

## MEMOIRE DE FIN D'ETUDES

### Développement d'applications Web

**PÉRIODE :**  
Du 07/05/2025 au 05/07/2025.

**Nom & Prénom de l'Etudiant** : M. ...  
**Classe, Filière et Option de l'Etudiant** : 3<sup>ème</sup> Licence Informatique  
Option Développement Informatique.

**ENCADRANT EN ENTREPRISE :**

**Nom et Prénom** : Mr. ;;;;

**Fonction** : ;;;;

**Visa** :



**ENCADRANT ECOLE :**

**Nom et Prénom** : ;;;.

**Visa** :



**Année Universitaire** : 2024/2025



Complete the installation of Docker Desktop

Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

Use advanced settings  
You manually set your preferred configurations.

Finish

## Dédicace

*Je dédie ce travail à :*

*À ma mère, pour son amour inconditionnel, son soutien sans faille et sa patience infinie. Tu es pour moi un modèle de courage et de dévouement. Que ce modeste travail soit le reflet de ma profonde reconnaissance et qu'il t'apporte joie et fierté.*

*À mon père, dont les sacrifices, la persévérance et les efforts constants pour mon éducation et mon bien-être sont inestimables. Ce projet est le fruit de ta générosité et de ta confiance en moi.*

*À mes frères et ma sœur, pour votre présence rassurante, votre soutien et tous ces moments de complicité qui ont illuminé ma vie. Vous êtes une source inépuisable de motivation.*

*À mes chers amis, pour les instants inoubliables, vos encouragements et votre aide précieuse tout au long de ce parcours.*

*À tous mes professeurs et formateurs, pour la qualité de leur enseignement, leurs conseils avisés et leur disponibilité qui m'ont permis de progresser et de mener à bien ce projet.*

*Enfin, toutes les personnes qui, de près ou de loin, ont contribué à ma formation, m'ont soutenu ou encouragé. Trouvez ici l'expression de ma plus profonde gratitude.*

**UN GRAND MERCI À VOUS TOUS.**

**ADNANE ZINEEDDINE**

## Remerciements

Au nom d'Allah, le Tout Miséricordieux, le Très Miséricordieux. Ce travail n'aurait pu être réalisé sans la grâce et la bonté de Dieu. Je tiens ici à remercier tous ceux qui m'ont aidé à mener à bien ce travail. Je remercie d'abord mon encadrant, **Mr ;;;;** d'avoir donné son accord de superviser ce travail, de m'avoir régulièrement guidé et assisté.

Mes remerciements vont également à mon encadrant de stage au sein de ;;;, pour sa disponibilité, ses précieux conseils et son soutien continu qui ont grandement contribué à la réussite de ce projet. Veuillez recevoir, Monsieur, l'expression de mon profond respect et de ma sincère reconnaissance.

J'exprime aussi ma gratitude au jury pour l'honneur qu'il m'accorde en évaluant ce travail.

Enfin, mes remerciements s'adressent à toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce projet, et qui m'ont apporté aide, encouragement et soutien.

## Résumé

La présente mémoire synthétise le travail que j'ai effectué au sein de ;;;;;, dans le cadre de mon projet de fin d'études pour l'obtention d'une **Licence en Développement Informatique** à l'École Supérieure d'Ingénierie et de Management – **SUP'EMIR**.

Le projet intitulé « **Développement d'une application de gestion d'inventaire** » vise à concevoir une plateforme web pour gérer efficacement les matériels et équipements de l'entreprise. Réalisée avec le framework **Laravel**, l'application intègre une API REST et des interfaces conviviales grâce à **Blade**, **Tailwind CSS** et **JavaScript**. Après une étude préliminaire définissant les besoins et le périmètre, un cahier des charges a été établi pour planifier le développement. Ce projet a permis de mettre en pratique les acquis de formation et de se confronter aux enjeux réels de la gestion d'inventaire.

# Glossaire

**API (Application Programming Interface)** : Interface de programmation permettant à différentes applications de communiquer entre elles, utilisée ici pour exposer les fonctionnalités de l'application sous forme de services web REST.

**Laravel** : Framework PHP open source basé sur le modèle MVC, utilisé pour développer l'application web de gestion d'inventaire.

**MVC (Modèle-Vue-Contrôleur)** : Architecture logicielle permettant de séparer la logique métier (Modèle), l'interface utilisateur (Vue) et le traitement des actions de l'utilisateur (Contrôleur).

**MySQL** : Système de gestion de base de données relationnelle utilisé pour stocker les informations sur les utilisateurs, matériels, affectations, etc.

**Sanctum** : Package Laravel permettant de gérer l'authentification par tokens API, utilisé pour sécuriser les accès aux endpoints de l'application.

**CRUD (Create, Read, Update, Delete)** : Opérations de base de manipulation des données, essentielles dans la gestion des ressources (utilisateurs, matériels, catégories...).

**Traçabilité** : Capacité de l'application à enregistrer l'historique des mouvements de matériels, en précisant les dates, utilisateurs et motifs des affectations et retours.

**Dashboard (Tableau de bord)** : Interface regroupant des graphiques et indicateurs clés pour visualiser l'état du stock, les affectations en cours et l'utilisation des matériels.

# Table des figures

Figure 1.1: BMCE Bank -----	16
Figure 1.2 : Organigramme BMCE Bank -----	18
Figure 3 : Architecture du projet-----	23
Figure 12 : Diagramme de cas d'utilisation général-----	35
Figure 13 : Diagramme de cas d'utilisation de Catégories-----	36
Figure 14 : Diagramme de cas d'utilisation de Matériels-----	37
Figure 15 : Diagramme de cas d'utilisation d'utilisateurs-----	37
Figure 15 : Diagramme de cas d'utilisation d'Affectations-----	37
Figure 22 : Diagramme de classe-----	43
Figure 23 : php -----	45
Figure 24 : laravel -----	45
Figure 25 : mysql -----	46
Figure 26 : Laravel+Sanetum-----	46
Figure 27 : JavaScript-----	47
Figure 28 : Tailwind CSS-----	47
Figure 29 : GitHub-----	48
Figure 30 : Apache http Server-----	48
Figure 31 : .Postman -----	49
Figure 32 : UML -----	50
Figure 33 : Visual Studio Code-----	50
Figure 41 : Page d'authentification-----	56
Figure 42 : Page Materiels -----	57
Figure 43 : Page d'information des affectations-----	57
Figure 44 : Page liste des utilisateurs -----	58
Figure 45 : Page Informations Categories-----	58
Figure 46 : Page liste des KPIs -----	59
Figure 47 : Répartitions Des Matériels-----	59
Figure 48 : Affectations Par Mois -----	60
Figure 49 : Status des Matériels -----	60
Figure 50 : Utilisateurs Par Rôle -----	61
Figure 51 : Durée Moyenne Par aFFECTATIONS-----	62
Figure 52 :Liste 10 Dernières-Affectations-----	63
Figure 54 : Page Gestion des administrations-----	64
Figure 55 : Page Profile -----	64



9



Complete the installation of Docker Desktop  
 Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.  
 Use advanced settings  
You manually set your preferred configurations.

Next

## Liste des tableaux

Tableau 1 : Liste des tâches

28

# Table des matières

<b>CHAPITRE I : PRÉSENTATION D'ORGANISME</b>	<b>14</b>
I. Organisme d'accueil	15
II. Domaine d'activité	16
III. Ses clients	17
IV. Organigramme d'entreprise	18
<b>CHAPITRE 2 : CONTEXTE GENERAL DU PROJET</b>	<b>19</b>
I. Introduction	20
II. Problématique	20
III. Solution	20
IV. Cahier de charge	21
V. Architecture du projet	23
<b>CHAPITRE 3 : GESTION DU PROJET</b>	<b>25</b>
I. Étapes d'élaboration du projet	26
II. Tâches effectuées	26
<b>CHAPITRE 4 : ETUDE CONCEPTUELLE</b>	<b>33</b>
I. Introduction	34
II. Définition UML	34
III. Diagramme de cas d'utilisation	34
IV. Diagramme de classe	43
<b>CHAPITRE 5 : OUTILS DE DÉVELOPPEMENT</b>	<b>44</b>
I. Introduction	45
II. Outils de développements	45
III. Outils de modélisations	52
IV. Editeurs de code	53
<b>CHAPITRE 6 : RÉALISATION</b>	<b>55</b>

# Introduction Générale

La maîtrise des actifs matériels et la rigueur des processus sont des composantes indispensables dans un domaine professionnel à forte responsabilité, où la gestion d'inventaire au sein des grandes institutions financières est un enjeu fondamental. Dans le cadre de l'organe d'audit à Bank of Africa Groupe, un outil fiable, sécurisé et convivial, nécessaire pour une gestion sécurisée et efficace des matériels, demeure un dispositif principal.

L'architecture oriente actuellement les solutions de développement de logiciels irisées vers une conception centrée sur l'audit, fournissant à ces solutions une infrastructure professionnelle. Ce document de projet élabore la conception et le développement d'une application web pour la gestion d'un inventaire des stocks de l'organe d'audit. L'application est conçue pour gérer, allouer aux utilisateurs, tracer les mouvements, et s'utiliser pour générer des rapports et des tableaux de bord précis à tout moment.

Développée en utilisant le framework Laravel avec une base de données MySQL, l'application intègre une architecture respectant le modèle MVC et des fonctionnalités avancées de reporting, tout en garantissant la sécurité des accès grâce à l'authentification via Laravel Sanctum.

Ce rapport présente l'ensemble des étapes de réalisation du projet, depuis l'analyse des besoins jusqu'à la conception, l'implémentation et l'évaluation de l'application. Il est structuré comme suit :

- **Chapitre 1 :** Présentation du contexte, de l'entreprise d'accueil (Bank of Africa Groupe) et des enjeux du projet.
- **Chapitre 2 :** Analyse des besoins, définition du cahier des charges et étude fonctionnelle du système.
- **Chapitre 3 :** Présentation des outils et technologies utilisés pour la conception et le développement de l'application.
- **Chapitre 4 :** Modélisation du système à l'aide de la méthode UML, incluant les diagrammes de cas d'utilisation, de classes et de séquence.
- **Chapitre 5 :** Conception et implémentation de l'application : architecture technique, création des tables, développement des modules principaux et mise en place de l'interface utilisateur.

- **Chapitre 6 :** Présentation des interfaces graphiques réalisées, bilan fonctionnel et perspectives d'amélioration.

À travers ce travail, l'objectif est de mettre en place un outil performant et adapté aux exigences strictes de l'audit interne, tout en développant des compétences techniques et organisationnelles indispensables dans le cadre professionnel.

# Chapitre I : Présentation d'organisme.

---

## I. Présentation de Bank of Africa Groupe

Bank of Africa Groupe est l'un des acteurs majeurs du secteur bancaire et financier en Afrique, avec une présence dans plus de 30 pays. Le groupe occupe une place de premier plan dans le financement de l'économie, le développement des services bancaires et l'accompagnement des entreprises et des particuliers.

Avec une forte implantation au Maroc et en Afrique subsaharienne, Bank of Africa Groupe se distingue par la diversité de ses activités et son engagement en faveur de la modernisation des systèmes d'information pour répondre aux enjeux de performance, de sécurité et de digitalisation.

Le groupe s'appuie sur une organisation solide et des équipes d'experts pour accompagner ses clients dans leurs besoins financiers et technologiques, notamment au sein de l'organe d'audit interne, garant de la conformité, du contrôle et de la traçabilité des opérations.

Bank of Africa, autrefois BMCE Bank ou Banque Marocaine du Commerce Extérieur, est une banque commerciale marocaine. Fondée en 1959 en tant que banque publique, elle a été privatisée en 1995 et rachetée par le financier Othman Benjelloun via la compagnie d'assurance RMA qu'il contrôle.

Aujourd'hui, BANK OF AFRICA est le groupe bancaire marocain ayant la plus grande présence internationale, couvrant 32 pays en Afrique, en Europe, en Asie et en Amérique du Nord.

Le groupe emploie plus de 14 900 personnes à travers le monde, avec plus de 2 000 points de vente desservant près de 6,6 millions de clients.

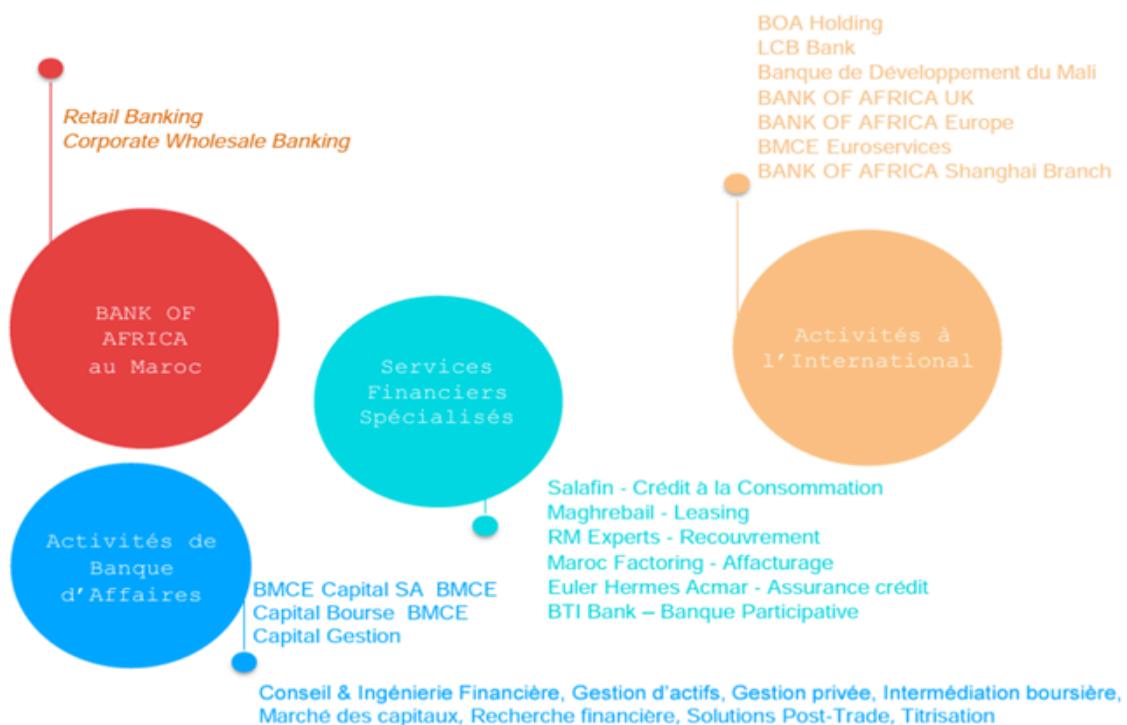


Figure 1.1: Activities of Bank of Africa <sup>1</sup>

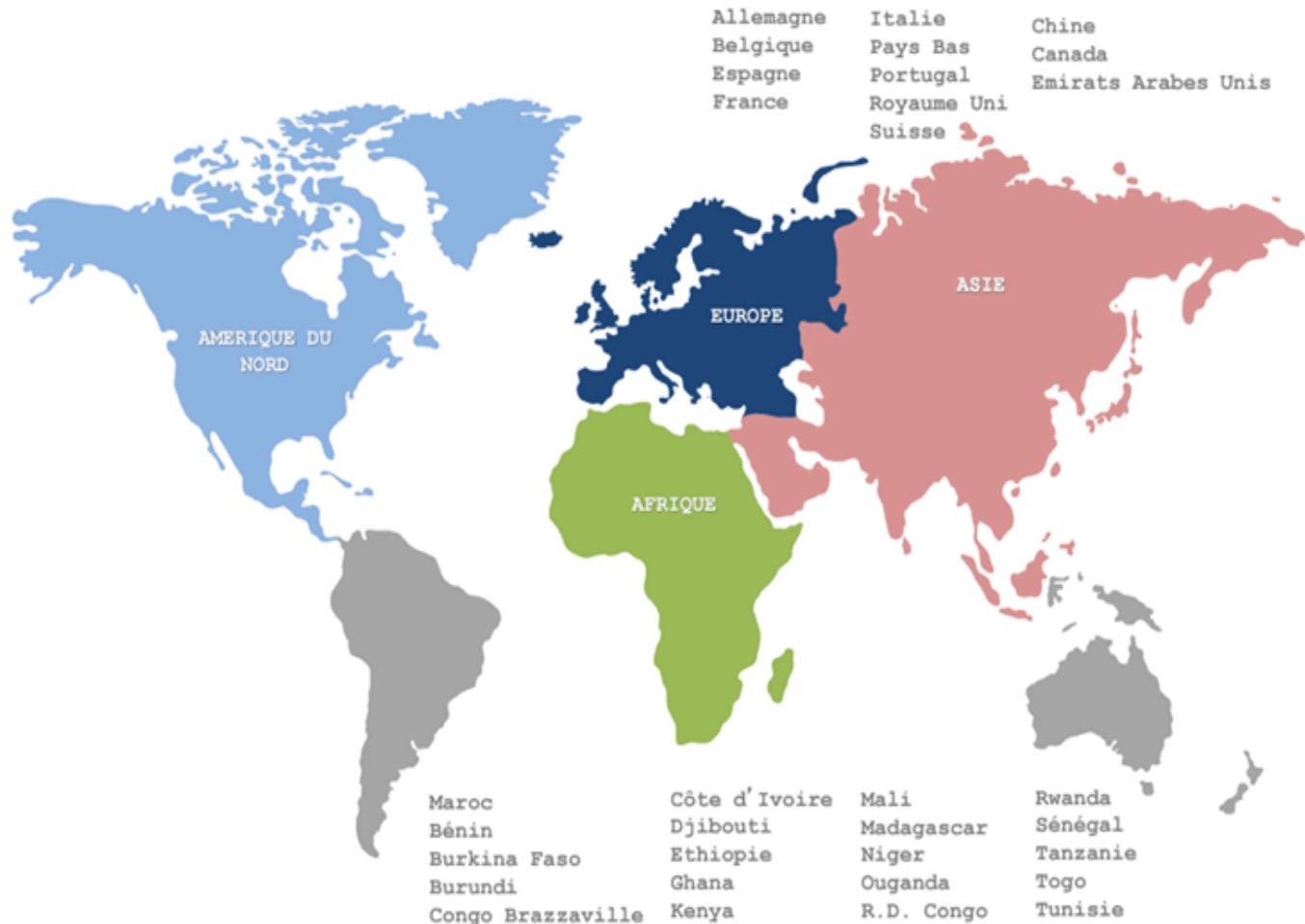


Figure 1.2: Bank of Africa worldwide <sup>2</sup>

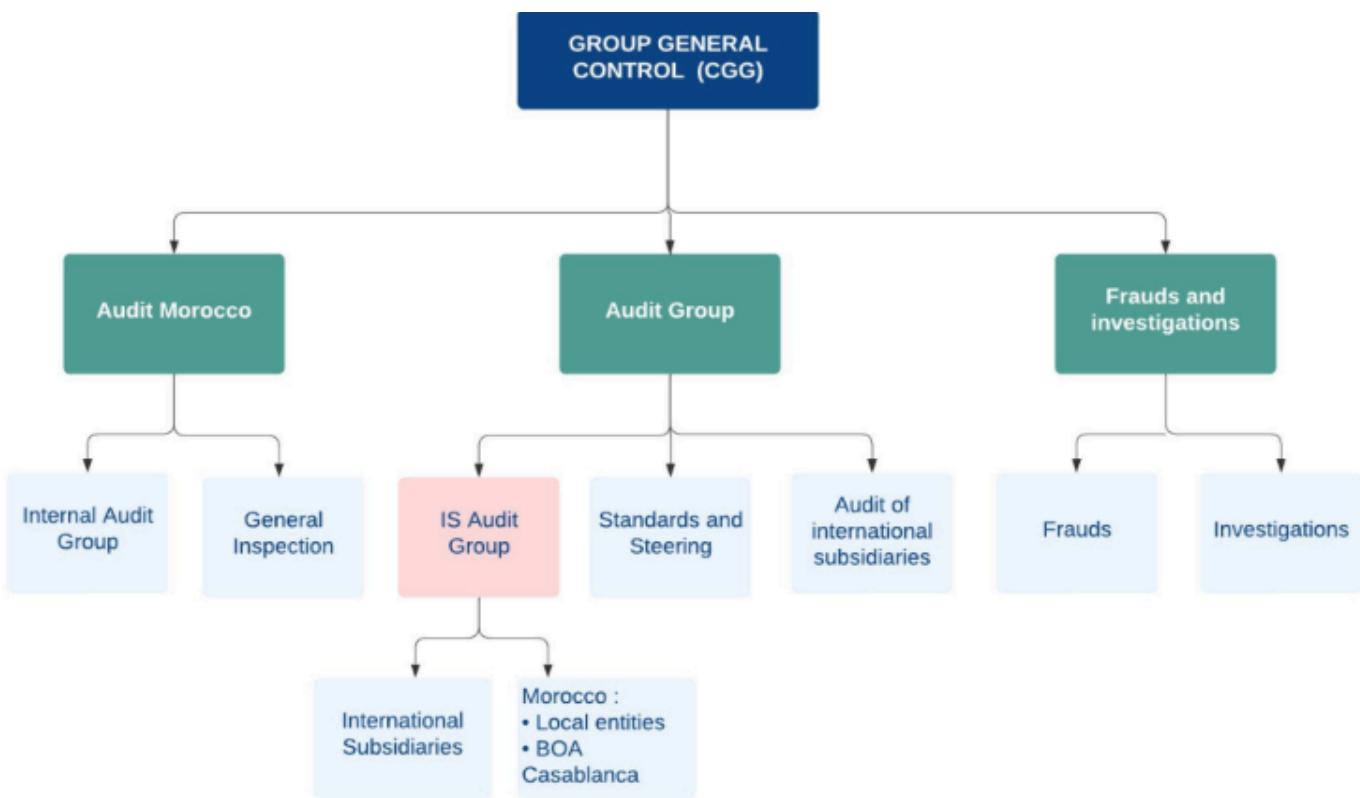


Figure 1.3: Group General Control Department chart

## I. Organigramme d'entreprise

Le Bank of Africa compte plusieurs départements essentiels à son activité. Ainsi, le département pour lequel nous avons fait notre stage est le département d'audit des systèmes d'information, qui est nommé « Informatique du Contrôle Général du Groupe » sur l'organigramme de l'entreprise. Ce département a pour mission le bon fonctionnement, la sécurité et le respect des normes réglementaires liés aux systèmes informatiques de la banque et de ses succursale

Il s'agit d'un département clé qui contribue à identifier et à atténuer les risques liés à la sécurité de l'information, aux cyberattaques, à la fraude informatique et à d'autres problèmes liés aux technologies de l'information.

Les principales responsabilités du département d'audit informatique de Bank of Africa sont les suivantes :

- **Évaluation des contrôles internes** : L'équipe d'audit évalue les contrôles internes relatifs aux systèmes et processus informatiques de la banque. Cela inclut l'analyse des politiques et procédures, l'examen des accès aux systèmes, l'identification des vulnérabilités et la recommandation d'actions correctives.

- **Audit de la sécurité des systèmes** : L'équipe réalise des audits de sécurité pour identifier les vulnérabilités potentielles dans les systèmes informatiques de la banque. Cela comprend la vérification des pare-feu, des systèmes de détection d'intrusion, des politiques de gestion des mots de passe, des protocoles de chiffrement et d'autres mesures de sécurité.
- **Audit de conformité** : Le département s'assure que la banque respecte les réglementations en matière de protection des données, de confidentialité des clients et d'autres lois en vigueur. Il vérifie également si les politiques et procédures internes de la banque sont conformes aux normes et bonnes pratiques en matière de sécurité informatique.
- **Évaluation des risques** : L'équipe évalue les risques associés aux systèmes informatiques et propose des recommandations pour les atténuer. Elle identifie les menaces potentielles, analyse leur impact sur les opérations de la banque et recommande des mesures préventives et de protection appropriées.
- **Tests de continuité d'activité** : Le département d'audit informatique participe également aux tests de continuité d'activité afin de s'assurer que la banque dispose de plans adéquats pour faire face aux interruptions de services informatiques telles que les pannes systèmes, les catastrophes naturelles ou les cyberattaques.

## Chapitre 2 : Contexte général du projet.

---

Ce chapitre a pour objectif de situer le projet dans son environnement organisationnel et contextuel. Il présente le cadre du projet, la problématique à résoudre et la solution proposée. Enfin, il détaille le cahier des charges défini pour la réalisation de l'application de gestion d'inventaire.

## I. Introduction

Dans un environnement où la rigueur, la traçabilité et la gestion optimale des ressources matérielles sont des impératifs, l'organe d'audit de Bank of Africa Groupe a ressenti la nécessité de disposer d'une application centralisée et sécurisée pour sa gestion de l'inventaire. Les obstacles induits par la méthode d'un procédé manuel consistaient au manque de visibilité sur l'état des stocks, l'impossibilité de suivre les mouvements de matériels et la détérioration de temps de contrôle et d'audit.

La mise en place d'une solution numérique moderne répond à la nécessité d'optimiser le suivi des matériels, de sécuriser les affectations aux utilisateurs et de générer des rapports fiables et exploitables par les responsables de l'audit.

## II. Problématique

Les problématiques identifiées au début du projet étaient les suivantes :

- Manque de visibilité sur l'inventaire du matériel de l'organe d'audit,
- Difficulté à tracer les mouvements de chaque équipement (affectation, retour...),
- Risques d'erreurs ou de pertes d'informations avec les enregistrements manuels,
- Impossibilité de générer des rapports dynamiques et des tableaux de bord précis,

Ces difficultés nuisent à l'efficacité des processus, augmentent les délais lors des contrôles internes et freinent la capacité de l'organe d'audit à garantir une gestion rigoureuse et sécurisée du parc matériel.

## III. Solution

Pour répondre aux problématiques identifiées, la solution envisagée consistait à développer une **application web centralisée de gestion d'inventaire**, accessible aux membres de l'organe d'audit et au personnel autorisé. Cette application offre les fonctionnalités suivantes :

## IV. **Enregistrement, mise à jour et consultation du stock de matériels** : Permettre une gestion précise et en temps réel du parc matériel.



Complete the installation of Docker Desktop  
 Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.  
 Use advanced settings  
You manually set your preferred configurations.

Next

**V. Affectation et suivi des équipements auprès des utilisateurs :** Gérer la distribution et le retour des matériels, avec un suivi détaillé.

**VI. Traçabilité complète des mouvements :** Chaque action liée au matériel est enregistrée avec un historique détaillé, assurant la transparence.

**VII. Génération de rapports et tableaux de bord :** Faciliter les audits grâce à des rapports clairs et des indicateurs de performance visuels.

**VIII. Interface intuitive :** Garantir une prise en main rapide pour les utilisateurs, même sans compétence technique poussée.

**IX. Contrôle d'accès sécurisé basé sur les rôles :** Assurer que chaque utilisateur ait des droits adaptés à ses responsabilités.

## X. Cahier de charge

À l'issue de l'analyse des besoins et des échanges avec les responsables de l'organe d'audit, le cahier des charges a été défini comme suit :

### Fonctionnalités principales :

- **Gestion des utilisateurs** : création, modification, suppression, et gestion des rôles (auditeur, administrateur...).
- **Gestion des comptes administrateurs** : création de comptes admin, modification des informations, suppression et attribution de priviléges spécifiques (droits de modification, gestion des utilisateurs, accès aux rapports, etc.).
- **Gestion des matériels** : enregistrement des équipements (catégorie, référence, état, série,...).
- **Gestion des affectations** : suivi des mouvements de matériel (affectation, retour,, avec enregistrement des dates, utilisateurs concernés et motifs).
- **Tableaux de bord et rapports** : affichage des statistiques clés telles que matériels en stock, taux d'utilisation, historiques par utilisateur.
- **Traçabilité** : conservation de l'historique complet des mouvements de chaque équipement.
- **Sécurité** : système d'authentification et de gestion des permissions basé sur les rôles, avec contrôle strict des accès.

### Règles de gestion principales :

## XI. Enregistrement du matériel :

- Chaque matériel possède un identifiant unique.
- Les informations obligatoires sont : catégorie, modèle, Hostname, serie , état initial.

## XII. Affectation :

- L'affectation précise l'utilisateur, la date de début et la date prévisionnelle de fin ainsi que le motif lors de désaffectation.
- Un matériel affecté ne peut être affecté à deux utilisateurs en même temps.

## XIII. Retour :

- Tout retour est consigné avec date, motif et état du matériel.
- Les opérations de retour sont enregistrées et mises à jour dans l'historique.

## XIV. Utilisateurs et rôles :

- Les permissions dépendent du rôle attribué.
- Les administrateurs ont un accès complet à toutes les fonctionnalités, y compris la gestion des comptes administrateurs.

## XV. Rapports et historiques :

- Les rapports peuvent être filtrés par période, utilisateur, catégorie, status.
- Les historiques sont consultables et exploitables en formats CSV ou PDF.
- Le système permet de **générer un fichier PDF récapitulatif** de l'état global de l'inventaire à tout moment.

## XVI. Sécurité et accès :

- Authentification obligatoire par login et mot de passe.
- Gestion fine des permissions pour limiter l'accès aux fonctionnalités sensibles.

## XVII. Architecture du projet

Dans cette partie je vais vous donner la structure architecturale du projet en générale

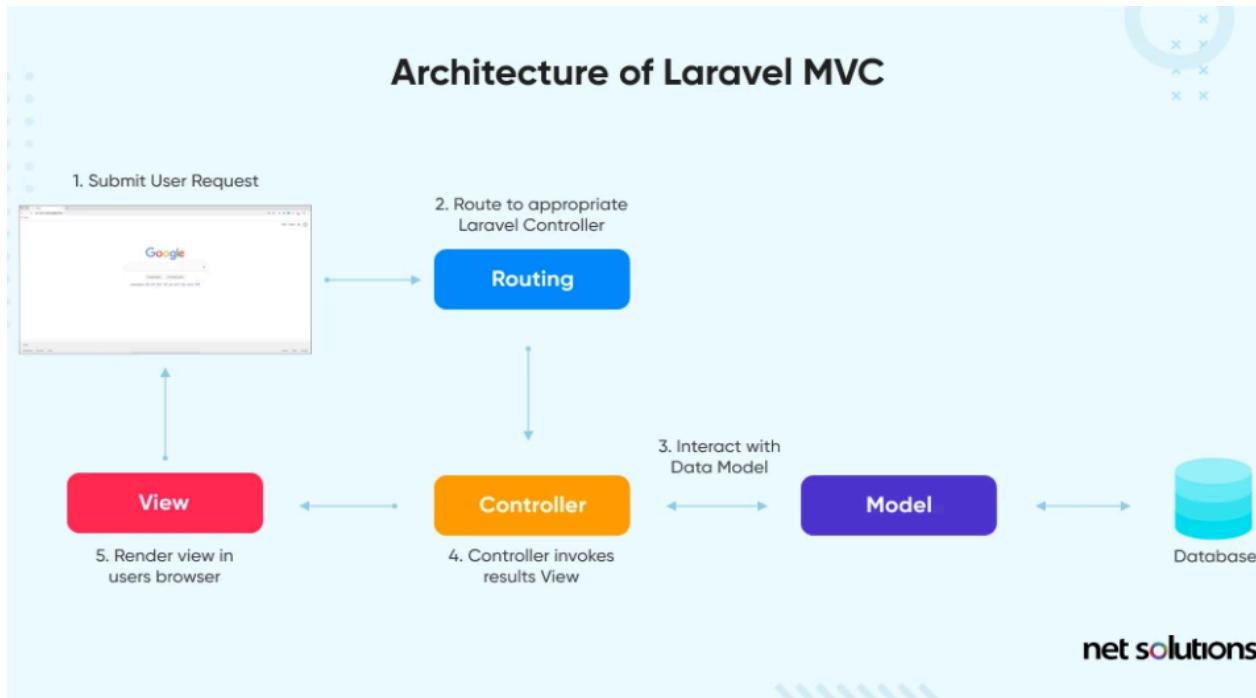


Figure 3 : Architecture du projet

Ce schéma représente les différents composants de l'application de gestion d'inventaire développée avec Laravel et MySQL. Ci-dessous, une définition de partie du projet:

- **Base de données (MySQL)** : Stocke toutes les informations relatives aux utilisateurs, matériels, catégories et affectations.
- **Front-end** : Interface utilisateur développée avec Blade (le moteur de templates de Laravel), HTML, CSS et JavaScript, permettant aux utilisateurs de gérer l'inventaire (ajout, modification, consultation).
- **Back-end (Laravel)** : Application développée en PHP avec le framework Laravel, qui implémente la logique métier, les contrôleurs, les routes et les API nécessaires à la gestion des inventaires.
- **Serveur web (Apache via XAMPP)** : Héberge l'application Laravel et permet l'accès via un navigateur en réseau local.

## Chapitre 3 : Gestion du projet.

---

Ce chapitre présente les principales étapes de réalisation de l'application de gestion d'inventaire, les tâches effectuées et leur planification, ainsi que la méthodologie de gestion agile adoptée.

## Collecte des besoins

- Analyse du cahier des charges
- Échanges avec les utilisateurs finaux pour cerner les attentes fonctionnelles

## Étude de l'existant et benchmark

- Analyse des outils similaires pour en extraire les bonnes pratiques et points d'amélioration

## Modélisation et conception

- Conception de la base de données MySQL
- Réalisation des diagrammes UML (cas d'utilisation, classes)

## Développement de l'application

- **Backend** : développement en Laravel (API, contrôleurs, logique métier)
- **Frontend** : interfaces avec HTML, Tailwind CSS et JavaScript
- **Sécurité** : mise en place de l'authentification et des permissions

## Tests et validation

- Exécution de tests fonctionnels et techniques
- Correction des bugs et validation du fonctionnement global

## Livraison et documentation

- Livraison de la version finale de l'application
- Rédaction de la documentation technique et utilisation.

## I. Tâches effectuées

### ✓ Liste des tâches réalisées durant le projet

#### 📌 1. Prise de connaissance du besoin

- Réunion avec le tuteur de stage pour comprendre les besoins de l'entreprise.
- Étude du fonctionnement actuel de la gestion d'inventaire.

#### 📌 2. Analyse et conception

- Analyse des utilisateurs et de leurs rôles (admin, utilisateur, etc.).
- Définition des fonctionnalités principales (CRUD utilisateurs, matériels, affectations...).
- Conception du schéma de la base de données (modèle conceptuel et relationnel).
- Création de diagrammes UML (cas d'utilisation, classes, etc.).
- Choix des technologies (Laravel, MySQL, Blade, Tailwind CSS, etc.).

#### 📌 3. Mise en place de l'environnement de développement

- Installation de XAMPP (Apache, MySQL, PHP).
- Création d'un projet Laravel.
- Configuration de la base de données (fichier `.env`).
- Initialisation du dépôt Git (optionnel).

#### 📌 4. Développement back-end (Laravel)



Complete the installation of Docker Desktop  
 Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.  
 Use advanced settings  
You manually set your preferred configurations.

Next

- Création des migrations pour les tables (**utilisateurs, matériels, catégories, affectations**, etc.).
- Développement des modèles Eloquent.
- Création des contrôleurs pour gérer les opérations CRUD.
- Mise en place des routes web et/ou API.
- Intégration de l’authentification (ex : Sanctum...).
- Implémentation des règles de validation et des messages d’erreurs.

## 📌 5. Développement front-end (Blade + Tailwind CSS)

- Création des interfaces utilisateur (pages Blade).
- Intégration des formulaires de saisie et tableaux de données.
- Mise en place du système de filtre, de recherche et de pagination.
- Utilisation de Tailwind CSS pour le design et la responsivité.
- Intégration de Chart.js pour les statistiques (dashboard).

## 📌 6. Fonctionnalités spécifiques

- Système d’affectation de matériels à des utilisateurs.
- Gestion des statuts (en stock, affecté).
- Génération de rapports ou export (ex :PDF, CSV).

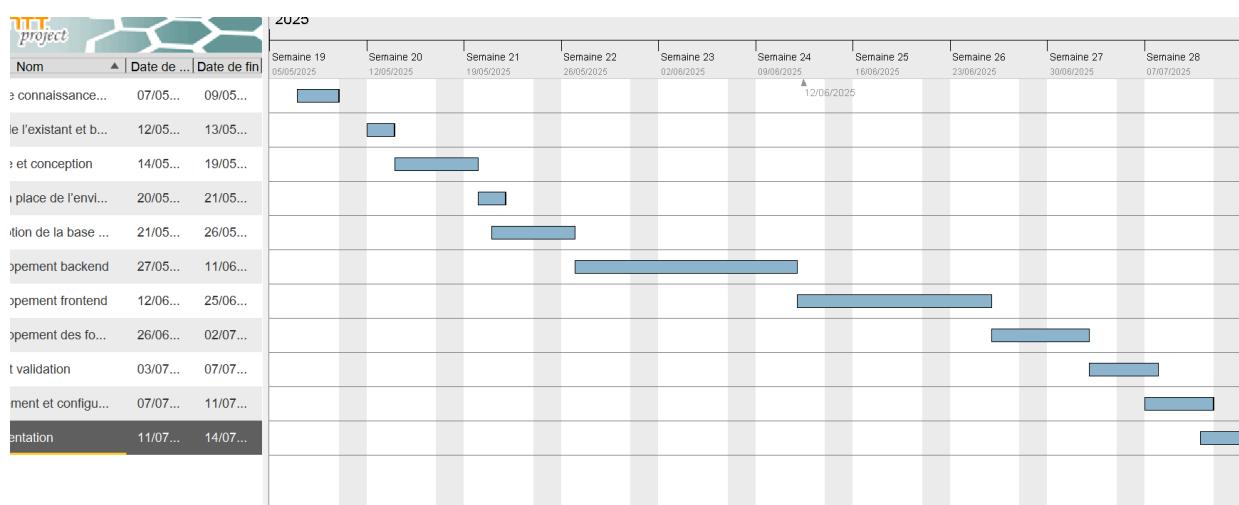
- Tableau de bord récapitulatif (avec statistiques, graphiques).
- Gestion des rôles et des droits d'accès.

## 📌 7. Tests et validation

- Test unitaire des fonctionnalités développées.
- Tests utilisateurs (vérification des cas d'usage réels).
- Correction des bugs et ajustements selon les retours.

## 📌 8. Déploiement

- Hébergement de l'application sur un serveur local (XAMPP).
- Configuration de l'accès depuis d'autres postes sur le réseau.
- Création d'un certificat SSL auto signé (si HTTPS requis).
- Documentation de l'installation et du fonctionnement.



**FIGURE Diagramme de GANTT**



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >

## Chapitre 4 : Etude Conceptuelle.

---

Ce chapitre est consacré à l'étude conceptuelle du projet, en utilisant la méthode UML. L'objectif est de fournir une modélisation fidèle aux besoins fonctionnels exprimés dans le cahier des charges. Cette étape permet de structurer et planifier l'implémentation de l'application de gestion d'inventaire.



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >

## I. Introduction

Au cours de notre formation, nous avons appris que la phase de conception logicielle est essentielle avant toute implémentation. Elle permet de mieux comprendre le fonctionnement global du système, d'anticiper les éventuels problèmes et de proposer des solutions en amont. Une conception soignée facilite le développement et contribue à la robustesse et la fiabilité de l'application.

Pour cette étape, nous avons utilisé UML, un langage de modélisation graphique largement reconnu pour la conception des systèmes orientés objet. Il permet de représenter de manière claire et structurée les différents aspects fonctionnels et techniques du système.

## II. Définition UML

UML (Unified Modeling Language) est un langage de modélisation graphique destiné à analyser, concevoir et documenter des systèmes logiciels, notamment orientés objet. Grâce à ses diagrammes, UML offre une vision simple et intuitive d'un système, permettant aux développeurs, analystes et parties prenantes de partager une compréhension commune.

UML comprend plusieurs types de diagrammes, permettant de décrire :

- III.  **Le comportement** du système (ex. : diagrammes de cas d'utilisation, de séquence).
- IV.  **La structure statique** (ex. : diagrammes de classes).
- V.  **L'interaction dynamique** entre les composants du système.

## VI. Diagramme de cas d'utilisation

-  **Diagramme de cas d'utilisation général**  
Il présente une vue d'ensemble des fonctionnalités offertes par l'application aux différents types d'utilisateurs (administrateur, employé, auditeur...).
-  **Diagramme de cas d'utilisation de gestion des utilisateurs**  
Il décrit les fonctionnalités permettant de créer, mettre à jour, désactiver ou supprimer des utilisateurs.
-  **Diagramme de cas d'utilisation de gestion des matériels**  
Il modélise les interactions liées à l'ajout, la modification, la suppression et la consultation des équipements dans le stock.

- **✓ Diagramme de cas d'utilisation de gestion des affectations**

Il décrit le processus d'affectation et de restitution de matériel aux utilisateurs, avec la traçabilité des mouvements.

- **✓ Diagramme de cas d'utilisation de gestion des catégories**

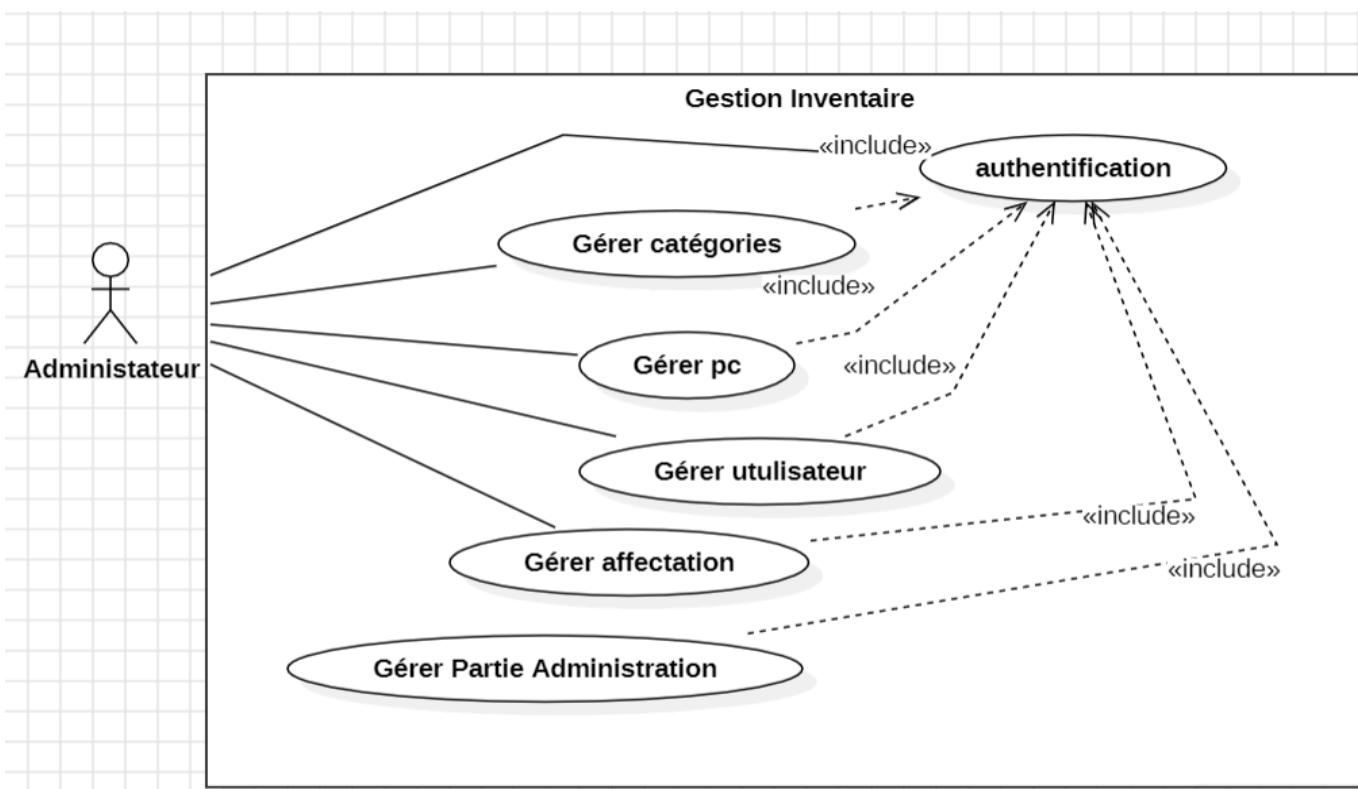
Il modélise les interactions liées à l'ajout, la modification, la suppression et la consultation des catégories.

- **✓ Diagramme de cas d'utilisation de gestion de l'administration**

Cette partie décrit les fonctions spécifiques réservées aux administrateurs du système, notamment la création, modification et suppression des comptes administrateurs, ainsi que la gestion des priviléges associés (droit de modification, gestion des rôles, contrôle d'accès aux fonctionnalités sensibles).

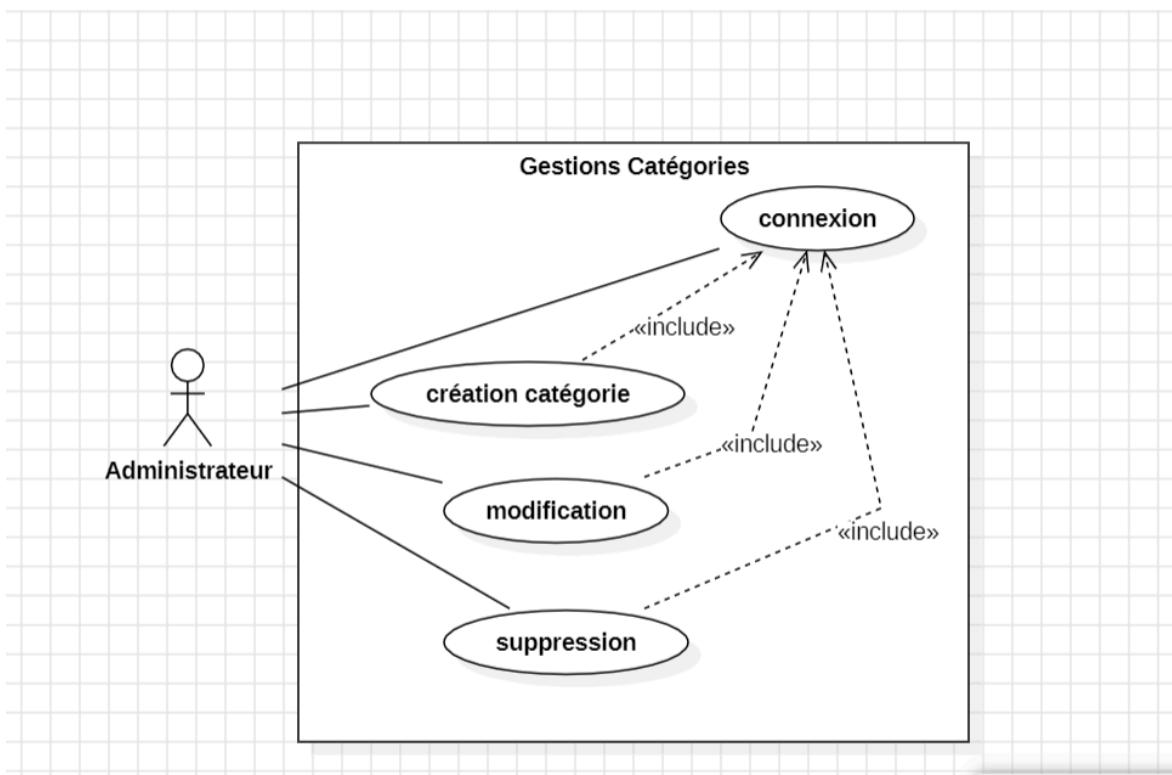
### 1) Diagramme de cas d'utilisation général

La figure suivante illustre le diagramme de cas d'utilisation général de l'application.



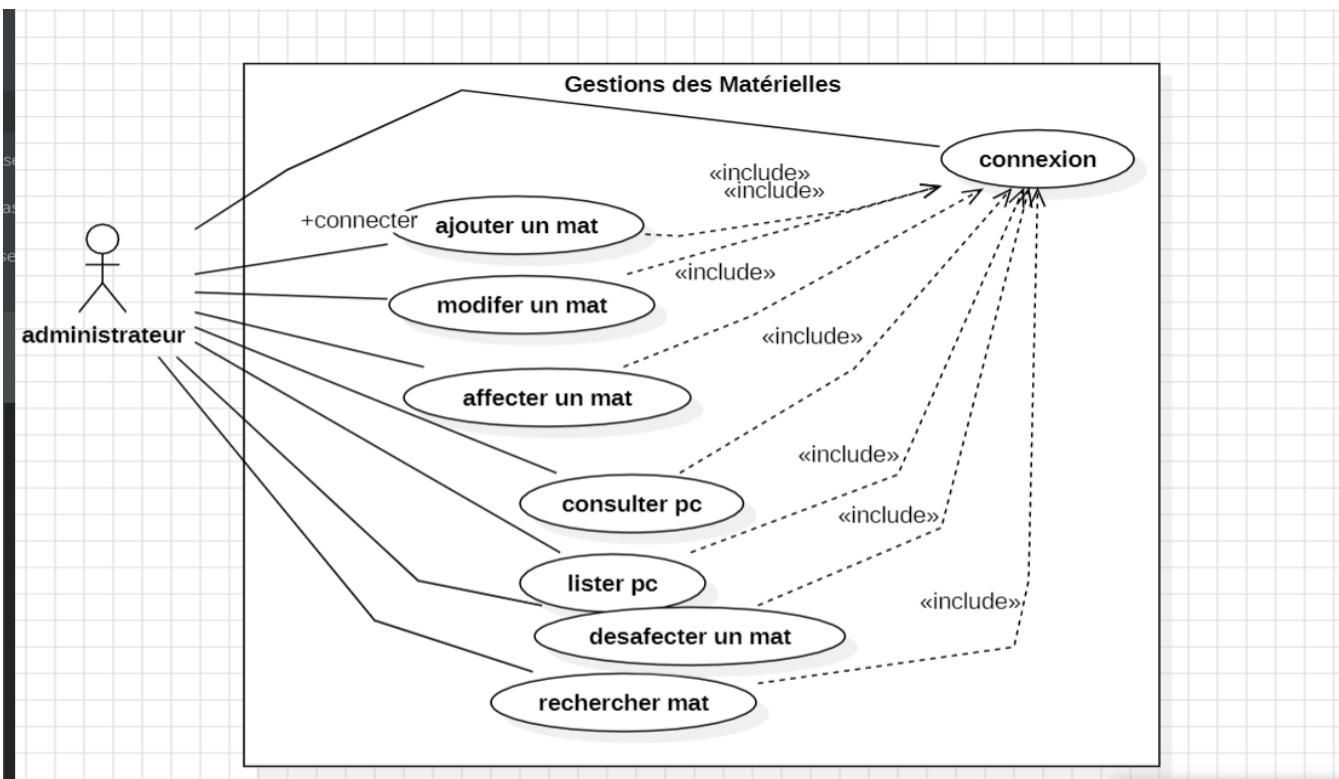
## 2) Diagramme de cas d'utilisation – Catégories

La figure suivante illustre les cas d'utilisation liés à la gestion des catégories.



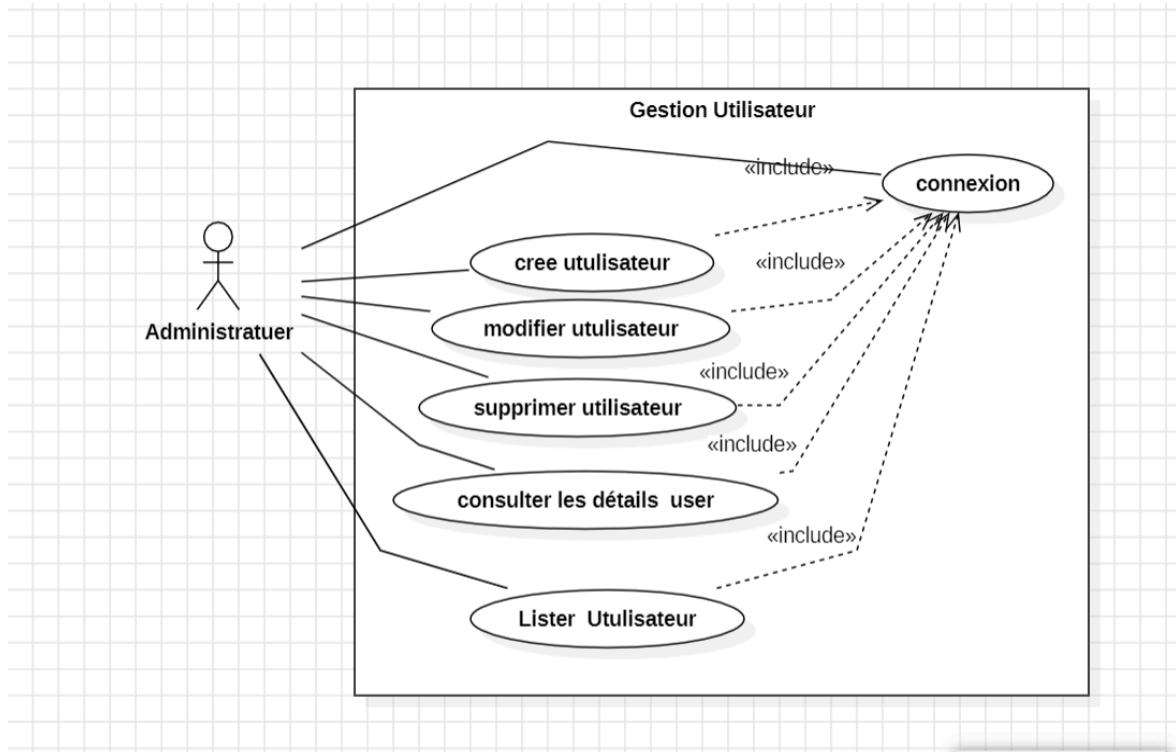
## 3) Diagramme de cas d'utilisation – Matériels

La figure suivante illustre les cas d'utilisation liés à la gestion des matériels



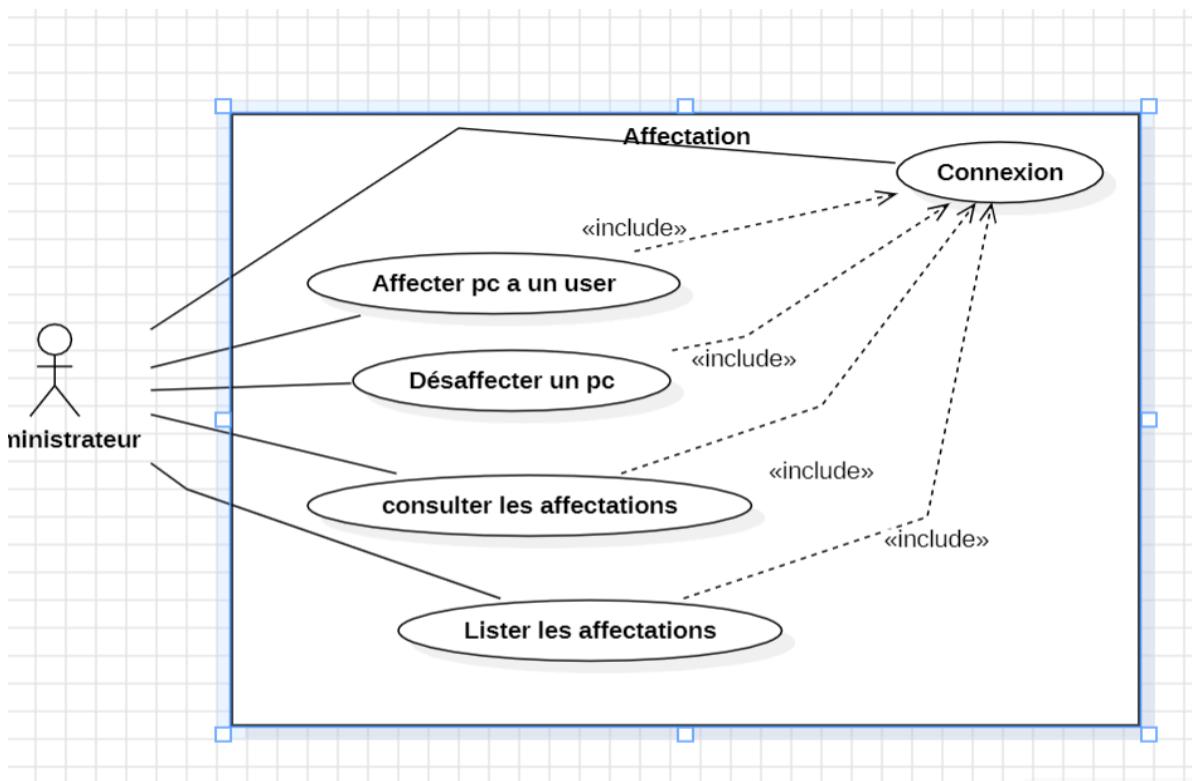
#### 4)Diagramme de cas d'utilisation – Utilisateurs.

La figure suivante présente les différents cas d'utilisation associés à la gestion des utilisateurs.



#### 5)Diagramme de cas d'utilisation – Affectations

La figure suivante illustre les cas d'utilisation liés à la gestion des affectations.





Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >

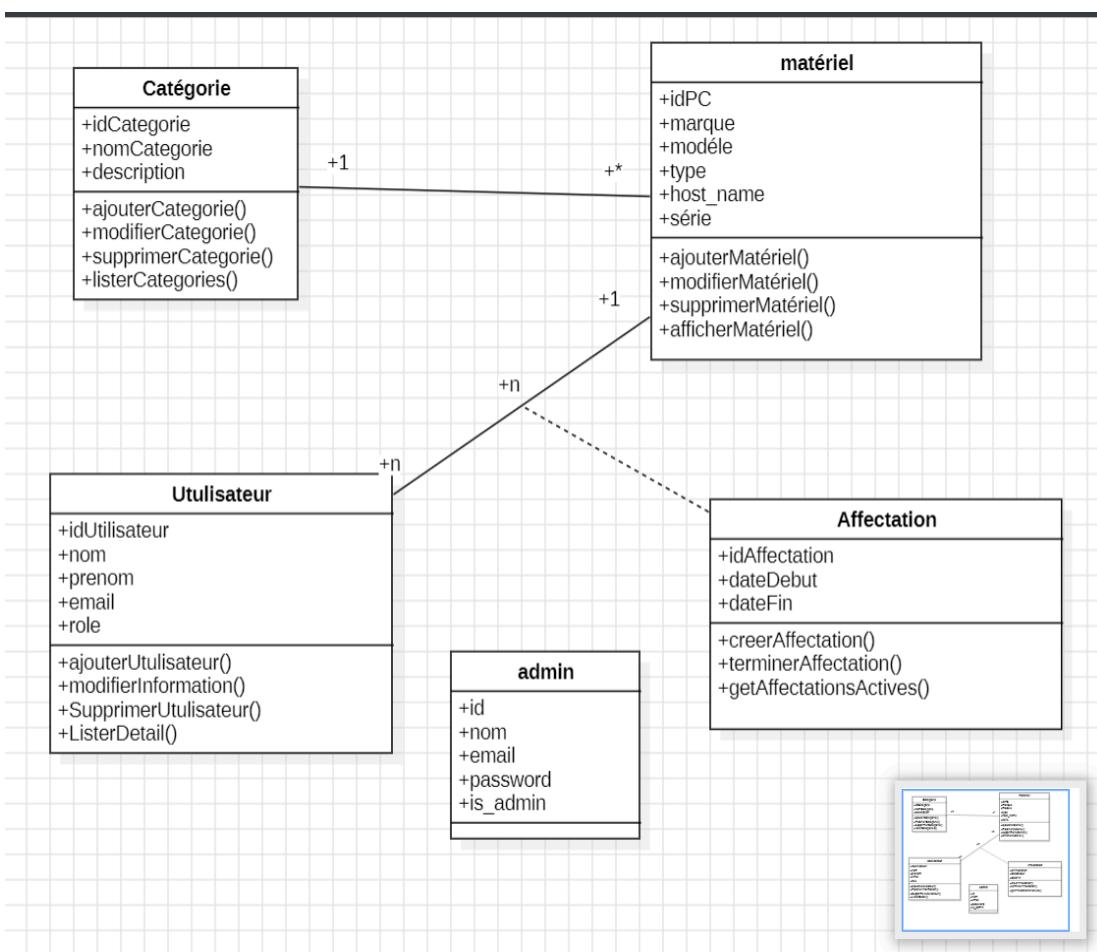
## Diagramme de classe

**Le diagramme de classes** est un diagramme statique. Il représente la vue statique d'une application. Le diagramme de classes n'est pas seulement utilisé pour visualiser, décrire et documenter différents aspects d'un système mais aussi pour construire du code exécutable de l'application logicielle.

**Le diagramme de classes** décrit les attributs et les opérations d'une classe ainsi que les contraintes imposées au système. Les diagrammes de classes sont largement utilisés dans la modélisation des systèmes orientés objet car ce sont les seuls diagrammes UML, qui peuvent être mappés directement avec des langages orientés objet.

**Le diagramme de classes** montre une collection de classes, interfaces, associations, collaborations et contraintes. Il est également connu sous le nom de diagramme structurel.

La figure suivante représente le diagramme de classe de l'application :





Complete the installation of Docker Desktop  
 Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.  
 Use advanced settings  
You manually set your preferred configurations.

Next

## 1.Catégorie

Représente les types de matériel (ex : ordinateurs, imprimantes).

- Attributs : identifiant, nom, description
- Méthodes : ajouter, modifier, supprimer, lister
- Relation : une catégorie contient plusieurs matériels ( $1 \rightarrow *$ ).

## 2.Matériel

Représente un équipement informatique (PC, serveur, etc.).

- Attributs : identifiant, marque, modèle, type, nom machine, numéro de série
- Méthodes : ajouter, modifier, supprimer, afficher
- Relation : appartient à une seule catégorie ( $* \rightarrow 1$ ), et peut être affecté à plusieurs utilisateurs via des affectations.

## 3.Utilisateur

Personne à qui un matériel peut être attribué.

- Attributs : identifiant, nom, prénom, email, rôle
- Méthodes : ajouter, modifier, supprimer, lister
- Relation : peut avoir plusieurs affectations ( $1 \rightarrow n$ ).

## 4.Affectation

Lien entre un utilisateur et un matériel pour une durée donnée.

- Attributs : identifiant, date début, date fin
- Méthodes : créer, terminer, obtenir affectations actives
- Relation : association many-to-many entre utilisateurs et matériels.

## 5.Admin

Administrateurs du système.

- Attributs : identifiant, nom, email, mot de passe, privilèges (booléen)
- Relation : généralement lié aux utilisateurs, mais ici autonome.

---

## Relations générales

- Une catégorie contient plusieurs matériels ( $1 \rightarrow *$ )
- Un matériel peut être affecté à plusieurs utilisateurs via l'affectation (relation many-to-many via Affectation)
- Affectation relie un utilisateur à un matériel sur une période définie.



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >

## VII. Diagramme de Séquence

Un **diagramme de séquence** est un diagramme UML qui représente la séquence de messages entre les objets au cours d'une interaction. Un diagramme de séquence comprend un groupe d'objets, représentés par des lignes de vie, et les messages que ces objets échangent lors de l'interaction.

D'après la conception que j'ai fait durant cette période j'ai réalisé 6 diagrammes de séquences.

- Diagramme de séquence pour l'authentification du fournisseur.
- Diagramme de séquence pour passer une transaction.
- Diagramme de séquence pour valider une transaction.
- Diagramme de séquence pour créer un compte.
- Diagramme de séquence pour ajouter un rôle
- Diagramme de séquence d'ajouter utilisateur



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >

## Chapitre 5 : Outils de développement.

---

Ce chapitre a pour but de définir tous les outils de développement, les outils de modélisations et l'éditeur de code utilisé dans ce projet.

## I. Introduction

La phase de choix des outils de développement, modélisation et éditeur de code est une phase indispensable dans la réalisation de chaque projet informatique. Ce chapitre a pour objectif de mettre en évidence les outils que ça soit de conception, de réalisation etc.

## II. Outils de développements

### 1. PHP



Figure : PH

**PHP** (officiellement, ce sigle est un acronyme récursif pour PHP Hypertext Preprocessor) est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement d'applications web.

### 2. Laravel



Figure : Laravel

**Laravel** est un framework Web utilisé pour créer des applications Web extensibles basées sur PHP à grande échelle

### 3. MYSQL



Figure : MySQL

**MySQL** est un système de gestion de bases de données relationnelles (SGBDR) open source utilisé pour stocker et gérer des données. Sa fiabilité, ses performances, son évolutivité et sa facilité d'utilisation font de MySQL un choix plébiscité par les développeurs. En réalité, vous le trouverez au cœur des applications exigeantes et à fort trafic telles que Facebook, Netflix, Uber, Airbnb, Shopify et [Booking.com](#).

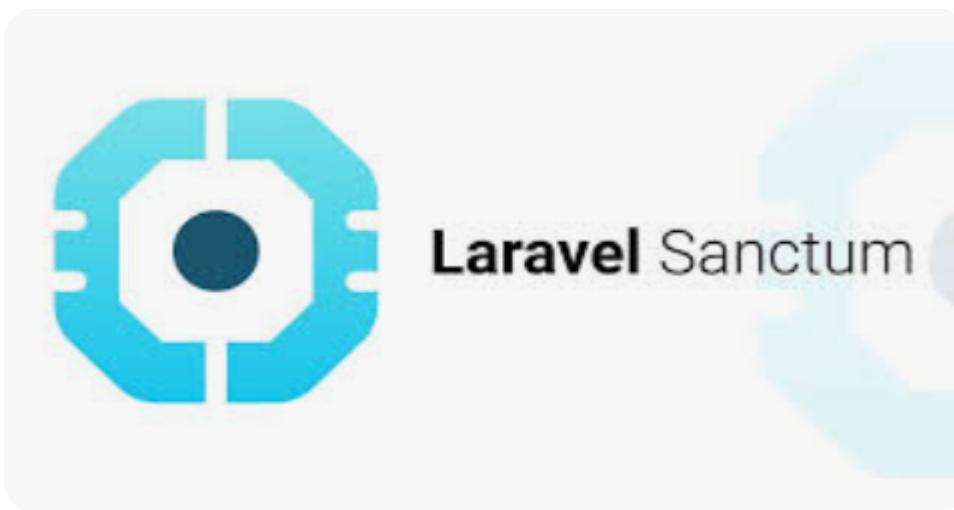


Figure : Laravel Sanctum

**Laravel Sanctum** est un système d'authentification pour les applications monopage (SPA), les applications mobiles et les API de base basées sur des jetons



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >

#### 4. JavaScript



Figure : JavaScript

**JavaScript** est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les langages HTML et CSS, JavaScript est au cœur des langages utilisés par les développeurs web.

#### 5. Tailwind CSS



Figure : Spring Boot

**Tailwind CSS** est un framework CSS open source. Contrairement à d'autres frameworks, comme Bootstrap, il ne fournit pas de classes prédéfinies pour des éléments tels que des boutons ou des tableaux. Il crée plutôt une liste de classes CSS « utilitaires » permettant de styliser chaque élément en les combinant



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >

## 6. Github



Figure : GitHub

**GitHub** est considéré comme un outil essentiel pour les ingénieurs logiciels, et sa popularité est inégalée. Il accueille actuellement plus de 25 millions d'utilisateurs.

## 7. Apache http Server



Figure : Apache

**Apache** est un logiciel pour les serveurs web gratuits et open-source qui sont environ 46% des sites web à travers le monde.

## 8.Postman



Figure : PostMan

**Postman** est un logiciel permettant de créer et de tester des requêtes HTTP. Il vous permet de les personnaliser dans les plus fins détails grâce à une interface ergonomique et intuitive. Vous pouvez choisir la méthode de la requête, entrer l'URL du serveur que vous voulez interroger, et ajouter tous les paramètres possibles pour une requête HTTP. Le logiciel tient un historique de vos requêtes. Il est très utile pour tester une api.

### III. Outils de modélisations

## 1. UML



Figure : UML

**Le Langage de Modélisation Unifié**, de l'anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les secteurs du développement logiciel et aussi dans conception orientée objet.

## IV. Editeurs de code

### 1. VS CODE



Figure : Eclipse

**Visual Studio Code**, communément appelé **VS Code**, est un environnement de développement intégré (EDI) développé par Microsoft.

## Chapitre 6 : Réalisation.

---

Ce chapitre a pour but de présenter les interfaces graphiques de la partie client et aussi la partie administration de l'application réalisée dans le projet avec une petite description décrive son fonctionnement.

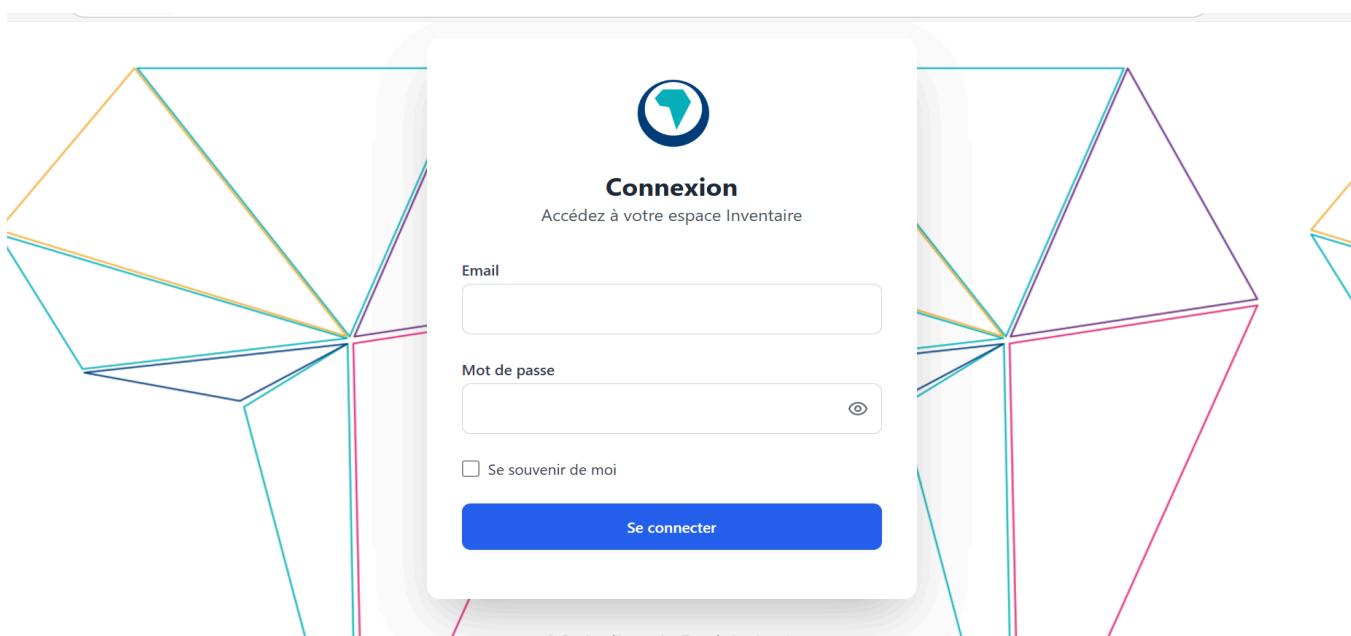
## I. Partie Admin

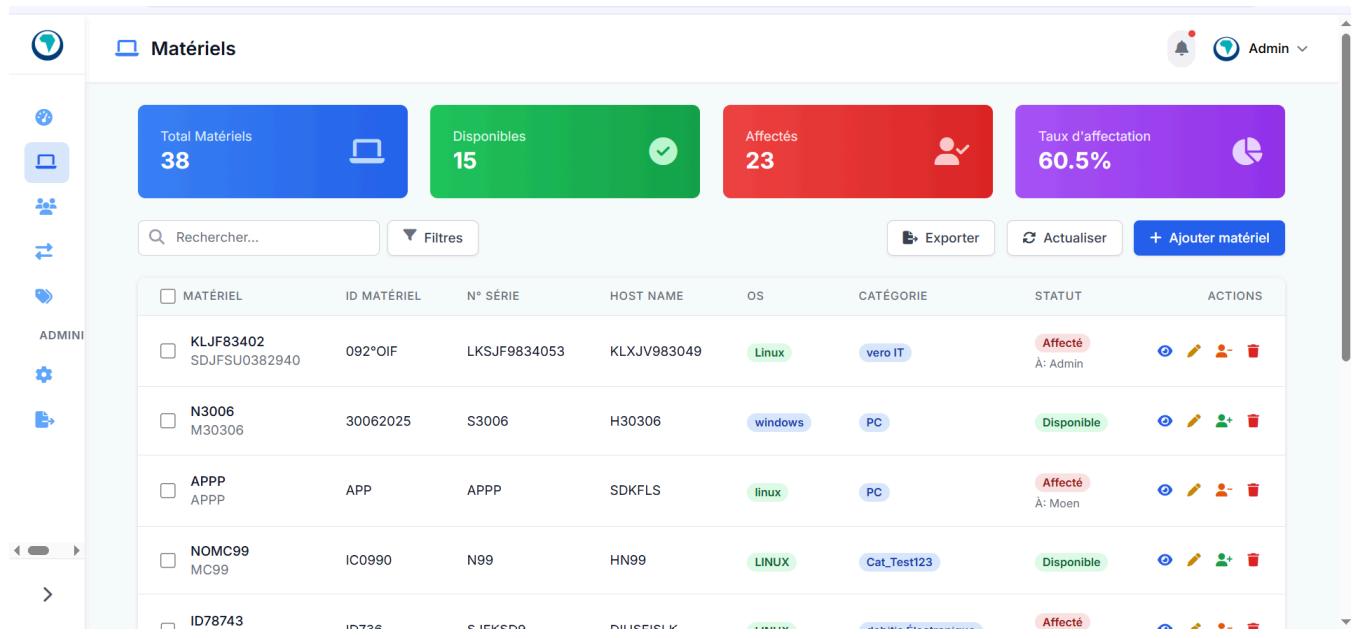
Dans la première partie de ce chapitre, je vais vous présenter les interfaces graphiques de la partie admin réalisées dans le projet.

### a) Authentification

Cette page est spécialisée à l'authentification par l'insertion du nom d'adminet le mot de passe pour s'authentifier à l'application.

**Figure : Page d'authentification**

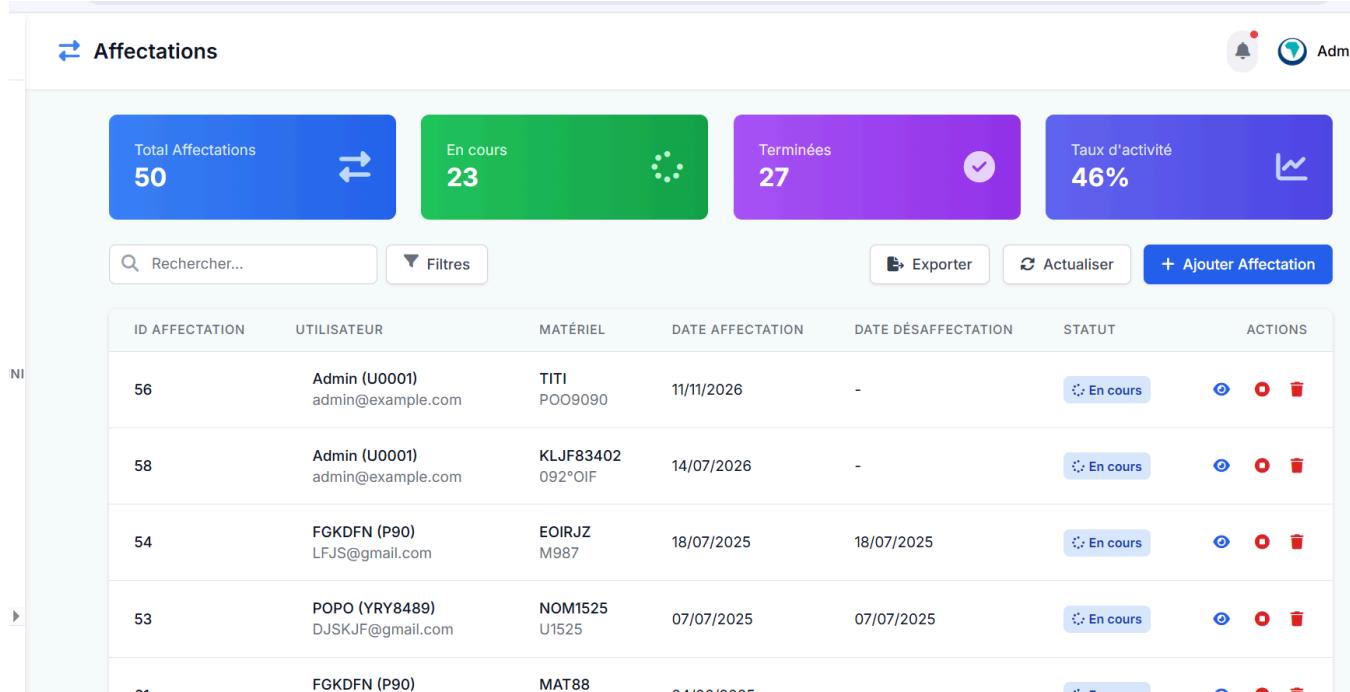




**Figure : Page Materials**

### b) Informations matériaux

Cette page a pour but d'afficher les informations des Matériels et aussi avec leurs statuts.



**Figure : Page d'information des affectation**

### c) Liste des Affectations

Sur cette page j'ai affiché la liste des affectations des utilisateurs avec les matériels



Complete the installation of Docker Desktop.  
 Use recommended settings (requires administrator password)  
 Docker Desktop automatically sets the necessary configurations that work for most developers.  
 Use advanced settings  
 You manually set your preferred configurations.

Install

**Utilisateurs**

Total Utilisateurs	Utilisateurs avec Affectation	Administrateurs	Matériaux Affectés
55	11	15	23

Rechercher... Filtres Exporter Actualiser Ajouter utilisateur

**ADMIN**

UTILISATEUR	ID_UTILISATEUR	NOM	PRÉNOM	EMAIL	RÔLE	ACTIONS
Yaya2026 etc2026	etc2026	Yaya2026	TAHA2026	Yaya@example.com	Chef	
NOM99 ID9999	ID9999	NOM99	PRENOM99	EMAIL99@gmail.com	Junior	
iuoiu iuoiu	iuoiu	iuoiu	uoioi	uoiu@gmail.com	Junior	
i;cvx: kd;,	kd;,	;,cvx:	;,,xcv	kdfsmfSDFs@gmail.com	Junior	

Figure 45 : Page liste des utilisateurs

e) Informations utilisateurs

Cette page a pour but d'afficher les informations des Utilisateurs

**Catégorie**

Total Categories	Total Matériaux	Catégorie la plus utilisée	Matériaux affectés
63	38	CAT104	23

Rechercher... Filtres Exporter Ajouter Catégorie

**ADMIN**

CATÉGORIE	ID_CATÉGORIE	LIBELLE	TOTAL MATERIELS	MATERIELS AFFECTÉS	MATERIELS DISPONIBLES	ACTIONS
Cat_Test123 C9999	C9999	Cat_Test123	1	0	1	
fugit Bureau01 CAT015	CAT015	fugit Bureau01	3	2	1	
debitis Bureau CAT035	CAT035	debitis Bureau	1	1	0	
facilis Bureau CAT050	CAT050	facilis Bureau	1	1	0	
ut Électronique CAT056	CAT056	ut Électronique	1	1	0	

Figure 46 : Page Informations Catégories

f) Informations Categories

Cette page a pour but d'afficher les informations des catégories.



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >


**Gestion d'Inventaire**

# Contrôle Général Groupe

## AUDIT SI



**PC total**  
 7

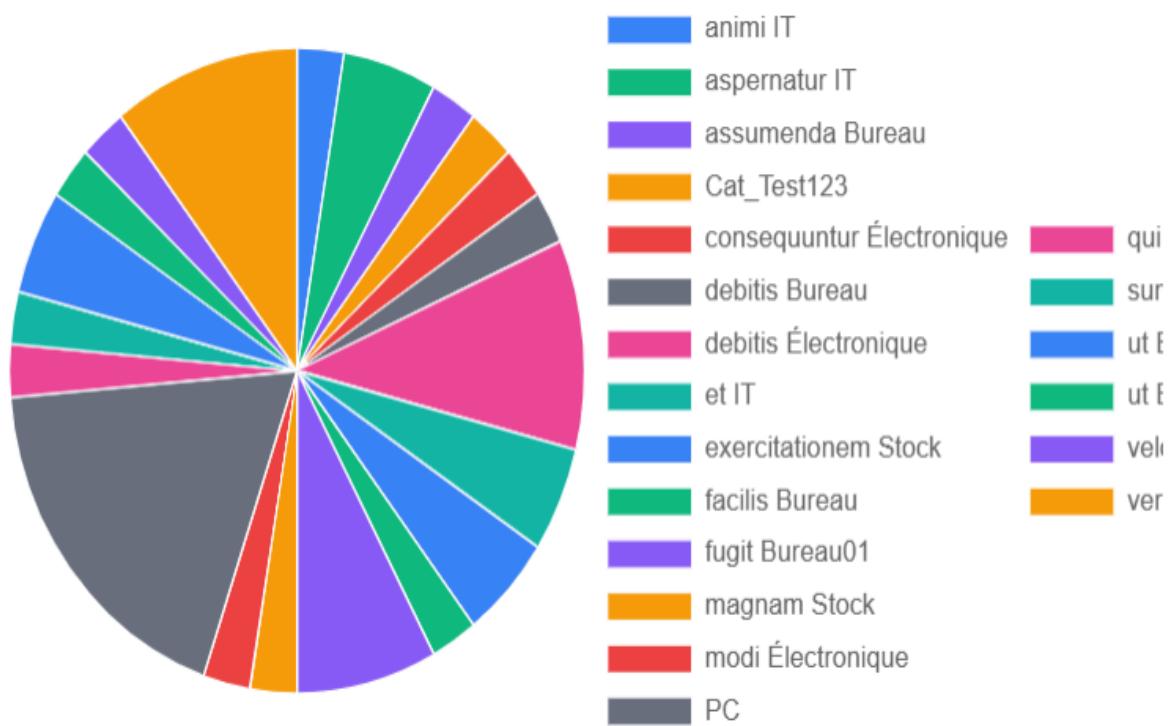

**Utilisateurs**  
 55


**PC actives**  
 2


**Catégories**  
 63

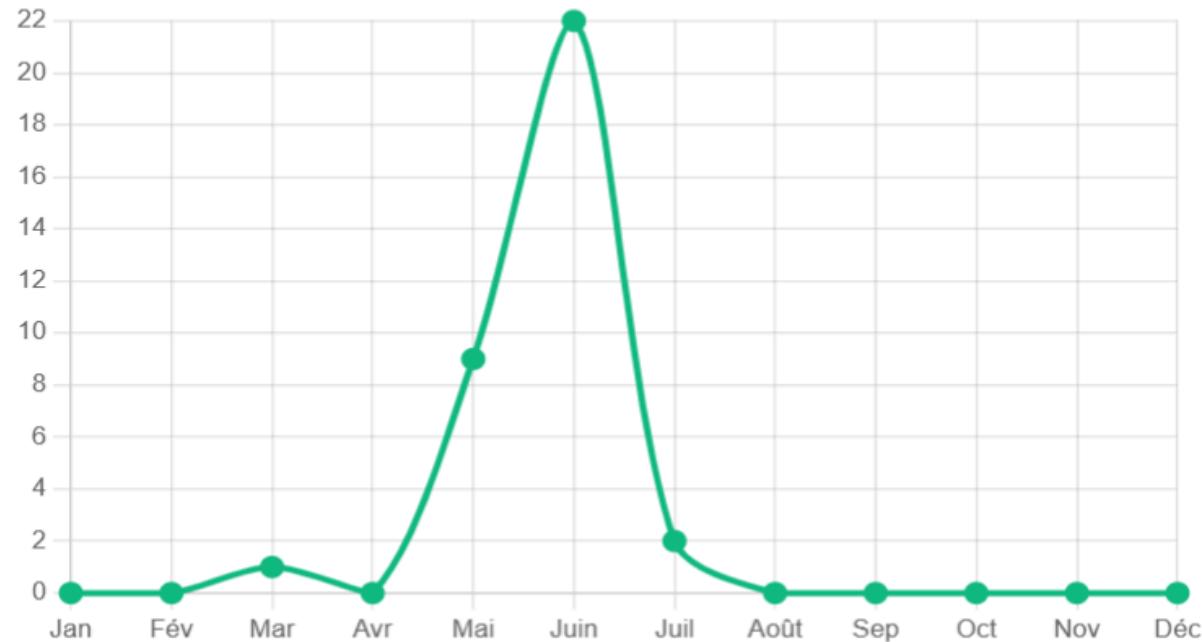
Figure : Page liste des KPIs

### Répartition des matériels par catégorie



Ce graphique permet de visualiser la distribution des différents types de matériels informatiques. Il est utile dans une application de gestion d'inventaire pour rapidement comprendre quelle catégorie domine ou est sous-représentée.

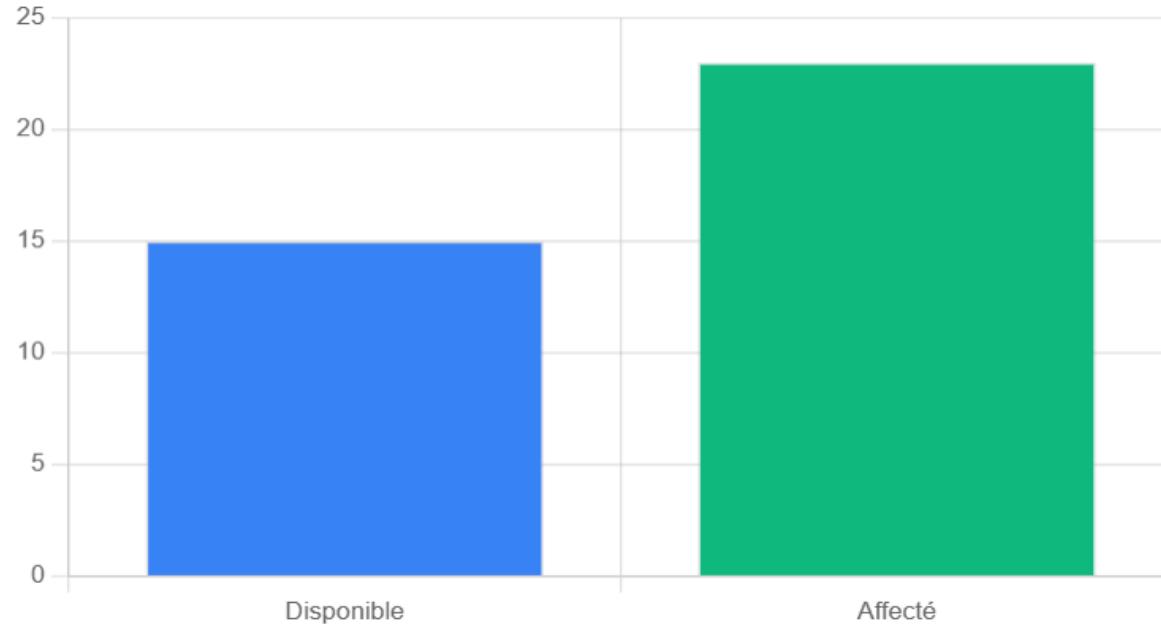
### Affectations par mois



Ce graphique permet d'**analyser la saisonnalité des affectations** :

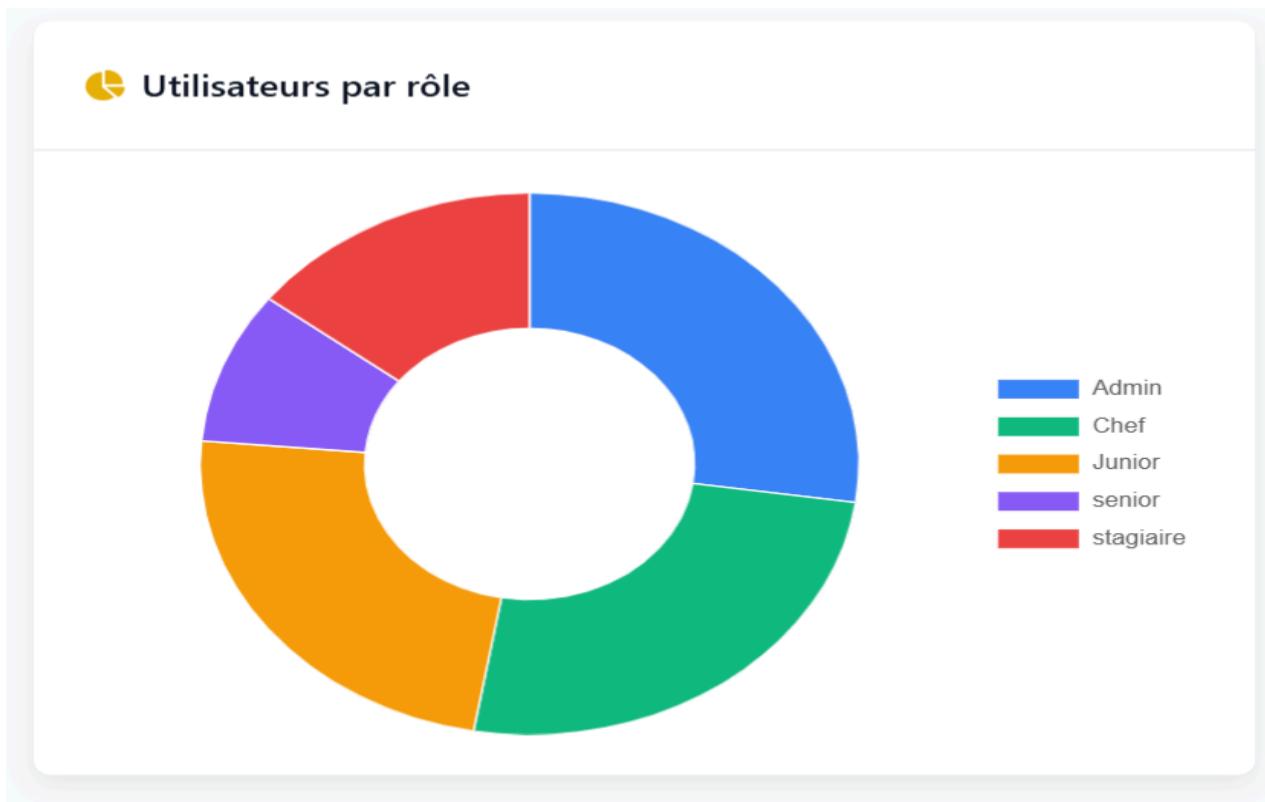
- Peut indiquer une planification trimestrielle ou liée à des périodes spécifiques (rentrée, fin de trimestre, etc.).
- Utile pour anticiper les besoins en matériel ou en personnel pendant les pics.

## Statut des matériels



Ce graphique offre une **vue rapide de l'état du stock** :

- Il permet d'équilibrer entre disponibilité et affectation.
- Pratique pour la gestion logistique dans un système d'inventaire.



Ce graphique offre une vue d'ensemble sur la répartition des utilisateurs par rôle :

- Permet de visualiser la répartition des rôles utilisateurs..
- Facilite la gestion des accès et des responsabilités.

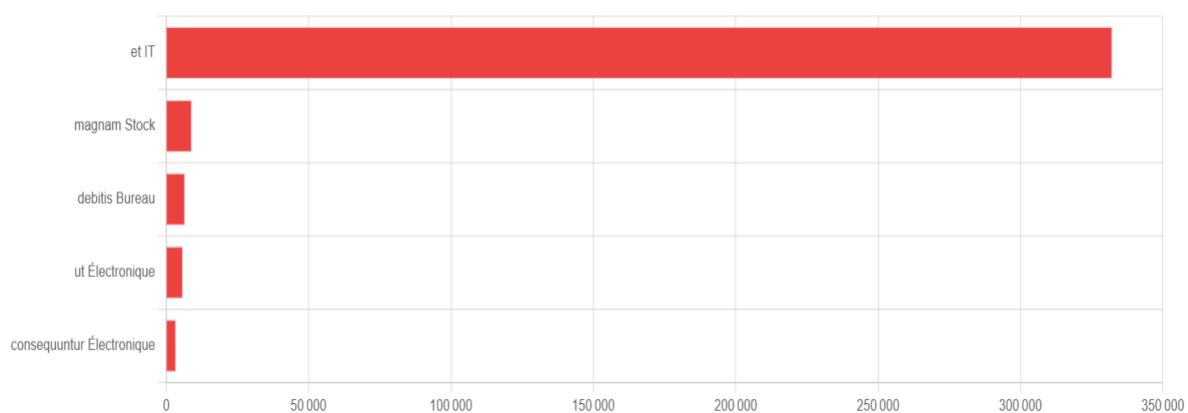


Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >

⌚ Durée moyenne des affectations par catégorie (en jours)


Ce graphique offre une vue d'ensemble sur la répartition des utilisateurs par rôle :

- Affiche les 5 catégories avec les durées d'affectation les plus longues.
- Utile pour repérer les usages prolongés et ajuster la gestion du matériel.

### g) Liste des 10 dernières Affectations

MATERIEL	CATEGORIE	UTILISATEUR	DATE AFFECTATION	STATUS
TITI	velo	Admin	11/11/2026	Active
KLJF83402	vero IT	Admin	14/07/2026	Active
EOIRJZ	et IT	FGKDFN	18/07/2025	Active
NOM1525	exercitationem Stock	POPO	07/07/2025	Active
MAT88	exercitationem Stock	FGKDFN	24/06/2025	Active
APPP	PC	Moen	23/06/2025	Active
Sanctum	PC	Admin	19/06/2025	Active
Scanner	facilis Bureau	Admin	07/06/2025	Active
Mat56	qui IT	Nelo	05/06/2025	Terminée

Affiche les 10 dernières affectations pour suivre les mouvements récents du matériel.



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >



Complete the installation of Docker Desktop.  
Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.  
Use advanced settings  
You manually set your preferred configurations.

Next



## Gestion des administrateurs

Dashboard

Déconnexion

### Liste des administrateurs

Gérez tous les administrateurs de votre application

+ Ajouter un administrateur

ID	NOM	EMAIL	ADMIN	ACTIONS
1	admin	admin@example.com	Admin	<button>Modifier</button> <button>Supprimer</button>
2	Admin	test@example.com	Admin	<button>Modifier</button> <button>Supprimer</button>
3	leo123	leo@example.com	Admin	<button>Modifier</button> <button>Supprimer</button>
4	hh	hh@example.com	Utilisateur	<button>Modifier</button> <button>Supprimer</button>
6	mm	mm@example.com	Admin	<button>Modifier</button> <button>Supprimer</button>
7	taha	taha@example.com	Utilisateur	<button>Modifier</button> <button>Supprimer</button>

### h) Ajouter administrateurs

Cette Page est dédiée pour ajouter un administrateur et donner des privilèges ainsi que modifier ou le supprimer.

Liste des administrateurs

Gérez tous les administrateurs de votre application

ID	NOM	EMAIL
1	admin	admin@example.com
2	Admin	test@example.com
3	leo123	leo@example.com
4	hh	hh@example.com
6	mm	mm@example.com

Ajouter un utilisateur

Remplissez les informations de l'utilisateur

Nom complet: Jean Dupont

Adresse email: jean.dupont@example.com

Mot de passe: \*\*\*\*\*

Confirmation: \*\*\*\*\*

Accès administrateur

Annuler Ajouter

Cette page est pour le Profil personnel des administrateurs.

Nom: admin

Email: admin@example.com

Mot de passe (laisser vide si inchangé)

Confirmer mot de passe

Enregistrer



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >

## Rapport de Synthèse - Inventaire

Indicateur	Valeur
Nombre total de matériels	37
Nombre total d'utilisateurs	55
Total des affectations (historique complet)	50
Total des affectations en cours	23
Total des affectations terminées	27
Matériels actuellement affectés	23
Matériels non affectés	14

## Utilité du Rapport d'Inventaire (PDF)

### 1. Vue d'ensemble centralisée

Résume l'état global du stock et des affectations en un seul document.

### 2. Suivi et traçabilité

Permet de conserver un historique des mouvements de matériel.

### 3. Communication facilitée

Partage rapide du rapport avec les équipes ou la direction.

### 4. Support pour les audits

Sert de preuve documentaire lors des contrôles ou vérifications internes.

### 5. Prise de décision

Aide à détecter les anomalies et à planifier les réaffectations ou renouvellements.

## Conclusion et Perspectives

Cette expérience de stage au sein de **Bmce Bank of Africa** m'a été très bénéfique. Elle m'a permis de découvrir les facettes du milieu professionnel. Au début du stage, j'avais un certain doute par rapport à ce que je voulais faire plus tard. Je sentais que j'avais besoin de réaliser quelque chose de concret. Grâce à ce stage, j'ai pu découvrir pour la première fois le monde de l'entreprise, et avec le projet que j'ai réalisé, j'ai vraiment eu l'impression d'avoir accompli quelque chose d'important.

Je tiens à souligner que si j'avais eu plus de temps, j'aurais aimé ajouter d'autres fonctionnalités à l'application, comme un système de notifications internes ou une interface mobile pour la gestion d'inventaire.

Ainsi, mon stage chez **Bmce Bank of Africa** a été très enrichissant. Il m'a permis de mettre en pratique ce que j'ai appris durant ma formation en Développement Informatique, et de renforcer mes compétences, notamment dans le développement web et la gestion des bases de données.

Un grand merci à **mon encadrant Mr Amine ESSAKALLI** ainsi qu'à l'ensemble du personnel de **Bmce Bank of Africa**, qui m'ont soutenu et accompagné tout au long de cette expérience professionnelle, laquelle restera gravée dans ma mémoire.



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next >



Complete the installation of Docker Desktop

- Use recommended settings (requires administrator password)  
Docker Desktop automatically sets the necessary configurations that work for most developers.

- Use advanced settings  
You manually set your preferred configurations.

Next