

**Faculté des Sciences Exactes et d'Informatique
Département de Mathématiques et informatique**

- **Filière : Informatique**
- **Spécialité : intelligence artificielle 4IOT**

- **Option: Master 2 IA**

Rapport Sur :

**Système de Gestion de Présence par Empreinte
Digitale**

✓ Etudiants: **BOUCHENA HAMZA**

BENSELAMA MOHAMMED SAID

✓ Encadrant: **BENZOHRA ABDELKADER**

Année Universitaire 2024-2025

REMERCIEMENTS

On dit souvent que le trajet est aussi important que la destination. Les années de master nous ont permis de bien comprendre la signification de cette phrase toute simple. Ce parcours, en effet, ne s'est pas réalisé sans défis et sans soulever de nombreuses questions pour lesquelles les réponses nécessitent de longues heures de travail.

Avant tout nous remercions ALLAH tout puissant pour la volonté, et la puissance qu'il nous a accordées durant toutes ces années d'études.

Nous remercions nos parents qui nous ont guidés pendant la vie, ils nous ont encouragés, ils nous ont soutenus afin d'être ici, à la veille de la fin de nos études universitaires.

*Au terme de ce projet de fin d'étude nous tenons à remercier vivement notre encadreur Mr **BENZOHRA ABDELKADER** pour tout le temps qu'il nous a consacré, sa générosité et de la qualité de son suivi, ainsi que pour tous ce qu'il a mis à nos dispositions, qui a été très utile durant nos recherches.*

De même nous remercions chaleureusement les membres du jury pour leurs remarques et la complémentarité de leurs jugements, nous donnent encore confiance et intérêt pour apprendre toujours et l'honneur qu'ils font pour juger et apprécier notre travail.

Nous remercions, toute personne qui a contribué de près ou de loin à la réalisation de notre projet de fin d'études.

Liste des figures

Figure 1 :	Architecture du système	14
Figure 2 :	Diagramme de contexte statique	16
Figure 3 :	Diagramme de cas d'utilisation « Administrateur »	17
Figure 4 :	Diagramme de cas d'utilisation « Enseignant »	18
Figure 5 :	Diagramme de cas d'utilisation « Etudiant »	19
Figure 6 :	Diagramme de séquence « Authentification »	20
Figure 7 :	Diagramme de séquence « Enregistrement d'un étudiant »	21
Figure 8 :	Diagramme de séquence « Enregistrement de la présence d'étudiant »	22
Figure 9 :	Diagramme de séquence « Consultation des présences par l'enseignant »	23
Figure 10 :	Diagramme de Classe	24
Figure 11 :	Diagramme de déploiement	25
Figure 12 :	Logo Springboot	28
Figure 13 :	Logo MongoDB	28
Figure 14 :	Logo React	29
Figure 15 :	Logo VS code	30
Figure 16 :	Logo IntelliJ IDEA	30
Figure 17 :	Logo Postman	31
Figure 18 :	Logo Tailwind CSS	32
Figure 19 :	Logo Node.js	32
Figure 20 :	ZK9500 (ZKTeco)	33
Figure 21 :	Document MongoDB Admin	34
Figure 22 :	Document MongoDB Enseignant	35
Figure 23 :	Document MongoDB Emploi du temps	35
Figure 24 :	Document MongoDB Etudiant	36
Figure 25 :	Document MongoDB Marque de présence	36
Figure 26 :	Smart Classroom d'accueil	37
Figure 27 :	Smart Classroom Se connecter	38
Figure 28 :	Tableau de Bord	38
Figure 29 :	Formulaire d'ajouter étudiant	39
Figure 30 :	Gestion d'étudiant	40
Figure 31 :	Formulaire d'ajouter enseignant	40
Figure 32 :	La liste des enseignants	41
Figure 33 :	Formulaire d'ajouter emploi du temps	41
Figure 34 :	Emploi du temps	42
Figure 35 :	Liste de présence	42
Figure 36 :	Page de présence par semestre	43
Figure 37 :	Rapport en PDF	43

Table des matières

Introduction Générale	5
Chapitre 1.....	7
Présentation générale et analyse des besoins	7
1.1 Introduction.....	7
1.2 Système de pointage actuel	7
1.3 Problématique	9
1.4 Solution.....	10
1.5 Spécification des besoins	11
1.5.1 Les besoins fonctionnels	12
1.5.2 Les besoins non fonctionnels	13
1.6 Architecture de notre système	14
1.7 Conclusion	14
Chapitre 02	15
Modélisation du système.....	15
2.1 Introduction.....	15
2.2 Choix de la méthode de conception	15
2.3 Conception détaillée.....	16
2.3.1 Diagramme de contexte statique	16
2.3.2 Diagramme de cas d'utilisation	17
2.3.3 Diagramme de séquence.....	19
2.3.4 Diagramme de classe	23
2.3.5 Diagramme de déploiement.....	24
2.4 Capteur ZK9500 (ZKTeco)[5]	25
2.5 L'algorithme utilisé par le capteur ZK9500	26
2.6 Conclusion	26
Chapitre 03	27
Réalisation & Implémentation.....	27
3.1 Introduction.....	27
3.2 Environnement logiciel	27
3.2.1 Java Spring Boot.....	27
3.2.2 MongoDB	28
3.2.3 React.....	29
3.2.4 VS code.....	29
3.2.5 IntelliJ IDEA Community Edition.....	30
3.2.5 Postman.....	31

3.2.7 Tailwind CSS	31
3.2.8 Node.js	32
3.3 Capteur utilisé	33
3.4 Structure de la base de données.....	33
3.5 Interfaces de l'application.....	37
3.5.1 Page d'accueil du site	37
3.5.2 Page de connexion	37
3.5.3 Page de Tableau de Bord.....	38
3.5.4 Page d'ajouter étudiant	39
3.5.5 Page de gérer l'étudiant.....	39
3.5.6 Page d'ajout d'un enseignant	40
3.5.7 Page d'ajouter l'emploi du temps	41
3.5.8 Page de présence.....	42
3.5.9 Page de Rapport	43
3.6 Conclusion.....	44
Conclusion générale et perspectives	45
Bibliographie.....	46

Introduction Générale

Dans un monde de plus en plus connecté et digitalisé, les établissements éducatifs sont confrontés à des défis majeurs pour moderniser leurs infrastructures et améliorer leur gestion quotidienne. Parmi ces défis, la gestion des salles de classe et le suivi des présences des étudiants occupent une place centrale. Traditionnellement, ces tâches reposent sur des méthodes manuelles, telles que les registres de présence papier ou les listes d'appel, qui s'avèrent chronophages, peu fiables et difficiles à exploiter.

Aujourd'hui, grâce aux avancées technologiques, il est possible de repenser entièrement la manière dont les salles de classe sont gérées. L'émergence de solutions intelligentes, basées sur des technologies comme l'authentification biométrique, ouvre la voie à une gestion plus efficace, plus sécurisée et plus adaptée aux besoins des établissements modernes. Parmi ces technologies, l'authentification par empreinte digitale se distingue par sa simplicité, sa fiabilité et son caractère unique, offrant une alternative robuste aux méthodes traditionnelles.

Le présent projet a pour objectif de présenter la conception et la réalisation d'un système intelligent de gestion de classe, intégrant une authentification biométrique par empreinte digitale. Ce système vise à moderniser et automatiser plusieurs aspects de la gestion quotidienne dans les établissements éducatifs, en mettant l'accent sur la sécurité, la fiabilité et la simplicité d'usage.

Grâce à l'authentification par empreinte digitale, le système permettra notamment de contrôler l'accès à la salle de classe, d'enregistrer automatiquement la présence des étudiants, et de générer des rapports détaillés sur les présences. Ce processus remplacera les méthodes traditionnelles, souvent manuelles, sujettes à des erreurs et des fraudes. De plus, la plateforme offrira une interface web conviviale qui permettra à l'administration de consulter et de gérer les données de manière centralisée.

Dans ce mémoire, nous décrirons les différentes étapes de la conception et du développement de ce système intelligent, incluant le choix du capteur biométrique (le ZK9500), la mise en place de la base de données, la création de l'interface utilisateur et l'intégration du module d'authentification. Nous aborderons également les défis rencontrés lors du développement et les solutions apportées pour garantir la fiabilité et la sécurité du système.

Enfin, nous évaluerons les performances du système en termes de précision de reconnaissance, de rapidité de traitement et de facilité d'utilisation. Nous concluons en soulignant l'impact de

ce projet sur l'amélioration de la gestion scolaire et en proposant des perspectives d'évolution pour enrichir le système avec d'autres fonctionnalités intelligentes.

Ce mémoire est structuré en trois chapitres :

- Le premier chapitre présente la généralité & spécification des besoins, avec un aperçu de notre solution.
- Le deuxième chapitre détaille la modélisation du système proposé.
- Enfin, le troisième chapitre est consacré à l'implémentation de la solution et à la présentation de l'application développée
- Une conclusion générale avec les perspectives futures de ce travail.

Chapitre 1

Présentation générale et analyse des besoins

1.1 Introduction

Dans ce chapitre, nous présentons le contexte général de notre projet, ainsi qu'une première analyse des besoins fonctionnels et techniques. Le projet s'inscrit dans une démarche d'innovation dans le domaine éducatif, visant à optimiser la gestion des classes à travers l'intégration de technologies intelligentes, notamment la biométrie.

Plus précisément, nous intéressons à l'authentification par empreinte digitale comme solution fiable pour automatiser la gestion des présences, renforcer la sécurité des accès, et fournir une traçabilité précise des activités en salle de classe.

1.2 Système de pointage actuel

Dans la majorité des établissements universitaires en Algérie, y compris à l'Université Abdelhamid Ibn Badis de Mostaganem, la gestion des présences des étudiants est réalisée par des méthodes traditionnelles principalement manuelles. Cette gestion repose généralement sur un système de pointage papier où chaque enseignant utilise une feuille de présence imprimée pour chaque cours.

Fonctionnement du système manuel de pointage :

1. Distribution des feuilles de présence

Avant ou au début de chaque séance, l'enseignant prépare une feuille comportant la liste des étudiants inscrits dans sa classe ou groupe. Cette feuille est ensuite distribuée aux étudiants pour qu'ils s'y inscrivent en signant leur nom.

2. **Signature des étudiants**

Chaque étudiant doit signer à côté de son nom ou indiquer sa présence sur la feuille. Dans certains cas, les étudiants peuvent aussi simplement écrire leur nom si la liste n'est pas fournie.

3. **Collecte et vérification**

À la fin de la séance, l'enseignant récupère la feuille de présence signée et vérifie que les signatures correspondent bien aux étudiants attendus.

4. **Saisie des données**

Souvent, ces feuilles sont ensuite transmises au secrétariat ou au service administratif, où les données de présence sont saisies manuellement dans un système informatique ou une base de données pour un traitement ultérieur.

5. **Archivage**

Les feuilles papier sont conservées comme preuve, mais peuvent être difficiles à gérer sur le long terme en raison du volume important.

Cette manière de faire est valide utilisée dans l'enregistrement des présences aux examens. Pendant les séances de TD/TP les enseignants font souvent l'appel des étudiants par leurs noms un par un et enregistrent la présence.

Limites et problèmes rencontrés

Ce système manuel, bien qu'encore largement utilisé, présente de nombreux inconvénients et limites qui affectent la qualité et la fiabilité de la gestion des présences :

- **Temps consommé** : La distribution, la collecte et la vérification des feuilles de présence prennent un temps non négligeable lors de chaque séance, réduisant le temps d'enseignement effectif.
- **Risque d'erreurs humaines** : La saisie manuelle des données dans les bases informatiques peut générer des erreurs, comme des oublis, des mauvaises saisies, ou des confusions sur les noms des étudiants.

- **Fraude et falsification** : L'un des problèmes majeurs est la possibilité pour certains étudiants de faire signer à leur place un camarade absent. La signature sur une feuille papier ne garantit pas l'identité réelle de l'étudiant présent.
- **Gestion lourde des documents** : La manipulation, le stockage et la conservation des feuilles papier sont contraignants, surtout quand le nombre d'étudiants et de cours est élevé.
- **Manque d'automatisation** : Ce système ne permet pas une exploitation facile et rapide des données de présence. Il est compliqué d'extraire des statistiques fiables, de suivre les tendances d'absentéisme, ou encore d'alerter rapidement les responsables.
- **Problèmes de traçabilité** : En cas de contestation ou de besoin de vérification, les feuilles papier peuvent être perdues ou endommagées, rendant difficile la preuve de présence ou d'absence.

1.3 Problématique

Dans le cadre universitaire, la gestion de la présence constitue un enjeu essentiel, tant sur le plan pédagogique qu'administratif. Elle permet aux enseignants de suivre l'engagement des étudiants, à l'administration de détecter les cas d'absentéisme prolongé, et de garantir une meilleure traçabilité du parcours de formation. Cependant, malgré l'importance de cet aspect, les méthodes de pointage utilisées aujourd'hui dans de nombreuses facultés demeurent basiques et peu adaptées aux exigences actuelles.

À l'Université Abdelhamid Ibn Badis de Mostaganem, comme dans beaucoup d'autres établissements, le suivi des présences repose encore sur des procédés manuels. Bien qu'acceptables dans un contexte de petite échelle, ces pratiques montrent rapidement leurs limites face à la croissance des effectifs étudiants, à la nécessité d'un accès rapide aux données, et aux exigences de fiabilité.

La multiplication des erreurs, les risques de fraude, le temps perdu lors de la collecte et de la saisie, ainsi que l'absence de statistiques en temps réel compromettent l'efficacité du système

actuel. De plus, l'absence d'un mécanisme sûr d'identification des étudiants complique la vérification de leur présence réelle, remettant en cause l'authenticité des feuilles de pointage.

Dans un environnement où les technologies numériques et biométriques sont devenues accessibles et efficaces, il apparaît légitime de s'interroger sur les moyens de moderniser ce processus.

1.4 Solution

Afin de répondre aux limites du système de pointage traditionnel et aux besoins croissants en matière de fiabilité, de rapidité et de sécurité dans la gestion de la présence des étudiants, ce mémoire propose la mise en place d'un système de gestion intelligent de la présence basé sur l'authentification biométrique, accessible via une plateforme web. L'idée centrale est de remplacer les feuilles de présence papier par un dispositif numérique automatisé, combinant une interface web conviviale pour les utilisateurs (étudiants, enseignants, administration) et un capteur biométrique d'empreintes digitales (en l'occurrence, le capteur ZK9500), afin de garantir l'authenticité et la traçabilité de chaque pointage.

Les principaux éléments de la solution sont :

- Authentification biométrique par empreinte digitale : chaque étudiant enregistre son empreinte digitale une seule fois au début de l'année. Lors de chaque cours, il s'identifie rapidement à l'aide du capteur, ce qui garantit sa présence réelle.
- Interface web centralisée : la plateforme web permet à chaque acteur (administration, enseignant, étudiant) d'interagir avec le système :
 - L'administration peut gérer les emplois du temps, les inscriptions, les groupes d'étudiants et les enseignants.
 - L'enseignant peut consulter les présences de ses étudiants, visualiser les absences, et extraire des statistiques.

- L'étudiant peut consulter son emploi du temps, voir son historique de présences et absences.
- **Base de données centralisée** : toutes les données de pointage sont enregistrées automatiquement dans une base de données, permettant un stockage sécurisé, une exploitation simplifiée, et une génération rapide de rapports.
- **Gain de temps et efficacité** : le système réduit considérablement le temps nécessaire à la prise de présence, tout en éliminant les risques d'erreurs humaines ou de fraude.

1.5 Spécification des besoins

Le présent projet consiste en la mise en œuvre d'un site web fonctionnel de gestion intelligente des présences dans les établissements scolaires, intitulé «*SmartClassroom*». Ce système repose sur l'authentification biométrique par empreinte digitale, assurée via le capteur **ZK9500**, permettant de garantir l'exactitude et la sécurité des pointages.

Le site web a été structuré autour de trois principaux espaces, chacun répondant à des besoins spécifiques :

- **Espace Administrateur**

Cet espace est réservé à la gestion centralisée du système. Il permet à l'administrateur :

- De gérer les comptes utilisateurs (enseignants et étudiant).
- D'enregistrer et associer les empreintes digitales aux identités des étudiant.
- De suivre les présences, absences à travers des tableaux de bord dynamiques.
- De configurer les emplois du temps et les modules.
- D'exporter les rapports de présence au format PDF (utile si on veut transmettre les tableaux par mail par exemple pour une raison ou une autre).

- **Espace Enseignant**

Les enseignants disposent d'un tableau de bord personnalisé leur permettant :

- De consulter la liste des étudiants de leurs cours.
- De suivre les présences enregistrées par empreinte.
- D'appliquer des filtres (par date, module) pour une analyse ciblée.

- **Espace Étudiant**

Les étudiant ont également un accès sécurisé qui leur permet de :

- S'authentifier par empreinte pour enregistrer leur présence.

Le système a été conçu pour être responsif, intuitif et sécurisé. Il permet une gestion automatisée et intelligente des présences sans recourir à des feuilles papier, tout en exploitant des données fiables issues de l'authentification biométrique.

1.5.1 Les besoins fonctionnels

Les besoins fonctionnels décrivent les actions principales que le système doit réaliser en réponse aux interactions des différents utilisateurs. Dans le cadre de notre site web de gestion de présence intelligente, ces besoins sont les suivants :

- **Espace d'accueil** : Permet d'accéder aux informations générales sur le système, telles que la description du projet, les avantages de la solution biométrique, ainsi qu'un aperçu des fonctionnalités principales.
- **Inscription des utilisateurs** : Les étudiant et enseignants sont enregistrés dans le système par l'administrateur avec leurs informations personnelles (nom, prénom, etc.). Les empreintes digitales sont ensuite associées aux étudiant via le capteur ZK9500.
- **Authentification biométrique** : Les étudiant valident leur présence en classe en s'authentifiant avec leur empreinte digitale. L'empreinte est comparée aux données stockées pour vérifier l'identité et enregistrer l'heure de pointage.
- **Accès au tableau de bord personnel** :
 - **Pour les enseignants** : Affichage de la liste des étudiant par classe, suivi des présences, consultation des absences et export des données.
 - **Pour les étudiant** : Consultation de l'historique de présence, statistiques personnelles, et notifications en cas d'anomalie.
- **Gestion des utilisateurs** : Permet à l'administrateur de gérer les comptes enseignants et étudiant ainsi que de modifier ou supprimer leurs informations selon les besoins.
- **Enregistrement des présences** : Le système enregistre automatiquement les présences à l'aide du lecteur d'empreintes. Chaque pointage est associé à un étudiant une date et une heure précise.
- **Filtrage des présences** : Possibilité d'effectuer des recherches avancées par étudiant par date ou par module, afin de faciliter le suivi ciblé des absences.

- **Génération de rapports** : Le système permet de générer des rapports de présence avec possibilité d'export au format PDF .
- **Recherche rapide** : L'administrateur ou l'enseignant peut rechercher un étudiant ou un pointage en saisissant un mot-clé dans un champ dédié.

1.5.2 Les besoins non fonctionnels

Les besoins non fonctionnels définissent les exigences qualitatives de l'application web. Bien que non liés directement aux fonctionnalités visibles, ils garantissent la stabilité, la sécurité, et la qualité d'utilisation de notre système de gestion de présence basé sur l'empreinte digitale. Voici ceux qui s'appliquent à notre site web :

- **Disponibilité continue** : Le site doit rester accessible à tout moment, notamment durant les horaires scolaires, afin de permettre l'enregistrement instantané de la présence via le capteur d'empreinte (ZK9500) connecté au système.
- **Fiabilité** : Le système doit offrir une précision étudiant dans la reconnaissance des empreintes et l'enregistrement de la présence, sans pertes ni duplications de données.
- **Gestion des erreurs** : Des messages clairs doivent guider l'utilisateur en cas d'erreurs échec d'authentification, empreinte inconnue, ou toute anomalie du système.
- **Sécurité** : Les données des étudiants (informations personnelles et empreintes digitales) doivent être protégées par des mécanismes robustes de chiffrement, d'authentification et de gestion des accès. Aucune donnée sensible ne doit être exposée.
- **Ergonomie et interface conviviale** : Le site doit présenter une interface simple, accessible et bien structurée, facilitant l'utilisation par les enseignants, les administrateurs et les utilisateurs sans connaissances techniques.
- **Facilité de maintenance et d'évolution** : L'architecture de l'application doit permettre l'ajout futur de nouvelles fonctionnalités (comme des statistiques d'assiduité ou alertes automatiques) sans modifier profondément l'existant.

1.6 Architecture de notre système

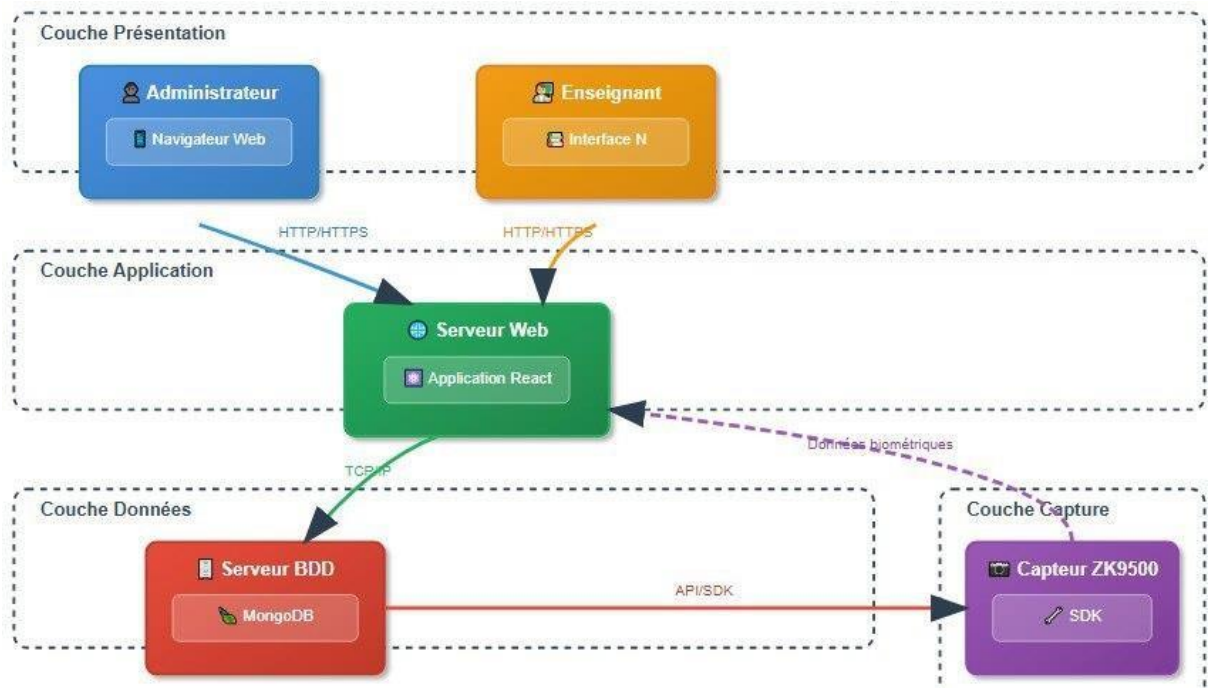


Figure 1: Architecture du système

1.7 Conclusion

Ce premier chapitre nous a permis d'introduire notre projet de gestion intelligente des présences dans un environnement éducatif, basé sur l'authentification biométrique via empreinte digitale. Nous avons identifié les enjeux liés à la digitalisation du suivi de présence et l'importance d'un système sécurisé, fiable et accessible à travers une plateforme web. Nous avons également analysé les besoins fonctionnels et non fonctionnels du système, en mettant l'accent sur l'interaction entre les utilisateurs (administrateurs, enseignants, étudiants) et les différentes fonctionnalités offertes par le site web. Cette analyse constitue la base essentielle pour la suite du développement, notamment la modélisation UML et la conception technique qui seront abordées dans le prochain chapitre.

Chapitre 02 :

Modélisation du système

2.1 Introduction

Dans ce chapitre, nous présentons la conception du système de gestion de présence scolaire basé sur l'authentification biométrique via empreinte digitale. Cette phase de conception est essentielle, car elle permet de structurer les différentes fonctionnalités du système, d'organiser les interactions entre les acteurs, et de modéliser les données qui seront traitées. La modélisation est réalisée à l'aide du langage UML (Unified Modeling Language), qui offre une représentation graphique claire et normalisée des éléments du système.

Comment modéliser notre système. On commence par les cas d'utilisation, et on utilisera les diagrammes de séquence pour expliquer le fonctionnement du système. Ensuite on va présenter le modèle conceptuel et le déploiement de l'application.

2.2 Choix de la méthode de conception

Lorsqu'on conçoit un système informatique, deux grandes approches sont possibles. La première, appelée approche fonctionnelle, considère le système comme un ensemble de fonctions à réaliser. La seconde, l'approche objet, voit le système comme un ensemble d'objets interagissant entre eux.

Dans notre projet, nous avons opté pour l'approche objet, car elle permet de mieux représenter les éléments réels du système, comme les étudiants, les enseignants ou encore les présences, sous forme d'objets ayant des attributs et des comportements spécifiques.

Pour modéliser notre projet de manière claire et structurée, nous avons choisi d'utiliser le langage UML. Ce langage universel nous permet de représenter les différents aspects du système de façon visuelle et compréhensible.

Notre démarche de conception se déroule en plusieurs étapes :

Nous commençons par le diagramme de contexte statique, qui offre une vue d'ensemble du système en identifiant ses principaux composants et leurs relations avec l'environnement extérieur. Ensuite, le diagramme de cas d'utilisation met en évidence les interactions principales entre les utilisateurs et le système. Nous poursuivons avec le diagramme de séquence, qui détaille l'enchaînement des échanges dynamiques entre les différents éléments du système lors de scénarios spécifiques. Puis, le diagramme de classes présente la structure interne du système à travers les classes, leurs attributs et leurs relations. Enfin, nous terminons par le diagramme de déploiement, qui illustre la répartition physique des composants logiciels et matériels du système.

2.3 Conception détaillée

2.3.1 Diagramme de contexte statique

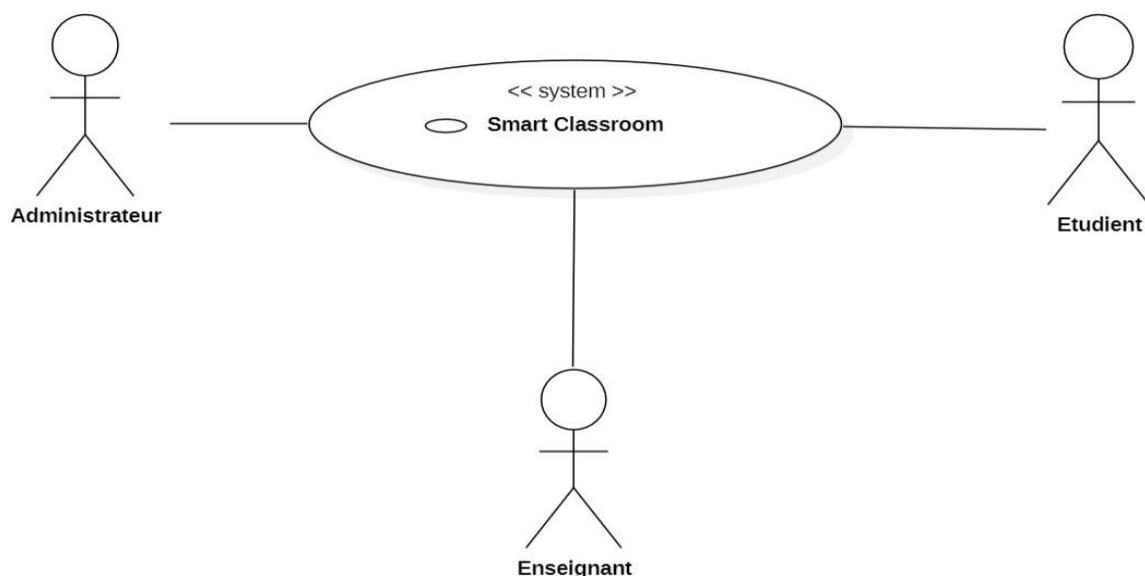


Figure 2 : Diagramme de contexte statique

2.3.2 Diagramme de cas d'utilisation

Le Diagramme de cas d'utilisation représente une fonctionnalité spécifique dans un système et est créé pour illustrer comment différentes fonctionnalités sont interconnectées et montrer leurs contrôleurs (ou acteurs) internes et externes. [1]

a. Diagramme de cas d'utilisation « Administrateur »

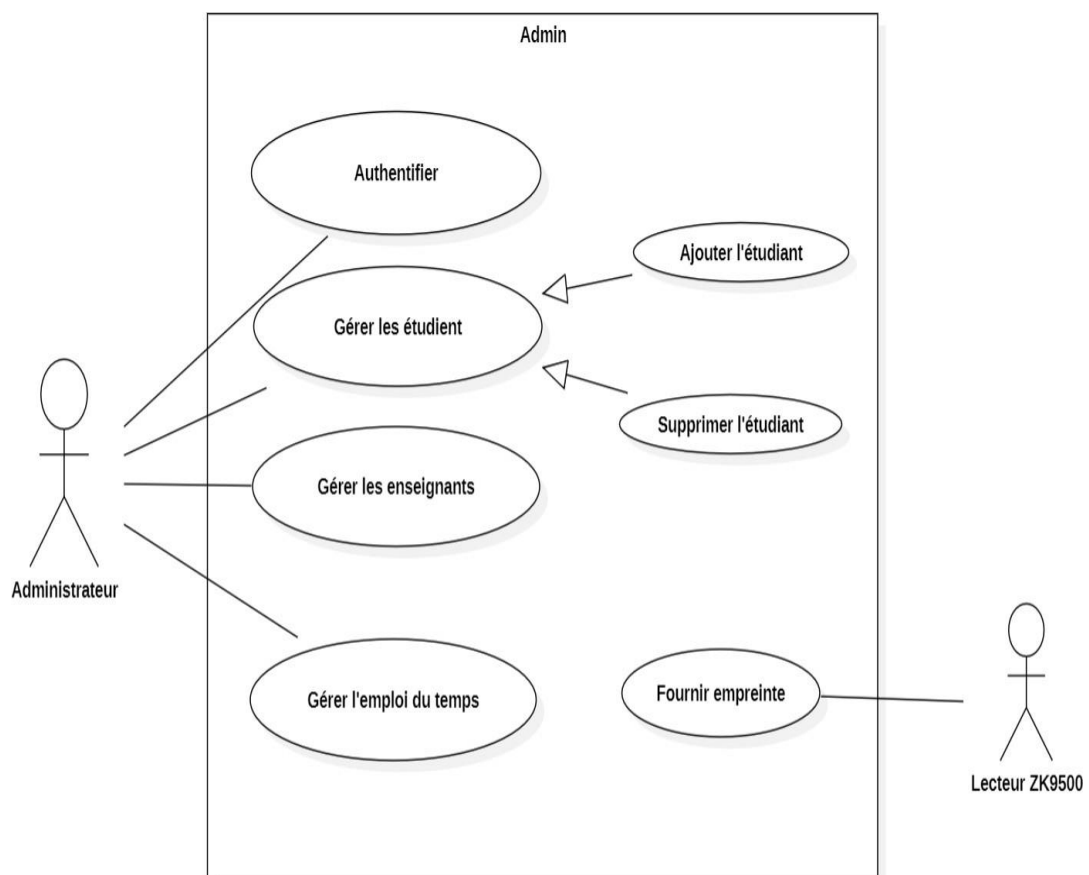


Figure 3 : Diagramme de cas d'utilisation « Administrateur »

b. Diagramme de cas d'utilisation « Enseignant »

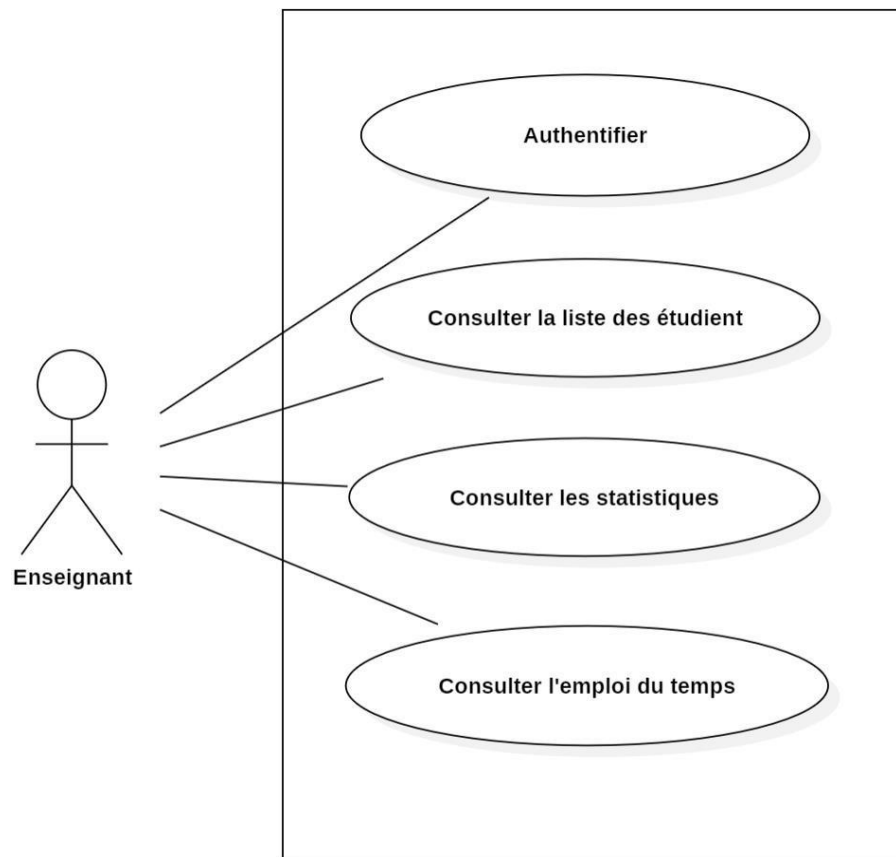


Figure 4 : Diagramme de cas d'utilisation « Enseignant »

c. Diagramme de cas d'utilisation « Étudiant »

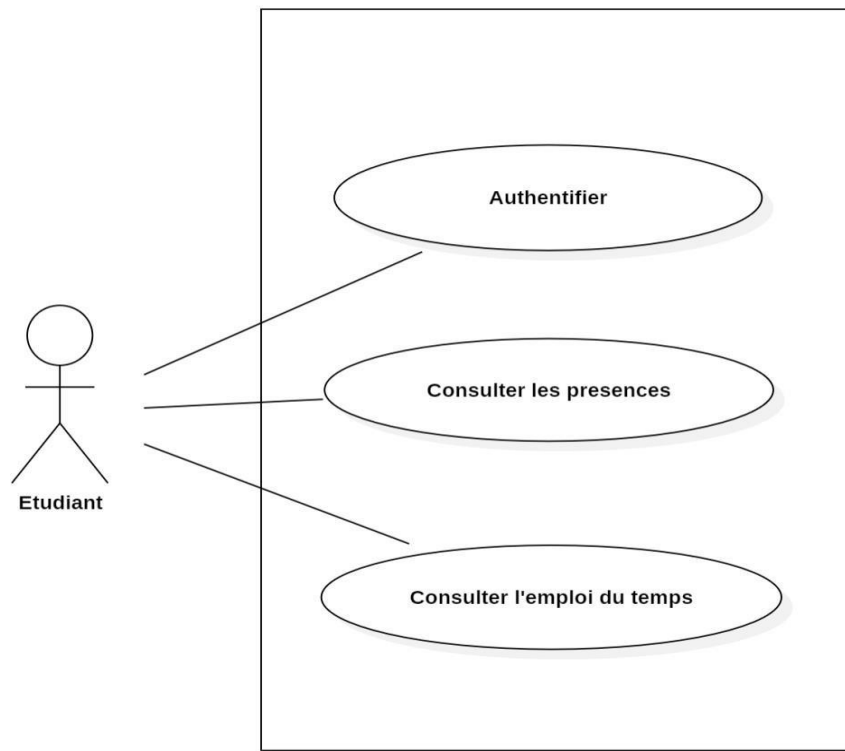


Figure 5 : Diagramme de cas d'utilisation « Etudiant »

2.3.3 Diagramme de séquence

Diagramme de séquence montre comment les objets interagissent les uns avec les autres et dans quel ordre. Ils représentent les interactions d'un scénario particulier.

Dans cette section, nous allons présenter quelques exemples de diagrammes de séquence. Ces schémas permettront de visualiser, de manière chronologique, comment les messages circulent entre les différents objets du système et les utilisateurs, selon les scénarios d'utilisation.[2]

a- Diagramme de séquence « Authentification »

L'authentification consiste à assurer la confidentialité des données, où le système effectue une vérification des informations associées à chaque utilisateur (généralement un nom d'utilisateur un mot de passe).

Ces informations sont préétablies dans le système. Si les informations fournies sont correctes, alors le système accorde l'accès à l'interface appropriée. En revanche, si l'utilisateur saisit des informations incorrectes, le système génère un message d'erreur et affiche un message d'erreur demandant de rectifier les informations saisies.

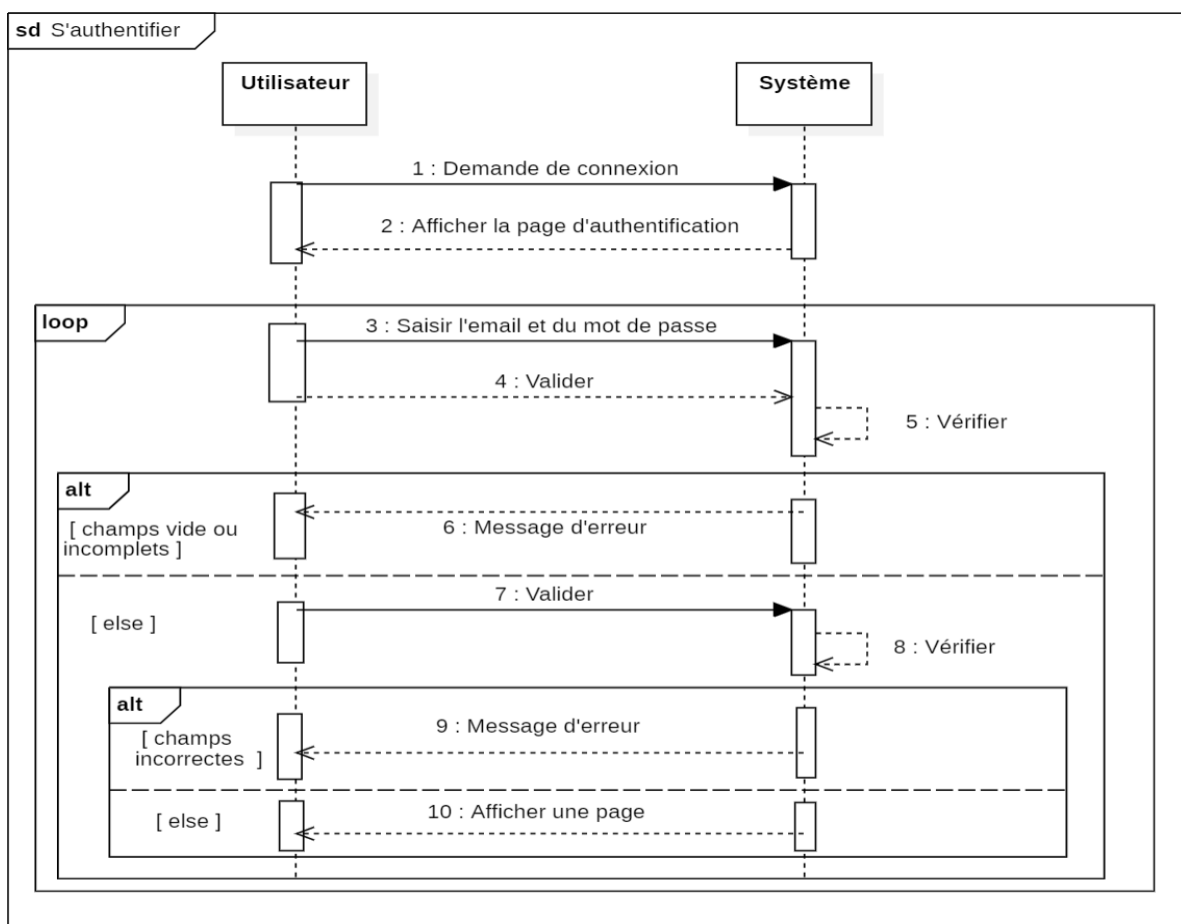


Figure 6 : Diagramme de séquence « Authentification »

b- Diagramme de séquence « Enregistrement d'un étudiant »

Ce diagramme illustre le processus d'enregistrement d'un nouvel étudiant dans le système. Il montre les interactions entre l'administrateur, le système et la base de données. Tout d'abord, l'administrateur saisit les informations d'étudiant. Le système procède alors à une vérification dans la base de données pour s'assurer de l'unicité de l'e-mail fourni.

Deux cas peuvent se présenter :

- Si les données sont valides et uniques, l'administrateur procède à la capture de l'empreinte digitale de l'étudiant. Une fois l'empreinte obtenue, le système enregistre l'ensemble des données dans la base de données et affiche un message de confirmation.
- Si les données sont invalides ou déjà enregistrées, le système informe immédiatement l'administrateur à travers un message d'erreur, l'invitant à corriger les informations saisies.

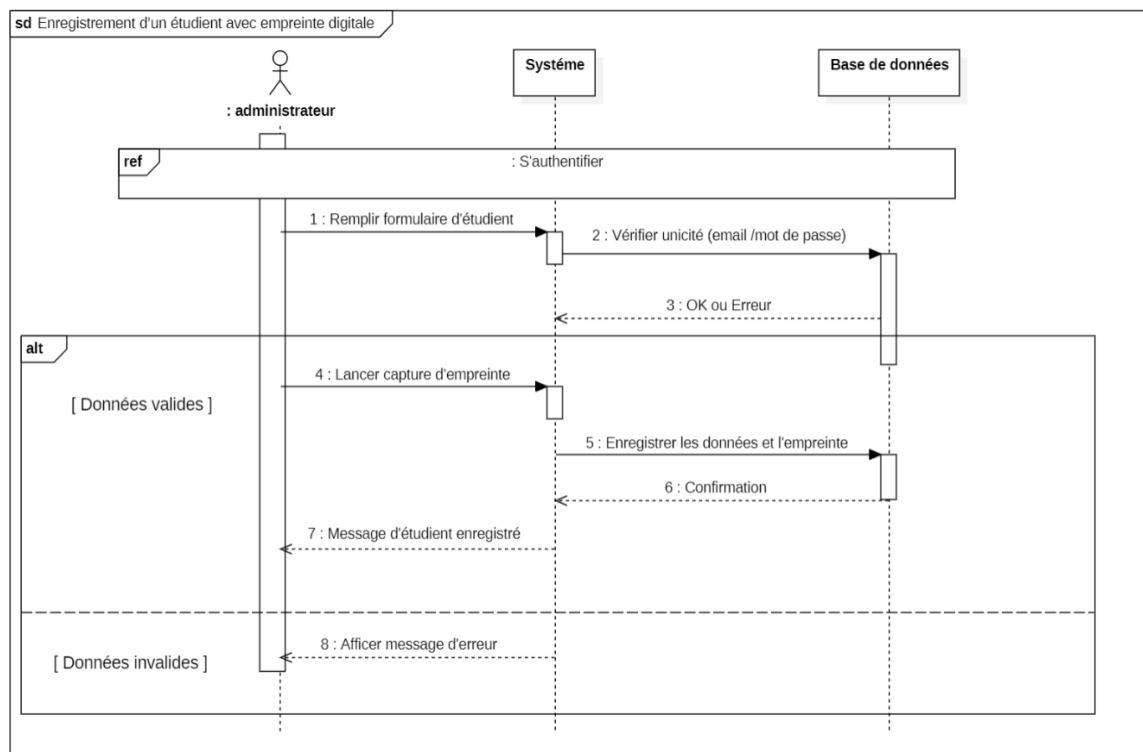


Figure 7 : Diagramme de séquence « Enregistrement d'un étudiant »

c- Diagramme de séquence « Enregistrement de la présence d'étudiant »

Ce diagramme illustre le processus par lequel un étudiant enregistre sa présence à un cours à l'aide du capteur biométrique ZK9500. L'étudiant commence par poser son doigt sur le capteur. Ce dernier lit l'empreinte digitale et la transmet au système central. Le système interroge alors la base de données pour comparer l'empreinte reçue avec celles déjà enregistrées.

Deux scénarios sont possibles :

- Empreinte reconnue : si une correspondance est trouvée, le système enregistre automatiquement la présence de l'étudiant, en incluant les informations telles que la date, l'heure et la séance concernée. Une confirmation est ensuite affichée à l'étudiant pour signaler la réussite de l'enregistrement.
- Empreinte non reconnue : si l'empreinte ne correspond à aucun étudiant enregistré, le système envoie un message d'erreur indiquant l'échec de l'identification.

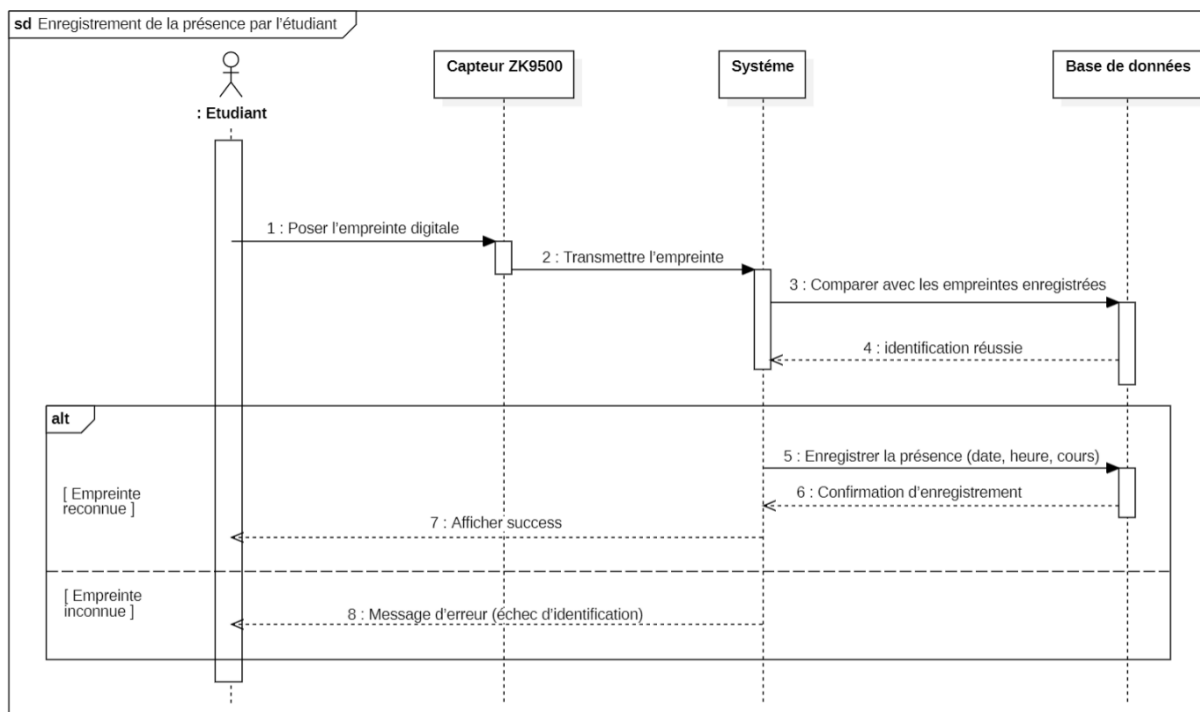


Figure 8 : Diagramme de séquence « Enregistrement de la présence d'étudiant »

d- Diagramme de séquence « Consultation des présences par l'enseignant »

Ce diagramme décrit le scénario dans lequel un enseignant souhaite consulter les présences des étudiant à ses cours. L'interaction débute lorsque l'enseignant effectue une requête vers le système afin d'obtenir la liste des présences. Le système interroge alors la base de données pour récupérer les données de présence correspondantes à la séance ou à la période demandée. Une fois les données extraites, la base de données renvoie les informations au système, qui se charge ensuite de les présenter à l'enseignant. L'interface affiche alors non seulement la liste des étudiant présents et absents, mais aussi des statistiques utiles, telles que les taux de présence ou d'absences.

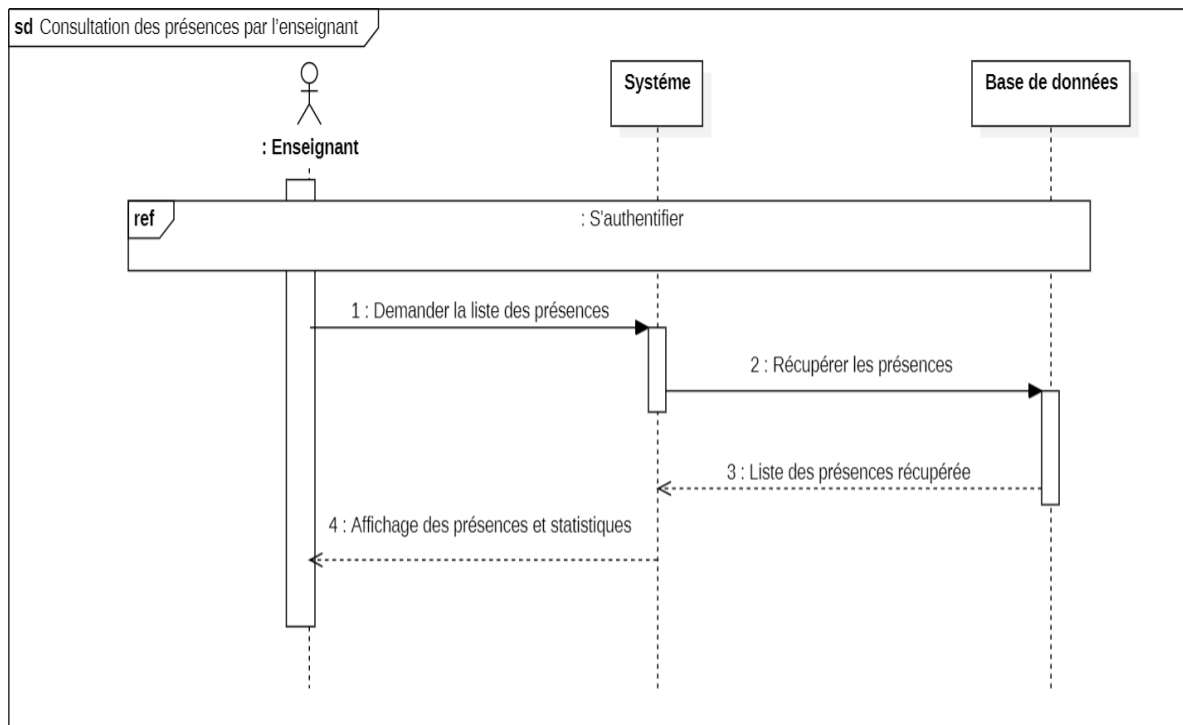


Figure 9 : Diagramme de séquence « Consultation des présences par l'enseignant »

2.3.4 Diagramme de classe

Le diagramme de classes est le diagramme UML le plus couramment utilisé et fondement de toute solution orientée objet. La figure ci-dessus illustre le diagramme de classe de notre système.[3]

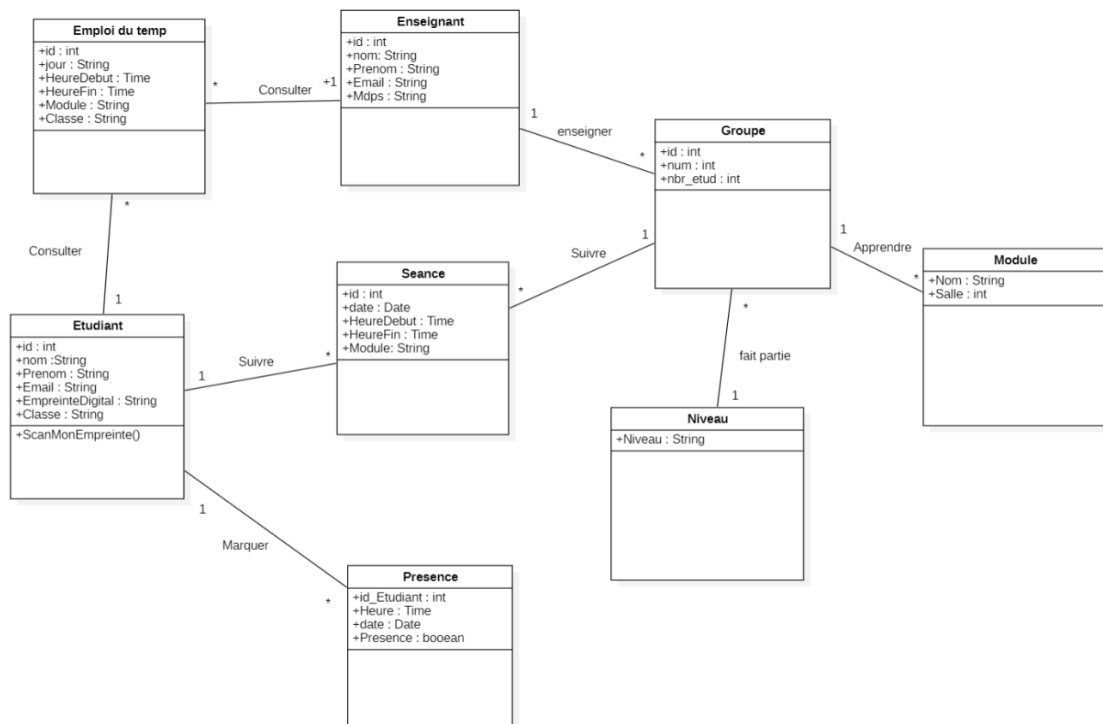


Figure 10 : Diagramme de Classe

2.3.5 Diagramme de déploiement

Diagramme de déploiement Illustre les infrastructures physique et logicielle d'un système. Ils sont utiles lorsqu'une solution logicielle est déployée sur de nombreuses machines avec des configurations uniques.[4]

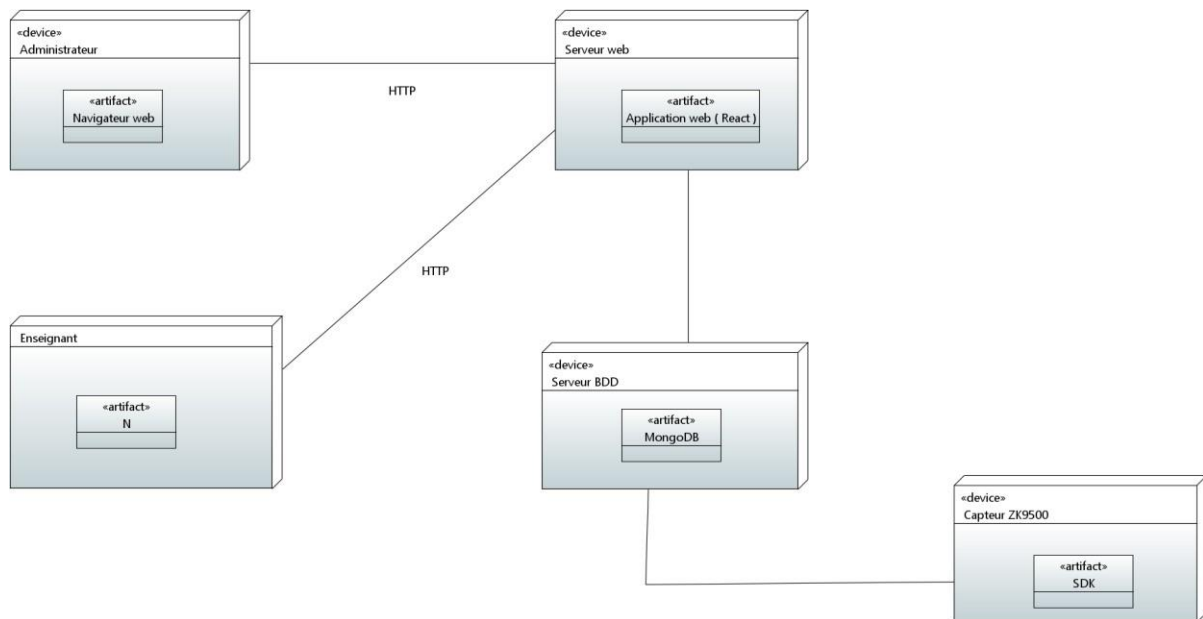


Figure 11 : Diagramme de déploiement

2.4 Capteur ZK9500 (ZKTeco)[5]

ZK9500, un dernier Lecteur optique d'empreintes digitales, est développé par ZKTeco. Il prend en charge la détection des doigts et capture rapidement les empreintes digitales, avec une haute résolution.

Le ZK9500 ZKTeco est petit, facile à utiliser pour l'enregistrement des empreintes digitales sur le bureau. Faible consommation d'énergie, il s'applique également aux tablettes et téléphones portables Android.

- Soutenir la fonction de détection des empreintes digitales
- Reconnaissance rapide des empreintes digitales sèches, humides et rugueuses
- SDK disponible pour le développement
- USB 2.0 haute vitesse .

2.5 L'algorithme utilisé par le capteur ZK9500

a. Enregistrement de l'étudiant (phase d'entraînement)

Cette étape se fait une seule fois pour chaque étudiant :

1. L'administrateur demande à l'étudiant de poser son doigt sur le capteur ZK9500.
2. Le capteur capture l'image de l'empreinte digitale.
3. Il transforme cette empreinte en **chaîne Base64**
4. L'image encodée est envoyée au backend via une **requête HTTP POST**.
5. Le backend enregistre l'empreinte dans la base de données **MongoDB**, accompagnée des informations de l'étudiant (nom, prénom, identifiant, etc.).

b. Prise de présence (reconnaissance de l'empreinte)

Cette étape se fait chaque fois que l'étudiant veut enregistrer sa présence :

1. L'étudiant pose son doigt sur le capteur.
2. Le capteur capture une nouvelle image de l'empreinte et la convertit en **Base64**.
3. L'empreinte encodée est envoyée au backend via une **requête HTTP**.
4. Le backend compare cette empreinte avec celles enregistrées dans la base de données.
5. Si l'empreinte est reconnue :
 - Le système trouve le bon étudiant.
 - Il enregistre la présence (avec la date, l'heure, etc.).
 - Un **son ou message de confirmation** est émis.
6. Si l'empreinte n'est pas reconnue :
 - Le système affiche un message d'erreur (Empreinte inconnue)

2.6 Conclusion

Dans ce chapitre, nous avons posé les fondations de la modélisation de notre application. Nous avons utilisé la méthode orienté objet pour le développement d'une application web structurée et efficace.

Nous avons également présenté les principaux diagrammes utilisés dans notre modélisation, qui nous ont permis de mieux comprendre le fonctionnement du système. Bien qu'une modélisation complète puisse inclure d'autres types de diagrammes, ceux abordés ici nous ont fourni une vision globale et cohérente du projet.

Cette phase de conception représente une étape clé avant de passer à la réalisation concrète de l'application, qui sera détaillée dans le chapitre suivant.

Chapitre 03 :

Réalisation & Implémentation

3.1 Introduction

Ce chapitre est consacré à la réalisation concrète de notre site web, en s'appuyant sur la modélisation effectuée précédemment. Nous allons d'abord présenter l'environnement de développement que nous avons choisi, ainsi que les outils et langages utilisés tout au long du projet. Ensuite, nous décrirons les différentes interfaces du site, afin de montrer comment les utilisateurs interagissent avec le système dans sa version finale.

3.2 Environnement logiciel

Au cours du processus de création de cette application, nous avons employé les logiciels suivants. React, VS code,..etc. Les différents logiciels seront exposés dans la section suivante .

3.2.1 Java Spring Boot

Spring Boot est un framework Java open source conçu pour simplifier le développement d'applications backend en proposant une configuration automatique, une architecture modulaire et une prise en charge native des services web RESTful. Il permet de créer rapidement des API robustes, prêtes pour le déploiement en production. Dans le cadre du projet SmartClassroom, Spring Boot est utilisé pour implémenter le cœur logique de l'application, assurant la communication entre l'interface utilisateur (développée en React.js) et la base de données MongoDB. Il gère les principales fonctionnalités du système telles que l'administration des comptes (étudiant, enseignants, administrateurs), la gestion des emplois

du temps, des modules, des classes, ainsi que l'enregistrement des présences via un lecteur d'empreintes digitales. Son intégration avec des outils de sécurité, de gestion des sessions et de déploiement rapide garantit un backend fiable, scalable et parfaitement adapté à un environnement éducatif intelligent.[6]



Figure 12 : Logo Springboot

3.2.2 MongoDB

MongoDB est une base de données NoSQL orientée documents, conçue pour stocker les données sous forme de documents BSON flexibles, ce qui élimine la contrainte d'un schéma rigide. Cette caractéristique facilite la gestion de structures de données complexes et évolutives, tout en assurant des performances élevées même pour de grands volumes de données non structurées. Dans le projet SmartClassroom, MongoDB est employée comme base de données principale pour centraliser et gérer les informations relatives aux étudiants, enseignants, emplois du temps, présences, modules et empreintes digitales. Son intégration avec Spring Data MongoDB permet d'optimiser les opérations de lecture, d'écriture et de mise à jour, garantissant ainsi une persistance fiable, rapide et scalable des données. Ce choix technologique répond aux exigences d'un système intelligent de gestion pédagogique, où la flexibilité, la cohérence et la performance sont essentielles. [7]



Figure 13 : Logo MongoDB

3.2.3 React

React.js est une bibliothèque JavaScript développée par Meta, conçue pour construire des interfaces utilisateurs dynamiques à partir de composants réutilisables. Son architecture basée sur un DOM virtuel permet d'optimiser les performances et d'assurer une mise à jour fluide de l'affichage en temps réel. Dans le cadre du projet SmartClassroom, React.js est utilisé pour développer l'interface utilisateur destinée aux administrateurs, enseignants et étudiant. Il permet de concevoir des pages modulaires comme le tableau de bord, la gestion des présences, des utilisateurs, des emplois du temps et des rapports. Pour faciliter la communication avec le backend, la bibliothèque Axios est intégrée, permettant d'envoyer efficacement des requêtes HTTP vers les API REST. De plus, la bibliothèque Recharts est utilisée pour la création de graphiques interactifs et dynamiques, notamment dans les rapports de présence, offrant ainsi une visualisation claire et engageante des données. L'approche composant de React facilite la maintenance et l'évolution de l'interface, tout en garantissant une navigation fluide et réactive entre les différentes sections du système. Ce choix technologique permet ainsi d'offrir une expérience utilisateur moderne, interactive et parfaitement adaptée aux exigences d'un système de gestion pédagogique intelligent. [8].

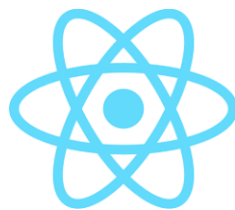


Figure 14 : Logo React

3.2.4 VS code

Visual Studio Code est un éditeur de code source léger et performant développé par Microsoft, compatible avec Windows, Linux et macOS. Il supporte de nombreux langages de programmation et offre une large gamme d'extensions, notamment pour Java, Spring Boot,

MongoDB et React, ce qui facilite le développement multi-technologies. Dans le cadre du projet SmartClassroom, VS Code est l'environnement principal utilisé pour écrire, organiser et maintenir le code du frontend et du backend. Ses fonctionnalités intégrées telles que l'autocomplétion intelligente, le débogage avancé, le contrôle de version Git et le terminal intégré optimisent la productivité des développeurs et simplifient la gestion du projet.[9]



Figure 15 : Logo VS code

3.2.5 IntelliJ IDEA Community Edition

IntelliJ IDEA Community Edition est une version gratuite de l'environnement de développement intégré (IDE) IntelliJ IDEA, spécialement conçue pour le développement d'applications Java et Android. Il offre des fonctionnalités clés telles que la complétion intelligente de code, la navigation rapide, le refactoring, la gestion de projets, ainsi que l'intégration avec des systèmes de contrôle de version comme Git. Dans le cadre du projet SmartClassroom, IntelliJ IDEA a été utilisé pour développer et tester l'application desktop liée au lecteur d'empreintes digitales ZKTeco, permettant la capture et l'envoi des données biométriques au frontend, ainsi que la vérification des présences des utilisateurs.[10]



Figure 16 : Logo IntelliJ IDEA

3.2.5 Postman

Postman est un outil puissant doté d'une interface graphique intuitive, qui permet aux développeurs d'envoyer des requêtes HTTP, de tester et valider les end points des API, d'automatiser les tests ainsi que de documenter les services web. Dans le cadre du projet SmartClassroom , Postman a été utilisé pour vérifier et déboguer les API RESTful développées avec Spring Boot, garantissant ainsi la fiabilité et la conformité des échanges entre le backend et le frontend. Son utilisation a été essentielle durant toutes les phases du développement, depuis la conception des API jusqu'à leur mise en production et leur maintenance.[11]



Figure 17 : Logo postman

3.2.7 Tailwind CSS

Tailwind CSS est un framework CSS utilitaire qui permet de styliser les interfaces utilisateur en appliquant directement des classes prédéfinies aux éléments HTML, évitant ainsi l'écriture de feuilles de style personnalisées. Contrairement aux frameworks traditionnels basés sur des composants tout prêts, Tailwind propose une approche “bas niveau” offrant une grande flexibilité de conception tout en maintenant une structure claire et modulable. Dans le cadre du projet SmartClassroom, Tailwind CSS a été utilisé pour créer une interface moderne, responsive et cohérente. Il a facilité le développement rapide des différents écrans du frontend, notamment le tableau de bord principal, la page de

connexion et les espaces dédiés aux différents profils utilisateurs. Grâce à Tailwind, la mise en page des éléments tels que grilles, boutons et cartes a pu être réalisée efficacement, avec un code plus maintenable et une personnalisation simplifiée.[12]



Figure 18 : Logo Tailwind CSS

3.2.8 Node.js

Node.js est une plateforme JavaScript côté serveur qui permet d'exécuter du code JavaScript en dehors d'un navigateur. Dans notre projet SmartClassroom, Node.js est utilisé principalement comme environnement d'exécution pour le développement frontend avec React.js, notamment pour gérer les paquets via npm, exécuter le serveur de développement et construire l'application. Cette utilisation facilite la gestion des dépendances, l'automatisation des tâches et le déploiement de l'interface utilisateur.

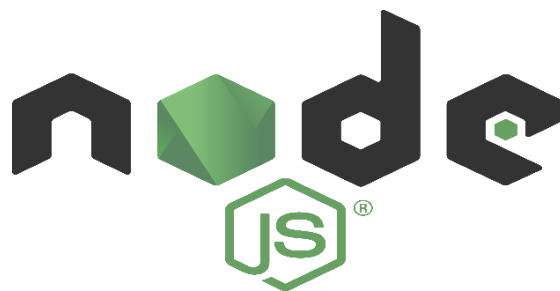


Figure 19 : Logo Node.js

3.3 Capteur utilisé

Le ZK9500 est un capteur biométrique d'empreintes digitales conçu par ZKTeco. Il se connecte à un ordinateur via USB et sert exclusivement à la capture d'empreintes, principalement pour l'enregistrement dans des systèmes d'authentification. Il ne fonctionne pas de manière autonome : il ne peut ni stocker des empreintes ni contrôler des dispositifs comme une porte. Pour son intégration dans une application, l'usage du SDK ZKTeco est indispensable. Ce capteur est prévu pour un usage en intérieur, avec une durée de vie d'environ 1 million de lectures. Il n'est ni étanche ni résistant aux chocs, et ne doit pas être installé à l'extérieur ou en environnement difficile.[13]



Figure 20 : Zk9500 (Zkteco)

3.4 Structure de la base de données

- **Pourquoi MongoDB ?**

Le choix de MongoDB comme système de gestion de base de données repose sur plusieurs considérations techniques et fonctionnelles. En tant que base de données NoSQL orientée documents, MongoDB offre une grande souplesse dans la modélisation des données, ce qui permet d'adapter aisément la structure des enregistrements aux besoins évolutifs de l'application. Contrairement aux bases relationnelles classiques, MongoDB ne nécessite pas de schéma rigide, facilitant ainsi l'intégration de nouveaux types de données sans refonte majeure. Cette flexibilité est particulièrement pertinente dans un projet tel que le nôtre, où les données biométriques, les horaires, les historiques de présence et les profils utilisateurs coexistent avec des structures variées. Par ailleurs, la performance en lecture/écriture, la scalabilité horizontale et la compatibilité avec les environnements JavaScript renforcent la pertinence de MongoDB dans un contexte de développement web moderne.

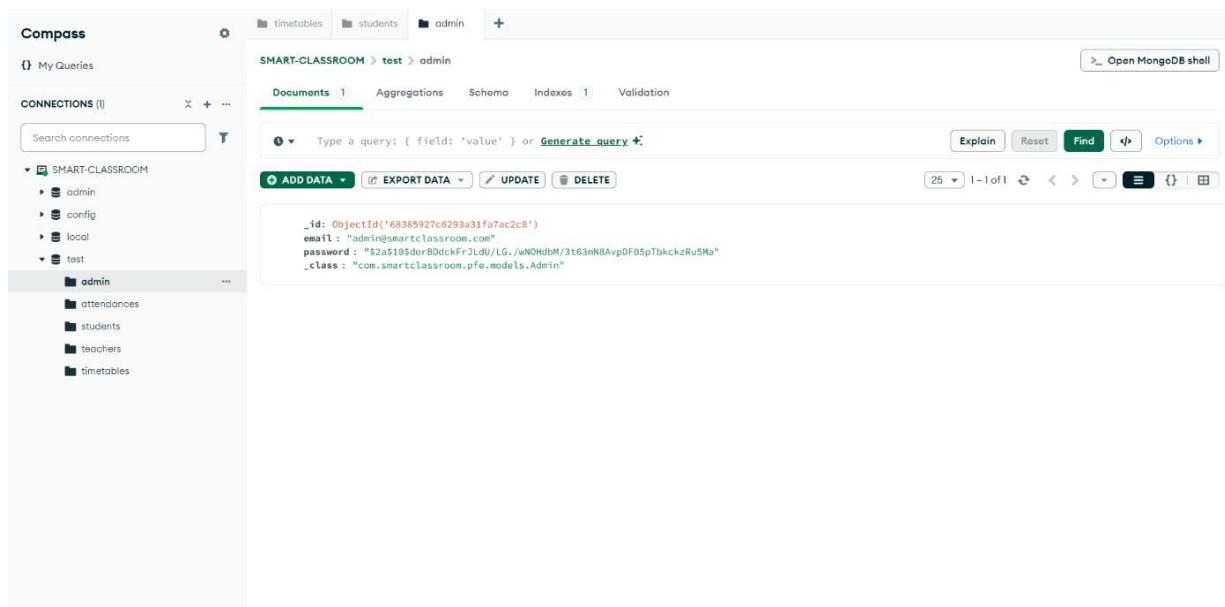


Figure 21 : Document MongoDB Admin

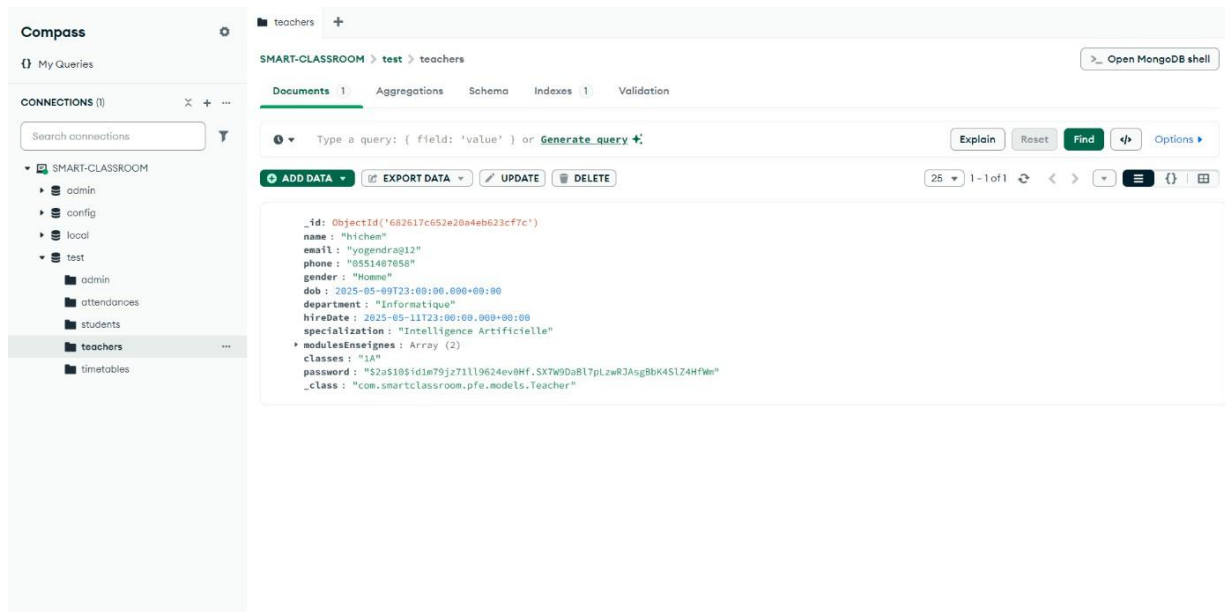


Figure 22 : Document MongoDB Enseignant

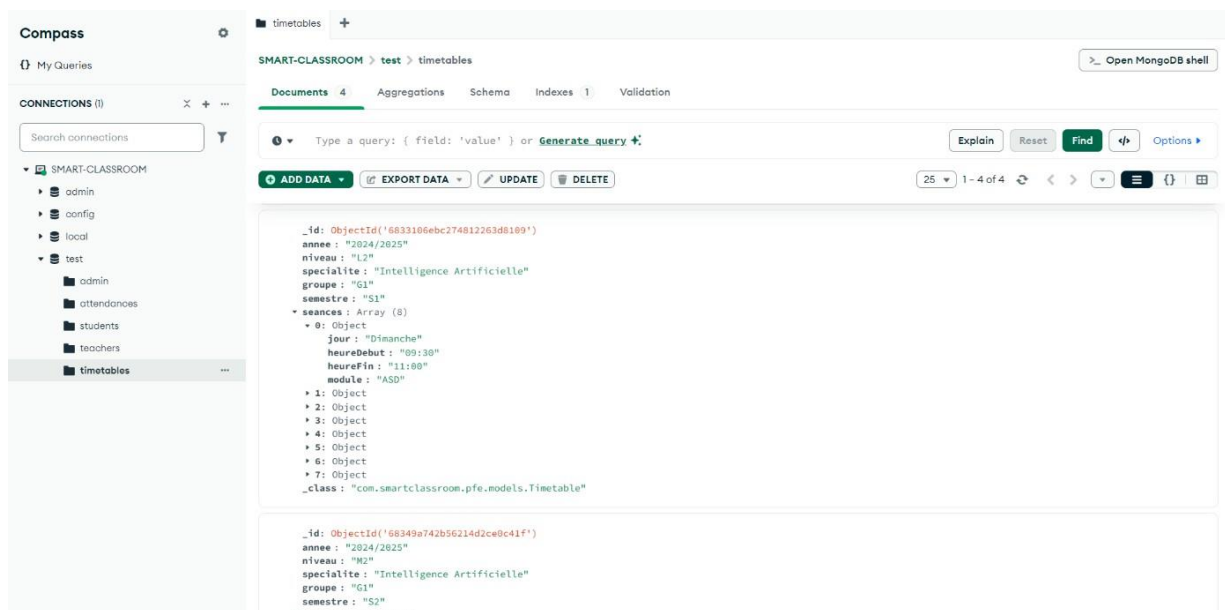


Figure 23 : Document MongoDB Emploi du temps

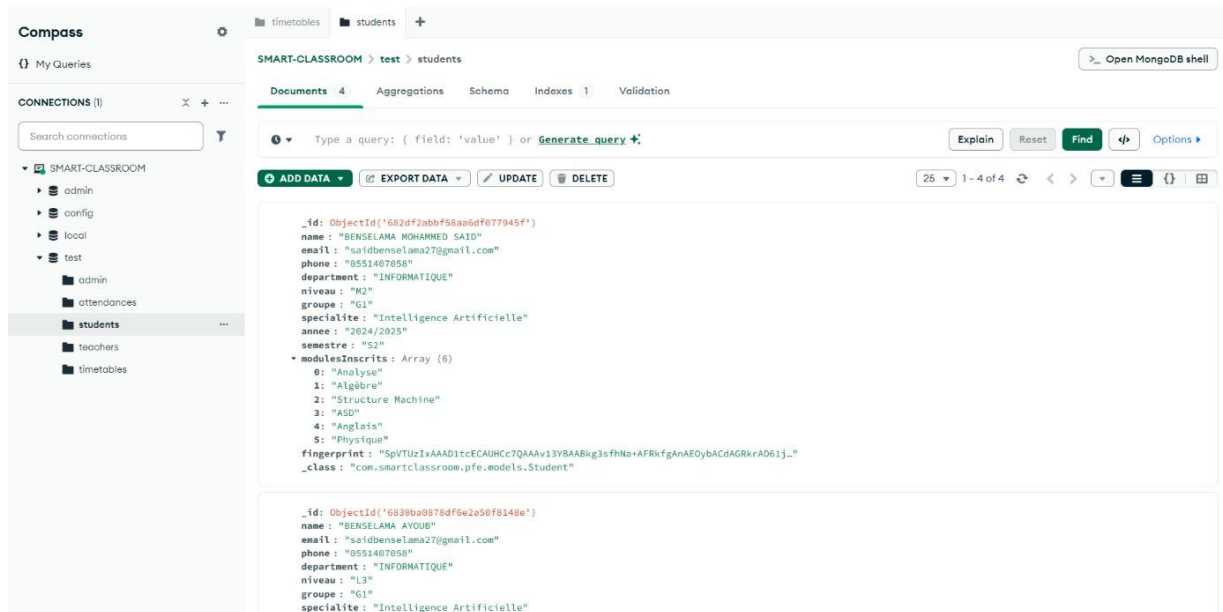


Figure 24 : Document MongoDB Etudiant

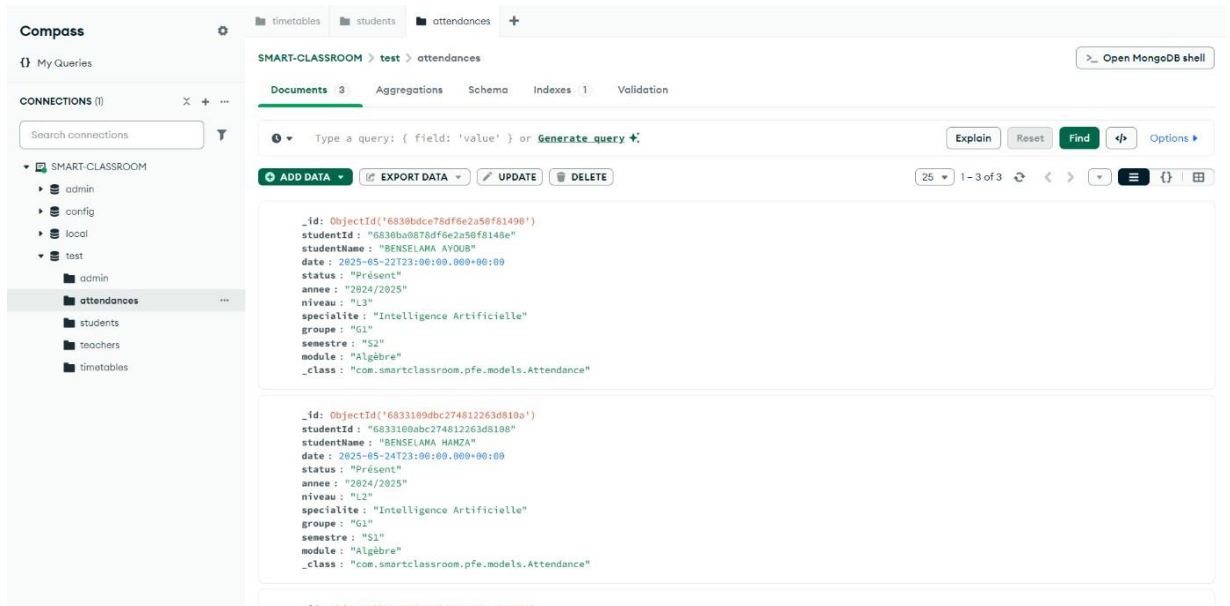


Figure 25 : Document MongoDB Marque de présence

3.5 Interfaces de l'application

Nous présenterons dans ce qui suit, les captures d'écran des principales interfaces réalisées sur notre site web.

3.5.1 Page d'accueil du site :

Cette interface est accessible à partir de n'importe quel type d'utilisateur. Elle représente l'idée du site.

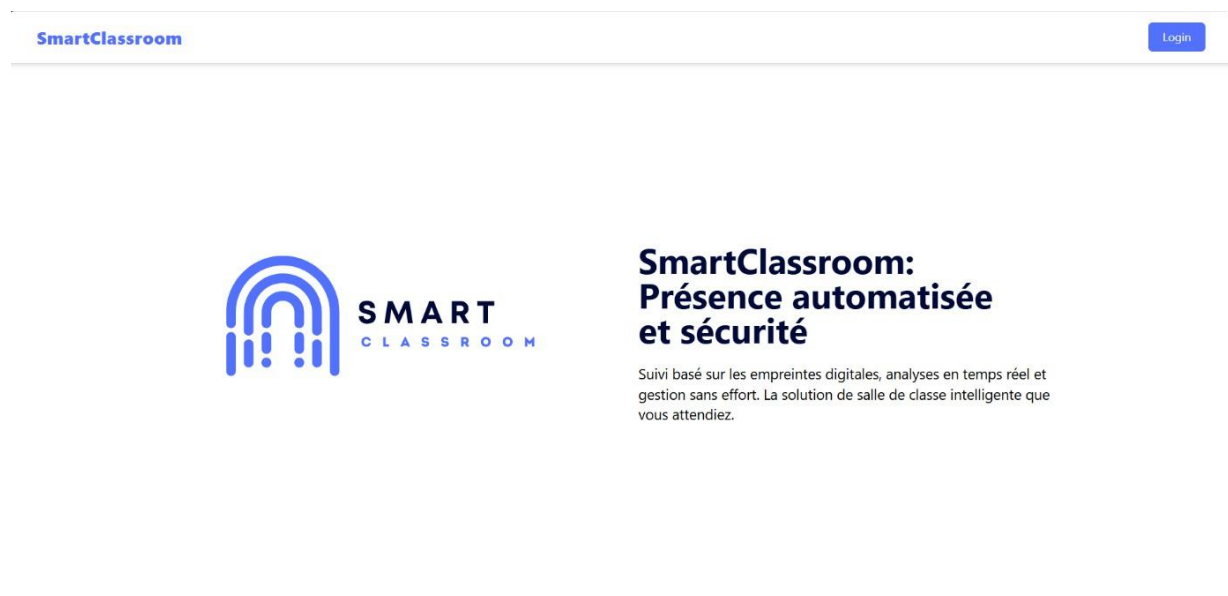


Figure 26 : Smart Classroom d'accueil [16]

3.5.2 Page de connexion :

Dans cette interface, l'utilisateur trouve un champ pour l'adresse e-mail et un autre pour le mot de passe et choisir un rôle afin d'accéder à son compte s'il en possède un sur notre site.



Bienvenue dans SmartClassroom

Email

admin@smartclassroom.com

Mot de passe

.....

Choisir un rôle

Administrateur Enseignant Élève

Se connecter

[Pas encore de compte ? Contactez l'administrateur](#)

Figure 27 : Smart Classroom Se connecter

3.5.3 Page de Tableau de Bord :

Dans cette interface affiche les statistiques de présence des étudiant.

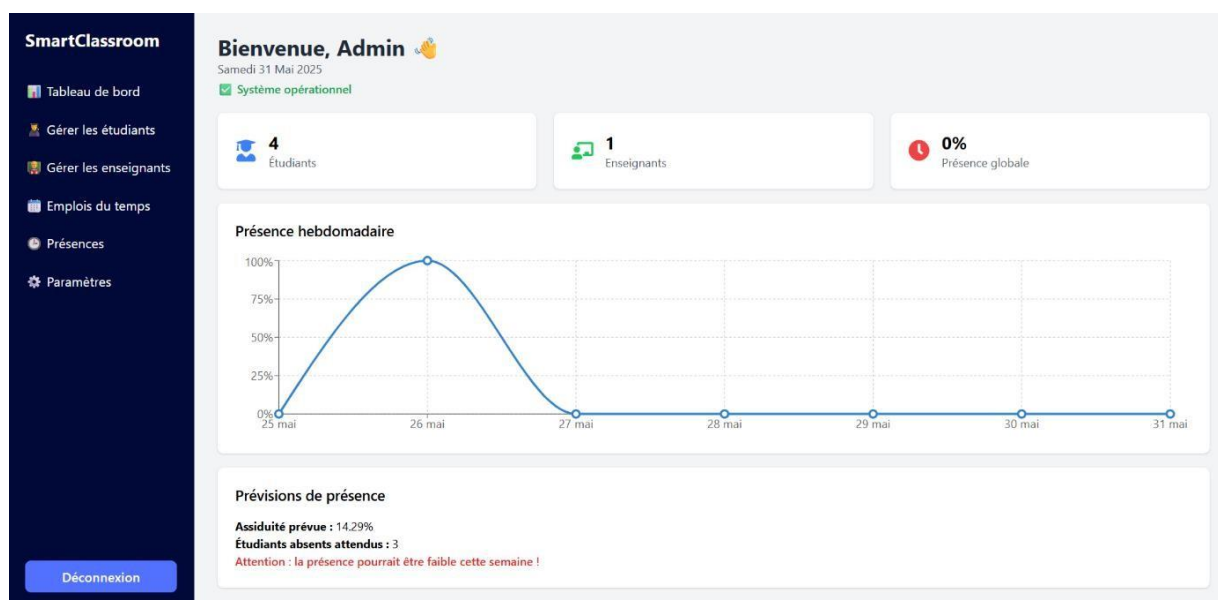


Figure 28 : Tableau de Bord

3.5.4 Page d'ajouter étudiant :

Dans cette interface l'admin doit remplir le formulaire qui convient pour ajouter un étudiant.

The screenshot shows the 'Ajouter un étudiant' (Add student) form in the SmartClassroom application. On the left is a dark blue sidebar with the 'SmartClassroom' logo and a menu containing: 'Tableau de bord', 'Gérer les étudiants', 'Gérer les enseignants', 'Emplois du temps', 'Présences', and 'Paramètres'. At the bottom of the sidebar is a 'Se déconnecter' button. The main content area is titled 'Ajouter un étudiant' and contains the following fields:

- Nom complet**: A text input field with the placeholder 'Nom complet'.
- Email**: A text input field with the placeholder 'Adresse email'.
- Téléphone**: A text input field with the placeholder 'Numéro de téléphone'.
- Département**: A dropdown menu with 'Sélectionner' as the placeholder.
- Niveau**: A dropdown menu with 'Sélectionner' as the placeholder.
- Groupe**: A dropdown menu with 'Sélectionner' as the placeholder.
- Spécialité**: A dropdown menu with 'Sélectionner' as the placeholder.
- Année académique**: A dropdown menu with 'Sélectionner' as the placeholder.
- Semestre**: A dropdown menu with 'Sélectionner' as the placeholder.
- Modules inscrits (Ctrl+clic pour multi-sélection)**: A list box with the placeholder '-- Sélectionnez les modules --' and the following items: 'Analyse', 'Algèbre', 'Structure Machine', and 'ASD'.

An 'Enregistrer l'étudiant' button is located at the bottom right of the form.

Figure 29 : Formulaire d'ajouter étudiant

3.5.5 Page de gestion d'étudiant :

Dans cette interface l'admin doit gérer les étudiant. Ajouter et Supprimer les informations et l'empreinte d'étudiant.

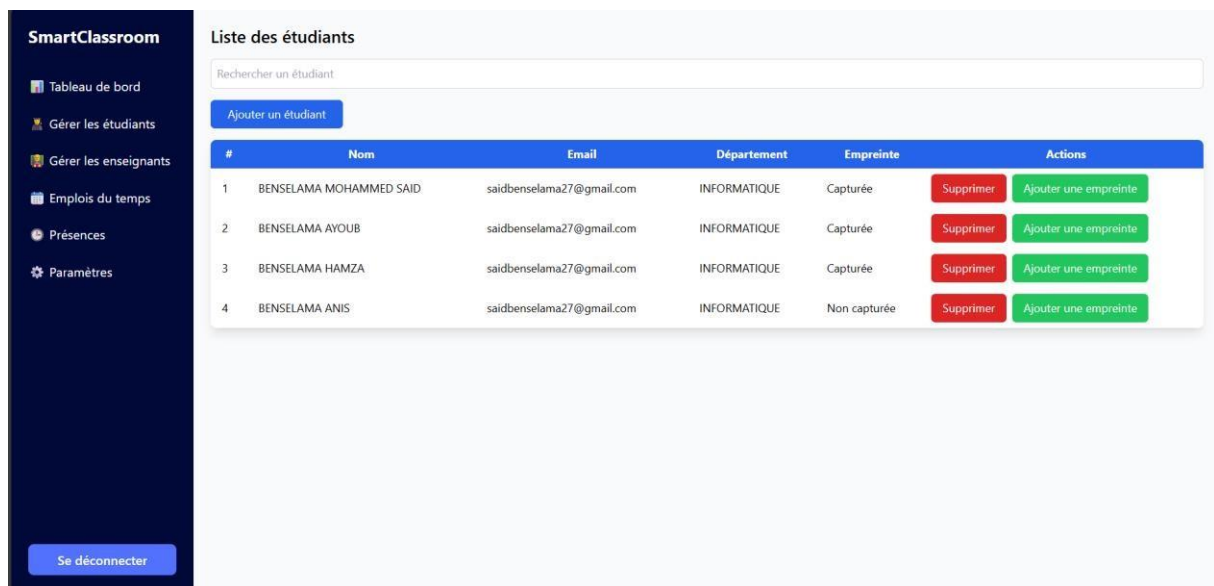


Figure 30 : Gestion d'étudiant

3.5.6 Page d'ajout d'un enseignant:

Dans cette interface l'admin doit remplir le formulaire qui convient pour ajouter et afficher la liste des enseignants.

Ajouter un enseignant

Nom complet:

Email:

Téléphone:

Genre:

Date de naissance:

Spécialisation:

Modules enseignés:

Département:

Date d'embauche:

Classes:

Figure 31 : Formulaire d'ajout d'enseignant

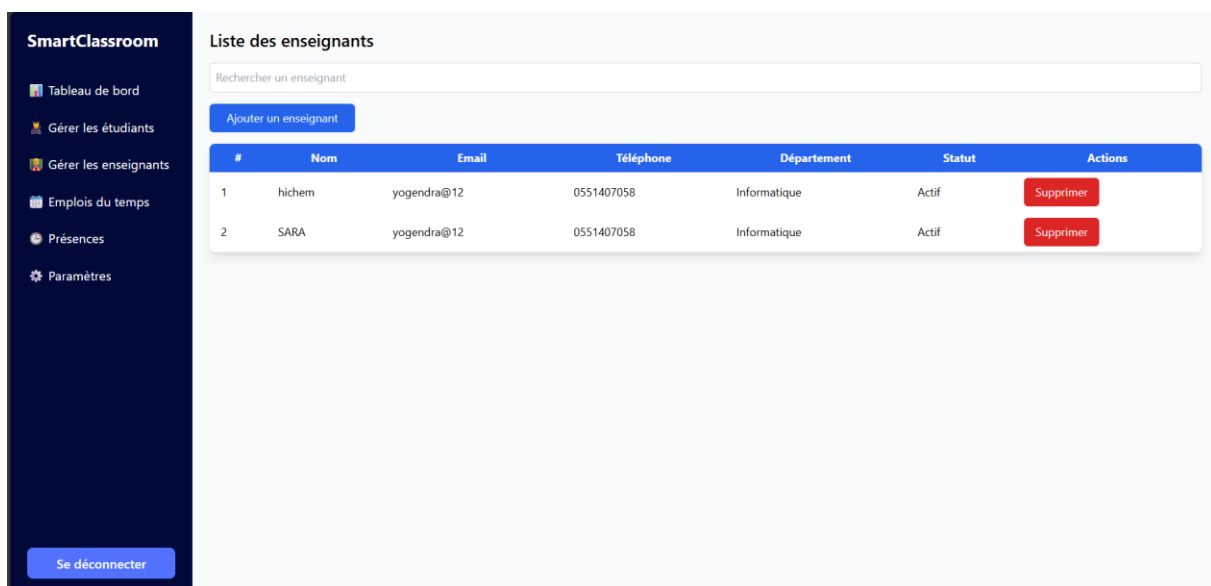


Figure 32 : La liste des enseignants

3.5.7 Page d'ajout d'emploi du temps :

Dans cette interface l'admin doit ajouter l'emploi du temps du semestre et l'afficher.

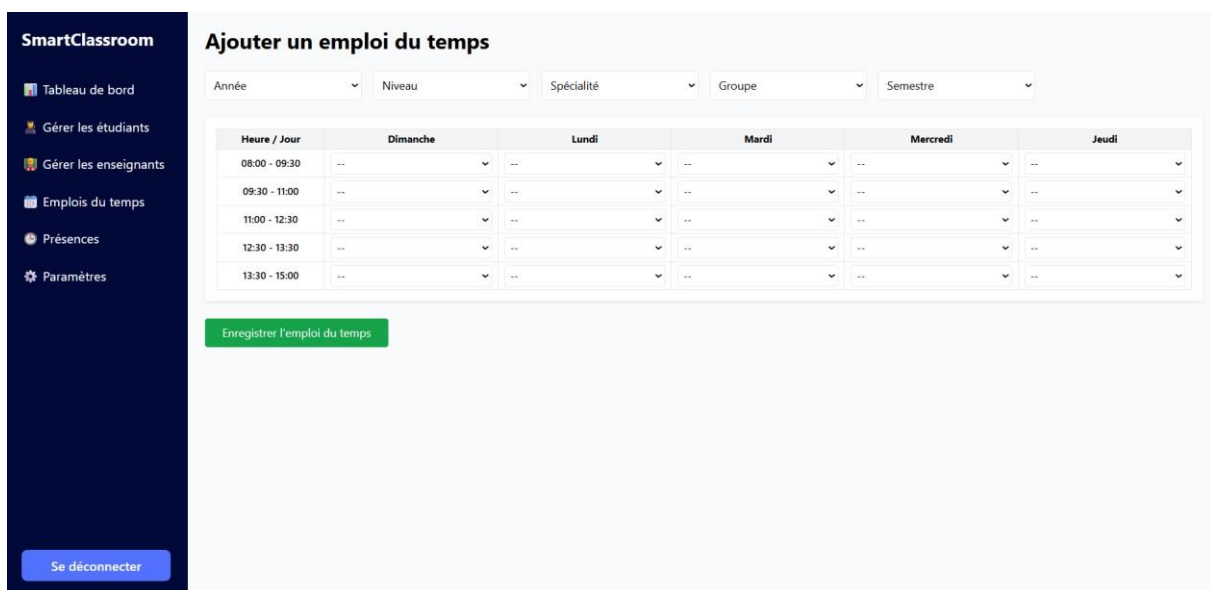


Figure 33 : Formulaire d'ajouter emploi du temps

SmartClassroom

Tableau de bord

Gérer les étudiants

Gérer les enseignants

Emplois du temps

Présences

Paramètres

Se déconnecter

Liste des emplois du temps

Ajouter un emploi du temps

2024/2025

M2

Intelligence Artificielle

G1

S2

Rechercher

M2 Intelligence Artificielle G1 | S2 (2024/2025)

Supprimer

Heure / Jour	Dimanche	Lundi	Mardi	Mercredi	Jeudi
08:00 - 09:30	--	--	--	--	--
09:30 - 11:00	Structure Machine	--	--	--	--
11:00 - 12:30	--	Structure Machine	Anglais	--	--
12:30 - 13:30	--	--	--	Structure Machine	Algèbre
13:30 - 15:00	--	Anglais	--	Algèbre	--

Figure 34 : Emploi du temps

3.5.8 Page de présence :

Dans cette interface on affiche la liste des présences des étudiant avec leur information et la liste par semestre.

SmartClassroom

Tableau de bord

Gérer les étudiants

Gérer les enseignants

Emplois du temps

Présences

Paramètres

Se déconnecter

Liste de présence

Aller à la page des Semestres

Année

Niveau

Spécialité

Groupe

Semestre

Module

mm/dd/yyyy

Rechercher

Nom	Date	Statut	Année	Niveau	Spécialité	Groupe	Semestre	Module
BENSELAMA AYOUB	2025-05-23	Présent	2024/2025	L3	Intelligence Artificielle	G1	S2	Algèbre
BENSELAMA HAMZA	2025-05-25	Présent	2024/2025	L2	Intelligence Artificielle	G1	S1	Algèbre
BENSELAMA MOHAMMED SAID	2025-05-26	Présent	2024/2025	M2	Intelligence Artificielle	G1	S2	Analyse

Figure 35 : Liste de présence

SmartClassroom

Tableau de bord

Gérer les étudiants

Gérer les enseignants

Emplois du temps

Présences

Paramètres

Se déconnecter

Présence par semestre

Retour à la liste des présences

Exporter en PDF

2024/2025

M2

Intelligence Artificielle

G1

S2

Analyse

Rechercher

Réinitialiser les filtres

Nom	Module	Semestre	Total Cours	Présences	Absences	Pourcentage	Année	Niveau	Spécialité	Groupe
BENSELAMA MOHAMMED SAID	Analyse	S2	1	1	0	100.00%	2024/2025	M2	Intelligence Artificielle	G1

Figure 36 : Page de présence par semestre

3.5.9 Page de Rapport :

A la fin l’admin ou l’enseignant peuvent générer un rapport en format PDF pour la présence des étudiants.

Rapport de présence par semestre

Nom	Module	Semestre	Total Cours	Présences	Absences	Pourcentage	Année	Niveau	Spécialité	Groupe
BENSELAMA MOHAMMED SAID	Analyse	S2	1	1	0	100.00%	2024/2025	M2	Intelligence Artificielle	G1

Figure 37: Rapport en PDF

3.6 Conclusion

La phase de réalisation constitue une étape essentielle dans le cycle de vie du projet, traduisant concrètement les besoins fonctionnels et techniques analysés en une solution opérationnelle. Ce chapitre a permis de présenter l'implémentation du système de gestion de présence scolaire « SmartClassroom », à travers une description détaillée des technologies utilisées, de l'architecture de la base de données, ainsi que des interfaces développées, notamment l'intégration du capteur biométrique ZK9500.

L'ensemble des composants de l'application ont été conçus dans le but d'assurer une gestion fiable, sécurisée et intuitive des présences scolaires. Ce chapitre met donc en évidence l'aspect pratique du projet, en illustrant la transition entre la conception théorique et sa concrétisation à travers une plateforme web fonctionnelle.

Conclusion générale et perspectives

Ce travail de fin d'études a porté sur la conception et la réalisation d'un système intelligent de gestion de présence scolaire, reposant sur une plateforme web intégrée à un dispositif biométrique. L'objectif principal de ce projet était de proposer une solution moderne, fiable et sécurisée, permettant de répondre aux problématiques de suivi des présences en milieu scolaire, tout en allégeant les tâches administratives. Tout au long de ce mémoire, nous avons abordé les différentes phases du développement, depuis l'analyse des besoins jusqu'à l'implémentation du système, en passant par la modélisation, le choix des technologies et la conception des interfaces. Le recours au capteur d'empreintes digitales ZK9500, couplé à l'utilisation de MongoDB pour la gestion des données, ainsi que des technologies web modernes, a permis de mettre en place un système performant, ergonomique et adapté aux exigences actuelles. Ce projet nous a permis de renforcer nos compétences en ingénierie logicielle, en base de données NoSQL, en développement web, ainsi qu'en intégration de périphériques matériels. Il ouvre également des perspectives d'amélioration et d'extension, notamment en matière d'analyse statistique, de notifications automatisées ou encore d'intégration mobile. En somme, cette expérience a été enrichissante tant sur le plan technique que professionnel, en mettant en application des concepts concrets pour répondre à un besoin réel et pertinent dans le domaine éducatif.

Parmi les perspectives d'évolution possibles, on peut envisager :

- Le développement d'une application mobile pour un accès plus pratique.
- L'utilisation d'autres méthodes d'identification comme la carte RFID ou la reconnaissance faciale.
- L'extension du système à plusieurs établissements scolaires.
- L'intégration d'un système de notifications par e-mail ou SMS pour informer les parents en cas d'absence.

Bibliographie

- [1] <https://www.lucidchart.com/pages/fr/langage-uml>
 - [2] <https://www.lucidchart.com/pages/fr/diagramme-de-sequence-uml>
 - [3] <https://www.lucidchart.com/pages/fr/diagramme-de-classes-uml>
 - [4] <https://www.lucidchart.com/pages/fr/diagramme-de-deploiement-uml>
 - [5] Kabir, M. H., Roy, S., Ahmed, M. T., & Alam, M. (2021). Smart attendance and leave management system using fingerprint recognition for students and employees in academic institute. International Journal of Scientific & Technology Research, 10(6), 268-276.
 - [6] <https://www.elwanjenigroup.com/lecteur-usb/241-zk9500-zkteco.html>
 - [7] <https://azure.microsoft.com/fr-ca/resources/cloud-computing-dictionary/what-is-java-spring-boot>
 - [8] <https://www.mongodb.com/docs/manual/introduction/>
 - [9] <https://react.dev/>
 - [10] <https://code.visualstudio.com/>
 - [11] <https://www.jetbrains.com/idea/download/?section=windows>
 - [12] <https://www.postman.com/>
 - [13] <https://tailwindcss.com/docs/installation/using-vite>
 - [14] <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>
 - [15] <https://zkteco.eu/downloads/zkfinger-sdk-v100-zk9500-usb-read>
-

جامعة عبد الحميد بن باديس-مستغانم-
كلية العلوم الدقيقة والإعلام الآلي

* ملحق القرار الوزاري رقم 933 المؤرخ في 28 جويلية 2016 الذي يحدد القواعد المتعلقة بالوقاية من السرقة العلمية ومكافحتها.

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

جامعة عبد الحميد بن باديس-مستغانم-
كلية العلوم الدقيقة والإعلام الآلي

تصريح شرفي خاص بالالتزام بقواعد النزاهة العلمية
لإنجاز البحث

أنا الممضي أدناه،

المطلوب(ة): يوليت حنوت رقم التسجيل الجامعي: A201937030575

الحامل لبطاقة التعريف الوطنية رقم: 10002963005 والصادرة بتاريخ: 01-03-2024

عن مستغانم 060002

المسجل ب: كلية العلوم الدقيقة والإعلام الآلي / قسم: الإعلام الآلي

شعبة: / التخصص: النكاح الأصناف في الأشياء الذكية

والمكلف بإنجاز مذكرة ماستر بعنوان:

نظام إدارة المحتوى في الإصبع

أصيح بشرفي أنني ألتزم بمراعاة المعايير العلمية والمنهجية ومعايير الأخلاقيات العلمية والنزاهة الأكاديمية
المطلوبة في إنجاز البحث ، وأتحمل المسؤولية الشخصية عن كل المحتوى المتضمن في البحث المذكور أعلاه .

التاريخ: 02/07/2024

إمضاء المعني



* ملحق القرار الوزاري رقم 933 المؤرخ في 28 جويلية 2016 الذي يحدد القواعد المتعلقة بالوقاية من السرقة العلمية ومكافحتها.