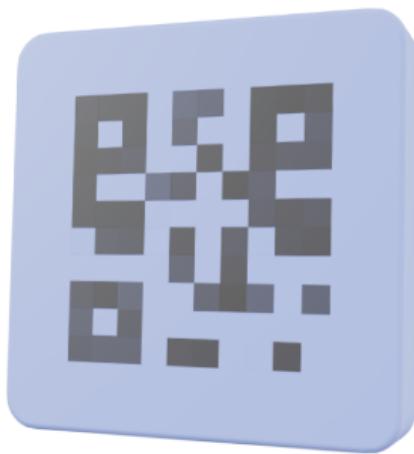


Cahier des charges

Feuille de présence numérique



Comment sécuriser et automatiser l'émargement universitaire tout en garantissant l'intégrité des données ?

Université Evry Paris-Saclay
IUT Evry | LP MIAW - DAWII

COLLETER Hyosua
COULIBALY Yahaya
NSOSSANI Emmanuelle



SOMMAIRE

1. INTRODUCTION

2. PRÉSENTATION DE L'ÉQUIPE

3. PÉRIMÈTRE DU PROJET

4. PERSONAE

5. ANALYSE DE L'EXISTANT

6. ANALYSE STRATÉGIQUE

7. SCÉNARIOS

8. SPÉCIFICATIONS FONCTIONNELLES

9. MODELISATION UML

10. DONNÉES ET ARCHITECTURE

11. TESTS ET VALIDATION

12. CONCLUSION

1. INTRODUCTION

1.1. Contexte

Actuellement, dans certaines universités, les feuilles d'émargement sont gérées manuellement : une impression pour chaque séance de cours, émargement manuscrit, traitement des feuilles par le secrétariat...

Notre application SYGMA vise à numériser la gestion de présence des étudiants de manière fiable et sécurisée.

1.2. Problématique

Comment sécuriser et automatiser l'émargement universitaire tout en garantissant l'intégrité des données ?

1.3. Objectifs généraux de SYGMA

- **Faciliter** la saisie et le suivi des présences et absences
- **Centraliser** les données de présence
- **Automatiser** les calculs tels que le taux de présence ou les absences cumulées.

1.4. Objectifs spécifiques de SYGMA

Pour les enseignants

- Accéder à une fiche personnelle
- Créer et gérer des cours et séances
- Saisir les présences et absences (manuel ou via QR Code)
- Modifier le statut des absences (présent, absent, retard, excusé)
- Ajouter des commentaires pédagogiques (retard, justification, remarque)
- Consulter l'historique des présences par cours et par élève
- Visualiser la liste des élèves présents à une séance donnée
- Ajouter ou retirer des élèves de manière ponctuelle et éphémère à un cours

Pour les gestionnaires pédagogiques

- Accéder à une fiche personnelle
- Créer et gérer les cours
- Corriger et modifier le statut des absences
- Gérer les justificatifs d'absences
- Ajouter des commentaires administratifs
- Consulter l'historique global des présences
- Ajouter ou supprimer des élèves
- Modifier le statut administratif des élèves (inscrit, suspendu, etc.)
- Exporter les données de présence
- Accéder aux données selon une périodicité définie

Pour les étudiants

- Accéder à une fiche personnelle
- Valider leur présence via l'application (QR Code)
- Consulter leur statut de présence et son historique (avec date de modification)
- Visualiser un dashboard personnel (retards, absences, cours suivis)
- Déclarer un retard
- Ajouter un justificatif d'absence

2. PRÉSENTATION DE L'ÉQUIPE

2.1. Rôles et responsabilités

Le projet est réalisé par une équipe de trois étudiants. Chaque membre dispose d'un rôle principal clairement défini, tout en collaborant activement aux différentes étapes du projet afin d'assurer la cohérence globale et la réussite de l'application.



Yahaya

Responsable Back-end

- **Développement**
 - Authentification & rôles
 - Gestion des utilisateurs
 - Logique métier (présence, absence, statuts)
 - API / contrôleurs
 - Sécurité & droits d'accès
- **Autres responsabilités**
 - Coordination technique
 - Validation du code
 - Support à l'équipe
 - Participation BDD



Emmanuelle

Responsable Front-end & UX
Cheffe de projet

- **Développement**
 - Interfaces utilisateur
 - Dashboards (enseignant / élève / admin)
 - Formulaires (présences, justificatifs)
 - Responsive design
 - Intégration des données
- **Autres responsabilités**
 - Maquettes (avant dev)
 - Parcours utilisateurs
 - Tests ergonomiques



Hyosua
Responsable Data &
fonctionnalités transverses

- **Développement**
 - Base de données (MCD, MLD)
 - Requêtes & statistiques
 - Export des données
 - Historique & traçabilité
 - QR Code (token, validité)
- **Autres responsabilités**
 - UML (classes, séquences)
 - RGPD & sécurité des données
 - Tests fonctionnels
 - Documentation technique

Commun

- Tests
- Corrections de bugs
- Améliorations
- Revue de code

2.2. Organisation et mode de travail

Pour la gestion et le suivi de notre projet, l'équipe a choisi d'adopter **la méthode agile**, permettant une approche itérative et flexible. Cette méthode favorise l'adaptation aux changements et assure un suivi régulier de l'avancement du projet.

Rythme de travail :

- L'équipe se réunit **toutes les deux semaines pour des points d'avancement**, permettant de faire le bilan des tâches réalisées, de planifier les prochaines étapes et d'identifier les obstacles rencontrés.

- Ces réunions favorisent le partage d'informations et la coordination entre les différents membres de l'équipe.

Outils utilisés :

 Trello	 GitHub	 Drive	 Discord
Pour le suivi des tâches, la planification des sprints et la gestion des priorités.	Pour le versioning et la gestion du code.	Pour le partage et la centralisation des documents et supports du projet.	Pour la communication rapide et quotidienne entre les membres de l'équipe.

3. PÉRIMÈTRE DU PROJET

3.1. Périmètre fonctionnel

Le périmètre fonctionnel définit les fonctionnalités et services que le projet doit inclure. Pour notre projet, cela comprend :

- **Gestion des utilisateurs** : création, modification, suppression des comptes (étudiants, enseignants, administrateurs).
- **Gestion des étudiants et groupes** : ajout, modification, affectation à des cours ou groupes.
- **Gestion des cours et des séances** : planification des séances, création et modification des événements.
- **Émargement des présences** : via QR Code et de manière manuelle si nécessaire.
- **Consultation et export des présences** : génération de rapports et suivi des statistiques.

- **Calculs automatiques** : décompte des présences, taux de présence par cours ou étudiant.

3.2. Hors périmètre

Le hors périmètre définit ce qui n'est pas prévu dans le projet pour éviter les dérives et clarifier les limites :

- Gestion financière ou comptable des cours (paiements, factures).
- Intégration avec des systèmes externes de l'école ou université.
- Développement d'une application mobile native (le projet est prévu pour le web uniquement).
- Gestion avancée des permissions ou profils complexes non nécessaires pour le fonctionnement de base.

3.3. Hypothèses de travail

Les hypothèses correspondent aux **conditions supposées pour que le projet fonctionne**, mais qui ne sont pas garanties :

- Tous les utilisateurs disposent d'un appareil avec connexion Internet pour accéder à l'application.
- Les étudiants et enseignants savent utiliser des outils numériques simples (QR Code, interface web).

4. ANALYSE DES BESOINS

4.1. Expression du besoin

Actuellement, la gestion des présences des étudiants est réalisée **manuellement**, ce qui entraîne :

- des erreurs dans le suivi des absences et retards,
- une perte de temps pour les enseignants et le personnel administratif,
- un accès limité aux statistiques sur la participation des étudiants.

Le projet a pour objectif de **numériser et automatiser la gestion des présences** afin de :

- simplifier l'émargement via QR Code ou manuel,
- fournir des rapports fiables et consultables à tout moment,
- faciliter la planification et le suivi des séances.

4.2. Besoins utilisateurs

Pour les enseignants :

- Pouvoir créer et gérer facilement leurs séances,
- Émarger rapidement les étudiants via QR Code ou manuellement,
- Consulter les présences et générer des rapports.

Pour les étudiants :

- Signer leur présence facilement avec un QR Code,
- Avoir la confirmation de leur présence enregistrée.

4.3. Besoins métiers

- **Fiabiliser** le suivi des présences et réduire les erreurs humaines,
- **Accélérer** les processus administratifs liés aux absences et retards,
- Permettre une meilleure **planification pédagogique** et un suivi statistique des étudiants.

4.4. Besoins techniques

- Application web accessible depuis desktop et mobile,
- Base de données sécurisée et centralisée pour stocker les présences et les informations utilisateurs,
- Interface simple et ergonomique,
- Gestion sécurisée des accès avec différents profils (étudiant, enseignant, administrateur),
- Export des données au format CSV ou PDF.

4.5. Priorisation des besoins (MoSCoW)

MoSCoW	Émargement via QR Code	Gestion des utilisateurs	Consultation et export des présences	Émargement manuel
Must (indispensable)				
Should (important mais pas critique)				
Could (optionnel)				
Won't (hors périmètre)				

MoSCoW	Calcul automatique du taux de présence	Notifications aux étudiants	Application mobile native	Intégration avec systèmes externes
Must (indispensable)				
Should (important mais pas critique)				
Could (optionnel)				
Won't (hors périmètre)				

5. PERSONAE



Mme. Dupont

Enseignante

Nom : Mme. Dupont
Âge : 42 ans
Rôle : Enseignante

Objectifs :

- Gérer rapidement l'appel en début de cours
- Disposer d'un suivi fiable des présences

Besoins :

- Outil simple et rapide
- Accès immédiat à la liste des étudiants
- Peu de manipulation pendant le cours

Freins / difficultés :

- Perte de temps avec les feuilles papier
- Risque d'erreurs manuelles

Attentes vis-à-vis de l'application :

- Gain de temps
- Fiabilité des données
- Simplicité d'utilisation



KEVIN

Étudiant

Nom : Kevin
Âge : 20 ans
Rôle : Étudiant

Objectifs :

- Valider facilement sa présence
- Suivre ses absences et retards

Besoins :

- Application accessible sur smartphone
- Procédure rapide
- Accès clair à son historique

Freins / difficultés :

- Oublis d'émargement
- Manque de visibilité sur son statut

Attentes vis-à-vis de l'application :

- Simplicité
- Transparence
- Rapidité

6. ANALYSE DE L'EXISTANT

6.1. Description du système actuel

Actuellement, le suivi des présences est réalisé de manière **manuelle**, à l'aide de **feuilles d'émargement papier**.

Pour chaque séance :

- L'enseignant procède à l'appel ou fait circuler une feuille.
- Les étudiants signent leur présence.
- Les informations sont ensuite conservées sous format papier ou saisies ultérieurement dans un fichier (Excel ou autre outil).

Ce mode de fonctionnement repose essentiellement sur :

- des documents **physiques**,
- des saisies **manuelles**,
- un **traitement non automatisé** des données.

6.2. Limites et problèmes identifiés

Le système actuel présente plusieurs limites :

- **Perte de temps** lors de l'appel en début de cours.
- **Risque d'erreurs humaines** (oublis, mauvaise saisie, signatures illisibles).
- **Difficulté de centralisation** des données (multiplication des supports).
- **Manque de traçabilité** (difficile de savoir quand et par qui une information a été modifiée).
- **Risque de perte** ou de dégradation des documents papier.
- **Traitemet long** des statistiques (calculs manuels des taux de présence).
- **Manque de réactivité** en cas de contrôle ou de demande administrative.

6.3. Opportunités d'amélioration

La mise en place d'une solution numérique permet :

- **L'automatisation** de la saisie des présences.
- **La centralisation** des données dans une base unique.
- Un **gain de temps** significatif pour les enseignants.
- Une **fiabilité** accrue des informations.
- Un **accès rapide** à l'historique et aux statistiques.
- Une meilleure **traçabilité** des actions (horodatage, auteur de la saisie).
- Une réduction de l'usage du papier (démarche écoresponsable).

6.4. Comparaison : système manuel vs solution numérique

Critère	Système papier	Solution numérique
Temps de saisie	Long	Rapide (QR Code / formulaire)
Fiabilité	Moyenne (erreurs possibles)	Élevée (règles automatiques)
Centralisation	Dispersée	Centralisée
Traçabilité	Faible	Complète (historique, horodatage)
Calcul des statistiques	Manuel	Automatisé
Sécurité des données	Faible	Renforcée (accès sécurisé)
Accessibilité	Limitée	Multi-support (PC / mobile)
Impact écologique	Moyen	Réduit

L'analyse de l'existant met en évidence les limites d'un système manuel peu adapté aux exigences de fiabilité, de rapidité et de traçabilité. La solution numérique proposée répond à ces enjeux en modernisant la gestion des présences.

7. ANALYSE STRATÉGIQUE

7.1. Analyse PESTEL

Politique	De plus en plus de démarches scolaires passent par le numérique, les établissements sont incités à moderniser leurs outils.
Économique	La solution permet de réduire les coûts liés au papier et au traitement administratif des présences.
Sociologique	Les utilisateurs ont l'habitude d'utiliser des outils numériques dans leur quotidien.
Technologique	Les technologies web et mobiles actuelles permettent une solution fiable et accessible.
Écologique	La dématérialisation contribue à la réduction de la consommation de papier.
Légal	Le projet doit respecter la réglementation sur la protection des données personnelles (RGPD).

7.2. Analyse SWOT

FORCES	FAIBLESSES
<ul style="list-style-type: none">• Solution numérique moderne• Gain de temps pour les enseignants• Centralisation et fiabilité des données• Automatisation des calculs de présence• Accessibilité multi-support (PC / mobile)	<ul style="list-style-type: none">• Dépendance à la connexion internet• Nécessité d'un accompagnement au changement• Courbe d'apprentissage pour certains utilisateurs
OPPORTUNITÉS	MENACES
<ul style="list-style-type: none">• Forte demande de digitalisation dans l'éducation• Possibilité d'évolution (statistiques avancées, intégration ENT, etc.)• Valorisation pédagogique du projet• Déploiement possible dans différents établissements	<ul style="list-style-type: none">• Résistance au changement de certains utilisateurs• Risques techniques (bugs, indisponibilité du service)• Contraintes réglementaires renforcées (données personnelles)• Concurrence d'outils existants sur le marché

8. SCÉNARIOS

8.1. Scénarios principaux

Scénario 1 : Connexion utilisateur

Acteur : Étudiant / Enseignant / Gestionnaire

Objectif : Accéder à l'application

Précondition : L'utilisateur possède un compte actif

Déroulement :

- a. L'utilisateur accède à la page de connexion.
- b. Il saisit son identifiant et son mot de passe.
- c. Le système vérifie les informations.
- d. L'utilisateur est redirigé vers son tableau de bord.

Résultat : L'utilisateur est authentifié et peut utiliser la plateforme.

Scénario 2 : Création d'une séance

Acteur : Enseignant

Objectif : Créer une séance de cours

Déroulement :

- a. L'enseignant sélectionne un cours.
- b. Il crée une nouvelle séance (date, heure, groupe).
- c. Le système enregistre la séance.

Résultat : La séance est enregistré dans le système

Scénario 3 : Émargement par QR Code

Acteur : Étudiant / Enseignant

Objectif : Enregistrer automatiquement la présence

Préconditions :

- La séance est ouverte.
- Le QR Code est affiché par l'enseignant.

Déroulement :

- a. L'enseignant génère le QR Code.
- b. Le QR Code est projeté en classe.
- c. L'étudiant le scanne depuis son appareil.
- d. Le système vérifie la validité (temps + localisation).
- e. La présence est enregistrée.

Résultat : L'étudiant est marqué Présent.

Scénario 4 : Émargement manuel

Acteur : Enseignant

Objectif : Marquer manuellement la présence

Déroulement :

- a. L'enseignant ouvre la liste des étudiants.
- b. Il coche les présents / absents / retard.
- c. Le système enregistre les modifications.

Résultat : Les présences sont mises à jour manuellement.

Scénario 5 : Consultation des présences

Acteur : Enseignant / Gestionnaire

Objectif : Consulter et exporter les présences

Déroulement :

- a. L'utilisateur sélectionne une séance ou un cours.
- b. Le système affiche la feuille de présence.
- c. L'utilisateur peut exporter les données.

Résultat : Les données sont consultables et exploitables.

8.2. Scénarios alternatifs

Scénario A : QR Code expiré

Acteur : Étudiant

- a. L'étudiant scanne un QR Code expiré.
- b. Le système refuse l'émargement.
- c. Un message d'erreur est affiché.
- d. L'étudiant doit utiliser l'émargement manuel.

Scénario B – Tentative de fraude

Acteur : Étudiant

- a. L'étudiant tente de scanner le QR Code hors salle ou hors temps.
- b. Le système détecte l'anomalie.
- c. L'émargement est refusé et signalé à l'enseignant.

Scénario C – Étudiant en retard / absent excusé

Acteur : Étudiant / Gestionnaire

- a. L'étudiant est marqué en retard ou absent.
- b. Il envoie un justificatif.
- c. Le gestionnaire reçoit une notification.
- d. Il valide ou refuse le justificatif.
- e. Le statut est mis à jour.

9. SPÉCIFICATIONS FONCTIONNELLES

9.1. Gestion des utilisateurs

Description :

Le système permet la gestion des comptes utilisateurs selon leur rôle (Étudiant, Enseignant, Gestionnaire).

Acteurs concernés : Administrateur / Gestionnaire

Entrées :

- Nom, prénom
- Email
- Rôle
- Identifiants de connexion

Sorties :

- Création, modification, suppression de comptes
- Attribution des droits d'accès

Règles de gestion :

- Chaque utilisateur possède un rôle unique.
- L'accès aux fonctionnalités dépend du rôle.
- L'authentification est obligatoire pour accéder à l'application.

9.2. Gestion des étudiants

Description :

Gestion des informations liées aux étudiants.

Acteurs concernés : Gestionnaire / Enseignant

Entrées :

- Identité de l'étudiant
- Groupe, formation

Sorties :

- Liste des étudiants
- Mise à jour des informations

Règles de gestion :

- Un étudiant est rattaché à un ou plusieurs groupes.
- Les données doivent être modifiables uniquement par les profils autorisés.

9.3. Gestion des cours et groupes

Description :

Création et administration des cours et des groupes.

Acteurs concernés : Gestionnaire / Enseignant

Entrées :

- Nom du cours
- Enseignant responsable
- Groupe(s) associé(s)

Sorties :

- Liste des cours
- Affectation des étudiants

Règles de gestion :

- Un cours est associé à au moins un groupe.
- Un enseignant peut gérer uniquement ses propres cours.

9.4 Gestion des séances

Description :

Création et planification des séances de cours.

Acteurs concernés : Enseignant

Entrées :

- Date, heure, durée
- Cours, groupe

Sorties :

- Séance créée
- QR Code généré

Règles de gestion :

- Une séance ne peut être ouverte qu'à une date valide.
- Le QR Code est actif uniquement pendant la durée définie.

9.5 Gestion des présences

Description :

Enregistrement et suivi des présences des étudiants.

Acteurs concernés : Étudiant / Enseignant / Gestionnaire

Entrées :

- Scan QR Code
- Saisie manuelle

Sorties :

- Statut : Présent / Retard / Absent / Excusé

Règles de gestion :

- Un étudiant ne peut émarger qu'une seule fois par séance.
- Toute modification est tracée.

9.6 Calculs automatiques

Description :

Le système calcule automatiquement les statistiques de présence.

Acteurs concernés : Enseignant / Gestionnaire

Sorties :

- Taux de présence par étudiant
- Taux de présence par cours
- Nombre d'absences et retards

Règles de gestion :

- Les calculs sont mis à jour en temps réel.
- Les données sont basées uniquement sur les séances validées.

9.7 Export et consultation des données

Description :

Consultation et extraction des données de présence.

Acteurs concernés : Enseignant / Gestionnaire

Entrées :

- Sélection d'un cours ou d'une séance

Sorties :

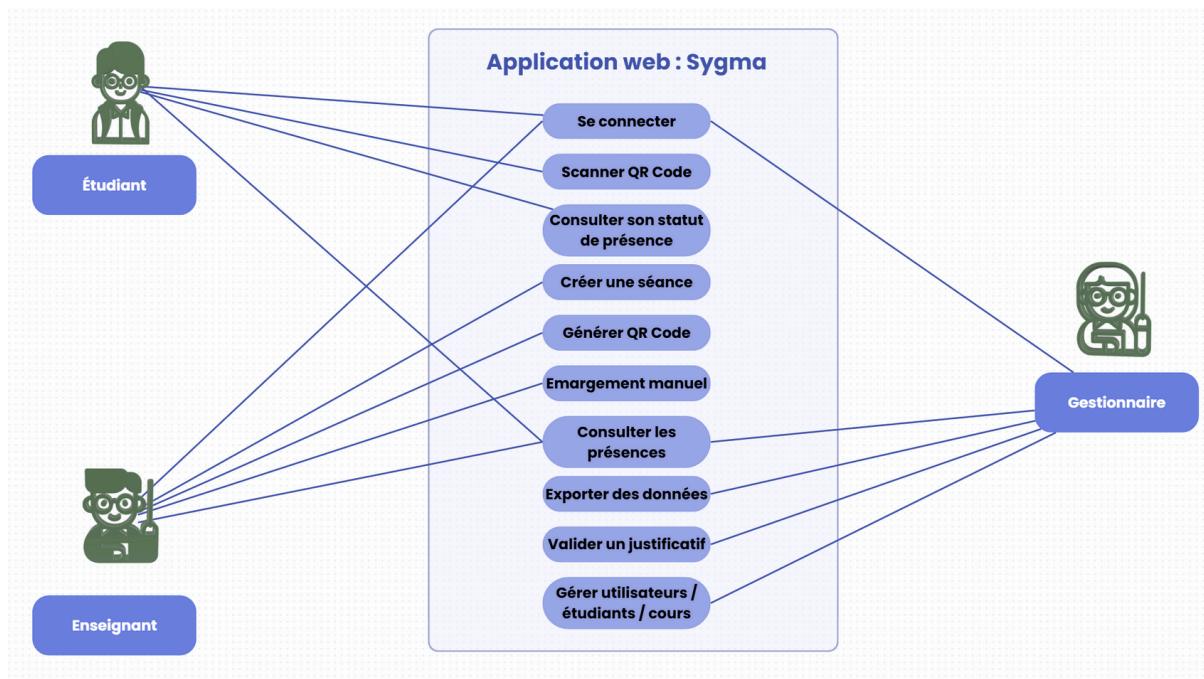
- Affichage des feuilles de présence
- Export en CSV / PDF

Règles de gestion :

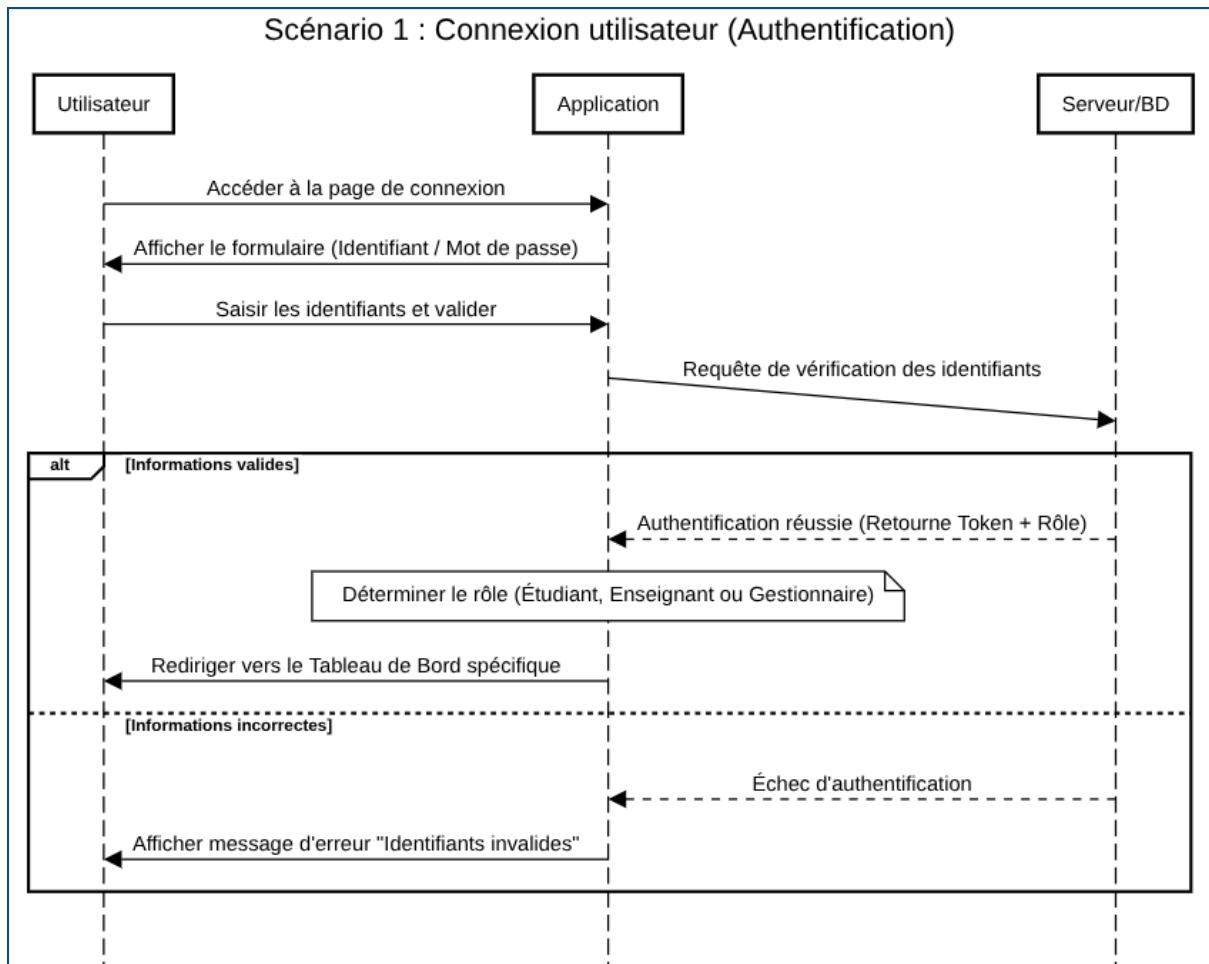
- Les exports respectent les règles RGPD.

10. MODELISATION UML

10.1. Diagramme de cas d'utilisation



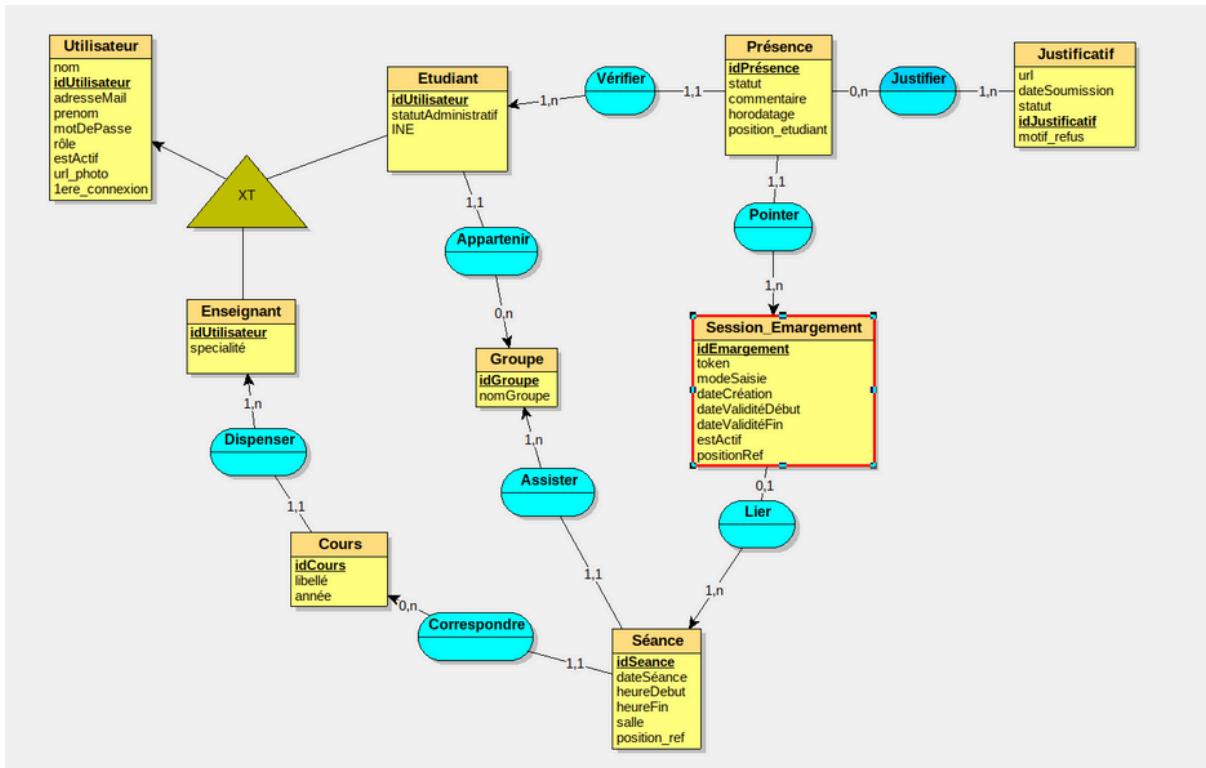
10.2. Diagramme de séquence



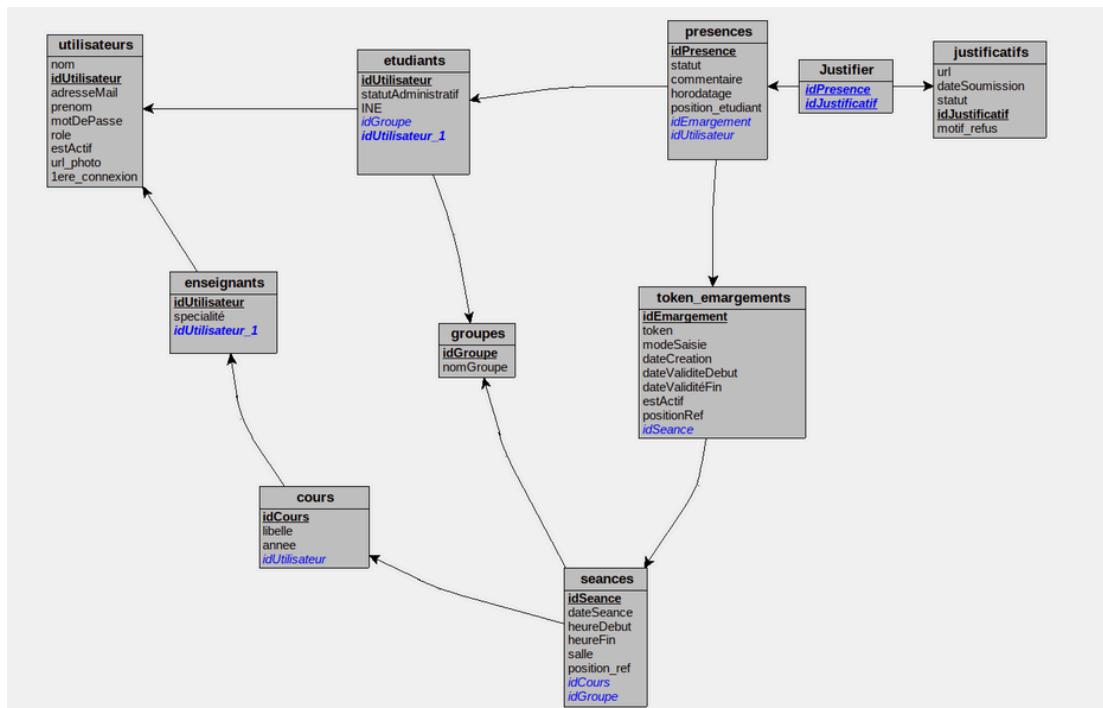
L'ensemble des diagrammes à trouver en annexe.

11. DONNÉES ET ARCHITECTURE

11.1. Modèle conceptuel de données (MCD)



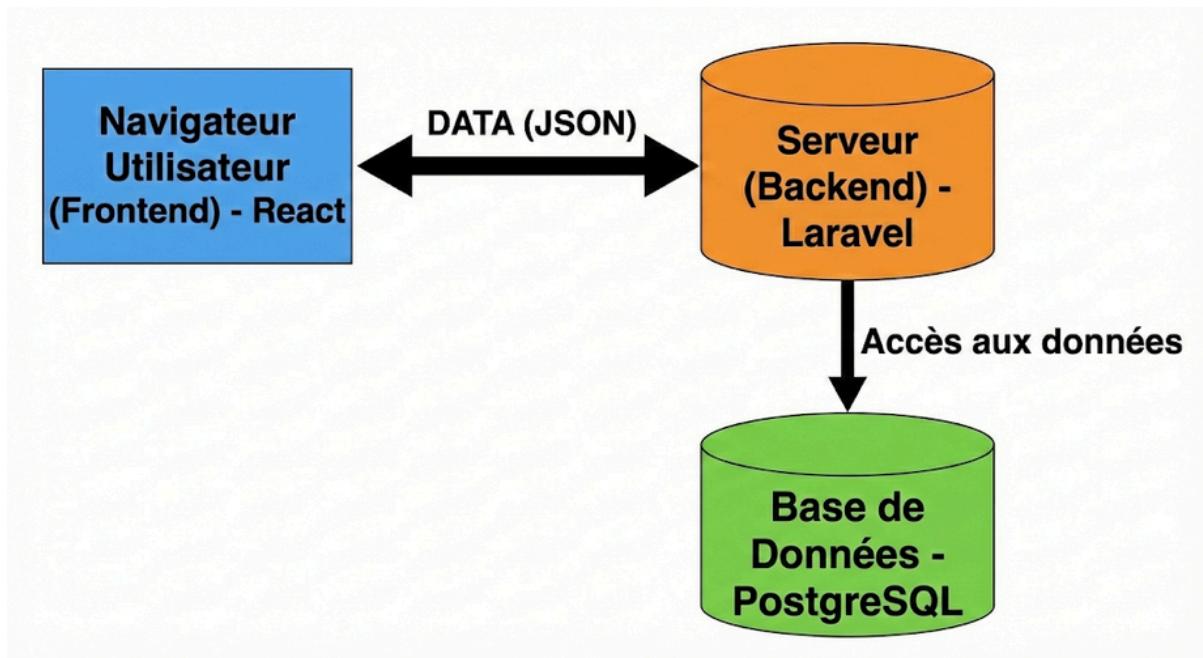
11.2. Modèle logique de données (MLD)



11.3. Architecture applicative

Architecture SPA (Single Page Application) avec API REST :

Cette approche sépare la partie **client (Front-end)** de la partie **Back-end (logique métier et accès à la base de données)**. Cette solution favorise l'**évolutivité** de l'application (ex. : si l'on souhaite passer à une **version mobile native**).



Architecture MVC (coté Back-End):

Permet de **séparer clairement les responsabilités** et d'organiser efficacement le code.

M — Modèle (Model)

Représentation des données et logique métier

Dialogue avec la **BDD (Base de Données)**

L'**ORM Eloquent de Laravel** permet d'effectuer les opérations sur les tables (**Select, Insert, Update, Delete...**)

Définit les **relations entre les entités**

Ex :

Le modèle "**Cours**" déclare qu'il a **plusieurs sessions de cours**.

Le modèle "**Étudiant**" déclare qu'il **appartient à plusieurs cours**.

Contient des **fonctions qui manipulent les données**

Ex : fonction **estTerminee()** qui renvoie **true** si la **date de fin d'une séance**

est dépassée.

C — Contrôleur (Controller)

Chef d'orchestre entre le **Modèle** et la **Vue**

Reçoit la **requête de l'utilisateur** et la redirige vers une **méthode spécifique**

Demande au **Modèle** de **récupérer ou modifier les données**

Une fois la réponse du Modèle reçue, il la **transmet à la Vue**

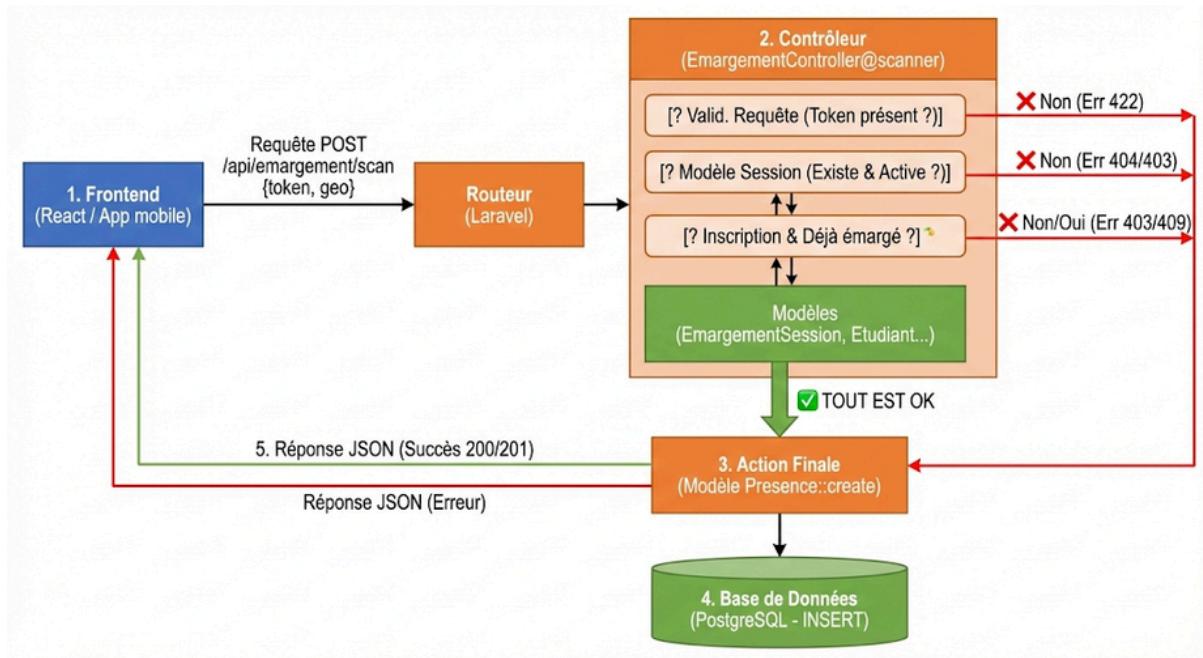
V — Vue (View)

Affichage destiné à l'utilisateur

Dans notre cas, le rôle de la **Vue** est pris en charge par la partie **React (Front-end)**.

Côté Back-end, la "Vue" correspond simplement aux **données formatées en JSON** qui sont **envoyées à React**.

Exemple: requête de Scan QRCode



11.4. Technologies envisagées

Front :

- **React** (Interfaces) avec Node.js pour configurer les outils de développement(npm, ESLINT...)
 - Permet de construire des interfaces utilisateurs dynamiques, réactives et modulaires
- **React-Router-Dom** (routes)
- **html5-qrcode** (Scan des QRCodes)

Back-end :

- **PHP** 8.2+ avec **Laravel** 11, (API, logique métier)
 - Laravel est une bibliothèque très complète avec des composants intégrés(authentification, gestion de requêtes HTTP...)
- **Apache** (serveur web)
- simple-qrcode (génération QRcode)
 - Packet conçu pour laravel pour la génération sécurisée de QRCode
- spatie-laravel/permission (gestion des rôles et permissions utilisateurs)
- maatwebsite/excel (Import/Export Excel)

BDD :

- **PostgreSql** 16
 - Système de Gestion de BDD 100% OpenSource, réputé pour sa fiabilité et robustesse (intégrité des données, Gestion des concurrences, requêtes complexes)
 - Possibilité ensuite d'ajouter l'extension PostGIS pour la gestion des données géospaciales (géolocalisation du QRCode)

Environnement de Développement:

- **Docker/Docker Compose**
 - permet de conteneuriser l'environnement et éviter les conflits entre les différentes configs des développeurs
- **Git** (Gestion de Versions)
 - pour une collaboration structurée, avec tracabilité (branches, pull requests)

12. Exigences non fonctionnelles

12.1. Ergonomie & UX

- Interface claire, intuitive et facile à prendre en main.
- Navigation simple avec un nombre limité de clics pour accéder aux fonctions principales.
- Design responsive et lisible.
- Messages d'erreur compréhensibles pour l'utilisateur.
- Temps d'apprentissage réduit.

12.2. Performance

- Temps de chargement des pages inférieur à 3 secondes.
- Enregistrement des présences en temps quasi réel.
- Capacité à supporter plusieurs connexions simultanées.
- Traitement rapide des exports et des statistiques.

12.3. Sécurité

- Authentification sécurisée (identifiant + mot de passe).
- Gestion des droits selon les rôles.
- Protection contre les accès non autorisés.
- Traçabilité des actions sensibles (modifications de présences).
- Chiffrement des données sensibles.

12.4. Disponibilité & sauvegarde

- Application accessible en continu (hors maintenance).
- Sauvegarde régulière de la base de données.
- Possibilité de restauration en cas de perte de données.
- Fiabilité du stockage des informations.

12.5 Compatibilité (desktop / mobile)

- Application accessible depuis les principaux navigateurs (Chrome, Firefox, Edge, Safari).
- Fonctionnement sur ordinateur, tablette et smartphone.
- Interface adaptée aux différentes tailles d'écran.

12.6 RGPD & protection des données

- Collecte uniquement des données nécessaires.
- Accès restreint aux données personnelles.
- Conservation limitée dans le temps.
- Possibilité de modification/suppression des données.
- Information claire des utilisateurs sur l'usage de leurs données.

13. Gestion des risques

13.1. Identification des risques

Les principaux risques identifiés pour ce projet sont :

- Retard dans le développement
- Problèmes techniques (QR Code, base de données, performances)
- Mauvaise répartition des tâches dans l'équipe
- Non-respect des obligations RGPD
- Manque de communication entre les membres du groupe
- Indisponibilité de certains membres de l'équipe

13.2. Analyse des risques

Risque	Probabilité	Impact	Niveau
Retard de développement	Moyennne	Élevé	Critique
Problèmes techniques	Moyennne	Élevé	Critique
Mauvaise répartition des tâches	Moyennne	Moyen	Important
Non-respect du RGPD	Faible	Élevé	Critique
Manque de communication	Moyennne	Moyen	Important

13.3. Plan de prévention

- Planification détaillée dès le début du projet.
- Utilisation de Trello pour le suivi des tâches.
- Réunions de suivi toutes les deux semaines.
- Répartition claire des rôles et responsabilités.
- Respect des bonnes pratiques de sécurité et RGPD.
- Tests réguliers des fonctionnalités.

13.4. Plan de mitigation

Risque	Action corrective
Retard de développement	Réorganisation du planning, priorisation des fonctionnalités.
Problèmes techniques	Support entre membres, recherche de solutions alternatives, simplification technique.
Mauvaise répartition des tâches	Rééquilibrage des tâches, entraide au sein de l'équipe.
Non-respect du RGPD	Correction immédiate, suppression des données non conformes.
Manque de communication	Augmentation des points d'échange et rappels réguliers.

14. PLANNING PREVISIONNEL

14.1. Découpage des phases

Analyse

- Recueil et analyse des besoins
- Rédaction du cahier des charges

Conception

- Maquettes
- UML
- Modélisation des données

Développement

- Front-end
- Back-end
- Base de données

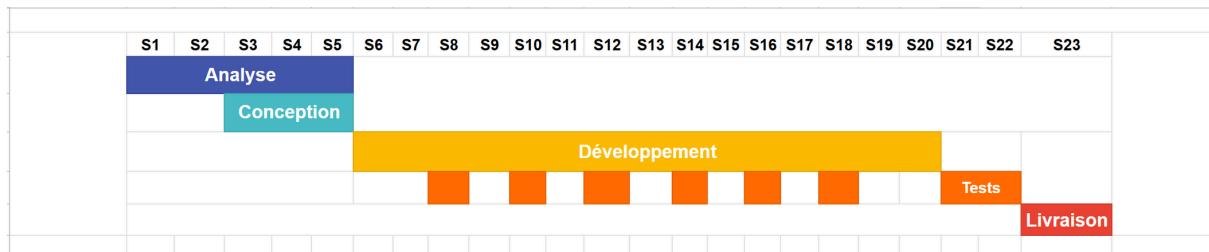
Tests

- Tests unitaires
- Tests fonctionnels
- Corrections

Livraison

- Mise en production
- Documentation
- Présentation finale

14.2. Diagramme de Gantt



4.3. Jalons du projet

- Validation du cahier des charges
- Validation des maquettes et UML
- Prototype fonctionnel
- Version finale testée
- Livraison du projet

15. TESTS ET VALIDATION

15.1. Stratégie de test

La stratégie de test vise à garantir la qualité, la fiabilité et la conformité de l'application avant sa livraison. Les tests sont réalisés tout au long du projet, et de manière plus intensive en fin de développement.

15.2. Tests unitaires

Vérification du bon fonctionnement de chaque fonctionnalité prise individuellement.

Exemples :

- Connexion utilisateur
- Génération du QR Code
- Enregistrement d'une présence

15.3. Tests fonctionnels

Vérification du respect des exigences du cahier des charges.

Exemples :

- Création d'une séance complète
- Émargement QR Code et manuel
- Consultation et export des présences

15.4. Tests utilisateurs

Tests réalisés par des utilisateurs représentatifs (enseignants, étudiants, gestionnaires).

Objectif : valider l'ergonomie et la compréhension de l'outil.

Recueil des retours pour amélioration.

15.5. Critères de validation

L'application est considérée comme validée si :

- Toutes les fonctionnalités principales fonctionnent.
- Les données sont fiables et sécurisées.
- Les performances sont satisfaisantes.
- Les exigences RGPD sont respectées.
- Les utilisateurs peuvent utiliser l'outil sans difficulté majeure.

16. LIVRABLES

À l'issue du projet, l'équipe s'engage à fournir les livrables suivants :

- Cahier des charges
- Maquettes / Wireframes de l'application
- Diagrammes UML (cas d'utilisation, classes, séquence, activités)
- Code source

- Base de données
- Documentation utilisateur
- Documentation technique (architecture, installation, maintenance)
- Présentation finale du projet

17. CONCLUSION

17.1. Synthèse du projet

Ce projet consiste à concevoir et développer une application web de gestion des présences basée sur l'utilisation d'un QR Code.

Il répond à un besoin réel d'optimisation du suivi des étudiants et de modernisation des pratiques pédagogiques.

L'ensemble du projet a été pensé de manière structurée, depuis l'analyse des besoins jusqu'à la définition des livrables, en intégrant des aspects fonctionnels, techniques, organisationnels et réglementaires.

17.2. Bénéfices attendus

- Gain de temps pour les enseignants et le personnel administratif.
- Fiabilité accrue des données de présence.
- Réduction des erreurs et des fraudes.
- Amélioration du suivi pédagogique des étudiants.
- Centralisation et traçabilité des informations.

17.3. Apports pédagogiques pour l'équipe

Ce projet permet à l'équipe de :

- Mettre en pratique les notions de gestion de projet.
- Développer des compétences en conception et développement web.
- Travailler en équipe avec une méthodologie professionnelle (Agile).
- Apprendre à produire une documentation technique et fonctionnelle complète.