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

Gestion des actifs

**Création d'une application de suivi des mouvements des stocks et des états des immobilisations de Samifin**

Par

RATIATIANA Jean Mirlin

Mémoire présenté  
en vue de l’obtention du grade de licence

en informatique

option de développement

Novembre 2024

Jury :

Mme Baovola RAJAONARISON, présidente  
 M. Tovo ANDRIAMBELOMA, examinateur  
 M. Tombo Herbert, encadreur professionnel

© RATIATIANA Jean Mirlin, 2024

**Table des matières**

[Liste des tableaux iii](#_Toc182035161)

[Liste des figures iv](#_Toc182035162)

[Glossaire vi](#_Toc182035163)

[Avant-propos 1](#_Toc182035164)

[L’IT University 1](#_Toc182035165)

[Samifin 2](#_Toc182035166)

[Remerciements 3](#_Toc182035167)

[Introduction 4](#_Toc182035168)

[1 Présentation du projet 5](#_Toc182035169)

[1.1 Objectifs du projet 5](#_Toc182035170)

[1.2 Planning de réalisation 5](#_Toc182035171)

[1.3 Architecture du projet 6](#_Toc182035172)

[1.4 Technologies utilisées 7](#_Toc182035173)

[2 Réalisation de l’application 12](#_Toc182035174)

[2.1 Analyse et conception 12](#_Toc182035175)

[2.1.1 Analyse de l'existant 12](#_Toc182035176)

[2.1.2 Conception de l’application 12](#_Toc182035177)

[2.2 Développement par fonctionnalité ou module 14](#_Toc182035178)

[2.2.1 Gestion d'utilisateur 14](#_Toc182035179)

[2.2.2 Gestion des biens 20](#_Toc182035180)

[2.2.3 Gestion des stocks 25](#_Toc182035181)

[2.2.4 Gestions des immobilisations 31](#_Toc182035182)

[2.2.5 Notification et recherche 36](#_Toc182035183)

[2.3 État d’Analyse et Statistiques 39](#_Toc182035184)

[2.3.1 Répartition des utilisateurs par direction 39](#_Toc182035185)

[2.3.2 Statistiques des dépenses par mois 39](#_Toc182035186)

[2.3.3 Quantité de stock 39](#_Toc182035187)

[2.4 Problèmes rencontrés et solutions 40](#_Toc182035188)

[3 Évaluation du projet et connaissances acquises 41](#_Toc182035189)

[3.1 Bilan pour l'entreprise 41](#_Toc182035190)

[3.2 Bilan personnel 41](#_Toc182035191)

[3.3 Extension et évolution de l’application 41](#_Toc182035192)

[Conclusion 43](#_Toc182035193)

[Bibliographie 44](#_Toc182035194)

[Annexe i](#_Toc182035195)

# Liste des tableaux

[Tableau 1 :Tableau comparatif entre ASP.NET Core et Spring Boot. 9](#_Toc182035207)

[Tableau 2 :Tableau comparatif entre PostgreSQL et MySQL. 11](#_Toc182035208)

# Liste des figures

[Figure 1 : Diagramme de Gantt pour la planification du projet 5](#_Toc182035216)

[Figure 2 : Structure générale du projet. 6](#_Toc182035217)

[Figure 3 : Structure détailés du projet 7](#_Toc182035218)

[Figure 4 : Comparaison d'ASP.NET Core avec ses alternatives 9](#_Toc182035219)

[Figure 5 : Comparaison PostgreSQL avec MySQL 11](#_Toc182035220)

[Figure 6 : Script MCD des tables 13](#_Toc182035221)

[Figure 7 : Script des views 14](#_Toc182035222)

[Figure 8 : Création d'utilisateur 16](#_Toc182035223)

[Figure 9 : Confirmation de la création d’utilisateur 16](#_Toc182035224)

[Figure 10 : Page d'accès refusé pour rôle 17](#_Toc182035225)

[Figure 11,12 : Connexion et Modification mot de passe 18](#_Toc182035226)

[Figure 13 : Liste des utilisateurs et leurs profiles 19](#_Toc182035227)

[Figure 14 : Profil de l’utilisateur connecté 20](#_Toc182035228)

[Figure 15 : Modal d'importation de fichier Excel 21](#_Toc182035229)

[Figure 16 : Confirmation de l’importation de fichier Excel 22](#_Toc182035230)

[Figure 17 :Ajout et Liste des biens 23](#_Toc182035231)

[Figure 18 : Ajout des nouveaux prix et liste des biens avec leur prix actuel 24](#_Toc182035232)

[Figure 19 : Historiques des prix d’un bien 25](#_Toc182035233)

[Figure 20 : Ajout et liste des biens avec leurs restes des stocks 27](#_Toc182035234)

[Figure 21 : Évaluation générale des stocks entre deux dates 28](#_Toc182035235)

[Figure 22 : Évaluation détaillée des stocks entre deux dates 29](#_Toc182035236)

[Figure 23 : Format PDF d’une évaluation des stocks 29](#_Toc182035237)

[Figure 24 : Module d’inventaire des stocks 30](#_Toc182035238)

[Figure 25 : Résultats de l'inventaire des stocks 31](#_Toc182035239)

[Figure 26 : Liste des inventaires effectués 31](#_Toc182035240)

[Figure 27 : Ajout et liste des immobilisations avec leurs états et propriétaires 32](#_Toc182035241)

[Figure 28 : Fiche de détention d’un employé 33](#_Toc182035242)

[Figure 29 : Fiche de détention d’un employé format PDF 34](#_Toc182035243)

[Figure 30 : Détails et historique des immobilisations 35](#_Toc182035244)

[Figure 31 : Ajout et liste des maintenances périodiques 36](#_Toc182035245)

[Figure 32 : Modal de modification d'une planification de maintenance 36](#_Toc182035246)

[Figure 33 : Page de messages pour les notifications. 37](#_Toc182035247)

[Figure 34 : Champ de recherche pour tous les tableaux de données 38](#_Toc182035248)

[Figure 35 : Tableau de bord statistique 39](#_Toc182035249)

[Figure 36 : Configuration d'affichage et de filtre pour les stocks i](#_Toc182035250)

[Figure 37 : Accès à la base de données et Paramètres SMTP i](#_Toc182035251)

# Glossaire

**Asynchrone.** Un mode d'exécution dans lequel les tâches ou les opérations peuvent être lancées et exécutées indépendamment les unes des autres, sans que l'application principale ou l'utilisateur ait à attendre leur achèvement.

**Backend**. Face cachée de l'application qui s'occupe des aspects invisibles pour l'utilisateur final, tels que le traitement des données, la gestion des bases de données, la logique métier, les serveurs, les opérations et les traitements en arrière-plan.

**Base de données**. Ensemble structurée et organisée d'informations généralement stockées électroniquement dans un ordinateur ou un système informatique de manière à être facilement accessible, géré et mis à jour.

**Bibliothèque**. Ensemble de ressources, de fonctions, de modules ou de composants logiciels réutilisables qui sont mis à disposition des développeurs pour faciliter leur travail.

**Codes-barres**. Désigne un système de codage graphique utilisé pour représenter des informations sous forme de barres noires et blanches. Ces codes peuvent être scannés à l'aide de lecteurs de codes-barres pour récupérer rapidement des informations, comme des prix, des identifiants de produits, ou des numéros de série.

**Claims**. un ensemble d'attributs ou de données, souvent sous forme de paires clé-valeur, qui sont associées à un utilisateur ou à une entité. Ces informations peuvent être diverses et servent à donner plus de contexte sur l'utilisateur authentifié, à vérifier son identité, ou à autoriser des actions spécifiques basées sur ses rôles ou ses attributs.

**Framework**. Ensemble d'outils qui aident les développeurs à créer des applications plus rapidement et plus facilement en fournissant des éléments prêts à l'emploi et des composants réutilisables.

**IActionResult**. Un type de retour utilisé dans les contrôleurs d'ASP.NET Core MVC .Il représente le résultat d'une action, c'est-à-dire la réponse que le serveur renvoie après avoir traité une requête.

**Jquery**. Un outil utilisé pour rendre le développement web plus facile et plus rapide. Il permet de manipuler les éléments d'une page web, comme modifier du texte, cacher ou afficher des éléments, et ajouter des animations. Avec jQuery, il est aussi plus simple d'ajouter des interactions comme des clics de boutons et de charger des données sans recharger la page.

**Open source**. Modèle de développement logiciel dans lequel le code source d'un logiciel est rendu public et est librement disponible pour être consulté, modifié, amélioré et redistribué par n'importe qui.

**Package**. Ensemble de fichiers, de bibliothèques ou de composants logiciels qui sont regroupés ensemble pour fournir une fonctionnalité spécifique au sein d’un projet ou d’une application.

**PBKDF2**. Une fonction cryptographique utilisée pour dériver une clé de chiffrement ou un hachage sécurisé à partir d'un mot de passe.

**Plateforme**. Ensemble de matériel ou de logiciel qui héberge et prend en charge l'exécution d'applications ou de services.

**Rainbow table.** Une méthode d'attaque utilisée pour casser des fonctions de hachage, en particulier celles utilisées pour protéger les mots de passe.

**Requête**. Demande ou instruction formulée par un programme ou un utilisateur pour obtenir des données, effectuer une opération ou communiquer avec un système distant.

**Routes**. Chemin ou URL spécifique qui correspond à une ressource ou à une action particulière au sein d’une application.

**Serveur**. Ordinateur ou système informatique spécialement conçu pour offrir des services, des ressources ou des données à d'autres ordinateurs en utilisant un réseau.

**Tables**. Structure de données qui organise et stocke des informations sous forme de lignes et de colonnes.

# Avant-propos

Le présent mémoire présente les résultats du travail effectué lors de mon stage de fin d’études de Licence en Informatique de l’IT University, stage effectué au « Direction administratif et financier » de « SAMIFIN » durant 3 mois, d’aout à novembre 2024.

Afin de poser clairement le contexte de ce mémoire, je vais présenter succinctement d’une part l’IT University et d’autre part mon institution d’accueil.

## L’IT University

Fondée en 2011, l’IT University (ou ITU) est une université privée, spécialisée en informatique, formant les jeunes bacheliers, de préférence scientifiques :

* En trois ans, pour l‘obtention d’une :
  + Licence, option Développement, Réseaux et Bases de Données ou Web et Design
  + Licence, option Graphic Design ou Communication Digitale
* En cinq ans, pour l’obtention d’un :
  + Master MBDS en coopération avec l’Université Côte d’Azur à Nice Sophia Antipolis – France
  + Master BIHAR en coopération avec l’ESTIA du Pays Basque - France

Étant une formation professionnalisante, l’ITU a tissé des liens forts avec ses partenaires industriels, dont l’opérateur convergent TELMA et la plupart des entreprises et institutions du secteur des TIC[[1]](#footnote-1). Ces partenaires participent effectivement à la formation par la fourniture de connexion Internet à haut débit, l’envoi de conférenciers ou par l’accueil des étudiants en stage. Beaucoup de ces partenaires recrutent aussi les sortants dès leur sortie d’École.  
D’autre part, le corps enseignant de l’ITU est constitué intégralement de spécialistes de très haut niveau et obligatoirement actifs professionnellement dans leurs domaines respectifs.

## Samifin

Le Service de Renseignement Financier (SRF), en abrégé « SAMIFIN », est l'entité nationale chargée de conduire la LBC/FT[[2]](#footnote-2) à Madagascar.

Crée en 2007 et réorganisé en 2015, le SAMIFIN est un service de renseignement financier de type administratif ,rattaché à la Présidence de la République. Ses rapports d’analyse ont simple valeur de renseignement d’intelligence. Disposant d’une indépendance opérationnelle et d’une autonomie de gestion ,le SAMIFIN s’article en 01 Direction Générale , 04 Directions techniques et 01 Direction d’appui.

Entérinée par la loi n° 2018- 043 du 13 février 2019 sur la LBC/ FT, la mission du SAMIFIN consiste à lutter contre les flux financiers  illicites liés à des faits de blanchiment de capitaux, de financement du terrorisme, des infractions économiques et financières et à tout crime organisé :

1. Prévention et Détection : Mettre en place des mécanismes pour identifier les activités suspectes et les transactions potentiellement liées au blanchiment d'argent ou au financement du terrorisme.
2. Analyse et Enquête : Analyser les données financières pour détecter les anomalies et collaborer avec les autorités judiciaires dans le cadre d'enquêtes.
3. Coordination et Formation : Travailler avec les institutions financières, les autorités publiques et les organisations internationales pour renforcer les capacités locales en matière de lutte contre le blanchiment d'argent et le financement du terrorisme.
4. Conformité : Assurer la conformité avec les normes et recommandations internationales en matière de lutte contre le blanchiment d'argent et le financement du terrorisme.

## Remerciements

Tout d’abord ,je tiens à remercier mon encadreur professionnel, M. Tombo Herbert, développeur chez Samifin, pour les conseils et les propositions qu’il a donné durant mon stage.

Je tiens également à remercier mon encadrreur pédagogique et examinateur, Monsieur Tovo ANDRIAMBELOMA, pour ses précieux conseils et son dévouement, ainsi que pour sa contribution à la correction de ce mémoire.

Après cela, je souhaite exprimer ma plus grande gratitude envers tous les professeurs de l'Université IT pour leurs efforts constants en matière d'enseignement, qui constituent les principales raisons de ma réussite universitaire.

Je souhaite également remercier mes collègues de travail chez Samifin pour leur accueil chaleureux, qui a grandement facilité mon intégration dans l'entité, chose qui a beaucoup contribué à ma productivité durant mon stage.

Enfin, je tiens à remercier ma famille pour leur soutien constant, leurs encouragements, leur force morale et économique, ainsi que pour leurs précieuses suggestions qui ont grandement contribué à la réalisation de ce stage et de ce mémoire.

# Introduction

L’essentiel du travail que j’ai réalisé a porté sur le thème :

« Gestion des actifs »

Le développement des technologies de l'information a profondément transformé la gestion des actifs dans les entités. Avec l'évolution des outils numériques, il est désormais essentiel de disposer de solutions fiables et efficaces pour suivre et gérer les stocks ainsi que les états des actifs. Ce projet, intitulé « Gestion des actifs », vise à répondre à cet besoin en développant une plateforme innovante capable de répondre aux défis posés par cette nouvelle donne.

L'objectif principal de cette application est de fournir une vue d'ensemble sur les actifs matériels et immatériels d'une organisation, tout en optimisant leur gestion à travers des outils de suivi adaptés. Ce système permettra de mieux gérer les mouvements d'inventaire, les actifs déployés, ainsi que l'historique et les statistiques d'utilisation. De plus, l'application offre des fonctionnalités avancées telles que la gestion des utilisateurs, le suivi des immobilisations et d'autres outils de gestion des actifs.

L'existant, la gestion se fait principalement sur support papier ou sur fichier Excel, ce qui rend la traçabilité difficile et ralentit la prise de décision. En revanche, l'application « Gestion des actifs » permet une gestion centralisée, une réduction des erreurs humaines et une optimisation du suivi des mouvements des stocks et des immobilisations.

Notre application regroupe les fonctionnalités suivantes :

* Gestion des utilisateurs
* Gestion des biens
* Gestion des mouvements des stocks
* Gestion des immobilisations
* Notifications et recherche
* Tableau de bord

Afin de bien développer les travaux effectués, ce présent mémoire sera structuré en 3 parties

* La présentation du projet
* La réalisation du projet
* L'évaluation du projet et les connaissances acquises

# Présentation du projet

## Objectifs du projet

L'objectif principal de l'application « Gestion des actifs » réside dans son potentiel à transcender les limites et les contraintes inhérentes à l'utilisation exclusive de fichiers Excel pour le stockage des données relatives aux actifs. En remplaçant ces outils par une application dédiée, notre intention est de fournir une plateforme plus robuste et conviviale tout en assurant la cohérence des données. Cela permettra non seulement une gestion plus efficace des actifs matériels et immatériels au sein de l'organisation, mais offrira également une vue d'ensemble plus détaillée et interactive des stocks et immobilisations. Grâce à cette application, les responsables de la gestion des actifs auront un accès simplifié aux informations essentielles, telles que les mouvements d'inventaire, les états des actifs, ainsi que des statistiques détaillées sur l'utilisation et la valeur des actifs. Cette transition vise à améliorer la traçabilité, réduire les erreurs humaines et optimiser le suivi des mouvements des stocks et des immobilisations, dans le but ultime d'accroître l'efficacité et la transparence de la gestion des actifs au sein de l'entité.

## Planning de réalisation

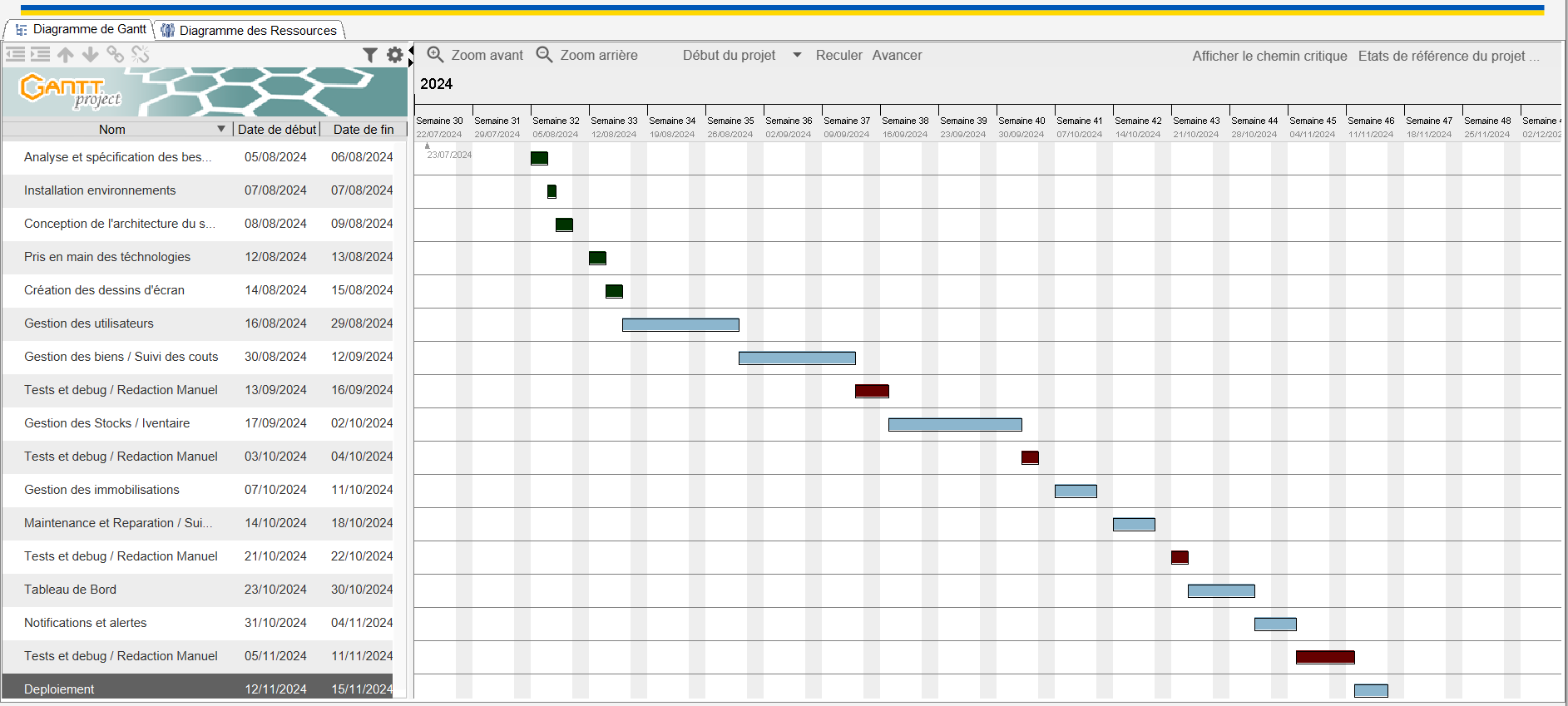


Figure 1 : Diagramme de Gantt pour la planification du projet

## Architecture du projet

Le projet « Gestion des actifs » utilise une **architecture monolithique**, un modèle où l'ensemble de l'application est regroupé dans un seul bloc. Dans ce modèle, le client envoie des requêtes HTTP[[3]](#footnote-3) au serveur via des URL[[4]](#footnote-4)s spécifiques pour accéder à des ressources. Le serveur reçoit ces requêtes et, en fonction des données nécessaires, interagit avec la base de données pour récupérer ou manipuler des informations. Les données renvoyées par le serveur sont encapsulées dans un objet IActionResult, qui permet de retourner différents types de résultats comme des vues, des redirections ou des réponses avec des données au format JSON[[5]](#footnote-5).

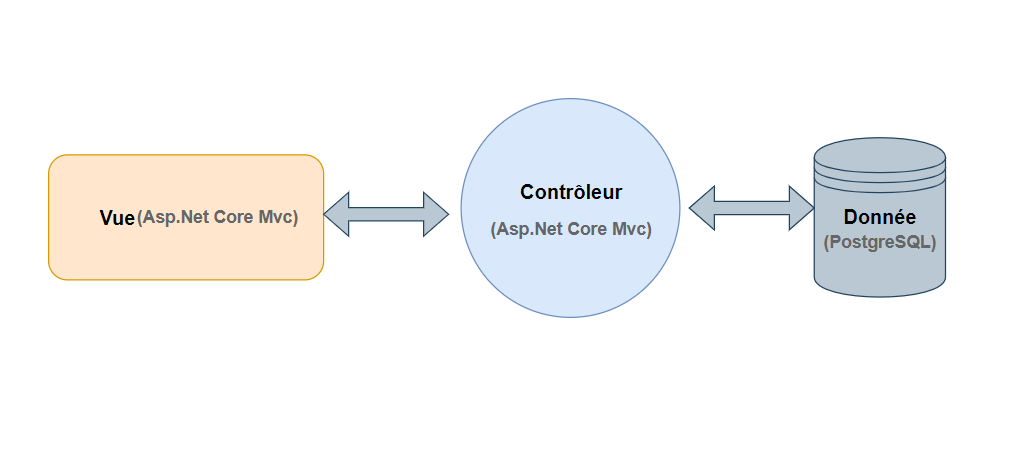


Figure 2 : Structure générale du projet.

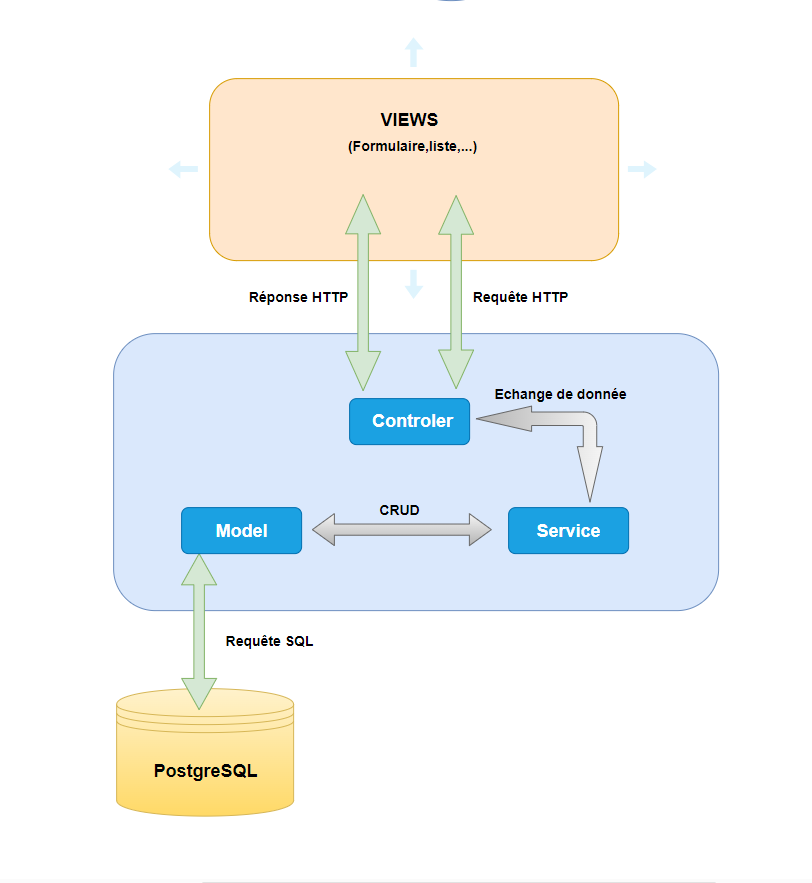


Figure 3 : Structure détailés du projet

Les deux figures précédentes offrent un aperçu du fonctionnement global de l'application. Avant leur insertion dans la base de données PostgreSQL, les données subissent un processus de traitement. Une fois reçues, ces données sont traitées, analysées et ensuite présentées dans l'interface graphique de l'application

## Technologies utilisées

Pour la conception de l'application « Gestion des actifs », le choix technologique s'est porté sur :

1. ASP.NET Core MVC[[6]](#footnote-6) pour gérer à la fois le front-end et le back-end. Cette architecture MVC permet de structurer le code en séparant la logique métier, la gestion des données et l'interface utilisateur, tout en offrant un rendu dynamique des pages web.
2. Entity Framework Core comme ORM[[7]](#footnote-7) pour faciliter l’interaction avec la base des données.
3. PostgreSQL pour la gestion des données, garantissant une haute performance et une comptabilité conforme aux exigences de l'application.
4. **ASP.NET Core MVC**

ASP.NET Core MVC est un framework web moderne et puissant développé par Microsoft. Il s'agit d'une version améliorée et restructurée d'ASP.NET MVC, offrant une architecture plus légère et modulaire. Ce framework permet de construire des applications web dynamiques en utilisant le modèle de conception MVC, favorisant ainsi une séparation claire des préoccupations.

Parmi les principaux avantages d'ASP.NET Core MVC figurent sa performance élevée, son caractère open-source et sa comptabilité multiplateforme. Il offre également un excellent support pour la programmation asynchrone, améliorant ainsi les performances des applications. De plus, ASP.NET Core MVC intègre des fonctionnalités de sécurité avancées et facilite le développement rapide grâce à des outils puissants comme Visual Studio et Rider.

ASP.NET Core MVC est particulièrement adapté pour le développement d'applications web robustes et évolutives. Il permet d'offrir une large gamme de bibliothèques et de frameworks tiers pour étendre ses capacités. Avec sa flexibilité et ses performances, ASP.NET Core MVC constitue un choix solide pour les développeurs souhaitant construire des applications web modernes et performantes.

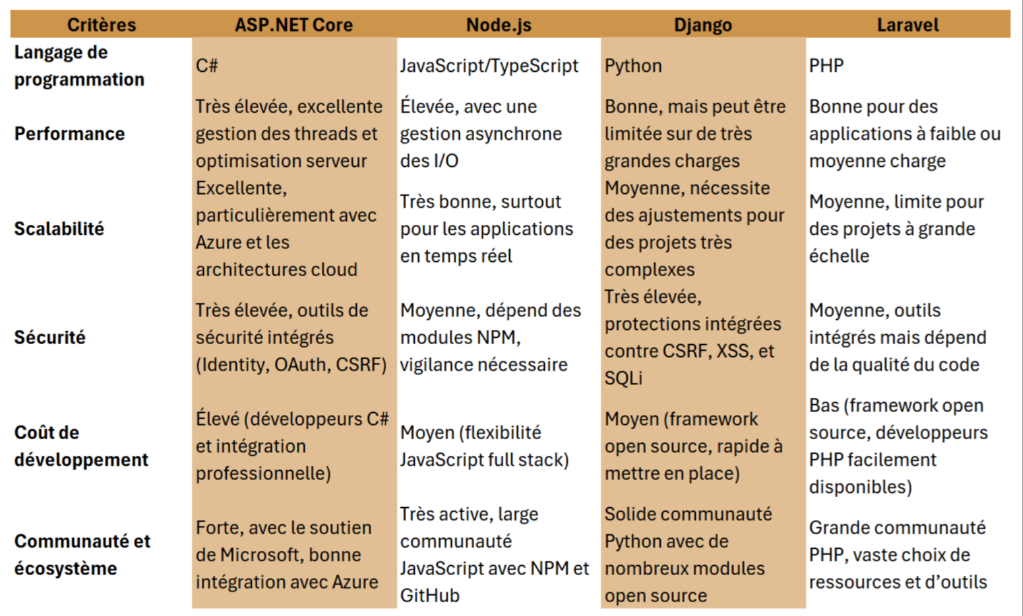


Figure 4 : Comparaison d'ASP.NET Core avec ses alternatives

|  |  |  |
| --- | --- | --- |
| **Critère** | **ASP.NET Core** | **Spring Boot** |
| **Langage** | C# | Java |
| **Performance** | Très élevée, conçu pour une exécution rapide | Bonnes performances, mais légèrement plus lentes que ASP.NET Core |
| **Facilité de démarrage** | Bon, surtout avec Visual Studio | Excellente, grâce à la configuration automatique |
| **Sécurité** | |  | | --- | |  |  |  | | --- | | Très sécurisé avec des outils intégrés | | Très sécurisé grâce à Spring Security |
| **Démarrage** | Très rapide | Moins rapide que ASP.NET Core |

Tableau 1 :Tableau comparatif entre ASP.NET Core et Spring Boot.

1. **Entity Framework Core**

Entity Framework Core (EF Core) est une version légère, extensible et open-source d'Entity Framework, une technologie populaire d'accès aux données. Il s'agit d'un mappage objet-relationnel (ORM) qui permet aux développeurs .NET de travailler avec une base de données en utilisant des objets .NET. EF Core élimine la nécessité d'écrire le plus grand partie du code d'accès aux données typiquement nécessaire, simplifiant ainsi considérablement l'interaction avec les bases de données.

EF Core présente plusieurs avantages pour les développeurs .NET. Il permet une approche orientée objet pour interagir avec les bases de données relationnelles, offre des fonctionnalités avancées comme les filtres de requête globaux et les conversions de valeurs personnalisées. De plus, son intégration fluide avec ADO[[8]](#footnote-8).NET et sa prise en charge des paramètres pour prévenir les injections SQL[[9]](#footnote-9) contribuent à la sécurité et à la performance des applications. En tant qu'outil puissant et flexible, EF Core est particulièrement adapté au développement d'applications .NET modernes et complexes.

1. **PostgreSQL**

PostgreSQL représente un choix solide pour la gestion de bases de données au sein de notre application. Réputé pour sa fiabilité, sa robustesse et sa conformité aux standards, PostgreSQL offre une large gamme de fonctionnalités avancées pour le stockage et la manipulation des données. Sa capacité à gérer des charges de travail complexes tout en maintenant des performances élevées en fait un choix idéal pour les applications exigeantes. La flexibilité de PostgreSQL lui permet de s'adapter à différents scénarios, que ce soit des applications web, mobiles ou d'entreprise. De plus, sa communauté active assure un rapport continu, des mises à jour régulières et une documentation riche, garantissant ainsi la stabilité et l'évolution de notre système de gestion de base de données. En optant pour PostgreSQL, nous visons à garantir la sécurité, la performance et la pérennité de la gestion de nos données au sein de l'application.

Parmi les alternatives à PostgreSQL, MySQL se distingue par son approche relationnelle robuste et sa facilité d'utilisation, offrant une solide plateforme de gestion de base de données avec une licence double et des versions communautaires gratuites. Cependant, il existe des différences notables en termes de fonctionnalités et de modèle de licence entre ces deux systèmes de gestion de base de données.

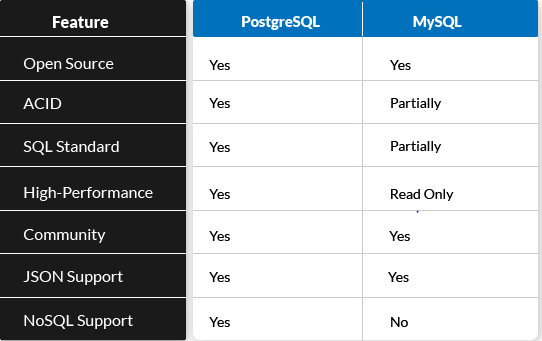


Figure 5 : Comparaison PostgreSQL avec MySQL

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Critère** | |  | | --- | |  |  |  | | --- | | **PostgreSQL** | | **MySQL** |
| **Conformité aux standards SQL** | Très conforme aux normes SQL. | Moins conforme, certaines différences par rapport au standard SQL. |
| **Types de données avancés** | Supporte des types de données spécialisés (JSON, géographique, etc.). | Moins riche en types de données natifs. |
| **Requêtes complexes et jointures** | Supporte les jointures complexes. | Moins performant sur les jointures multiples. |
| **Performance** | Très performant avec des données complexes et des requêtes avancées. | Meilleur pour des requêtes simples et une gestion rapide des données transactionnelles. |

Tableau 2 :Tableau comparatif entre PostgreSQL et MySQL.

# Réalisation de l’application

## Analyse et conception

### Analyse de l'existant

Avant l’introduction de la nouvelle application « Gestion des actifs », les employés du service logistique géraient manuellement les stocks et les immobilisations en utilisant Excel ou d'autres éditeurs de documents. Cette approche manuelle augmentait le risque d’erreurs, rendait la recherche d’informations longue et limitait l'efficacité opérationnelle. Cette nouvelle application marque donc un tournant majeur en automatisant ces processus et en fournissant une solution centralisée, plus fiable et plus rapide pour la gestion des stocks et des immobilisations.

### Conception de l’application

La conception est une étape fondamentale pour le développement d’une application. En effet, c’est durant la conception qu’on analyse la grandeur du projet, que l’on étudie les problèmes majeurs à qui l’on pourra faire face pendant le développement et qui détermine les tâches à effectuer pour atteindre les objectifs du projet. Tout cela pour pouvoir mettre en place un bon planning de développement.

A la réalisation de notre développement, on a créé au total :

* Nombre de tables, views et functions créés : 45
* Nombre de contrôleurs créés : 19
* Nombre de models et services créés : 53
* Nombre d'écrans créés : 45

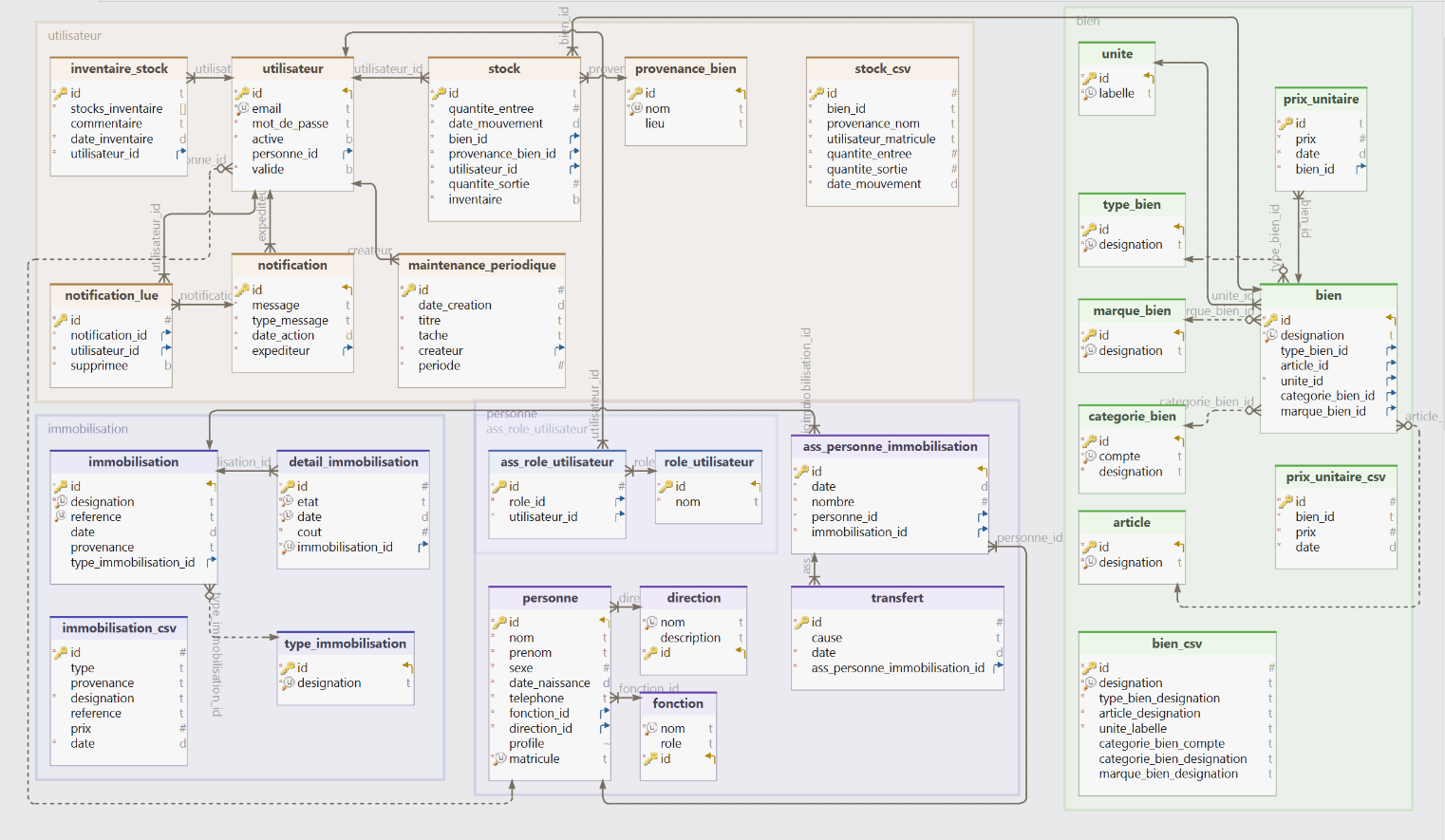


Figure 6 : Script MCD des tables

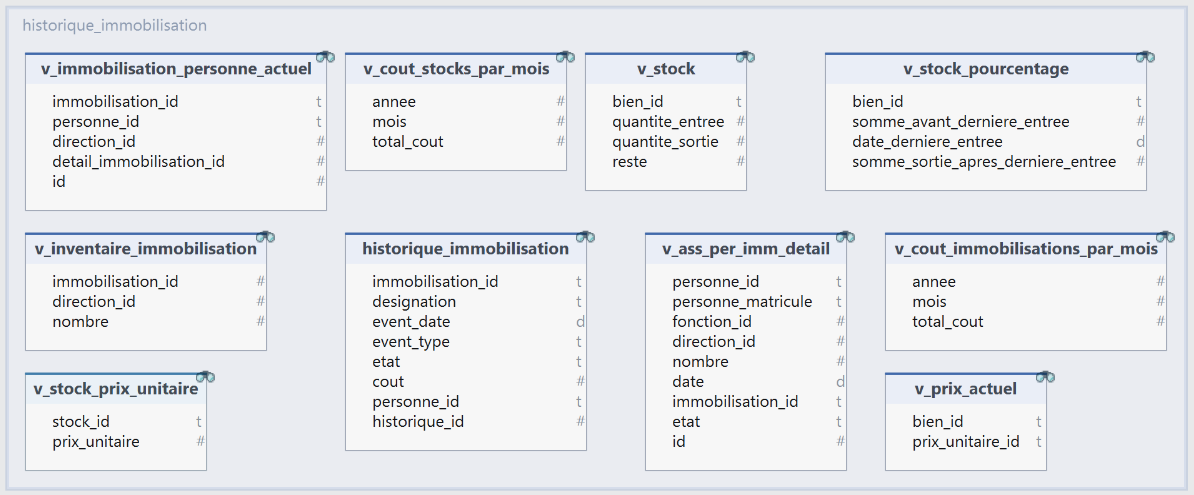


Figure 7 : Script des views

## Développement par fonctionnalité ou module

### Gestion d'utilisateur

#### Inscription

* **Description et objectif**

Dans le cadre des règles de gestion de l’entité, le processus d’inscription est strictement contrôlé et réservé aux administrateurs. Les utilisateurs ne peuvent pas s’inscrire de manière autonome ni accéder à une fonctionnalité d’inscription au sein de l'application. Cette restriction vise à assurer un contrôle total sur la gestion des comptes utilisateurs et à prévenir tout accès non autorisé à l'application. L'objectif est de garantir que seuls les employés autorisés puissent utiliser le système, tout en appliquant des mesures de sécurité appropriées.

* **Scenario**

Les administrateurs sont chargés de créer manuellement les comptes des utilisateurs en saisissant des informations personnelles telles que : nom, prénom, poste, direction, sexe, date de naissance, et rôle (utilisateur, admin, super admin). Une fois le compte créé, un identifiant et un mot de passe provisoire sont générés et communiqués aux utilisateurs, soit en main propre, soit par email. Après l'inscription, le mot de passe est chiffré à l'aide de l'algorithme PBKDF2[[10]](#footnote-10) fourni par ASP.NET Core Identity, qui applique plusieurs milliers d'itérations de hachage avec un sel unique. Cette approche renforce la sécurité en protégeant les mots de passe contre les attaques, notamment de type rainbow table.

Cette gestion centralisée des inscriptions permet de restreindre l’accès à l'application aux seuls employés de Samifin, garantissant ainsi que seuls des utilisateurs autorisés peuvent accéder au système.

Gestion des formulaires d'ajout d'utilisateurs :

* Remplissage obligatoire des champs : nom, prénom, date de naissance, matricule, direction, fonction, téléphone, email, sexe, rôles
* Validation des règles de gestion :
* Âge minimum de 18 ans
* Matricule unique
* Format téléphonique conformément aux normes
* Email unique et conforme au format standard
* Attachement d'au moins un rôle (SuperAdmin, Admin, Utilisateur)
* Affichage de messages d'erreur en cas de non-conformité.
* Création automatique d'un mot de passe temporaire.
* Possibilité  d’ajout de la photo de profil par l'administrateur.

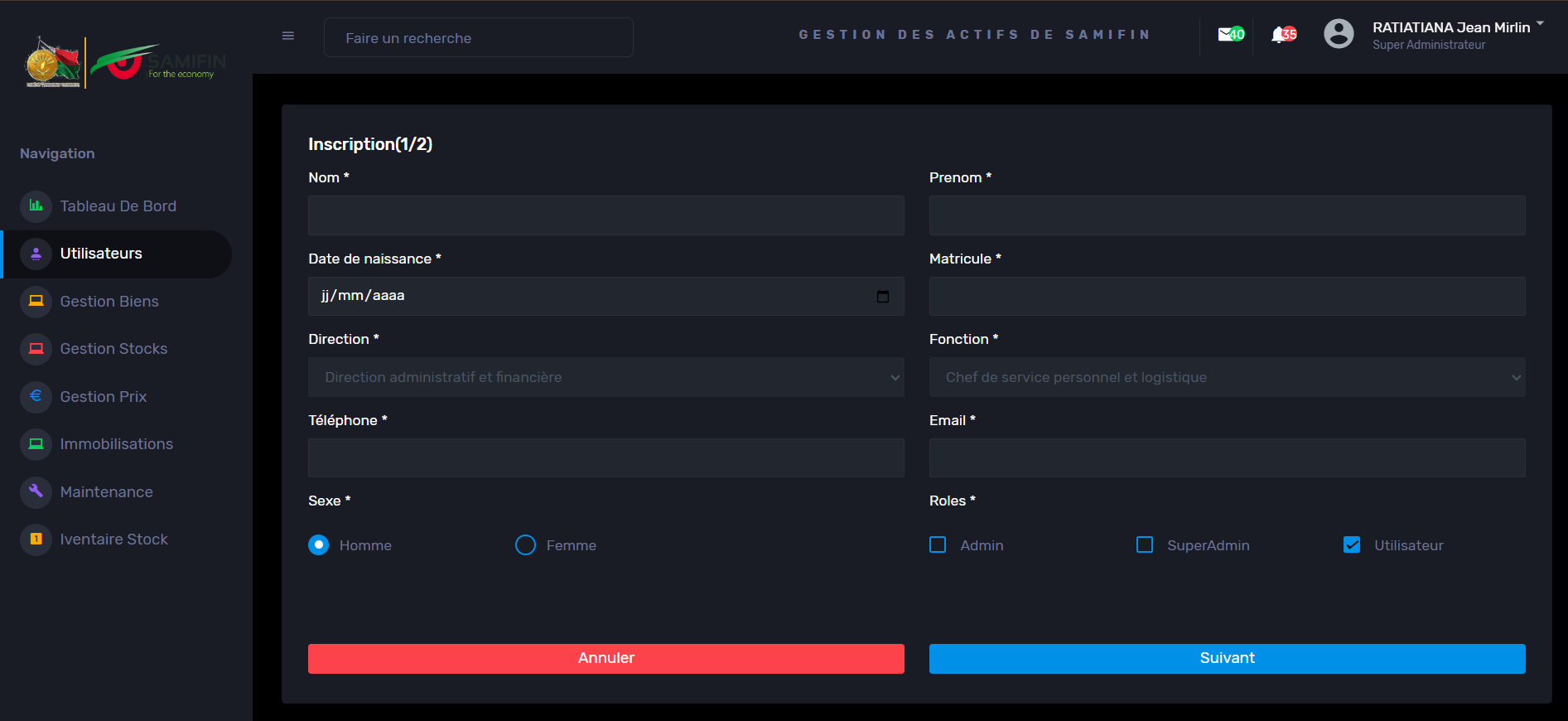


Figure 8 : Création d'utilisateur

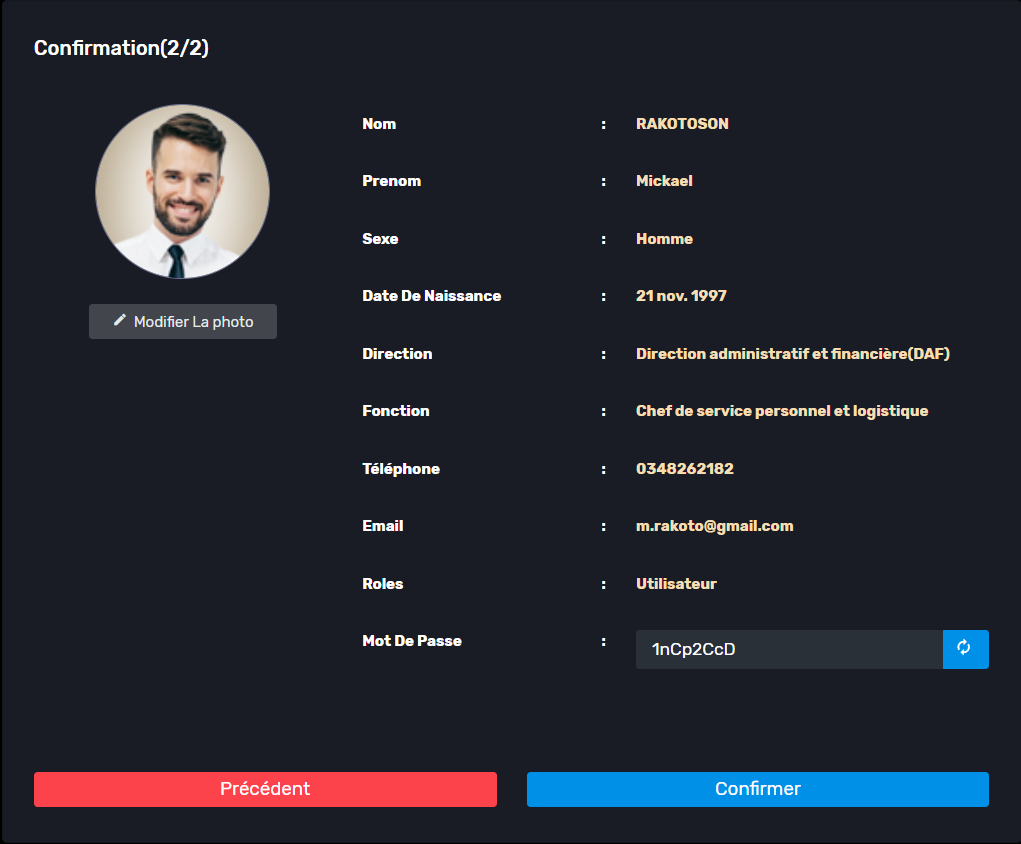


Figure 9 : Confirmation de la création d’utilisateur

#### Connexion

* **Description et objectif**

Le processus de connexion vise à renforcer la sécurité des comptes utilisateurs en exigeant un changement de mot de passe lors de la première connexion. L'objectif est de protéger les comptes en imposant une mise à jour immédiate du mot de passe provisoire. De plus, la gestion des rôles, qui est intégrée via ASP.NET Core Identity, permet de définir des niveaux d'accès différenciés, garantissant que chaque utilisateur accède uniquement aux fonctionnalités appropriées en fonction de son rôle (utilisateur, admin ou super admin).

* **Scenario**

Lors de sa première connexion après récupération des identifiants, l'utilisateur est redirigé vers une page de modification du mot de passe ; tant que le mot de passe provisoire n'est pas changé, l'application bloque toute action pour des raisons de sécurité. Le nom, prénom, et rôle de l'utilisateur sont stockés dans des claims, permettant de gérer son identité et ses droits. Les données volumineuses, comme la photo de profil, sont conservées dans la HttpSession pour rester disponibles côté serveur durant la session de l'utilisateur.

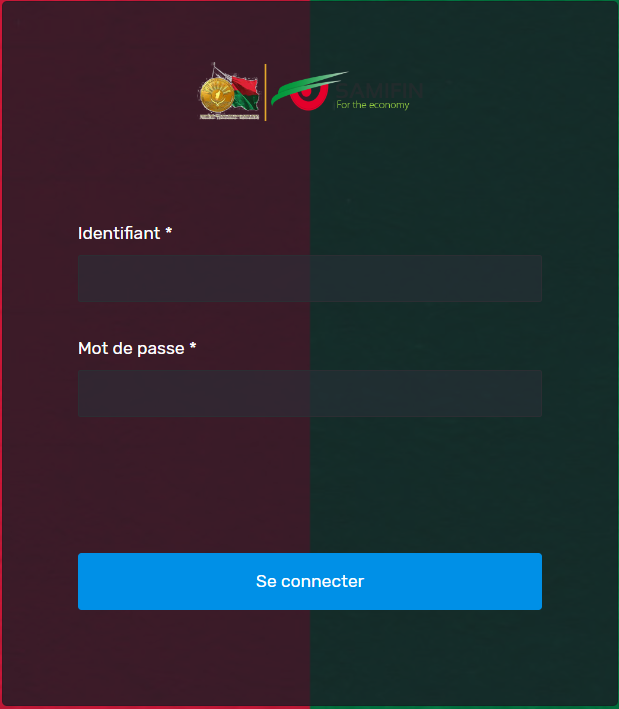
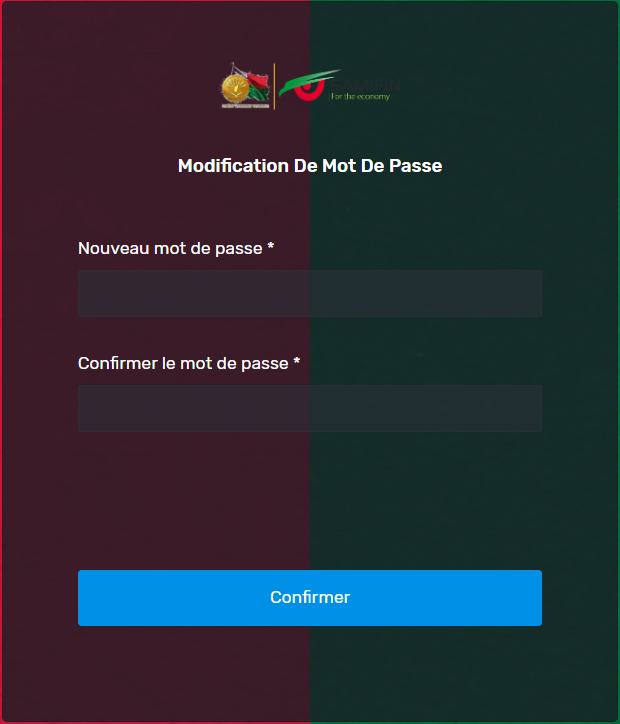
Les rôles sont gérés via ASP.NET Core Identity, qui permet de définir différents niveaux d’accès. Dans votre application, les trois rôles définis sont les suivants :

* Utilisateur : Accès limité aux fonctionnalités de base.
* Admin : Accès à des fonctionnalités avancées de gestion.
* Super admin : Accès total avec des droits de gestion sur tous les aspects de l’application, y compris l’administration des utilisateurs et des rôles.



Figure 10 : Page d'accès refusé pour rôle

L’application utilise ces rôles pour restreindre ou autoriser l’accès à certaines parties de l’interface, en fonction des permissions attribuées. Cette gestion des rôles garantit que seuls les utilisateurs autorisés peuvent accéder aux fonctionnalités sensibles.

Figure 11,12 : Connexion et Modification mot de passe

En conclusion, le processus d’inscription et de connexion de l’application a été conçu pour garantir la sécurité et l’intégrité des comptes utilisateurs. En limitant l’inscription aux administrateurs et en imposant la modification du mot de passe initial, l’application prévient les accès non autorisés. L’utilisation de claims pour stocker les informations essentielles et de rôles pour définir les niveaux d’accès permet une gestion fine des permissions, adaptée aux besoins spécifiques de l’entité. De plus, le recours à la HttpSession pour conserver les données volumineuses, comme le profil utilisateur, assure une gestion efficace et performante des sessions, tout en maintenant une expérience utilisateur fluide et sécurisée.

#### Gestion des utilisateurs

* **Description et objectif**

La gestion des utilisateurs est un composant essentiel du système, responsable de la création, la gestion et la surveillance des comptes utilisateurs. Ses principaux objectifs sont :

* Permettre aux administrateurs de gérer efficacement la base d'utilisateurs de l'application.
* Assurer la sécurité et l'intégrité des données personnelles des utilisateurs.
* Offrir une expérience utilisateur fluide et sécurisée.
* Implémenter un système de rôle et de permission robuste.
* Faciliter la création et la modification des profils d'utilisateur.
* **Scenario**

1. Affichage de la liste des utilisateurs avec leurs profils, emails, matricules et statuts.
2. Affichage et mise à jour des informations personnelles de l'utilisateur :

* Accès aux détails complets via le bouton «Voir »
* Modification des informations personnelles via le bouton «Modifier »
* Gestion des mot de passes oubliés via le bouton «Mot De Passe »: c’est de régénérer un nouveau mot de passe temporaire par l’administrateur.

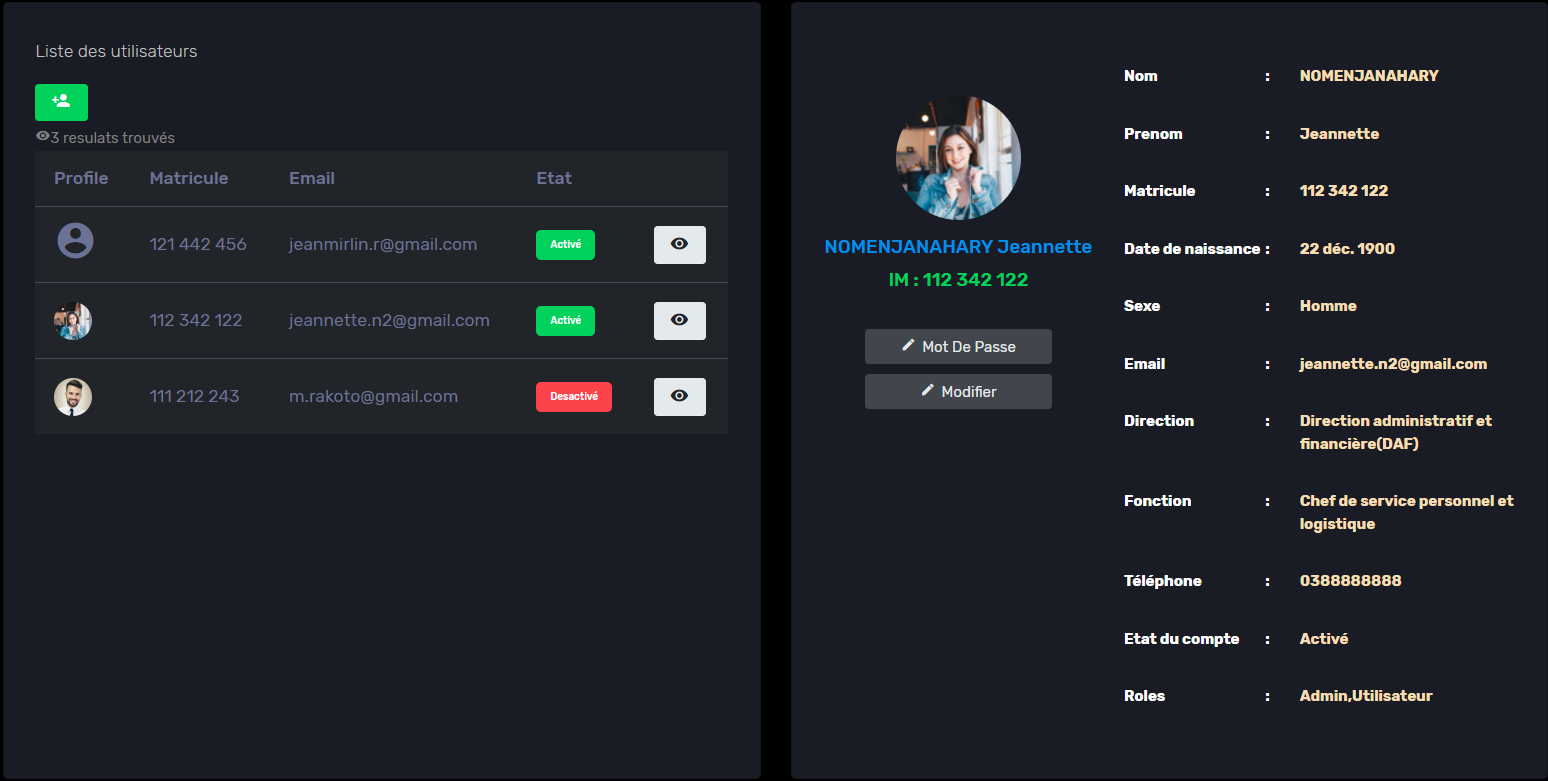


Figure 12 : Liste des utilisateurs et leurs profiles

#### Profile d’utilisateur

* **Description et objectif**

Une page dédiée à l’affichage des informations personnelles de l’utilisateur, offrant la possibilité de modifier la photo de profil et le mot de passe.

* **Scenario**

1. Pour changer la photo, l’utilisateur clique sur l’icône de caméra, ce qui lui permet de sélectionner une image depuis son ordinateur. Une fois l’image choisie, une page de confirmation s’affiche. L’utilisateur peut alors cliquer sur le bouton « Mettre à jour » pour valider la modification.
2. L’utilisateur peut changer son mot de passe en cliquant sur le bouton « Modifier Mot de Passe ». Il doit entrer son mot de passe actuel, puis le nouveau mot de passe, qu’il saisira à nouveau pour confirmer.
3. Seuls les administrateurs peuvent modifier les informations personnelles complètes de l’utilisateur. Si un utilisateur sans privilèges administratifs tente de modifier des informations autres que la photo de profil et le mot de passe, il sera redirigé vers une page d’accès refusé.

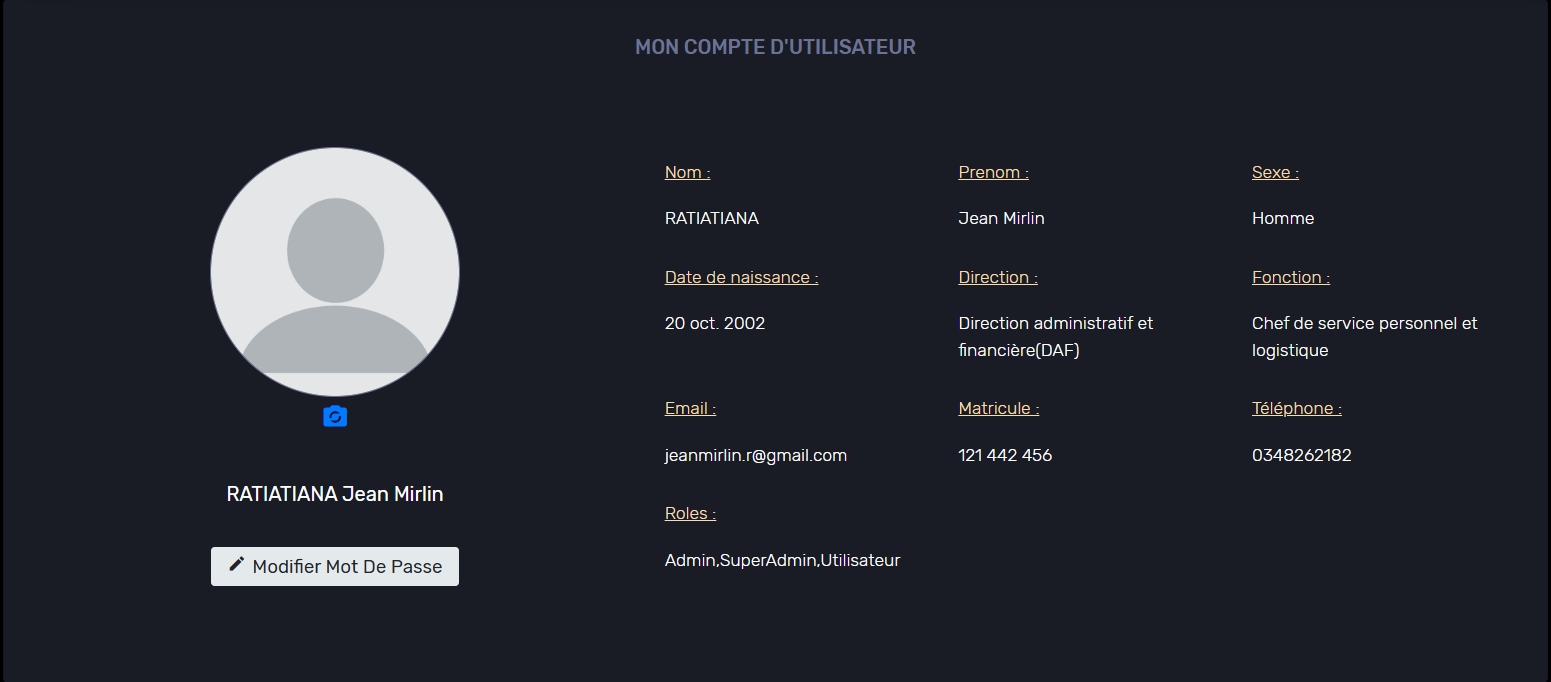


Figure 13 : Profil de l’utilisateur connecté

### Gestion des biens

#### Insertion des biens

* **Description et objectif**

La fonctionnalité d'insertion des biens permet à l'utilisateur de gérer l'ajout de nouveaux actifs au système. Deux options sont proposées pour répondre à différents besoins :

* La première option permet l'insertion manuelle d'un bien avec ses caractéristiques, telles que la catégorie, l'article, le type, la marque, et l'unité.
* La seconde option permet l'importation de plusieurs biens simultanément via un fichier Excel. Cette méthode est particulièrement utile pour accélérer le processus d'ajout en cas de volumes importants, tout en garantissant une saisie précise des informations.

L'objectif est de faciliter et d'optimiser l'enregistrement des biens, qu'il s'agisse d'une saisie unitaire ou massive, tout en offrant une flexibilité dans l'ajout des caractéristiques si elles ne sont pas déjà présentes dans le système.

* **Scenario**

L'utilisateur sélectionne les caractéristiques du bien à ajouter, telles que la catégorie, l'article, la marque, le type, l'unité et la désignation. Si certaines caractéristiques (par exemple, une nouvelle marque ou un nouveau type) ne figurent pas encore dans la base de données, l'utilisateur a la possibilité de les créer directement depuis cette interface. Cela garantit une gestion souple des informations et la mise à jour continue des caractéristiques des biens.

Pour l'importation en masse, l'utilisateur télécharge un fichier au format Excel contenant les informations des biens à ajouter. Une fois le fichier sélectionné, une page de confirmation s'affiche, permettant à l'utilisateur de vérifier que les données importées sont exactes et conformes aux attentes. Cette étape de validation garantit l'intégrité des informations et évite l'insertion d'erreurs dans la base de données.

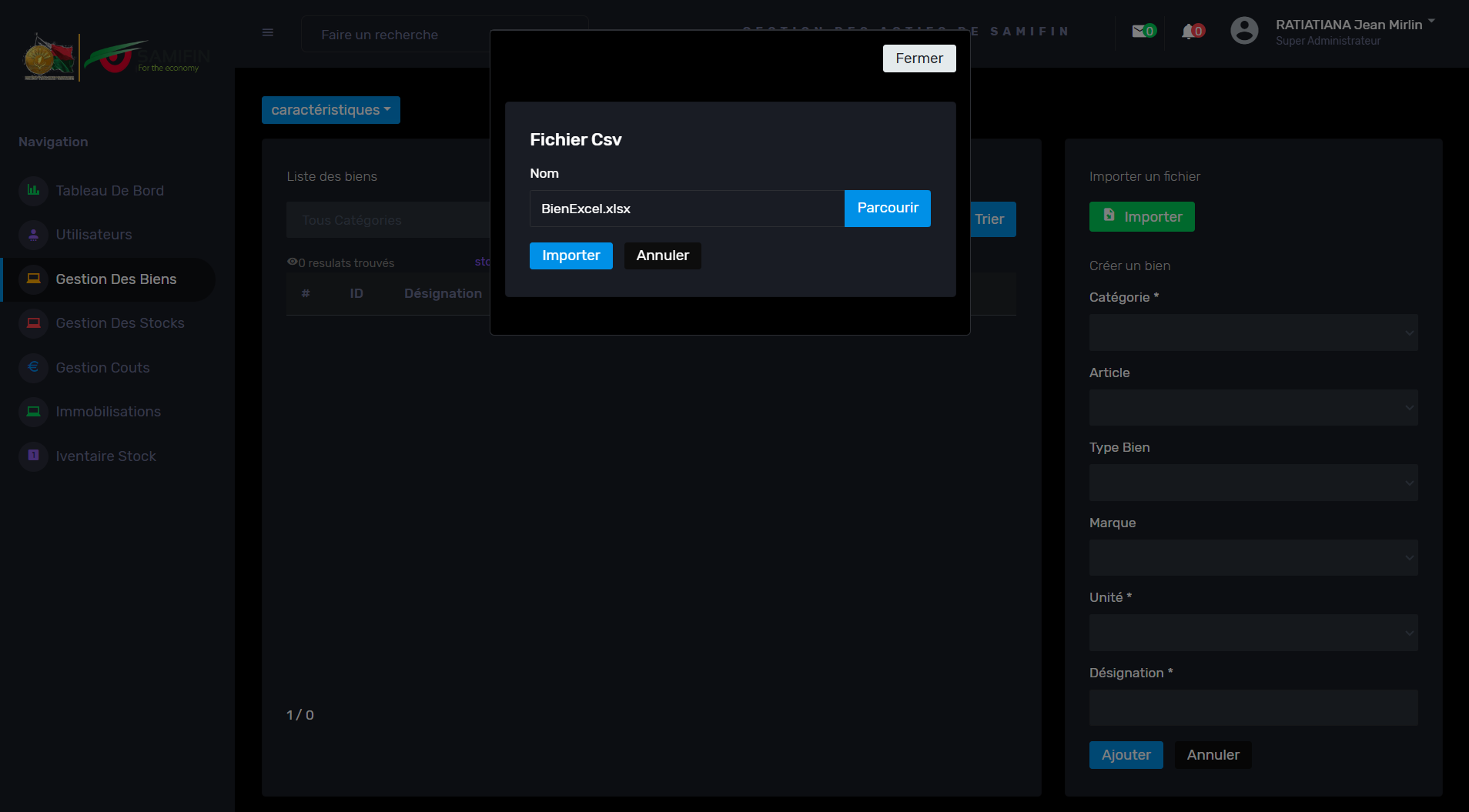


Figure 14 : Modal d'importation de fichier Excel

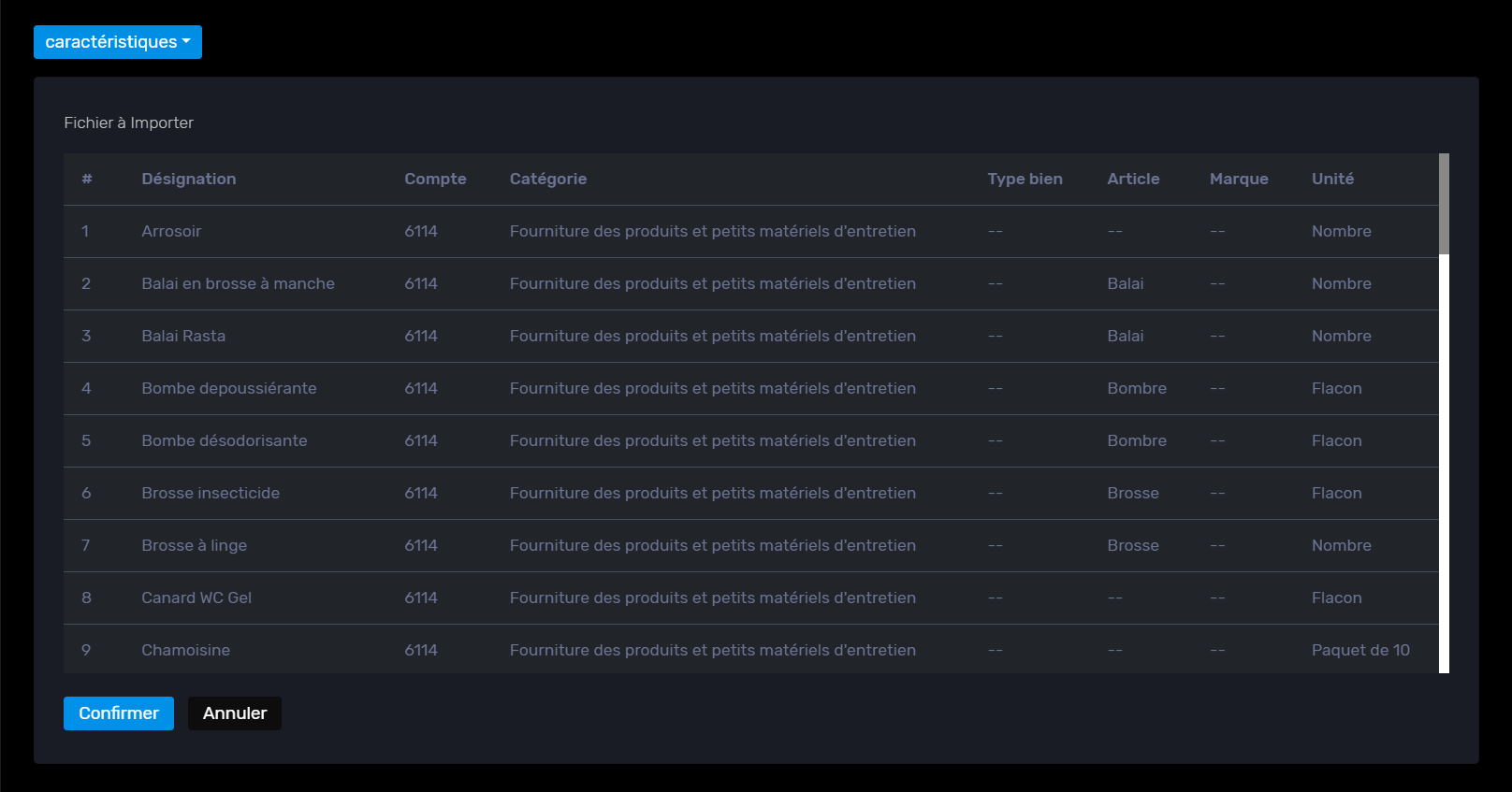


Figure 15 : Confirmation de l’importation de fichier Excel

#### Liste des biens

* **Description et objectif**

Cette page présente l'ensemble des biens disponibles dans le système, accompagnés de leurs principales caractéristiques (catégorie, article, type, marque, unité). Elle permet à l'utilisateur de modifier ou de supprimer un bien, à condition qu'aucun mouvement (stock ou transaction) lié à ce bien n'existe. Pour faciliter la recherche et l'accès rapide aux informations, un système de recherche multicritères ainsi que des options de tri par catégorie, type et unité sont disponibles.

L'objectif est de fournir une vue d'ensemble claire et facilement filtrable des biens, tout en assurant un contrôle strict sur la gestion des modifications et suppressions.

* **Scenario**

Les biens sont présentés dans un tableau paginé, avec un système de pagination permettant de naviguer facilement à travers les différentes pages. Le nombre de biens affichés par page est configurable et stocké dans un fichier XML[[11]](#footnote-11), permettant ainsi une modification flexible sans nécessiter de changements dans le code source. Ce paramètre peut être ajusté en fonction des besoins de l'utilisateur ou des performances souhaitées du système.

Un système de recherche multicritères permet de filtrer les biens selon plusieurs dimensions, notamment la catégorie, l'article, et le type. Ce mécanisme facilite la localisation rapide d'un bien spécifique ou d'un groupe de biens répondant à des critères définis.

Chaque ligne du tableau propose des actions permettant de modifier ou de supprimer un bien. Toutefois, la suppression n'est autorisée que pour les biens qui ne sont pas liés à des mouvements de stock ou d'autres transactions. Si un utilisateur tente de supprimer un bien ayant des mouvements associés, un message d'erreur s'affiche, interdisant l'opération. Cela garantit l'intégrité des données et évite toute suppression accidentelle d'un bien encore actif dans le système.

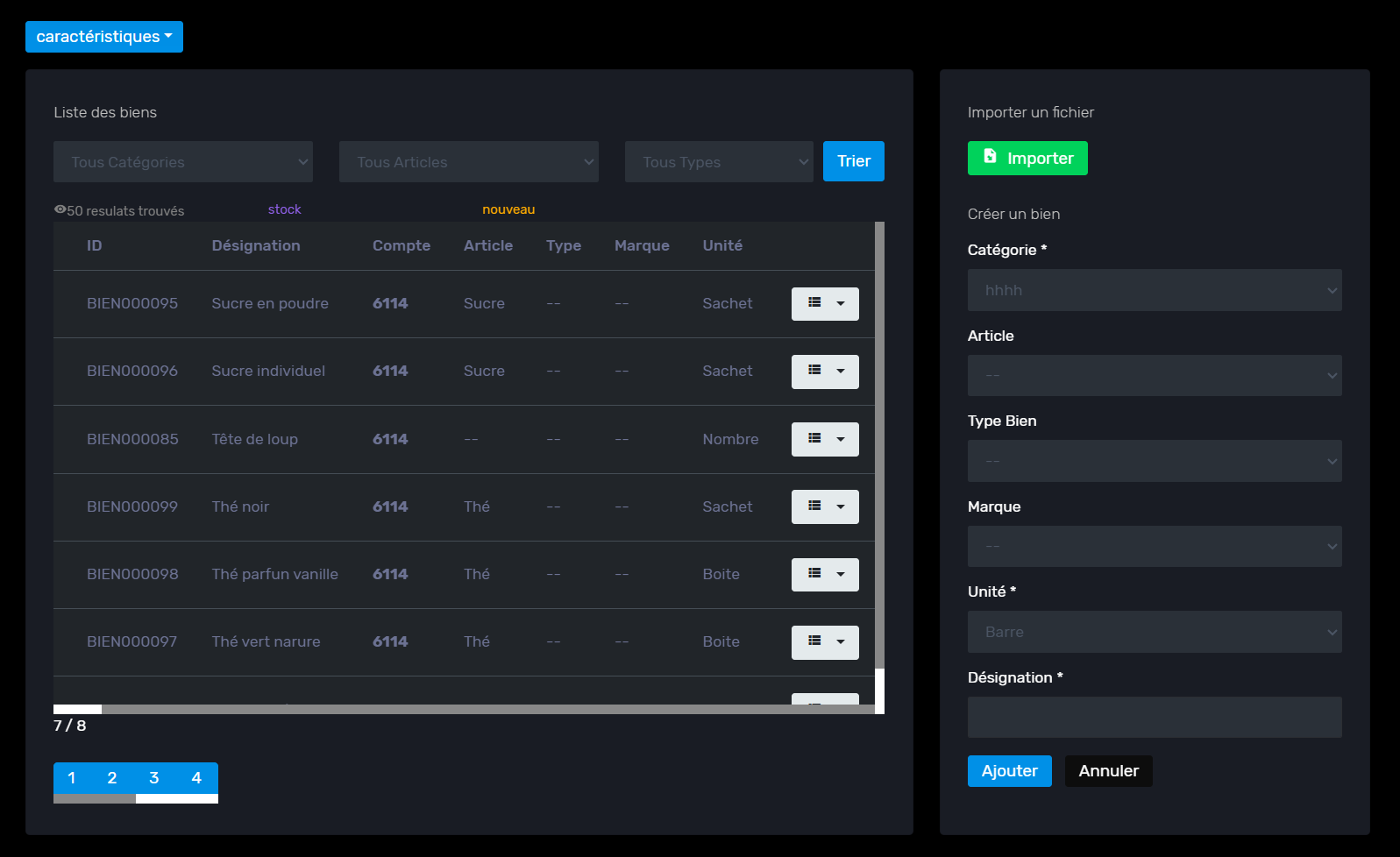


Figure 16 :Ajout et Liste des biens

#### Gestion des prix unitaires de biens

* **Description et objectif**

Permet la gestion des prix unitaires des biens, incluant la modification des prix en cas de variation, ainsi que le suivi et la gestion de l’historique des changements de prix.

* **Scenario**
* Sur la liste des biens et de leurs prix unitaires, l’utilisateur peut ajuster les prix en cas d’augmentation ou de diminution. L’historique des prix est également disponible, affiché par ordre décroissant de date, incluant les prix et les taux de variation. Il est possible de modifier ou supprimer un événement de l’historique en cas d’erreur d’insertion.
* Pour insérer les prix unitaires, l’utilisateur peut importer un fichier Excel contenant l’identifiant du bien, le prix et la date, ou bien saisir directement les données sur la page. Dans ce dernier cas, l’utilisateur sélectionne le bien et le prix unitaire, ce qui permet l’ajout de plusieurs biens avec leurs prix dans un tableau récapitulatif. Une fois la saisie terminée, l’utilisateur soumet le formulaire via le bouton « Ajouter », puis entre la date de modification via une fenêtre modale avant de confirmer.

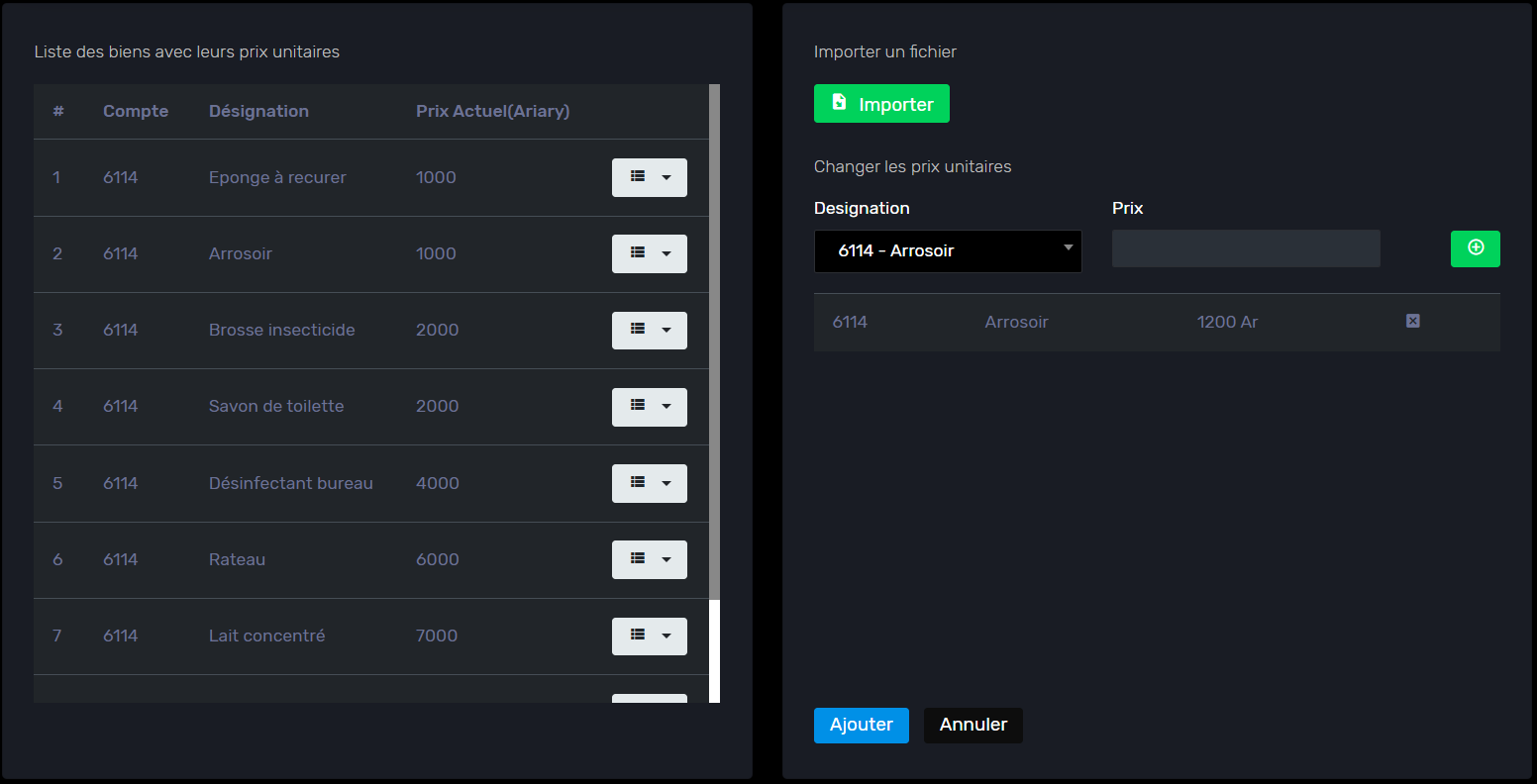


Figure 17 : Ajout des nouveaux prix et liste des biens avec leur prix actuel

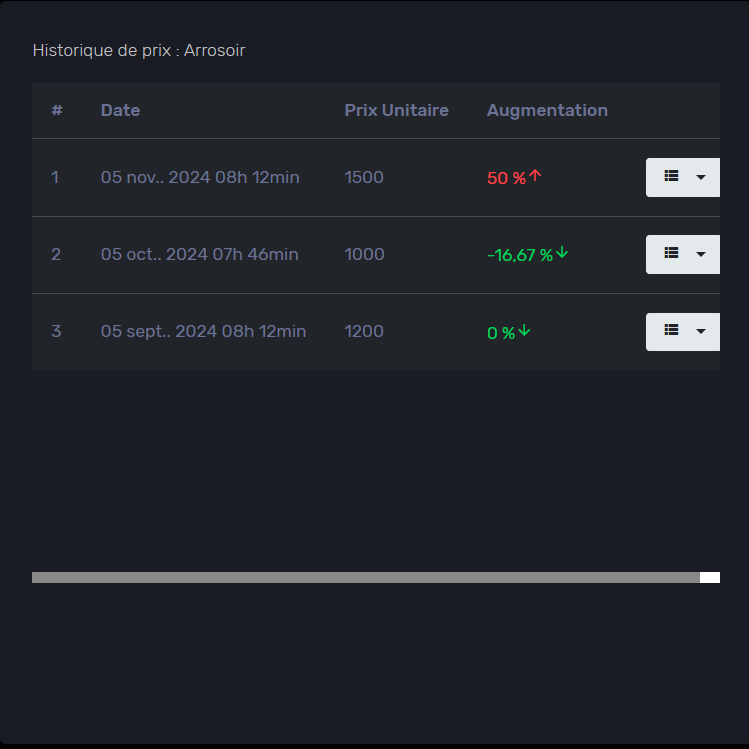


Figure 18 : Historiques des prix d’un bien

### Gestion des stocks

#### Mouvement des stocks

* **Description et objectif**

Cette fonctionnalité permet de gérer les mouvements de stocks, c'est-à-dire d'enregistrer les entrées et les sorties de stock pour chaque bien. Il est également possible d'importer un fichier Excel pour effectuer des mouvements en masse, ce qui accélère le processus d'insertion en cas de nombreuses transactions. En outre, une liste est disponible pour afficher les quantités actuelles des stocks de tous les biens, avec des filtres permettant de définir des plages de quantités, facilitant ainsi l'évaluation des niveaux de stock.

L'objectif de cette fonctionnalité est de garantir une gestion fluide et efficace des stocks tout en offrant des options d'optimisation, comme l'importation en masse et le filtrage avancé pour l'évaluation des quantités en stock.

* **Scenario**

1. **Insertion d’un mouvement de stock :**

Pour ajouter un mouvement de stock (entrée ou sortie), l'utilisateur doit :

* Sélectionner le bien concerné.
* Indiquer la provenance ou la destination du bien (selon qu'il s'agit d'une entrée ou d'une sortie).
* Saisir la quantité de biens impliquée.
* Spécifier la date du mouvement.
* Choisir le type de mouvement : entrée ou sortie.

Une fois les informations saisies, la validation du mouvement se fait en respectant les règles de gestion suivantes :

* La quantité ne peut pas être négative.
* Après une sortie de stock, la quantité restante doit toujours être positive ou nulle. Cela garantit que le stock ne passe jamais en négatif, assurant ainsi la cohérence des inventaires.

1. **Importation via fichier Excel :**

Pour accélérer le processus d'ajout, l'utilisateur peut importer un fichier au format Excel contenant plusieurs mouvements de stock. Ce fichier doit respecter un format prédéfini afin que le système puisse traiter correctement les données. Après l'importation, une page de confirmation s'affiche pour valider les mouvements avant leur enregistrement définitif.

1. **Affichage et filtrage des stocks :**

Une liste des biens, avec leurs quantités de stock actuelles, est disponible. L'utilisateur peut appliquer des filtres pour effectuer des recherches précises, comme afficher uniquement les biens dont les quantités sont comprises entre deux valeurs. Cela permet d'évaluer facilement les niveaux de stock critiques ou excédentaires, et d'anticiper des ajustements (réapprovisionnement ou déstockage).

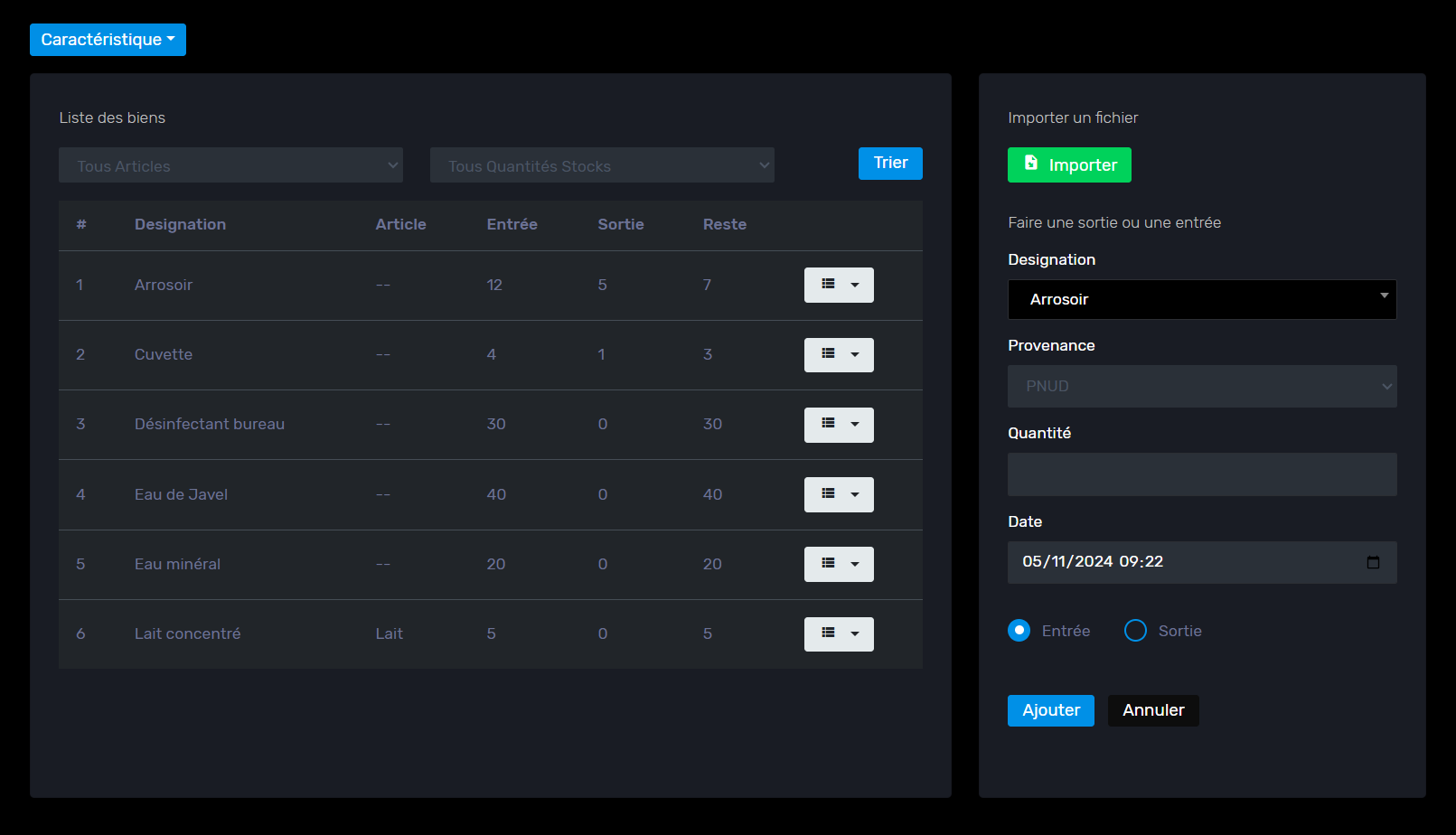


Figure 19 : Ajout et liste des biens avec leurs restes des stocks

#### Evaluation des stocks

* **Description et objectif**

La fonctionnalité d'évaluation des stocks permet d'analyser l'état des stocks sur une période définie, filtrée par catégorie ou bien spécifique. L'utilisateur peut obtenir deux types de résultats :

* **Résultat général** : un tableau synthétique affichant les biens avec la quantité initiale, le total des entrées, le total des sorties, et le stock restant sur la période sélectionnée.
* **Résultat détaillé** : des tableaux individuels pour chaque bien, indiquant la date des mouvements, l'éditeur, la provenance/destination, et les quantités déplacées (entrées/sorties).

En plus de la visualisation des résultats à l'écran, la fonctionnalité d'évaluation des stocks permet d'exporter ces données sous forme de fichiers PDF[[12]](#footnote-12), facilitant ainsi leur partage et archivage.

* **Scenario**

L'utilisateur sélectionne une période (date de début et date de fin), et peut choisir de filtrer l'évaluation par une catégorie spécifique ou l’ensemble des catégories ou un bien particulier au sein d'une catégorie donnée.

Il est possible de modifier ou d'annuler un mouvement de stock directement à partir des résultats détaillés, mais cette action est réservée uniquement aux administrateurs. Si un utilisateur non autorisé tente de le faire, il sera automatiquement redirigé vers une page d'accès refusé.

De plus, un bouton permet d'exporter les résultats de l'évaluation en PDF, facilitant ainsi le partage et l'archivage des données d'inventaire.

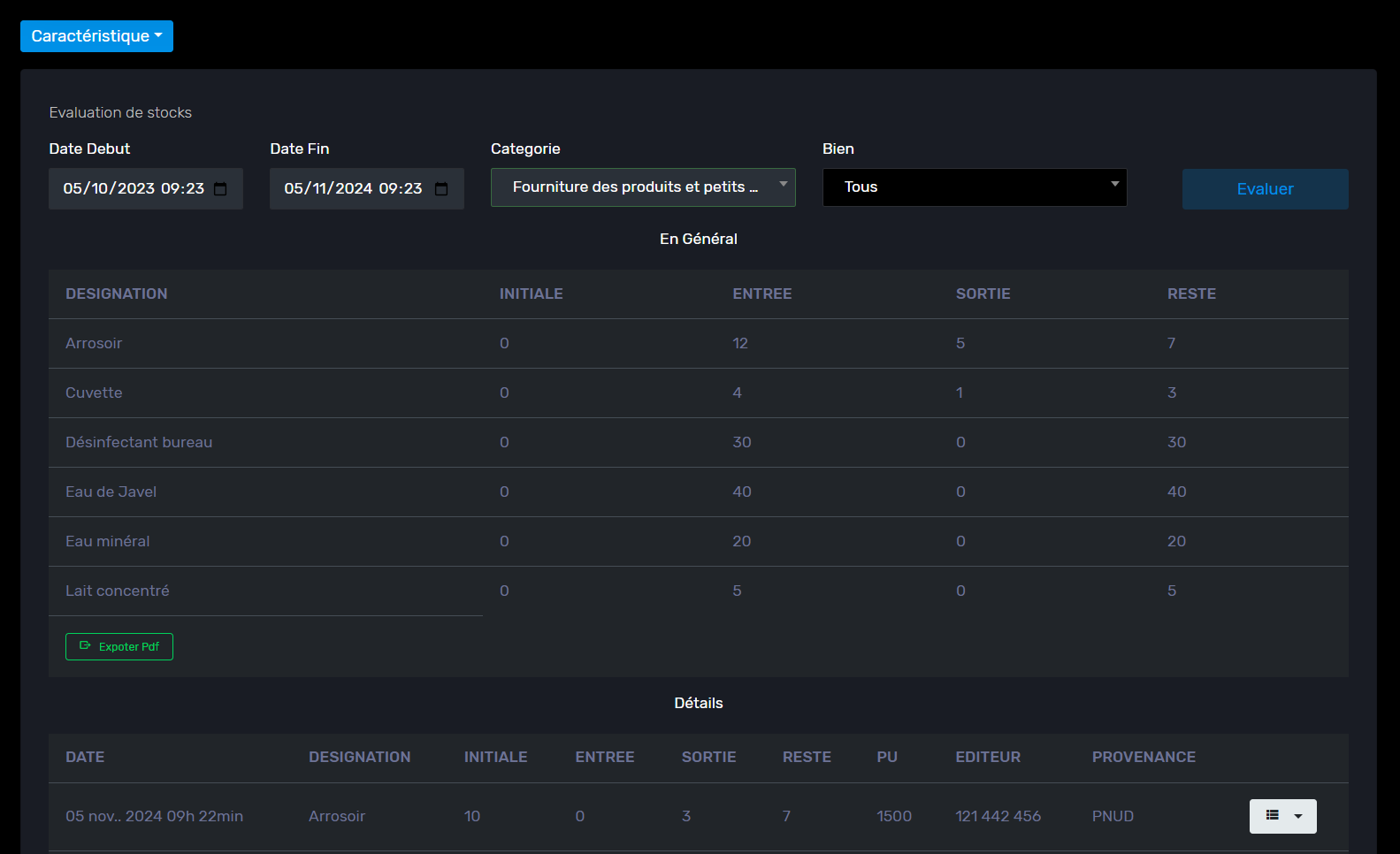


Figure 20 : Évaluation générale des stocks entre deux dates

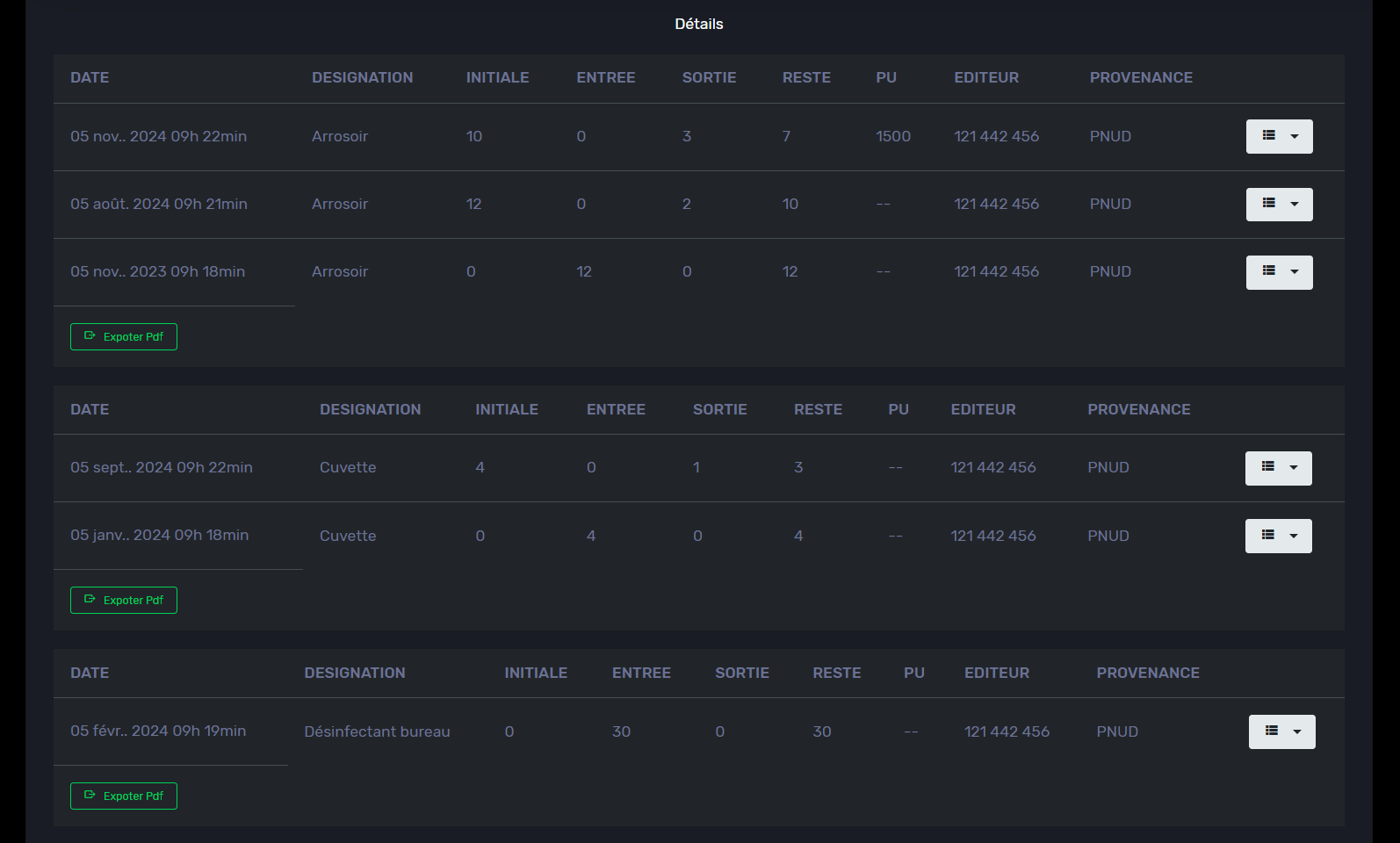


Figure 21 : Évaluation détaillée des stocks entre deux dates

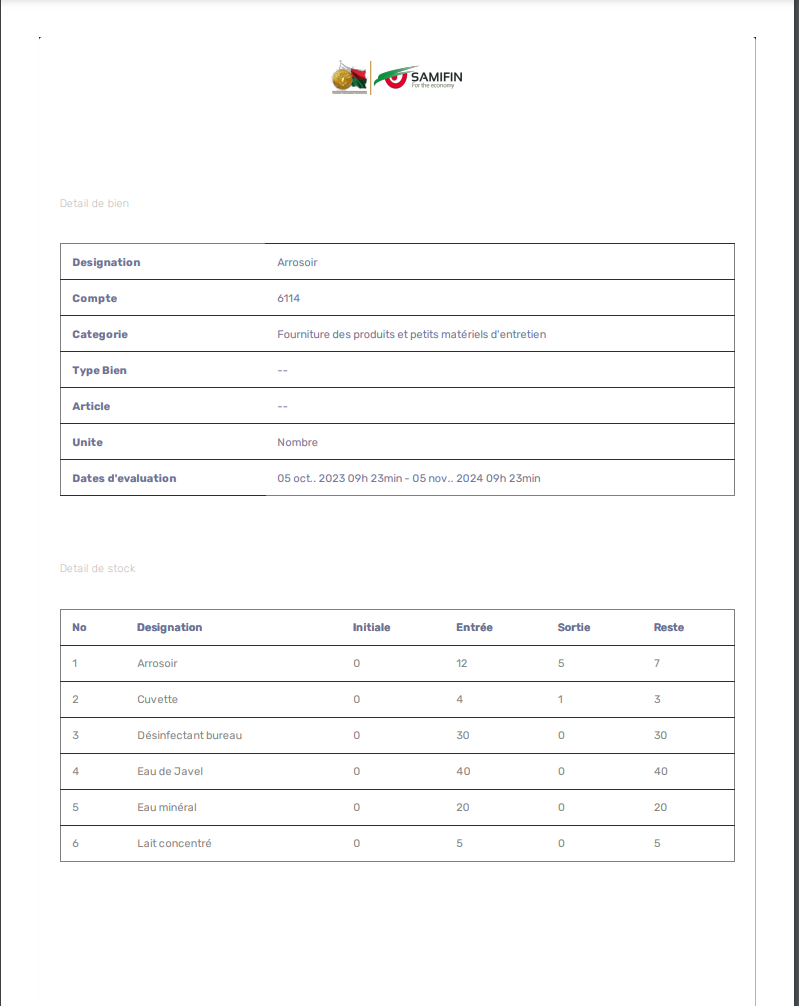


Figure 22 : Format PDF d’une évaluation des stocks

#### Inventaire des stocks

* **Description et objectif**

La fonctionnalité d'inventaire des stocks permet de réaliser un comptage physique des biens disponibles dans le système. L'utilisateur a la possibilité de comptabiliser tous les biens en enregistrant leurs quantités. Un bouton est prévu pour afficher les résultats de l'inventaire en comparant les données physiques avec celles du système.

L'objectif de cette fonctionnalité est de s'assurer que les quantités physiques correspondent aux enregistrements du système, tout en offrant un moyen de documenter les écarts et d'améliorer la gestion des stocks.

* **Scenario**

L'utilisateur effectue un comptage physique des stocks, en prenant soin de noter les quantités de chaque bien. Une fois le comptage terminé, il peut enregistrer l'inventaire dans le système en saisissant la date de l'inventaire ainsi qu'un commentaire ou une remarque pour contextualiser les résultats.

Lors de l'enregistrement de l'inventaire, les valeurs constatées sont prises comme nouvelles valeurs initiales pour tous les stocks, remplaçant ainsi les enregistrements précédents. Cela permet de mettre à jour le système avec des données récentes et précises.

Une page dédiée affiche la liste des inventaires effectués, comprenant les détails de chaque inventaire ainsi que les incohérences relevées lors du comptage. Cette fonctionnalité permet de garder un historique des mouvements de stocks et de suivre l'évolution des inventaires au fil du temps, facilitant ainsi les audits et les analyses.

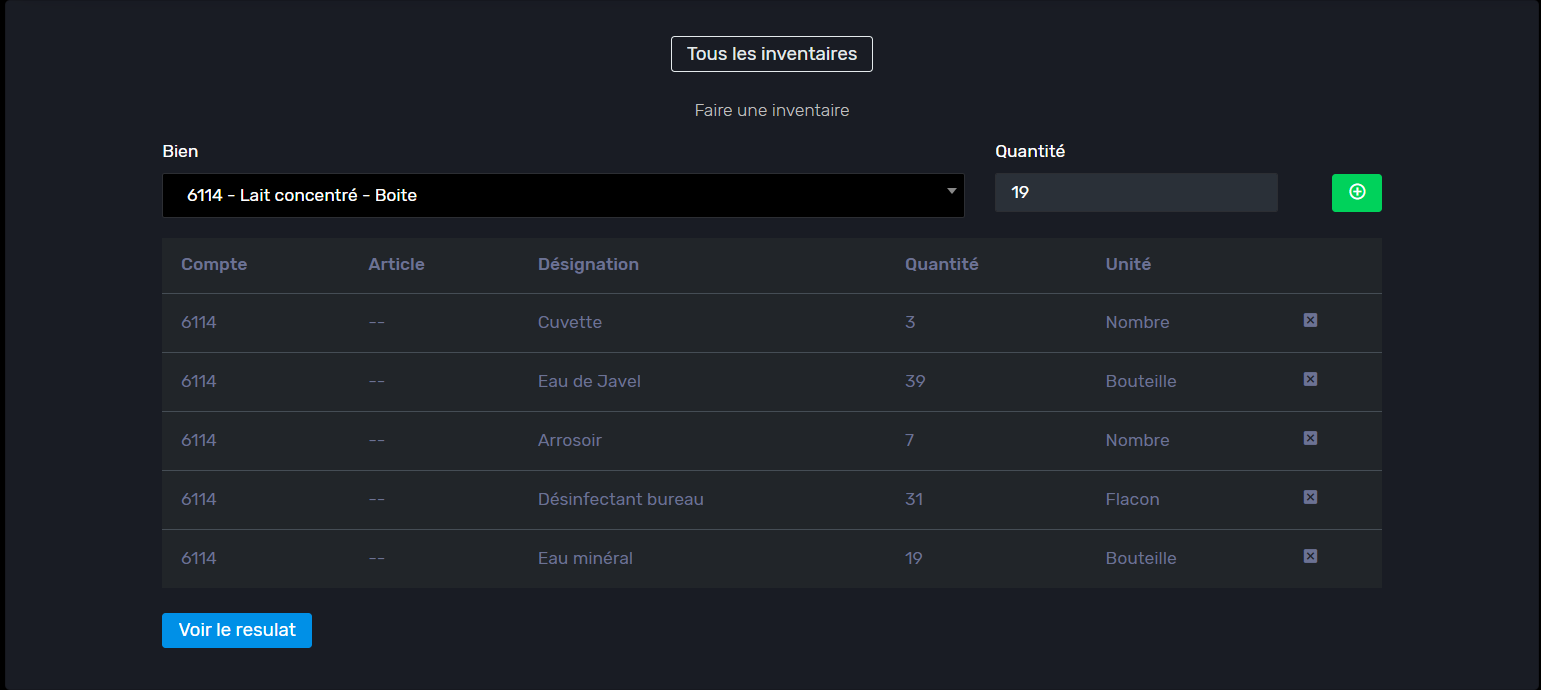


Figure 23 : Module d’inventaire des stocks



Figure 24 : Résultats de l'inventaire des stocks

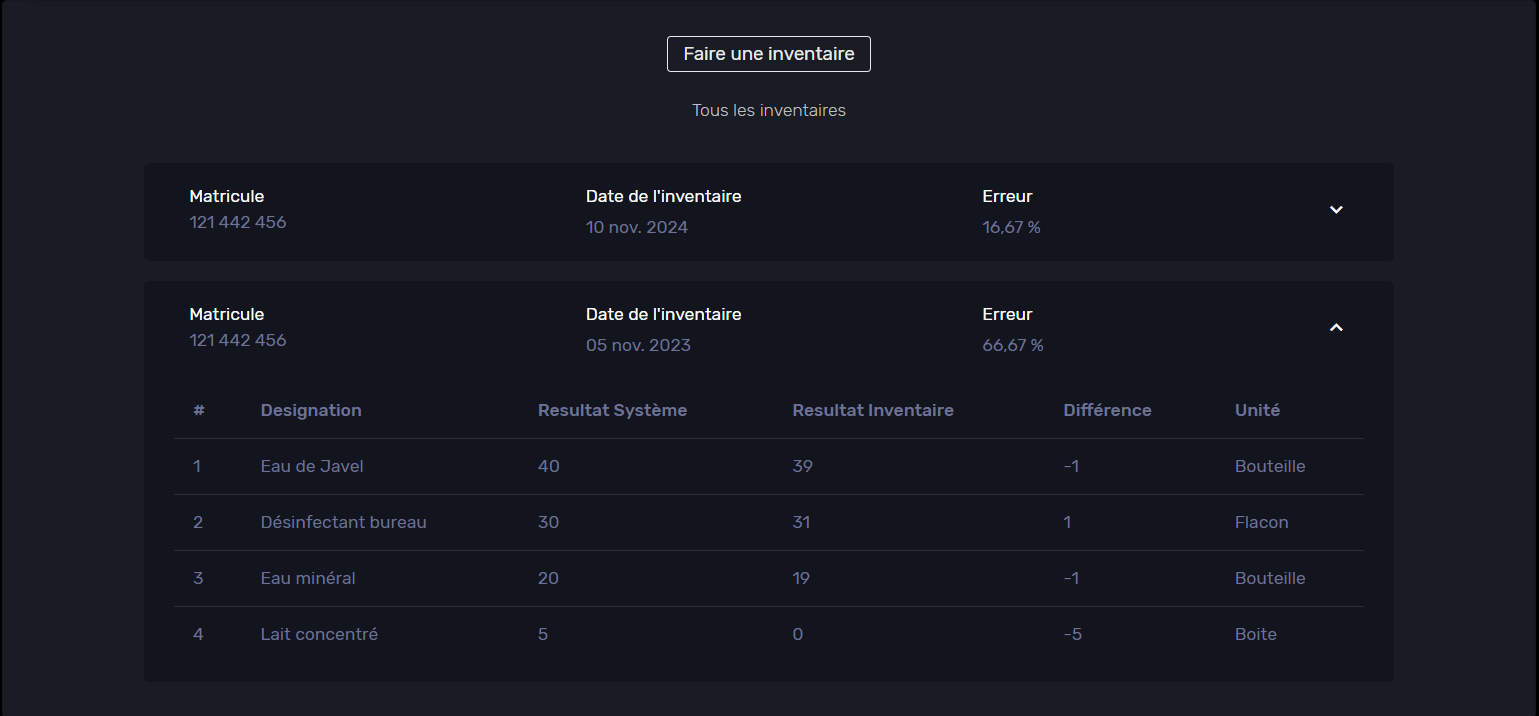


Figure 25 : Liste des inventaires effectués

### Gestions des immobilisations

#### Gestions des immobilisations

* **Description et objectif**

Cette fonctionnalité permet :

* L'ajout d'une ou plusieurs immobilisations à la fois
* La gestion d'une liste des immobilisations avec leurs propriétaires actuels
* L'exécution d'actions liées à l'état de l'immobilisation, comme : mettre en panne, réparer
* **Scenario**

Le système doit offrir une interface utilisateur intuitive pour :

* Ajouter facilement une nouvelle immobilisation ou plusieurs simultanément
* Visualiser et gérer l'inventaire des immobilisations existantes
* Afficher les propriétaires actuels pour chaque immobilisation
* Effectuer des actions spécifiques en fonction de l'état de l'immobilisation

Le système doit également permettre une gestion efficace des données, incluant :

* La création et la mise à jour automatique des états d'immobilisation
* L'historisation des actions effectuées sur chaque immobilisation
* Des rapports détaillés sur la liste des immobilisations et leurs propriétaires

Cette fonctionnalité devrait être conçue pour faciliter le suivi et la maintenance des immobilisations tout en offrant une visibilité claire aux responsables.

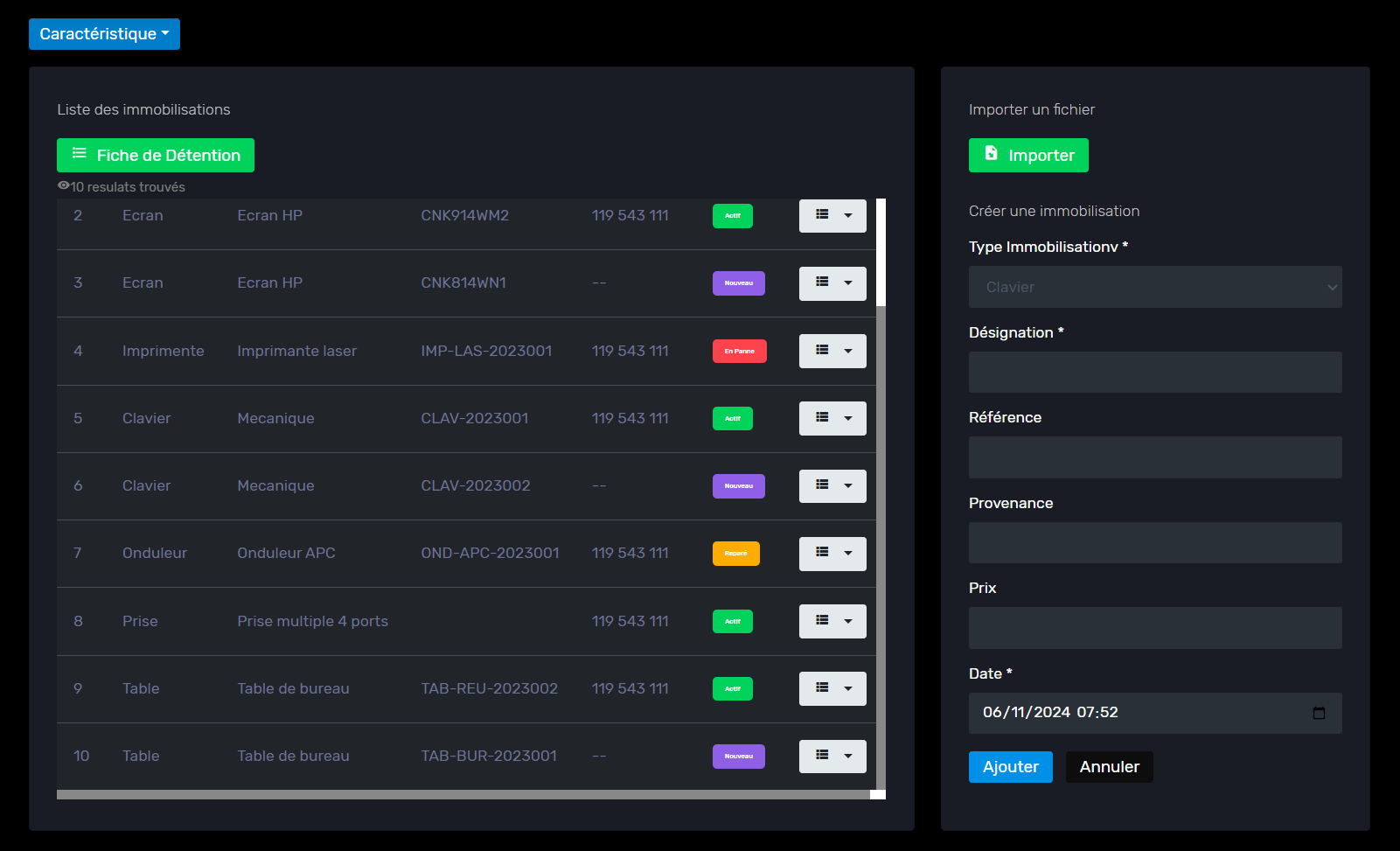


Figure 26 : Ajout et liste des immobilisations avec leurs états et propriétaires

#### Fiche de détention

* **Description et objectif**

Une page affiche les immobilisations et les quantités attribuées à un employé donné et permet de les modifier. Les administrateurs peuvent voir les fiches de détention de tous les employés, tandis que les utilisateurs ne peuvent voir que les leurs.

* **Scenario**
* Saisissez la matricule de l’employé pour accéder à sa fiche de détention
* Un bouton « Ajouter des immobilisations » permet d’attribuer d’autres immobilisations à l’employé, tandis qu’un bouton « Retirer » est disponible pour chaque immobilisation afin de la retirer, par exemple en cas de déplacement ou de réaffectation à un autre employé.
* En cliquant sur le bouton « Ajouter des immobilisations », l’utilisateur est redirigé vers une page d’attribution. Seules les immobilisations disponibles (celles qui ne sont pas déjà assignées) apparaissent dans la liste déroulante. L’utilisateur sélectionne les immobilisations à attribuer, saisit la quantité, et valide l’opération.

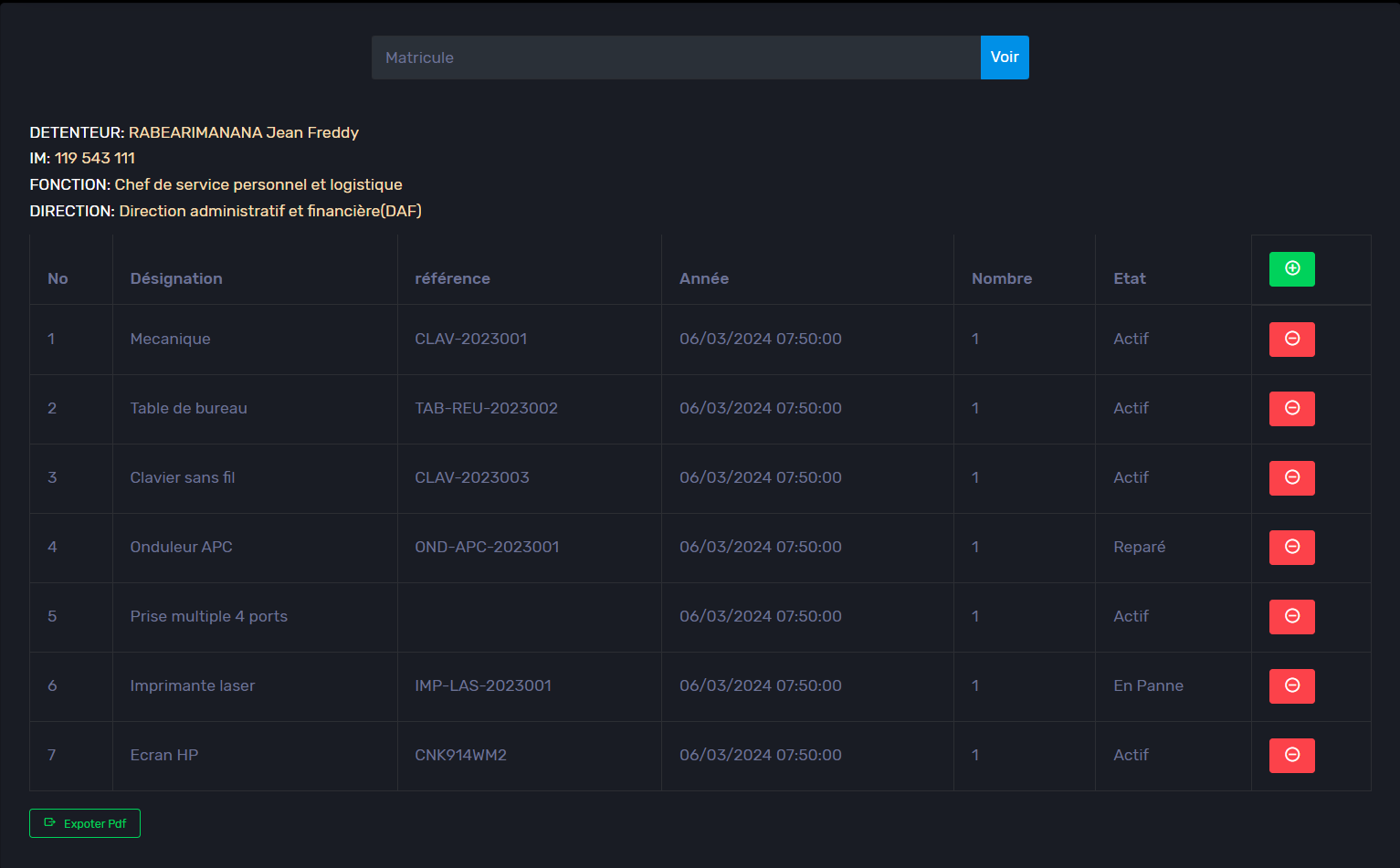


Figure 27 : Fiche de détention d’un employé



Figure 28 : Fiche de détention d’un employé format PDF

#### Historiques des immobilisations

* **Description et objectif**

Chaque immobilisation possède un bouton qui ouvre une page dédiée à son historique et à ses détails. Cette page permet de consulter, modifier les caractéristiques, gérer l'historique, ou d'annuler un ou plusieurs événements en cas d’erreur d’insertion. Cette fonctionnalité a été mise en place pour permettre une visualisation et un suivi précis de l’état des immobilisations.

* **Scenario**

En cliquant sur le bouton « Détails », l’utilisateur peut consulter les informations de l’immobilisation, son historique, et les événements qu’elle a subis, accompagnés des dates et des coûts. Pour modifier les caractéristiques de l’immobilisation, il suffit de cliquer sur le bouton « Modifier » et de mettre à jour les informations souhaitées : type, désignation, référence, date d’achat, provenance, et prix d’achat.

L’utilisateur peut également corriger un événement de l’historique (date ou coût) ou l’annuler (ex. en panne, réparé, attribué) en cas d’erreur lors de l’insertion.

Un bouton « Exporter PDF » permet d’exporter l’immobilisation avec son historique au format PDF.

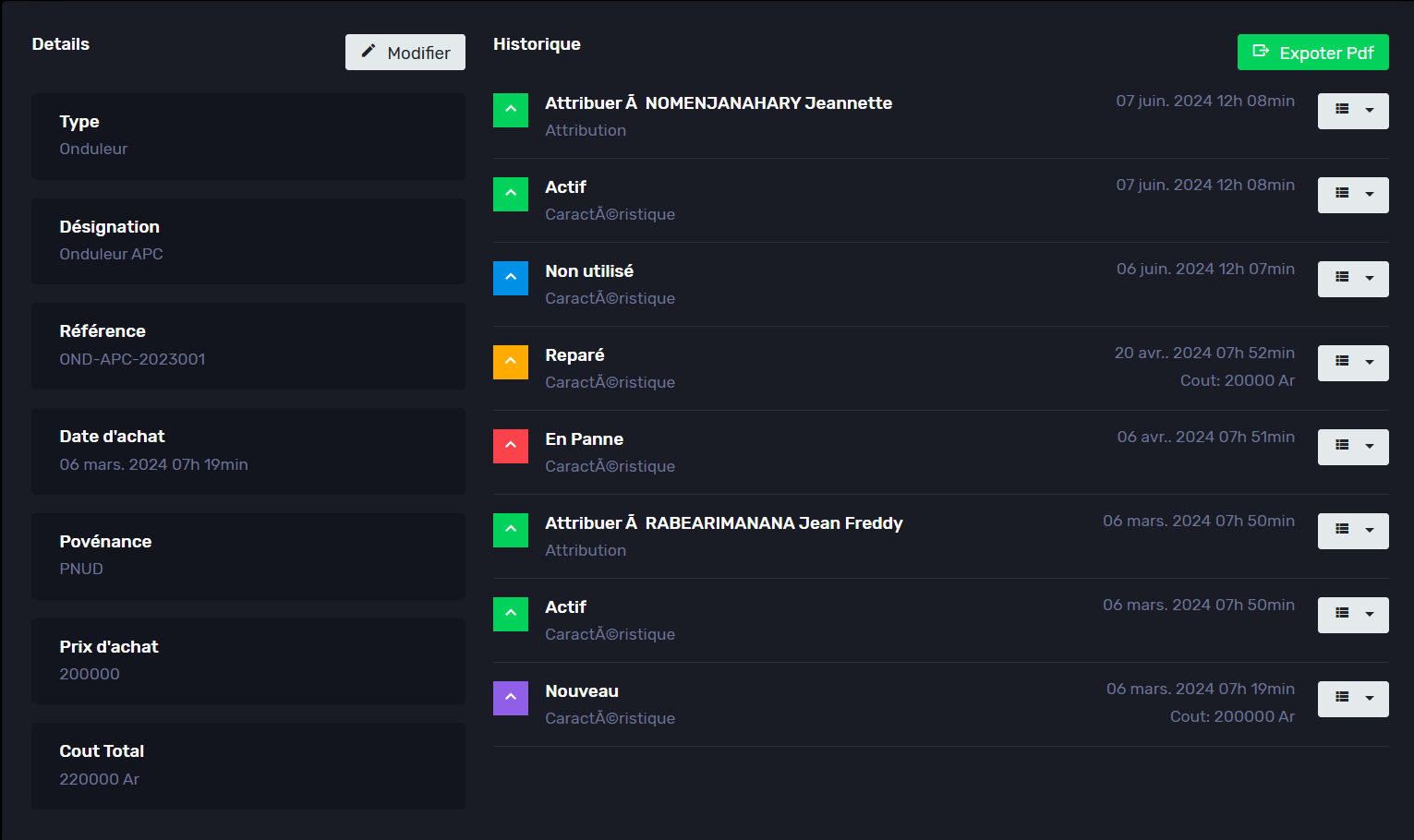


Figure 29 : Détails et historique des immobilisations

#### Maintenance périodique

* **Description et objectif**

Ce module de maintenance périodique permet de gérer les interventions d'entretien récurrentes pour les actifs de l'entité. Son but est de faciliter le suivi et l'organisation des maintenances à intervalles réguliers, garantissant ainsi la longévité et le bon fonctionnement des actifs.

* **Scenario**

Un administrateur accède au module de maintenance périodique et saisit les détails de la maintenance (titre, fréquence, date de début, tâches). Il enregistre ensuite les informations afin que la maintenance soit programmée automatiquement.

Un utilisateur consulte la liste des maintenances périodiques programmées pour les actifs. Il peut visualiser les dates prévues des interventions.

Un administrateur peut modifier les informations d'une maintenance périodique pour ajuster la fréquence ou la date de la prochaine intervention ou les taches.

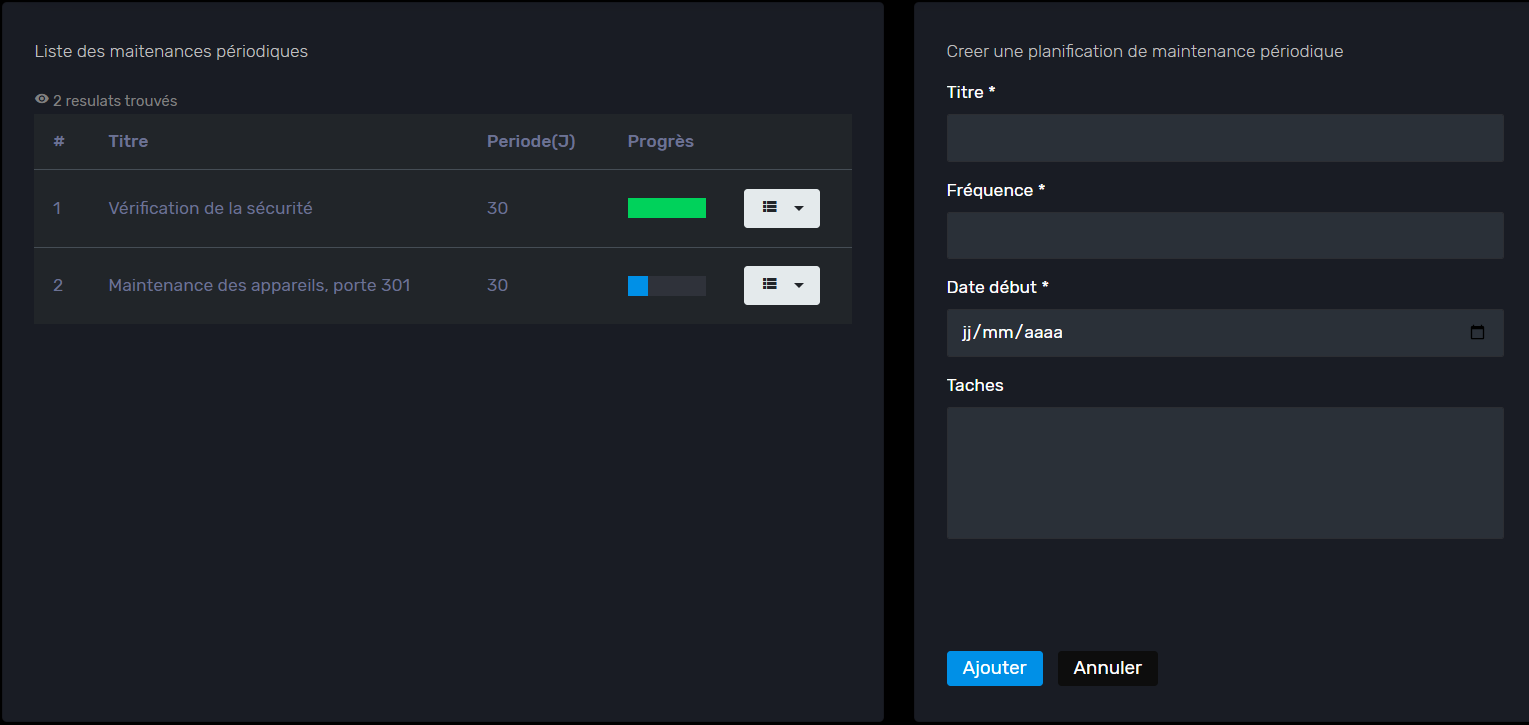


Figure 30 : Ajout et liste des maintenances périodiques

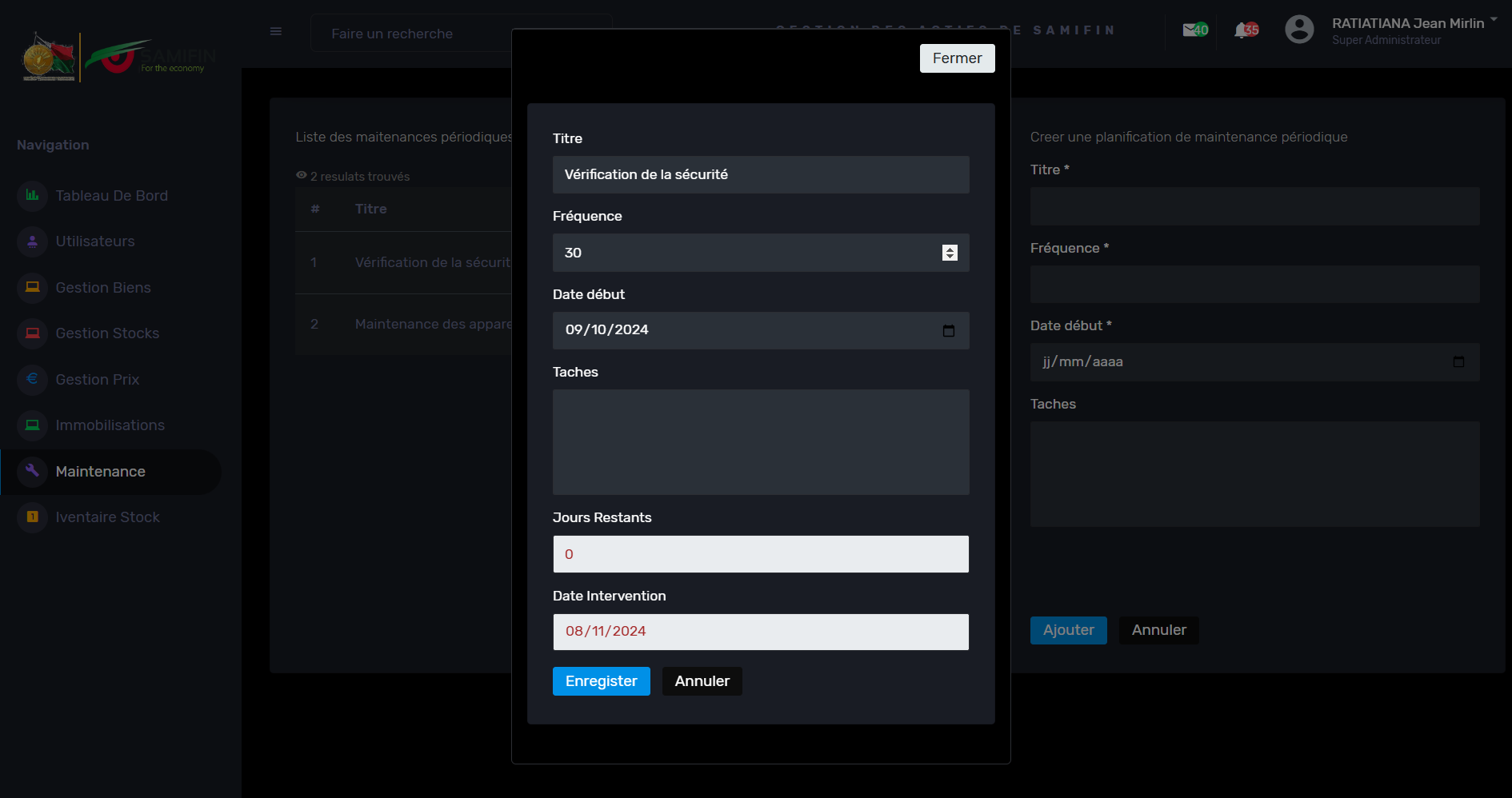


Figure 31 : Modal de modification d'une planification de maintenance

### Notification et recherche

#### Notification

* **Description et objectif**

Il s'agit d'un message d'alerte informant tous les utilisateurs de tout changement de données, qu'il s'agisse d'une entrée, d'une sortie de stock, d'une modification, d'une annulation de mouvement, d'une modification des immobilisations, ainsi que d'une notification en cas de maintenance.

* **Scenario**

En cliquant sur une notification dans la liste, tous ses détails s’affichent à droite et elle est automatiquement marquée comme lue. Pour supprimer des notifications, l’utilisateur peut sélectionner les messages qu'il souhaite supprimer ou choisir de sélectionner toutes les notifications en une seule fois, puis cliquer sur le bouton « Supprimer ». Cette suppression n’affecte que le compte de l’utilisateur.

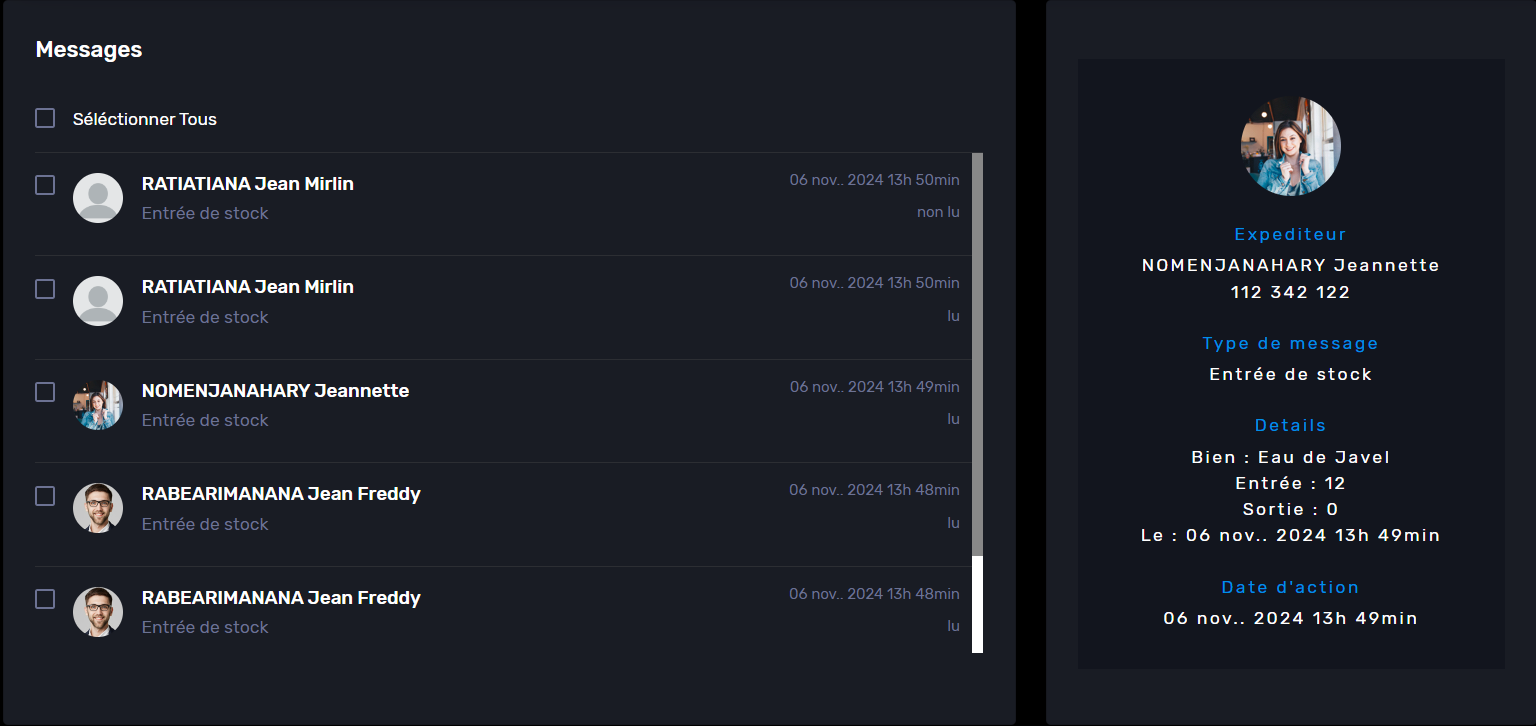


Figure 32 : Page de messages pour les notifications.

#### Recherche

Un module de recherche a été ajouté dans l'en-tête de la page pour permettre de rechercher sur n'importe quelle page contenant un tableau ou une liste. Cette recherche prend en compte toutes les colonnes de la liste et est exécutée via un processus jQuery pour assurer la rapidité des résultats.

Ici on prend ,par exemple, d’appliquer le fonctionnement de la recherche par la liste des utilisateurs.

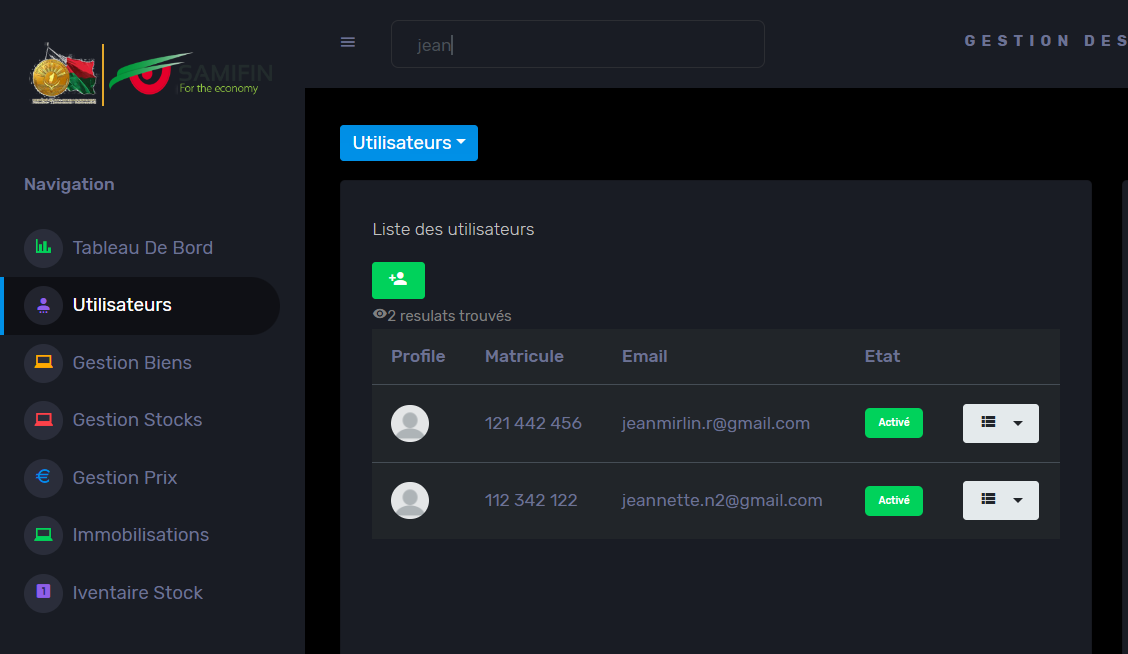


Figure 33 : Champ de recherche pour tous les tableaux de données

## État d’Analyse et Statistiques

### Répartition des utilisateurs par direction

Cette statistique montre la répartition des utilisateurs par direction. Elle permet de comprendre quelles directions suscitent le plus d'intérêt chez nos utilisateurs, ce qui peut guider les améliorations à apporter pour chaque direction. Et lorsqu’on sélectionne une direction , une page s’affiche avec les employés de cette direction.

### Statistiques des dépenses par mois

Ces données statistiques fournissent une vue d’ensemble des dépenses mensuelles liées aux stocks et aux immobilisations. Pour plus de flexibilité, un filtre permet de sélectionner les données par année. Et chaque point du graphique représente ses détails lorsqu’on clique dessus.

### Quantité de stock

Il s’agit d’une liste des cinq premiers biens ayant les quantités de stock les plus faibles, avec leur pourcentage par rapport à leurs dernières entrées.

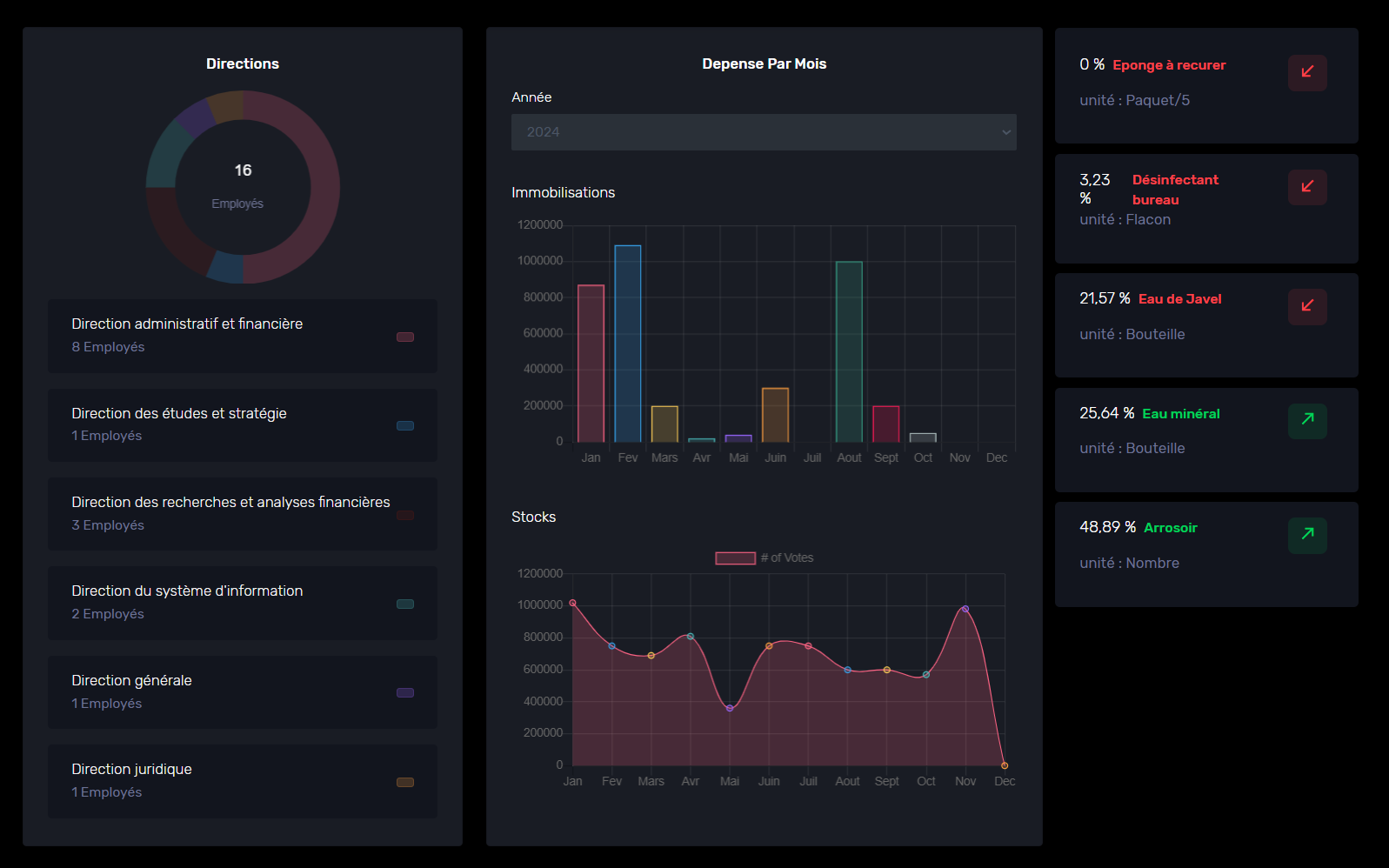


Figure 34 : Tableau de bord statistique

## Problèmes rencontrés et solutions

Au cours de mon stage, **deux principaux problèmes** ont été rencontrés dans le processus de mise en œuvre de l’application.

Le premier problème concernait **le manque de clarté sur les besoins spécifiques de l’entité**, en particulier concernant les données d’entrée et la nature précise de leurs défis opérationnels. Cette incertitude a parfois ralenti les progrès, car il était crucial de bien comprendre les attentes pour éviter de développer des fonctionnalités qui ne répondraient pas aux besoins réels. Pour surmonter cette difficulté, nous avons adopté une approche collaborative en effectuant des tests réguliers sur chaque petit module développé. Cela a permis de valider chaque étape avec les utilisateurs finaux, assurant que les fonctionnalités mises en place correspondaient aux exigences.

Le deuxième **défi consistait à concevoir une interface utilisateur intuitive**, simple et claire, afin que les utilisateurs puissent utiliser l’application sans difficulté et ainsi améliorer leur efficacité au travail. Pour y parvenir, j’ai opté pour une méthode itérative en présentant régulièrement l’interface et les fonctionnalités développées aux utilisateurs pour recueillir leurs retours. Ces retours étaient ensuite intégrés rapidement, permettant ainsi des ajustements en temps réel pour garantir que l’interface répondait aux attentes en termes d’ergonomie et de convivialité.

Ces solutions m’ont permis non seulement de mieux comprendre les besoins des utilisateurs, mais aussi de renforcer mes compétences en gestion de projet et en conception centrée sur l’utilisateur. Au final, cette approche a contribué à la création d’une application adaptée et fonctionnelle, répondant efficacement aux besoins de l’entité tout en optimisant l’expérience utilisateur.

# Évaluation du projet et connaissances acquises

## Bilan pour l'entreprise

L’ensemble des objectifs de « Samifin » a été pleinement atteint dans le cadre du projet de gestion des actifs. La mise en œuvre de cette solution a permis d’améliorer significativement la rapidité et l'efficacité des processus, réduisant ainsi le temps nécessaire pour traiter les informations et minimisant les erreurs humaines. Grâce à cette automatisation, les équipes ont pu se concentrer davantage sur des tâches à forte valeur ajoutée, tout en assurant une meilleure précision et une cohérence dans la gestion des actifs.

## Bilan personnel

Mon stage au sein de Samifin a marqué une étape clé dans mon parcours professionnel en représentant mes premiers pas dans le monde du travail. Il m'a permis de mettre en pratique les connaissances acquises lors de mes études à l'IT University et de me familiariser avec la réalité d'un environnement professionnel, comme le respect des horaires et la participation aux réunions. Ces expériences m'ont sensibilisé à l'importance des compétences interpersonnelles qui sont aussi essentielles que les compétences techniques dans un cadre professionnel.

Sur le plan technique, ce stage m'a permis de renforcer ma maîtrise de la technologie .NET à travers des projets concrets de développement back-end. J’ai approfondi des concepts comme l’architecture MVC et le design de bases de données, obtenant ainsi une vision complète du cycle de développement d'une application. En travaillant sur des projets orientés vers les besoins des utilisateurs finaux, j'ai également pris conscience de l'importance de l'expérience utilisateur et de la satisfaction client dans le développement de logiciels.

## Extension et évolution de l’application

Étant donné que le projet « Gestion des actifs » est en constante évolution, plusieurs pistes d’améliorations et d’extensions peuvent être envisagées pour répondre aux besoins futurs et optimiser l’efficacité de l’application. Parmi les extensions potentielles :

* Le développement de la version mobile de l’application.
* Utilisation de codes-barres pour l’entrée ou la sortie des stocks.
* L’ajout d’un système de notifications par SMS[[13]](#footnote-13).
* L’intégration d’un système de messagerie .
* Module d’optimisation des stocks en fonction du budget.
* Enregistrement des documents liés aux biens ou aux immobilisations.

# Conclusion

En conclusion, mon stage au sein de Samifin a été une expérience profondément enrichissante et formatrice, marquant le début de mon parcours dans le monde professionnel. Ce projet m’a permis d’appliquer et d’approfondir mes connaissances en développement informatique, notamment avec la technologie .NET, tout en relevant des défis concrets. J’ai eu l’opportunité de travailler sur un projet à valeur ajoutée pour l’entité, en contribuant activement au développement d’une application de « Gestion des actifs » qui facilitera le suivi et l’optimisation de leurs ressources.

Ce stage m’a non seulement permis de renforcer mes compétences techniques, mais également d’acquérir une réelle compréhension des attentes professionnelles : écoute des besoins, communication, rigueur et collaboration. En participant aux différentes phases du projet, de l’analyse des besoins à la conception des interfaces, j’ai appris à faire preuve de polyvalence et d’adaptabilité, deux qualités essentielles pour un développeur.

De plus, cette expérience m’a permis d’identifier plusieurs pistes d’améliorations et d’évolutions pour l’application, notamment le développement d'une version mobile, l’ajout de notifications par SMS pour améliorer la réactivité des utilisateurs, l’intégration d’une messagerie interne pour faciliter la communication, un module d’optimisation des stocks en fonction du budget, ainsi que l’enregistrement de documents liés aux biens pour une meilleure gestion documentaire. Ces extensions potentiellement à venir visent à enrichir l’application en la rendant plus flexible, accessible, et alignée sur les besoins évolutifs de l’organisation.

Enfin, cette expérience m’a sensibilisé à l’importance de l’orientation utilisateur dans le développement d’applications et à l’impact de la satisfaction client sur la réussite d’un projet. Ce stage a solidifié mon engagement à poursuivre dans cette voie et à perfectionner mes compétences en développement de logiciels, avec une attention particulière à l’expérience utilisateur et aux exigences métier. Je suis reconnaissant envers Samifin et mes encadrants pour leur encadrement et les responsabilités qui m’ont été confiées, qui ont toutes contribué à faire de ce stage une étape clé de ma formation et de ma carrière future.

# Bibliographie

Microsoft Learn.,Qu'est-ce que ASP.NET Core MVC? Pourquoi l'utiliser ? Disponible sur

<https://learn.microsoft.com/fr-fr/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-8.0> (Consulté le 10-08-2024).

stack overflow., Qu'est-ce que le modèle MVC et pourquoi l'utiliser ? Disponible sur :

<https://stackoverflow.com/questions/tagged/asp.net-core-mvc> (Consulté le 09-08-2024).

Axopen.,Net Core vs Spring Boot. Disponible sur :

<https://www.axopen.com/technos/versus/.net%20core-vs-spring%20boot/> (Consulté le 09-08-2024).

Microsoft Learn, Comment fonctionne la gestion des routes dans ASP.NET Core MVC ? Disponible sur :

<https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-8.0> (Consulté le 09-08-2024).

FullStack café., Quelles sont les principales différences entre ActionResult et ViewResult en ASP.NET Core MVC ? Disponible sur :

<https://www.fullstack.cafe/blog/dot-net-core-interview-questions-and-answers> (Consulté le 09-08-2024).

Microsoft Learn .,Comment fonctionne Entity Framework ? Disponible sur :

<https://learn.microsoft.com/fr-fr/dotnet/framework/data/adonet/ef/overview>

nuget.,Create PDFs and images with .NET. Disponible sur :

<https://www.nuget.org/packages/rotativa.aspnetcore> (Consulté le 12-08-2024).

Digital Ocean .,Article de Comparaison entre MySQL et PostgreSQL . Disponible sur :

<https://www.digitalocean.com/community/tutorials/mysql-vs-postgresql> (Consulté le 12-08-2024).

Aws., Quelle est la différence entre MySQL et PostgreSQL ? Disponible sur :

<https://aws.amazon.com/fr/compare/the-difference-between-mysql-vs-postgresql/> Consulté le 12-08-2024)

PostgreSQL Official Documentation, Disponible sur :

<https://www.postgresql.org/docs/> (Consulté le 12-08-2024)

MySQL Official Documentation. Disponible sur :<https://dev.mysql.com/doc/> (Consulté le 12-08-2024).

# Annexe

- Configuration d'affichage et de filtre pour les stocks dans le fichier Configuration.xml

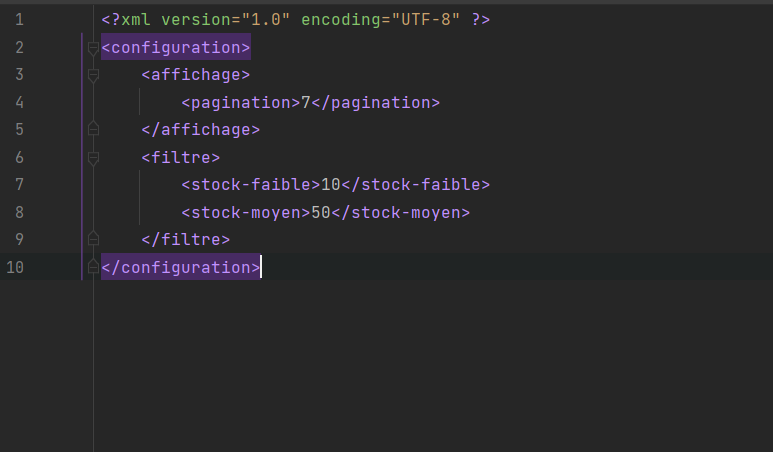


Figure 35 : Configuration d'affichage et de filtre pour les stocks

-Configuration de l'application : Accès à la base de données et Paramètres SMTP



Figure 36 : Accès à la base de données et Paramètres SMTP

1. TIC : Technologies de l’Information et de la Communication [↑](#footnote-ref-1)
2. LBC/FT : Lutte contre le Blanchiment de Capitaux et le Financement du Terrorisme [↑](#footnote-ref-2)
3. HTTP : Hypertext Transfer Protocol [↑](#footnote-ref-3)
4. URL : Uniform Resource Locator [↑](#footnote-ref-4)
5. JSON : JavaScript Object Notation [↑](#footnote-ref-5)
6. MVC : Model-View-Controller [↑](#footnote-ref-6)
7. ORM : Object-Relational Mapping [↑](#footnote-ref-7)
8. ADO : ActiveX Data Objects [↑](#footnote-ref-8)
9. SQL : Structured Query Language [↑](#footnote-ref-9)
10. PBKDF2 : Password-Based Key Derivation Function 2 [↑](#footnote-ref-10)
11. XML : Extensible Markup Language [↑](#footnote-ref-11)
12. PDF : Portable Document Format [↑](#footnote-ref-12)
13. SMS : Short Message Service [↑](#footnote-ref-13)