



Application Web de Réservation des Ressources d'un Data Center

Realise par:

Chahid Zaid

El boualiti Oussama

El kadi Monssif

El bahlouli Adil

Encadre par :

Prof. M'hamed AIT KBIR

Prof. Yasyn EL YUSUFI

2025/2026

Résumé

Ce projet vise à concevoir et développer une application Web de gestion des ressources informatiques d'un Data Center, basée sur Laravel (PHP) et MySQL. L'objectif principal est d'assurer une réservation efficace, sécurisée et transparente des ressources telles que les serveurs, machines virtuelles, équipements de stockage et matériels réseau.

L'application propose un système de réservation en temps réel avec gestion des disponibilités, des conflits et des périodes de maintenance. Elle repose sur une gestion multi-profils (invité, utilisateur interne, responsable technique et administrateur), chacun disposant de rôles et de droits adaptés. Des fonctionnalités de suivi des demandes, de notifications, de traçabilité et de statistiques sont intégrées afin d'améliorer l'exploitation et l'optimisation des ressources du Data Center.

Abstract

This project aims to design and develop a web-based application for managing Data Center IT resources using Laravel (PHP) and MySQL. The main objective is to provide an efficient, secure, and transparent system for reserving resources such as physical servers, virtual machines, storage systems, and network equipment.

The application includes a real-time reservation system with availability checks, conflict management, and maintenance scheduling. It supports multiple user roles (guest, internal user, technical manager, and administrator), each with specific permissions. Additional features such as request tracking, notifications, activity logging, and statistical dashboards are implemented to enhance resource monitoring and overall Data Center efficiency.

Table de Matières

Résumé.....	1
Abstract	2
Introduction générale	6
Chapitre 1 : Analyse et conception.....	7
1.1 Analyse du contexte	7
1.2 Problématique	7
1.3 Objectifs du projet	8
1.3.1 Objectif général	8
1.3.2 Objectifs fonctionnels	8
1.3.3 Objectifs non fonctionnels	11
Chapitre 2 : Réalisation et implémentation.....	12
2.1 Introduction du chapitre.....	12
2.2 Environnement de développement	12
2.2.1 Langage de programmation : PHP 8.x	12
2.2.2 Framework : Laravel.....	13
2.2.3 Base de données : MySQL.....	14
2.2.4 Technologies Front-end : HTML5, CSS3, JavaScript.....	14
2.2.5 Gestionnaire de dépendances : Composer.....	14
2.2.6 Outils de développement : Visual Studio Code et Git	14
2.4 Implémentation de la base de données.....	15
2.4.1 Migrations Laravel	15

2.4.2 Relations entre les tables	15
2.5 Gestion des utilisateurs et authentification	16
2.5.1 Authentification.....	16
2.5.2 Gestion des rôles et permissions	16
2.6 Implémentation du système de réservation.....	17
2.6.1 Création d'une réservation.....	17
2.6.2 Vérification des disponibilités	18
2.6.3 Gestion du cycle de vie d'une réservation.....	19
2.7 Gestion des ressources du Data Center	19
2.8 Notifications et journalisation.....	20
2.8.1 Notifications	20
2.8.2 Journalisation	21
2.9 Tableaux de bord et statistiques	22
2.10 Sécurité de l'application	25
Conclusion générale.....	25

Table des figures

Figure 1 Diagramme de cas d'utilisation global.....	9
Figure 2 Diagramme de séquence : processus de réservation	10
Figure 3 Diagramme de classes	11
Figure 4 Diagramme architecture MVC.....	12
Figure 5 Schema relationnel de base de données	15
Figure 6 Page de connexion.....	16
Figure 7 Interface de Gestion des Utilisateurs.....	17
Figure 8 Formulaire de création d'une réservation.....	18
Figure 9 Cycle de vie d'une réservation	19
Figure 10 Notification interne suite à l'approbation ou au refus d'une réservation	20
Figure 11 Notification interne adressée au responsable technique lors d'une demande de réservation	21
Figure 12 Journal d'audit	22
Figure 13 dashboard d'un administrateur	24

Introduction générale

Avec l'évolution rapide des technologies de l'information et la multiplication des services numériques, les Data Centers sont devenus des infrastructures critiques pour les établissements d'enseignement, les centres de recherche et les entreprises. Ils hébergent un ensemble de ressources informatiques hétérogènes telles que des serveurs physiques, des machines virtuelles, des systèmes de stockage et des équipements réseau, indispensables au bon fonctionnement des activités pédagogiques, scientifiques et administratives.

Cependant, la gestion manuelle ou partiellement automatisée de ces ressources pose plusieurs défis, notamment en matière de disponibilité, de planification, de traçabilité et de sécurité. L'absence d'un système centralisé de réservation peut entraîner des conflits d'utilisation, une sous-exploitation ou une surcharge des ressources, ainsi qu'un manque de visibilité pour les responsables techniques et les administrateurs du Data Center.

Dans ce contexte, la mise en place d'une application Web dédiée à la gestion et à la réservation des ressources informatiques apparaît comme une solution pertinente et nécessaire. Une telle plateforme permet non seulement de centraliser l'information, mais aussi d'automatiser les processus de demande, de validation et de suivi des ressources, tout en garantissant un accès contrôlé selon les rôles et responsabilités des utilisateurs.

Ce projet s'inscrit dans cette dynamique et vise à concevoir et développer une application Web sécurisée et performante, basée sur le framework **Laravel (PHP)** et le système de gestion de base de données **MySQL**. L'application offre une interface ergonomique et responsive, intégrant un système de réservation en temps réel, une gestion multi-profils, des mécanismes de notification, ainsi que des outils de suivi et de statistiques. L'objectif final est d'améliorer l'efficacité opérationnelle du Data Center, d'optimiser l'utilisation des ressources et d'assurer une gestion transparente et fiable au service des utilisateurs.

Chapitre 1 : Analyse et conception

1.1 Analyse du contexte

Un Data Center regroupe un ensemble de ressources informatiques critiques mises à disposition de différents utilisateurs pour des besoins variés tels que l'enseignement, la recherche scientifique ou l'administration. Ces ressources incluent notamment des serveurs physiques, des machines virtuelles, des systèmes de stockage et des équipements réseau. Leur gestion nécessite une organisation rigoureuse afin d'assurer une disponibilité optimale, une utilisation équitable et une traçabilité complète.

Dans de nombreux environnements, les demandes de ressources sont encore traitées de manière informelle (emails, formulaires papier ou échanges directs), ce qui engendre des problèmes de coordination, des conflits de réservation et un manque de visibilité globale. Il devient alors indispensable de disposer d'un système informatisé permettant de centraliser les informations, d'automatiser les processus et de sécuriser les accès.

1.2 Problématique

La problématique principale de ce projet peut être formulée comme suit :

Comment mettre en place une application Web permettant de gérer efficacement la réservation, l'allocation et le suivi des ressources informatiques d'un Data Center, tout en tenant compte des rôles des utilisateurs, des contraintes de disponibilité et des exigences de sécurité ?

Cette problématique soulève plusieurs défis :

- La gestion des conflits de réservation et des périodes de maintenance.
- La différenciation des accès selon les profils utilisateurs.
- La traçabilité des actions et de l'historique d'utilisation.
- La présentation claire et synthétique des informations pour faciliter la prise de décision.

1.3 Objectifs du projet

Le projet de gestion des ressources informatiques d'un Data Center s'inscrit dans une démarche d'amélioration de la gouvernance, de l'optimisation et de la traçabilité des infrastructures informatiques. Les objectifs définis tiennent compte des contraintes techniques, organisationnelles et sécuritaires propres à un environnement de Data Center.

1.3.1 Objectif général

L'objectif général de ce projet est de concevoir et développer une application Web centralisée permettant la **réservation, l'allocation et le suivi des ressources informatiques d'un Data Center**, tout en garantissant un accès sécurisé et contrôlé selon les profils des utilisateurs.

Cette application vise à remplacer les méthodes traditionnelles de gestion manuelle par une solution numérique fiable, capable de :

- Centraliser toutes les informations relatives aux ressources informatiques.
- Automatiser les processus de demande, de validation et de suivi des réservations.
- Améliorer la visibilité globale sur l'état et le taux d'occupation du Data Center.
- Réduire les conflits d'utilisation et les erreurs humaines.

1.3.2 Objectifs fonctionnels

Les objectifs fonctionnels décrivent les fonctionnalités principales que l'application doit offrir afin de répondre aux besoins des utilisateurs.

a) Diagramme de cas d'utilisation global

Pour représenter de manière globale les fonctionnalités proposées par l'application ainsi que les interactions entre le système et les différents profils d'utilisateurs, un diagramme des

cas d'utilisation a été créé. Ce dernier permet d'identifier clairement les actions principales réalisables par chaque acteur (invité, utilisateur interne, responsable technique, et administrateur), ainsi que de définir les limites fonctionnelles du système. Il s'agit d'une vue synthétique essentielle pour la compréhension du fonctionnement général de l'application.

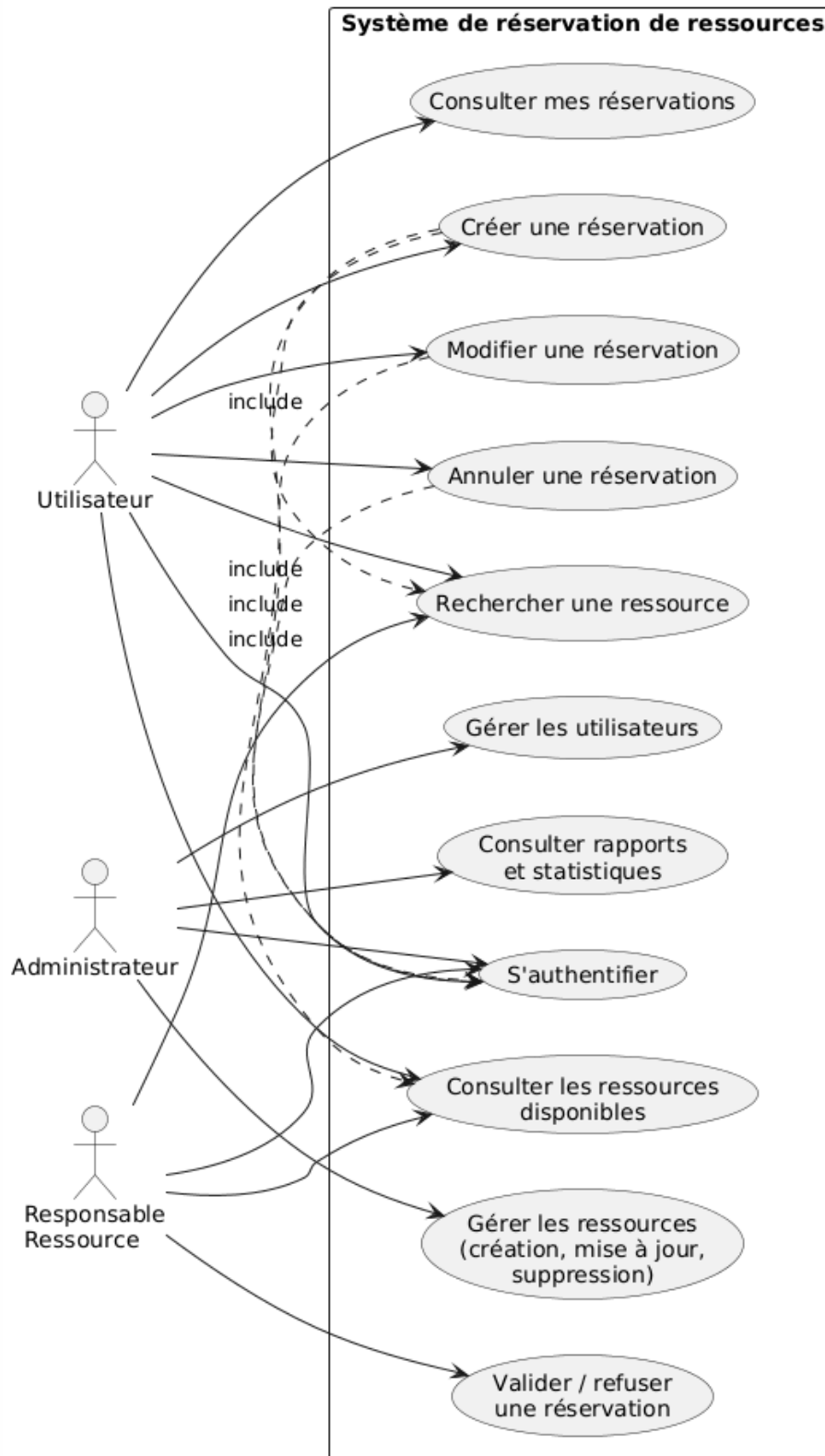


Figure 1 Diagramme de cas d'utilisation global

b) Diagramme de séquence

Dans le but d'étudier les interactions entre les différents composants du système lors de l'exécution d'une fonctionnalité spécifique, un diagramme de séquence a été conçu. Ce diagramme montre, de manière chronologique, les interactions entre les différents composants du système, à savoir l'utilisateur, l'interface, les contrôleurs, et la base de données, lors de la création d'une demande de réservation.

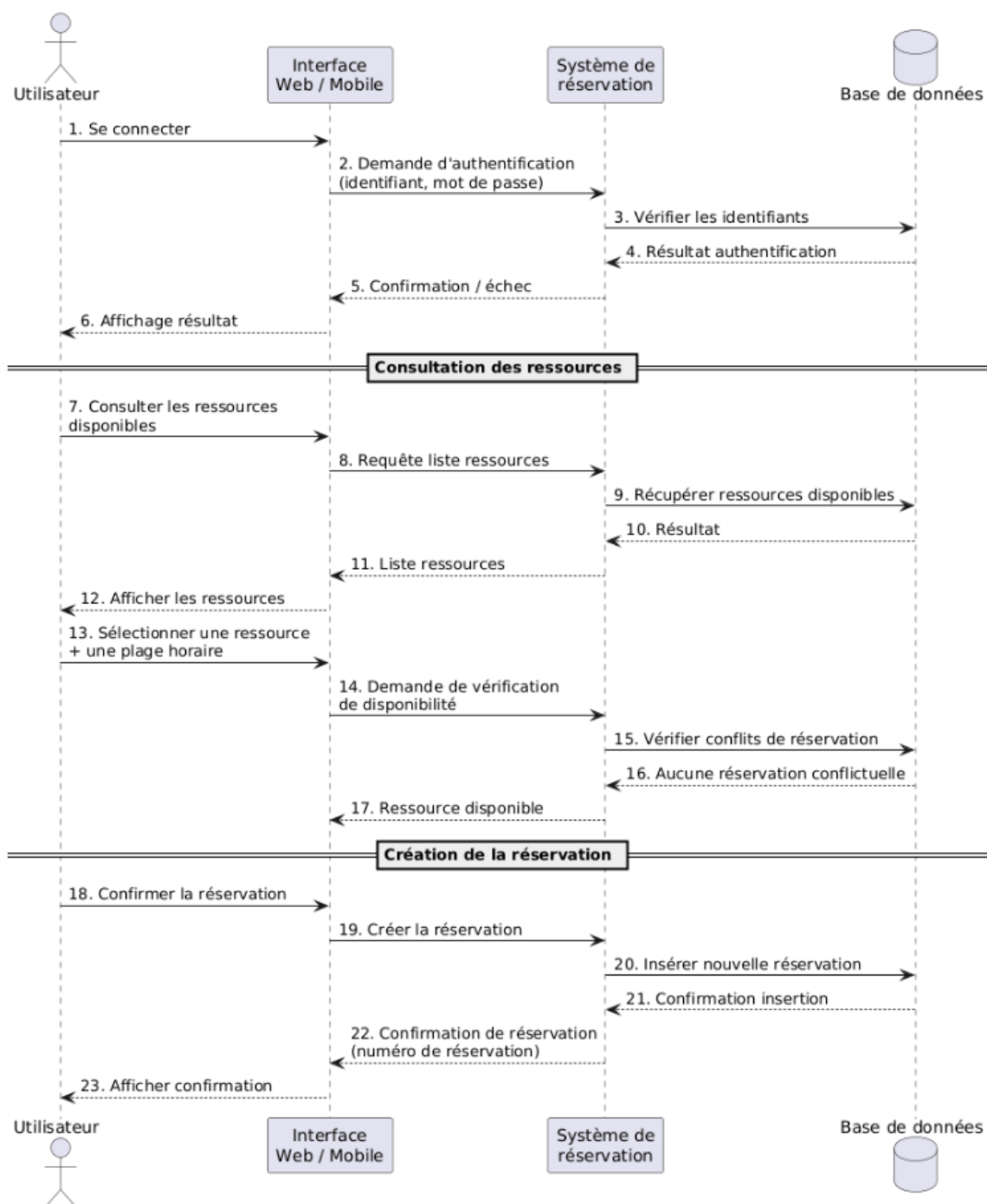


Figure 2 Diagramme de séquence : processus de réservation

c) Diagramme de classes

Pour définir la structure statique de l'application et les relations entre les différentes entités du système, un diagramme de classes a été développé. Ce dernier montre les différentes classes, les attributs et les méthodes de ces classes, ainsi que les relations existantes entre les entités du système. Il s'agit d'un élément essentiel dans la définition de la base de données et l'implémentation du modèle avec le framework Laravel.

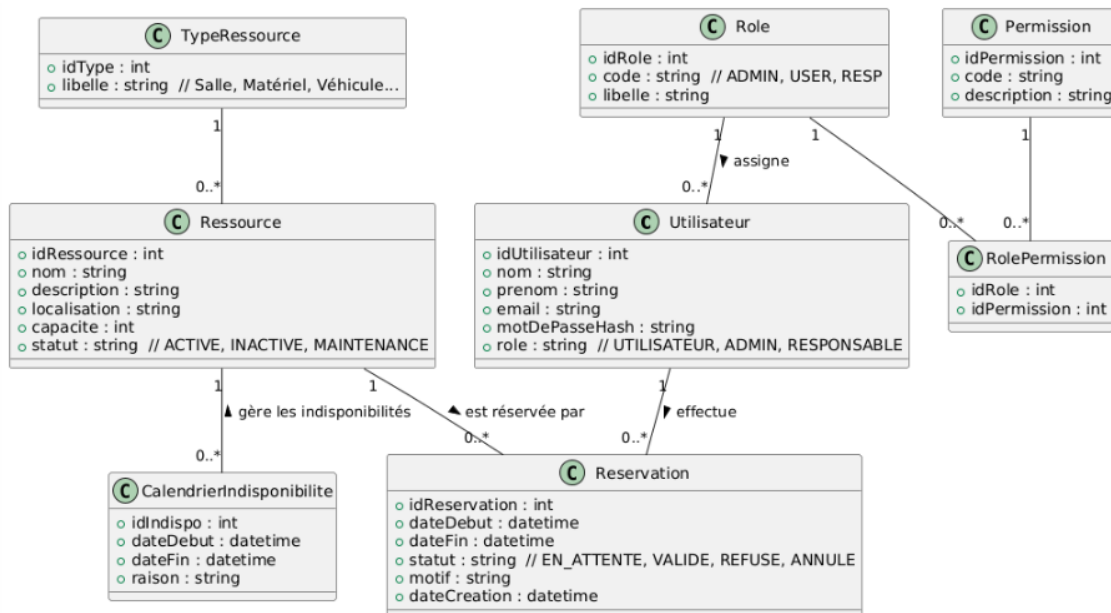


Figure 3 Diagramme de classes

1.3.3 Objectifs non fonctionnels

Les objectifs non fonctionnels garantissent la qualité et la pérennité de l'application :

- **Sécurité** : contrôle des accès, protection des données, journalisation.
- **Performance** : temps de réponse optimisé même en cas de forte charge.
- **Ergonomie** : interface intuitive, responsive et accessible.
- **Maintenabilité** : code structuré, évolutif et conforme aux bonnes pratiques Laravel.

- **Scalabilité** : possibilité d'ajouter de nouvelles ressources et fonctionnalités.

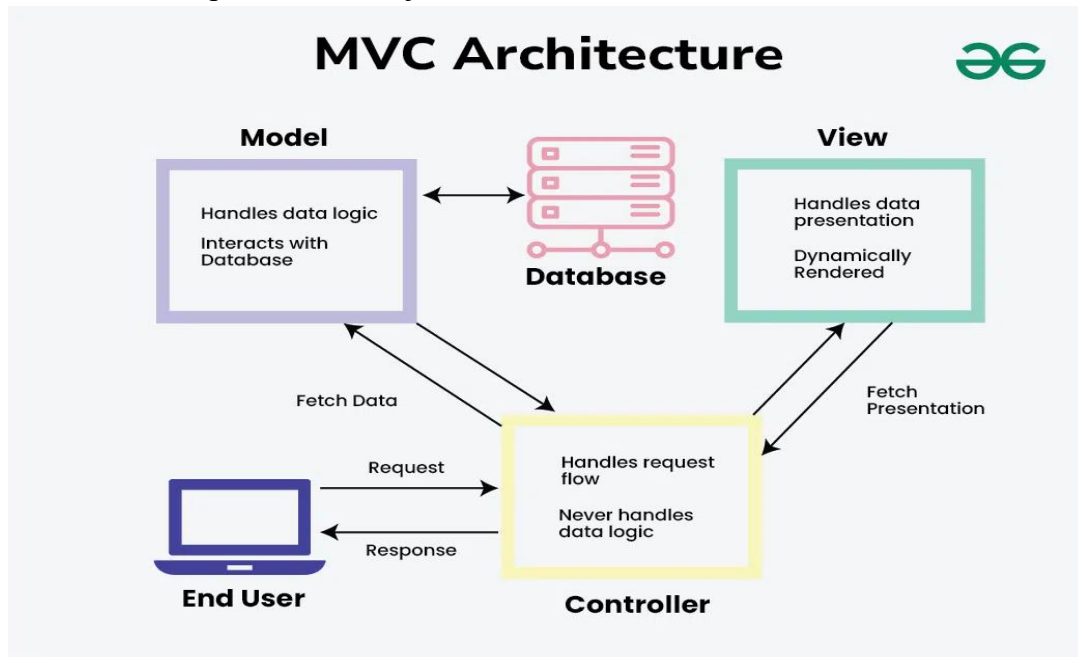


Figure 4 Diagramme architecture MVC

Chapitre 2 : Réalisation et implémentation

2.1 Introduction du chapitre

Après l'analyse des besoins et la conception générale de l'application présentées dans le chapitre précédent, ce chapitre est consacré à la **réalisation technique et à l'implémentation** de la solution proposée. Il décrit les choix technologiques effectués, l'architecture mise en place ainsi que les différentes fonctionnalités développées. L'objectif est de montrer comment la conception théorique a été traduite en une application Web fonctionnelle, sécurisée et performante.

2.2 Environnement de développement

Le développement de l'application a été réalisé en s'appuyant sur un ensemble de technologies modernes, robustes et largement utilisées dans le domaine du développement Web. Ces choix technologiques visent à garantir la performance, la sécurité, la maintenabilité et l'évolutivité de l'application, tout en facilitant son déploiement et sa prise en main par les différents acteurs du projet.

2.2.1 Langage de programmation : PHP 8.x



PHP est un langage de programmation côté serveur largement utilisé pour le développement d'applications Web dynamiques. La version 8.x apporte des améliorations significatives en termes de performance, de gestion de la mémoire et de sécurité. Son intégration native avec le framework Laravel permet de structurer efficacement la logique métier, de simplifier l'accès aux bases de données et de renforcer la fiabilité globale de l'application.

2.2.2 Framework : Laravel



Laravel est un framework PHP moderne basé sur l'architecture MVC (Model–View–Controller). Il offre un ensemble complet d'outils facilitant le développement d'applications Web sécurisées et maintenables. Parmi ses principales fonctionnalités figurent la gestion de l'authentification, des routes, des middlewares, ainsi que l'ORM Eloquent pour l'interaction avec la base de données. Laravel permet également d'appliquer de bonnes pratiques de développement et d'accélérer la mise en œuvre des fonctionnalités complexes.

2.2.3 Base de données : MySQL



MySQL est un système de gestion de base de données relationnelle reconnu pour sa stabilité, ses performances et sa fiabilité. Il est particulièrement adapté aux applications Web nécessitant une gestion structurée et cohérente des données. Dans ce projet, MySQL est utilisé pour stocker l'ensemble des informations relatives aux utilisateurs, aux ressources, aux réservations, aux maintenances et à la journalisation, tout en garantissant l'intégrité et la cohérence des données grâce aux contraintes relationnelles.

2.2.4 Technologies Front-end : HTML5, CSS3, JavaScript

Le développement de l'interface utilisateur repose sur les technologies standards du Web. HTML5 est utilisé pour structurer le contenu des pages, tandis que CSS3 permet de définir leur apparence visuelle. Le framework Bootstrap facilite la création d'interfaces ergonomiques et responsives, adaptées à différents types d'écrans. JavaScript est utilisé pour améliorer l'interactivité, la validation des formulaires et l'expérience utilisateur globale.

2.2.5 Gestionnaire de dépendances : Composer

Composer est un gestionnaire de dépendances pour PHP qui permet de gérer efficacement les bibliothèques et les packages nécessaires au projet. Il simplifie l'installation, la mise à jour et la maintenance des dépendances utilisées par Laravel. Grâce à Composer, le projet bénéficie d'une meilleure organisation du code et d'une compatibilité optimale entre les différentes bibliothèques.

2.2.6 Outils de développement : Visual Studio Code et Git



2.4.1 Migrations Laravel

- Une gestion versionnée du schéma

- Chaque table a été définie avec ses clés primaires, étrangères et contraintes d'intégrité.

- Un utilisateur peut effectuer plusieurs réservations
- Une ressource peut être réservée plusieurs fois à des périodes différentes
- Une ressource appartient à une catégorie
- Une réservation est liée à un utilisateur et à une ressource

- [illegible]

Figure 5 Schema relationnel de base de données

2.5 Gestion des utilisateurs et authentification

2.5.1 Authentification

Laravel fournit un système d'authentification sécurisé incluant :

- Inscription
- Connexion
- Déconnexion
- Gestion des mots de passe

Les mots de passe sont chiffrés à l'aide de l'algorithme **bcrypt**.

2.5.2 Gestion des rôles et permissions

Chaque utilisateur est associé à un rôle :

- Invité
- Utilisateur interne
- Responsable technique
- Administrateur

Les accès sont contrôlés à l'aide de **middlewares**, garantissant que chaque utilisateur n'accède qu'aux fonctionnalités autorisées.

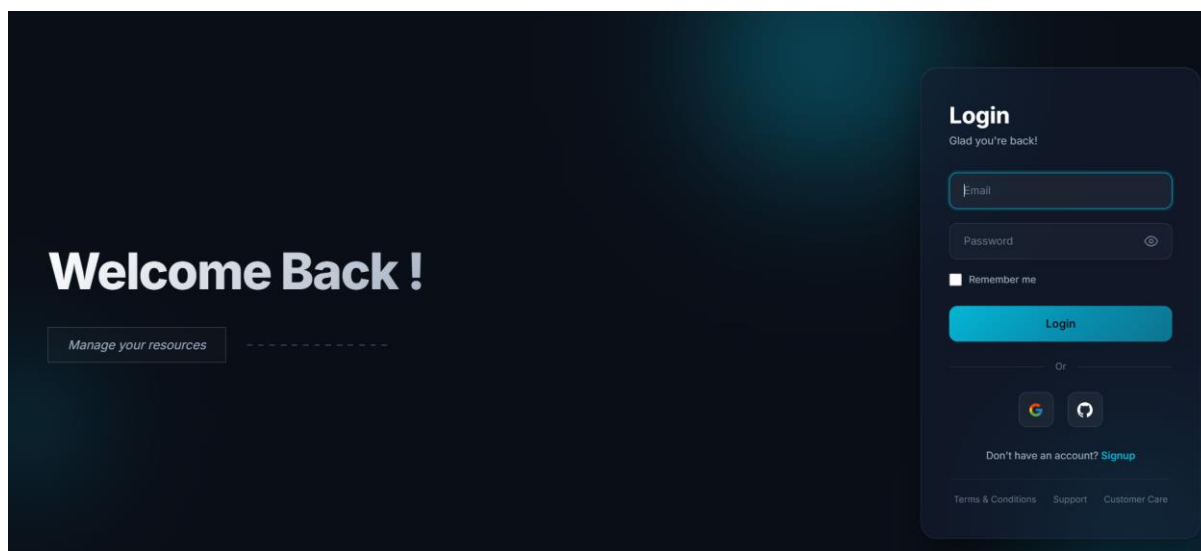


Figure 6 Page de connexion

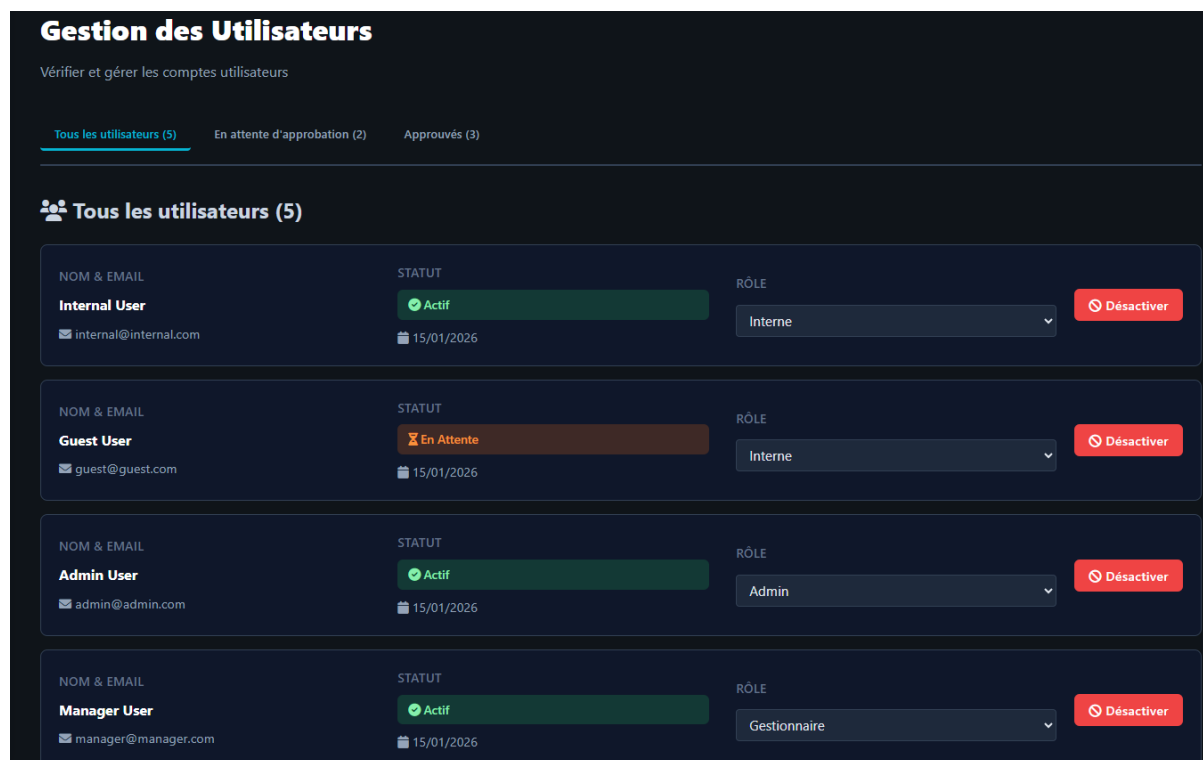


Figure 7 Interface de Gestion des Utilisateurs

2.6 Implémentation du système de réservation

2.6.1 Création d'une réservation

Un utilisateur interne peut soumettre une demande de réservation en précisant :

- La ressource souhaitée
- La période de réservation
- Une justification

Réserver la ressource : Server Alpha

Type : Serveur

Description : Serveur physique principal

Période d'occupation et disponibilité :

- Occupé : 25/01/2026 → 25/01/2026
- Occupé : 25/01/2026 → 29/01/2026
- ☑ Libre en dehors de ces périodes

Date de début :

mm/dd/yyyy --:-- --

Date de fin :

mm/dd/yyyy --:-- --

Justification (Pourquoi avez-vous besoin de cette ressource ?) :

Envoyer la demande

Figure 8 Formulaire de création d'une réservation

2.6.2 Vérification des disponibilités

Le système vérifie automatiquement :

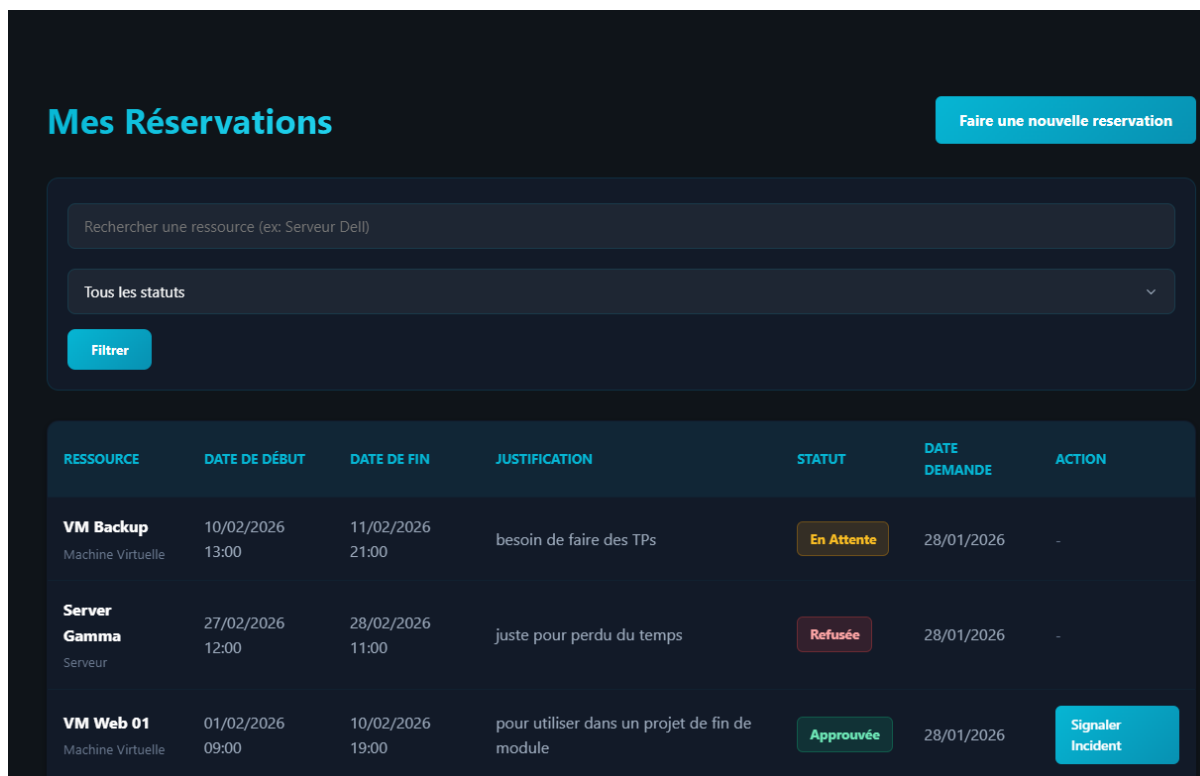
- La disponibilité de la ressource

- Les conflits de réservation (chevauchement des dates)
- Les périodes de maintenance

2.6.3 Gestion du cycle de vie d'une réservation

La réservation peut se trouver dans différents états en fonction de son niveau d'avancement. Ces états peuvent être En attente, Approuvée ou Refusée.

Ces états aident à gérer les ressources du Data Center de manière précise et à tenir les utilisateurs renseignés sur l'état de leur réservation.



Mes Réservations [Faire une nouvelle reservation](#)

Rechercher une ressource (ex: Serveur Dell)

Tous les statuts

Filtrer

RESSOURCE	DATE DE DÉBUT	DATE DE FIN	JUSTIFICATION	STATUT	DATE DEMANDE	ACTION
VM Backup Machine Virtuelle	10/02/2026 13:00	11/02/2026 21:00	besoin de faire des TPs	En Attente	28/01/2026	-
Server Gamma Serveur	27/02/2026 12:00	28/02/2026 11:00	juste pour perdu du temps	Refusée	28/01/2026	-
VM Web 01 Machine Virtuelle	01/02/2026 09:00	10/02/2026 19:00	pour utiliser dans un projet de fin de module	Approuvée	28/01/2026	Signaler Incident

Figure 9 Cycle de vie d'une reservation


2.7 Gestion des ressources du Data Center

Les responsables techniques peuvent :

- Ajouter de nouvelles ressources
- Modifier leurs caractéristiques
- Mettre une ressource en maintenance

- Désactiver une ressource

Chaque modification est enregistrée dans le journal des actions.

 Image à insérer : Interface de gestion des ressources

2.8 Notifications et journalisation

2.8.1 Notifications

Le système envoie des notifications internes lors :

- De l'approbation ou du refus d'une réservation

Lorsqu'une demande de réservation est traitée par le responsable technique, le système notifie automatiquement l'utilisateur concerné. Ces notifications internes permettent d'informer l'utilisateur, en temps réel, de l'approbation ou du refus de sa demande, garantissant ainsi une communication rapide et efficace.

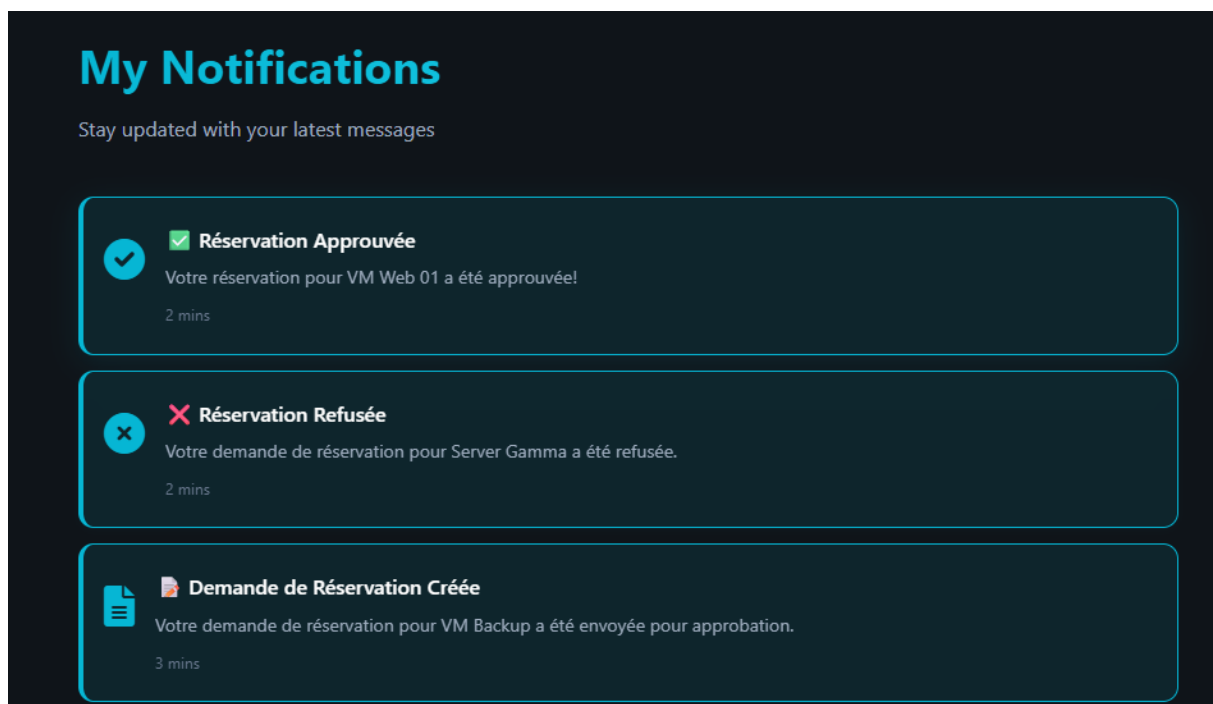


Figure 10 Notification interne suite à l'approbation ou au refus d'une réservation

- De demande d'une reservation

Lorsqu'une nouvelle demande de réservation est soumise, le système envoie automatiquement une notification interne au responsable technique concerné. Cette notification lui permet d'être informé rapidement des nouvelles sollicitations et de prendre une décision d'approbation ou de refus en fonction de la disponibilité des ressources. Ce

mécanisme améliore la réactivité et l'efficacité dans la gestion des ressources du Data Center.

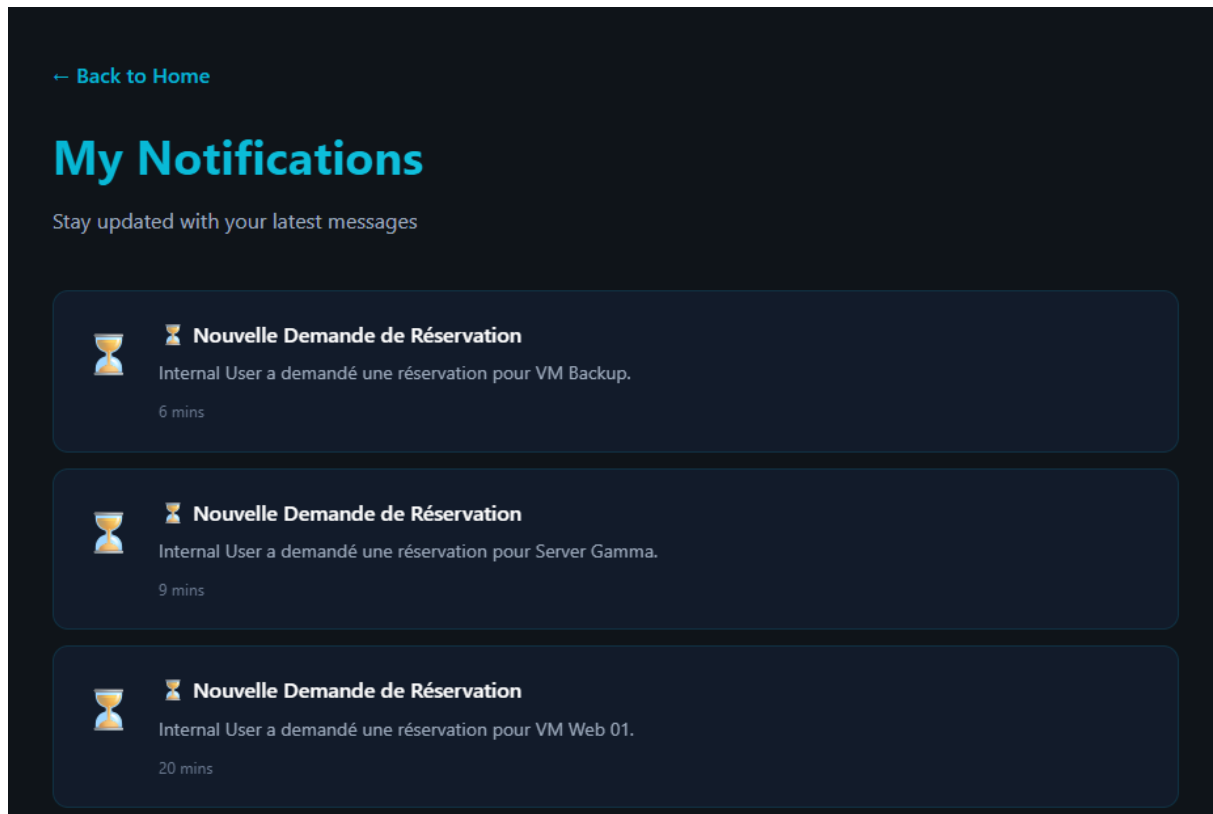


Figure 11 Notification interne adressée au responsable technique lors d'une demande de réservation

2.8.2 Journalisation

Toutes les actions importantes sont enregistrées :

- Connexions
- Créations
- Modifications
- Suppressions

Journal d'audit

HORODATAGE	UTILISATEUR	ACTION	DÉTAILS
28/01/2026 01:00:13	Manager User	Mise à jour Réserve	Réserve ID: 9 mise à jour (Statut: pending → approved)
28/01/2026 01:00:00	Manager User	Mise à jour Réserve	Réserve ID: 10 mise à jour (Statut: pending → refused)
28/01/2026 00:58:47	Internal User	Création Réserve	Nouvelle réserve pour la ressource ID: 6
28/01/2026 00:55:46	Internal User	Création Réserve	Nouvelle réserve pour la ressource ID: 3
28/01/2026 00:44:37	Internal User	Création Réserve	Nouvelle réserve pour la ressource ID: 4
28/01/2026 00:40:44	Internal User	Création Réserve	Nouvelle réserve pour la ressource ID: 6
27/01/2026 23:44:37	Manager User	Mise à jour Réserve	Réserve ID: 7 mise à jour (Statut: pending → approved)
27/01/2026 22:54:11	Internal User	Création Réserve	Nouvelle réserve pour la ressource ID: 7
27/01/2026 20:51:22	Internal User	Création Réserve	Nouvelle réserve pour la ressource ID: 7
27/01/2026 20:47:41	Manager User	Mise à jour Réserve	Réserve ID: 5 mise à jour (Statut: approved → refused)

Figure 12 Journal d'audit

2.9 Tableaux de bord et statistiques

L'application propose des tableaux de bord personnalisés :

- Utilisateur : réservations et historique
- Responsable technique : ressources supervisées
- Administrateur : vue globale du Data Center

Les statistiques incluent :

- Taux d'occupation
- Répartition par type de ressource
- Historique d'utilisation

✂ Images à insérer :

- Dashboard utilisateur

Le tableau de bord utilisateur permet à chaque utilisateur authentifié d'accéder rapidement aux informations liées à ses réservations. Il affiche les demandes en cours, leur état, l'historique des réservations ainsi que les notifications récentes. Cette interface simplifie le suivi des activités et améliore l'expérience utilisateur.



- Dashboard administrateur.

Le tableau de bord administrateur offre une vue globale sur l'ensemble du Data Center. Il permet de consulter les statistiques générales telles que le nombre de ressources, les réservations en cours, les taux d'occupation et l'état des utilisateurs. Cet espace facilite la supervision, la prise de décision et le pilotage stratégique des ressources informatiques

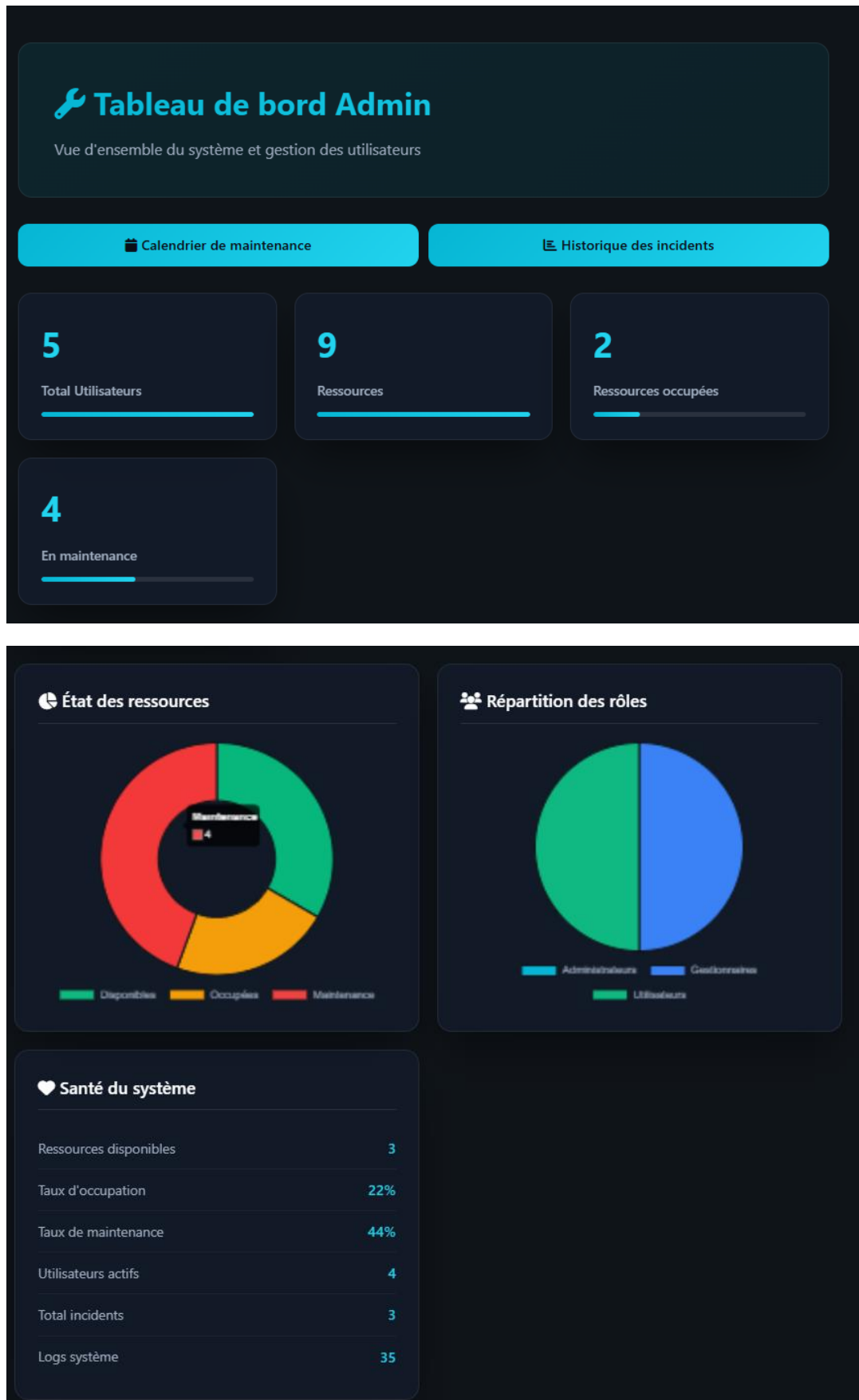


Figure 13 Dashboard d'un administrateur

2.10 Sécurité de l'application

La sécurité de l'application est assurée à travers plusieurs mécanismes qui sont intégrés dans le framework de Laravel. L'accès au système est assuré par une authentification et une autorisation par rôles, ce qui garantit que chaque utilisateur n'accède qu'aux fonctionnalités qui lui sont autorisées.

La protection contre les attaques de CSRF, la validation des données côté serveur, et la gestion des sessions de manière sécurisée permettent de prévenir les accès non autorisés et les erreurs de manipulation.

Enfin, la journalisation des accès et des actions importantes assure une traçabilité complète, renforçant ainsi la fiabilité globale du système.

Les mécanismes de sécurité mis en place incluent :

- Authentification et autorisation
- Protection CSRF
- Validation des données
- Gestion des sessions
- Journalisation des accès

Conclusion générale

La gestion efficace des ressources informatiques constitue un enjeu stratégique majeur pour les Data Centers, en particulier dans les environnements académiques et de recherche où

les besoins sont variés et en constante évolution. L'objectif principal de ce projet était de concevoir et de développer une application Web permettant de centraliser, organiser et optimiser la réservation, l'allocation et le suivi des ressources informatiques d'un Data Center de manière sécurisée et transparente.

À travers une phase d'analyse approfondie des besoins et une conception rigoureuse, une solution adaptée aux contraintes organisationnelles et techniques a été proposée. L'application développée, basée sur les technologies **Laravel (PHP)** et **MySQL**, offre une gestion multi-profils avec des rôles et permissions différenciés, garantissant un contrôle d'accès strict et une meilleure gouvernance des ressources. Le système de réservation en temps réel, la gestion des conflits, la planification des maintenances ainsi que les mécanismes de notification et de journalisation contribuent à une exploitation plus fiable et plus efficace de l'infrastructure informatique.

La réalisation du projet a également permis de mettre en pratique les bonnes pratiques du développement Web moderne, notamment l'utilisation de l'architecture MVC, la sécurisation des données, la conception d'interfaces ergonomiques et la mise en place de tableaux de bord statistiques facilitant la prise de décision. Les tests effectués ont démontré que l'application répond aux exigences fonctionnelles et non fonctionnelles définies, tout en offrant une expérience utilisateur satisfaisante.

Malgré les résultats obtenus, certaines limites subsistent, notamment en ce qui concerne l'automatisation avancée de l'allocation des ressources et l'intégration avec des outils externes de supervision ou de virtualisation. Ces aspects ouvrent la voie à plusieurs perspectives d'amélioration, telles que l'ajout de mécanismes d'intelligence artificielle pour l'optimisation des ressources, l'intégration de solutions de monitoring en temps réel ou encore l'ouverture de l'application via des API.

En conclusion, ce projet constitue une contribution significative à l'amélioration de la gestion des ressources d'un Data Center. Il offre une solution évolutive et fiable, capable de s'adapter aux besoins futurs, tout en posant des bases solides pour des développements ultérieurs.