UNIVERSITÉ MOHAMMED V DE RABAT

Faculté des Sciences



Département d'informatique

Filière Licence Fondamentale

en Sciences Mathématiques et Informatique

PROJET DE FIN D'ÉTUDES

Intitulé:

Conception et réalisation d'un système de Contrôle de Systèmes d'Information

Présenté par :

OTHMANE HRIMAT
NOUAMAN JABER

Soutenu le 24 Juin 2024 devant le Jury

M. Mohammed El Amrani	Professeur à la Faculté des Sciences - Rabat	Président
M. Hamid Khalifi	Professeur à la Faculté des Sciences - Rabat	Encadrant

Année Universitaire : 2023-2024

Remerciement

Nous tenons tout d'abord à remercier **DIEU**, qui nous a donné la santé et la force de mener à bien ce stage de fin d'études.

Nous remercions également nos parents, pour leur soutien inconditionnel et leur encouragement tout au long de notre parcours académique.

Nous tenons à exprimer chaleureusement notre gratitude envers nos encadrants, M. Outman Ziati et M. Hamid Khalifi, pour leur soutien indéfectible, leurs conseils avisés et leur disponibilité tout au long de ce projet. Leur expertise et leur encadrement ont été essentiels à l'aboutissement de ce travail.

Nous souhaitons également adresser nos remerciements à nos professeurs de la filière SMI des semestres S5 et S6. Vos enseignements de qualité et vos implications ont joué un rôle crucial dans notre formation et dans la réussite de ce projet.

Enfin, nous remercions tous les membres de l'équipe pédagogique qui ont contribué à notre parcours académique et à l'acquisition des compétences nécessaires à la réalisation de ce projet.

Merci à tous pour votre précieuse aide et votre soutien constant.

Résumé

Le présent rapport a pour objectif de présenter notre projet de développement d'une application web et mobile appelée FixMate, destinée à améliorer la gestion et le contrôle des systèmes d'information. Notre objectif était de fournir une solution intégrée pour les entreprises souhaitant surveiller en temps réel leurs infrastructures numériques, y compris les applications, les serveurs, les équipements et les réseaux.

Au cours de ce projet, nous avons effectué une analyse minutieuse des besoins des utilisateurs afin de concevoir une application répondant à leurs attentes. Nous avons développé des interfaces utilisateurs conviviales pour faciliter la navigation et l'utilisation de l'application, ainsi que des fonctionnalités essentielles telles que la gestion des droits d'accès, la recherche multicritères et l'envoi automatisé de notifications.

Le rapport présente en détail les différentes étapes de développement du projet, allant de l'analyse des besoins à la conception, en passant par la spécification des exigences et le développement de l'application. Nous avons également abordé les outils et technologies utilisés pour le développement de l'application, ainsi que les résultats finaux obtenus. FixMate offre une solution complète et efficace pour la gestion des systèmes d'information, assurant une performance optimale, une sécurité renforcée et une expérience utilisateur enrichie.

Liste des acronymes

IOS : iPhone Operating System.

HTML : HyperText Markup Language.

CSS : Cascading Style Sheets.

JS : JavaScript.

UML : Unified Modeling Language.

UI : User Interface.

PHP : Hypertext Preprocessor.

SQL : Structured Query Language.

API : Application Programming Interface.

SDK : Software Development Kit.

SMTP : Simple Mail Transfer Protocol.

MVC : Model-View-Controller.

JSON : JavaScript Object Notation.

XML : eXtensible Markup Language.

Liste des figures

Figure 1: Logo de FixMate

Figure 2: Diagramme de Gantt

Figure 3: Diagramme de navigation

Figure 4: Logo de StarUML

Figure 5: Site-Map

Figure 6: Diagramme de cas d'utilisation d'un Admin

Figure 7: Diagramme de cas d'utilisation d'un Manager

Figure 8: Diagramme de cas d'utilisation d'un utilisateur

Figure 9: Diagramme de classes de projet

Figure 10 : Logo du langage MySql

Figure 11: Logo du framework Laravel

Figure 12: Architecture MVC

Figure 13: Schéma des technologies utilisés dans le projet

Figure 14 : L'écran de chargement

Figure 15 : L'écran de connexion

Figure 16 : Tableau de Bord

Figure 17 : La page des fournisseurs

Figure 18 : Formulaire d'ajout d'un fournisseur

Figure 19 : La page des utilisateurs

Figure 20 : Formulaire d'ajout d'un utilisateur

Figure 21 : La page des équipes

Figure 22 : La page des sites web

Figure 23 : La page des Équipements

Figure 24 : La page des Locations

Figure 25 : Onglet Sous-Locations

Figure 26 : Onglet Équipements

Figure 27 : Onglet Contrôles Effectués

Figure 28 : Ajout de Contrôle dans une Location

Figure 29 : Page des Contrôles

Figure 30 : Formulaire d'Ajout de Contrôle

Figure 31 : Page Principale d'un Contrôle

Figure 32 : Modification d'un Contrôle

Figure 33: Page SplashScreen de FixMate

Figure 34: Page d'accueil de FixMate

Figure 35 : Page de Location de FixMate

Figure 36 : Page de liste des contrôles de FixMate

Figure 37 : page de liste des contrôles terminés

Figure 38 : Page de l'ajout de contrôles

Figure 39 : page de la liste des équipements de FixMate

igure 40 : Page introductive 1 de FixMate

Figure 41 : Page introductive 2 de FixMate

Figure 42 : Page introductive 3 de FixMate

Figure 43 : Page de début d'opération de contrôle

Figure 44 : Page d’authentification

Figure 45 : Page vous informe que vous avez bien s'authentifier

Sommaire

Remerciement **3**
Résumé **4**
Liste des acronymes **5**
Liste des figures **6**
Sommaire **8**
Introduction **10**
Partie I **11**
Présentation générale du projet **11**
1. Problématique 12
2. Présentation de l'application 12
3. Étude des exigences 13
Conclusion 17
Partie II **18**
Planification de gestion de projet **18**
1. Identification des objectifs de projet 19
2. Méthodologie de gestion de projet 20
3. Planification de projet 21
4. Diagramme de Gantt : 23
5. Diagramme de navigation 24
6. Gestion des risques 24
Conclusion 27
Partie III **28**
I. Conception de FixMate **29**
1. Outils de conception et modélisation 29
2. Site-Map 30
3. Diagramme de cas d'utilisation 30
3.1. Définition 30
4. Diagramme de class 34
4.1. Définition 34
4.2. Diagramme de classes de projet 35
Partie IV **36**
1. Outils et technologies utilisés 37
1.1. Back-End 37
1.2. Architecture MVC 43

1.3. Utilisation des API 44
Conclusion 46
2. Développement de l'application web 47
2.1. Technologies et outils pour les interfaces web 47
.2.2 Interfaces du l'application web 48
3. Développement de l'application mobile 65
3.1. Technologies et outils pour les interfaces mobiles 65
3.2. Interfaces du l'application mobile 65
Conclusion 78
Conclusion générale 79
Bibliographie 80

Introduction

Les systèmes d'information sont devenus des éléments incontournables pour le bon fonctionnement des entreprises modernes. Ils permettent de collecter, traiter et diffuser des informations essentielles à la prise de décision. Cependant, la complexité croissante de ces systèmes nécessite des solutions de contrôle robustes et flexibles pour assurer leur bon fonctionnement et leur sécurité.

Notre travail s'articule autour de trois parties principales :

- **Partie 1 : Spécification fonctionnelle des besoins** : Cette première étape consiste à recueillir et à définir les besoins fonctionnels du système. Il s'agit de comprendre les exigences des utilisateurs finaux et de formaliser les fonctionnalités attendues de l'application. Cette phase est cruciale car elle pose les bases sur lesquelles reposera l'ensemble du projet.
- **Partie 2 : Analyse et conception** : Une fois les besoins clairement définis, nous avons procédé à une analyse approfondie pour élaborer une architecture cohérente et efficace. Cette partie comprend la modélisation des données, la conception des interfaces, ainsi que l'établissement des flux de travail nécessaires pour répondre aux besoins spécifiés.
- **Partie 3 : Réalisation et mise en œuvre** : La dernière phase de notre projet consiste à la concrétisation de l'analyse et de la conception sous forme d'applications opérationnelles. Nous avons développé et intégré les fonctionnalités spécifiées dans les environnements web et mobile, en veillant à la qualité du code et à la performance des applications.

Ce rapport vise à présenter de manière détaillée chacune de ces étapes, en illustrant les choix techniques et méthodologiques que nous avons fait tout au long de notre projet. Nous espérons que notre travail apportera une contribution significative à la gestion et au contrôle des systèmes d'information et qu'il servira de référence pour des projets futurs similaires.

Partie I

Présentation générale du projet

1. Problématique

Les systèmes d'information modernes incluent un large éventail de composants, tels que des applications, des réseaux wifi, des équipements, des infrastructures, des serveurs, et des matériels audiovisuels. Assurer la disponibilité, la performance et la sécurité de ces systèmes nécessite des contrôles réguliers et efficaces. Cependant, l'absence d'une solution intégrée de gestion et de contrôle complique le suivi et la vérification de l'état de ces systèmes. Les contrôles manuels et les solutions disparates entraînent des inefficacités, des retards dans l'identification des problèmes et des risques accrus d'interruptions de service, affectant ainsi la performance globale de l'entreprise.

2. Présentation de l'application

Figure 1 : Logo de FixMate

FixMate est une application de gestion et de contrôle des systèmes d'information conçue pour offrir une solution intégrée et efficace. Elle permet aux utilisateurs de surveiller et de gérer divers composants des systèmes d'information, tels que les applications, les équipements, les serveurs, les infrastructures réseau et les matériels audiovisuels, en garantissant un suivi précis et en temps réel de leur statut et de leur performance.

FixMate aide les entreprises à maintenir la performance et la sécurité de leurs systèmes d'information en offrant des fonctionnalités complètes de contrôle et de gestion. Les utilisateurs peuvent paramétrer les différentes ressources, effectuer des recherches multicritères, gérer les droits d'accès et visualiser l'état des systèmes via un tableau de bord interactif. Chaque contrôle réalisé peut inclure des détails tels que la date, l'heure, l'identité du contrôleur et des observations spécifiques avec des indicateurs visuels pour signaler les états de fonctionnement (flèche verte pour OK, flèche rouge pour problème).

FixMate offre également la possibilité aux administrateurs de configurer des contrôles, de gérer les droits d'accès, de planifier et d'exécuter des contrôles détaillés et d'envoyer automatiquement des notifications par email aux parties prenantes. De plus, l'application permet

d'imprimer des rapports détaillés des contrôles effectués pour une documentation facile et structurée.

Une des fonctionnalités uniques de FixMate est sa version mobile pour Android, qui inclut les modules essentiels comme le tableau de bord, la gestion des contrôles et la recherche multicritères. Cette version mobile permet aux utilisateurs de continuer à surveiller et à gérer les systèmes d'information même lorsqu'ils sont en déplacement, assurant ainsi une réactivité accrue et une supervision continue.

Pour les administrateurs et les gestionnaires, FixMate propose une interface back office complète qui permet de créer et d'attribuer des profils d'utilisateur, de gérer les modules et sous-modules et de personnaliser les paramètres de contrôle.

FixMate est l'outil idéal pour les entreprises cherchant à optimiser la gestion et la supervision de leurs systèmes d'information, en offrant une solution complète et intuitive pour assurer le bon fonctionnement de leurs infrastructures digitales et atteindre leurs objectifs opérationnels de manière efficace.

3. Étude des exigences

Exigences Fonctionnelles

1. **Gestion des Paramétrages**
    - Les utilisateurs doivent pouvoir configurer les applications, serveurs, équipements et autres ressources.
    - Chaque ressource doit pouvoir être définie avec des champs spécifiques (intitulé, lien, description, etc.).
2. **Recherche Multicritères**
    - L'application doit permettre aux utilisateurs de rechercher des informations en utilisant divers critères tels que les dates, les modules, et autres champs personnalisés.
3. **Gestion des Droits d'Accès**
    - Les administrateurs doivent pouvoir créer des profils d'utilisateur et leur attribuer des fonctionnalités spécifiques.
    - Les administrateurs doivent pouvoir gérer les permissions de création, visualisation, modification et suppression des données.
4. **Tableau de Bord** :

- L'application doit fournir une vue d'ensemble en temps réel des statuts de fonctionnement des différents modules.
- Les utilisateurs doivent pouvoir visualiser rapidement les résultats des derniers contrôles.

5. **Gestion des Contrôles**
   - Les utilisateurs doivent pouvoir planifier, créer et exécuter des contrôles détaillés.
   - Chaque contrôle doit inclure des informations telles que la date, l'heure, l'identité du contrôleur et un champ libre pour les observations.
   - Les états des modules contrôlés doivent être affichés avec des indicateurs visuels (flèche verte pour OK, flèche rouge pour problème).
6. **Envoi des Emails**
   - L'application doit automatiser l'envoi de notifications par email aux parties prenantes concernant les résultats des contrôles.
7. **Impression des États de Sortie**
   - Les utilisateurs doivent pouvoir imprimer des rapports détaillés des contrôles effectués.

8. **Version Mobile**
   - L'application mobile doit inclure les modules tableau de bord, gestion des contrôles et recherche multicritère.
   - Les utilisateurs doivent pouvoir effectuer des contrôles et des recherches même en déplacement.
9. **Interfaces Utilisateur**
   - Le front office doit être accessible aux utilisateurs autorisés pour effectuer des contrôles, des recherches et visualiser le tableau de bord.
   - Le back office doit être réservé aux administrateurs pour la gestion des droits d'accès, des contrôles, des modules et sous-modules, ainsi que l'impression des fiches de contrôle.

Exigences Non Fonctionnelles

1. **Performance**

- L'application doit être capable de gérer simultanément plusieurs utilisateurs sans dégradation significative des performances.
- Les requêtes de recherche multicritères doivent être traitées rapidement pour garantir une expérience utilisateur fluide.

2. **Sécurité**
   - Les données sensibles doivent être protégées par des mécanismes de sécurité robustes, incluant le chiffrement des données en transit et au repos.
   - Les utilisateurs doivent être authentifiés et autorisés de manière sécurisée pour accéder aux différentes fonctionnalités de l'application.
3. **Fiabilité**
   - L'application doit être disponible et fonctionnelle avec un minimum de temps d'arrêt.
   - Des mécanismes de sauvegarde et de récupération des données doivent être en place pour assurer la continuité des opérations en cas de panne.
4. **Scalabilité**
   - L'application doit pouvoir évoluer pour gérer une augmentation du nombre d'utilisateurs et des volumes de données sans nécessiter de modifications majeures.
5. **Compatibilité**
   - L'application web doit être compatible avec les principaux navigateurs web (Chrome, Firefox, Edge, Safari).
   - L'application mobile doit être compatible avec les principales versions d'Android.
6. **Utilisabilité**
   - L'interface utilisateur doit être intuitive et facile à utiliser, avec une navigation claire et cohérente.
   - Des guides et des tutoriels doivent être disponibles pour aider les utilisateurs à comprendre et à utiliser les fonctionnalités de l'application.
7. **Maintenance**
   - L'application doit être conçue de manière modulaire pour faciliter les mises à jour et la maintenance.
   - Des outils de surveillance et de diagnostic doivent être intégrés pour aider les administrateurs à détecter et résoudre les problèmes rapidement.

En combinant ces exigences fonctionnelles et non fonctionnelles, FixMate sera capable de fournir une solution efficace, sécurisée et performante pour la gestion et le contrôle des systèmes d'information, répondant ainsi aux besoins des entreprises modernes.

Conclusion

FixMate offre une solution complète et intégrée pour la gestion et le contrôle des systèmes d'information, répondant à une gamme étendue d'exigences fonctionnelles et non fonctionnelles. Grâce à ses fonctionnalités robustes, incluant la gestion des paramétrages, la recherche multicritères, la gestion des droits d'accès, le tableau de bord interactif, la gestion des contrôles, l'envoi automatisé des emails et l'impression des rapports, FixMate assure une supervision continue et efficace des infrastructures digitales. La version mobile de l'application étend ces capacités, permettant une réactivité accrue et une gestion flexible. En respectant les critères de performance, de sécurité, de fiabilité, de scalabilité, de compatibilité, d'utilisabilité et de maintenance, FixMate se positionne comme un outil indispensable pour les entreprises souhaitant optimiser la performance et la sécurité de leurs systèmes d'information.

Partie II

Planification de gestion de projet

Introduction:

Lors du développement d'un projet logiciel, la planification et la gestion du projet sont des étapes cruciales pour assurer la réussite du projet. Ces étapes permettent de définir les objectifs, les ressources nécessaires, les délais et les responsabilités de chaque membre de l'équipe de développement. Dans ce chapitre, nous allons présenter la planification et la gestion de projet dans le cadre de notre application FixMate. Nous allons décrire les étapes que nous avons suivi pour planifier et gérer notre projet, ainsi que les outils que nous avons utilisé pour faciliter cette tâche.

1. Identification des objectifs de projet

Objectifs Fonctionnels

L'objectif principal de FixMate est de fournir une surveillance en temps réel des systèmes d'information, incluant les applications, les serveurs, les équipements et les infrastructures réseau. Cette application permettra une configuration détaillée et personnalisable des ressources, offrant des champs spécifiques pour chaque type de ressource, ce qui facilitera une gestion précise et adaptée aux besoins de l'entreprise. FixMate intégrera également une fonctionnalité de recherche multicritères avancée, permettant aux utilisateurs de filtrer les informations selon divers critères tels que les dates, les modules et les sous-modules. La gestion des droits d'accès sera un autre aspect clé, permettant de créer et gérer des profils d'utilisateur avec des permissions spécifiques, garantissant ainsi la sécurité et la confidentialité des informations.

Le tableau de bord interactif de FixMate permettra une visualisation en temps réel des statuts de fonctionnement des différents modules et des résultats des contrôles. Les utilisateurs pourront planifier, créer, exécuter et documenter les contrôles réguliers des systèmes d'information, en incluant des informations détaillées sur les dates, les contrôleurs et les observations. L'application automatisera l'envoi de notifications par email pour informer les parties prenantes des résultats des contrôles et des problèmes détectés. De plus, FixMate permettra l'impression de rapports détaillés des contrôles effectués, facilitant ainsi la documentation et la référence future. Une version mobile de l'application sera développée pour Android, incluant les modules essentiels tels que le tableau de bord, la gestion des contrôles et la recherche multicritères, permettant aux utilisateurs de surveiller et gérer les systèmes d'information même en déplacement.

Objectifs de Délai

Le respect des délais de développement et de déploiement fixés par le plan de projet est crucial pour FixMate. Le projet prévoira des livraisons intermédiaires à des étapes clés, permettant des tests et des ajustements avant la version finale, assurant ainsi que chaque phase du développement est conforme aux attentes et aux exigences. Des mises à jour régulières seront également planifiées et exécutées pour améliorer les fonctionnalités de l'application et corriger les bugs identifiés, garantissant une évolution continue de la qualité et de la performance de FixMate.

Objectifs de Qualité

FixMate vise à assurer une fiabilité et une disponibilité maximales, en mettant en place des mécanismes de sauvegarde et de récupération des données pour minimiser le temps d'arrêt. La sécurité sera renforcée par l'implémentation de mesures robustes, incluant le chiffrement des données en transit et au repos, protégeant ainsi les informations sensibles contre les accès non autorisés. L'application garantira des performances élevées, permettant aux requêtes de recherche multicritères et aux opérations de contrôle de s'exécuter rapidement même avec un nombre élevé d'utilisateurs simultanés. L'interface utilisateur sera conçue pour être intuitive et facile à utiliser, avec une navigation claire et cohérente, assurant une expérience utilisateur fluide et satisfaisante. En outre, FixMate sera conçu pour être scalable, pouvant évoluer facilement en réponse à une augmentation du nombre d'utilisateurs et des volumes de données sans nécessiter de modifications majeures. Enfin, toutes les activités et modifications seront enregistrées pour assurer une traçabilité complète, facilitant ainsi les audits de sécurité et garantissant la conformité aux normes et réglementations en vigueur.

2. Méthodologie de gestion de projet

Dans le cadre du développement de FixMate, nous avons choisi d'adopter la méthodologie Agile pour la gestion de projet. La méthodologie Agile est une approche itérative et collaborative qui se concentre sur la livraison rapide de fonctionnalités fonctionnelles tout en s'adaptant aux changements et aux feedbacks des utilisateurs tout au long du processus de développement. Le choix de cette méthodologie s'explique par plusieurs facteurs. Tout d'abord, étant donné que notre équipe est composée de deux personnes et que nous avons un temps limité pour le développement du projet, nous avons opté pour une méthode de travail flexible et adaptative qui nous permettrait de livrer rapidement des fonctionnalités fonctionnelles et d'itérer

sur celles-ci tout au long du processus de développement. Et en adoptant cette méthodologie, nous avons également pu bénéficier d'une meilleure communication et collaboration entre les membres de l'équipe, donc cette méthode met l'accent sur la communication continue entre les membres de l'équipe et sur la collaboration pour résoudre les problèmes qui peuvent survenir tout au long du processus de développement.

3. Planification de projet

Dans le cadre du projet FixMate, nous avons élaboré un plan de projet détaillé qui comprend toutes les tâches nécessaires pour la conception, le développement et la mise en production de l'application. Le plan de projet est divisé en plusieurs étapes, avec des délais et des livrables spécifiques pour chaque étape. Pour FixMate, nous avons choisi de travailler avec des sprints de 2 semaines. Au début de chaque sprint, nous avons défini les objectifs à atteindre, en tenant compte des priorités des exigences et des délais fixés pour le projet. Nous avons ensuite identifié les tâches nécessaires pour atteindre ces objectifs et les avons assigné à chaque membre de l'équipe en fonction de leurs compétences. Pendant chaque sprint, nous avons tenu des réunions quotidiennes de suivi de l'avancement. Ces réunions nous ont permis de discuter de l'avancement des tâches, des problèmes rencontrés et des solutions proposées. Nous avons également organisé une réunion de revue de sprint à la fin de chaque période de 2 semaines pour évaluer les résultats obtenus et planifier les prochaines étapes.

TABLEAU DE PLANNIFICATION DE PROJET

Sprint	Durée	Objectifs	Tâches	Livrables
1	1 semaine	Définir les objectifs de l'application	- Effectuer une étude de marché pour comprendre les besoins des utilisateurs et les lacunes du marché. - Établir les fonctionnalités principales de l'application en se basant sur l'étude de marché. - Élaborer une liste complète des exigences fonctionnelles et non fonctionnelles de l'application.	- Document des fonctionnalités de l'application. - Document des exigences de l'application.
2	1 semaine	Sélection de la méthode de travail et Planification des tâches.	- Comparaison des méthodes et choix de la méthode de travail convenable. - Définition des tâches à réaliser - Estimation de la durée de chaque tâche	- Décision de la méthode de travail. - Liste des tâches avec l'estimation de la durée de chacune.
3	1 semaine	Modélisation de l'application	- Réaliser un diagramme de cas d'utilisation pour l'application. - Concevoir un diagramme de classes pour représenter les entités et les relations dans l'application. - Créer un diagramme de navigation pour savoir comment naviguer dans l'application	- Diagramme de cas d'utilisation. - Diagramme de classes. - Diagrammes de navigation.
4	4 semaines	Développement de l'application web	Mise en place du backend avec Laravel - Intégration de MySQL pour la gestion des données	Application web fonctionnelle avec backend et base de données
5	4 semaines	Développement de l'application mobile	- Développement des fonctionnalités mobiles en React Native - Intégration avec les services web	Applications web et mobiles testées et validées

6	1 semaine	Tests et validation	- Réalisation de tests unitaires et d'intégration - Tests utilisateurs initiaux sur les interfaces	Applications web et mobiles prêtes pour le déploiement
7	1 semaine	Finalisation et déploiement	- Correction des bugs et optimisation des performances - Préparation pour le déploiement final	Applications web et mobiles prêtes pour le déploiement

4. Diagramme de Gantt :

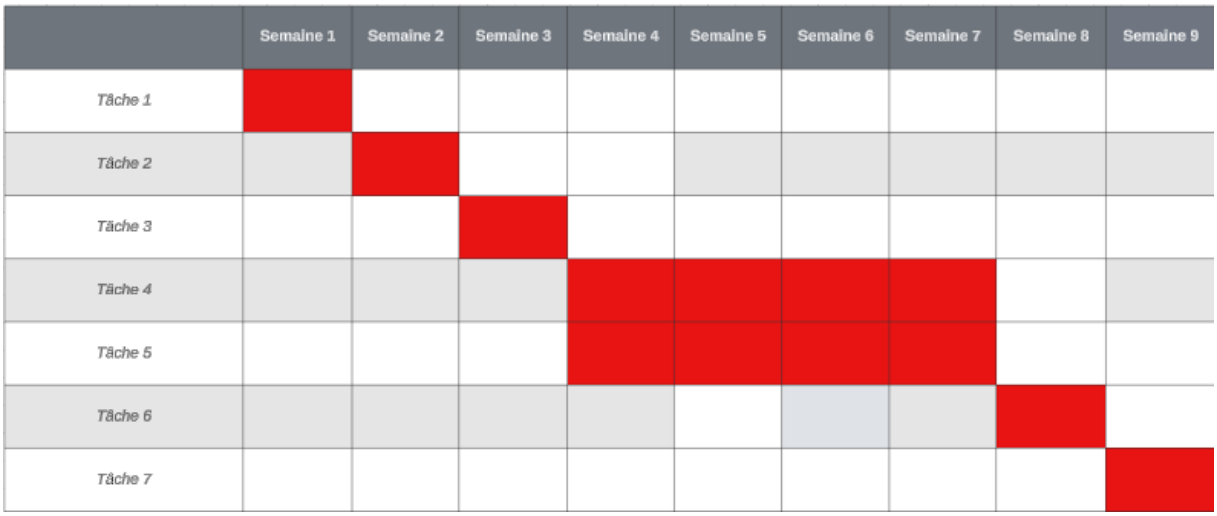


Figure 2 : Diagramme de Gantt

5. Diagramme de navigation

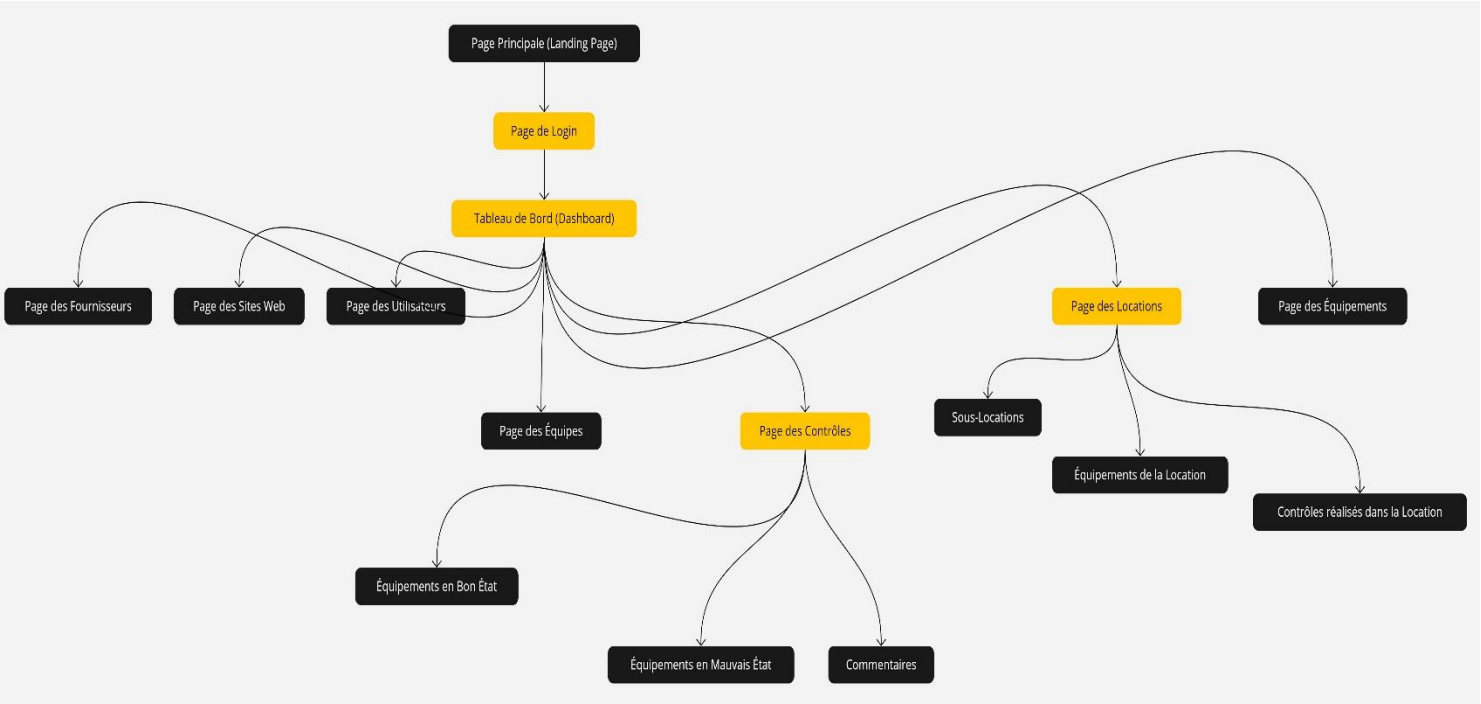


Figure 3 : Diagramme de navigation

6. Gestion des risques

Risques identifiés

1. **Retards de Développement**
   - Les retards dans le développement des fonctionnalités peuvent survenir en raison de problèmes techniques imprévus, de la complexité du projet ou de la sous-estimation des efforts nécessaires.
2. **Problèmes de Performance**
   - L'application pourrait rencontrer des problèmes de performance, tels que des temps de réponse lents ou des défaillances sous une forte charge affectant l'expérience utilisateur.
3. **Sécurité des Données**
   - Des vulnérabilités de sécurité pourraient être exploitées, compromettant les données sensibles des utilisateurs et de l'entreprise.

4. **Non-Conformité aux Exigences**
    - L'application pourrait ne pas répondre entièrement aux exigences fonctionnelles et non fonctionnelles spécifiées, entraînant une insatisfaction des utilisateurs et des clients.
5. **Problèmes d'Intégration**
    - Des difficultés pourraient survenir lors de l'intégration de FixMate avec les systèmes existants de l'entreprise, retardant le déploiement ou causant des dysfonctionnements.
6. **Risques de Scalabilité**
    - L'application pourrait ne pas évoluer efficacement avec l'augmentation du nombre d'utilisateurs et de données, limitant sa capacité à répondre aux besoins futurs.
7. **Gestion des Changements**
    - Les modifications des exigences ou des priorités en cours de développement peuvent perturber le projet et provoquer des retards ou des dépassements de budget.

Stratégies de Mitigation Prévues

1. **Gestion Rigoureuse du Projet**
    - Mettre en place une gestion de projet rigoureuse avec des plans de projet détaillés, des jalons clairs et des échéances réalistes pour suivre l'avancement et identifier les retards potentiels tôt.
    - Utiliser des méthodologies agiles pour permettre des ajustements rapides et une meilleure gestion des ressources.
2. **Tests de Performance et d'Endurance**
    - Effectuer des tests de performance et d'endurance réguliers pour identifier et résoudre les problèmes de performance avant le déploiement. Optimiser le code et l'infrastructure pour assurer des temps de réponse rapides et une gestion efficace de la charge.
3. **Sécurité Renforcée**
    - Implémenter des mesures de sécurité robustes dès le début du développement, incluant le chiffrement des données, des audits de sécurité réguliers et des tests de pénétration pour identifier et corriger les vulnérabilités.

4. **Validation Continue des Exigences**
    - Maintenir une communication continue avec les parties prenantes pour valider régulièrement que les développements en cours répondent aux exigences spécifiées.
    - Utiliser des prototypes et des démonstrations pour recueillir des retours et ajuster les fonctionnalités selon les besoins.
5. **Plan d'Intégration Structuré**
    - Développer un plan d'intégration structuré avec des tests d'intégration réguliers pour assurer la compatibilité avec les systèmes existants.
    - Collaborer étroitement avec les équipes techniques de l'entreprise pour faciliter une intégration en douceur.
6. **Architecture Scalable**
    - Concevoir l'application avec une architecture scalable, capable de gérer l'augmentation du nombre d'utilisateurs et des volumes de données.
7. **Gestion du Changement**
    - Mettre en place un processus de gestion des changements formel pour évaluer, approuver et intégrer les modifications des exigences ou des priorités de manière structurée, minimisant ainsi les perturbations.

Conclusion

La planification et la gestion de projet sont des étapes essentielles dans le développement de tout projet logiciel. Dans le cas de FixMate, nous avons identifié des objectifs clairs, choisi une méthodologie Agile pour la gestion de projet et élaboré un plan de projet détaillé. Nous avons également identifié des risques potentiels et prévu des stratégies de mitigation pour minimiser leur impact sur le projet. La mise en place de ces étapes nous a permis de développer une application fonctionnelle et de qualité, répondant aux besoins des utilisateurs et des coachs professionnels. Cela a également permis de respecter les délais de livraison fixés pour le projet.

Partie III

Conception de l'Application FixMate

Introduction:

Dans ce chapitre, nous allons aborder la conception et la réalisation de l'application FixMate. Nous allons détailler les différentes étapes qui ont permis de mettre en place cette application, en nous concentrant sur les outils utilisés pour la conception, ainsi que sur le Framework utilisé pour la réalisation et la mise en œuvre de l'application. Nous aborderons également les fonctionnalités mises en place pour tous les utilisateurs de l'application. Enfin, nous évaluerons les résultats obtenus, en analysant les performances et l'utilisabilité du système, ainsi que les problèmes rencontrés et les solutions apportées.

I. Conception de FixMate

1. Outils de conception et modélisation

Dans cette partie, nous allons aborder les différents outils que nous avons utilisé pour la conception de notre application FixMate :

UML :

Une grande partie dans notre conception, nous avons utilisé les diagrammes d'UML, alors l'UML (Unified Modeling Language) est un langage de modélisation visuelle utilisé pour représenter graphiquement des systèmes logiciels. Il fournit un ensemble de diagrammes et de notations qui permettent aux développeurs de communiquer efficacement les différentes composantes du système, leurs interactions, leurs comportements et leurs contraintes. UML est largement utilisé dans l'industrie du développement logiciel pour concevoir, documenter et tester les systèmes informatiques.



Figure 4: Logo de StarUML

Nous avons utilisé StarUML pour la création des diagrammes de classes, de cas d'utilisation et des activités. Ce logiciel de modélisation UML nous a permis de représenter de

manière visuelle l'architecture de notre application et de mieux appréhender les différentes fonctionnalités.

2. Site-Map

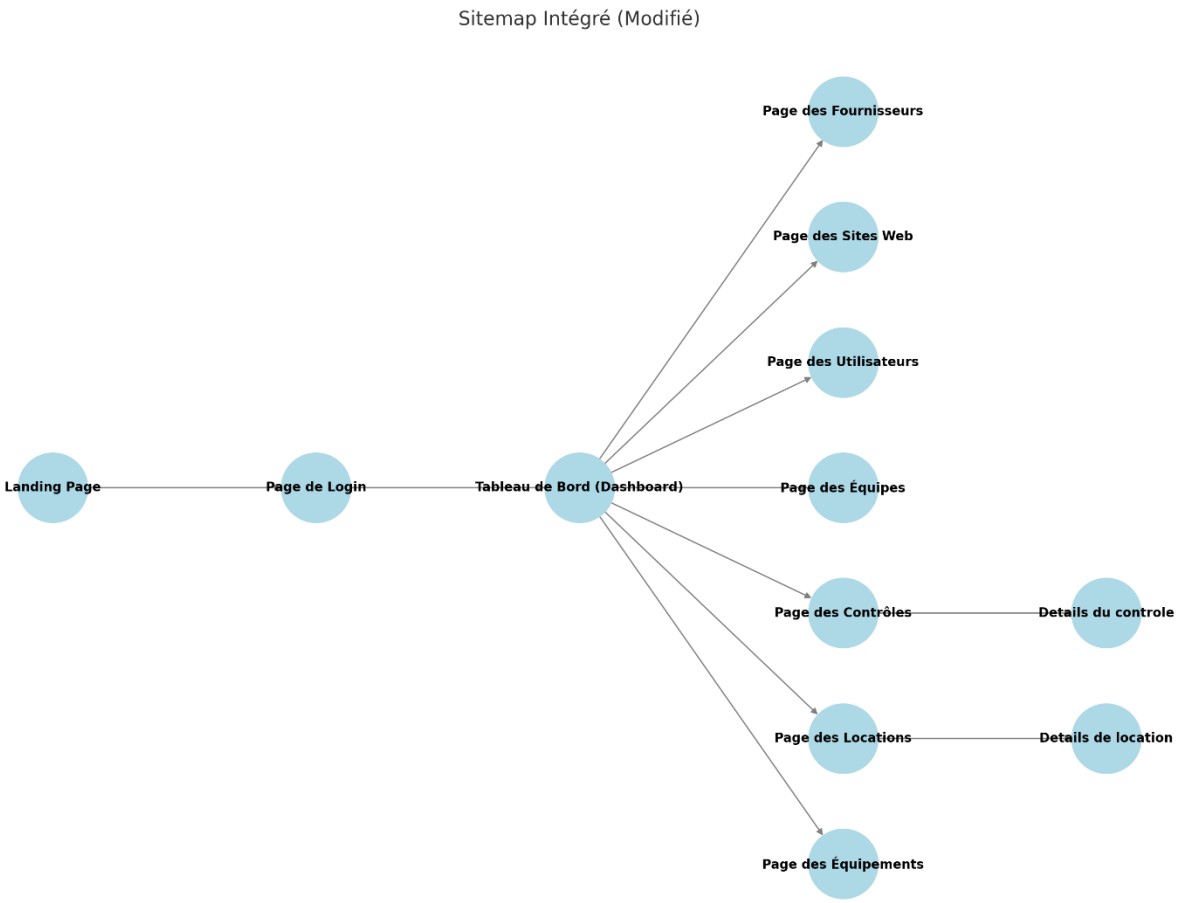


Figure 5 : Site-Map

3. Diagramme de cas d'utilisation

3.1. Définition

Un diagramme de cas d'utilisation est un diagramme qui permet de représenter graphiquement les interactions entre les acteurs et le système et il représente les différents cas

d'utilisation (ou scénarios d'utilisation) d'un système, ainsi que les acteurs (utilisateurs, systèmes externes ou autres entités qui interagissent avec le système).

3.2. Diagramme de cas d'utilisation d'un Admin

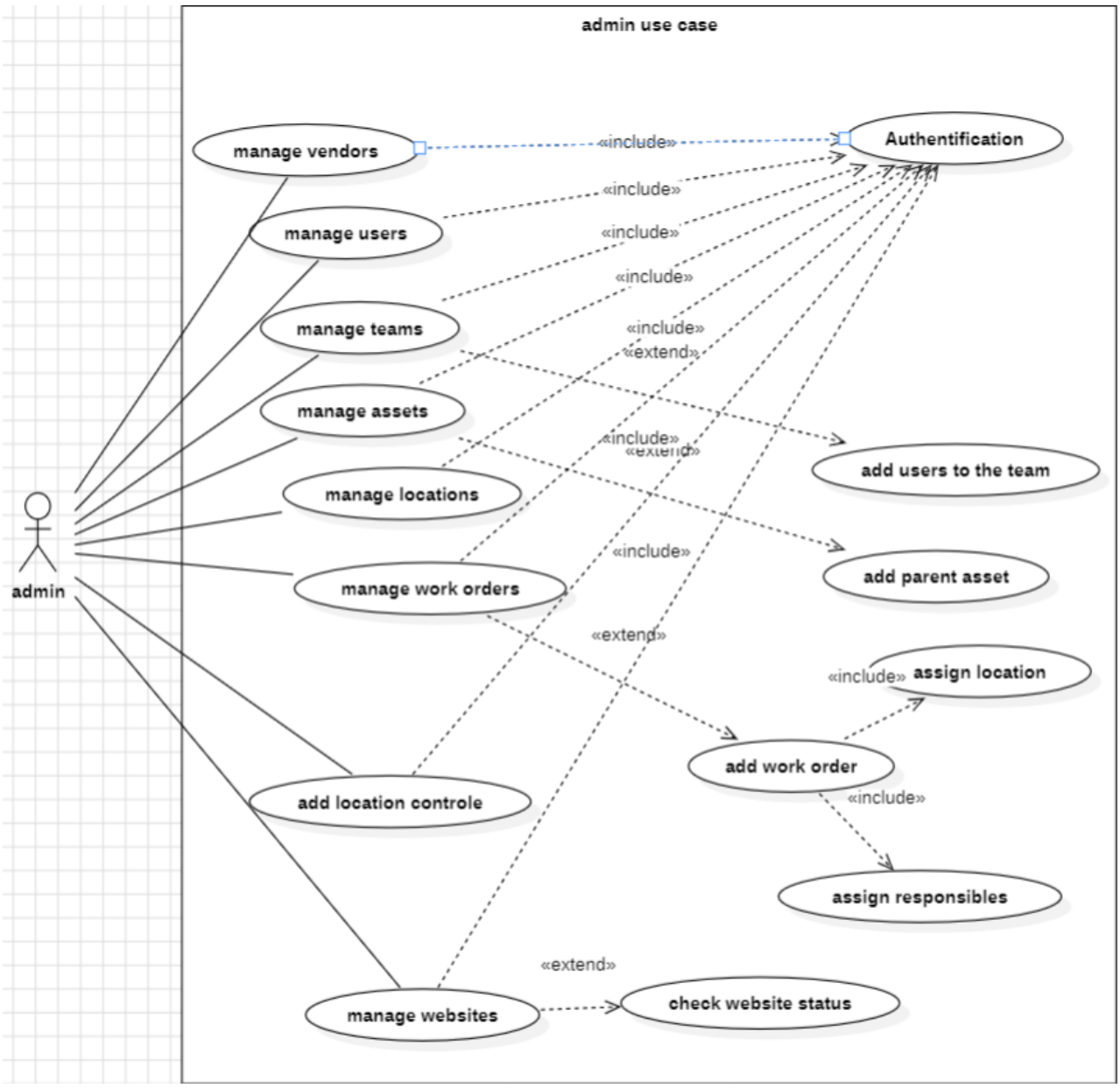


Figure 6: Diagramme de cas d'utilisation d'un Admin

3.3. Diagramme de cas d'utilisation d'un Manager

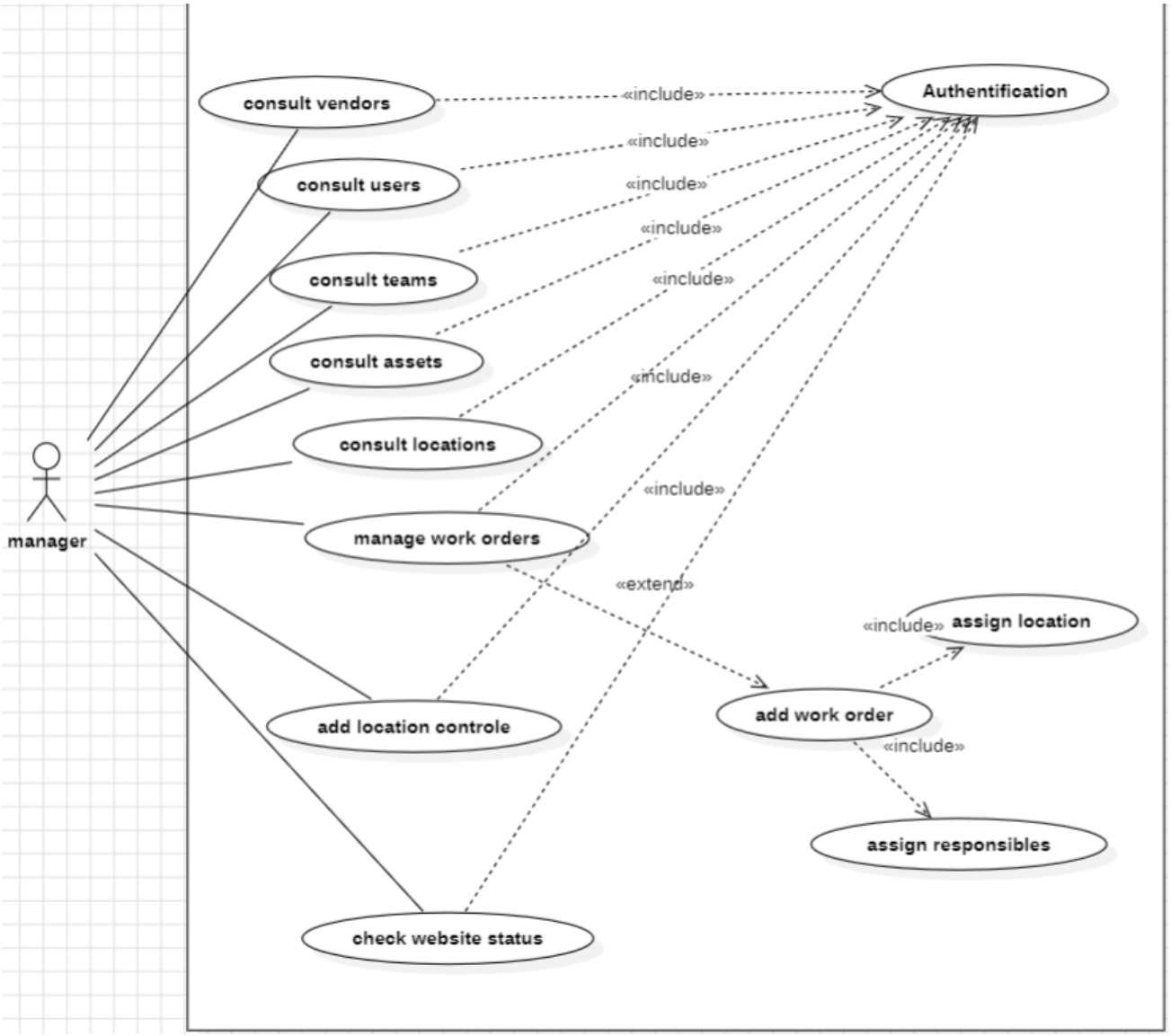


Figure 7: Diagramme de cas d'utilisation d'un Manager

3.4. Diagramme de cas d'utilisation d'un utilisateur

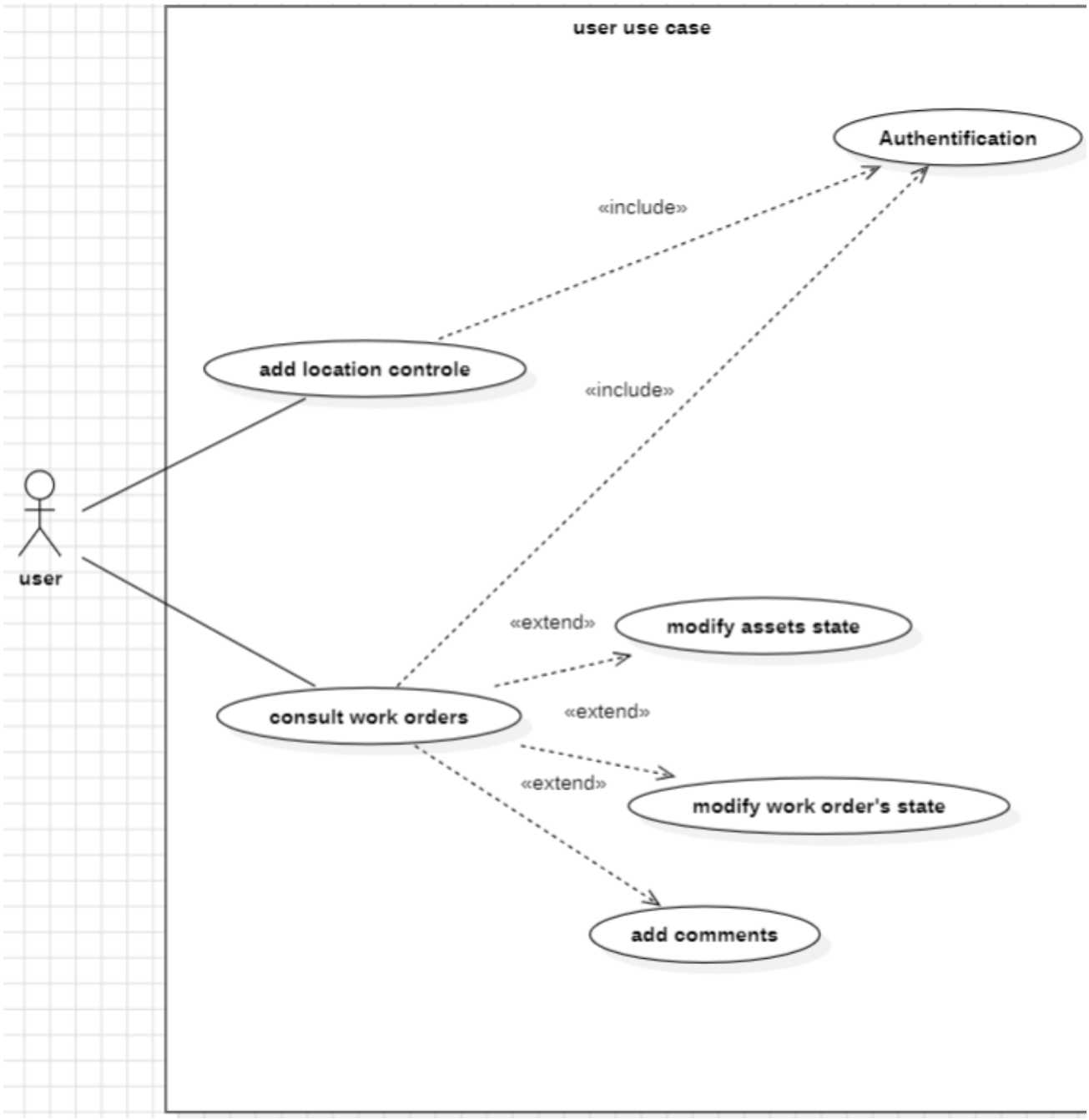


Figure 8: Diagramme de cas d'utilisation d'un utilisateur

4. Diagramme de class

4.1. Définition

Un diagramme de classes est un type de diagramme UML qui représente la structure et les relations entre les classes d'un système.

Il est utilisé pour visualiser les différents objets du système, les attributs qu'ils possèdent, ainsi que les méthodes et les relations entre eux.

Les classes peuvent être représentées sous forme de rectangles contenant le nom de la classe, ses attributs et ses méthodes.

Les relations entre les classes peuvent être représentées par des flèches et des annotations pour indiquer la nature de la relation (agrégation, composition, association, héritage, etc.).

Le diagramme de classes est souvent utilisé lors de la phase de conception d'un système pour représenter les objets qui seront implémentés dans le code source.

4.2. Diagramme de classes de projet

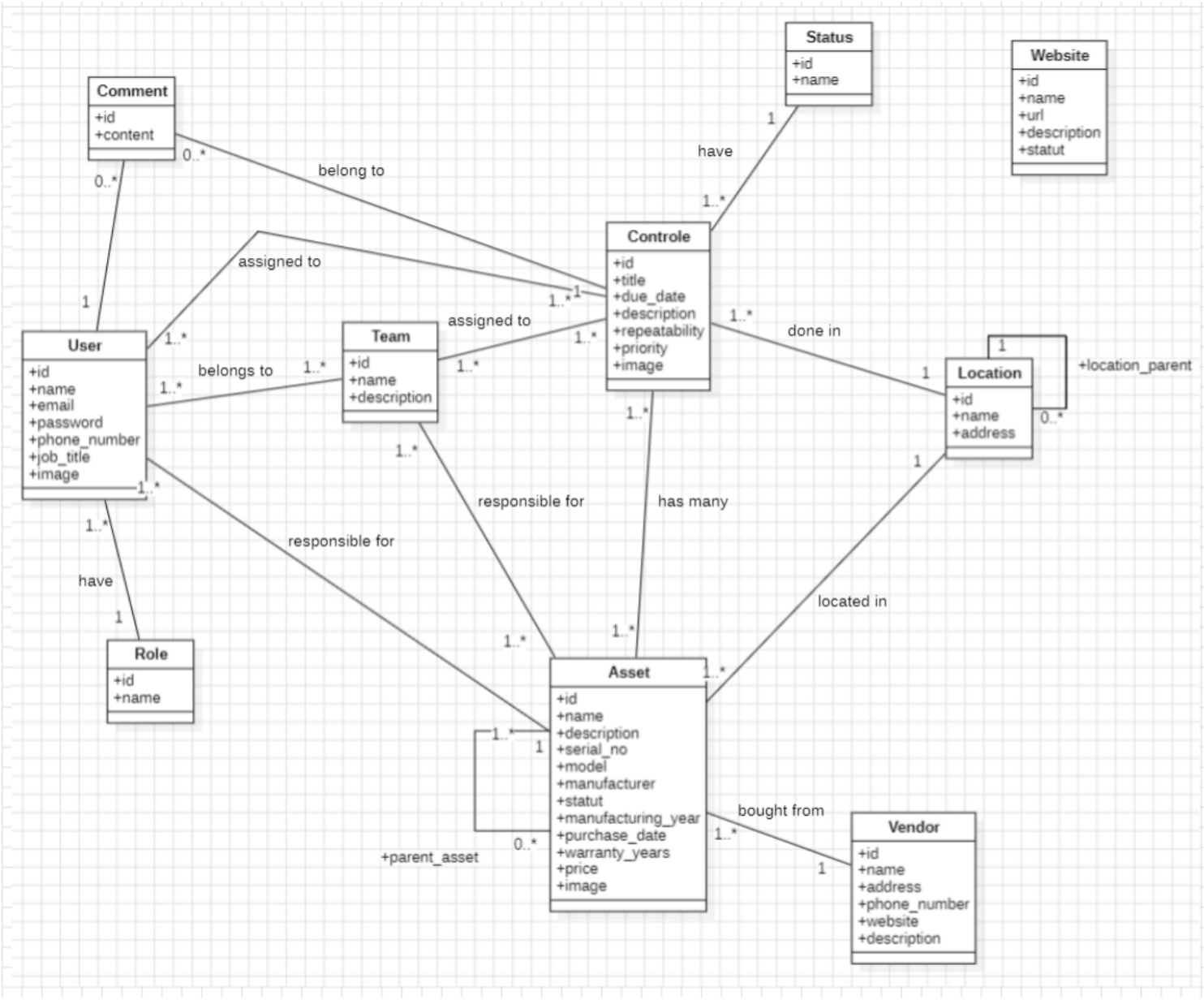


Figure 9: Diagramme de classes de projet

Partie IV

Réalisation de l'Application FixMate

1. Outils et technologies utilisés

1.1. Back-End

1.1.1. PHP (Hypertext Preprocessor)

PHP est un langage de script côté serveur très populaire et largement utilisé dans le développement web. Dans notre projet, nous avons utilisé PHP pour le développement de l'application web. Voici quelques points importants à noter :

- Langage de script côté serveur : PHP est un langage de script spécialement conçu pour le développement web côté serveur. Il permet de créer des pages web dynamiques en générant du contenu HTML à partir de scripts PHP exécutés sur le serveur.
- Large compatibilité : PHP est compatible avec de nombreux serveurs web et systèmes d'exploitation, ce qui facilite son déploiement sur différentes plateformes. Il peut être utilisé avec des serveurs web tels qu'Apache, Nginx, IIS, etc.
- Intégration avec les bases de données : PHP offre une intégration facile avec différents systèmes de gestion de bases de données tels que MySQL, SQL Server, Oracle, etc. Cela permet de stocker et de récupérer des données à partir de la base de données, ce qui est essentiel pour les applications web qui nécessitent une interaction avec une base de données.
- Prise en charge de nombreuses fonctionnalités : PHP dispose d'une large gamme de fonctionnalités intégrées, telles que le traitement des formulaires, la manipulation des fichiers, l'envoi d'e-mails, la gestion des sessions, etc. Ces fonctionnalités facilitent le développement d'applications web complexes et interactives.
- Communauté active : PHP bénéficie d'une communauté de développeurs très active et de nombreuses ressources en ligne. Cela permet d'accéder à une vaste bibliothèque de scripts, de Framework et de plugins, ce qui accélère le développement et facilite la résolution des problèmes.

En utilisant PHP comme langage de programmation principal dans notre projet, nous avons pu créer une application web robuste, dynamique et interconnectée. PHP nous a offert la flexibilité nécessaire pour développer des fonctionnalités avancées, interagir avec des bases de données et fournir une expérience utilisateur optimale.

1.1.2. MySQL

MySQL est un système de gestion de base de données relationnelle open source largement utilisée. Voici quelques caractéristiques clés de MySQL :



Figure 10 : Logo du langage MySql

- Fiabilité et stabilité : MySQL est réputé pour sa fiabilité et sa stabilité, ce qui en fait un choix populaire pour les applications web et les systèmes d'information.
- Performance élevée : MySQL offre de bonnes performances, même avec des charges de travail importantes et une grande quantité de données.
- Prise en charge des langages de programmation : MySQL est compatible avec un large éventail de langages de programmation, ce qui facilite l'intégration avec les applications développées dans différents environnements.
- Facilité d'utilisation : MySQL est apprécié pour sa facilité d'installation, de configuration et d'utilisation, ce qui permet aux développeurs de se concentrer davantage sur la logique métier de leurs applications.
- Communauté active : MySQL bénéficie d'une communauté active d'utilisateurs et de développeurs, offrant un support technique et de nombreuses ressources en ligne.

Dans notre projet, nous avons utilisé MySQL comme système de gestion de base de données pour stocker et gérer les informations. Nous avons exploité les fonctionnalités de MySQL, telles que la création de tables, les relations entre les entités et les requêtes SQL, pour implémenter la logique de notre application.

L'utilisation de MySQL nous a permis de bénéficier d'un système de gestion de base de données fiable et performant lors de la phase initiale de notre projet.

1.1.3. SMTP (Simple Mail Transfer Protocol)

SMTP est un protocole largement utilisé pour l'envoi de messages électroniques sur Internet. Dans notre application web, nous avons intégré un service SMTP pour faciliter l'envoi d'e-mails aux utilisateurs.

- Livraison fiable des e-mails : En utilisant un service SMTP, nous nous assurons que les messages électroniques sont livrés de manière fiable aux serveurs de messagerie des destinataires. Cela permet de réduire les risques que les e-mails soient considérés comme du spam ou ne parviennent pas aux destinataires prévus.
- Authentification et sécurité : Les services SMTP fournissent généralement des mécanismes d'authentification, tels que nom d'utilisateur et mot de passe, pour garantir un accès sécurisé au serveur

de messagerie. Cela permet de prévenir les accès non autorisés et de protéger la confidentialité des informations des utilisateurs.

- Personnalisation du contenu des e-mails : Avec SMTP, nous pouvons personnaliser le contenu des messages électroniques, y compris le sujet, le corps et les pièces jointes. Cela nous permet de personnaliser les e-mails et de fournir des informations pertinentes aux utilisateurs.
- Intégration avec notre application web : Nous avons intégré le service SMTP dans notre application web, ce qui permet l'envoi transparent d'e-mails à partir de différentes fonctionnalités. Cela inclut l'envoi d'e-mails d'activation de compte, de réinitialisation de mot de passe et de notifications aux utilisateurs.
- Surveillance et suivi : De nombreux services SMTP offrent des fonctionnalités de surveillance et de suivi, nous permettant de suivre l'état de livraison des e-mails envoyés et de suivre les taux d'ouverture ou de clics. Cela nous aide à évaluer l'efficacité de nos communications par e-mail.

1.1.4. Laravel

a. Définition



Figure 11 : Logo du framework Laravel

Laravel est un framework PHP open-source, robuste et facile à comprendre. Il suit un modèle de conception MVC (Modèle-Vue-Contrôleur). Laravel réutilise les composants existants de différents frameworks, ce qui aide à créer une application web. L'application web ainsi conçue est plus structurée et pragmatique. Laravel offre un ensemble complet de fonctionnalités qui incorpore les fonctionnalités de base des frameworks PHP tels que CodeIgniter, Yii et d'autres langages de programmation tels que Ruby on Rails. Laravel possède un ensemble très complet de fonctionnalités qui accélèrent le développement web tel que :

Architecture MVC : Laravel suit l'architecture Modèle-Vue-Contrôleur (MVC), qui permet de séparer les préoccupations de la logique de l'application, de l'interface utilisateur et de la gestion des données. Cela rend le code plus facile à maintenir et à étendre.

Facilité de développement : Laravel est conçu pour faciliter le développement. Il offre une syntaxe concise et expressive, des fonctionnalités de migration de base de données, des modèles Eloquent pour interagir avec la base de données et une API de routage simple à utiliser.

Sécurité : Laravel intègre des fonctionnalités de sécurité de base pour aider à protéger votre application contre les attaques de sécurité courantes. Il offre également une fonctionnalité de hachage de mot de passe par défaut, ce qui signifie que les mots de passe des utilisateurs sont stockés de manière sécurisée.

Communauté active : Laravel est soutenu par une grande communauté de développeurs actifs qui contribuent régulièrement au framework. Cette communauté fournit une documentation complète, des didacticiels, des packages et des forums d'aide.

Artisan : Laravel offre un outil de ligne de commande appelé Artisan, qui facilite les tâches courantes de développement. Il peut être utilisé pour créer des modèles, des migrations de base de données, des contrôleurs et bien plus encore.

Intégration facile : Laravel peut être facilement intégré avec d'autres bibliothèques et services tiers, tels que AWS, Stripe, PayPal, Mandrill et bien d'autres encore. Cela facilite l'ajout de fonctionnalités avancées à votre application.

Tests automatisés : Laravel offre un système de test automatisé intégré qui permet de tester rapidement et facilement votre code pour détecter les erreurs et les bogues.

b. Architecture de Laravel :

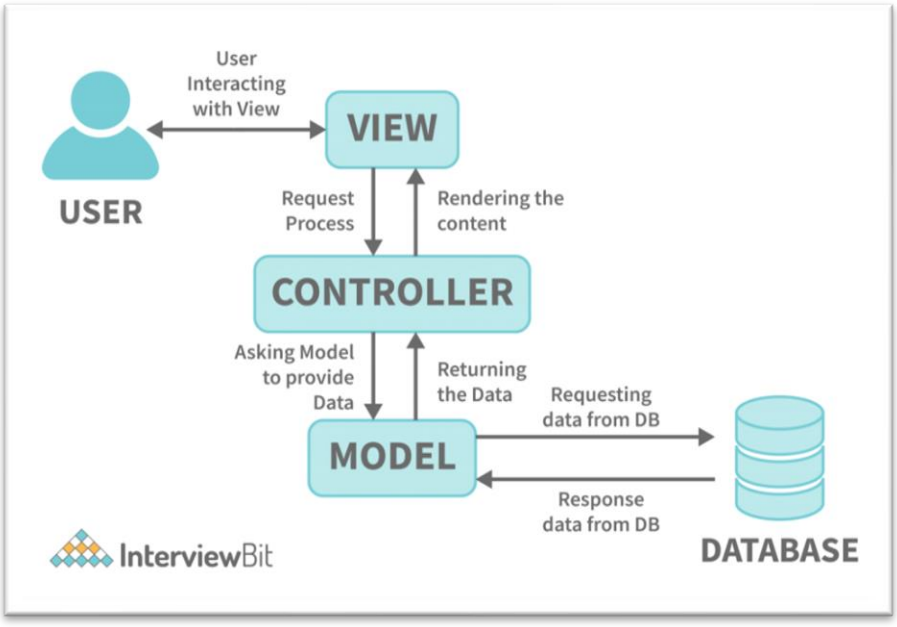


Figure 12: Architecture MVC

Laravel suit l'architecture MVC (Modèle-Vue-Contrôleur) qui est une approche populaire pour le développement d'applications Web. Voici un aperçu de chaque composant de l'architecture :

Modèle : Le modèle représente la couche de données de l'application. Il est responsable de la récupération, de la manipulation et de l'enregistrement des données. Dans Laravel, la couche de modèle est généralement implémentée à l'aide du ORM (Object-Relational Mapping) Eloquent.

Vue : La vue représente la couche de présentation de l'application. Elle est responsable de l'affichage des données à l'utilisateur final. Dans Laravel, les vues sont généralement écrites en utilisant le langage de modèle Blade.

Contrôleur : Le contrôleur représente la couche de logique de l'application. Il est responsable de la gestion des requêtes HTTP entrantes, de la récupération des données nécessaires à partir de la couche de modèle et de la génération de la réponse HTTP appropriée en utilisant la vue.

En utilisant l'architecture MVC, les différents composants de l'application sont séparés les uns des autres, ce qui facilite la maintenance et l'extension du code. Par exemple, si vous devez apporter des modifications à la logique de l'application, vous pouvez modifier le contrôleur sans toucher au modèle ou à la vue. De même, si vous devez modifier l'apparence de l'application, vous pouvez modifier la vue sans modifier le modèle ou le contrôleur.

Laravel offre également des fonctionnalités supplémentaires pour faciliter le développement d'applications Web, telles que le routage, la migration de base de données, la gestion des sessions, la validation des données, la sécurité et bien plus encore. Toutes ces fonctionnalités sont conçues pour fonctionner harmonieusement avec l'architecture MVC de Laravel, afin de simplifier le développement d'applications Web PHP.

c. La structure d’une application Laravel

La structure d'une application Laravel est organisée de manière logique et cohérente, ce qui facilite la gestion des différents aspects de l'application.

La capture d'écran montrée ici se réfère au dossier racine de Laravel nommé "laravel-project". Il comprend divers sous-dossiers et fichiers.



Voici une description des principaux dossiers et fichiers de la structure d'une application Laravel :

Dossier "app" : Le dossier "app" contient les fichiers de l'application, tels que les modèles, les contrôleurs, les vues et les fichiers de migration de base de données. Les sous-dossiers de "app" comprennent également "Http" qui contient les fichiers de routage, les fichiers de middleware et les contrôleurs de l'application, et "Providers" qui contient les fichiers de fournisseurs de services pour l'application.

Dossier "bootstrap" : Le dossier "bootstrap" contient les fichiers qui permettent à l'application de démarrer, tels que le fichier "app.php" qui charge les dépendances de l'application.

Dossier "config" : Le dossier "config" contient les fichiers de configuration pour l'application, tels que les fichiers de configuration de la base de données, de l'authentification et de la session.

Figure 12: Structure d'une application Laravel

Dossier "database" : Le dossier "database" contient les fichiers de migration de base de données, les fichiers de semences (pour ajouter des données initiales à la base de données) et le dossier "factories" (pour générer des données de test).

Dossier "public" : Le dossier "public" contient les fichiers qui sont accessibles au public, tels que les fichiers CSS, JavaScript et les fichiers d'images. Le fichier "index.php" dans ce dossier est le point d'entrée de l'application.

Dossier "resources" : Le dossier "resources" contient les fichiers qui sont utilisés pour construire l'application, tels que les fichiers de vues, les fichiers de langage et les fichiers de compilation (pour compiler les fichiers CSS et JavaScript).

Dossier "routes" : Le dossier "routes" contient les fichiers de routage de l'application. Les fichiers de routage définissent les points d'entrée de l'application et les actions associées à ces points d'entrée.

Dossier "storage" : Le dossier "storage" contient les fichiers générés par l'application, tels que les fichiers journaux, les fichiers de cache et les fichiers de session.

Fichier ".env" : Le fichier ".env" contient les variables d'environnement de l'application, telles que les informations de connexion à la base de données.

Dans l'ensemble, la structure d'une application Laravel est bien organisée et facilite la gestion de différents aspects de l'application. Elle permet également de se concentrer sur la logique de l'application et de ne pas avoir à se soucier des détails de la structure du projet.

Pourquoi on a choisi de travailler avec Laravel ?

Nous avons choisi d'utiliser Laravel pour développer FitMate en raison de ses nombreux avantages et fonctionnalités. Tout d'abord, Laravel est un framework PHP open-source et robuste qui suit le modèle de conception MVC, ce qui le rend facile à comprendre et à utiliser. Il offre également une grande richesse de fonctionnalités, telles que la modularité, la testabilité, la gestion de configuration, le générateur de requêtes, l'ORM Eloquent, le moteur de templates Blade, l'authentification utilisateur, la gestion des files d'attente et bien plus encore. De plus, Laravel est très sécurisé et prévient plusieurs attaques Web. En utilisant Laravel, nous avons pu développer rapidement une application Web structurée et pragmatique tout en assurant la sécurité de nos utilisateurs.

1.2. Architecture MVC

L'architecture MVC est un modèle de conception largement utilisé dans le développement web pour séparer les préoccupations et organiser le code de manière logique et modulaire. Dans notre projet, nous avons adopté cette architecture pour le back-end de notre application. Voici un aperçu du rôle de chaque élément :

- **Le modèle (Model) :** représente les données de l'application et la logique métier associée. Il est responsable de la récupération, de la manipulation et de la persistance des données. Dans notre cas, les modèles sont utilisés pour représenter les entités. ls interagissent directement avec la base de données et fournissent les fonctionnalités de manipulation des données.
- **La vue (View) :** est responsable de l'affichage des données à l'utilisateur. Elle présente les informations de manière conviviale et réactive. Dans notre projet, les vues sont utilisées pour générer les pages web et les réponses API, en utilisant des Template ou des composants réutilisables. Elles reflètent l'état actuel des données et permettent aux utilisateurs d'interagir avec l'application.
- **Le contrôleur (Controller) :** agit comme un intermédiaire entre le modèle et la vue. Il reçoit les requêtes de l'utilisateur, déclenche les actions appropriées sur les modèles, et renvoie les résultats à la vue pour l'affichage. Dans notre projet, les contrôleurs sont responsables de la logique de traitement des requêtes, de la validation des données, de l'appel aux services et de la gestion des réponses.

L'architecture MVC offre plusieurs avantages, tels que la séparation claire des responsabilités, la facilité de maintenance et de test, ainsi que la réutilisabilité du code.

1.3. Utilisation des API

Dans notre projet, nous avons utilisé des **API (Application Programming Interfaces)** pour faciliter la communication entre les différentes parties de notre application, notamment entre le back-end et les applications web et mobiles. Les API nous ont permis d'établir des points d'interaction standardisés pour l'échange de données.

Conception et développement des API

Nous avons conçu et développé des API RESTful (Representational State Transfer) en utilisant Laravel. Les API RESTful fournissent une approche simple et cohérente pour exposer les fonctionnalités de notre back-end sous la forme de services web.

Endpoints et méthodes HTTP

Nous avons défini différents endpoints pour chaque fonctionnalité exposée par nos API. Chaque endpoint correspond à une action spécifique, comme la création d'un utilisateur, la récupération des équipements d’une location, etc. Nous avons utilisé les méthodes HTTP appropriées, telles que POST, GET, PUT, DELETE, pour interagir avec ces endpoints et effectuer les opérations correspondantes.

Formats de données

Les API utilisent généralement le format JSON (JavaScript Object Notation) pour l'échange de données. Nous avons donc utilisé JSON comme format de données pour les requêtes et les réponses des API. Cela permet une compatibilité et une lisibilité élevées, et facilite l'intégration avec les applications front-end.

Authentification et sécurité

Pour garantir la sécurité des API, nous avons mis en place des mécanismes d'authentification et d'autorisation. Dans notre cas, nous avons utilisé des tokens d'authentification basés sur le protocole JWT (JSON Web Tokens) pour sécuriser l'accès aux API. Cela permet de vérifier l'identité des utilisateurs et de limiter l'accès aux ressources sensibles.

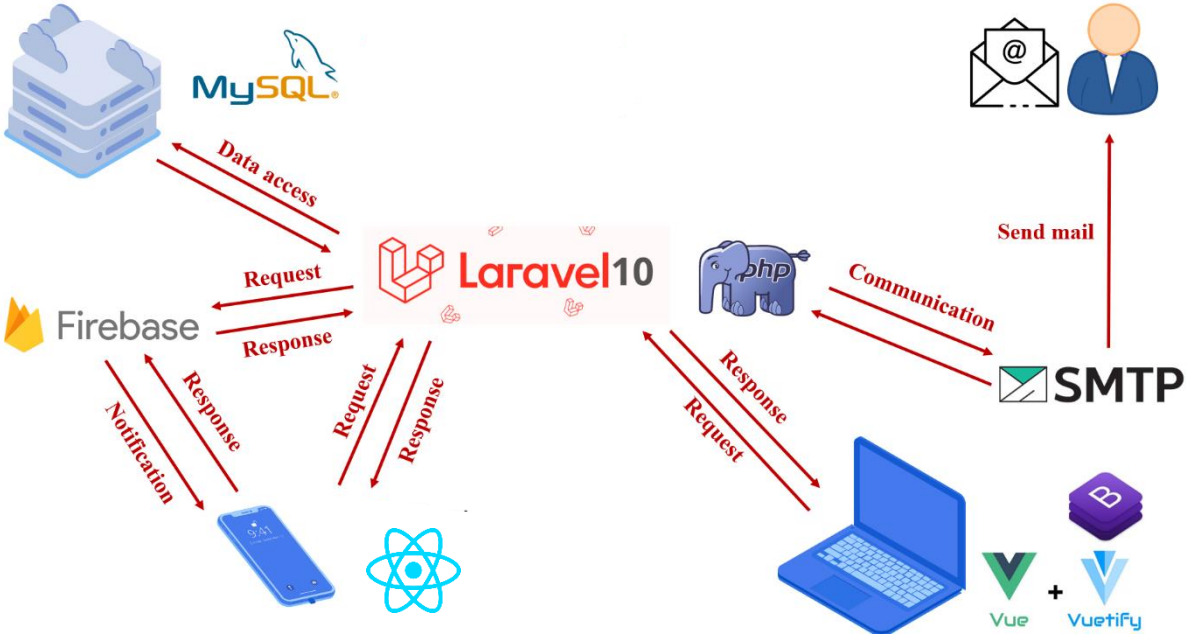


Figure 13: Schéma des technologies utilisés dans le projet

Conclusion

La sélection des outils et technologies pour le projet FixMate a été cruciale pour son succès. En utilisant PHP pour le développement côté serveur, nous avons pu créer une application web dynamique et robuste, capable de s'adapter à différents environnements et de gérer efficacement les interactions avec la base de données. MySQL a servi de système de gestion de base de données, offrant une fiabilité et des performances élevées, L'intégration du protocole SMTP a permis une communication par e-mail fiable et sécurisée, essentielle pour diverses fonctionnalités de l'application.

Laravel, avec son architecture MVC et ses fonctionnalités complètes a simplifié le développement et la maintenance de l'application tout en assurant une sécurité et une modularité optimales. L'utilisation des API RESTful a facilité l'interaction entre le back-end et les applications web et mobiles, garantissant une communication fluide et sécurisée. Enfin, le front-end développé avec HTML, CSS, JavaScript, et Bootstrap a permis de créer une interface utilisateur intuitive et responsive, offrant une expérience utilisateur optimale.

L'ensemble de ces technologies a permis de développer une application web performante, sécurisée et facile à utiliser, répondant aux besoins des utilisateurs tout en étant alignée sur les standards modernes de développement.

2. Développement de l’application web

2.1. Technologies et outils pour les interfaces web

Dans le développement front-end de notre application web, nous avons utilisé plusieurs technologies et outils clés pour créer une interface utilisateur attrayante et interactive. Voici les principales technologies et outils que nous avons utilisés :

HTML (HyperText Markup Language)

Nous avons utilisé HTML pour définir la structure et les éléments de base de chaque page de l'application. HTML nous a permis de créer une hiérarchie logique des contenus, tels que les en-têtes, les paragraphes, les formulaires et les tableaux.

CSS (Cascading Style Sheets)

CSS a été utilisé pour styliser et mettre en forme les éléments HTML de l'application. Nous avons défini les couleurs, les polices, les marges, les espacements et d'autres propriétés visuelles pour créer une présentation cohérente et esthétique.

JavaScript

Nous avons utilisé JavaScript pour ajouter des fonctionnalités interactives et dynamiques à l'application. Cela inclut la manipulation des éléments de la page, la gestion des événements, les requêtes AJAX et la validation des formulaires.

Bootstrap

Bootstrap est un framework CSS populaire qui offre une collection de composants et de styles prédéfinis pour le développement web. Nous avons utilisé Bootstrap pour créer un design responsif et harmonieux, ainsi que pour accélérer le développement en utilisant ses classes CSS réutilisables.

Vue.js

Vue.js est un Framework JavaScript progressif qui permet de construire des interfaces utilisateur réactives. Nous avons utilisé Vue.js pour la gestion de l'état de l'application, la création de composants réutilisables et la mise en œuvre de fonctionnalités dynamiques telles que le filtrage des données et la mise à jour en temps réel.

Vuetify

Vuetify est une bibliothèque de composants basée sur Vue.js qui offre une gamme complète de composants prêts à l'emploi et de styles préconfigurés. Nous avons utilisé Vuetify pour créer une interface utilisateur cohérente et moderne, en utilisant ses composants esthétiques tels que les boutons, les formulaires, les cartes et les menus.

Vite.js

Vite.js est un outil de build rapide et léger pour le développement d'applications web modernes. Nous avons utilisé Vite.js pour améliorer les performances du développement front-end en fournissant un environnement de développement instantané. Grâce à sa compilation rapide des modules, Vite.js nous a permis de bénéficier d'un temps de rechargement rapide lors du développement et de réduire les temps d'attente entre les modifications de code.

Ces technologies et outils nous ont permis de développer une interface utilisateur riche, réactive et esthétique pour notre application web. Ils ont facilité la mise en place de fonctionnalités avancées et ont accéléré le développement grâce à leurs fonctionnalités prêtes à l'emploi et leur documentation complète.

Vue I18n

Pour prendre en charge la localisation et la traduction dans notre application web, nous avons utilisé Vue I18n. Vue I18n est un plugin de localisation officiel pour Vue.js qui facilite la gestion des traductions dans les différentes langues. Grâce à Vue I18n, nous avons pu fournir une expérience multilingue aux utilisateurs de l'application, en leur permettant de sélectionner leur langue préférée et de voir les contenus traduits correspondants.

2.2. Interfaces du l'application web



Figure 14 : L'écran de chargement

L'écran de chargement sert de page intermédiaire qui apparaît pendant que l'application récupère des données ou initialise ses composants. Cet écran assure à l'utilisateur que l'application est en cours de traitement et sera prête sous peu. L'icône animée de chargement

fournit une indication visuelle du processus en cours, empêchant les utilisateurs de penser que l'application ne répond pas.

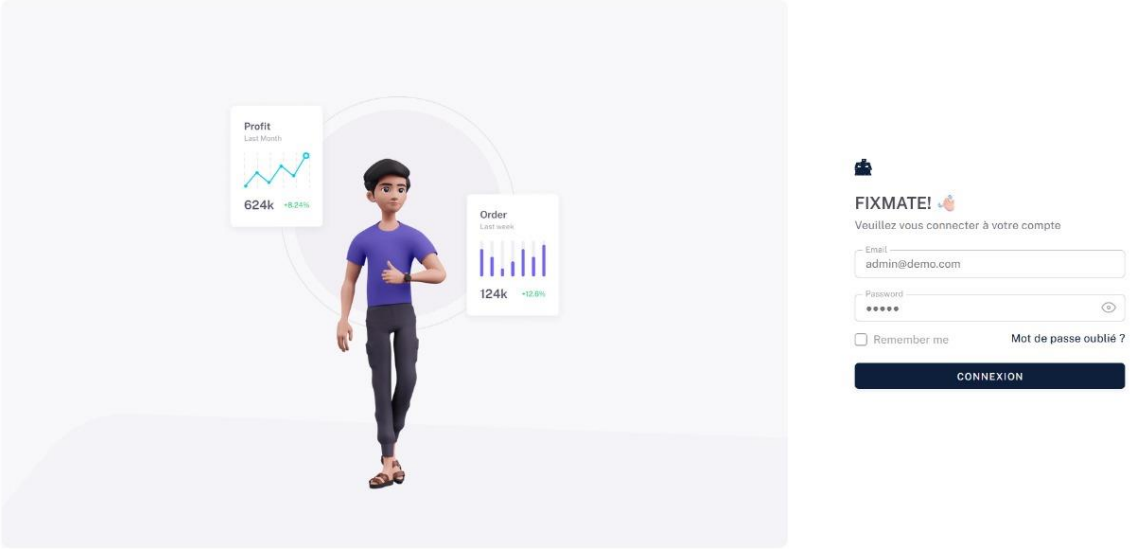


Figure 15 : L'écran de connexion

L'écran de connexion permet aux utilisateurs de s'authentifier et d'accéder à l'application. Cette page est essentielle pour sécuriser l'accès aux fonctionnalités et aux données de l'application. Les utilisateurs doivent saisir leur adresse e-mail et leur mot de passe pour se connecter. Il y a aussi une option "Se souvenir de moi" pour garder les utilisateurs connectés et un lien "Mot de passe oublié ?" pour récupérer leur mot de passe en cas de besoin.

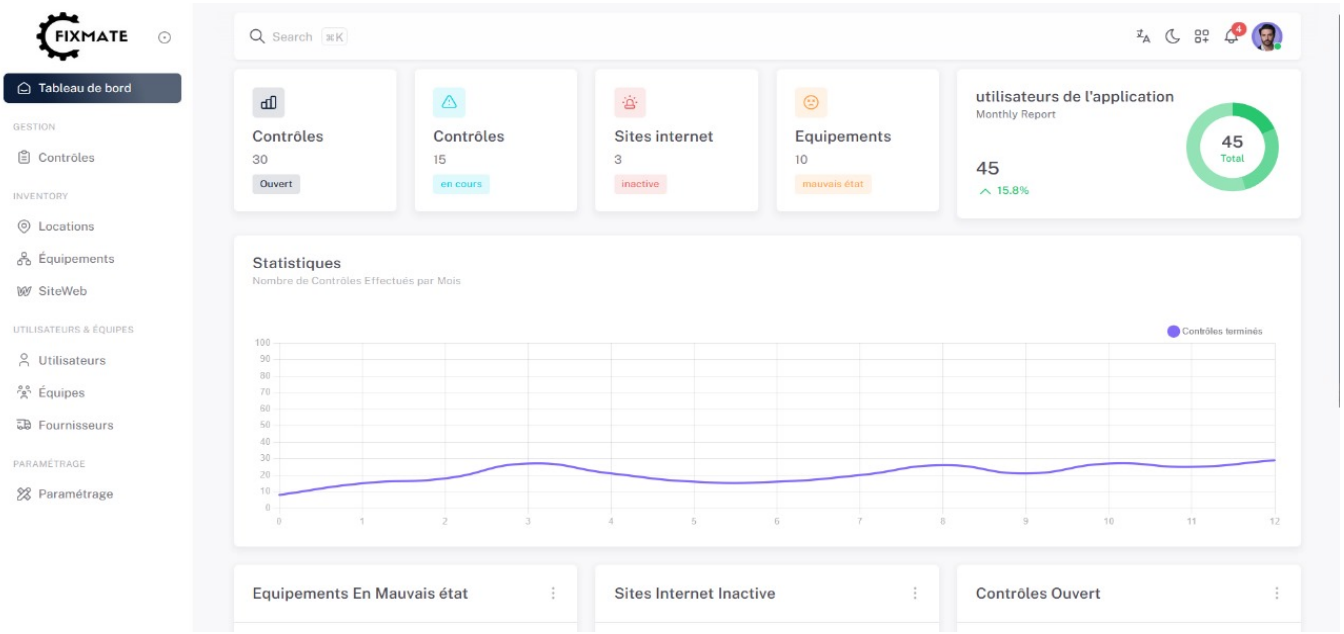


Figure 16 : Tableau de Bord

Pour la page intitulée "Tableau de Bord", il s'agit d'une vue d'ensemble permettant de visualiser et d'analyser divers indicateurs clés liés à la gestion et au suivi des activités pertinentes pour le projet. Cette page présente un tableau synthétique des données, regroupant des informations telles que le nombre de contrôles effectués, l'état des équipements, la disponibilité des sites internet, ainsi que d'autres métriques importantes. L'objectif principal est de fournir aux utilisateurs une vision claire et concise de la performance globale des différents aspects du projet, facilitant ainsi la prise de décisions informées et l'identification des domaines nécessitant une attention particulière.

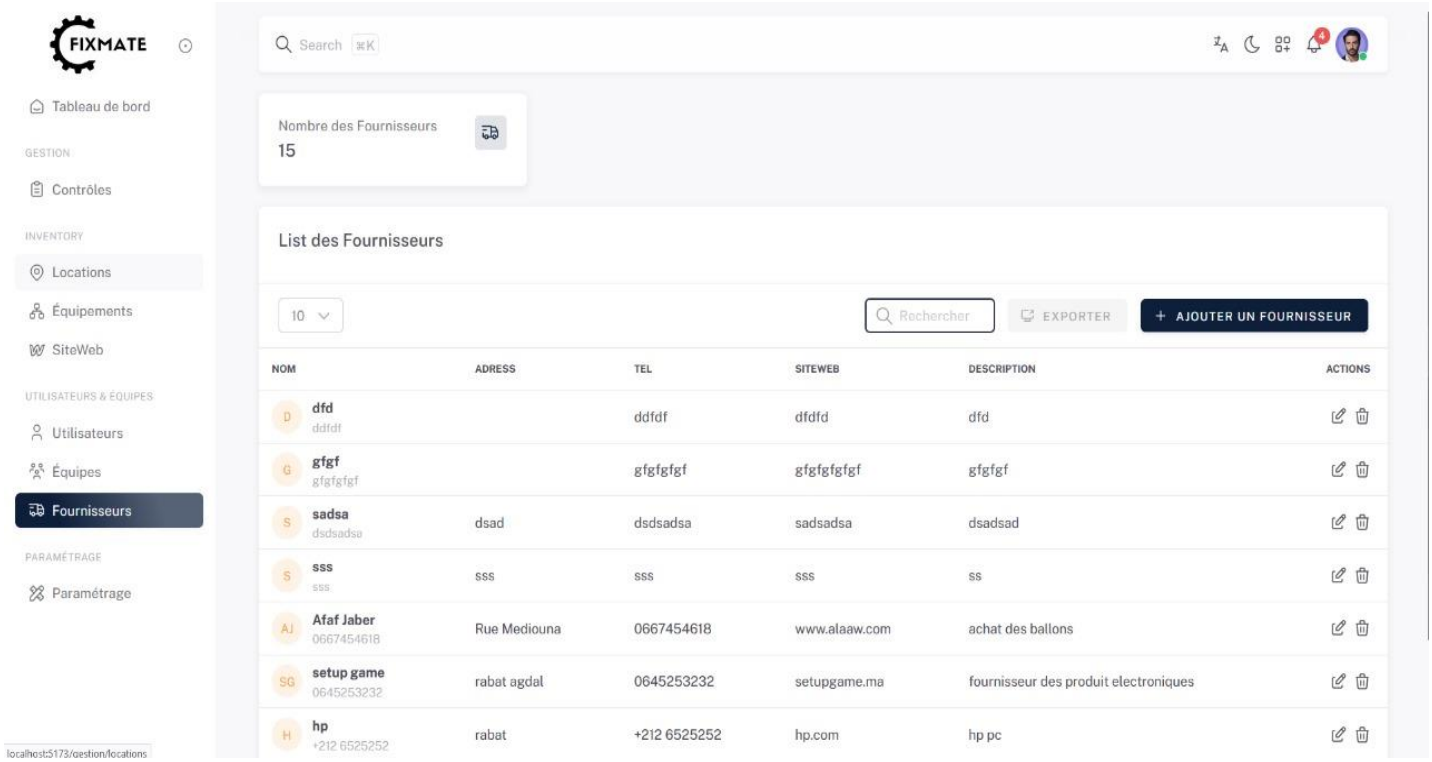


Figure 17 : La page des fournisseurs

La page des fournisseurs est une section cruciale au sein de l'application, conçue pour faciliter la gestion et la collaboration avec les fournisseurs externes. Son rôle principal est de centraliser toutes les informations pertinentes relatives aux fournisseurs avec lesquels l'entreprise travaille. Cette page offre une interface conviviale permettant aux utilisateurs autorisés, tels que les gestionnaires des achats ou les responsables des relations fournisseurs, d'accéder rapidement et efficacement à ces informations.

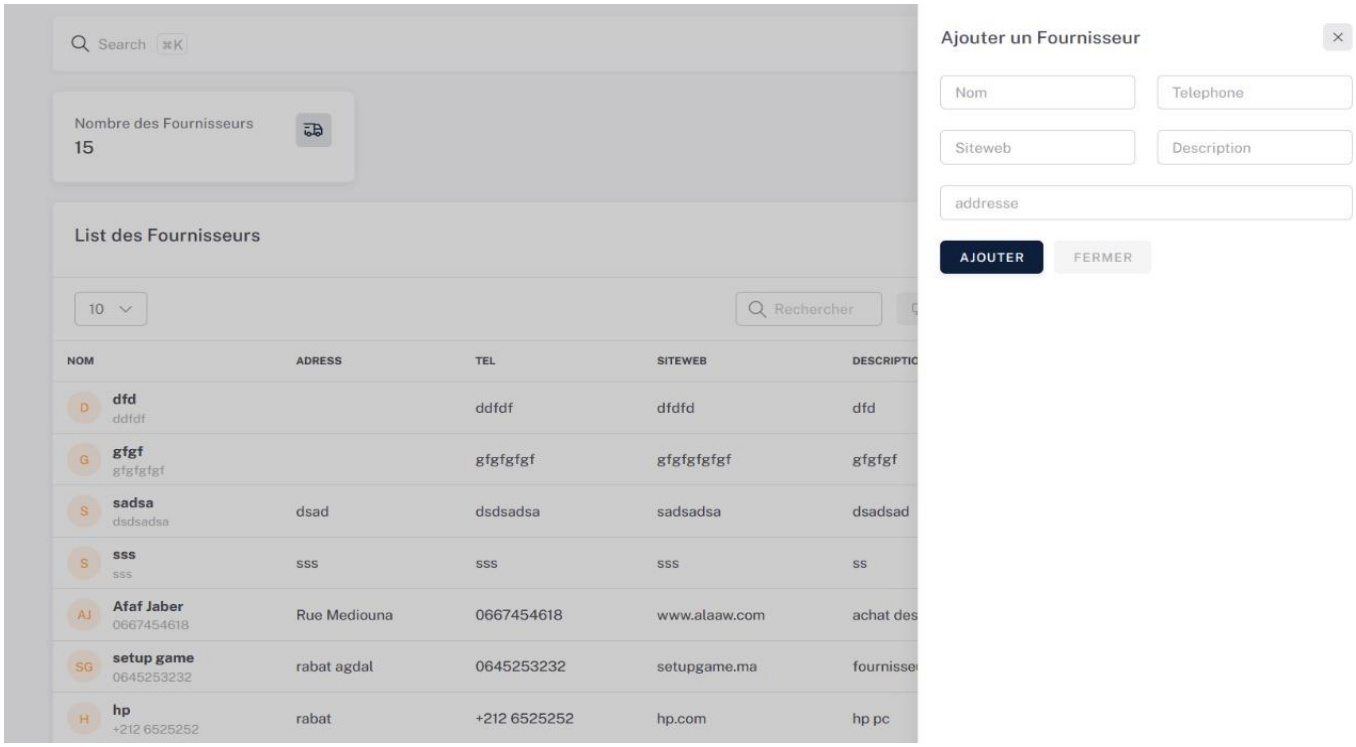


Figure 18 : Formulaire d'ajout d'un fournisseur

Pour ajouter un nouveau fournisseur, les utilisateurs peuvent simplement cliquer sur un bouton dédié. Cela fait apparaître un formulaire intuitif à droite de l'écran, où ils peuvent saisir les informations requises telles que le nom du fournisseur, les coordonnées de contact, et d'autres spécifications pertinentes.

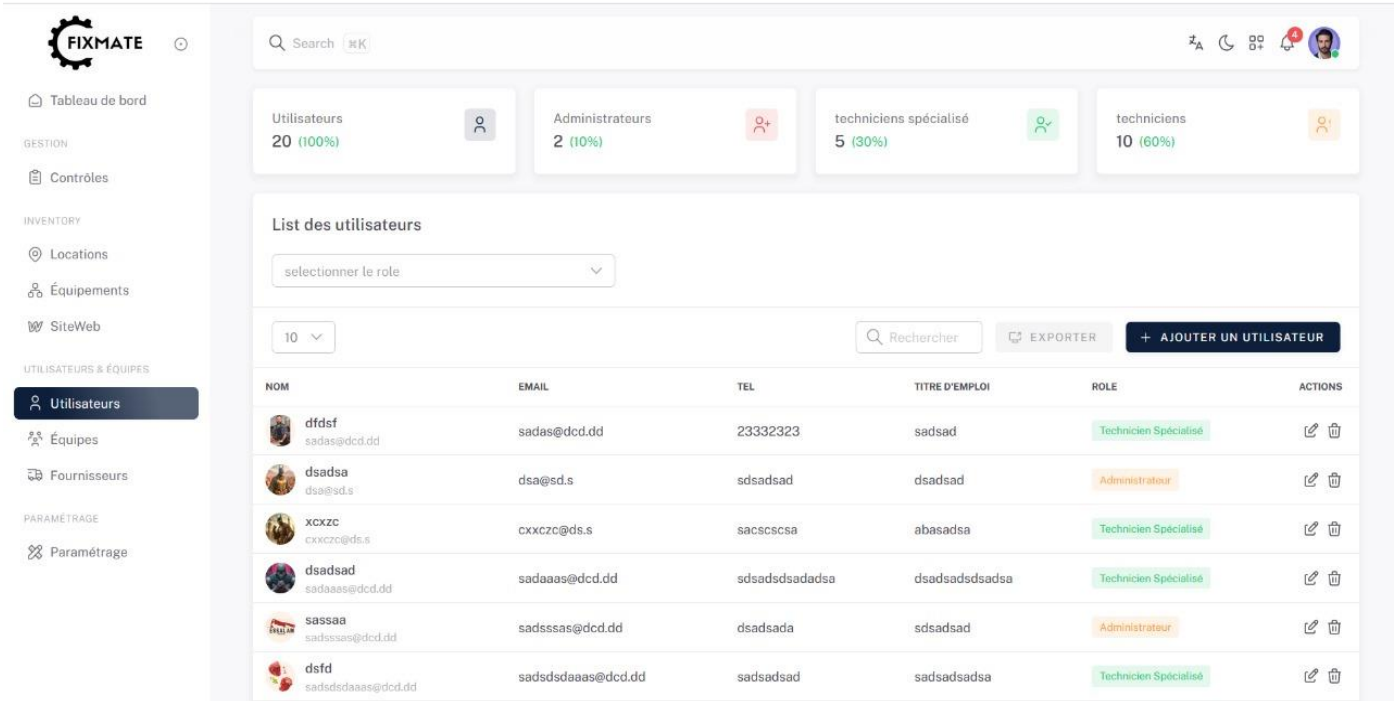


Figure 19 : La page des utilisateurs

La page des utilisateurs est une composante centrale de l'application, conçue pour gérer efficacement les comptes et les rôles des utilisateurs. Elle offre une vue d'ensemble claire et détaillée de tous les utilisateurs enregistrés, ainsi que de leurs rôles attribués au sein du système. Cette page permet aux administrateurs et aux gestionnaires de contrôler les accès et les permissions, assurant ainsi la sécurité et la gestion optimale des ressources.

Voici quelques fonctionnalités clés de la page des utilisateurs :

Vue d'ensemble des Utilisateurs : La page affiche un tableau récapitulatif avec le nombre total d'utilisateurs et la répartition par rôles. Cela permet de visualiser rapidement la structure des utilisateurs dans l'organisation.

Gestion des Rôles : Les utilisateurs sont classés en différents rôles tels que Administrateurs, Techniciens Spécialisés, et Techniciens. Chaque rôle a des permissions spécifiques définies par l'administrateur.

Liste des Utilisateurs : Une liste détaillée des utilisateurs enregistrés est disponible, avec des informations telles que l'adresse e-mail, le numéro de téléphone, et le rôle attribué à chaque utilisateur. Cela facilite la recherche et la gestion individuelle des comptes.

Filtre et Recherche : Les utilisateurs peuvent être filtrés par rôle ou recherchés par nom ou par adresse e-mail. Cela permet de trouver rapidement un utilisateur spécifique ou de voir tous les utilisateurs d'un rôle particulier.

Ajout et Modification des Utilisateurs : Les administrateurs ont la possibilité d'ajouter de nouveaux utilisateurs en spécifiant leur rôle et en remplissant les informations nécessaires. De plus, ils peuvent modifier les détails des utilisateurs existants, y compris leur rôle et leurs informations de contact.

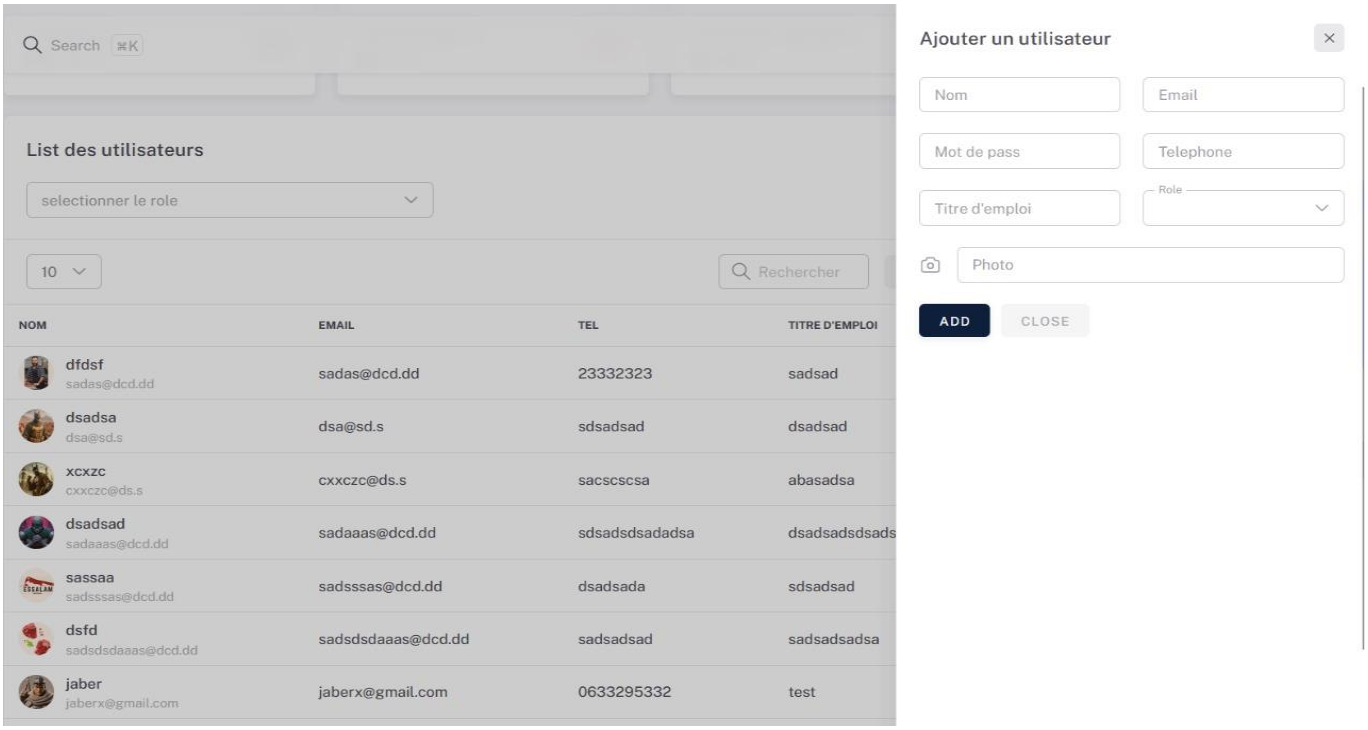


Figure 20 : Formulaire d'ajout d'un utilisateur

Ce formulaire permet aux administrateurs de saisir les informations nécessaires pour créer un nouveau compte utilisateur. Les champs obligatoires incluent le nom, l'email, le mot de passe, le téléphone, le titre d'emploi et le rôle. L'option pour télécharger une photo est également disponible pour personnaliser le profil de l'utilisateur. Une fois les informations saisies, l'administrateur peut soumettre le formulaire en cliquant sur le bouton "Ajouter".

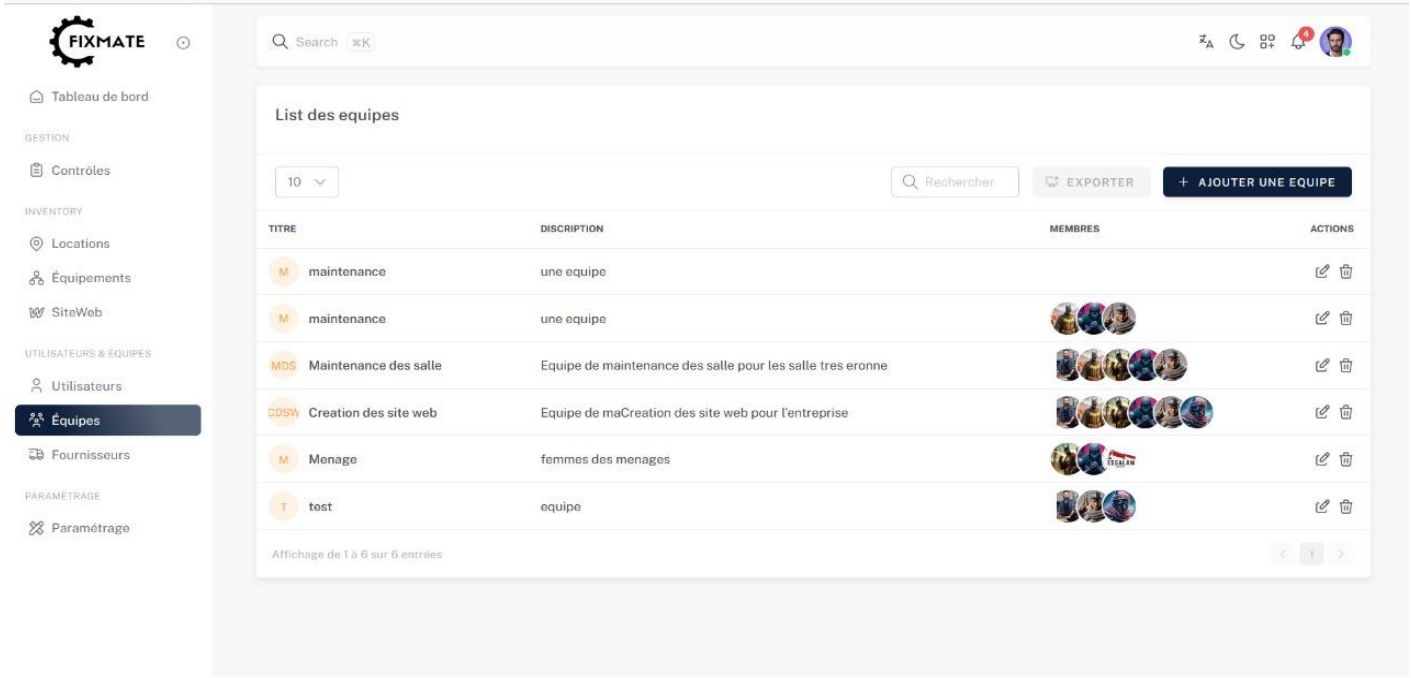


Figure 21 : La page des équipes

La page des équipes constitue une interface centrale pour gérer toutes les équipes au sein de l'application. Elle permet de visualiser, rechercher, exporter et ajouter des équipes.

Chaque équipe est représentée par les éléments suivants :

Titre de l'équipe : Un titre descriptif identifiant clairement l'équipe.

Description : Une brève description détaillant les objectifs ou le domaine d'activité de l'équipe.

Membres de l'équipe : La liste des membres qui composent l'équipe

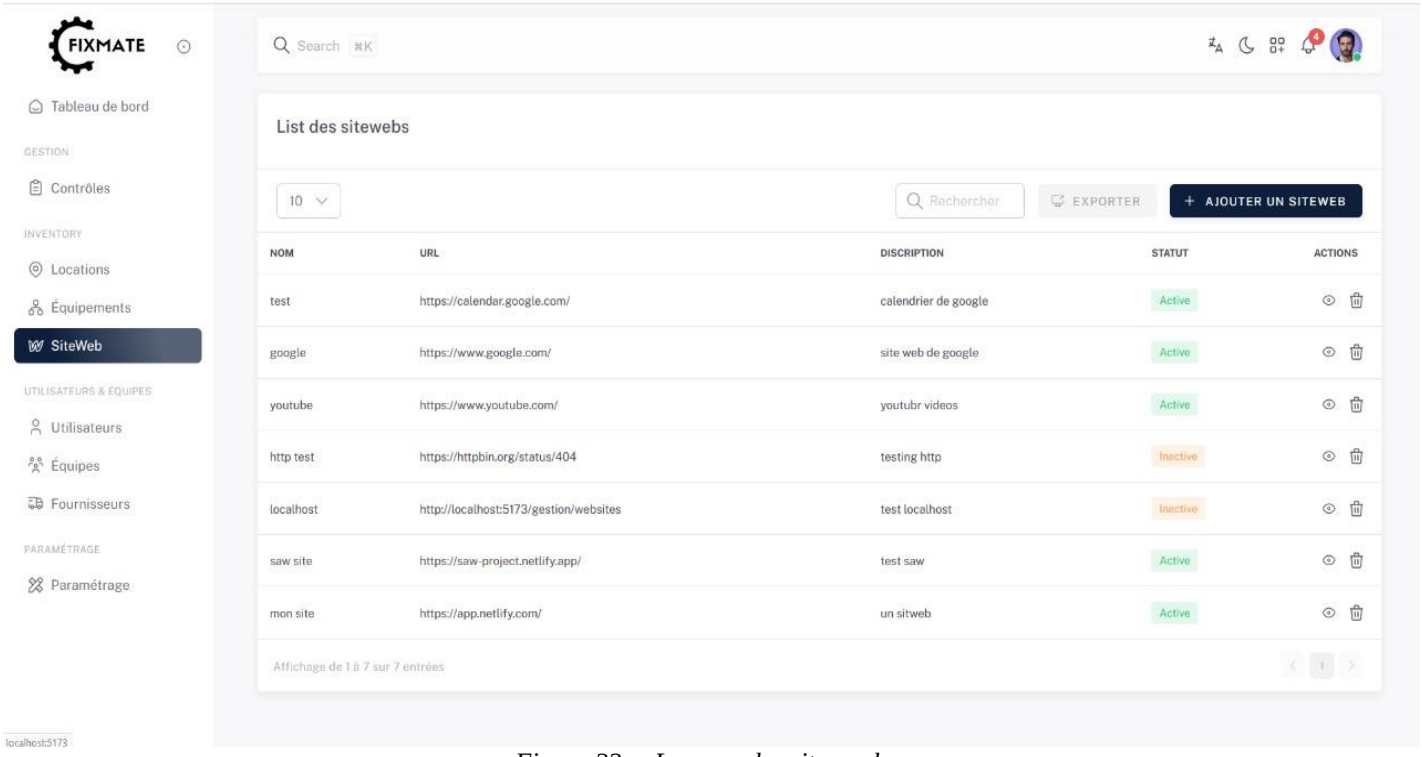


Figure 22 : La page des sites web

La page des sites web est une fonctionnalité essentielle de l'application permettant de gérer et de surveiller les différents sites enregistrés. Chaque site est répertorié avec les détails suivants : nom, URL, description et statut (actif ou inactif). Cela permet une gestion efficace des ressources et une surveillance continue de la disponibilité des sites.

Fonctionnalités de la Page

Surveillance Automatique : Toutes les 30 minutes, le système vérifie automatiquement l'état de chaque site web enregistré. Si un site devient inactif, le statut est mis à jour et une notification par e-mail est envoyée aux administrateurs pour les informer du problème.

Vérification Manuelle : Les utilisateurs peuvent également vérifier manuellement l'état d'un site en cliquant sur une icône d'œil. Cela permet de s'assurer que les sites critiques sont opérationnels à tout moment.

Changement Automatique du Statut : Si un site qui était actif devient inactif et qu'un utilisateur clique sur le bouton de vérification, le statut est automatiquement mis à jour dans la base de données.

Ajout de Site Web : Pour ajouter un nouveau site, les utilisateurs saisissent le nom, l'URL et une brève description. En cliquant sur le bouton d'ajout, le système vérifie la disponibilité du site et ajuste le statut en conséquence. Le nouvel enregistrement est ensuite intégré à la base de données pour une gestion continue.

Recherche et Exportation : Les utilisateurs peuvent rechercher des sites par nom ou URL et exporter les données pour une analyse externe ou une sauvegarde.

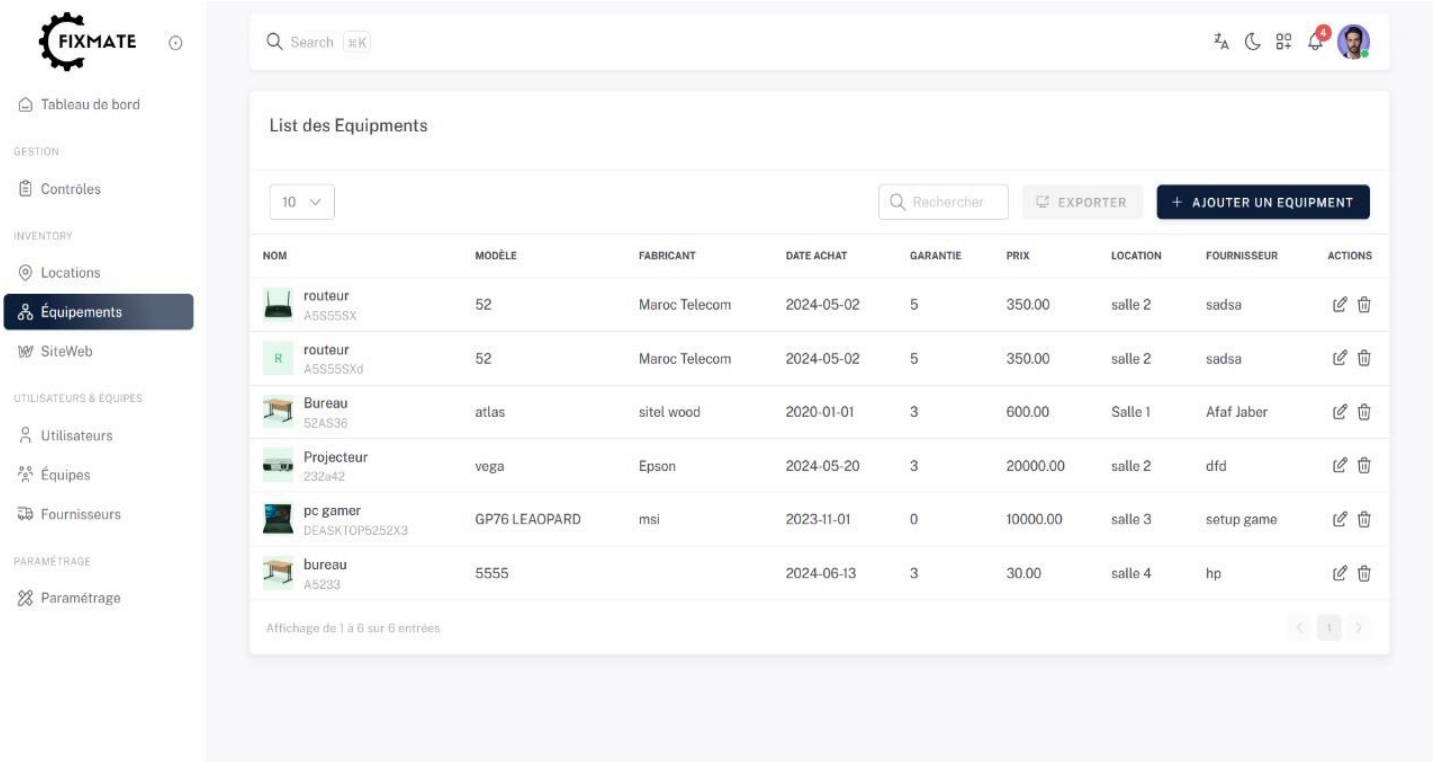


Figure 23 : La page des Équipements

La page des équipements centralise les informations essentielles sur chaque équipement utilisé dans l'entreprise. Chaque équipement est répertorié avec les détails suivants : nom, modèle, fabricant, date d'achat, garantie, prix, statut de location, fournisseur, et inclut une image représentative de l'équipement.

Fonctionnalités de la Page

Gestion Complète des Données : Les utilisateurs peuvent gérer efficacement les équipements en visualisant et en mettant à jour les informations telles que le nom, le modèle, le fabricant, la date d'achat, la période de garantie, le prix, et le statut de location.

Recherche Avancée : La fonction de recherche permet de trouver rapidement un équipement par son nom, son modèle, son numéro de série, etc., facilitant la localisation des informations spécifiques.

Exportation des Données : Les données des équipements peuvent être exportées pour des analyses externes ou pour une sauvegarde sécurisée.

Ajout d'Équipement : En cliquant sur le bouton "Ajouter un équipement", un formulaire apparaît sur la droite. Les utilisateurs saisissent les détails tels que le nom, le modèle, l'année de fabrication, le prix, une description détaillée, le statut de location, le fournisseur, et téléchargent une photo de l'équipement.

Attribution des Responsabilités : Chaque équipement peut être associé à une personne responsable et à une équipe chargée de son entretien et de sa gestion, assurant une supervision efficace

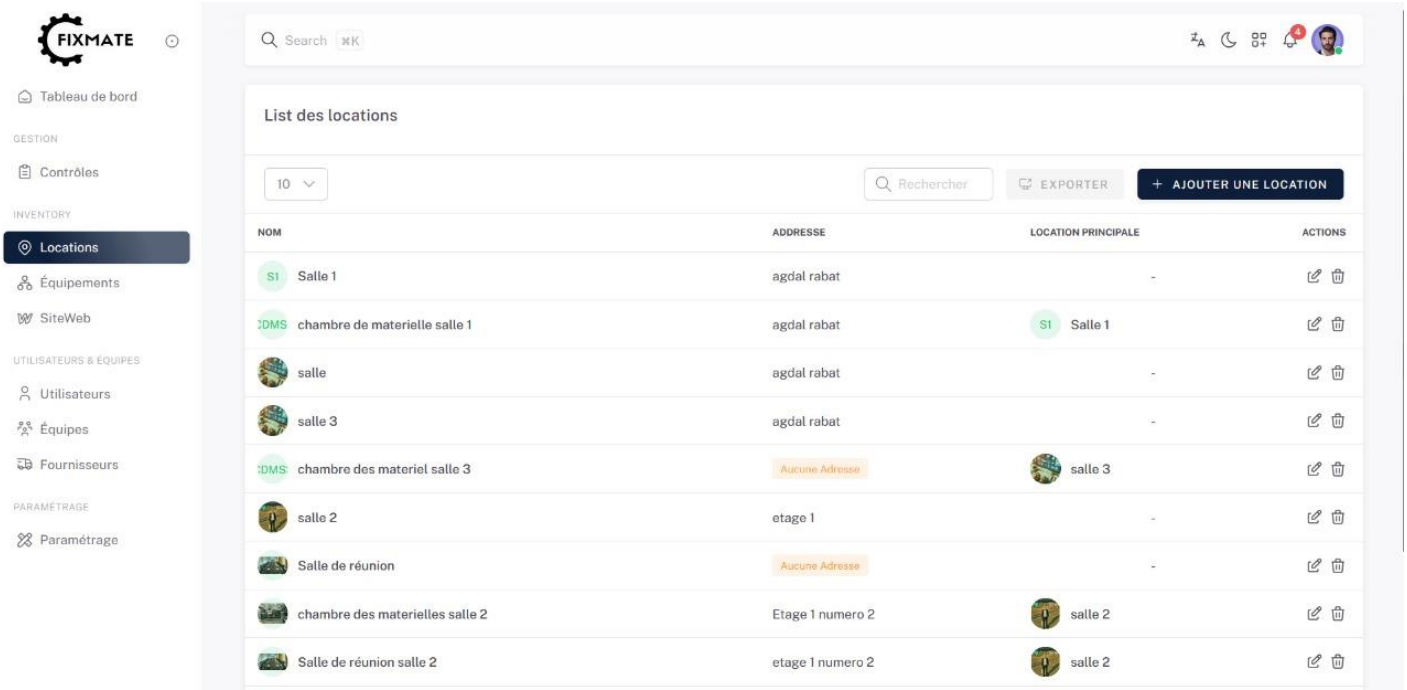


Figure 24 : La page des Locations

La page des locations répertorie toutes les locations disponibles dans l'entreprise, telles que les salles de réunion, les chambres de matériel, etc. Chaque location est définie par son nom, son adresse et une éventuelle salle parente (location principale).

Fonctionnalités de la Page

Liste des Locations : Affiche toutes les locations avec leurs détails, y compris le nom, l'adresse et la salle parente le cas échéant.

Gestion des Locations : Permet aux utilisateurs de gérer les locations existantes en visualisant, modifiant ou supprimant leurs informations.

Ajout de Nouvelle Location : Les utilisateurs peuvent ajouter une nouvelle location en cliquant sur le bouton "Ajouter une location". Ils saisissent le nom et l'adresse de la nouvelle location, ainsi que la salle parente si applicable. Ce champ n'est pas obligatoire.

Recherche Avancée : Facilite la recherche d'une location spécifique par son nom ou son adresse, offrant une navigation rapide et efficace à travers les différentes options de location.

Exportation des Données : Permet d'exporter les données des locations pour des besoins externes ou pour une analyse détaillée.

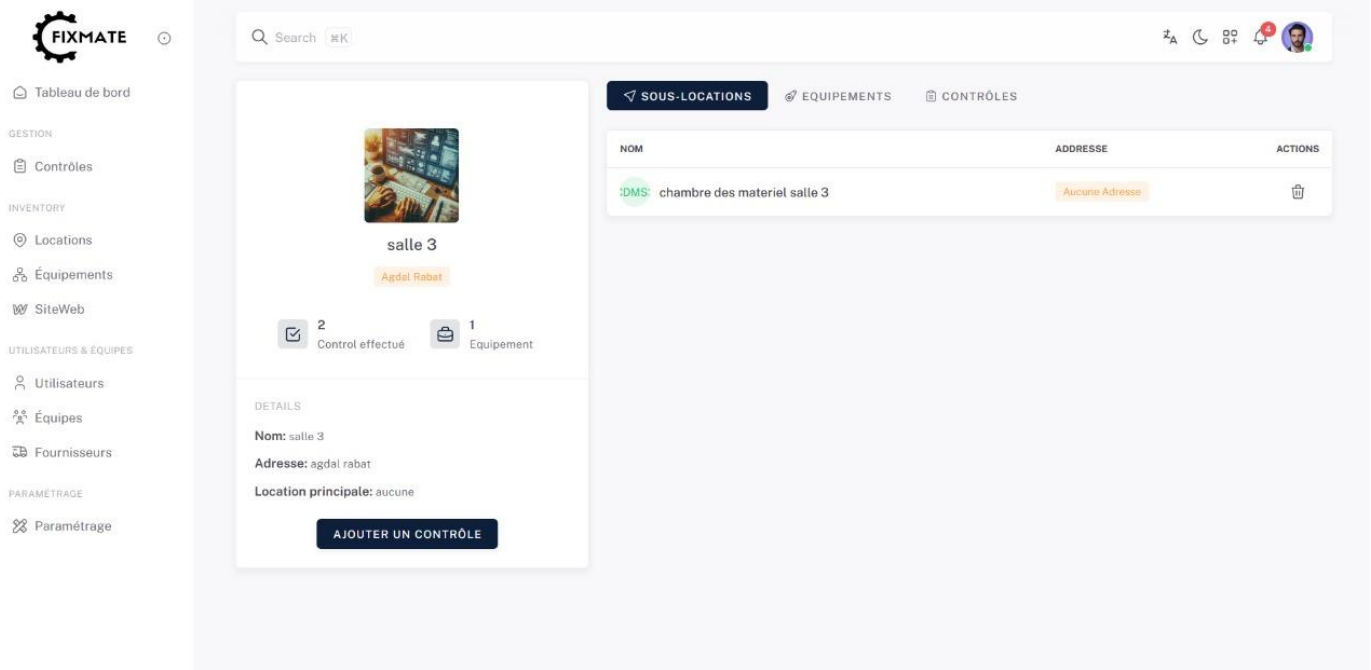


Figure 25 : Onglet Sous-Locations

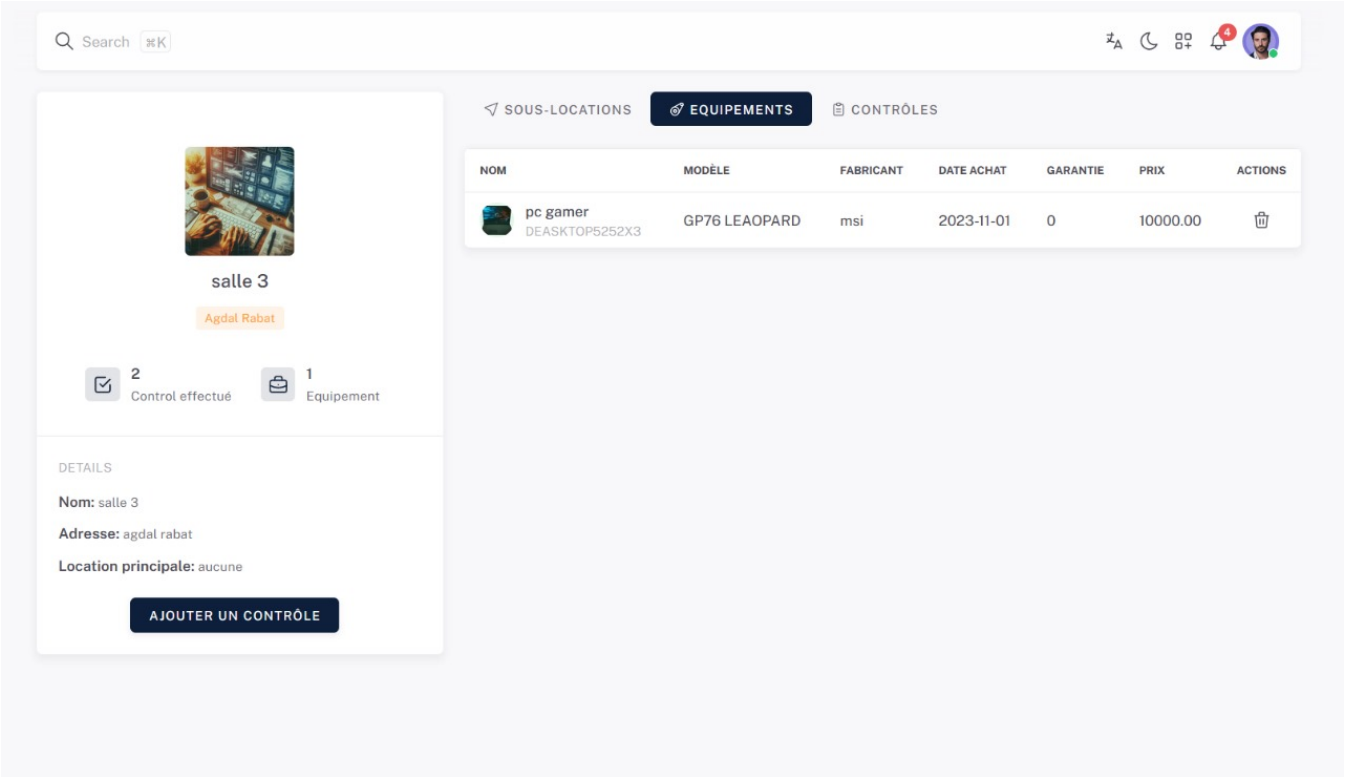


Figure 26 : Onglet Équipements

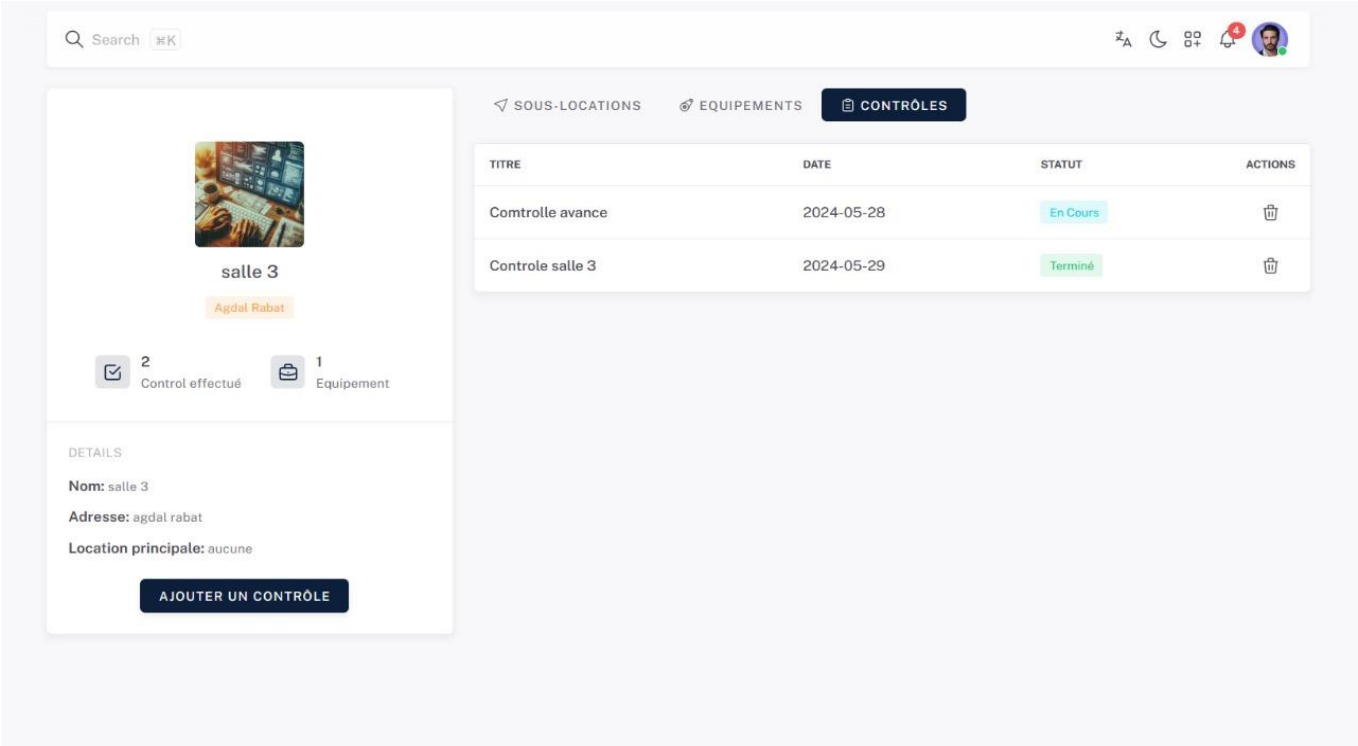


Figure 27 : Onglet Contrôles Effectués

Page Principale d'une Location

En cliquant sur le nom d'une location, les utilisateurs sont redirigés vers sa page principale. Voici ce à quoi cette page ressemble :

Section Gauche (Carte)

- Image : Une représentation visuelle de la location.
- Nom de la Location : Le nom spécifique de la location.
- Adresse : L'adresse physique de la location.
- Statistiques :

    - Nombre de contrôles effectués : Indique combien de fois des vérifications ont été effectuées dans cette location.

    - Nombre d'équipements : Montre le nombre d'équipements présents dans cette location.

Section Droite (Onglets)

- Onglet Sous-Locations : Liste toutes les sous-locations associées à cette location principale.
- Onglet Équipements : Affiche tous les équipements qui sont localisés dans cette location.
- Onglet Contrôles Effectués : Répertorie tous les contrôles et vérifications effectués spécifiquement dans cette location.

Fonctionnalités de la Page

- Navigation Intuitive : Facilite la navigation entre les différentes sections de la location.
- Visualisation Détaillée : Permet d'explorer les détails spécifiques de la location, y compris ses sous-divisions, ses équipements et les activités de contrôle associées.
- Gestion des Données : Offre aux utilisateurs la possibilité de gérer et de mettre à jour les informations relatives à la location principale et ses composants.

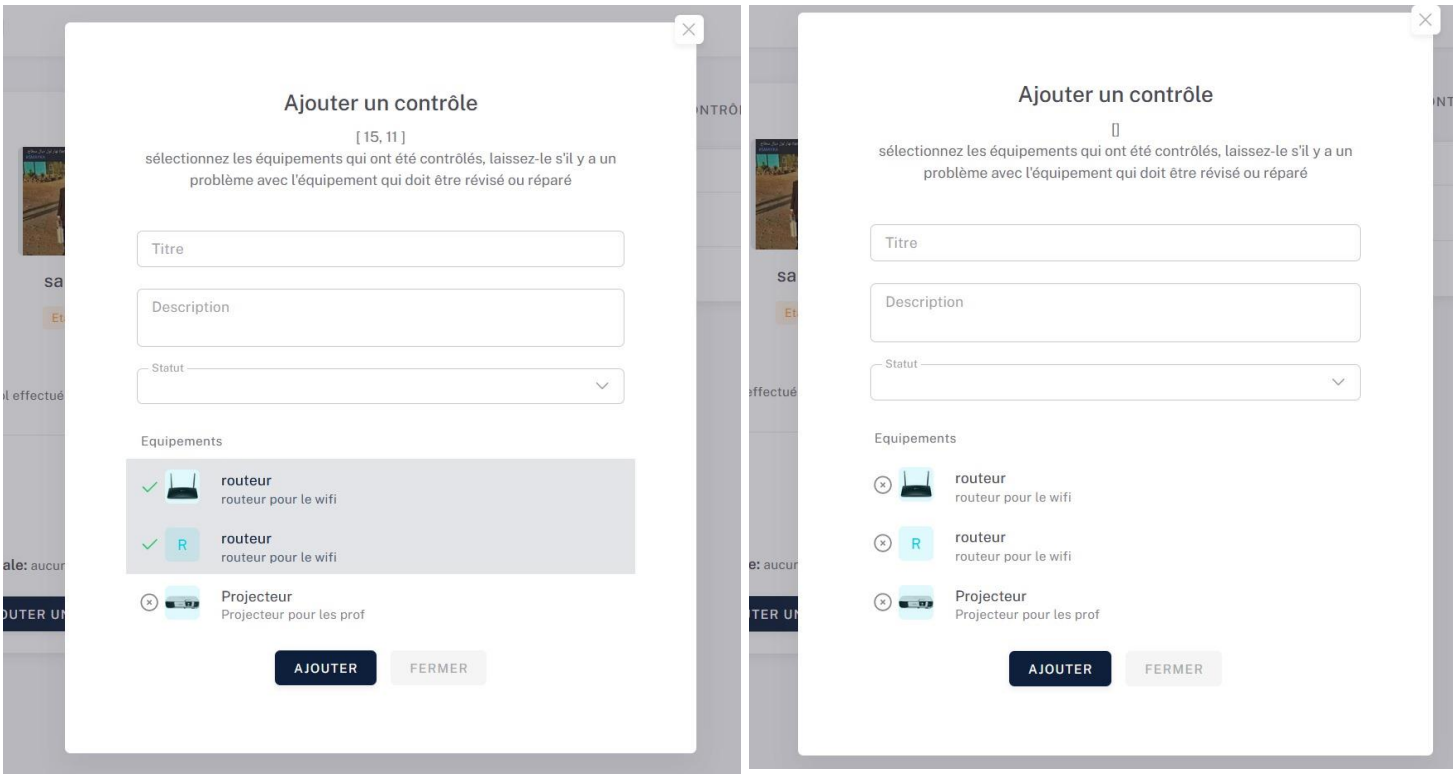


Figure 28 : Ajout de Contrôle dans une Location

Fonctionnalité d'Ajout de Contrôle dans une Location

La fonctionnalité principale sur la page d'une allocation est l'ajout de contrôle. Lorsqu'un technicien effectue une vérification des équipements dans l'allocation, il peut ajouter un contrôle en suivant ces étapes :

- Ajouter Contrôle : En cliquant sur le bouton « Ajouter contrôle », un pop-up ou un dialogue s'affiche.
- Saisie des Informations

Titre : Le technicien doit saisir un titre pour décrire succinctement le contrôle effectué.

Description : Une description détaillée peut être ajoutée pour documenter les détails spécifiques du contrôle.

Statut : Le statut du contrôle peut être choisi parmi « En cours », « Terminé » ou « Annulé », pour indiquer l'état actuel du processus de vérification.

- Sélection des Équipements :Une liste des équipements disponibles dans cette salle est affichée.Le technicien sélectionne uniquement les équipements qui sont en bon état pour le contrôle. Les équipements en mauvais état ne doivent pas être sélectionnés.

- Validation et Enregistrement :Une fois les informations nécessaires saisies et les équipements appropriés sélectionnés, le technicien peut valider et enregistrer le contrôle.

Cette action permet de documenter officiellement la vérification des équipements dans la location spécifique.Cette fonctionnalité assure une gestion efficace des contrôles effectués dans les différentes allocations, offrant ainsi un suivi précis de l'état des équipements et des actions de maintenance entreprises.

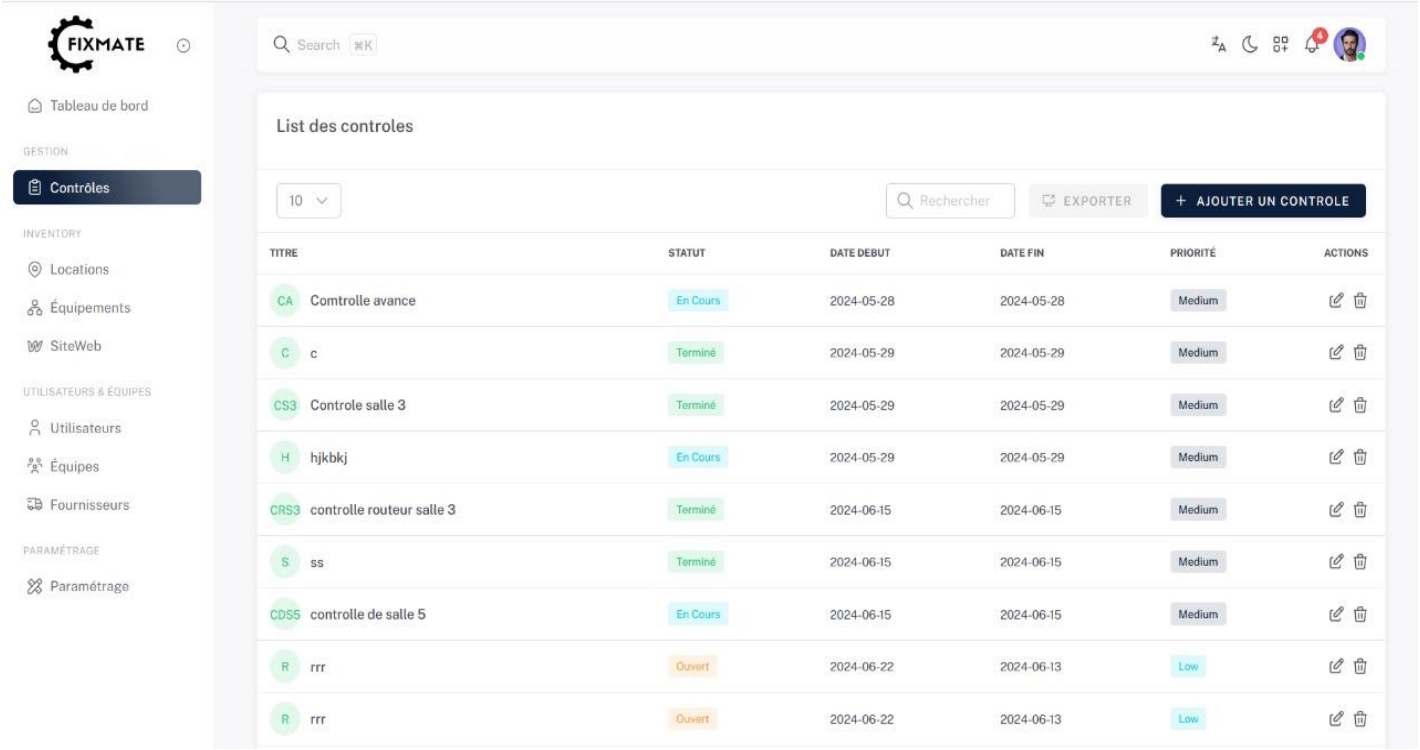


Figure 29 : Page des Contrôles

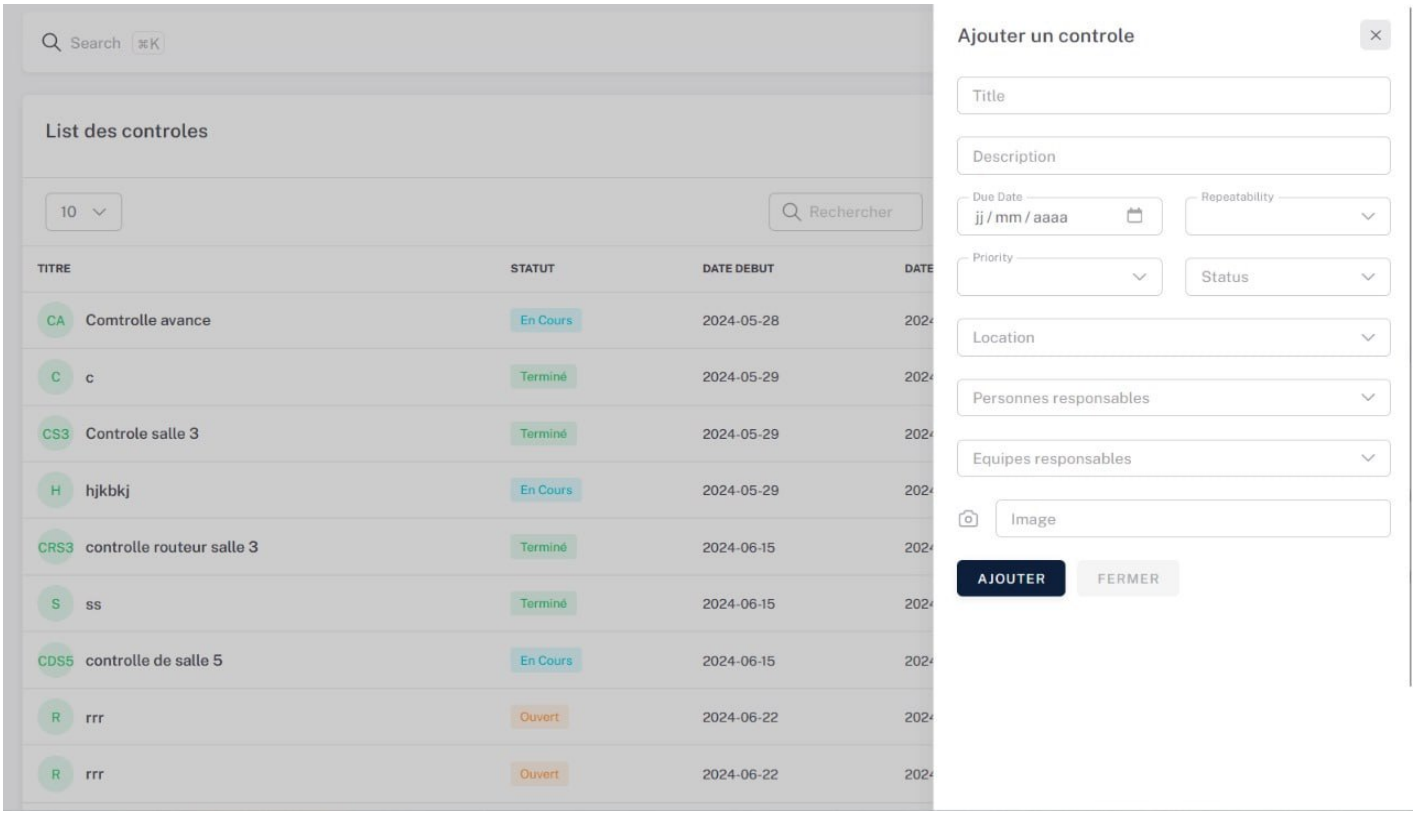


Figure 30 : Formulaire d'Ajout de Contrôle

Page des Contrôles

La page des contrôles centralise toutes les informations concernant les contrôles effectués, en cours, terminés ou en attente.

Fonctionnalités

Consultation des Contrôles :

Titre : Nom du contrôle ou de l'ordre de contrôle.

Statut : Indique si le contrôle est en cours, terminé, ouvert, etc.

Date de Début : Date de début prévue pour le contrôle.

Date de Fin : Date de fin prévue pour le contrôle.

Priorité : Niveau de priorité attribué au contrôle.

Recherche : Possibilité de rechercher les contrôles par nom, statut ou priorité pour filtrer les résultats.

Exportation : Permet d'exporter les données des contrôles pour une analyse plus approfondie.

Ajout de Contrôle : En cliquant sur le bouton « Ajouter un Contrôle », un formulaire apparaît à droite pour saisir les informations suivantes :

Nom : Nom du contrôle ou de l'ordre à contrôler.

Description : Détails et instructions spécifiques concernant le contrôle.

Date de Réalisation : Date prévue pour l'exécution du contrôle.

Répétabilité : Fréquence ou intervalle à laquelle le contrôle doit être effectué (optionnel).

Priorité : Importance ou urgence du contrôle.Statut : État actuel du contrôle (en cours, terminé, etc.).

Location : Lieu où le contrôle doit être effectué.

Personnes Responsables : Individus chargés de la réalisation du contrôle.

Équipes Responsables : Groupes ou équipes impliqués dans le contrôle.

Image : Optionnelle, une image associée au contrôle (facultatif).

Une fois ajouté, les personnes et équipes responsables sont automatiquement notifiées par e-mail. Les utilisateurs autorisés peuvent accéder et modifier ces données directement via l'application pour assurer la gestion efficace des contrôles.

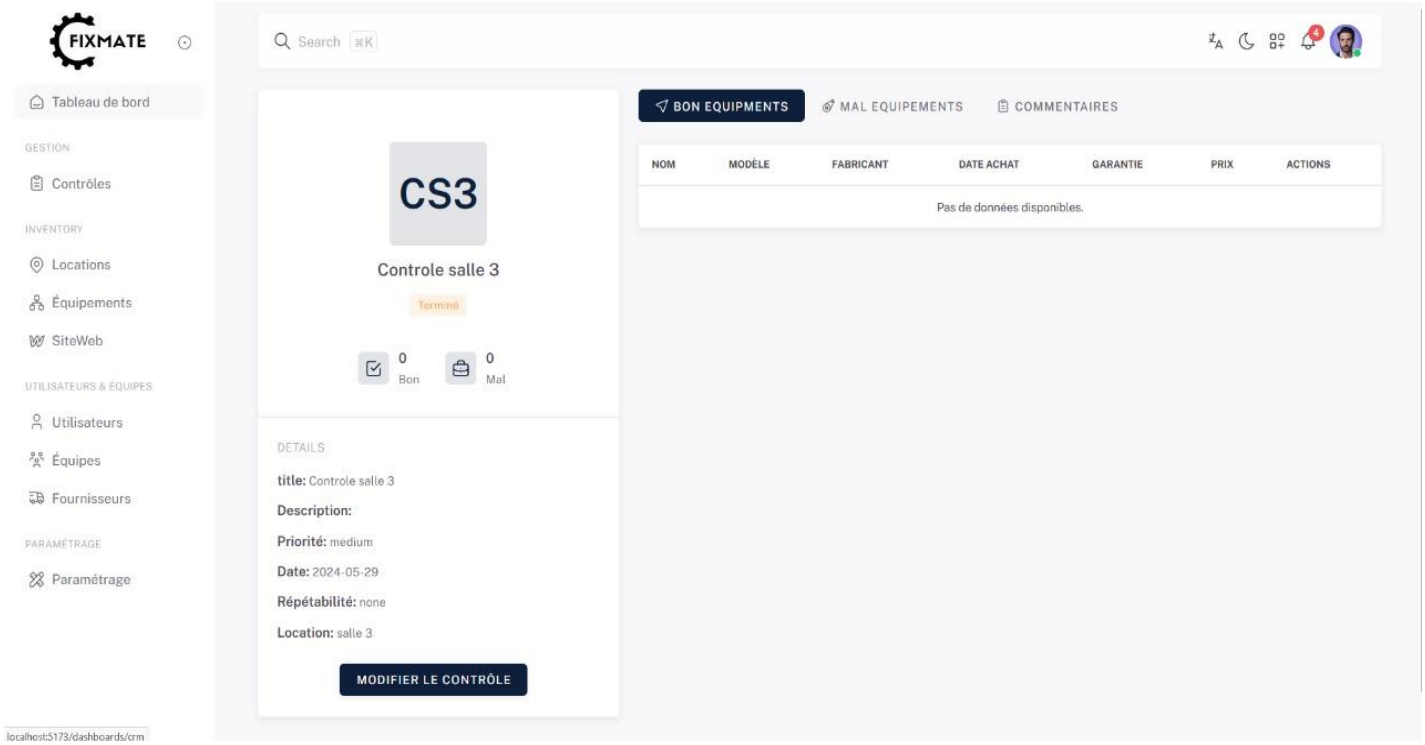


Figure 31 : Page Principale d'un Contrôle

Page Principale d'un Contrôle

Lorsqu'on clique sur le nom d'un contrôle dans la page des contrôles, on est redirigé vers la page principale de ce contrôle. La structure de cette page est similaire à celle d'une page principale de location :

Détails Généraux

À gauche, les détails généraux du contrôle sont affichés, comprenant des statistiques telles que le nombre d'équipements en bon état et le nombre d'équipements en mauvais état.

Les informations spécifiques comme le titre du contrôle, la description, la priorité, la répétabilité, la location associée, etc., sont également présentées pour une vue d'ensemble complète.

À droite, Trois onglets distincts sont disponibles :

Équipements en Bon État : Cette section liste les équipements qui ont été vérifiés et sont en bon état lors du contrôle.

Équipements en Mauvais État : Ici sont répertoriés les équipements qui nécessitent une attention supplémentaire ou qui ont été notés comme étant en mauvais état.

Commentaires : Un onglet dédié aux commentaires permet aux utilisateurs d'ajouter des notes pertinentes ou des observations spécifiques sur le contrôle effectué.

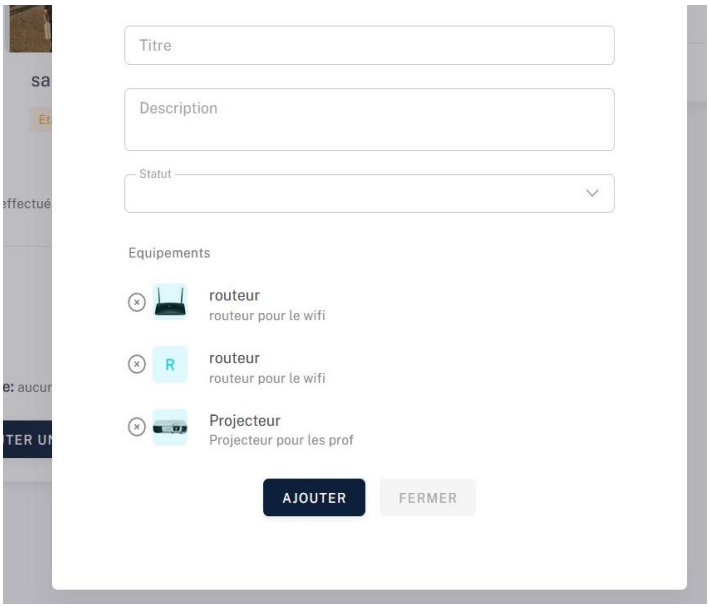


Figure 32 : Modification d'un Contrôle

Modification d'un Contrôle

Pour modifier un contrôle existant, vous pouvez cliquer sur le bouton « Modifier le contrôle ». Un dialogue supplémentaire s'affichera pour vous permettre de changer l'état du contrôle, par exemple de « En cours » à « Terminé » ou de « Ouvert » à « En cours », selon les options disponibles.

Sélection des Équipements Contrôlés :

Pour sélectionner les équipements spécifiques qui ont été contrôlés, vous utiliserez une méthode similaire à celle de la page principale de la location.

3. Développement de l'application mobile

3.1. Technologies et outils pour les interfaces mobiles

Dans le développement de l'application mobile, nous avons utilisé plusieurs outils et technologies clés pour créer une expérience utilisateur optimale. Voici les principaux outils et technologies que nous avons utilisés :

React-native :

React Native est un framework de développement d'applications mobiles open-source créé par Facebook. Il permet aux développeurs de construire des applications mobiles pour iOS et Android en utilisant JavaScript et React. Contrairement aux frameworks traditionnels de développement mobile qui nécessitent des langages de programmation spécifiques pour chaque plateforme (Swift ou Objective-C pour iOS et Java pour Android), React Native permet d'écrire la majeure partie du code une seule fois et de le déployer sur les deux plateformes.

3.2. Interfaces du l'application mobile

Le splash screen de l'application FixMate présente un design épuré et moderne, dominé par une couleur verte apaisante. Au centre de l'écran, le nom de l'application "FIXMATE" est affiché en majuscules, suivi du slogan "Contrôler vos systèmes" qui souligne la fonction principale de l'application. En bas de l'écran, une icône de chargement indique que l'application est en train de se lancer. Ce splash screen est conçu pour offrir une première impression

professionnelle et rassurante aux utilisateurs, en les informant immédiatement de la finalité de l'application : la gestion et le contrôle des systèmes d'information.



Figure 33 : Page SplashScreen de FixMate

La page d'accueil de l'application FixMate affiche un design convivial et fonctionnel, conçu pour offrir une expérience utilisateur fluide. En haut de l'écran, un message de bienvenue personnalisé "Hi, Othmane" est suivi de la localisation de l'utilisateur, "Maroc". Une barre de recherche permet aux utilisateurs de trouver rapidement des contrôles ou des informations spécifiques.

La section "Les derniers contrôles" présente les contrôles récents effectués, illustrés par des images pertinentes et des descriptions succinctes, comme "Contrôle de la buvette" et "Contrôle total", avec les emplacements respectifs. Cette section aide les utilisateurs à avoir une vue d'ensemble rapide des contrôles récents et leur statut.

Sous cette section, la partie "Les équipes" met en avant les équipes disponibles, comme l'"Equipe technique" notée 4.0 étoiles, offrant des détails sur leur localisation. Cela permet aux utilisateurs de visualiser rapidement les équipes disponibles et leur évaluation.

Enfin, la section "Equipements" est présente avec un lien "Voir tout" pour accéder à la liste complète des équipements gérés par l'application. Les icônes de navigation en bas de l'écran offrent un accès facile aux différentes sections de l'application, améliorant ainsi l'accessibilité et la navigation. Cette page d'accueil est conçue pour fournir une interface intuitive et riche en informations, facilitant la gestion des contrôles des systèmes d'information.

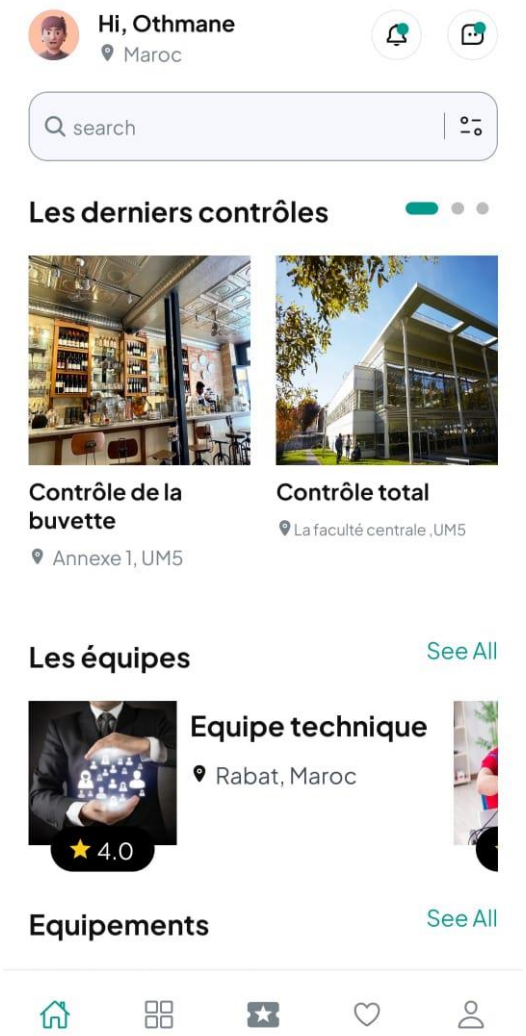


Figure 34: Page d'accueil de FixMate

La page de locations de l'application FixMate présente une interface claire et organisée pour faciliter la gestion et la recherche des différents lieux contrôlés. En haut de l'écran, une barre de recherche permet aux utilisateurs de trouver rapidement des lieux spécifiques en tapant des mots-clés. À côté de la barre de recherche, une icône de filtre offre des options pour affiner les résultats.

La section principale affiche les lieux avec des images et des descriptions succinctes. Par exemple, le "Bureau" situé à CGI, Maroc, et l'"Amphi D" situé à FSR, sont présentés avec des images pertinentes et leur localisation, permettant aux utilisateurs de visualiser rapidement les lieux contrôlés et leur état.

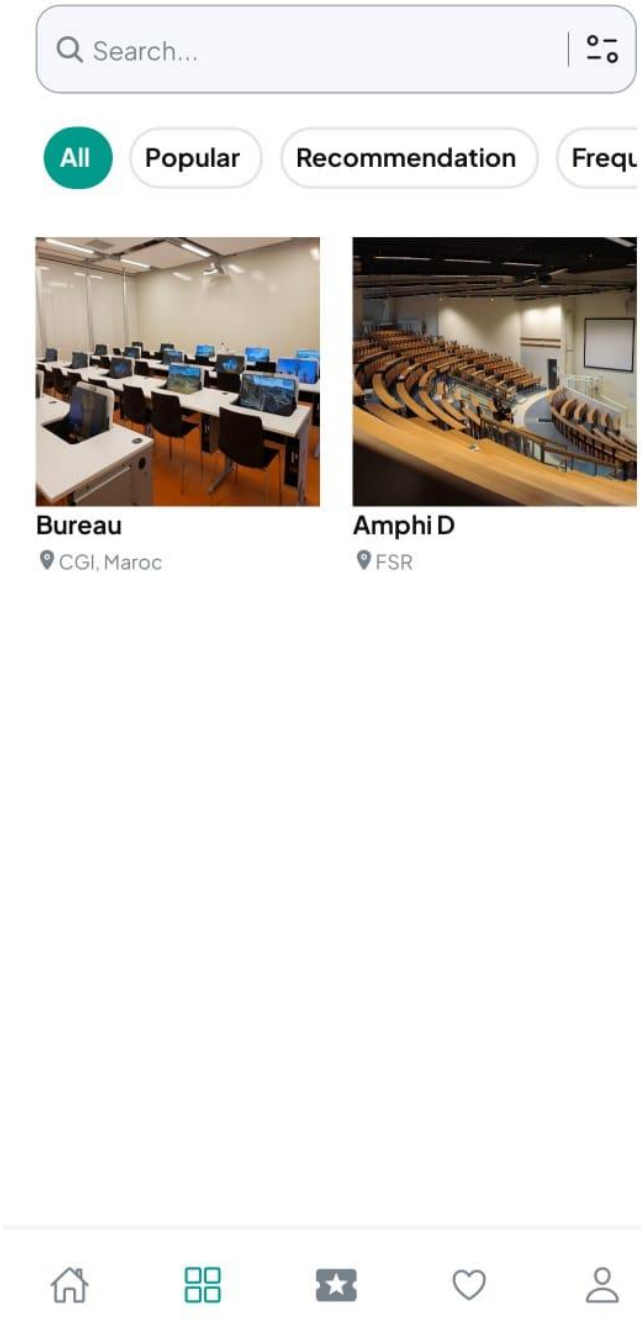


Figure 35 : Page de Location de FixMate

La page de Contrôles est conçue pour aider les utilisateurs à gérer et à suivre les activités de maintenance de leurs actifs. L'écran d'accueil affiche un résumé des activités de maintenance de l'utilisateur, notamment :

- **Contrôles à venir :** Une liste des contrôles de maintenance à venir qui sont prévus.
- **Contrôles récents :** Une liste des contrôles de maintenance récents qui ont été terminés.
- **Contrôles en retard :** Une liste des contrôles de maintenance en retard qui n'ont pas encore été terminés.

L'écran comprend également un bouton pour ajouter un nouveau contrôle. Il fournit aux utilisateurs un aperçu rapide de leurs activités de maintenance et facilite l'ajout de nouveaux contrôles. L'application serait un outil précieux pour les utilisateurs qui ont besoin de gérer et de suivre les activités de maintenance de leurs actifs.

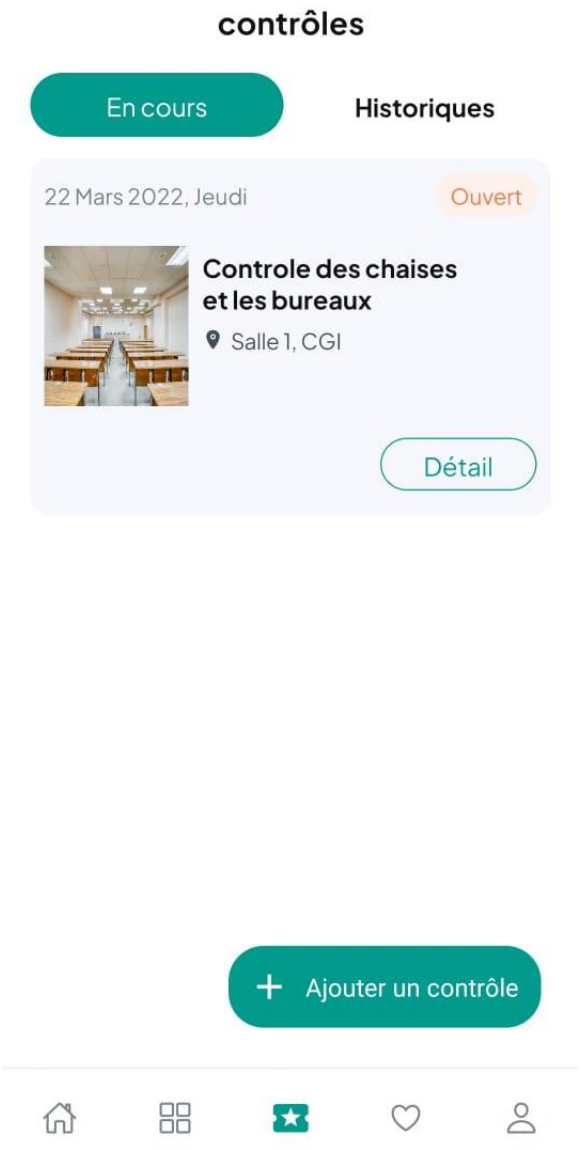


Figure 36 : Page de liste des contrôles de FixMate

Cette page fournit une liste des contrôles de maintenance récents qui ont été terminés.

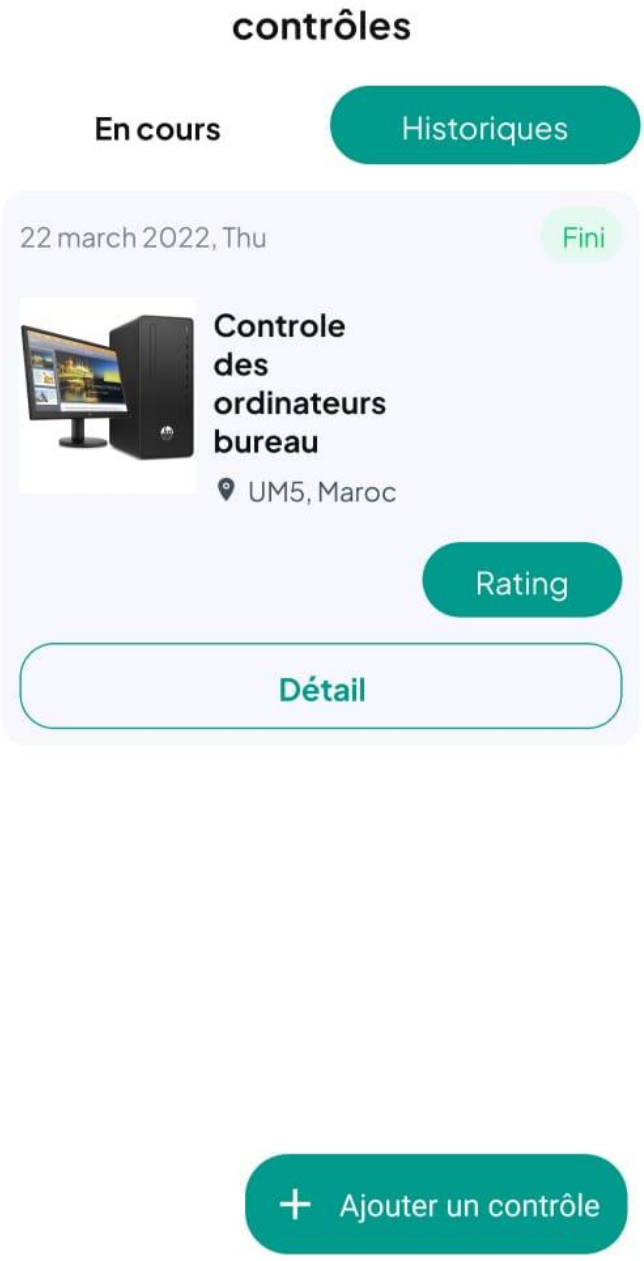


Figure 37 : page de liste des contrôles terminés

La page "Ajouter un contrôle" de l'application de gestion de la maintenance "Contrôles" est bien conçue et facile à utiliser. Elle fournit aux utilisateurs un ensemble complet de champs pour capturer toutes les informations pertinentes sur un nouveau contrôle. La page est également bien organisée, ce qui permet aux utilisateurs de trouver facilement les informations dont ils ont besoin. L'application serait un outil précieux pour les utilisateurs qui ont besoin de créer et de gérer des contrôles de maintenance.

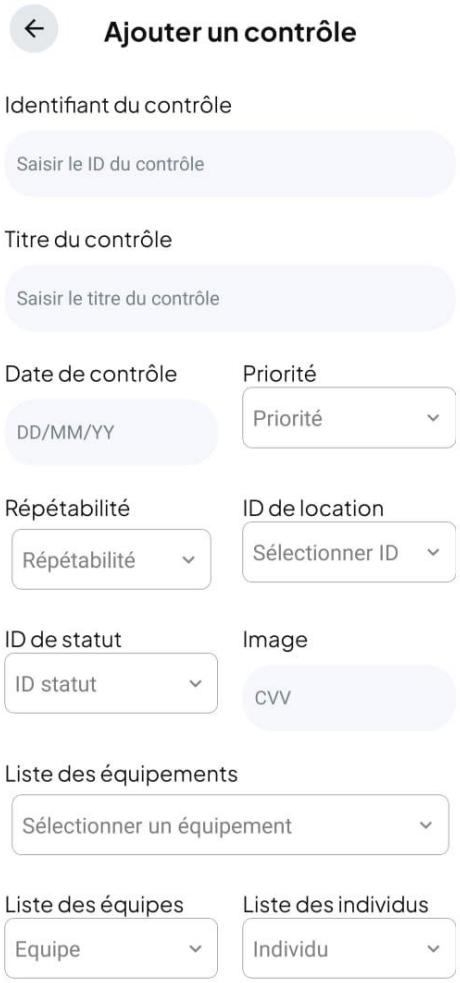


Figure 38 : Page de l'ajout de contrôles

La liste des équipements est bien organisée et facile à lire. Elle comprend toutes les informations dont les utilisateurs ont besoin pour identifier et suivre rapidement les équipements de leur entreprise.



Figure 39 : page de la liste des équipements de FixMate

L'application mobile "FixMate" pour la gestion des contrôles des systèmes d'information commence par une introduction claire et convaincante à travers ses trois premières pages, chacune ayant un rôle spécifique pour accueillir et informer les utilisateurs.

Ces trois pages d'introduction jouent un rôle crucial en posant les bases de l'application. Elles accueillent les utilisateurs, expliquent les avantages économiques et opérationnels et décrivent les fonctionnalités clés, permettant ainsi une compréhension rapide et une adoption facile de FixMate.



Figure 40 : Page introductive 1 de FixMate



Figure 41 : Page introductive 2 de FixMate

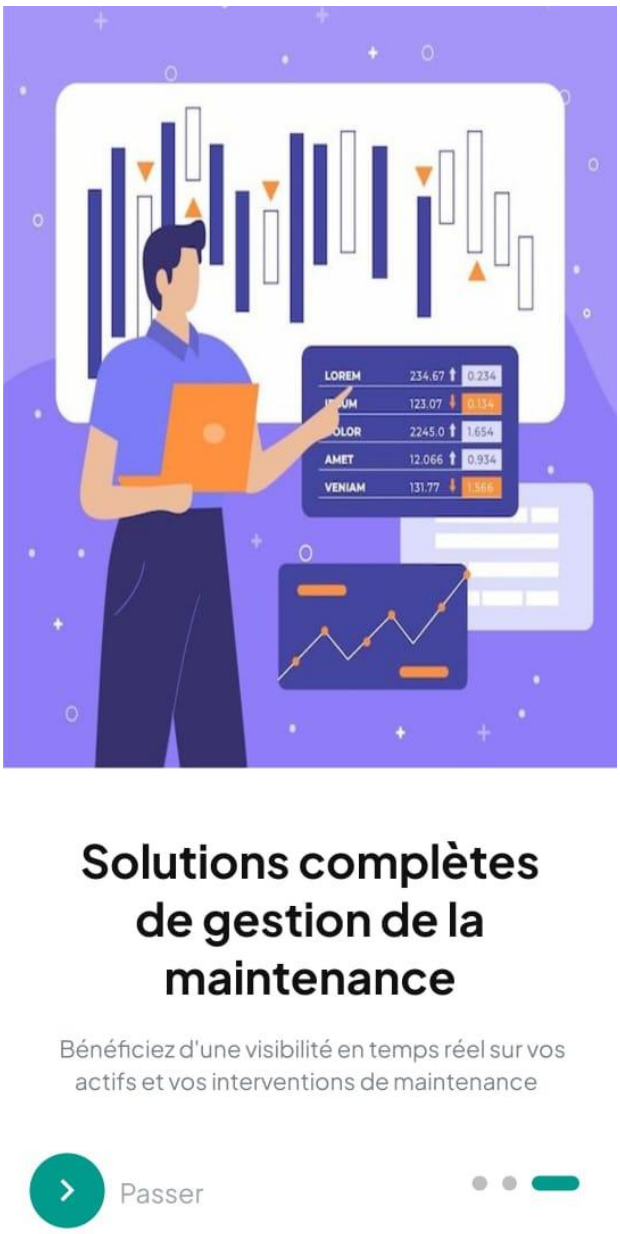


Figure 42 : Page introductive 3 de FixMate

Cette page joue un rôle crucial en incitant les utilisateurs à passer à l'action après avoir compris les avantages et les fonctionnalités de l'application. Elle réaffirme l'engagement de FixMate à simplifier la gestion des systèmes d'information tout en soulignant les bénéfices attendus tels que l'optimisation des opérations et la réduction des coûts. Le bouton "Commencez maintenant" encourage les utilisateurs à se connecter et à débuter leur utilisation de l'application, facilitant ainsi une transition fluide vers l'interface principale de gestion des contrôles.



Figure 43 : Page de début d'opération de contrôle

Cette page web est une page de connexion permettant aux utilisateurs d'accéder à leur compte. Elle est composée d'un formulaire de connexion qui demande à l'utilisateur d'entrer son adresse e-mail et son mot de passe. Le formulaire propose également une case à cocher pour permettre à l'utilisateur de se souvenir de sa préférence de connexion et une option pour réinitialiser le mot de passe en cas d'oubli.



Figure 44 : Page d'authentification

Une fois l'authentification réussie, les utilisateurs reçoivent un message de confirmation.

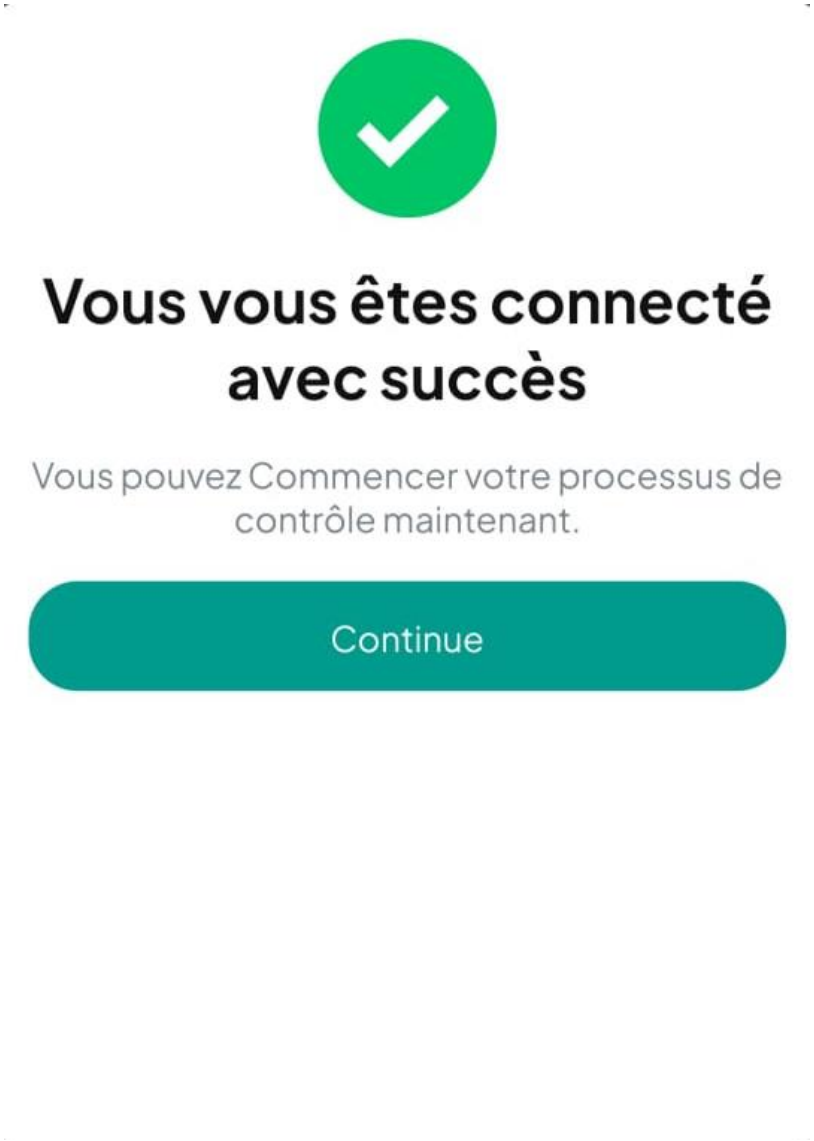


Figure 45 : Page vous informe que vous avez bien s'authentifier

Conclusion

L'utilisation combinée de ces technologies et outils modernes nous a permis de développer des applications web et mobiles de haute qualité, offrant une expérience utilisateur optimale et cohérente sur toutes les plateformes. Le choix de frameworks comme Vue.js et React Native a facilité le développement réactif et interactif, tandis que des outils comme Bootstrap, Vuetify, et Vite.js ont accéléré et optimisé le processus. Enfin, l'intégration de solutions comme Vue I18n a permis de rendre l'application accessible à une audience internationale, répondant ainsi aux besoins variés de nos utilisateurs. Grâce à cette approche, nous avons pu livrer des produits finaux robustes, modernes et prêts à répondre aux exigences du marché actuel.

Conclusion générale

Le projet FixMate représente une étape importante dans l'amélioration de la gestion et du contrôle des systèmes d'information. En combinant une application web robuste et une application mobile intuitive, nous avons créé une solution intégrée capable de répondre aux besoins complexes des entreprises modernes en matière de surveillance et de maintenance des infrastructures numériques.

Tout au long de ce projet, nous avons mis l'accent sur la performance, la sécurité et la convivialité. Nous avons développé des fonctionnalités essentielles telles que le suivi en temps réel, la gestion des droits d'accès, la recherche multicritères et l'envoi automatisé de notifications, tout en assurant une interface utilisateur intuitive et facile à utiliser. Ces efforts garantissent que FixMate peut offrir une expérience utilisateur fluide tout en répondant aux exigences fonctionnelles et non fonctionnelles spécifiées.

Nous avons également identifié et mitigé divers risques potentiels, notamment les retards de développement, les problèmes de performance et les vulnérabilités de sécurité. En adoptant une gestion de projet rigoureuse, en effectuant des tests approfondis et en mettant en œuvre des mesures de sécurité robustes, nous avons minimisé ces risques et assuré la fiabilité et la scalabilité de l'application.

La version mobile de FixMate, développée pour Android et IOS, permet aux utilisateurs de gérer les contrôles et d'accéder aux informations critiques même en déplacement, ce qui renforce encore l'utilité et l'accessibilité de la solution.

En conclusion, FixMate offre une solution complète et efficace pour la gestion et le contrôle des systèmes d'information, capable de s'adapter aux besoins évolutifs des entreprises. Nous sommes convaincus que cette application contribuera de manière significative à l'amélioration de la performance, de la sécurité et de la gestion des infrastructures numériques. Nous sommes fiers de ce travail accompli.

Bibliographie

https://www.php.net/

https://www.mysql.com/

https://reactnative.dev/

https://vuetifyjs.com/en/

https://vuejs.org/guide/

https://laravel.com/docs/11.x

https://en.wikipedia.org/wiki/Laravel

https://en.wikipedia.org/wiki/vuejs