

Système IoT de Gestion Automatique de Présence



Par Empreinte Digitale avec ESP32 et Visualisation en Temps Réel



Réalisé par :

Bouba Ahmed & Lkhaldi Mohamed



Encadré par :

Pr. Omari Slimane

Master 2 - Systèmes Intelligents pour l'Éducation



Meknès, le February 6, 2026

Année Universitaire 2025-2026

Plan de la présentation

2

- 1. Introduction & Contexte**
- 2. Problématique & Objectifs**
- 3. Solution Proposée**
- 4. Architecture Technique**
- 5. Flux de Fonctionnement**

- 6. Choix Technologiques**
- 7. Résultats & Apports**
- 8. Limites & Améliorations**
- 9. Conclusion**



CHAPITRE 1

Introduction & Contexte

1



Le Défi Actuel

Les limites des méthodes traditionnelles de présence

3

⚠ Problèmes Critiques

- ▶ **Temps perdu** : 5-15 min par séance pour l'appel
- ▶ **Fraudes fréquentes** : Signatures pour autrui
- ▶ **Retards ignorés** : Seuls les absents comptabilisés
- ▶ **Erreurs humaines** : Oublis, mauvaises interprétations
- ▶ **Aucune traçabilité** : Données papier non exploitables

↗ Impact Éducatif

- ▶ **Perte d'attention** : Appel interrompt le cours
- ▶ **Données non fiables** : Statistiques impossibles
- ▶ **Charge administrative** : Saisie manuelle répétitive
- ▶ **Absence d'analyse** : Pas de suivi des tendances
- ▶ **Désorganisation** : Fichiers multiples, versionnage

Contexte Universitaire

Pourquoi l'automatisation est-elle nécessaire ?

4

III Environnement Académique

- ▶ **Classes nombreuses** : 50-200 étudiants par groupe
- ▶ **Emplois du temps denses** : Cours de 1h30 à 3h
- ▶ **Multi-salles** : Plusieurs amphis/laboratoires
- ▶ **Gestion complexe** : Groupes, sections, filières
- ▶ **Besoin de reporting** : Statistiques pour décisions

◎ Objectifs Pédagogiques

- ▶ Récupérer 100% du temps d'enseignement
- ▶ Garantir 100% de fiabilité des données
- ▶ Générer des statistiques en temps réel
- ▶ Faciliter le travail administratif

Transition vers une solution automatisée

De la gestion manuelle à l'intelligence embarquée

5

Méthode Manuelle

- Appel oral chronophage
- Feuilles de présence papier
- Saisie Excel manuelle
- Risque élevé de fraude

TRANSITION NUMÉRIQUE



Solution Automatisée

- Biométrie (empreinte digitale)
- Enregistrement automatique
- Dashboard temps réel
- Données sécurisées cloud

 ≈ 5-15 min / séance

 < 2 sec / étudiant

 Gain de productivité : 95% de temps récupéré pour l'enseignement



CHAPITRE 2

Problématique & Objectifs

2



Système IoT de Présence → February 6, 2026

Problématique

Le cœur du défi à résoudre

6

Comment assurer une gestion fiable, rapide et automatisée de la présence des étudiants ?



Fiabilité

- ✓ Authentification
- ✓ Données sûres
- ✓ Traçabilité



Rapidité

- ✓ Temps réel
- ✓ Flux fluide
- ✓ Sans attente



Automatisation

- ✓ Autonomie
- ✓ Zéro papier
- ✓ Sans erreur



Triple défi : Authentifier sans erreur, enregistrer sans délai, gérer sans effort.

Objectifs du projet

Répondre aux enjeux d'accessibilité et d'autonomie

7

Objectifs Techniques

► Automatiser la prise de présence

Système IoT avec reconnaissance biométrique

► Réduire la fraude

Authentification par empreinte digitale

► Centraliser les données

Stockage cloud avec Firebase

► Interface temps réel

Dashboard web avec Flask/React

Objectifs Pédagogiques

► Gain de temps enseignant

Récupération du temps d'enseignement

► Données fiables

Prise de décision basée sur des stats précises

► Simplicité d'utilisation

Interface intuitive pour tous les acteurs

► Scalabilité

Adaptable à plusieurs salles et groupes



CHAPITRE 3

Solution Proposée

3



Système IoT de Présence → February 6, 2026

Écosystème Complet du Système

Une architecture intégrée IoT + Cloud + Web

8

Couche Matérielle

- ▶ **ESP32 DevKit** : Microcontrôleur WiFi
- ▶ **Capteur AS608** : Biométrie
- ▶ **Interface** : Écran LCD + Buzzer

 DISPOSITIF EMBARQUÉ

Couche Backend

- ▶ **Flask API** : Serveur REST Python
- ▶ **Logique** : Traitement des données
- ▶ **Sécurité** : Authentification API

 API RESTFUL

Couche Cloud

- ▶ **Firebase RTDB** : NoSQL temps réel
- ▶ **Synchronisation** : Flux instantanés
- ▶ **Scalabilité** : Multi-salles / Multi-sites

 INFRASTRUCTURE CLOUD

Couche Frontend

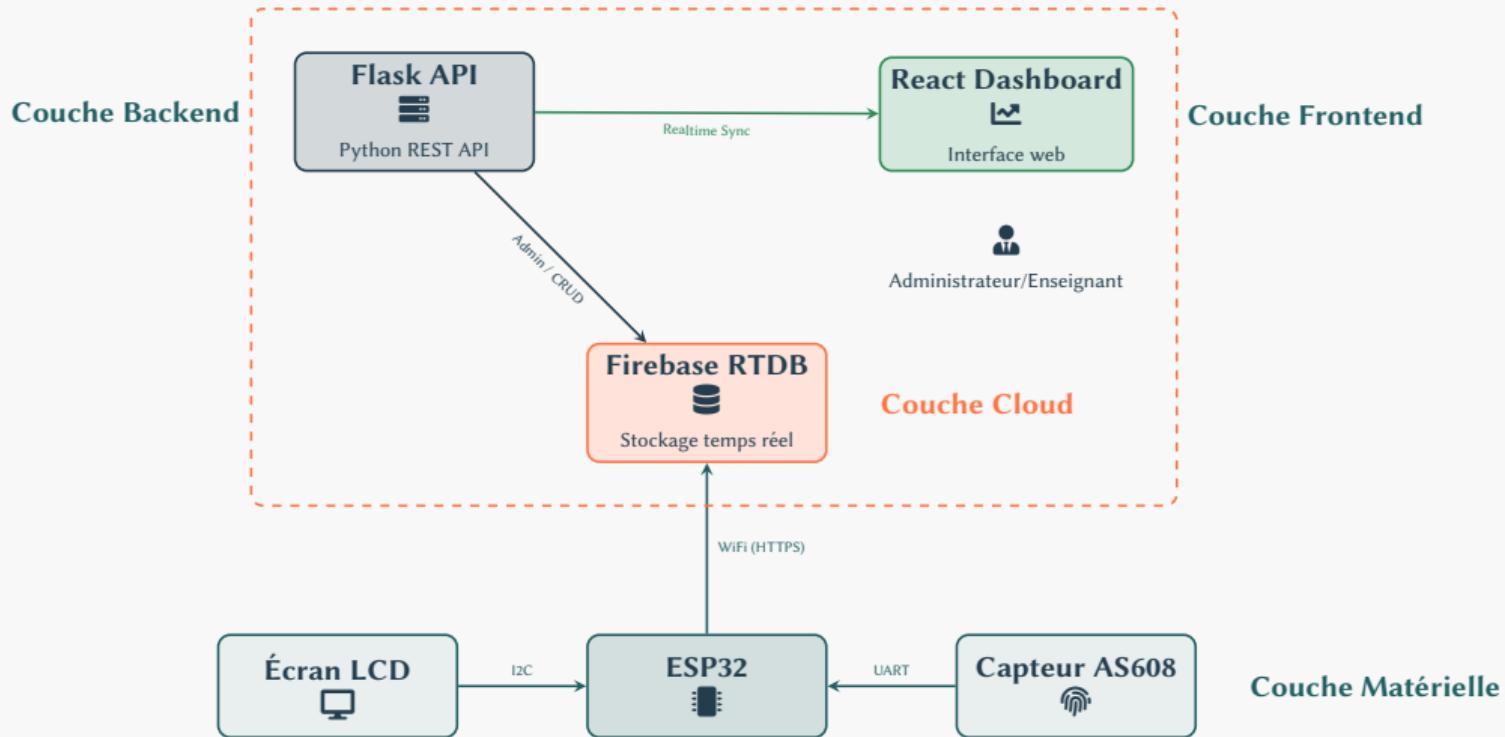
- ▶ **React.js** : Interface Single Page
- ▶ **Material-UI** : Design Responsive
- ▶ **Chart.js** : Visualisation stats

 DASHBOARD INTERACTIF

Schéma d'Architecture

Flux de données et interactions entre les composants

9





CHAPITRE 4

Architecture Technique

4



Système IoT de Présence → February 6, 2026

Couche IoT : Identification Biométrique

Unité de traitement embarquée et interface utilisateur

10

ESP32 : Le Cœur du Système

Performance Dual-Core (240 MHz)

Permet de gérer simultanément la lecture série du capteur et la pile TCP/IP pour le Cloud.

- ▶ WiFi Intégré : Protocole sécurisé pour Firebase.
- ▶ GPIOs : Gestion synchrone du LCD et du Buzzer.

Capteur AS608

Précision : 99.6%

- ▶ Extraction : Prétraitement d'image local.
- ▶ Vitesse : Match en < 1s.
- ▶ Stockage : 127 templates.

Feedback Utilisateur

- ▶ LCD 16x2 : Instructions en direct.
- ▶ Signaux Sonores : Validation acoustique.



UART (Serial)
↓

Communication ESP32

Note technique : L'utilisation de l'UART pour l'AS608 libère les ressources SPI pour d'autres extensions futures.

10/32

Couche Backend : Flask API

Le pivot central de la logique métier et du routage

11

Architecture de l'API

Python 3.8+ & Flask Micro-framework

- ▶ **Normalisation** : Prétraitement des données brutes IoT en objets JSON structurés.
- ▶ **Middleware** : Couche de validation vérifiant l'intégrité de chaque requête.
- ▶ **Scalabilité** : Architecture découpée facilitant l'ajout de nouvelles fonctionnalités.

Endpoints REST

POST	/api/attendance
GET	/api/students
GET	/api/stats
POST	/api/register



Sécurité & Intégrité

- ▶ **Filtrage** : Rejet automatique des trames malformées ou incomplètes.
- ▶ **Abstraction** : Le frontend ne communique jamais directement avec le hardware.

Évolution : Préparé pour l'implémentation de **JWT** (JSON Web Tokens).

Couche Frontend : Interface d'Administration

Dashboard réactif et pilotage des données sous React.js

12

Architecture logicielle

- ▶ **Composants Réutilisables :** Maintenance facilitée.
- ▶ **Material-UI (MUI) :** Rendu *Enterprise-ready*.
- ▶ **Single Page Application :** Navigation via React Router.

Écosystème de Gestion

- ➊ **Analytique** : Taux de présence.
- ➋ **Acteurs** : CRUD Étudiants.
- ➌ **Organisation** : Planning.
- ➍ **Live-Sync** : Pointage réel.

⚠ État du Prototype

Note : Focus sur les fonctionnalités métiers (Phase 1).
Auth prévue pour la Phase 2.



Performance : Zéro rechargement.



Dashboard



Étudiants



Salles



Présences



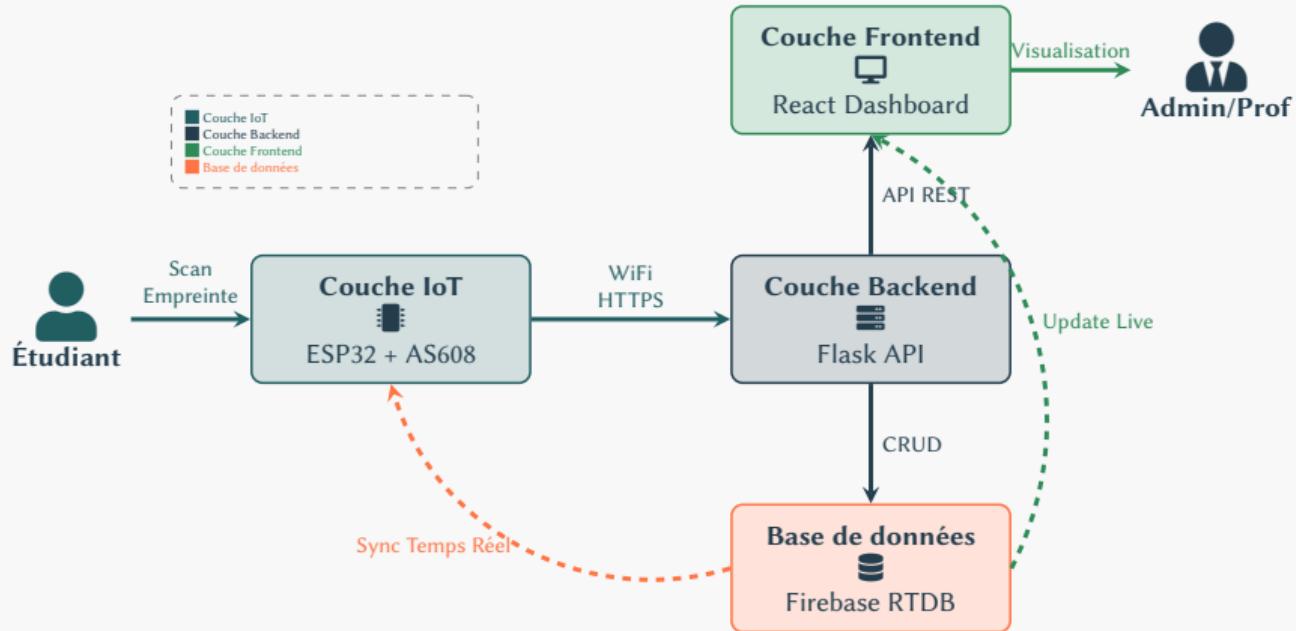
Planning

12/32

Synthèse Architecture

Interaction entre les différentes couches techniques

13



Conclusion architecture : Une structure modulaire permettant une **scalabilité** totale.



CHAPITRE 5

Flux de Fonctionnement

5



Système IoT de Présence → February 6, 2026

Workflow Complet du Système

De l'empreinte digitale à l'affichage en temps réel

14

1. Capture



Scan AS608
ID Biométrique
 1s



2. ESP32



Logique
Paquet JSON
 0.5s



3. Cloud



Firebase
Sync NoSQL
 0.7s



4. Backend



API Flask
Traitement
 0.3s



5. Frontend



React UI
Dashboard Live
 0.5s

14/32

Détail des Étapes Techniques

Analyse approfondie du cycle de vie d'une donnée

15

1. Scan Biométrique

- ▶ Scan optique haute précision (AS608)
- ▶ Conversion en modèle mathématique
- ▶ **Performance :** < 1s — Précision 99.6%

3. Transmission Cloud

- ▶ Connexion WiFi sécurisée (HTTPS)
- ▶ Envoi JSON structuré vers Firebase
- ▶ **Données :** ID, Horodatage, Salle

2. Traitement ESP32

- ▶ Identification par ID unique
- ▶ Vérification des doublons en local
- ▶ **Feedback :** Signal Buzzer + LCD

4. Traitement Backend

- ▶ API Flask : Validation & Cohérence
- ▶ Agrégation des statistiques live
- ▶ **Sortie :** Mise à jour du Dashboard React

 Chaque étape est optimisée pour garantir une latence totale inférieure à 3 secondes.

Étape 5 : Visualisation en Temps Réel

Interface d'administration et de suivi (Dashboard React)

16

► Interface React.js

- Synchronisation **Live** (Firebase)
- Filtres par groupe & salle
- Export PDF/Excel instantané

TAUX DE PRÉSENCE

94%

EFFECTIF RÉEL

45 / 50

Mise à jour : Temps réel





CHAPITRE 6

Choix Technologiques

6



Système IoT de Présence → February 6, 2026

Pourquoi ces technologies ?

Une sélection réfléchie pour chaque composant du système

17

Critères de Sélection

- ▶ **Performance** : Temps de traitement optimal
- ▶ **Fiabilité** : Stabilité et robustesse
- ▶ **Cout** : Budget maîtrisé pour le matériel
- ▶ **Communauté** : Support et documentation
- ▶ **Intégration** : Compatibilité entre composants
- ▶ **Scalabilité** : Adaptation à la croissance
- ▶ **Sécurité** : Protection des données sensibles

Philosophie

- ▶ **Simplicité** : Technologies éprouvées
- ▶ **Modernité** : Stack à jour et maintenue
- ▶ **Écosystème** : Bibliothèques et outils riches



ESP32



Flask



React



Firebase

17/32

Choix Technologique : ESP32

Le microcontrôleur idéal pour l'Internet des Objets (IoT)

18

Pourquoi l'ESP32 ?

- ▶ WiFi & Bluetooth intégrés
- ▶ Dual-Core 240 MHz (Rapidité)
- ▶ Économie (Mode Deep Sleep)
- ▶ Rapport Qualité/Prix imbattable
- ▶ Large support (Arduino/MicroPython)

vs Alternatives

(Arduino / STM32)

Critère	ESP32	Autre
WiFi/BT	✓	✗
Vitesse	240 MHz	16 MHz
RAM	520 KB	2 KB
Prix	5 - 8€	15€+

Conclusion

Solution robuste, compacte et parfaitement adaptée à une transmission cloud en temps réel.

Note : L'ESP32 permet de supprimer les modules WiFi externes (coût & place).

💡 L'ESP32 réduit la complexité matérielle tout en augmentant les performances de calcul.

Architecture Logicielle : Pourquoi Ce Stack ?

L'alliance de la performance Python et de la réactivité JavaScript

19

BACKEND : FLASK

- ▶ **Micro-framework** : Agilité maximale
- ▶ **API REST** : Communication JSON
- ▶ **Python** : Traitement efficace

⌚ Setup : Instantané

FRONTEND : REACT.JS

- ▶ **Virtual DOM** : Sync temps réel
- ▶ **Composants** : Code maintenable
- ▶ **State Hooks** : Gestion fluide

⚡ Rendu : 60 FPS

Critère de choix

Notre Stack

Verdict

Poids de l'architecture

Léger (Flask)

⬆ Optimisé

Expérience Utilisateur

Réactif (React)

⬆ Moderne

Temps de Développement

Réduit (Agile)

⬆ Rapide

Requête HTTP (GET/POST)



Infrastructure Cloud : Pourquoi Firebase ?

Le moteur de l'instantanéité et de la fiabilité IoT

20

Flux "Push" vs "Pull"



💡 La valeur ajoutée métier

Temps Réel : Contrairement au SQL (Polling), Firebase "pousse" la donnée. L'étudiant voit son nom s'afficher **immédiatement**.

⚠ Analyse Comparative

Caractéristique	Firebase	MySQL/SQL
Mise à jour	Push (Auto)	Pull (Requête)
Mode Hors-ligne	Natif (SDK)	À développer
Structure Data	JSON (Flexible)	Tables (Rigide)
Maintenance	Serverless	Serveur dédié

- ▶ **Résilience** : Le cache local gère les coupures Wi-Fi sans perte.
- ▶ **Scalabilité** : Architecture capable d'absorber des pics de pointage simultanés.

💡 Conclusion : Firebase a été choisi pour sa capacité à transformer un simple enregistrement en une **expérience interactive et fiable** pour l'utilisateur final.



CHAPITRE 7

Résultats & Apports

7



Système IoT de Présence → February 6, 2026

Résultats et Impact du Système

Optimisation radicale de la gestion des présences

21

⌚ Efficacité Temporelle

- ▶ Avant : 15-20 min / séance
- ▶ Après : ≈ 2 sec / étudiant
- ▶ Gain : 95% de temps récupéré

🛡 Sécurité & Fraude

- ▶ Fraude : Éliminée (Biométrie)
- ▶ Erreurs : 0% (Saisie auto)

↗️ Fiabilité des Données

- ▶ Exactitude : Authentification 99.6%
- ▶ Centralisation : Cloud Firebase

↔️ Scalabilité

- ▶ Capacité : Jusqu'à 1000 IDs
- ▶ Multi-sites : Déploiement facile



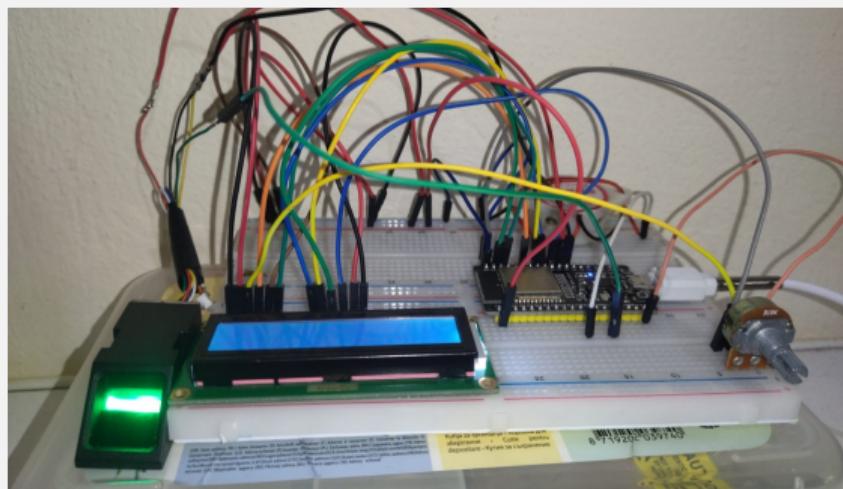
Bilan : Passage d'un processus manuel à une automatisation totale.

Réalisation Matérielle : Prototype IoT

Intégration des composants et tests de connectivité

22

Vue d'ensemble du prototype



Fiabilité : Les tests valident une reconnaissance précise (< 1s) même avec des doigts secs ou légèrement humides.

Architecture

- ▶ **Cœur :** ESP32 (WiFi + Dual Core)
- ▶ **ID :** Capteur AS608 (UART)
- ▶ **Feedback :** LCD I2C + Buzzer
- ▶ **Alim :** 5V/3.3V régulé

Détail Capteur AS608

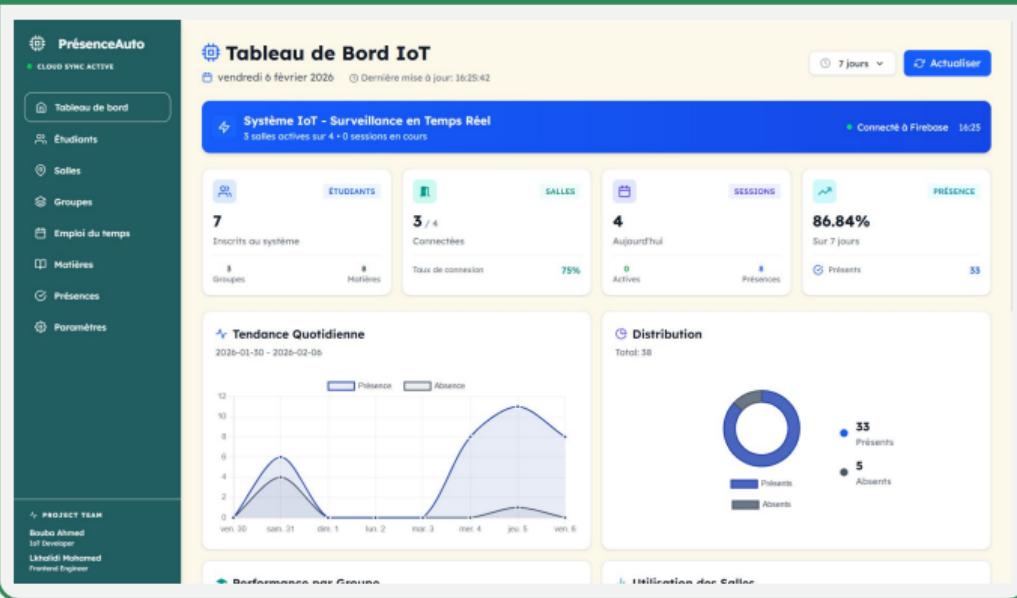


Interface Web : Dashboard de Supervision

Analyse des données et monitoring en temps réel

23

Tableau de bord analytique



Visualisation

- **Statistiques Live** : Taux de présence global.
- **Graphiques** : Suivi hebdomadaire (Chart.js).
- **Alertes** : Notifications en cas d'absence.

i Aperçu : Le système comporte d'autres modules non affichés ici.

23/32

Interface Web : Suivi des Présences

Registre dynamique et historique des pointages

24

Liste des présences et absences

Gestion des Présences

vendredi 6 février 2026 | Dernière mise à jour: 16:26:58

Auto-refresh

Présents	Absents	Sessions	Performance
8 Étudiants présents 8 Total	0 Étudiants absents 0% Taux d'absence	4 Sessions programmées 0 Actives	10:07 Heure moyenne d'arrivée 4 Terminées Données filtrées

Répartition Présence/Absence
Distribution globale des présences

Taux par Salle
Performance des salles de classe

Salle	Taux de présence (%)
roomB	100%
roomA	100%

Présences par Heure
Distribution horaire des arrivées

Heure	Nombre d'arrivées
08:00	4
10:00	2
14:00	2

Rechercher étudiant, salle ou groupe... 06/02/2026 Toutes les salles CSV

Registre des Présences 8 enregistrements

PROJECT TEAM

Bouba Ahmed
Software Developer
Lahdioui Mohamed
Frontend Engineer

Contrôle

- **Temps Réel** : Affichage instantané dès le scan du doigt.
- **Filtres** : Tri par date, groupe ou état (Présent/Absent).
- **Traçabilité** : Heure exacte de passage enregistrée.

Note : Le système gère également le CRUD étudiant, salles etc...

Administration du Système

Gestion des ressources et planification scolaire

25

1 Emplois du Temps

The screenshot shows the 'Gestion de l'emploi du temps' (Time Management) section. It includes a summary of scheduled programs (26), available rooms (3), active groups (3), and direct bookings (0). Below this is a navigation bar for days of the week, with 'Lun' (Monday) highlighted. A search bar for rooms and a button to add new ones are also present. At the bottom, a detailed view for 'Lundi - Emploi du temps' shows 3 rooms and 4 credit hours programmed between 08:00 and 16:00.

1 Gestion des Salles

The screenshot shows the 'Gestion des Salles' (Room Management) section. It displays room statistics: 8 configured rooms, 29 total credits, and 5 assigned teachers. A pie chart shows room distribution by level: Niveau 1 (blue), Niveau 2 (orange), and Niveau 3 (green). A bar chart shows the number of rooms per credit level (3, 4, and 5).

2 Planification des modules

Occupation des locaux

25/32



CHAPITRE 8

Limites & Améliorations

8



Système IoT de Présence → February 6, 2026

Analyse critique du système actuel

Conscience des limites pour une amélioration continue

26

Limites Identifiées

- ▶ **Prototype unique** : Interface uniquement admin
- ▶ **Aucune authentification** : Accès non sécurisé au dashboard
- ▶ **Dépendance Internet** : Nécessite connexion stable pour Firebase
- ▶ **Capacité limitée** : 127 empreintes maximum par capteur
- ▶ **Monofactoriel** : Juste l'empreinte digitale comme identification

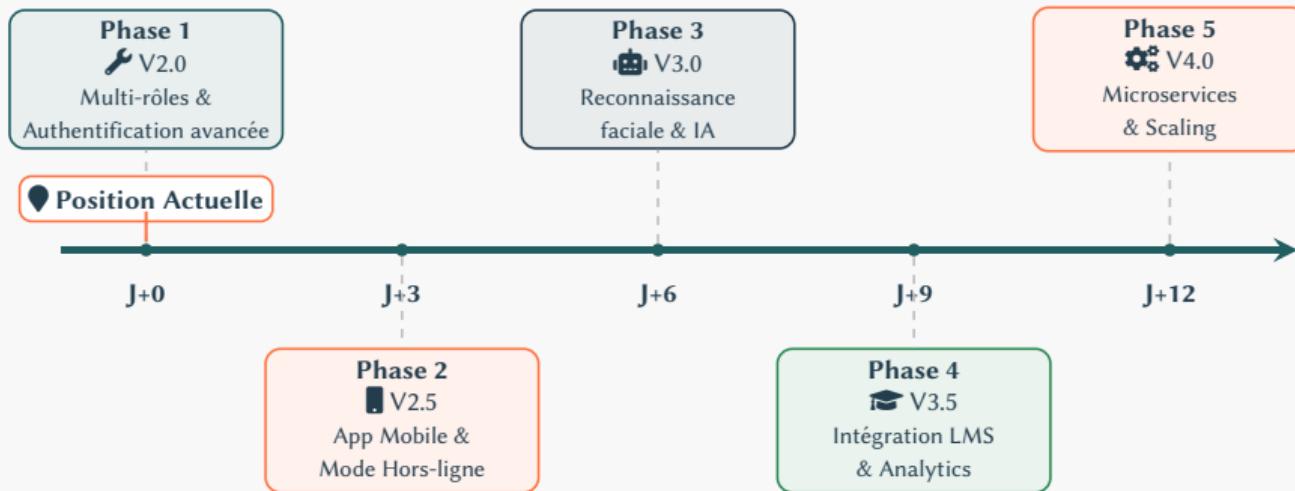
Philosophie d'amélioration

- ▶ **Approche itérative** : Premier prototype fonctionnel
- ▶ **Design évolutif** : Architecture modulaire
- ▶ **Feedback utilisateur** : Intégration des retours terrain
- ▶ **Veille technologique** : Suivi des nouvelles technologies
- ▶ **Open Source** : Contribution communautaire possible

Feuille de Route Détailée

Évolution prévue et perspectives d'avenir

27



- **Court terme :** Finalisation de l'écosystème mobile et amélioration de l'UX.
- **Long terme :** Transition vers une architecture distribuée pour supporter plusieurs établissements.

Objectif Final : Vers un Produit Industriel

De la preuve de concept au dispositif autonome

28

Optimisation Matérielle

- ▶ **Transition PCB** : Conception d'une carte mère dédiée pour éliminer les câblages manuels.
- ▶ **Autonomie** : Intégration d'un module de gestion d'énergie et batterie de secours.
- ▶ **Standardisation** : Utilisation de composants CMS pour une production en série.

Design & Ergonomie

- ▶ **Boîtier 3D** : Enveloppe sur mesure protégeant l'électronique contre la poussière et les chocs.
- ▶ **Interface** : Support mural incliné facilitant le scan biométrique.



Simulation de l'unité finale installée



Ambition : Transformer un prototype fonctionnel en une solution de gestion "Plug & Play".

9



CHAPITRE 9

Conclusion



Système IoT de Présence → February 6, 2026

Conclusion - Récapitulatif

De la problématique à la solution fonctionnelle

29

Objectifs Atteints

- ▶ **Automatisation complète :** Prise de présence biométrique
- ▶ **Stockage cloud :** Centralisation Firebase temps réel
- ▶ **Interface web :** Dashboard React fonctionnel
- ▶ **Scalabilité :** Architecture extensible multi-salles
- ▶ **Fiabilité :** 99.6% de reconnaissance d'empreintes

Résultats Concrets

- ▶ **Temps :** Réduction de 95% du temps de gestion
- ▶ **Fiabilité :** Élimination des fraudes et erreurs
- ▶ **Traçabilité :** Historique complet horodaté
- ▶ **Analyse :** Statistiques générées automatiquement
- ▶ **Scalabilité :** Architecture prête pour expansion

Bilan et Apports du Projet

Une solution globale bénéfique à tout l'écosystème

30



Enseignants

- ▶ Gain de temps pédagogique
- ▶ Zéro paperasse manuelle
- ▶ Rapports auto-générés



Administration

- ▶ Traçabilité & Sécurité
- ▶ Statistiques d'absentéisme
- ▶ Centralisation des données



Étudiants

- ▶ Équité (anti-fraude)
- ▶ Rapidité de passage
- ▶ Transparence du suivi



Apports pour l'Équipe de Développement

Ce projet nous a permis de maîtriser en binôme l'intégralité de la chaîne de valeur :

Conception IoT → Firmware ESP32 → Architecture Cloud → Fullstack Web



Expertise Technique

Gestion des protocoles (UART/I2C), APIs REST et synchronisation temps réel.



Gestion de Projet

Collaboration Agile, versioning Git et intégration matériel/logiciel.

Synthèse et Remerciements

Bilan du projet et clôture de la présentation

31

✓ Conclusion du Projet

Ce système IoT démontre comment la synergie entre l'**ESP32** et le **Cloud Firebase** permet de moderniser l'administration scolaire via une solution biométrique fiable, rapide et scalable.



Innovation



Technique



Pédagogie



Avenir

❤️ Remerciements

- ▶ **Pr. Omari Slimane**
Encadrant du projet
- ▶ **Pr. Abdelaoui Arbi Alaoui**
Coordinateur du Master SIE

🏛️ Institution

- ▶ **Équipe Pédagogique du Master**
- ▶ **ENS de Meknès**
- ▶ **À tous ceux qui ont contribué à la réussite de ce projet.**

Questions & Discussion

Clôture de la présentation — Système de gestion des présences

32



Merci de votre aimable attention

Nous sommes à votre disposition pour tout complément d'information.



Bouba Ahmed

✉ ah.bouba@edu.umi.ac.ma



Lkhalidi Mohamed

✉ m.lkhalidi@edu.umi.ac.ma