



كلية العلوم و التقنيات بطنجة
Faculté des Sciences et Techniques de Tanger



UNIVERSITÉ ABDELMALEK ESSAÂDI - FACULTÉ DES SCIENCES ET TECHNIQUES DE TANGER

Filière : Analytique des données - Module : Développement Web

Titre du projet

Gestion des ressources informatiques d'un Data Center

Réalisé par :

Ferdaouss Bniyech
Assia Mabrouk
Mariam chahaba
Meryem el khoumri

Encadré par :

Prof. AIT KBIR M'hamed

Prof. EL YUSUFI Yasyn

La Faculté des Sciences et Techniques de Tanger (FSTT)

Année universitaire : 2025 – 2026

Table des matières

Page de garde.....	1
Table des matières.....	2
Résumé (en français).....	4
Abstract (in English).....	4
Chapitre 1 : Étude et analyse du système.....	4
Introduction du chapitre.....	5
1.1 Présentation du Data Center.....	5
1.2 Problématique de gestion des ressources.....	5
1.3 Objectifs fonctionnels de l'application.....	6
1.4 Solution proposée.....	7
Chapitre 2 : Analyse fonctionnelle.....	7
Introduction du chapitre.....	8
2.1 Acteurs du système.....	8
Invité.....	8
Utilisateur interne.....	8
Responsable technique.....	8
Administrateur.....	9
2.2 Rôles et permissions.....	9
2.3 Cas d'utilisation.....	10
2.3.1 Diagramme de cas d'utilisation.....	10
2.4 Description des cas d'utilisation.....	11
Cas d'utilisation 1 : Consulter les informations publiques.....	11
Cas d'utilisation 2 : Effectuer une demande de réservation.....	11
Cas d'utilisation 3 : Traiter une demande de réservation.....	12
Cas d'utilisation 4 : Gérer les utilisateurs.....	12
Conclusion du chapitre.....	12
Chapitre 3 : Conception du système.....	13
Introduction.....	13
3.1 Architecture générale de l'application.....	13
3.2 Modèle de données.....	14
3.2.1 Entités principales et relations.....	14
3.2.2 Modèle logique et relationnel.....	15
3.3 Organisation des modèles.....	16
Conclusion du chapitre.....	17
Chapitre 4 : Réalisation de l'application.....	18
Introduction.....	18
4.1 Environnement de développement.....	18
4.2 Technologies utilisées.....	19
4.3 Organisation du projet Laravel.....	20
4.4 Implémentation des fonctionnalités principales.....	21
4.4.1 Gestion des utilisateurs.....	21
4.4.2 Gestion des ressources.....	21
4.4.3 Système de réservation.....	21
4.4.4 Notifications internes.....	22
4.4.5 Statistiques et tableaux de bord.....	22

4.4.6 Journalisation et traçabilité.....	22
4.4.7 Sécurité et protection des accès.....	22
4.5 Organisation des routes et contrôleurs.....	23
Conclusion du chapitre.....	23
Chapitre 5 : Sécurité et gestion des accès.....	24
Introduction.....	24
5.1 Authentification et autorisation.....	24
Authentification.....	24
Autorisation.....	24
5.2 Gestion des rôles et permissions.....	25
5.3 Protection des routes (Middleware).....	25
5.4 Journalisation et traçabilité.....	26
5.5 Sécurité supplémentaire.....	26
5.5.1 Hachage des mots de passe.....	26
5.5.2 Protection des données.....	26
5.5.3 Gestion des sessions.....	27
Conclusion du chapitre.....	27
Chapitre 6 : Interfaces de l'application.....	28
Introduction.....	28
6.1 Interface Invité.....	28
Fonctionnalités principales :.....	28
Caractéristiques de l'interface :.....	28
6.2 Interface Utilisateur interne.....	30
Fonctionnalités principales :.....	30
Caractéristiques de l'interface :.....	32
6.3 Interface Responsable technique.....	32
Fonctionnalités principales :.....	32
Caractéristiques de l'interface :.....	34
6.4 Interface Administrateur.....	34
Fonctionnalités principales :.....	34
Caractéristiques de l'interface :.....	36
6.5 Ergonomie et design.....	36
Conclusion du chapitre.....	37
Conclusion générale.....	37
Bilan du projet.....	38
Objectifs atteints.....	38
Limites du système.....	39
Perspectives d'évolution.....	39
Conclusion finale.....	39
Bibliographie / Webographie.....	40

Résumé (en français)

Ce projet a pour objectif de développer une application web de gestion des ressources informatiques d'un Data Center. L'application permet de centraliser la réservation, l'allocation et le suivi des ressources telles que les serveurs, machines virtuelles, équipements réseau et systèmes de stockage.

Elle propose une gestion différenciée selon les profils utilisateurs : invités, utilisateurs internes, responsables techniques et administrateurs. Les principales fonctionnalités incluent la gestion des utilisateurs et des rôles, un système de réservation en ligne avec suivi des statuts, la notification interne des utilisateurs, et des tableaux de bord statistiques pour analyser l'utilisation des ressources.

Le développement repose sur le framework Laravel avec une base de données MySQL, utilisant uniquement du CSS et JavaScript personnalisés pour l'interface utilisateur. Le projet assure ainsi une gestion centralisée, sécurisée et transparente des ressources du Data Center, tout en offrant une interface ergonomique et des fonctionnalités évolutives pour l'avenir.

Abstract (in English)

This project aims to develop a web application for managing IT resources in a Data Center. The application centralizes the reservation, allocation, and monitoring of resources such as servers, virtual machines, network equipment, and storage systems.

It provides differentiated management based on user profiles: guests, internal users, technical managers, and administrators. Key functionalities include user and role management, an online reservation system with status tracking, internal notifications, and statistical dashboards to analyze resource usage.

The development is based on the Laravel framework with a MySQL database, using only custom CSS and JavaScript for the user interface. The project ensures a centralized, secure, and transparent management of Data Center resources, while offering an ergonomic interface and scalable features for future enhancements

Chapitre 1 : Étude et analyse du système

Introduction du chapitre

Ce premier chapitre est consacré à l'étude et à l'analyse du système existant ainsi qu'à la compréhension du contexte général du projet. Il permet de présenter le Data Center concerné, de mettre en évidence les problématiques liées à la gestion des ressources informatiques et de définir les objectifs fonctionnels attendus de l'application. Enfin, une solution globale est proposée afin de répondre aux besoins identifiés.

1.1 Présentation du Data Center

Un Data Center est une infrastructure informatique centralisée destinée à héberger, traiter, stocker et distribuer des ressources numériques essentielles. Il regroupe un ensemble d'équipements critiques tels que les serveurs physiques, les machines virtuelles, les systèmes de stockage et les équipements réseau. Ces infrastructures jouent un rôle central dans le fonctionnement des systèmes d'information modernes, en garantissant la disponibilité, la performance et la sécurité des services informatiques.

Dans le cadre de ce projet, le Data Center étudié met à disposition diverses ressources informatiques utilisées par plusieurs catégories d'utilisateurs, notamment les ingénieurs, les enseignants, les doctorants ainsi que les responsables techniques et les administrateurs. Ces ressources sont exploitées pour des besoins pédagogiques, scientifiques et techniques tels que l'hébergement d'applications, la réalisation d'environnements de test, le calcul intensif ou encore le stockage de données.

La diversité des ressources et des profils utilisateurs rend leur gestion complexe. Il devient donc indispensable de disposer d'un système structuré permettant d'assurer une allocation équitable des ressources, une visibilité claire sur leur disponibilité et une exploitation optimale du parc informatique du Data Center.

1.2 Problématique de gestion des ressources

La gestion des ressources d'un Data Center est un défi, surtout lorsqu'elle repose sur des procédures manuelles ou des outils non centralisés. L'absence d'une plateforme unique peut entraîner plusieurs difficultés :

- Vision limitée en temps réel de la disponibilité et de l'état des ressources.
- Conflits de réservation dus au chevauchement des périodes d'utilisation.
- Manque de traçabilité de l'historique des réservations et de l'exploitation des ressources.
- Processus de validation lent et peu transparent.
- Absence de statistiques fiables sur le taux d'occupation global.

Ces contraintes peuvent conduire à une mauvaise allocation des ressources, à une surcharge de certains équipements, à une sous-utilisation d'autres, ainsi qu'à une perte de temps importante pour l'ensemble des acteurs impliqués. Il devient alors nécessaire de mettre en place une solution informatique capable de centraliser et d'automatiser la gestion des ressources.

1.3 Objectifs fonctionnels de l'application

L'objectif principal de ce projet est de concevoir et de développer une application web centralisée permettant d'assurer une gestion efficace, sécurisée et transparente des ressources informatiques d'un Data Center.

Les objectifs fonctionnels de l'application sont les suivants :

- Consulter les ressources disponibles avec leurs caractéristiques.
- Réserver des ressources en ligne selon des périodes définies.
- Vérifier la disponibilité et gérer les conflits de réservation.
- Suivre les demandes selon différents statuts (en attente, approuvée, refusée, active...).
- Gérer les utilisateurs avec rôles et permissions.
- Administrer les ressources, utilisateurs et périodes de maintenance.

- Afficher des tableaux de bord et statistiques d'utilisation.
- Assurer la sécurité des accès et la journalisation des actions.

Ces objectifs visent à améliorer l'organisation globale du Data Center tout en offrant une meilleure expérience aux utilisateurs et aux gestionnaires.

1.4 Solution proposée

Afin de répondre aux problématiques identifiées et d'atteindre les objectifs fixés, il est proposé de développer une application web moderne basée sur le framework Laravel et une base de données relationnelle MySQL.

La solution proposée repose sur les principes suivants :

- Une base de données pour gérer utilisateurs, rôles, ressources, réservations et historiques.
- Une interface web ergonomique, sécurisée et responsive, accessible depuis un navigateur.
- Une gestion précise des rôles et permissions pour contrôler l'accès selon le profil utilisateur.
- Un système de réservation intelligent détectant automatiquement les conflits et indisponibilités.
- L'exploitation des fonctionnalités natives de Laravel : authentification, middlewares, validation, migrations et ORM Eloquent.

Cette solution permet d'améliorer significativement la gestion du Data Center, d'optimiser l'utilisation des ressources disponibles et de renforcer la transparence et la traçabilité des opérations.

Chapitre 2 : Analyse fonctionnelle

Introduction du chapitre

Ce chapitre est consacré à l'analyse fonctionnelle de l'application de gestion des ressources du Data Center. Il vise à identifier les différents acteurs du système, à définir leurs rôles et permissions, et à décrire les principales interactions entre les utilisateurs et l'application à travers les cas d'utilisation. Cette analyse constitue une étape essentielle avant la phase de conception technique.

2.1 Acteurs du système

L'application de gestion des ressources du Data Center fait intervenir plusieurs catégories d'acteurs, chacune disposant de droits et de responsabilités spécifiques. Les principaux acteurs identifiés sont les suivants :

Invité

L'invité est un utilisateur externe qui accède à l'application sans authentification. Il dispose de droits très limités et ne peut effectuer que des actions en lecture seule. Son rôle principal consiste à consulter les informations publiques relatives au Data Center et aux ressources disponibles, ainsi qu'à déposer une demande d'ouverture de compte.

Utilisateur interne

L'utilisateur interne est un membre de l'organisation (ingénieur, enseignant ou doctorant) disposant d'un compte authentifié. Il utilise l'application pour effectuer des réservations de ressources informatiques nécessaires à ses activités professionnelles ou académiques. Il peut également consulter l'état de ses demandes et l'historique de ses réservations.

Responsable technique

Le responsable technique est chargé de la supervision et de la gestion technique d'un ensemble de ressources du Data Center. Il joue un rôle clé dans la validation des demandes de réservation et dans le maintien du bon fonctionnement des équipements dont il a la charge.

Administrateur

L'administrateur est l'acteur disposant des priviléges les plus élevés. Il est responsable de la gestion globale du système, de la sécurité, des utilisateurs et du catalogue des ressources. Il assure également le suivi global de l'activité du Data Center.

2.2 Rôles et permissions

Le système repose sur une gestion des rôles et des permissions afin de garantir un accès sécurisé et contrôlé aux différentes fonctionnalités de l'application. Chaque acteur dispose de droits spécifiques en fonction de son rôle.

- **Invité :**

- Consulter les informations publiques.
- Visualiser les ressources disponibles en lecture seule.
- Soumettre une demande d'ouverture de compte.

- **Utilisateur interne :**

- Accéder à un espace personnel.
- Consulter les ressources avec filtres.
- Soumettre des demandes de réservation.
- Suivre l'état de ses réservations.
- Consulter l'historique de ses demandes.
- Recevoir des notifications internes.
- Signaler un problème ou un incident.

- **Responsable technique :**

- Gérer les ressources sous sa responsabilité.
- Consulter les demandes de réservation liées à ses ressources.
- Approuver, planifier ou refuser les demandes avec justification.
- Mettre une ressource en maintenance ou la désactiver.
- Modérer les échanges liés à ses ressources.

- **Administrateur :**

- Gérer les utilisateurs, les rôles et les permissions.
- Gérer le catalogue global des ressources.
- Activer ou désactiver des comptes utilisateurs.
- Planifier les périodes de maintenance.
- Consulter les statistiques globales du Data Center.
- Superviser la sécurité et la traçabilité du système.

Cette répartition des rôles permet d'assurer une séparation claire des responsabilités et de renforcer la sécurité globale de l'application.

2.3 Cas d'utilisation

Les cas d'utilisation décrivent les interactions entre les acteurs et le système. Ils permettent de représenter les fonctionnalités offertes par l'application du point de vue de l'utilisateur.

2.3.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation met en évidence les principales fonctionnalités accessibles à chaque acteur du système. Il illustre les relations entre les utilisateurs (invité, utilisateur interne, responsable technique, administrateur) et les différents modules de l'application tels que la consultation des ressources, la gestion des réservations, l'administration et la supervision.

2.4 Description des cas d'utilisation

Cas d'utilisation 1 : Consulter les informations publiques

- **Acteur principal :** Invité
- **Précondition :** L'utilisateur accède à l'application sans authentification.

Scénario principal :

1. L'invité accède à la page d'accueil de l'application.
2. Le système affiche les informations publiques disponibles.
3. L'invité consulte les données sans possibilité de modification.

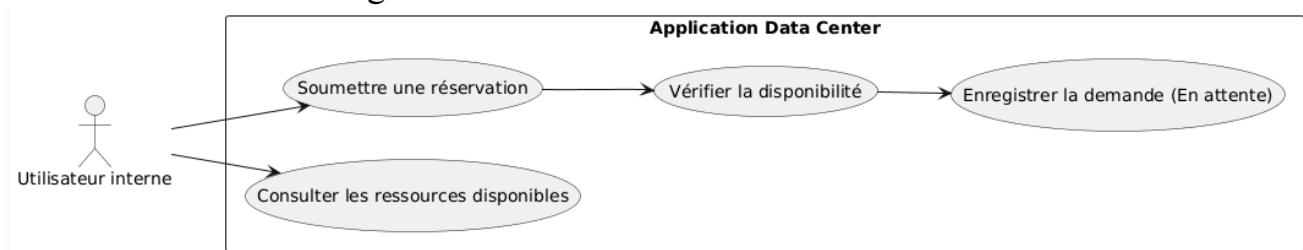


Cas d'utilisation 2 : Effectuer une demande de réservation

- **Acteur principal :** Utilisateur interne
- **Précondition :** L'utilisateur est authentifié.

Scénario principal :

1. L'utilisateur accède à son espace personnel.
2. Il consulte la liste des ressources disponibles.
3. Il sélectionne une ressource et définit la période de réservation.
4. Il saisit la justification de la demande.
5. Le système vérifie la disponibilité de la ressource.
6. La demande est enregistrée avec le statut *En attente*.

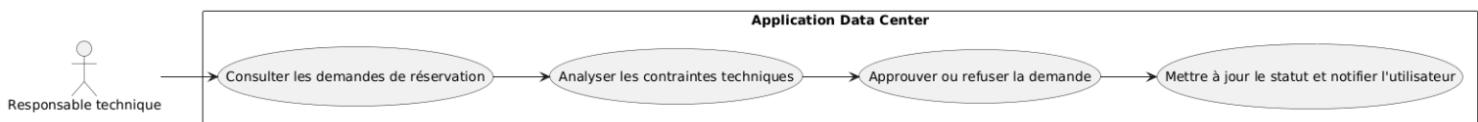


Cas d'utilisation 3 : Traiter une demande de réservation

- **Acteur principal :** Responsable technique
- **Précondition :** Le responsable technique est authentifié et possède les droits nécessaires.

Scénario principal :

1. Le responsable technique consulte les demandes associées à ses ressources.
2. Il analyse la demande et les contraintes techniques.
3. Il approuve ou refuse la demande en fournissant une justification.
4. Le système met à jour le statut de la réservation et notifie l'utilisateur.



Cas d'utilisation 4 : Gérer les utilisateurs

- **Acteur principal :** Administrateur
- **Précondition :** L'administrateur est authentifié.

Scénario principal :

1. L'administrateur accède au module de gestion des utilisateurs.
2. Il crée, modifie ou supprime un compte utilisateur.
3. Il attribue les rôles et permissions appropriés.
4. Le système met à jour la base de données et confirme l'opération.

Conclusion du chapitre

L'analyse fonctionnelle présentée dans ce chapitre a permis d'identifier clairement les acteurs du système, leurs rôles et les principales interactions avec l'application. Cette étape constitue une base solide pour la phase de conception du système, qui sera abordée dans le chapitre suivant à travers l'architecture générale, les modèles de données et les diagrammes UML.

Chapitre 3 : Conception du système

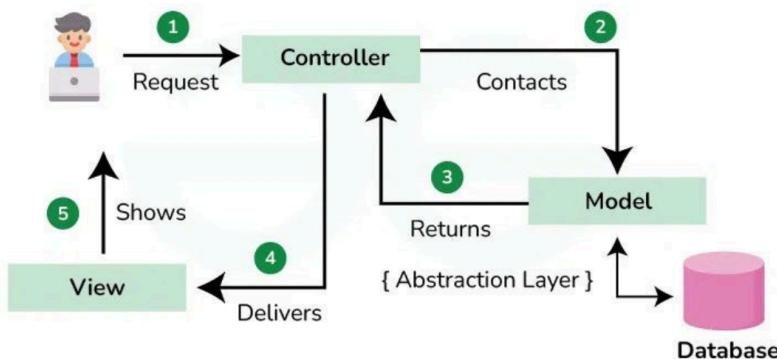
Introduction

Ce chapitre décrit la **conception technique et structurelle** de l'application web de gestion des ressources du Data Center. Il présente l'architecture adoptée, le modèle de données, ainsi que l'organisation réelle du projet Laravel, y compris les **dossiers des migrations et des modèles**, et les relations entre les entités principales. Cette conception constitue la base pour la phase de réalisation décrite dans le chapitre suivant.

3.1 Architecture générale de l'application

L'application repose sur l'architecture **MVC (Model – View – Controller)** fournie par Laravel :

- **Model** : couche d'accès aux données, via l'ORM Eloquent, représentant les entités telles que User, Resource, Reservation, Notification, etc.
- **View** : interface utilisateur (vues Blade), responsive et ergonomique, permettant l'interaction avec les différentes fonctionnalités selon le rôle de l'utilisateur.
- **Controller** : gestion de la logique métier et coordination entre modèles et vues (ex. ReservationController, AdminController).



Des **middlewares** sont utilisés pour contrôler l'accès aux routes selon les rôles (Invité, Utilisateur interne, Responsable technique, Administrateur) et renforcer la sécurité de l'application.

3.2 Modèle de données

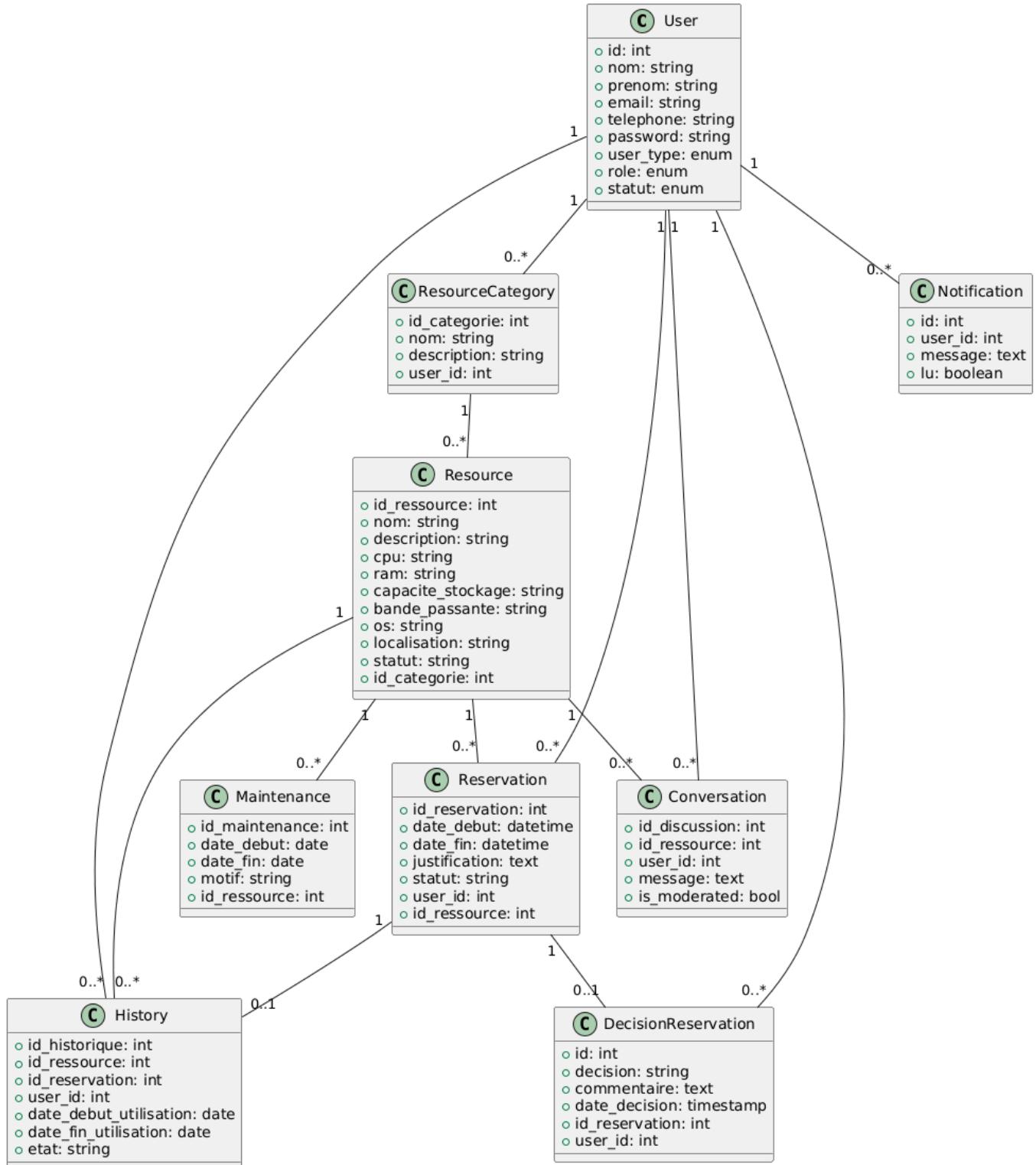
3.2.1 Entités principales et relations

Le système manipule les entités suivantes :

Entité	Description	Relations
User	Représente les utilisateurs du système (invité, utilisateur interne, responsable, administrateur)	Possède plusieurs réservations, notifications, conversations et historyLogs
Role	Définit les rôles et permissions	Un rôle peut être associé à plusieurs utilisateurs
Resource	Représente une ressource du Data Center	Appartient à une catégorie, possède plusieurs réservations et maintenances
ResourceCategory	Classe les ressources par type (serveur, VM, stockage, réseau)	Contient plusieurs ressources
Reservation	Représente une demande de réservation	Appartient à un utilisateur et une ressource, possède des conversations
Conversation	Messages associés à une réservation	Appartient à un utilisateur et une réservation
Notification	Messages internes envoyés aux utilisateurs	Appartient à un utilisateur
MaintenanceSchedule	Périodes d'indisponibilité planifiée	Appartient à une ressource
HistoryLog	Journalisation des actions importantes	Appartient à un utilisateur

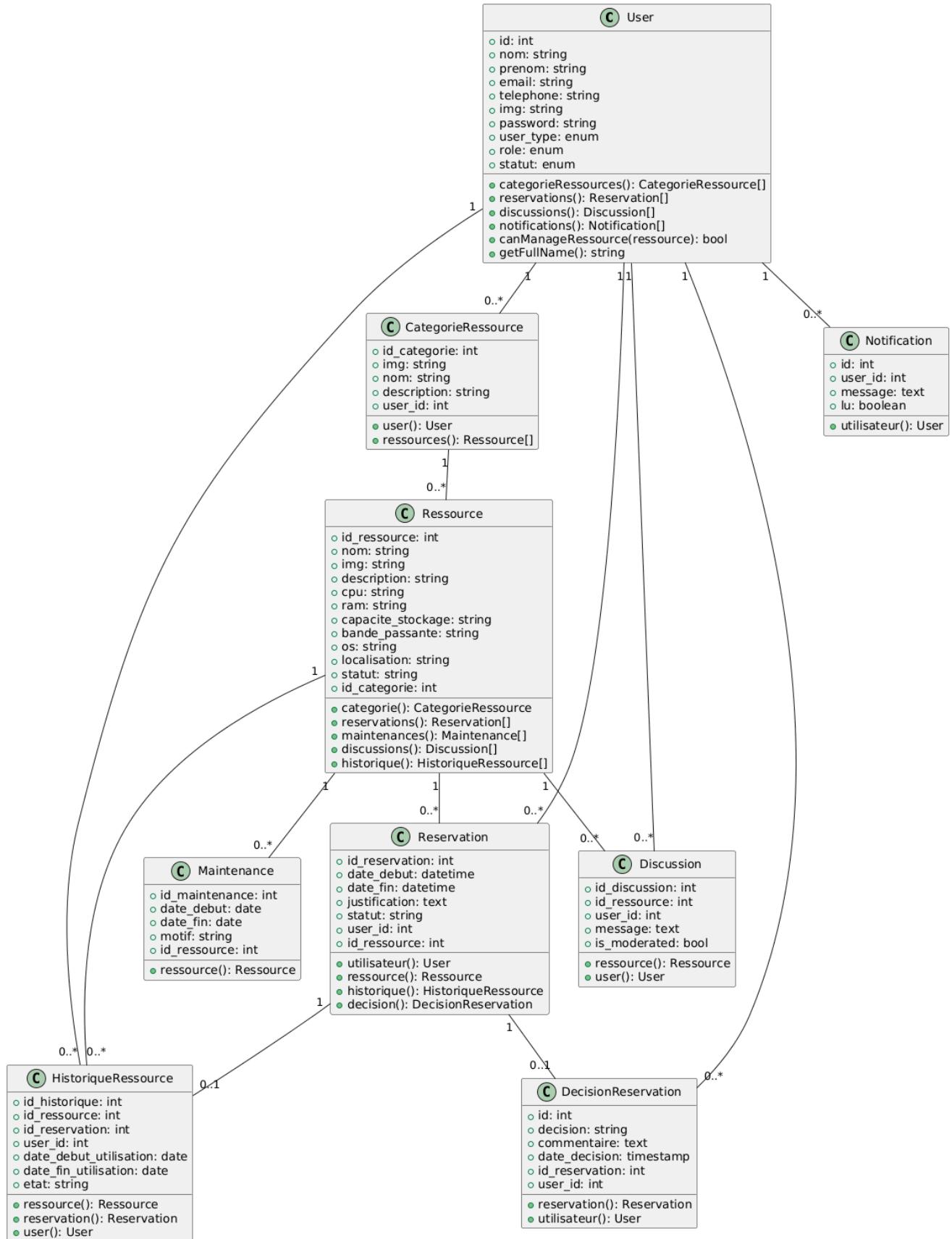
3.2.2 Modèle logique et relationnel

Chaque entité du MCD est traduite en une table MySQL avec ses relations :



3.3 Organisation des modèles

Le dossier `app/Models` contient les modèles Eloquent représentant chaque entité.



Représente les entités principales (`User`, `Role`, `Resource`, `Reservation`, `Notification`, etc.) et leurs relations (1:N ou N:1).

Chaque classe correspond directement à un modèle Eloquent

Chaque modèle définit :

- les **attributs remplissables** (`$fillable`),
- les **relations avec d'autres modèles** (`belongsTo`, `hasMany`),
- les **castings** pour les champs spéciaux (`boolean`, `datetime`, `array`).

Exemples :

- `User` → possède plusieurs `reservations`, `notifications`, `resourcesGerees`.
- `Resource` → appartient à une `ResourceCategory` et un `User` responsable, possède plusieurs `reservations` et `maintenanceSchedules`.
- `Reservation` → appartient à un `User` et une `Resource`, possède plusieurs conversations

Conclusion du chapitre

Ce chapitre a présenté la **conception technique complète** de l'application : architecture MVC, modèle de données, relations entre entités, organisation réelle des **migrations et modèles Laravel**. Cette conception fournit une base solide pour le développement effectif du système, détaillé dans le **Chapitre 4 : Réalisation de l'application**.

Chapitre 4 : Réalisation de l'application

Introduction

Ce chapitre décrit la phase de réalisation technique de l'application web de gestion des ressources du Data Center. Il présente l'environnement de développement, les technologies utilisées, l'organisation du projet Laravel ainsi que l'implémentation des principales fonctionnalités mises en place pour assurer la gestion efficace des ressources.

4.1 Environnement de développement

Pour garantir un développement stable et efficace, l'environnement de travail suivant a été mis en place :

- **Système d'exploitation :**
 - Windows
- **Serveur local :**
 - XAMPP (Apache + MySQL)
- **Éditeur de code :**
 - Visual Studio Code, avec extensions Laravel Blade et PHP Intelephense
- **Navigateur web :**
 - Google Chrome (pour le débogage et l'inspection CSS/JS)
- **Gestion de version :**
 - Git, pour le suivi des modifications et la sauvegarde du code source

Cet environnement a permis de développer, tester et valider les différentes fonctionnalités de l'application de manière efficace et sécurisée.

4.2 Technologies utilisées

L'application repose sur un ensemble de technologies modernes adaptées au développement web dynamique :

- **Laravel** : Laravel est un framework PHP qui facilite le développement d'applications web en suivant le modèle **MVC (Model-View-Controller)**. Il permet de séparer clairement la logique métier, l'interface utilisateur et l'accès aux données, ce qui rend le code plus organisé et facile à maintenir. Laravel intègre aussi des fonctionnalités de sécurité prêtes à l'emploi, comme la protection contre les attaques CSRF, la validation des formulaires et un système d'auth complet.
- **PHP** : PHP est un langage de programmation côté serveur utilisé pour créer la logique métier des applications web. Il permet de traiter les données envoyées par les utilisateurs, de gérer les interactions avec la base de données.
- **MySQL** : MySQL est un système de gestion de bases de données relationnelles. Il est utilisé pour stocker et organiser toutes les informations d'une application, telles que les utilisateurs, les rôles, les ressources disponibles, les réservations, les notifications et les journaux d'historique. MySQL permet d'effectuer des opérations de lecture, d'écriture, de modification et de suppression de données (CRUD) de manière rapide et fiable.
- **CSS personnalisé** : Le CSS (Cascading Style Sheets) est utilisé pour styliser et mettre en forme les pages web. Dans ce projet, le CSS est écrit de manière personnalisée, sans frameworks externes comme Bootstrap, pour créer une interface ergonomique et responsive.
- **JavaScript personnalisé** : JavaScript est un langage côté client qui rend les pages web interactives. Ici, il est utilisé pour effectuer la **validation des formulaires** avant l'envoi au serveur, pour afficher dynamiquement des tableaux ou des notifications, et pour améliorer l'expérience utilisateur sans recharger la page.

4.3 Organisation du projet Laravel

Le projet respecte l’arborescence standard de Laravel, facilitant la maintenance et l’évolution du système :

- **app/Models** : contient les modèles Eloquent représentant les entités principales :
 - CategorieRessource.php
 - Discussion.php
 - HistoriqueRessource.php
 - Maintenance.php
 - Notification.php
 - Reservation.php
 - Ressource.php
 - User.php
 - etc..
- **app/Http/Controllers** : regroupe les contrôleurs responsables de la logique métier :
 - CategorieRessourceController.php
 - Controller.php
 - DashboardController.php
 - DiscussionController.php
 - NotificationController.php
 - ReservationController.php
 - etc..
- **database/migrations** : contient les fichiers de migration pour créer et modifier les tables de la base de données
- **routes/web.php** : définition des routes accessibles via le navigateur
- **resources/views** : contient les vues Blade représentant les interfaces utilisateur
- **public/** : regroupe les fichiers CSS, JavaScript et les ressources statiques

Des **middlewares** ont été mis en place pour contrôler l'accès aux fonctionnalités selon le rôle : Invité, Utilisateur interne, Responsable technique, Administrateur.

4.4 Implémentation des fonctionnalités principales

4.4.1 Gestion des utilisateurs

- **Authentification** : Les utilisateurs doivent se connecter pour accéder aux fonctionnalités. Les invités peuvent créer une demande de compte, activée uniquement par l'administrateur.
- **Gestion des rôles** : Chaque utilisateur a un rôle (`Guest`, `Utilisateur`, `Responsable`, `Admin`) défini dans la table `roles`.
- **Middlewares** : Limitent l'accès aux routes selon le rôle.
- **Relations** : `User` possède plusieurs `reservations`, `resourcesGerees`, `notifications`..

4.4.2 Gestion des ressources

- **CRUD complet** : Les responsables techniques peuvent ajouter, modifier, mettre en maintenance ou désactiver des ressources.
- **Fiche technique** : Chaque ressource contient des informations comme CPU, RAM, stockage, OS, emplacement et statut.
- **Disponibilité dynamique** : Le champ `statut` et la fonction `isDisponible()` déterminent si une ressource est libre pour réservation.
- **Relations** : Chaque ressource appartient à une catégorie (`ResourceCategory`) et à un responsable (`User`).
- **Maintenance** : Périodes d'indisponibilité gérées via `MaintenanceSchedule`.

4.4.3 Système de réservation

- Les utilisateurs internes peuvent soumettre des demandes de réservation avec période et justification.
 - Vérification automatique de la disponibilité pour éviter les conflits (`overlapping`).
 - Cycle de vie d'une réservation :
1. **Demande** → 2. **En attente** → 3. **Validation/Refus** → 4. **Activation** → 5.

Terminaison

4.4.4 Notifications internes

- Les utilisateurs reçoivent automatiquement des notifications lors :
 - de la validation ou du refus d'une réservation,
 - des changements de statut,
 - de l'approche de la fin de réservation.
- Les notifications sont stockées dans la table `notifications` et liées à un utilisateur.

4.4.5 Statistiques et tableaux de bord

- Tableau de bord pour les utilisateurs autorisés affichant des statistiques graphiques :
 - taux d'occupation des ressources,
 - nombre de réservations par période,
 - ressources les plus sollicitées.
- Ces données aident à analyser l'utilisation du Data Center et faciliter la prise de décision.

4.4.6 Journalisation et traçabilité

- Toutes les actions importantes (connexion, création ou modification de ressources, validation de réservation) sont enregistrées dans `HistoryLog`.
- Chaque entrée contient l'utilisateur, la table concernée, l'action, les anciennes et nouvelles valeurs.
- Cette journalisation améliore la sécurité et la fiabilité du système.

4.4.7 Sécurité et protection des accès

- **Authentification** Laravel pour sécuriser l'accès.
- **Gestion des rôles et permissions** pour limiter les actions selon le profil.
- **Middlewares** pour protéger les routes et refuser l'accès aux utilisateurs non autorisés.
- **Traçabilité** via `HistoryLog` pour suivre toutes les modifications

4.5 Organisation des routes et contrôleurs

- **Routes principales (web.php) :**
 - /login, /register → Authentification
 - /resources → Consultation et gestion des ressources
 - /reservations → Gestion des réservations
 - /notifications → Liste des notifications
 - /admin/* → Administration et gestion des utilisateurs/ressources
- **Contrôleurs principaux :**
 - UserController → Gestion des utilisateurs
 - ResourceController → Gestion des ressources
 - ReservationController → Gestion des réservations et notifications
 - AdminController → Administration générale et statistiques
 - NotificationController → Gestion et affichage des notifications

Conclusion du chapitre

La réalisation de l'application a permis de mettre en place un **système complet et fonctionnel** pour gérer, réserver et suivre les ressources informatiques du Data Center.

Le projet respecte l'architecture MVC de Laravel, avec une base de données relationnelle MySQL, une interface web responsive et sécurisée, un système de notifications, un journal de traçabilité et des tableaux de bord statistiques.

Cette phase de réalisation montre la mise en pratique des concepts de **modélisation, développement web et sécurité** pour un projet concret.

Chapitre 5 : Sécurité et gestion des accès

Introduction

La sécurité est un aspect essentiel de toute application web, surtout pour un système de gestion des ressources d'un Data Center, où des informations sensibles et des ressources critiques sont manipulées. Ce chapitre présente les mécanismes mis en place pour sécuriser l'accès à l'application, contrôler les permissions des utilisateurs et assurer la traçabilité des actions.

5.1 Authentification et autorisation

Authentification

L'authentification permet de vérifier l'identité des utilisateurs avant de leur donner accès au système.

- Chaque utilisateur doit se connecter avec un **email** et un **mot de passe** pour accéder à ses fonctionnalités.
- Les invités peuvent créer une demande de compte, mais **l'activation de ce compte est réalisée uniquement par l'administrateur**.
- Laravel gère l'authentification via son système natif (**Auth**), incluant :
 - Protection contre les attaques CSRF
 - Hachage sécurisé des mots de passe

Autorisation

Une fois l'utilisateur authentifié, l'autorisation détermine **ce qu'il peut faire** :

- Chaque action est limitée selon le **rôle de l'utilisateur** (Guest, Utilisateur, Responsable, Administrateur).
- L'autorisation empêche un utilisateur d'accéder à des fonctionnalités qui ne correspondent pas à son rôle.

5.2 Gestion des rôles et permissions

L'application repose sur un **système de rôles et permissions** pour organiser l'accès aux fonctionnalités :

- **Rôles définis :**

1. **Guest (Invité)** : accès en lecture seule aux informations publiques, demande de création de compte
2. **Utilisateur interne** : consultation et réservation des ressources, suivi de ses demandes, signalement de problèmes
3. **Responsable technique** : gestion des ressources qu'il supervise, validation ou refus des demandes, modération des conversations
4. **Administrateur** : gestion globale des utilisateurs, des ressources, des permissions et des statistiques

- Chaque rôle dispose d'un ensemble spécifique de **permissions**, contrôlées via le champ `role` dans la table `users` et appliquées grâce aux **middlewares Laravel**.
- Les middlewares vérifient le rôle avant d'accéder à une route ou d'exécuter une action, garantissant ainsi **un contrôle fin des accès**.

5.3 Protection des routes (Middleware)

Les **middlewares** jouent un rôle clé dans la sécurité :

- Ils contrôlent l'accès aux routes et fonctionnalités selon le rôle de l'utilisateur.
- Exemples d'application :
 - `/admin/*` → accessible uniquement par l'administrateur
 - `/resources/manage` → accessible par le responsable technique
 - `/reservations/create` → accessible par les utilisateurs internes
- Lorsqu'un utilisateur tente d'accéder à une route sans permission, **l'accès est refusé** et il est redirigé vers une page sécurisée.

5.4 Journalisation et traçabilité

La traçabilité permet de conserver un **historique détaillé des actions importantes** réalisées dans le système.

- Les actions enregistrées incluent :
 - Création, modification ou suppression de ressources
 - Validation ou refus des réservations
 - Modification des comptes utilisateurs et rôles
- Les journaux sont stockés dans la table `history_logs` avec les informations suivantes
 - **Utilisateur** ayant effectué l'action
 - **Description détaillée**
- Cette journalisation permet :
 - De **déetecter les anomalies** et comportements suspects
 - De faciliter la **maintenance et le suivi**
 - D'assurer la **transparence** et la sécurité du système

5.5 Sécurité supplémentaire

5.5.1 Hachage des mots de passe

Tous les mots de passe sont hachés automatiquement par Laravel avant d'être enregistrés dans la base de données, empêchant toute lecture directe en cas de fuite de données.

5.5.2 Protection des données

- Les données sensibles (comme les informations personnelles et historiques) sont **accessibles uniquement aux utilisateurs autorisés**.
- Les relations Eloquent (`User`, `Reservation`, `Resource`, etc.) respectent les règles de visibilité pour éviter les fuites d'informations.

5.5.3 Gestion des sessions

- Les sessions utilisateur sont sécurisées par Laravel avec expiration automatique et invalidation lors de la déconnexion.
- Une seule session active par utilisateur peut être configurée pour éviter l'usurpation de compte.

Conclusion du chapitre

La sécurité et la gestion des accès sont des aspects fondamentaux pour la fiabilité du système de gestion des ressources.

Grâce aux mécanismes suivants :

- Authentification et autorisation robustes
- Gestion des rôles et permissions via middlewares
- Journalisation complète des actions
- Protection des données et hachage des mots de passe

Chapitre 6 : Interfaces de l'application

Introduction

Ce chapitre présente les différentes interfaces de l'application web développée pour la gestion des ressources du Data Center. Chaque interface est conçue pour répondre aux besoins spécifiques de chaque profil utilisateur : Invité, Utilisateur interne, Responsable technique et Administrateur. L'objectif est d'offrir une expérience utilisateur claire, ergonomique et fonctionnelle, tout en respectant les règles de sécurité et les permissions associées à chaque rôle.

6.1 Interface Invité

L'interface **Invité** est destinée aux utilisateurs externes qui n'ont pas encore de compte ou qui souhaitent consulter les informations publiques.

Fonctionnalités principales :

- **Consultation des ressources disponibles** en lecture seule, avec filtres de recherche par type ou catégorie de ressources.
- **Demande de création de compte** pour accéder aux fonctionnalités réservées aux utilisateurs internes.
- Accès aux **règles d'utilisation** des ressources et aux informations générales sur le Data Center.

Caractéristiques de l'interface :

- Design simple et clair, centré sur la lisibilité des informations.
- Navigation intuitive permettant de trouver rapidement les ressources ou les informations sur la création de compte.



figure : welcome page

The screenshot shows the 'Notre Mission' section of the DataCenterPro website. The navigation bar at the top includes 'Règles' which is underlined. Below the navigation, the title 'Notre Mission' is centered above four cards. Each card has a circular icon and a title: 'Optimiser' (rocket icon), 'Sécuriser' (shield icon), 'Superviser' (monitor icon), and 'Simplifier' (people icon). Each card also contains a brief description of its purpose.

figure : Regle page

The screenshot shows the 'Connexion au Dashboard' (Login) page. The left side features a large blue banner with the DataCenterPro logo and the text 'Rejoignez la plateforme de gestion du Data Center'. It also includes a registration form for new users. The right side contains a login form with fields for 'Adresse Email' (Email address) and 'Mot de passe' (Password), both marked with a red asterisk indicating they are required. There are also checkboxes for 'Se souvenir de moi' (Remember me) and 'Mot de passe oublié ?' (Forgot password?). At the bottom, there's a 'Se connecter' button and a link to 'Créer un compte' (Create account).

figure : page de connexion

6.2 Interface Utilisateur interne

L'interface **Utilisateur interne** (ingénieur, enseignant, doctorant) est conçue pour faciliter la **réservation et le suivi des ressources**.

Fonctionnalités principales :

The screenshot shows the DataCenterPro dashboard. On the left, a table titled "Mes Réservations Récentes" lists recent reservations with columns for RESOURCE, STATUS, and DATE. On the right, a section titled "Mon Activité" displays four summary boxes: "Total Réservations" (10), "En attente" (4), "Approuvées" (4), and "Actives" (0). Below this are "Actions Rapides" buttons for "Nouvelle Réservation" and "Voir mes réservations".

Mes Réservations Récentes		
RESSOURCE	STATUT	DATE
VLAN Réseau virtuel	En attente	12/02/2026
Stockage NAS Stockage	Approuvée	05/03/2026
Serveur de calcul Serveurs physiques	Refusée	16/02/2026
Pare-feu Équipements réseau	Approuvée	07/02/2026
VLAN Réseau virtuel	Refusée	12/02/2026

- **Espace personnel** avec tableau de bord :

- Vue synthétique des réservations en cours et historiques.
- Notifications sur l'état des demandes.

The screenshot shows the "Ressources dans cette catégorie" section. It displays three resource cards: "Baie de stockage SAN" (available), "Stockage NAS" (available), and "Stockage objet" (available). Each card includes a thumbnail image, the resource name, a brief description, and filters for CPU and RAM.

Ressources dans cette catégorie	
 Baie de stockage SAN Stockage centralisé performant pour les serveurs.	 Stockage NAS Stockage réseau accessible par plusieurs utilisateurs simultanément.
 Stockage objet Stockage flexible pour les fichiers volumineux et non structurés.	

- **Consultation des ressources disponibles** avec filtres avancés par catégorie, état, disponibilité, ou caractéristiques techniques.

Effectuer une demande de réservation

Ressource souhaitée
Sélectionnez une ressource...

Date de début
jj/mm/aaaa

Date de fin
jj/mm/aaaa

Justification de la demande
Veuillez expliquer pourquoi vous avez besoin de cette ressource...

- **Soumission de demandes de réservation :**

- Sélection de la ressource, période souhaitée et justification.
- Vérification automatique de la disponibilité avant enregistrement.

Ressource	Début	Fin	Statut	Date de demande	Actions
Stockage NAS	05/03/2026	05/03/2026	En attente	29/01/2026 07:37	
Volume de sauvegarde	04/03/2026	04/03/2026	En attente	29/01/2026 07:37	
Pare-feu	03/03/2026	03/03/2026	En attente	29/01/2026 07:37	
VM Linux	23/02/2026	23/02/2026	Approuvée	29/01/2026 07:37	
Réseau privé virtuel (VPC)	19/02/2026	19/02/2026	Approuvée	29/01/2026 07:37	
Serveur de calcul	16/02/2026	16/02/2026	En attente	29/01/2026 07:37	
Certificat SSL	14/02/2026	14/02/2026	En attente	29/01/2026 07:37	
VLAN	12/02/2026	12/02/2026	En attente	29/01/2026 07:37	
VLAN	12/02/2026	12/02/2026	En attente	29/01/2026 07:37	
Pare-feu	07/02/2026	07/02/2026	En attente	29/01/2026 07:37	

- **Suivi des demandes :**

- États possibles : En attente, Approuvée, Refusée, Active, Terminée.

Mon Historique

- Réservation #1 - VLAN**
Statut: En attente • Crée le 29/01/2026 07:31
- Réservation #2 - Stockage NAS**
Statut: Approuvée • Crée le 29/01/2026 07:31
- Réservation #3 - Serveur de calcul**
Statut: Refusée • Crée le 29/01/2026 07:31
- Réservation #4 - Pare-feu**
Statut: Approuvée • Crée le 29/01/2026 07:31
- Réservation #5 - VLAN**
Statut: Refusée • Crée le 29/01/2026 07:31

- **Historique complet :**

- Consultation filtrée par date, ressource ou statut.

Caractéristiques de l'interface :

- Tableau de bord interactif avec notifications dynamiques.
- Affichage des informations des ressources de manière claire et structurée.
- Formulaires de réservation simples et sécurisés.

6.3 Interface Responsable technique

L'interface **Responsable technique** est destinée aux gestionnaires de ressources. Elle permet de superviser et de contrôler les ressources sous leur responsabilité.

Fonctionnalités principales :

Réservations de ma Catégorie				Voir tout
RESSOURCE	UTILISATEUR	STATUT	DATE	
Équipement	Utilisateur			0

Mon Activité	
0	0

- **Gestion des ressources supervisées :**
 - Ajouter, modifier ou désactiver une ressource.
 - Mettre en maintenance une ressource temporairement indisponible.

The screenshot shows the DataCenterPro software interface. At the top, there is a navigation bar with icons for DataCenterPro, Categories, Resources, Dashboard, Reserv (selected), and Notifications (with 2 notifications). On the right, there is a user profile for 'filan Respo' (Responsible) with a dropdown arrow.

Informations Reservation

Statut: En attente

Date Début: 07/03/2026
Date Fin: 07/03/2026
Date Demande: 29/01/2026 07:50

Justification: Test reservation 1

Ressource Concernee

Nom: Équipement VPN
Catégorie: Équipements réseau
Description: Permet des connexions sécurisées à distance au réseau interne.

Demandeur

Nom: Utilisateur Standard
Email: user@test.com
Type: enseignant

Traitement de la demande

Commentaire (Requis pour refus)

Action Buttons: ✓ Approver (green button) and ✗ Refuser (red button)

- **Validation des réservations :**

- Consulter toutes les demandes liées aux ressources qu'il supervise.
- Approuver, planifier ou refuser les réservations avec justification.

The screenshot shows a list of messages or comments from users. The first message is from 'Standard Utilisateur' (4 minutes ago) with the text: 'Molestiae aut saepe pariatur molestias dignissimos rerum perspiciatis a quo reprehenderit.' The second message is from 'Jalal Ahmed' (4 minutes ago) with the text: 'Eum rerum nihil quia dolor odio beatae aut esse sit odit eligendi rerum.'

- **Modération des conversations liées aux ressources :**

- Supprimer les messages inappropriés ou envoyer des alertes.

- **Tableau de bord des ressources :**

- Vue d'ensemble des réservations, disponibilité et maintenance.

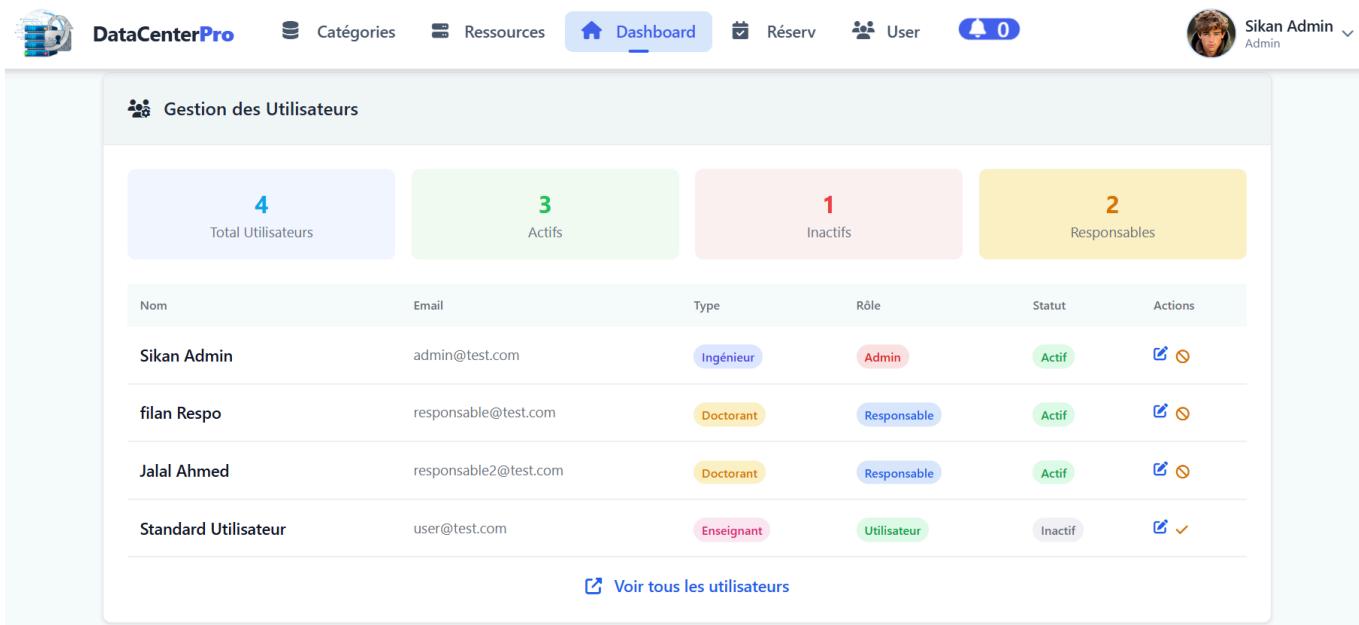
Caractéristiques de l'interface :

- Accès sécurisé via authentification et rôle.
- Affichage clair des actions nécessitant une intervention (réservation à valider, alertes).
- Navigation intuitive entre les différentes ressources et demandes.

6.4 Interface Administrateur

L'interface **Administrateur** offre un contrôle complet sur l'application et le Data Center. Elle est destinée aux utilisateurs ayant les droits les plus élevés.

Fonctionnalités principales :



The screenshot shows the DataCenterPro Admin interface. At the top, there is a navigation bar with links for 'Catégories', 'Ressources', 'Dashboard' (which is highlighted in blue), 'Réserv.', 'User', and a notification bell with '0' notifications. On the far right, there is a user profile for 'Sikan Admin' (Admin) with a dropdown arrow.

The main content area is titled 'Gestion des Utilisateurs'. It displays four summary statistics in colored boxes: '4 Total Utilisateurs' (blue), '3 Actifs' (green), '1 Inactifs' (pink), and '2 Responsables' (yellow). Below this is a table listing five users:

Nom	Email	Type	Rôle	Statut	Actions
Sikan Admin	admin@test.com	Ingénieur	Admin	Actif	<input checked="" type="checkbox"/> <input type="checkbox"/>
filan Respo	responsable@test.com	Doctorant	Responsable	Actif	<input checked="" type="checkbox"/> <input type="checkbox"/>
Jalal Ahmed	responsable2@test.com	Doctorant	Responsable	Actif	<input checked="" type="checkbox"/> <input type="checkbox"/>
Standard Utilisateur	user@test.com	Enseignant	Utilisateur	Inactif	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>

At the bottom of the table, there is a link 'Voir tous les utilisateurs' with a magnifying glass icon.

- **Gestion des utilisateurs :**

- Création, modification et suppression de comptes.
- Attribution de rôles et gestion des permissions.

- **Gestion complète des ressources :**

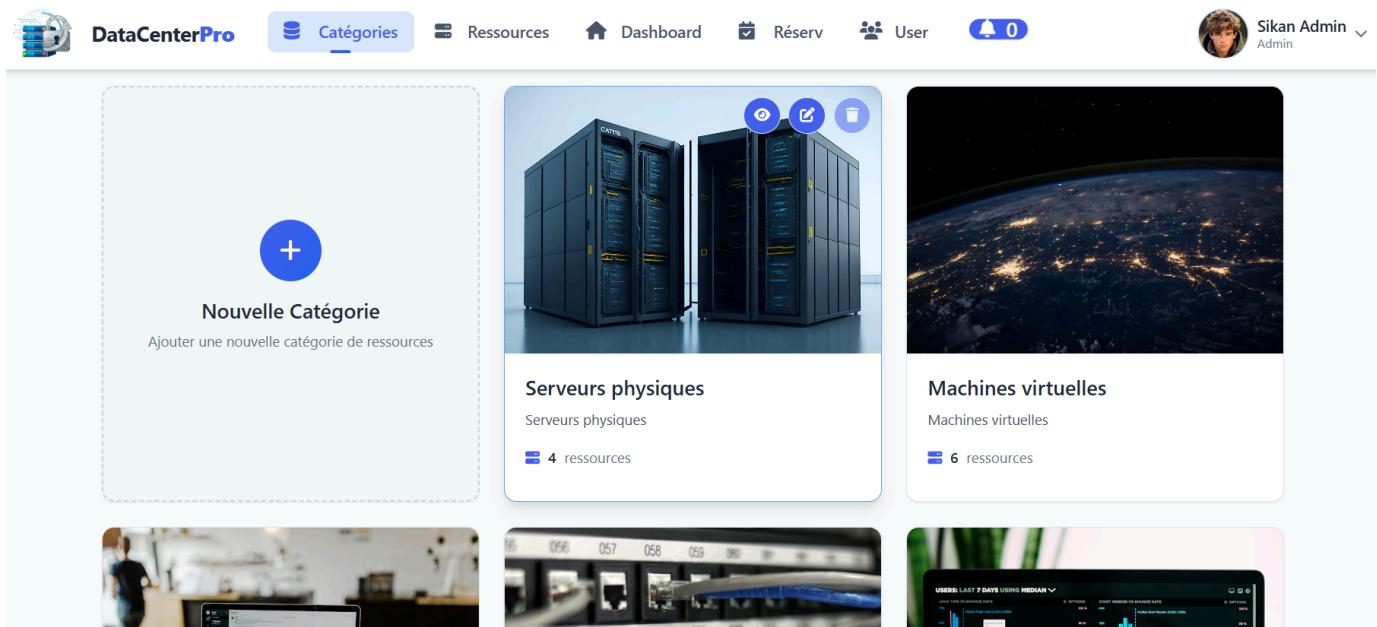
- Ajout et modification des catégories de ressources et de leurs caractéristiques.
- Activation ou désactivation des ressources.

- **Consultation des statistiques globales :**

- Taux d'occupation, nombre de réservations par période, ressources les plus utilisées.

Caractéristiques de l'interface :

- Tableau de bord complet avec statistiques graphiques.
- Navigation rapide entre la gestion des utilisateurs, des ressources et des réservations.
- Outils de recherche et de filtrage pour faciliter la supervision globale.



6.5 Ergonomie et design

L'ensemble des interfaces a été conçu pour être **ergonomique, clair et responsive**, sans l'utilisation de frameworks CSS externes :

- **Flexbox et Grid** pour structurer les pages et les tableaux.
- **CSS personnalisé** pour l'esthétique et la cohérence visuelle.
- **JavaScript** pour l'interactivité :
 - Validation des formulaires côté client
 - Affichage dynamique des notifications et des tableaux de bord
 - Filtrage et tri des listes de ressources ou de réservations

Ces choix permettent une **navigation fluide** et une **expérience utilisateur homogène** sur tous les profils.

Conclusion du chapitre

Les interfaces développées répondent aux besoins spécifiques de chaque rôle utilisateur, assurant :

- Une consultation rapide et claire des informations pour les invités et utilisateurs internes.
- Une gestion efficace et centralisée des ressources pour les responsables techniques et administrateurs.
- Une expérience utilisateur intuitive et sécurisée, respectant les rôles et permissions.

L'ergonomie et l'interactivité des interfaces contribuent ainsi à **l'efficacité de la gestion du Data Center** et à la satisfaction des utilisateurs.

Conclusion générale

Le projet avait pour objectif la conception et le développement d'une **application web de gestion des ressources d'un Data Center**, permettant de centraliser, sécuriser et optimiser l'utilisation des serveurs, machines virtuelles, équipements réseau et systèmes de stockage.

À travers l'utilisation du **framework Laravel**, associé à une **base de données MySQL**, ce projet a permis de créer une solution complète répondant aux besoins de différents profils d'utilisateurs : invités, utilisateurs internes, responsables techniques et administrateurs.

Bilan du projet

La plateforme développée offre :

- **Gestion sécurisée des utilisateurs** selon leur rôle et permissions, avec authentification et journalisation des actions.
- **Gestion complète des ressources du Data Center**, incluant ajout, modification, maintenance et désactivation des équipements.
- **Système de réservation en ligne** permettant de vérifier automatiquement la disponibilité des ressources et d'éviter les conflits.
- **Suivi détaillé des demandes de réservation** avec différents statuts : en attente, approuvée, refusée, active, terminée.
- **Notifications internes et tableaux de bord statistiques** pour assurer la traçabilité et faciliter la prise de décision.

Le développement a respecté les contraintes du projet, notamment l'absence de frameworks CSS/Javascript externes, et a permis de mettre en pratique des connaissances en **architecture MVC, gestion des bases de données relationnelles, sécurité et développement web interactif**.

Objectifs atteints

Les objectifs fixés au début du projet ont été globalement atteints :

- Consultation simple et rapide des ressources disponibles.
- Gestion sécurisée des utilisateurs et contrôle fin des accès selon les rôles.
- Système de réservation fonctionnel avec workflow complet et gestion des conflits.
- Traçabilité des actions et suivi historique des réservations et des modifications.
- Tableaux de bord et statistiques pour l'analyse de l'utilisation des ressources.

Ces fonctionnalités contribuent à **une utilisation optimale et rationnelle des ressources informatiques** du Data Center.

Limites du système

Malgré les résultats satisfaisants, certaines **limites** ont été identifiées :

- Absence d'intégration directe avec les outils de virtualisation ou de supervision du Data Center.
- Notifications limitées à la messagerie interne (pas d'email ou SMS).
- Performances et gestion de charge non testées à grande échelle.
- Interface utilisateur pouvant être améliorée en termes d'ergonomie et d'accessibilité.

Ces limites sont principalement dues aux **contraintes de temps et de périmètre** du projet.

Perspectives d'évolution

Pour améliorer et enrichir l'application à l'avenir, plusieurs **évolutions** peuvent être envisagées :

- Intégration d'un système de notifications par **email et SMS**.
- Connexion avec des outils de **supervision et gestion automatique** des ressources.
- Ajout de fonctionnalités avancées de **reporting et analyses prédictives**.
- Amélioration de l'interface avec des **tableaux de bord plus interactifs et graphiques avancés**.
- Développement d'une **API** pour permettre l'interopérabilité avec d'autres systèmes ou applications.

Ces évolutions permettront de renforcer la **performance, la sécurité et l'efficacité globale** de la gestion du Data Center.

Conclusion finale

La réalisation de ce projet a constitué une étape importante dans l'acquisition de compétences techniques et méthodologiques en développement web moderne. Elle a permis de mettre en pratique plusieurs notions théoriques étudiées, notamment la conception et la gestion de bases de données relationnelles, la programmation orientée objet, la sécurisation des accès ainsi que la structuration d'une application web selon une architecture claire et maintenable.

Le développement de cette application a également renforcé la compréhension des mécanismes essentiels liés à la gestion des ressources informatiques d'un Data Center, en proposant une solution centralisée permettant de suivre la disponibilité des ressources, de gérer efficacement les réservations et d'assurer une traçabilité complète des actions effectuées par les utilisateurs. Grâce à l'intégration d'un système de rôles et de permissions, l'application garantit un contrôle précis des accès et une meilleure répartition des responsabilités entre les différents acteurs.

Par ailleurs, l'utilisation du framework Laravel a facilité l'implémentation de fonctionnalités avancées telles que l'authentification, la validation des données, la gestion des transactions et la sécurisation globale du système, tout en assurant une bonne évolutivité de l'application. L'interface utilisateur ergonomique et responsive contribue à améliorer l'expérience utilisateur et à rendre le système accessible sur différents supports.

En conclusion, l'application développée répond efficacement aux besoins actuels de gestion centralisée et sécurisée des ressources d'un Data Center. Elle constitue une solution fiable, moderne et évolutive, pouvant être enrichie ultérieurement par l'ajout de nouvelles fonctionnalités telles que l'automatisation avancée des rapports, l'intégration d'outils de supervision ou encore l'amélioration des statistiques d'exploitation. Ce projet représente ainsi une base solide pour des développements futurs et une contribution pertinente à l'optimisation de la gestion des infrastructures informatiques.

Bibliographie / Webographie

1. Framework Laravel et PHP :

- Laravel Documentation – <https://laravel.com/docs/10.x>
- PHP Documentation – <https://www.php.net/manual/en/>

2. Base de données et SQL :

- MySQL Documentation – <https://dev.mysql.com/doc/>
- C. J. Date, *An Introduction to Database Systems*, 8th Edition, 2003.

3. CSS et JavaScript :

- Mozilla Developer Network (MDN) – CSS :
<https://developer.mozilla.org/en-US/docs/Web/CSS>
- Mozilla Developer Network (MDN) – JavaScript :
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

4. Articles et ressources web sur la gestion des Data Centers :

- Uptime Institute, *Data Center Management Best Practices*, 2020.
- TechTarget, *Data Center Resource Management*, <https://www.techtarget.com/>