

École Marocaine des Sciences de l'Ingénieur

Rapport de Projet Attendify

Application Mobile de Gestion de Présence par QR Code

Réalisé par : Nacer Allah Majdoul et Anas Idrissi

Encadré par Pr. BE.ELBAGHAZAOUI :

Année Universitaire : 2025 – 2026

Table des matières

1	Introduction Générale	4
2	Présentation Générale de l'Application	5
3	Stack Technologique	6
3.1	Framework et Plateforme de Développement	6
3.2	Langage de Programmation	6
3.3	Backend et Gestion des Données	7
3.4	Navigation et Interface Utilisateur	7
3.5	Gestion des QR Codes	7
3.6	Récapitulatif de la Stack Technologique	7
4	Rôles et Utilisateurs	9
4.1	Administrateur	9
4.2	Professeur	9
4.3	Étudiant	10
5	Architecture du Projet	11
5.1	Structure des Dossiers	11
6	Modélisation des Données	13
6.1	Utilisateur (User)	13
6.2	Groupes, Cours et Sessions	13
6.3	Présence (Attendance)	14
7	Conception du Projet	15
7.1	Diagramme de Cas d'Utilisation	15
7.2	Diagramme de Classes	16
7.3	Diagramme de Séquence	17
8	Fonctionnalités par Rôle	18
8.1	Administrateur	18
8.2	Professeur	18

8.3 Étudiant	18
9 Système d'Authentification	20
10 Flux de Gestion de Présence	21
11 Conclusion Générale	22

Table des figures

7.1	Diagramme de cas d'utilisation	16
7.2	Diagramme de classes UML de l'application Attendify	17

Chapitre 1

Introduction Générale

La gestion de la présence des étudiants occupe une place importante dans les établissements d'enseignement, car elle permet d'assurer le suivi de l'assiduité et de mesurer l'implication des apprenants dans leur parcours académique. Elle constitue également un outil essentiel pour les enseignants et l'administration afin de garantir le bon déroulement des activités pédagogiques et le respect des règlements internes.

Traditionnellement, la présence est gérée à l'aide de méthodes manuelles telles que l'appel nominatif ou les feuilles d'émargement. Bien que simples à mettre en œuvre, ces méthodes présentent plusieurs inconvénients. Elles sont souvent chronophages, surtout dans les classes à effectif élevé, et peuvent interrompre le rythme des séances. De plus, elles sont sujettes à des erreurs humaines, à des oublis et parfois à des pratiques frauduleuses, ce qui affecte la fiabilité des données collectées. La conservation et l'exploitation de ces informations deviennent également complexes lorsqu'elles sont gérées de manière non informatisée.

Avec l'évolution des technologies numériques, en particulier des applications mobiles et des solutions cloud, de nouvelles approches ont vu le jour pour moderniser les processus de gestion académique. L'utilisation généralisée des smartphones par les étudiants et les enseignants offre une opportunité pertinente pour automatiser certaines tâches administratives. Par ailleurs, les bases de données en temps réel permettent un accès rapide et sécurisé aux informations, tout en facilitant leur mise à jour et leur analyse.

Dans ce contexte, le projet Attendify propose une solution mobile innovante pour la gestion de la présence des étudiants. L'application repose sur l'utilisation de QR codes dynamiques, générés par les professeurs pour chaque séance, et scannés par les étudiants afin d'enregistrer automatiquement leur présence. Les données sont stockées et synchronisées en temps réel grâce à un backend cloud basé sur Firebase, garantissant ainsi la fiabilité, la sécurité et la rapidité du système.

Ainsi, Attendify vise à simplifier et optimiser la gestion de la présence académique en offrant une solution moderne, efficace et adaptée aux besoins actuels des établissements d'enseignement, tout en améliorant l'expérience des enseignants et des étudiants.

Chapitre 2

Présentation Générale de l'Application

Attendify est une application mobile conçue pour moderniser et automatiser la gestion de la présence des étudiants au sein des établissements d'enseignement. Elle a été développée à l'aide du framework React Native, en s'appuyant sur la plateforme Expo, ce qui permet d'assurer une compatibilité multiplateforme, notamment sur les systèmes Android et iOS, tout en garantissant de bonnes performances et une expérience utilisateur fluide.

Le principe de fonctionnement de l'application repose sur l'utilisation de QR codes dynamiques générés par les professeurs pour chaque séance de cours. Lorsqu'une session de présence est créée, un QR code unique, limité dans le temps, est affiché par le professeur. Les étudiants présents scannent ce QR code à l'aide de l'application mobile, ce qui permet d'enregistrer automatiquement leur présence dans le système. Cette approche réduit considérablement le temps consacré à l'appel manuel et limite les risques d'erreurs ou de fraudes.

Afin d'assurer une organisation claire et sécurisée des fonctionnalités, Attendify intègre un système de gestion des rôles. Trois profils d'utilisateurs sont définis : l'administrateur, le professeur et l'étudiant. Chaque rôle dispose d'autorisations spécifiques adaptées à ses responsabilités. L'administrateur assure la gestion globale du système, notamment la création des utilisateurs, des groupes et des cours. Le professeur est chargé de la gestion des séances, de la génération des QR codes et du suivi des présences. Quant à l'étudiant, il peut scanner les QR codes et consulter son historique de présence.

Grâce à cette architecture basée sur les rôles, l'application garantit une séparation claire des responsabilités, une meilleure sécurité des données et une expérience utilisateur adaptée à chaque type d'utilisateur. Attendify se positionne ainsi comme une solution mobile moderne, fiable et intuitive, répondant aux besoins actuels des établissements d'enseignement en matière de gestion de la présence.

Chapitre 3

Stack Technologique

Le choix des technologies constitue une étape essentielle dans la réalisation d'une application mobile performante, évolutive et adaptée aux besoins des utilisateurs. Pour le projet **Attendify**, une stack technologique moderne a été adoptée afin de garantir une bonne expérience utilisateur, une maintenance facilitée et une synchronisation des données en temps réel.

Les technologies sélectionnées permettent le développement d'une application multi-plateforme, accessible aussi bien sur les appareils Android que iOS, tout en s'appuyant sur un backend cloud sécurisé. Cette combinaison assure une séparation claire entre la logique métier, l'interface utilisateur et la gestion des données.

3.1 Framework et Plateforme de Développement

React Native a été choisi comme framework principal pour le développement de l'application mobile. Il permet de développer des applications multiplateformes à partir d'une base de code unique, tout en offrant des performances proches des applications natives. Ce choix permet de réduire le temps de développement et de faciliter la maintenance du projet.

La plateforme **Expo** vient compléter React Native en simplifiant la configuration du projet et l'accès aux fonctionnalités natives du smartphone, telles que la caméra, indispensable pour le scan des QR codes. Expo facilite également le déploiement et les tests de l'application sur différents appareils.

3.2 Langage de Programmation

Le projet est développé en **TypeScript**, un sur-ensemble de JavaScript intégrant le typage statique. L'utilisation de TypeScript permet d'améliorer la qualité du code, de réduire les erreurs à l'exécution et de faciliter la collaboration ainsi que la maintenance

du projet à long terme.

3.3 Backend et Gestion des Données

Pour la gestion du backend et des données, **Firebase** a été retenu. Firebase Authentication est utilisé pour la gestion sécurisée des connexions et des rôles utilisateurs, tandis que **Cloud Firestore** assure le stockage et la synchronisation en temps réel des données. Cette solution cloud permet une mise à jour instantanée des informations de présence et garantit une bonne scalabilité du système.

3.4 Navigation et Interface Utilisateur

La navigation entre les différentes interfaces de l'application est gérée à l'aide de **React Navigation**. Cette bibliothèque permet une gestion fluide des écrans et une redirection automatique selon le rôle de l'utilisateur après authentification.

Pour l'aspect visuel, la bibliothèque **expo-linear-gradient** est utilisée afin d'améliorer l'expérience utilisateur grâce à des interfaces modernes et harmonieuses.

3.5 Gestion des QR Codes

La génération et le scan des QR codes reposent sur les bibliothèques **react-native-qrcode-svg** pour la création des QR codes dynamiques et **expo-camera** pour l'accès à la caméra du smartphone. Cette combinaison permet une gestion rapide, fiable et sécurisée du processus de présence.

3.6 Récapitulatif de la Stack Technologique

Catégorie	Technologie	Version
Framework Mobile	React Native	0.81.5
Plateforme	Expo	54.0.30
Langage	TypeScript	5.9.3
Backend / Base de données	Firebase (Firestore + Auth)	12.7.0
Navigation	React Navigation	6.x
Interface Utilisateur	expo-linear-gradient	15.0.8
QR Code	react-native-qrcode-svg, expo-camera	—

Ce choix technologique permet à Attendify de bénéficier d'une architecture moderne, robuste et évolutive, tout en assurant une expérience utilisateur fluide et une gestion efficace des données en temps réel.

Chapitre 4

Rôles et Utilisateurs

L'application Attendify repose sur un système de gestion des utilisateurs basé sur des rôles distincts. Cette approche permet de définir clairement les responsabilités, les droits d'accès et les fonctionnalités disponibles pour chaque catégorie d'utilisateur. La séparation des rôles contribue à renforcer la sécurité du système, à améliorer l'organisation globale de l'application et à offrir une expérience utilisateur adaptée à chaque profil.

Trois types d'utilisateurs sont pris en charge par l'application : l'administrateur, le professeur et l'étudiant. Chaque rôle dispose de permissions spécifiques, déterminées en fonction de ses missions au sein du système.

4.1 Administrateur

L'administrateur représente le rôle ayant le plus haut niveau d'autorisation dans l'application. Il est responsable de la gestion globale du système et de la configuration initiale de la plateforme. Ses principales missions consistent à créer, modifier et supprimer les comptes utilisateurs, qu'il s'agisse des professeurs ou des étudiants.

L'administrateur assure également la gestion des groupes pédagogiques en assignant les étudiants aux groupes correspondants, ainsi que la création des cours et leur association aux professeurs et aux groupes concernés. En outre, il dispose de fonctionnalités permettant la réinitialisation des mots de passe et la supervision générale de l'application. Ce rôle garantit ainsi le bon fonctionnement, la cohérence et la sécurité du système.

4.2 Professeur

Le professeur est un utilisateur central dans le processus de gestion de la présence. Il a accès aux cours qui lui sont assignés par l'administrateur et peut gérer les séances de présence associées à ces cours. Pour chaque séance, le professeur est en mesure de générer un QR code dynamique, valide pour une durée limitée.

Grâce à cette fonctionnalité, le professeur peut suivre la présence des étudiants en temps réel et consulter les listes des étudiants présents, absents ou en retard. Il a également la possibilité de clôturer une session de présence, ce qui permet de marquer automatiquement les étudiants n'ayant pas scanné le QR code comme absents. Ce rôle vise à simplifier le travail de l'enseignant tout en garantissant la fiabilité des données de présence.

4.3 Étudiant

L'étudiant est l'utilisateur final du système. Son rôle principal consiste à enregistrer sa présence en scannant le QR code généré par le professeur à l'aide de l'application mobile. Cette action permet de marquer automatiquement sa présence pour la séance concernée.

En plus de cette fonctionnalité, l'étudiant peut consulter son historique de présence, lui permettant de suivre son assiduité au fil du temps. Le système peut également indiquer le statut de chaque séance, tel que présent, en retard ou absent. Ce rôle offre ainsi à l'étudiant une meilleure visibilité sur son parcours académique et favorise une plus grande responsabilisation.

Chapitre 5

Architecture du Projet

L'architecture du projet Attendify a été conçue selon une approche modulaire afin de garantir la maintenabilité, la lisibilité et l'évolutivité de l'application. Cette organisation permet de séparer les différentes responsabilités du système, facilitant ainsi le développement, les tests et les futures améliorations.

La structure du projet est basée sur une répartition claire entre les composants d'interface, les écrans, la logique métier, la gestion des données et les outils utilitaires. Cette organisation favorise une meilleure collaboration entre les développeurs et réduit les risques d'erreurs liées à un code monolithique.

5.1 Structure des Dossiers

Le projet est organisé selon l'arborescence suivante :

```
attendify-mobile/  
  src/  
    components/      % Composants réutilisables de l'interface  
    screens/         % Écrans principaux de l'application  
    services/        % Services de communication avec Firebase  
    contexts/        % Contextes React pour la gestion de l'état global  
    navigation/      % Configuration de la navigation  
    types/           % Modèles et types TypeScript  
    utils/           % Fonctions utilitaires  
  assets/            % Ressources graphiques  
  scripts/           % Scripts utilitaires  
  package.json       % Dépendances et configuration du projet
```

Le dossier `components` contient les composants réutilisables de l'interface utilisateur, ce qui permet d'assurer une cohérence visuelle et de réduire la duplication du code. Le

dossier **screens** regroupe les différentes interfaces correspondant aux écrans accessibles par les utilisateurs selon leur rôle.

Les services Firebase sont centralisés dans le dossier **services**, facilitant ainsi la gestion des appels à la base de données et à l'authentification. Les contextes React, placés dans le dossier **contexts**, permettent de gérer l'état global de l'application, notamment l'authentification et les informations utilisateur.

Cette architecture modulaire permet à Attendify d'être facilement extensible et adaptable à de nouvelles fonctionnalités, tout en maintenant une structure claire et bien organisée.

Chapitre 6

Modélisation des Données

La modélisation des données constitue une étape fondamentale dans la conception de l'application Attendify. Elle permet de définir la structure des informations à stocker, les relations entre les différentes entités et les règles de gestion associées. Une modélisation claire et cohérente garantit la fiabilité des données, facilite leur exploitation et assure une bonne évolutivité du système.

Dans Attendify, les données sont stockées dans une base de données NoSQL, à savoir **Cloud Firestore**, qui offre une synchronisation en temps réel et une grande flexibilité dans la gestion des collections et des documents.

6.1 Utilisateur (User)

L'entité **Utilisateur** représente l'ensemble des personnes ayant accès à l'application. Chaque utilisateur est enregistré dans la collection **users** de Firestore et est identifié de manière unique par un identifiant (**uid**) fourni par Firebase Authentication.

Les informations stockées pour chaque utilisateur incluent les données personnelles essentielles ainsi que le rôle attribué (administrateur, professeur ou étudiant). Ce rôle détermine les fonctionnalités accessibles après authentification. Pour les étudiants, une association à un groupe pédagogique peut être définie afin de faciliter la gestion des cours et des présences.

6.2 Groupes, Cours et Sessions

Les **groupes** permettent d'organiser les étudiants selon des critères pédagogiques définis par l'établissement. Chaque groupe regroupe un ensemble d'étudiants et peut être associé à un ou plusieurs cours.

Les **cours** représentent les unités d'enseignement dispensées par les professeurs. Chaque cours est associé à un professeur responsable ainsi qu'aux groupes concernés. Cette asso-

ciation permet de déterminer quels étudiants sont autorisés à participer aux séances de présence.

Les **sessions** correspondent aux séances de cours pour lesquelles la présence doit être enregistrée. Une session est liée à un cours, à un professeur et à un groupe spécifique. Elle est caractérisée par une durée limitée et par la génération d'un QR code dynamique utilisé pour l'enregistrement de la présence.

6.3 Présence (Attendance)

L'entité **Présence** permet de tracer la participation des étudiants aux différentes sessions. Chaque scan de QR code par un étudiant génère un enregistrement de présence stocké dans la collection **attendances**. Cet enregistrement est lié à une session et à un étudiant spécifiques.

Les informations enregistrées permettent d'indiquer le statut de l'étudiant (présent, en retard ou absent) ainsi que l'horodatage du scan. Cette modélisation permet un suivi précis de l'assiduité et facilite la génération de statistiques et de rapports de présence.

Chapitre 7

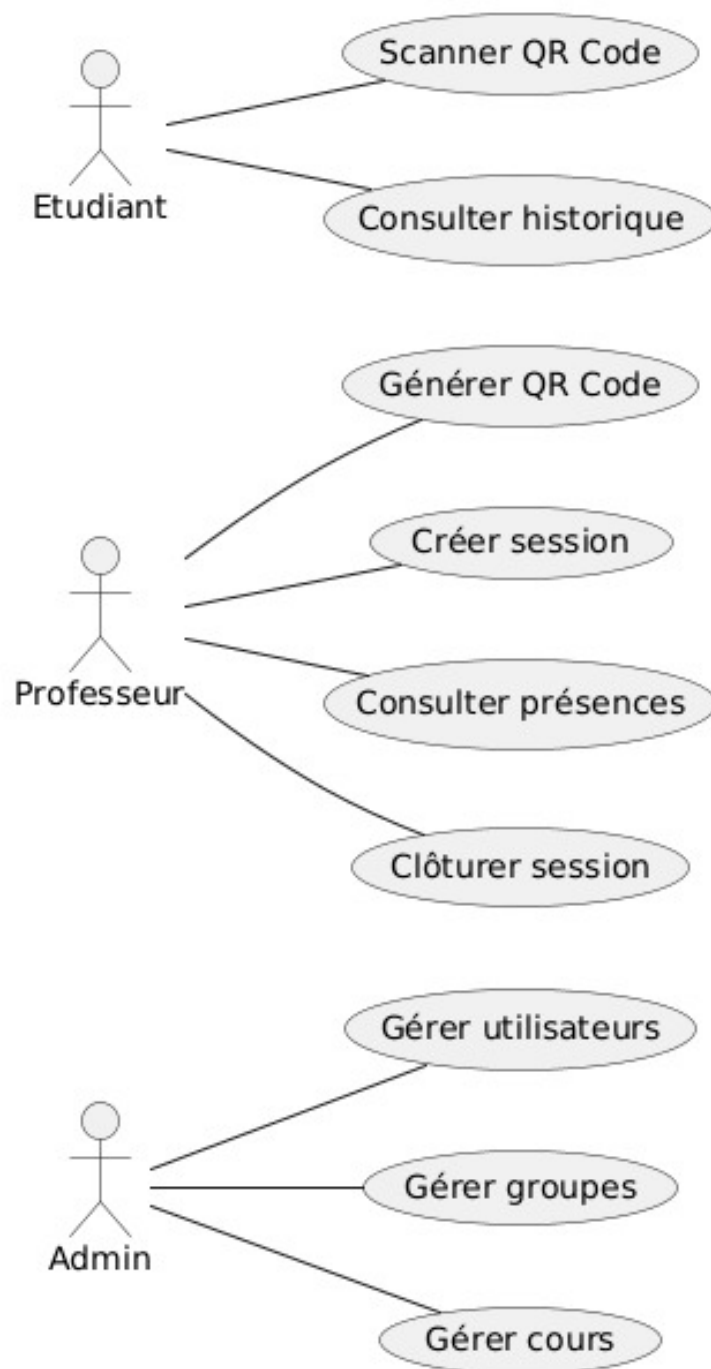
Conception du Projet

La phase de conception vise à traduire les besoins fonctionnels du système en modèles clairs et structurés. Elle permet de visualiser le fonctionnement global de l'application, d'identifier les acteurs impliqués et de définir les interactions entre les différentes composantes du système.

Dans le cadre du projet Attendify, la conception repose sur l'utilisation de diagrammes UML, qui offrent une représentation graphique standardisée et compréhensible de l'architecture et du comportement de l'application.

7.1 Diagramme de Cas d'Utilisation

Le diagramme de cas d'utilisation permet d'illustrer les différentes fonctionnalités offertes par le système ainsi que les interactions entre les utilisateurs et l'application. Il met en évidence les rôles principaux (administrateur, professeur et étudiant) et les actions qu'ils peuvent effectuer.



beginfigure[H]

FIGURE 7.1 – Diagramme de cas d'utilisation

Diagramme de cas d'utilisation de l'application Attendify

7.2 Diagramme de Classes

Le diagramme de classes représente la structure statique du système. Il décrit les différentes classes du domaine, leurs attributs, leurs méthodes ainsi que les relations existantes entre elles. Ce diagramme permet de comprendre l'organisation des données et les liens

entre les entités telles que les utilisateurs, les cours, les groupes et les sessions.

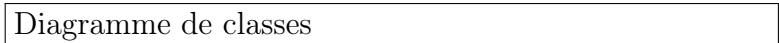
A rectangular box containing the text "Diagramme de classes".

Diagramme de classes

FIGURE 7.2 – Diagramme de classes UML de l'application Attendify

7.3 Diagramme de Séquence

Le diagramme de séquence illustre le déroulement temporel des interactions entre les acteurs et le système pour un scénario donné. Dans le cadre de ce projet, il permet notamment de représenter le flux de gestion de la présence, depuis la génération du QR code par le professeur jusqu'à l'enregistrement de la présence de l'étudiant.

Chapitre 8

Fonctionnalités par Rôle

Ce chapitre présente de manière détaillée les fonctionnalités offertes par l'application Attendify en fonction du rôle de l'utilisateur. Cette approche permet de mieux comprendre les responsabilités de chaque acteur et les services mis à sa disposition.

8.1 Administrateur

L'administrateur dispose d'un accès complet à l'ensemble des fonctionnalités de l'application. Il est chargé de la gestion des utilisateurs, incluant la création, la modification et la suppression des comptes des professeurs et des étudiants.

Il assure également la gestion des groupes pédagogiques et des cours, en attribuant les professeurs aux cours et en associant les groupes correspondants. En outre, l'administrateur peut procéder à la réinitialisation des mots de passe et superviser le bon fonctionnement global du système.

8.2 Professeur

Le professeur est responsable de la gestion des séances de présence liées à ses cours. Il peut consulter la liste des cours qui lui sont assignés et créer des sessions de présence pour chaque séance.

Pour chaque session, le professeur génère un QR code dynamique, valide pour une durée limitée. Il peut suivre en temps réel la présence des étudiants, consulter les listes des présents et des absents, et clôturer la session à la fin de la séance.

8.3 Étudiant

L'étudiant utilise l'application principalement pour enregistrer sa présence. Il scanne le QR code généré par le professeur à l'aide de son smartphone afin de valider sa participation

à la séance.

L'application permet également à l'étudiant de consulter son historique de présence, lui offrant une visibilité sur son assiduité et son statut pour chaque session. Cette fonctionnalité favorise une meilleure responsabilisation de l'étudiant vis-à-vis de son parcours académique.

Chapitre 9

Système d'Authentification

Le système d'authentification constitue un élément central de l'application Attendify, car il garantit la sécurité des accès et la protection des données utilisateurs. Pour répondre à ces exigences, l'application s'appuie sur le service **Firebase Authentication**, qui offre une solution fiable et sécurisée pour la gestion des connexions.

Chaque utilisateur dispose d'un compte unique identifié par une adresse électronique et un mot de passe. Lors de la connexion, Firebase Authentication vérifie les informations fournies et établit une session sécurisée. Une fois l'utilisateur authentifié, les informations complémentaires, notamment le rôle associé (administrateur, professeur ou étudiant), sont récupérées depuis la base de données **Cloud Firestore**.

En fonction du rôle identifié, une redirection automatique est effectuée vers l'interface correspondante. Cette approche permet de garantir que chaque utilisateur accède uniquement aux fonctionnalités qui lui sont autorisées. Elle renforce également la sécurité du système en empêchant l'accès non autorisé à certaines parties de l'application.

Ainsi, l'intégration de Firebase Authentication permet d'assurer une gestion centralisée, sécurisée et évolutive des utilisateurs, tout en simplifiant l'implémentation du contrôle d'accès basé sur les rôles.

Chapitre 10

Flux de Gestion de Présence

Le flux de gestion de la présence dans l'application Attendify repose sur un processus simple, rapide et sécurisé, basé sur l'utilisation de QR codes dynamiques. Ce mécanisme vise à automatiser l'enregistrement de la présence tout en limitant les erreurs et les tentatives de fraude.

Le processus débute lorsque le professeur sélectionne un cours et un groupe, puis crée une session de présence pour une durée déterminée. À partir de cette session, l'application génère un QR code unique et temporaire. Ce QR code est affiché à l'écran du professeur et reste valide uniquement pendant la durée de la session.

Les étudiants présents scannent ce QR code à l'aide de l'application mobile. Lors du scan, le système vérifie la validité du QR code, l'identité de l'étudiant ainsi que l'appartenance de celui-ci au groupe concerné. Une fois ces vérifications effectuées, la présence est enregistrée en temps réel dans la base de données.

À la fin de la séance, le professeur peut clôturer la session. Les étudiants n'ayant pas scanné le QR code sont alors automatiquement marqués comme absents. Ce flux permet d'assurer une gestion fiable, rapide et transparente de la présence académique.

Chapitre 11

Conclusion Générale

Le projet Attendify a permis de concevoir et de développer une application mobile moderne dédiée à la gestion de la présence académique. En s'appuyant sur des technologies récentes telles que React Native, Expo et Firebase, l'application propose une solution efficace répondant aux besoins actuels des établissements d'enseignement.

Grâce à son architecture cloud, son système de gestion des rôles et l'utilisation de QR codes dynamiques, Attendify offre une expérience utilisateur fluide, sécurisée et fiable pour l'ensemble des acteurs du système. L'automatisation du processus de présence permet de réduire le temps consacré aux tâches administratives tout en améliorant la précision des données collectées.

Ce projet constitue une base solide pouvant être enrichie par de futures améliorations, telles que l'intégration de statistiques avancées, de notifications en temps réel ou encore l'exportation des rapports de présence. Attendify représente ainsi une solution évolutive et pertinente pour la digitalisation des processus académiques.