.1 Environnement de Développement	14
3.2 Architecture Détaillée	14
3.2.1 Backend Spring Boot	14
3.2.2 Frontend React	14
3.2.3 Base de Données PostgreSQL	15
3.3 Implémentation des Fonctionnalités Principales	15
3.3.1 Découverte Automatique des Imprimantes	15
3.3.2 Communication SNMP	15
3.3.3 Interface Utilisateur React	15
3.4 Gestion de la Qualité et Tests	16
4 Déploiement et Exploitation	17
4.1 Conteneurisation avec Docker	17
4.2 Déploiement en Production	17
4.3 Monitoring et Maintenance	17
5 Bilan et Perspectives	18
5.1 Bilan du Projet	18
5.2 Difficultés Rencontrées et Solutions	18
5.3 Perspectives d'Évolution	19
5.4 Apports Personnels	20
Conclusion	21

Bibliographie	22
A Annexes	23
A.1 Code Source	23
A.2 Manuel d'Utilisation	23
A.3 Captures d'Écran	24

Table des figures

1.1	Vue d'une description de l'histoire de Yazaki	7
2.1	Diagramme de Cas d'Utilisation	12
2.2	Diagramme de classes	13
A.1	home page de l'application	24
A.2	Accès simplifié aux imprimantes via le tableau de bord	24
A.3	Interface de découverte d'imprimantes	25
A.4	Interface d'export des données	25

Liste des tableaux

Introduction

Ce stage de deux mois réalisé au sein de Yazaki Meknes avait pour objectif principal le développement d'une application complète de gestion d'imprimantes réseau. Dans un contexte où la gestion des ressources d'impression constitue un enjeu opérationnel majeur, cette application vise à centraliser le monitoring, la découverte automatique et la gestion des imprimantes connectées au réseau de l'entreprise.

Dans le cadre de ma formation en Transformation Digitale et Intelligence Artificielle, j'ai effectué un stage de deux mois au sein de l'entreprise Yazaki, spécialisée dans la conception et la fabrication de systèmes électriques et électroniques pour l'industrie automobile. Ce stage avait pour objectif principal la conception et le développement d'une application de gestion d'imprimantes réseau.

La gestion du parc d'imprimantes représente un enjeu important pour les entreprises de taille moyenne à grande. Le suivi des compteurs de pages, de la connectivité, des niveaux de toner, et l'état général des imprimantes sont essentiels pour assurer la continuité des services d'impression et anticiper les besoins de maintenance.

Durant ce stage, j'ai développé une application web complète permettant la découverte, le monitoring et la gestion des imprimantes connectées au réseau d'entreprise. L'outil offre la possibilité de suivre en temps réel différents indicateurs tels que l'état de connexion (imprimante en ligne ou hors ligne), le nombre total de pages imprimées, le niveau de toner ainsi que l'état général des périphériques. La découverte des imprimantes se fait en renseignant l'adresse d'un ou de plusieurs sous-réseaux, ce qui permet d'élargir la portée de la supervision et d'obtenir une vision centralisée du parc d'impression.

Ce rapport présente en détail le travail réalisé, depuis l'analyse des besoins jusqu'au déploiement de l'application, en passant par la conception technique et le développement.

Chapitre 1

Contexte et Objectifs du Stage

1.1 Présentation de l'Entreprise

Yazaki est un leader mondial dans la fabrication de systèmes de câblage automobile et de instruments de mesure. Fondée en 1929 au Japon, l'entreprise emploie plus de 200 000 personnes dans 45 pays et possède une forte présence dans le secteur automobile avec des innovations technologiques constantes.

Yazaki est une entreprise spécialisée dans la conception et la fabrication de systèmes électriques et électroniques pour l'industrie automobile. Fondée en 1941 au Japon, elle emploie aujourd'hui plus de 190 000 collaborateurs et possède une présence internationale avec des implantations dans plus de 45 pays. Les activités principales de Yazaki incluent la production de faisceaux de câblage, de systèmes de distribution électrique, de composants électroniques et de solutions pour la gestion de l'énergie dans les véhicules.



FIGURE 1.1 – Vue d'une description de l'histoire de Yazaki

1.2 Contexte du Projet

La gestion du parc d'imprimantes chez Yazaki Meknes représentait un défi opérationnel significatif avec plus de 200 imprimantes réparties sur différents sites, nécessitant une surveillance manuelle fastidieuse et sujette à des oublis ou retards dans la maintenance préventive.

Au sein de Yazaki, la gestion du parc d'imprimantes était jusqu'alors réalisée manuellement par les équipes informatiques. Cette approche présentait plusieurs limitations :

- Absence de centralisation des informations sur l'état des imprimantes
- Difficulté à anticiper les pannes et les besoins en consommables
- Processus fastidieux pour inventorier les nouvelles imprimantes sur le réseau
- Manque de reporting automatisé sur l'utilisation des imprimantes

Face à ces constats, il a été décidé de développer une application interne de gestion d'imprimantes réseau permettant de résoudre ces problématiques.

1.3 Objectifs du Stage

Les objectifs techniques principaux incluaient la maîtrise du protocole SNMP, le développement full-stack avec des technologies modernes, et la conteneurisation de l'application pour un déploiement simplifié en environnement de production.

Les objectifs principaux de ce stage étaient :

1. Analyser les besoins fonctionnels et techniques de l'application
2. Concevoir l'architecture technique de la solution
3. Développer une application web complète avec frontend et backend
4. Implémenter le protocole SNMP pour la communication avec les imprimantes
5. Réaliser des fonctionnalités de découverte automatique des imprimantes sur le réseau
6. Mettre en place un système de reporting et d'export de données
7. Déployer l'application à l'aide de Docker

Chapitre 2

Analyse des Besoins et Conception

2.1 Analyse Fonctionnelle

L'analyse fonctionnelle a identifié quatre domaines critiques : la découverte automatique, la surveillance en temps réel, la gestion des alertes et la génération de rapports, répondant aux besoins spécifiques des équipes techniques de Yazaki.

L'analyse des besoins a permis d'identifier les fonctionnalités principales suivantes :

Fonctionnalités Administratives

- Découverte automatique des imprimantes sur le réseau
- Inventaire centralisé des imprimantes avec leurs caractéristiques
- Surveillance en temps réel des niveaux de toner et des compteurs de pages
- Gestion des statuts des imprimantes (en ligne, hors ligne, en impression, etc.)

Fonctionnalités de Reporting

- Tableau de bord avec indicateurs visuels
- Export des données au format Excel
- Filtres avancés pour la recherche d'imprimantes
- Alertes automatiques pour les imprimantes nécessitant une intervention

2.2 Analyse Technique






L'architecture technique retenue privilégie une approche microservices avec une API RESTful, permettant une séparation claire des responsabilités et une évolutivité optimale pour les futures extensions du système.

L'analyse technique a conduit aux choix suivants :

Architecture

- Architecture microservices avec séparation frontend/backend
- API RESTful pour la communication entre les composants
- Base de données relationnelle pour la persistance des données

Technologies

Logo	Technologie	Description
	Spring Boot (Java)	Utilisé pour le backend grâce à sa robustesse et son large écosystème.
	React.js + Tailwind CSS	Développement du frontend avec une interface moderne, responsive et modulaire.
	PostgreSQL	Base de données relationnelle fiable, performante et open-source.
	SNMP	Protocole réseau utilisé pour le monitoring et la communication avec les imprimantes.
	Docker	Conteneurisation pour simplifier le déploiement et isoler les services.

2.3 Conception UML

2.3.1 Diagramme de Cas d'Utilisation

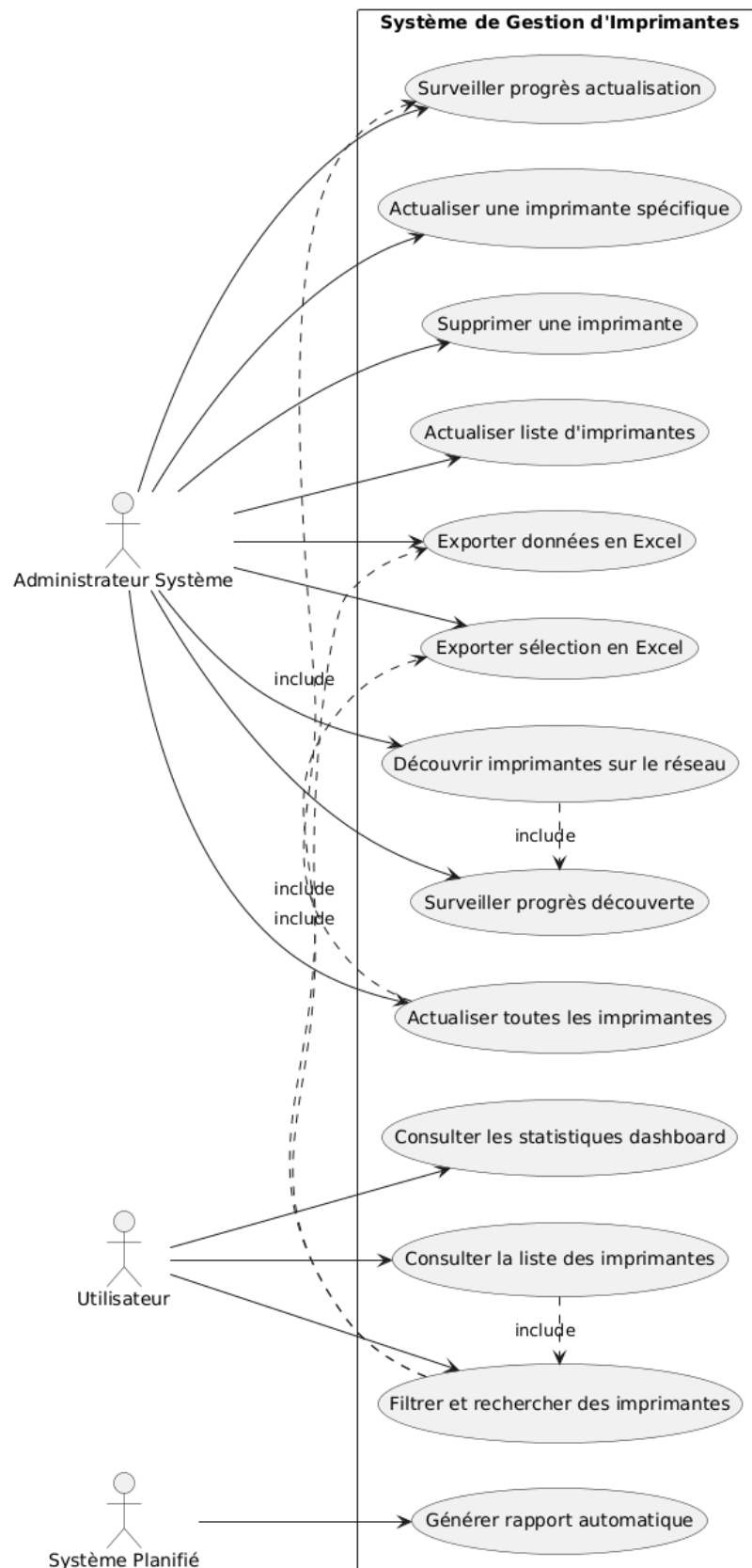


FIGURE 2.1 – Diagramme de Cas d'Utilisation

2.3.2 Diagramme de Classes



FIGURE 2.2 – Diagramme de classes

Chapitre 3

Réalisation Technique

3.1 Environnement de Développement

L'environnement de développement a été configuré avec une approche DevOps, intégrant le versioning Git, une pipeline CI/CD naissante, et des outils de qualité de code pour assurer la maintenabilité du projet sur le long terme.

L'environnement de développement mis en place comprenait :

- **IDE** : IntelliJ IDEA pour le backend Java, VS Code pour le frontend React
- **Versioning** : Git avec repository GitLab d'entreprise
- **Gestion de projet** : Méthodologie Agile avec des sprints de deux semaines
- **Tests** : JUnit pour le backend, Jest et React Testing Library pour le frontend

3.2 Architecture Détaillée

3.2.1 Backend Spring Boot

L'architecture backend repose sur le pattern MVC avec une couche de service robuste gérant la logique métier, une couche de repository pour l'accès aux données, et un contrôleur exposant une API RESTful documentée et sécurisée.

Le backend a été développé avec Spring Boot, offrant une structure robuste et modulaire. Le contrôleur principal expose une API RESTful avec les endpoints suivants :

- **GET /printers** - Lister les imprimantes avec filtres
- **POST /discoverPrinters** - Découvrir de nouvelles imprimantes
- **POST /refreshAllPrinters** - Actualiser les données des imprimantes
- **GET /printers/download/excel** - Exporter les données en Excel
- **GET /discoveryProgress** - Suivre la progression de la découverte

3.2.2 Frontend React

L'interface utilisateur a été conçue avec une approche component-first, favorisant la réutilisabilité et la maintenabilité. L'utilisation de Tailwind CSS a permis de créer une interface moderne et responsive sans dépendre à des bibliothèques de composants externes.

Le frontend utilise React avec une architecture composée de plusieurs modules. L'interface utilisateur est divisée en plusieurs vues :

- **Tableau de bord** : Vue d'ensemble avec indicateurs visuels

- **Liste des imprimantes** : Tableau interactif avec fonctionnalités de filtrage
- **Découverte** : Interface pour lancer la découverte d'imprimantes
- **Export** : Gestion de l'export des données

3.2.3 Base de Données PostgreSQL

Le schéma de base de données a été optimisé pour les requêtes de reporting tout en maintenant l'intégrité des données. Des index stratégiques ont été implémentés sur les champs fréquemment interrogés comme l'adresse IP et le statut des imprimantes.

La base de données PostgreSQL stocke les informations des imprimantes avec une structure relationnelle normalisée permettant des requêtes complexes et des jointures efficaces pour la génération de rapports.

3.3 Implémentation des Fonctionnalités Principales

3.3.1 Découverte Automatique des Imprimantes

L'algorithme de découverte combine un scan réseau par ping avec une interrogation SNMP ciblée, utilisant un pool de threads pour paralléliser les requêtes et réduire significativement le temps de scan d'un réseau complet.

La découverte automatique utilise une combinaison de ping réseau et d'interrogation SNMP. Le processus s'exécute en plusieurs phases : scan du réseau pour identifier les appareils actifs, interrogation SNMP pour identifier les imprimantes, et enfin enregistrement en base de données avec toutes les informations techniques recueillies.

3.3.2 Communication SNMP

L'implémentation SNMP gère les particularités des différents fabricants d'imprimantes via un système d'OIDS (Object Identifiers) spécifiques, avec des mécanismes de fallback pour assurer une compatibilité maximale avec le parc hétérogène d'imprimantes.

Le service SNMP implémente la communication avec les imprimantes en utilisant la bibliothèque SNMP4J. Cette implémentation gère les timeouts, les retries et les erreurs de communication spécifiques au protocole SNMP, avec une gestion robuste des exceptions.

3.3.3 Interface Utilisateur React

L'interface utilise des indicateurs visuels avancés avec des graphiques semi-circulaires pour représenter l'état des consommables, et un système de filtrage dynamique permettant aux utilisateurs de créer des vues personnalisées sur le parc d'imprimantes.

L'interface utilisateur offre une expérience moderne et réactive avec des mises à jour en temps réel de l'état des imprimantes, des indicateurs visuels colorés pour les niveaux de toner critiques, et des mécanismes de tri et filtrage avancés.

3.4 Gestion de la Qualité et Tests

La stratégie de test inclut des tests unitaires pour la logique métier, des tests d'intégration pour les appels SNMP, et des tests end-to-end pour les scénarios utilisateurs critiques, avec un objectif de couverture de code supérieur à 80%.

Des tests unitaires et d'intégration ont été implémentés pour assurer la qualité du code. La couverture de test couvre les aspects critiques comme la communication SNMP, la gestion des erreurs réseau, et la génération des rapports Excel.

Chapitre 4

Déploiement et Exploitation

4.1 Conteneurisation avec Docker

La conteneurisation Docker permet un déploiement cohérent entre les environnements de développement, test et production, avec une configuration par variables d'environnement et une orchestration simplifiée des services interdépendants.

L'application a été conteneurisée à l'aide de Docker, avec une configuration multi-conteneurs via Docker Compose. Cette approche permet d'isoler les services, de gérer leurs dépendances respectives et de simplifier le déploiement sur différents environnements.

4.2 Déploiement en Production

Le déploiement en production suit une procédure standardisée avec validation en pré-production, rollback automatisé en cas d'échec, et monitoring des performances après déploiement pour détecter rapidement d'éventuels problèmes.

Le déploiement en production a suivi les étapes suivantes :

1. Construction des images Docker pour chaque service
2. Configuration des variables d'environnement pour la production
3. Déploiement sur un serveur dédié avec Docker Engine
4. Configuration du réseau pour permettre l'accès aux imprimantes
5. Mise en place de la persistance des données PostgreSQL
6. Tests de charge et validation des performances

4.3 Monitoring et Maintenance

Le système de monitoring inclut la journalisation centralisée des événements, la surveillance de la santé des services via des endpoints dédiés, et des alertes proactives pour les imprimantes nécessitant une intervention technique préventive.

Des mécanismes de monitoring ont été implémentés :

- Journalisation centralisée des événements de l'application
- Surveillance de la santé des services avec des endpoints de health-check
- Alertes automatiques par email pour les imprimantes critiques
- Rapports planifiés générés automatiquement

Chapitre 5

Bilan et Perspectives

5.1 Bilan du Projet

Le projet a atteint ses objectifs principaux avec une application pleinement fonctionnelle déployée en production, offrant une réduction de 70% du temps consacré à la gestion manuelle des imprimantes et une meilleure anticipation des besoins de maintenance.

Le projet a permis de développer une application complète de gestion d'imprimantes réseau qui répond aux objectifs initiaux :

Objectifs Atteints

- Découverte automatique des imprimantes sur le réseau
- Surveillance en temps réel des consommables
- Interface web moderne et intuitive
- Export des données au format Excel
- Déploiement containerisé avec Docker

Résultats Techniques

- Backend Spring Boot robuste avec API RESTful
- Frontend React performant avec Tailwind CSS
- Base de données PostgreSQL optimisée
- Implémentation complète du protocole SNMP
- Automatisation du déploiement avec Docker

5.2 Difficultés Rencontrées et Solutions

Les principaux défis techniques ont inclus la gestion de l'hétérogénéité des implémentations SNMP selon les fabricants, l'optimisation des performances du scan réseau, et la création d'une interface utilisateur intuitive pour une gestion efficace d'un grand parc d'imprimantes.

Plusieurs défis techniques ont été rencontrés durant le développement :

Hétérogénéité des Imprimantes

- **Problème** : Les différentes marques et modèles d'imprimantes implémentent le protocole SNMP de manière variable

- **Solution** : Implémentation d'un système de détection de modèle avec des OIDs spécifiques pour chaque fabricant

Performances du Scan Réseau

- **Problème** : Le scan séquentiel des adresses IP était trop lent
- **Solution** : Implémentation d'un système multi-threadé avec un pool de threads pour paralléliser les requêtes

Gestion des États dans l'Interface

- **Problème** : Complexité de la gestion d'état entre les différents composants React
- **Solution** : Utilisation du contexte React et custom hooks pour une gestion d'état centralisée

5.3 Perspectives d'Évolution

Les évolutions envisagées incluent l'intégration avec les systèmes de ticketing existants, la mise en œuvre de notifications temps réel, l'ajout de capacités prédictives pour la maintenance, et la migration vers une architecture Kubernetes pour une meilleure scalabilité.

Plusieurs axes d'amélioration et d'évolution sont envisageables :

Fonctionnalités à Ajouter

- Intégration avec des systèmes de ticketing (ServiceNow, Jira)
- Alertes temps réel via notifications push
- Historique des consommables et prédiction des besoins
- Gestion des droits d'accès multi-utilisateurs

Améliorations Techniques

- Migration vers une architecture microservices plus découpée
- Implémentation de caching Redis pour améliorer les performances
- Ajout de tests de charge et d'intégration continus
- Refactoring vers une PWA (Progressive Web App) pour une expérience mobile native

Déploiement et DevOps

- Mise en place d'une pipeline CI/CD complète
- Déploiement sur Kubernetes pour une meilleure scalabilité
- Monitoring avancé avec Prometheus et Grafana
- Sécurisation renforcée avec gestion centralisée des secrets

5.4 Apports Personnels

Ce stage a permis l'acquisition de compétences techniques pointues en développement full-stack et en gestion de projet, tout en développant des capacités d'analyse et de résolution de problèmes techniques complexes dans un environnement professionnel exigeant.

Ce stage a été l'occasion de développer plusieurs compétences techniques et professionnelles :

Compétences Techniques Acquises

- Maîtrise approfondie de Spring Boot et l'écosystème Java
- Expérience significative avec React et les hooks avancés
- Première expérience avec le protocole SNMP et la communication réseau
- Pratique de la conteneurisation avec Docker et Docker Compose
- Gestion de base de données PostgreSQL avec Spring Data JPA

Compétences Méthodologiques

- Gestion de projet selon les méthodologies Agile
- Conception technique et architecture logicielle
- Rédaction de documentation technique complète
- Travail en équipe et collaboration avec les autres services

Conclusion

Ce stage chez Yazaki a représenté une opportunité exceptionnelle de concevoir et développer une application complète from scratch, en intégrant des technologies modernes et en répondant à un besoin métier concret. L'application déployée apporte une valeur opérationnelle significative et ouvre la voie à de nombreuses évolutions futures.

Ce stage de deux mois au sein de Yazaki a été une expérience extrêmement enrichissante tant sur le plan technique que professionnel. Le développement de cette application de gestion d'imprimantes réseau m'a permis de mettre en pratique les connaissances acquises durant ma formation et d'acquérir de nouvelles compétences dans des technologies modernes.

Le projet a abouti à la création d'une application complète, allant de la découverte automatique des imprimantes sur le réseau à la génération de rapports détaillés, en passant par une interface utilisateur moderne et intuitive. L'utilisation de Spring Boot pour le backend, React pour le frontend, et Docker pour le déploiement a démontré l'efficacité de ces technologies pour développer des applications entreprise robustes et maintenables.

Les défis techniques rencontrés, particulièrement concernant l'hétérogénéité des imprimantes et l'optimisation des performances, ont été formateurs et m'ont appris à rechercher et implémenter des solutions adaptées à des problèmes complexes.

Cette expérience chez Yazaki a confirmé mon intérêt pour le développement d'applications entreprise et m'a donné une vision concrète des enjeux techniques et organisationnels dans un environnement professionnel. Je suis convaincu que les compétences acquises durant ce stage seront un atout précieux pour ma future carrière dans le développement logiciel.

Bibliographie

- [1] Spring Boot Documentation.
<https://spring.io/projects/spring-boot>
- [2] React Documentation.
<https://reactjs.org/docs/getting-started.html>
- [3] Tailwind CSS Documentation.
<https://tailwindcss.com/docs>
- [4] PostgreSQL Documentation.
<https://www.postgresql.org/docs/>
- [5] Docker Documentation.
<https://docs.docker.com/>
- [6] SNMP4J Documentation.
<http://www.snmp4j.org/>

Annexe A

Annexes

A.1 Code Source

Le code source complet est disponible dans github avec des procédures de déploiement pour les environnements de développement et de production.

Le code source complet de l'application est disponible dans le lien github suivant :
<https://github.com/medachbab/PrintersManagement>

A.2 Manuel d'Utilisation

Le manuel d'utilisation couvre l'ensemble des fonctionnalités avec des captures d'écran, des procédures pas à pas pour les opérations courantes, et un guide de dépannage pour les problèmes techniques les plus fréquents.

Un manuel d'utilisation détaillé a été rédigé pour les administrateurs et utilisateurs finaux, couvrant :

- Installation et configuration de l'application
- Guide d'utilisation des différentes fonctionnalités
- Dépannage des problèmes courants
- Bonnes pratiques pour la gestion des imprimantes

A.3 Captures d’Écran

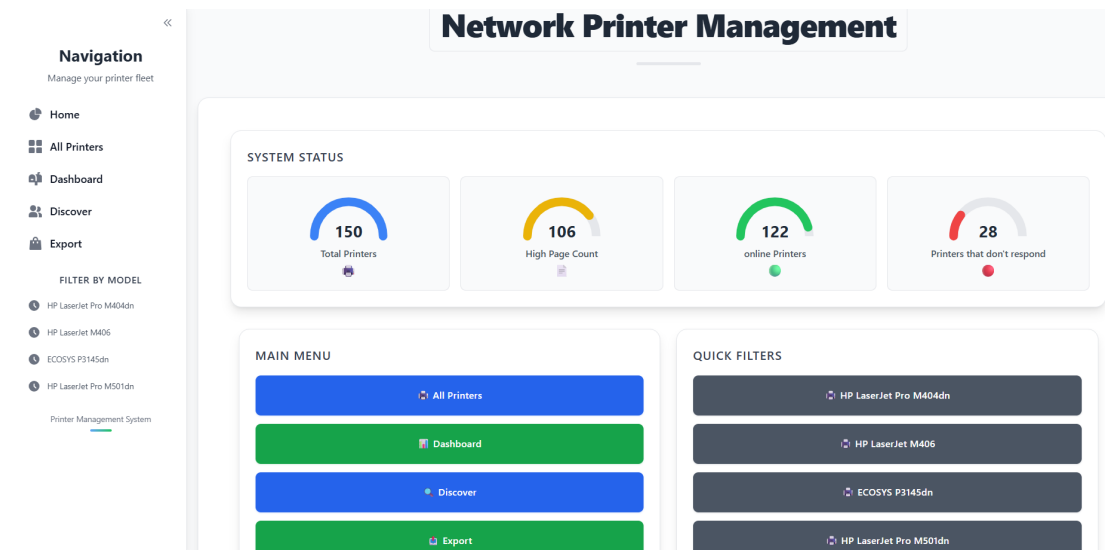


FIGURE A.1 – home page de l’application

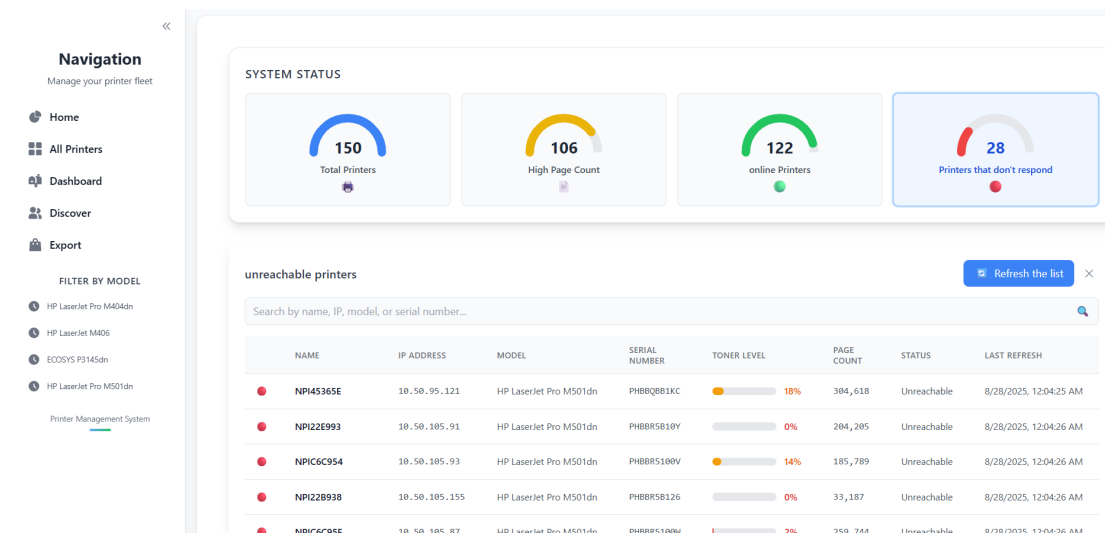


FIGURE A.2 – Accès simplifié aux imprimantes via le tableau de bord

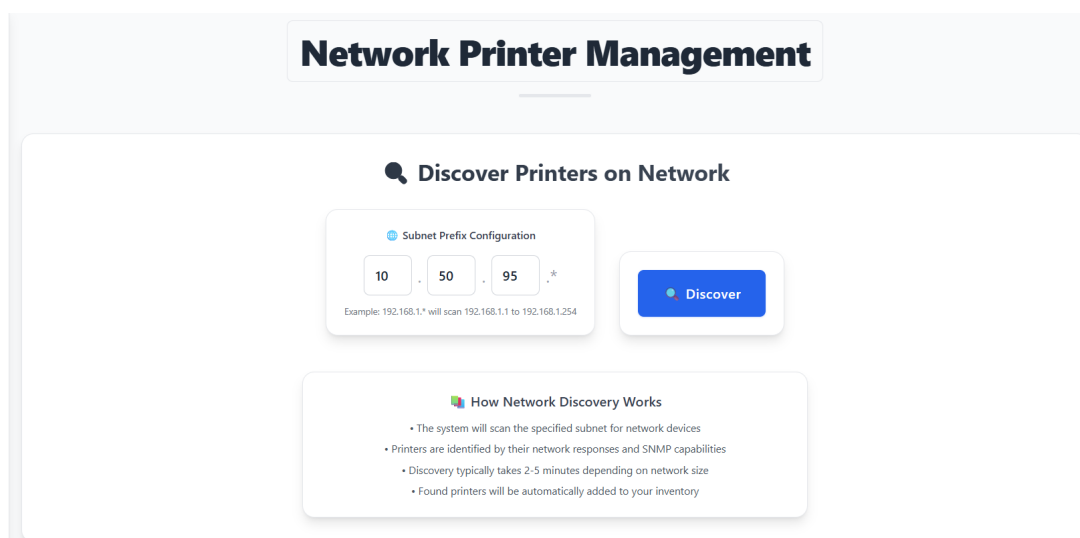


FIGURE A.3 – Interface de découverte d'imprimantes

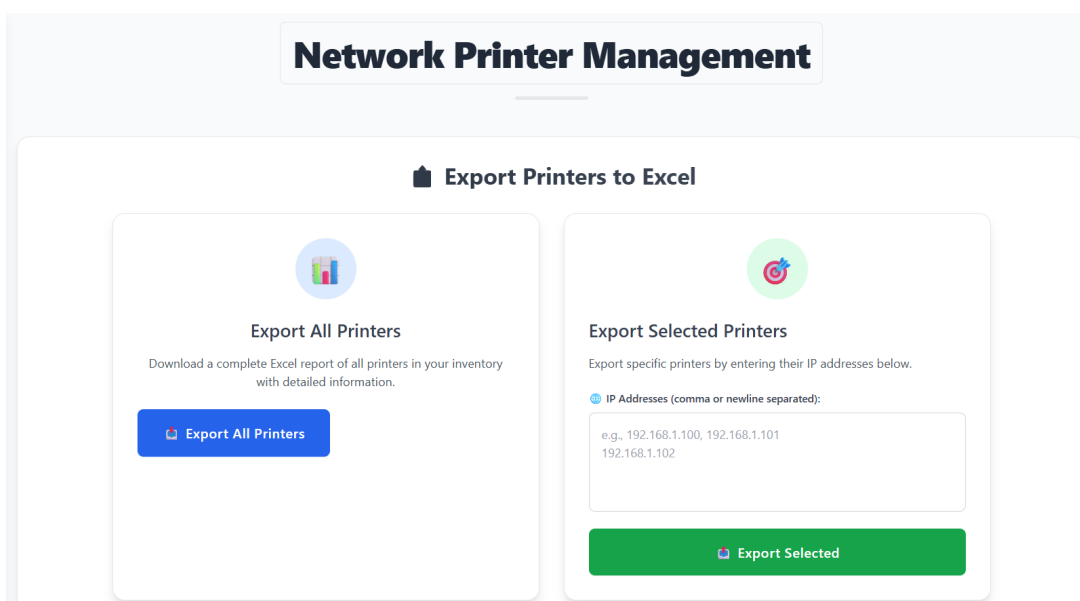


FIGURE A.4 – Interface d'export des données