

RÉPUBLIQUE ALGERIENNE DÉMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي و البحث العلمي
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
جامعة عمار ثليجي بالاغواط
UNIVERSITÉ AMAR TELIDJI DE LAGHOUAT
كلية العلوم
FACULTÉ DES SCIENCES
DÉPARTEMENT D'INFORMATIQUE

MÉMOIRE DE MASTER

Domaine : Mathématiques et Informatique

Filière : Informatique

Option : Réseaux, Systèmes et Applications Distribuées

Présenté par : Rayane Dalia Yasmine

THÈME

**CONCEPTION ET RÉALISATION D'UN SAAS INNOVANT
POUR LA GESTION DE L'ASSIDUITÉ DES ÉLÈVES BASÉ SUR DES
RÉSEAUX DE NEURONES RNN ET UNE DÉTECTION DE
PRÉSENCE IoT**

Membres du Jury

Dr. SAHLAOUI Atika	M.C.(A)	(Université de Laghouat)	Présidente
Dr. BENKOUIDER Sara	M.C.(A)	(Université de Laghouat)	Examineur
Dr. ABEDLHAFIDI Zohra	M.C.(A)	(Université de Laghouat)	Examineur
Dr. HAMINI Narjes	M.C.(A)	(Université de Laghouat)	Encadreur

NAnnée Universitaire 2022/2023

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Dédicaces

Je commence par remercier Allah pour m'avoir donné la force et la santé de mener à bien ce travail.

Je dédie ce travail :

À mes chers parents, ma mère **Khadija** et mon père **Tahar** pour leur amour, leur soutien et leurs encouragements infailibles tout au long de ma vie.

À mes sœurs : Fatima zohra et Amina.

À mes frères : Ali et Hadjaissa.

À mes cousines : Nesrine et Nada.

Un grand merci à mon encadrant, **Dr Hamini Nardjes** pour son aide inestimable et son contribution significative à la réalisation de ce travail.

Enfin, j'aimerais exprimer ma gratitude à mes amis proches et à tous ceux qui m'ont aidé directement ou indirectement dans la réalisation de ce projet.

Rayane Dalia Yassmine

Remerciements

Je tiens, tout d'abord, à remercier Dieu pour toutes les bénédictions et pour m'avoir donné la patience et la détermination qui m'ont permis d'accomplir ce modeste travail.

Ce travail n'aurait pas été possible sans mon encadrente, Dr. Nardjes Hamini, professeur au Département d'Informatique de la Faculté des Sciences de l'Université de Laghouat, pour ses précieux commentaires et surtout pour sa gentillesse. Je remercie, également, les membres du jury pour avoir accepté d'évaluer mon travail.

Je souhaite également exprimer ma gratitude à chaque personne qui m'a aidée et conseillée tout au long de ce voyage difficile mais magnifique. Je n'aurais pas pu entreprendre cela sans leur aide et leur soutien.

Enfin, je tiens à exprimer ma profonde appréciation envers tous mes professeurs du Département d'Informatique de l'Université de Laghouat qui ont contribué de manière sincère à mon éducation et à ma formation.

ملخص

تعتمد تطور الأمم على الاهتمام الذي يوليه المسؤولون للتعليم والتدريس. ومع ذلك، يمكن أن تنجح سياسة التعليم الوطنية فقط إذا قمنا بإدارة حضور الطلاب والانتظام في المؤسسات التعليمية بفعالية. هذا هو ما دفع بأبحاثنا، حيث اخترنا استكشاف آفاق تحسين إدارة حضور الطلاب من خلال نهج مبتكر يجمع بين تنفيذ تطبيق بوصفه خدمة (SAAS) وذكاء اصطناعي مستند إلى الشبكات العصبية المتكررة (RNN) مع الاهتمام وكذلك التكامل مع تقنية الكشف عن الحضور (IOT) لتحقيق ذلك، قمنا بتصميم وتطوير تطبيق قوي ومرن مخصص لإدارة حضور الطلاب باستخدام أدوات حديث مثل (UML) و (Angular)، (Node)، (MySQL). يتيح هذا التطبيق جمع البيانات وتخزينها وإدارتها بكفاءة. بالإضافة إلى ذلك، قمنا بتطوير جهاز كشف الحضور الذي يعتمد على بصمات الأصابع باستخدام تقنية (ايوة). ولضمان الوصول الواسع وإدارة موارد فعالة، قمنا بنشر تطبيقنا كخدمة (SAAS) باستخدام تقنية Docker. تمثل الإنجاز الرئيسي لأبحاثنا تنفيذ نموذج معتمد على الشبكات العصبية المتكررة (RNN) يتنبأ بمعدلات الغياب لدى الطلاب من خلال تحديد الأسابيع الحرجة. وهذا يوفر للإداريين رؤى قيمة لإدارة الحضور بشكل استباقي

الكلمات الرئيسية : الذكاء الاصطناعي، الطلاب، المدارس، تطوير الويب، SAAS ، الشبكات العصبية الذكية، Docker ، Angular ، التطوير.

Résumé

Le développement d'une nation est tributaire de l'intérêt que portent les responsables à l'éducation et à l'enseignement. Or, une politique d'éducation nationale ne peut réussir que si nous maîtrisons l'assiduité et la ponctualité dans les établissements scolaires. Et c'est, justement, ce qui nous a motivé pour ce travail de recherche dans lequel, nous avons essayé d'explorer de manière approfondie les perspectives d'une optimisation de la gestion de l'assiduité des élèves grâce à une approche novatrice combinant la mise en œuvre d'une application en tant que service (SAAS) et une intelligence artificielle basée sur les réseaux de neurones RNN avec attention et sur la détection de présence intégrant (IoT). Pour ce faire, nous avons conçu et développé une application robuste et flexible dédiée à la gestion de l'assiduité des apprenants, fondée sur des outils modernes tels que l'UML, Node, Angular et MySQL. Cette application permet de collecter, de stocker et de gérer efficacement les données. Nous avons, également, développé un appareil de détection de présence par empreinte digitale utilisant l'IoT. Et pour garantir une accessibilité étendue et une gestion efficace des ressources, nous avons déployé notre application en tant que service SAAS grâce à Docker. Enfin, la réalisation essentielle de notre travail fut l'implémentation d'un modèle basé sur les réseaux neurones récurrents (RNN) qui prédit le taux d'absentéisme des élèves en identifiant les semaines critiques, fournissant, ainsi, aux administrateurs des établissements des informations importantes pour une gestion proactive de l'assiduité.

Mots Clés : Développement web, Elèves, Développement, Angular, SAAS, internet des objets, Réseaux de neurones artificiels, Intelligence artificielle.

Abstract

The development of a nation is contingent upon the interest shown by authorities in education and teaching. However, a national education policy can only succeed if we effectively manage student attendance and punctuality in educational institutions. This motivation drove our research, where we delved deep into the prospects of optimizing student attendance management through an innovative approach that combines the implementation of a Software as a Service (SAAS) application with artificial intelligence based on Recurrent Neural Networks (RNN) with attention, and integrated presence detection (IoT). To achieve this, we designed and developed a robust and flexible application dedicated to student attendance management, using modern tools such as UML, Node, Angular and MySQL. This application allows for the efficient collection, storage, and management of data. Additionally, we developed a fingerprint-based presence detection device using (IoT) technology. To ensure widespread accessibility and efficient resource management, we deployed our application as a SAAS service using Docker. The central achievement of our work was the implementation of an (RNN) based model that predicts student absenteeism rates by identifying critical weeks. This provides administrators with valuable insights for proactive attendance management.

keywords : Developement, Web developement, Students, Angular, SAAS, Internet of things , Artificial neural networks, Artificial intelligence.

Table des matières

Liste des Figures

Liste des Tableaux

Liste d'abréviations

1	Introduction générale	1
1	Contexte	1
2	Problématique et objectifs	1
3	Contenu des chapitres	2
2	État de l'art	3
2.1	Introduction	3
2.2	Le SAAS	3
2.2.1	A propos le cloud computing	3
2.2.2	Définition de SAAS	4
2.2.3	Architecture et fonctionnement du SaaS	5
2.2.4	Intégration et interopérabilité avec d'autres systèmes SAAS	6
2.2.5	Sécurité et confidentialité dans les solutions SaaS	6
2.2.6	Les fournisseurs de SAAS	7
2.3	Les réseaux de neurones artificiels	7
2.3.1	Noeud d'un réseau de neurones	7
2.3.2	Structure des réseaux de neurones	8
2.3.3	La définition des réseaux de neurones artificiels	8
2.3.4	Les domaines d'application des réseaux de neurones artificiels	8
2.3.5	Apprentissage des réseaux de neurones	9
2.3.6	Les types des réseaux de neurones artificiels	9
2.3.7	Les Avancées récentes des réseaux de neurones	10
2.4	Internet des Objets (IoT)	10
2.4.1	Définition	10
2.4.2	Fonctionnement de l'IoT	10
2.4.3	Les avantages de l'IoT	11
2.4.4	Les solutions de l'IoT sur le marché	11
2.5	conclusion	11

3	Conception, réalisation et le déploiement de l'Application SaaS	12
3.1	Introduction	12
3.2	Conception de l'application	12
3.2.1	Spécification des Exigences	12
3.2.2	Phase d'Analyse et de Conception	13
3.2.3	La conception de logiciel	14
3.3	Développement de l'application	14
3.3.1	Environnement et Outils de Développement	14
3.3.2	Les mesures de sécurité pour l'application	16
3.4	Déploiement de l'application SAAS	17
3.4.1	Objectifs de déploiement	17
3.4.2	Description de l'Architecture de l'application SAAS	17
3.4.3	Comparaison entre infrastructure basé sur (VM) et une infrastructure basée sur des conteneurs	18
3.4.4	Présentation le déploiement d'une application SaaS sur Google Cloud Platform (GCP)	19
3.4.5	Déploiement avec Docker	22
3.4.6	Comparaison entre la réalisation de l'auto-hébergement SaaS avec Docker et l'hébergement sur Google Cloud pour le SaaS	24
3.5	Conclusion	25
4	Réalisation de l'Appareil de Détection de Présence par l'Empreinte	26
4.1	Introduction	26
4.2	Description de l'appareil	26
4.2.1	Présentation les composants de l'appareil	26
4.3	La conception matérielle et logicielle de l'appareil	28
4.4	Fonctionnement de l'Appareil de Détection de Présence	31
4.5	Résultats des tests	32
4.6	La figure de l'appareil	33
4.7	La sécurité et la protection des données biométriques	33
4.8	conclusion	33
5	Implémentation des reseaux neurones artificiel RNN	34
5.1	Introduction	34
5.2	Objetifs de travail	34
5.3	Présentations de quelques travaux antérieur	34
5.3.1	Article1 : Predicting and Analyzing Student Absenteeism Using Machine Learning Algorithm	34
5.3.2	Article2 : Factors Affecting Pupils' Absenteeism at Felicormfort Junior High School (JHS) in Cape Coast, Ghana	35
5.3.3	Article3 : Revealing underlying factors of absenteeism : A machine learning approach	35
5.4	Architecture du Modèle de Réseau de Neurones	35
5.5	Implémentation d'un modèle de prédiction basé sur RNN	36
5.5.1	Discussion des résultats	39
5.6	Conclusion	41

6 Conclusion générale	42
6.1 Conclusion	42
6.2 Futures améliorations	43
Bibliographie	44
Annexe : Détail de la conception et la réalisation de l'application	47

Table des figures

2.1	Différence entre les couches SAAS,PAAS et IASS	4
2.2	Différence entre les modèles SAAS,PAAS et IASS	4
2.3	Structure générale de réseau de neurones	8
2.4	Fonctionnement de l'Internet des Objets	10
3.1	Interface de l'admin console	20
3.2	Diagramme de l'architecture de l'application SAAS	21
3.3	Script de Dockerfile	23
3.4	La commande de docker build	23
3.5	La commande de docker run	23
3.6	La commande pour docker login	23
3.7	La commande pour push une image	23
3.8	Docker Hub	24
3.9	La commande pour pull une image	24
3.10	La page Home après le déploiement	24
4.1	Capteur D'empreinte Digitale DY50	26
4.2	Module WIFI ESP8266-E12 NodeMCU V2	26
4.3	Brochage du capteur	27
4.4	Câblage des composants de l'appareil	29
4.5	Script d'enregistrement d'Empreintes Digitales	30
4.6	Script de reconnaissance d'Empreintes Digitales	31
4.7	Interface d'enregistrer l'Empreinte	32
4.8	Résultat d'enregistrement d'Empreinte Digitale	32
4.9	Résultat de reconnaissance pour l'Empreinte Digitale	32
4.10	L'appareil de l'empreinte	33
5.1	Bibliothèques importées dans script de prédiction	36
5.2	Code de collecte de data	36
5.3	Code de définir la longueur de séquence	37
5.4	Code de prétraitement des données	37
5.5	Code de préparation des séquences et des labels	37
5.6	Code de division des données	38
5.7	Code de création du modèle LSTM	38
5.8	Code de compilation du modèle	38
5.9	Code de mise en œuvre de l'arrêt anticipé	38
5.10	Code de l'entraînement du modèle	39

5.11	Code de l'évaluation du modèle	39
5.12	Code de l'utilisation du modèle pour les prédictions	39
5.13	Le résultat de prédiction	39
5.14	Graphique en secteur de prédiction	40
5.15	Graphique en courbes de l'apprentissage	40
5.16	Graphique en courbes de nombres de neurones	41
6.1	Diagramme de classes	47
6.2	Diagramme de cas d'utilisation	48
6.3	Diagramme de séquence login	48
6.4	Diagramme de séquence ajouter étudiant	49
6.5	Diagramme de séquence modifier étudiant	49
6.6	Diagramme de séquence supprimer étudiant	50
6.7	Diagramme de séquence ajouter enseignant	50
6.8	Diagramme de séquence modifier enseignant	51
6.9	Diagramme de séquence supprimer enseignant	51
6.10	Diagramme de séquence recherche étudiant	52
6.11	Diagramme de séquence d'assiduité dans classe	52
6.12	Diagramme de séquence recherche de présence	53
6.13	Diagramme de séquence de marker la présence	53
6.14	Diagramme de séquence de voir la présence	54
6.15	Page de connexion	54
6.16	Page d'ajouter l'élève	55
6.17	Page de modifier l'élève	56
6.18	Page de voir les informations de l'élève	56
6.19	Page de rechercher les élèves par classe	57
6.20	Page d'ajouter enseignant	57
6.21	Page d'enseignant	58
6.22	Page d'ajouter les classes	58
6.23	Page de rechercher la présence des élèves dans la classe	59
6.24	Page de voir la présence de l'élève dans l'école	59
6.25	Page de l'enseignant marque la présence de sa classe	60
6.26	Page de voir l'assiduité de les enfants par leur parent	60

Liste des tableaux

4.1 Câblage de capteur avec l'Arduino	27
---	----

Liste d'Abréviations

Acronyme	Signification
SQL	Structured Query Language
JWT	JSON Web Tokens
UML	Unified Modeling Language
API	Interfaces de Programmation Applicative
REST	Representational State Transfer
DOI	Digital Object Identifier
SAAS	Software As a Service
PAAS	Platform as a Service
IAAS	Infrastructure as a Service
IOT	Internet Of Things
TTL	Time to Live
CRM	Gestion de la Relation Client
GRH	Gestion des Ressources Humaines
VM	Virtual Machine
GCP	Google Cloud Platform
GKE	Google Kubernetes Engine
IAP	Identity-Aware Proxy

IAM	Identity and Access Management
VPC	Virtual Private Cloud
RNN	Recurrent Neural Network
MLP	Multilayer Perceptron
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
ROC	Receiver Operating Characteristic

Introduction générale

1 Contexte

Dans le domaine de l'éducation, l'assiduité des élèves est un aspect essentiel dans leur parcours éducatif. La présence régulière en classe favorise l'engagement et la participation des étudiants, ce qui facilite grandement leur apprentissage. Cependant, la gestion manuelle, souvent sujette à des erreurs humaines, ne suffit plus à répondre aux défis contemporains de l'éducation.

Aujourd'hui, le développement technologique offre des possibilités extraordinaires, il est impératif de réinventer la manière dont nous suivons l'assiduité des élèves. De plus, la présence a une importance capitale pour les parents qui souhaitent savoir si leurs enfants sont présents ou non.

2 Problématique et objectifs

La gestion traditionnelle, manuelle de l'absentéisme présente plusieurs problèmes et défis. Les systèmes de gestion manuelle sont sujets à des erreurs humaines lors de la prise de présence, de la tenue des registres, ou du calcul des taux d'absentéisme. Peut être aussi chronophage pour les enseignants et le personnel administratif, qui doivent consacrer du temps à des tâches répétitives et administratives, et ne permet pas un suivi en temps réel de la présence des élèves car les processus manuels peuvent être inefficaces, en particulier dans les grandes écoles ou les établissements avec un grand nombre d'élèves. Ajoutons à cela, le manque de transparence et le retard dans la communication pour informer les parents des absences de leurs enfants car le processus manuel est difficile. D'autre part, la gestion de données sur l'absentéisme peut être difficile; elle limite la possibilité d'identifier des tendances ou des modèles. Cependant, Les méthodes traditionnelles ne sont souvent pas adaptées aux besoins actuels des établissements scolaires, qui nécessitent des solutions plus flexibles et informatisées. A cause de ces problèmes, on a mis en place une solution technologique innovante et nécessaire pour améliorer l'éducation et simplifier le suivi des élèves.

L'objectif de cette thèse est de concevoir et développer une application de gestion de suivi de l'assiduité des élèves en utilisant MySQL, Node.js et Angular, réputé pour ses performances, sa robustesse et sa flexibilité du système. Intégrer également, des capteurs d'empreintes digitales dans le processus de prise de présence pour automatiser et simplifier la gestion de l'assiduité, et déployer cette application en tant que Service as a Service (SAAS) en utilisant la technologie Docker. De plus, mettre en place un modèle basé sur les Réseaux de Neurones Récurrents (RNN) avec mécanisme d'attention pour anticiper l'absentéisme des élèves et offrir une perspective proactive aux écoles. Enfin, Fournir des recommandations pour l'avenir de la gestion de l'assiduité des élèves en intégrant davantage de technologies émergentes et en poursuivant la recherche en vue d'une amélioration continue de l'éducation.

3 Contenu des chapitres

- **Chapitre 1** : Contexte, problématique et objectifs.
- **Chapitre 2** : Introduction et exploration en détail les concepts théoriques des technologies : SaaS, de l'IA basée sur les réseaux de neurones et de l'IoT.
- **Chapitre 3** : Conception de l'application de gestion de l'assiduité des élèves en utilisant UML comme outil de modélisation pour concevoir les diagrammes nécessaire.
- **Chapitre 4** : Présentation en profondeur le processus de développement et de résultat final de l'application en utilisant des technologies telles qu'Angular, Node.js et MySQL.
- **Chapitre 5** : Réalisation de l'appareil de détection de présence par empreinte digitale, mettant en lumière l'intégration de l'IoT dans notre solution.
- **Chapitre 6** : Présentation de les deux approches de déploiement google cloud platform et Docker, et les étapes de déploiement de l'application en tant que service SaaS à l'aide de Docker.
- **Chapitre 7** : Ce dernier chapitre se concentre sur l'implémentation d'un modèle basé sur les réseaux neuronaux RNN avec attention pour prédire le taux d'absentéisme des élèves en utilisant Python.
- **Conclusion générale**

CHAPITRE 2

ÉTAT DE L'ART

2.1 Introduction

Le Cloud Computing, l'Intelligence Artificielle (IA) basée sur les réseaux de neurones, et l'Internet des Objets (IoT) sont trois technologies majeures qui ont considérablement façonné notre monde moderne. Ensemble, elles ont ouvert la voie à de nouvelles opportunités et ont impacté divers secteurs, de l'industrie à la santé, en passant par l'éducation. Et pour cette raison on a choisi de les appliquer dans notre logiciel qui va gérer l'assiduité des élèves dans l'établissements scolaires. Dans ce cadre Nous apprendrons à connaître ces technologies.

2.2 Le SAAS

2.2.1 A propos le cloud computing

CISCO: « Le Cloud Computing est une plateforme de mutualisation informatique fournissant aux entreprises des services à la demande avec l'illusion d'une infinité des ressources »

Le Cloud Computing est un concept qui consiste à déporter sur des serveurs distants des stockages et des traitements informatiques traditionnellement localisés sur des serveurs locaux ou sur le poste de l'utilisateur. Il consiste à proposer des services informatiques sous forme de service à la demande, accessible de n'importe où, n'importe quand et par n'importe qui. L'idée principale à retenir est que le Cloud n'est pas un ensemble de technologies, mais un modèle de fourniture, de gestion et de consommation de services et de ressources informatiques localisé dans des Datacenter [1].

Le cloud computing peut être décomposé en trois couches :

- Application (SaaS, Software as a Service)
- Platform (PaaS, Platform as a Service)
- Infrastructure (IaaS, Infrastructure as a Service)

La Figure ci-dessous représente les différentes couches du cloud computing de la couche la moins visible pour les utilisateurs finaux à la plus visible. L'infrastructure as a Service (IaaS) est plutôt gérée par les architectes réseaux, la couche PaaS est destinée aux développeurs d'applications et finalement le logiciel comme un service (SaaS) est le « produit final » pour les utilisateurs.

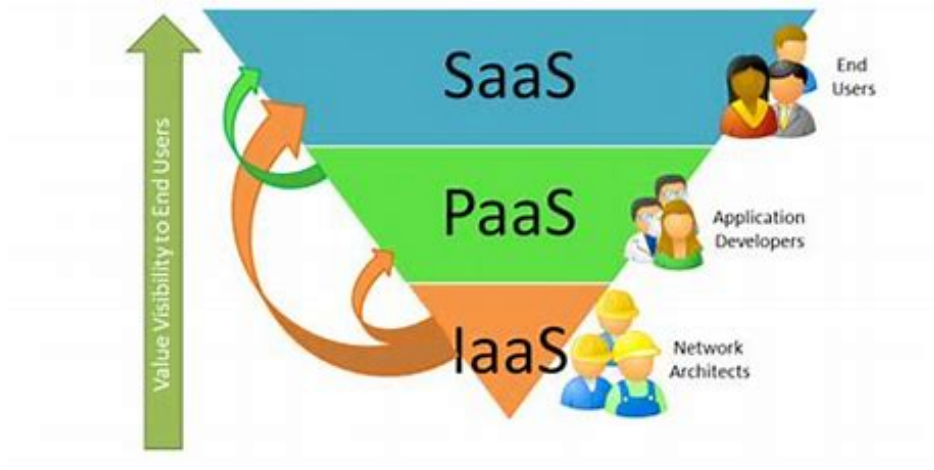


FIGURE 2.1 – Différence entre les couches SAAS,PAAS et IAAS

La Figure ci-dessous représente les différents modèles et la manière d'utilisation pour créer un environnement de cloud computing qui répond aux besoins des clients [34].

On-site	IaaS	PaaS	SaaS
Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
O/S	O/S	O/S	O/S
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking

■ You manage
■ Service provider manages

FIGURE 2.2 – Différence entre les modèles SAAS,PAAS et IAAS

2.2.2 Définition de SAAS

C'est un modèle de distribution de logiciels qui offre aux utilisateurs un accès à des applications via Internet. Contrairement aux méthodes traditionnelles où les

logiciels étaient installés localement sur les ordinateurs, le SAAS permet aux utilisateurs d'utiliser les applications directement depuis le cloud, sans avoir besoin de les installer ou de les gérer localement.

Dans ce modèle le fournisseur de services gère l'ensemble de l'infrastructure, de la maintenance et des mises à jour du logiciel. Les utilisateurs peuvent accéder aux applications via un navigateur web ou une application spécifique, en fonction de leurs besoins, sans se soucier des problèmes de compatibilité ou de la gestion des ressources matérielles.

2.2.3 Architecture et fonctionnement du SaaS

L'architecture et le fonctionnement du modèle SaaS (Software as a Service) reposent sur la fourniture de logiciels via le cloud, accessibles aux utilisateurs via Internet.

- **Architecture de modèle SAAS :**

1. **Infrastructure Cloud :** Les fournisseurs de modèle SaaS utilisent des infrastructures cloud robustes pour héberger et gérer les applications. Ces infrastructures comprennent des serveurs, des bases de données, des systèmes de stockage et des réseaux.
2. **Multitenant :** Cette architecture est souvent multitenant, c'est à dire une seule instance de l'application est partagée par plusieurs clients (locataires). Chaque client dispose d'un espace dédié, isolé et sécurisé dans l'application.
3. **Mises à jour centralisées :** Tous les clients bénéficient des mises à jour automatiques, ce qui garantit que les dernières fonctionnalités et correctifs sont disponibles pour tous.

- **Fonctionnement du SaaS :**

1. **Accès via Internet :** Les applications SaaS sont accessibles de n'importe où, ce qui élimine la nécessité d'installer et de maintenir les logiciels localement.
2. **Évolutivité :** Les solutions SaaS sont conçues pour être évolutives. Adapter les ressources de les utilisateurs par rapport le besoin et la demande.
3. **Sécurité :** Des mesures de sécurité sont conçues par les fournisseurs afin de protéger les données des utilisateurs, y compris le chiffrement des données, les contrôles d'accès et les certifications de sécurité.
4. **Économies d'Échelle :** Les fournisseurs de SaaS peuvent bénéficier des économies d'échelle en gérant une seule instance de l'application pour plusieurs clients, ce qui peut réduire les coûts pour chaque client individuel.

En résumé, le modèle SaaS offre des avantages tels que **la flexibilité, l'accessibilité, la réduction des coûts initiaux et la délégation de la maintenance**. Il est largement utilisé dans divers domaines tels que la gestion de la relation client (**CRM**), la gestion des ressources humaines (**GRH**), la gestion de projet et la productivité en entreprise.

2.2.4 Intégration et interopérabilité avec d'autres systèmes SAAS

L'intégration et l'interopérabilité avec d'autres systèmes SaaS sont des aspects essentiels pour permettre aux applications de collaborer de manière transparente.

- **Intégration des Systèmes SaaS**

1. **API (Interfaces de Programmation Applicative)** : Les API qui sont proposés par la plupart des fournisseurs de SAAS permettant aux développeurs d'intégrer leurs applications avec d'autres systèmes. Ces API définissent les méthodes et les protocoles pour l'échange de données et la communication entre applications.
2. **Middleware et Solutions d'Intégration** : Les plateformes de middleware permettent de connecter plusieurs systèmes SaaS en utilisant des flux de travail automatisés. Ils peuvent simplifier la création de connexions entre différentes applications.
3. **Connecteurs Prédéfinis** : Certains fournisseurs SaaS proposent des connecteurs prédéfinis pour les applications courantes. Ces connecteurs simplifient l'intégration et réduisent le besoin de développement personnalisé.

- **Interopérabilité entre Systèmes SaaS**

1. **Formats de Données Standard** : Pour échanger des informations entre systèmes par l'utilisation des formats de données standard comme JSON (JavaScript Object Notation) ou XML (eXtensible Markup Language) afin de faciliter la compréhension des données par différentes applications.
2. **Normes et Protocoles** : Respecter les normes et protocoles reconnus pour la communication et l'échange de données. Par exemple, OAuth pour l'authentification et REST (Representational State Transfer) pour les API.
3. **Synchronisation en Temps Réel** : Pour maintenir les données à jour entre différents systèmes.
4. **Sécurité et Confidentialité** : Les échanges de données entre systèmes sont sécurisés et conformes aux réglementations en matière de protection des données.

L'intégration et l'interopérabilité efficaces entre systèmes SaaS qui permettent de faciliter l'automatisation des flux de travail, améliorer l'efficacité opérationnelle et optimiser les processus métier.

2.2.5 Sécurité et confidentialité dans les solutions SaaS

La sécurité et la confidentialité sont essentielles dans les solutions SaaS (Software as a Service) en raison de la nature des données sensibles et de l'accès à distance aux applications via Internet. La sécurité et la confidentialité sont abordées dans les solutions SaaS par :

- **Chiffrement des Données** : Lorsque les données sont stockées et en transit entre l'utilisateur et le fournisseur SaaS elles sont souvent chiffrées. Ce chiffrement garantit que seules les parties autorisées peuvent accéder aux données.
- **Authentification et Autorisation** : Les mécanismes d'authentification et d'autorisation afin d'assurer que seules les personnes autorisées peuvent accéder aux applications et aux données.
- **Sécurité au Niveau de l'Application** : Les fournisseurs SaaS mettent des fonctionnalités de sécurité directement dans les applications pour protéger les données et les interactions utilisateur.
- **Sauvegardes Régulières** : Les fournisseurs SaaS effectuent régulièrement des sauvegardes des données pour assurer la disponibilité en cas de panne ou de perte de données.
- **Surveillance et Détection des Intrusions** : Ils mettent Des systèmes de surveillance en temps réel et de détection des intrusions afin d'identifier rapidement les activités suspectes.

2.2.6 Les fournisseurs de SAAS

Pour répondre aux besoins des entreprise et des particulier, il exist multitude de fournisseurs dans plusieurs domaines. Voici quelques exemples de fournisseurs SAAS populaires.

- **Stockage et Partage de Fichiers** : Dropbox et Box parmi les services populaire de stockage et partage de fichier en ligne.
- **Productivité et Collaboration** : Il exist plusieurs fournisseurs de SAAS dans ce service par exemple : Microsoft, Google, Slack...etc.
- **CRM (Gestion de la Relation Client)** : Salesforce l'un des principaux fournisseurs dans le service de gestion de ventes et du marketing.
- **Services Financiers** : Inuit pour les services de compatibilité de gestion financière pour les petites entreprises et Xero pour les grandes entreprises.

2.3 Les réseaux de neurones artificiels

2.3.1 Noeud d'un réseau de neurones

Chaque fois que nous apprenons quelque chose, notre cerveau stocke les connaissances. L'ordinateur utilise la mémoire pour stocker des informations. Bien qu'ils stockent tous les deux des informations, leurs mécanismes sont très différents. L'ordinateur stocke des informations à des emplacements spécifiés de la mémoire, tandis que le cerveau modifie l'association des neurones. Le neurone lui-même n'a pas de capacité de stockage; il ne fait que transmettre des signaux d'un neurone à l'autre. Le cerveau est un réseau gigantesque de ces neurones et l'association des neurones forme une information spécifique[4].

Les réseaux de neurones tentent d'imiter le cerveau humain, en combinant l'informatique et les statistiques pour résoudre des problèmes courants dans le domaine de l'intelligence artificielle.

2.3.2 Structure des réseaux de neurones

Les réseaux de neurones sont constitués de couches de neurones interconnectés.

- La première couche est la couche d'entrée, qui reçoit les données d'entrée.
- Ensuite, il y a une ou plusieurs couches cachées, qui effectuent des calculs et des transformations non linéaires sur les données.
- Enfin, il y a la couche de sortie, qui fournit les résultats finaux du réseau.

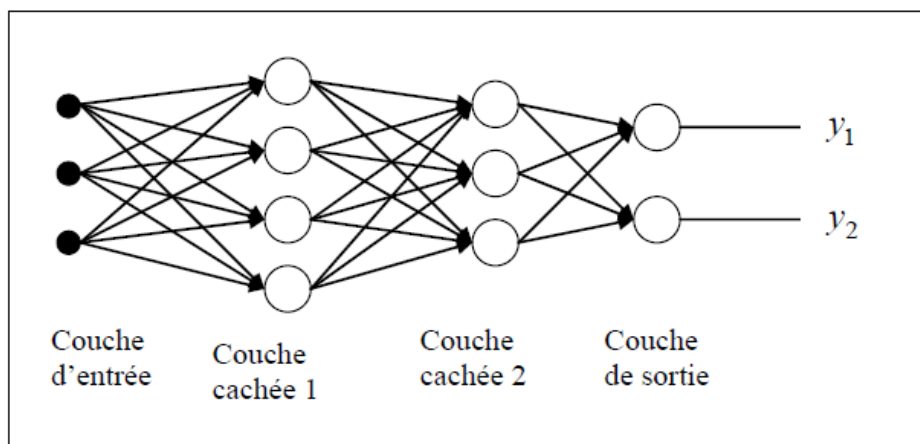


FIGURE 2.3 – Structure générale de réseau de neurones

2.3.3 La définition des réseaux de neurones artificiels

un réseau de neurones artificiels est un ensemble organisé de neurones interconnectés permettant la résolution de problèmes complexes tels que la vision par ordinateur ou le traitement du langage naturel. Tel que les neurones artificiels sont les blocs de construction fondamentaux des réseaux de neurones. Chaque neurone reçoit des entrées pondérées, les passe à travers une fonction d'activation, puis envoie la sortie à d'autres neurones. Les poids des connexions entre les neurones sont ajustés lors de l'apprentissage pour optimiser les performances du réseau.

Un réseau de neurone ne se programme pas, il est entraîné grâce à un mécanisme d'apprentissage. Les tâches particulièrement adaptées au traitement par réseau de neurones sont : l'association, la classification, la discrimination, la prévision ou l'estimation, et la commande de processus complexes[2].

2.3.4 Les domaines d'application des réseaux de neurones artificiels

Aujourd'hui, les réseaux de neurones artificiels ont de nombreuses applications dans des secteurs très variés :

- **Traitement d'images** : reconnaissance de caractères et de signatures, compression d'images, reconnaissance de forme, cryptage, classification, etc.
- **Traitement du signal** : filtrage, classification, identification de source, traitement de la parole. . .etc.

- **Contrôle** : commande de processus, diagnostic, contrôle qualité, asservissement des robots, systèmes de guidage automatique des automobiles et des avions. . .etc.
- **Défense** : guidage des missiles, suivi de cible, reconnaissance du visage, radar, sonar, lidar, compression de données, suppression du bruit. ..etc.
- **Optimisation** : planification, allocation de ressource, gestion et finances, etc.
- **Simulation** : simulation du vol, simulation de boîte noire, prévision météorologique, recopie de modèle. . .etc.

2.3.5 Apprentissage des réseaux de neurones

L'apprentissage de réseaux de neurones, également connu sous le nom d'apprentissage profond ou d'apprentissage automatique profond, est un domaine de l'intelligence artificielle qui consiste à entraîner des modèles de réseaux de neurones artificiels pour effectuer des tâches complexes en utilisant des données. Il existe deux principaux types d'apprentissage : **supervisé** et **non supervisé**. La principale différence entre les deux méthodes réside dans les données.

- **Apprentissage Supervisé :**

L'apprentissage supervisé consiste à entraîner un modèle de réseau de neurones d'utiliser des ensembles de **données étiquetées** pour former un algorithme de machine learning. Ladite supervision tient au fait que les étiquettes sur les données aident le modèle à prédire des résultats avec précision[5].

Les tâches principales d'un algorithme entraîné à l'aide de données étiquetées correspondent généralement à la **classification**(catégorisation d'images, détection de spam ou à la **régression**(prévision de valeurs numériques).

- **Apprentissage Non Supervisé :**

L'apprentissage non supervisé implique l'entraînement d'un modèle de réseau de neurones sur l'utilisation de **données non étiquetées** pour former un modèle(sans avoir besoin de sorties attendues). En d'autres termes, l'algorithme apprend par lui-même à découvrir des informations à partir d'un ensemble de données[5]. Il est surtout utilisé pour le **clustering**(regroupement de données similaires) et l'**association**.

2.3.6 Les types des réseaux de neurones artificiels

Les réseaux de neurones peuvent être classés en différents types, qui sont utilisés à des fins différentes. Voici quelques-uns des principaux types de réseaux de neurones :

- Le perceptron est le plus ancien réseau de neurones, créé par Frank Rosenblatt en 1958.
- Les réseaux de neurones multicouches (**MLP**) : Sont constitués d'une couche en entrée, d'une ou de plusieurs couches cachées et d'une couche en sortie.
- Les réseaux de neurones convolutifs (**CNN**) : sont généralement utilisés pour la reconnaissance d'images, la reconnaissance de formes et/ou la vision par ordinateur. Ces réseaux exploitent les principes de l'algèbre linéaire, en particulier la multiplication des matrices, pour identifier les motifs dans une image.

- réseaux de neurones récurrents (RNN) : Sont identifiés par leurs boucles de rétroaction. Ces algorithmes d'apprentissage sont principalement exploités lorsqu'on utilise des données de séries chronologiques pour faire des prédictions sur des résultats futurs, comme les prédictions boursières ou les prévisions de ventes[3].

2.3.7 Les Avancées récentes des réseaux de neurones

Les réseaux de neurones ont connu de nombreuses avancées récentes. Par exemple, l'utilisation de l'apprentissage profond (deep learning) permet de construire des réseaux de neurones plus profonds et plus complexes. De plus, les réseaux de neurones adversaires génératifs (GAN), ont contribué à des progrès significatifs dans des domaines tels que la génération d'images réalistes ou la traduction automatique.

2.4 Internet des Objets (IoT)

2.4.1 Définition

L'IoT (Internet des objets), c'est l'interconnexion d'objets physiques équipés de logiciels et de capteurs spéciaux avec Internet.

2.4.2 Fonctionnement de l'IoT

L'IoT peut être envisagé comme un ensemble de technologies reliées entre elles dans trois domaines principaux (Figure 2.4) :



FIGURE 2.4 – Fonctionnement de l'Internet des Objets

- **Things** : sont des **éléments physiques disposant de capteurs intégrés** et qui sont connectés à Internet.
- **Insights** : il s'agit des **résultats du traitement et de l'analyse** des données brutes envoyées par les Things.
- **Actions** : il s'agit de la **réponse automatisée ou manuelle** aux informations remontées par l'Insight.

2.4.3 Les avantages de l'IoT

- Une **vision en temps réel** sur tous les processus de production ;
- L'**optimisation** du temps, et donc des dépenses ;
- L'**amélioration** de la productivité ;
- Une **prise de décision** améliorée.

2.4.4 Les solutions de l'IoT sur le marché

Plusieurs solutions existent sur le marché :

- **Ethernet** : dans cette solution, les objets sont connectés physiquement au réseau interne. Elle est très peu utilisée en raison de la complexité de sa mise en place avec un grand nombre d'objets et surtout à cause des restrictions géographiques qu'elle impose.
- **Wifi** : avec une portée plus large (une centaine de mètres), cette solution n'impose aucune connexion physique et facilite le déploiement sur un périmètre plus ou moins étendu.
- **4G/5G** : solution avec une couverture presque mondiale. Les objets peuvent être connectés pratiquement n'importe où dans le monde et communiquer sans problème avec les systèmes centraux.
- **L'Intelligence artificiel et le Machine Learning** : on est maintenant capable de développer des systèmes intelligents proches du raisonnement humain, capables de s'adapter à leur environnement et d'apprendre au fil du temps.
- **L'analyse de données** : il existe sur le marché des systèmes capables de lire, analyser et traiter des millions d'informations en temps réel.
- **Les protocoles de communication** : est l'une des composantes essentielles de l'IoT. Que ce soit entre des objets connectés, avec un système central.

2.5 conclusion

En concluant ce chapitre introductif, nous avons exploré les concepts essentiels du SAAS (Software As Service), de l'Intelligence Artificielle basée sur les réseaux de neurones et de l'Internet des Objets. Cette compréhension approfondie de ces technologies servira de fondement solide pour les chapitres d'implémentation à venir. Le chapitre suivant se concentrera sur le design, en utilisant des méthodologies telles qu'UML (Unified Modeling Language) pour définir l'architecture, les composants et les interactions du logiciel de l'assiduité des élèves dans l'établissements scolaires. Suite à cette étape, nous aborderons les interfaces du logiciel, plongeant dans la manière dont les utilisateurs interagiront avec l'application. Enfin on va implémenter notre compréhension approfondie de ces technologies en une solution concrète et fonctionnelle.

CHAPITRE 3

CONCEPTION, RÉALISATION ET LE DÉPLOIEMENT DE L'APPLICATION SAAS

3.1 Introduction

Ce chapitre est consacré sur la conception, la réalisation et le déploiement de l'application SAAS et ces objectifs, à la présentation de l'architecture. Il sera également, question de déploiement sur google cloud et docker ainsi que la comparaison entre ces deux technologies.

3.2 Conception de l'application

L'objectif de la conception de l'application de Suivi de l'Assiduité des Élèves est de mettre en place un système complet qui gère et surveille efficacement la présence des élèves au établissements scolaires. Ce logiciel vise à rationaliser le processus d'enregistrement de la présence tout en offrant une plateforme permettant aux éducateurs, aux administrateurs et aux parents de collaborer de manière efficace. Au cœur de ce logiciel se trouve la fonction de suivi de la présence, qui permet aux enseignants de marquer efficacement la présence et l'absence des élèves. Cette fonction garantit non seulement des enregistrements de présence précis, mais sert également de base pour la génération de rapports qui permettant d'informer les parents l'état de présence de leurs enfants. De plus, le logiciel propose une méthode de suivi de présence basée sur les empreintes digitales. Cette fonction permet aux élèves de marquer leur présence en scannant leurs empreintes digitales à leur arrivée à l'école. En parallèle, les administrateurs, notamment le directeur, peuvent également utiliser ce logiciel pour obtenir une vue d'ensemble sur la présence des étudiants. Grâce aux fonctionnalités de génération de rapports et d'analyses, le directeur peut avoir une idée générale de la présence des étudiants pour prendre des mesures en conséquence.

3.2.1 Spécification des Exigences

Dans cette partie, afin d'identifier les services que notre logiciel doit fournir, nous détaillerons les exigences spécifiques, qui se composent de besoins fonctionnels et de besoins non fonctionnels.

1. Les besoins fonctionnels

- Enregistrement de la Présence : Le système permet aux enseignants d’enregistrer la présence des élèves en classe.
- Suivi des Absences : Le système permet aux enseignants de suivre et de gérer les absences des élèves, en incluant les durées des absences.
- Génération de Rapports : Le système est capable de générer des rapports d’assiduité pour différentes périodes.
- Intégration des Empreintes Digitales : Le système intègre une fonctionnalité de reconnaissance des empreintes digitales pour permettre aux élèves de marquer leur présence de manière biométrique.
- Interface d’Administration : Le système comporte une interface d’administration pour les administrateurs, permettant la gestion des données d’assiduité.
- Manager les utilisateurs : Le système permet aux administrateur de manager les utilisateurs (les enseignants et les parents).

2. Les besoins non fonctionnels

- Évolutivité : L’application est conçue pour gérer une augmentation progressive du nombre d’utilisateurs, les élèves, de données et d’activités simultanées.
- Sécurité : Afin de protéger les données des anciens élèves, l’application est empêcher l’accès non autorisé et garantir la confidentialité, l’intégrité et la disponibilité de l’application et même stockées les données de manière sécurisée et confidentielle.
- Sécurité Biométrique : La fonctionnalité de reconnaissance des empreintes digitales doit respecter les normes de sécurité biométrique pour protéger les données sensibles des élèves. Performances : L’application doit fonctionner de manière efficace et réagir rapidement aux entrées des utilisateurs, offrant ainsi une expérience conviviale.
- Maintenabilité : Avec un code source modulaire et bien documenté (Angular), l’application est conçue pour faciliter la maintenance, les mises à jour et l’ajout de nouvelles fonctionnalités à l’avenir.

3.2.2 Phase d’Analyse et de Conception

Langage de Modélisation Unifié(UML) Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il fournit un ensemble de notations et de conventions pour décrire la structure, le comportement, les interactions et les caractéristiques d’un système, ce qui en fait un outil essentiel dans le domaine du développement logiciel et de l’ingénierie des systèmes[6].

Les types de diagrammes UML

Les diagrammes UML se divisent principalement en deux catégories principales : les diagrammes de structure et les diagrammes comportementaux.

1. Diagrammes de Structure

- Class Diagram
- Component Diagram
- Deployment Diagram
- Object Diagram

2. Diagrammes Comportementaux

- Diagramme de Cas d'Utilisation
- Diagramme d'Activité
- Diagramme de Séquence
- Diagramme d'État de Machine[7].

3.2.3 La conception de logiciel

Cette section traite de la phase de conception de l'application. Cette conception joue un rôle important dans la définition de la fonctionnalité et de l'expérience utilisateur de l'application.

Pour élaborer les diagrammes de l'application, nous ferons usage de StarUML. StarUML est un logiciel de modélisation UML (Unified Modeling Language) open source, qui nous permet de créer différents types de diagrammes en utilisant le Langage de Modélisation Unifié (UML)[8].

3.3 Développement de l'application

3.3.1 Environnement et Outils de Développement

MySQL

MySQL est un système de gestion de bases de données relationnelles SQL open source développé et supporté par Oracle[11].

PhpMyAdmin

phpMyAdmin est un logiciel libre écrit en PHP qui a pour mission de s'occuper de l'administration d'un serveur de base de données MySQL ou MariaDB. Vous pouvez utiliser phpMyAdmin pour réaliser la plupart des tâches d'administration, ceci incluant la création de base de données, l'exécution de demandes, et l'ajout de comptes utilisateur[12].

Postman

Postman est une plateforme d'API permettant de construire, tester et d'utiliser des APIS. Postman simplifie chaque étape du cycle de vie des API et facilite la collaboration afin que vous puissiez créer de meilleures API plus rapidement[13].

Angular

Angular est un Framework JavaScript open-source écrit en TypeScript, développé par Google qui permet la création d'applications Web et plus particulièrement de ce qu'on appelle des « Single Page Applications, SPAs » en utilisant HTML et TypeScript[14].

HTML

L'HTML (HyperText Markup Language) est un langage de balisage d'hypertexte utilisé pour créer des pages web. Ce langage permet de réaliser de l'hypertexte à base d'une structure de balisage[15].

CSS

Le CSS, est l'acronyme anglais de Cascading Style Sheets qui peut se traduire par "feuilles de style en cascade", est utilisé pour décrire l'apparence et la mise en forme d'un document écrit dans un langage de balisage afin de modifier le style des pages web et des interfaces utilisateur[16].

Typescript

TypeScript est un surensemble de JavaScript qui ajoute un typage statique optionnel et des fonctionnalités avancées à JavaScript. Il a été développé par Microsoft et a été initialement publié en octobre 2012. Depuis sa sortie en 2012, il a rapidement été adopté par la communauté des développeurs web[17].

Bootstrap 5

Bootstrap est un framework développé par l'équipe du réseau social Twitter. Proposé en open source (sous licence MIT), ce framework utilisant les langages HTML, CSS et JavaScript fournit aux développeurs des outils pour créer un site facilement. Ce framework est pensé pour développer des sites avec un design responsive, qui s'adapte à tout type d'écran, et en priorité pour les smartphones[18].

Angular Material

Angular Material est une bibliothèque de composants d'interface utilisateur pour les développeurs Angular JS, développé par Google en 2014[19].

VS code

Visual Studio Code est un éditeur de code simplifié, qui est gratuit et développé en open source par Microsoft. Il fonctionne sous Windows, mac OS et Linux. De plus, il offre une prise en charge intégrée pour JavaScript, TypeScript et Node.js. Il possède un écosystème riche d'extensions pour d'autres langages et environnements d'exécution tels que C++, C, Java, Python et PHP[20].

NodeJs

Node.js est un environnement d'exécution single-thread, open-source et multi-plateforme permettant de créer des applications rapides et évolutives côté serveur et en réseau. Il fonctionne avec le moteur d'exécution JavaScript V8[28].

3.3.2 Les mesures de sécurité pour l'application

L'application de gestion de l'assiduité des élèves dans l'établissements scolaires intègre des mesures de sécurité visant à améliorer la sécurité des données, en mettant l'accent sur l'authentification via JSON Web Tokens (JWT), le hachage des mots de passe et la prévention des attaques par injection SQL.

1. Authentification avec JSON Web Tokens (JWT)

JSON Web Token (JWT) est une norme ouverte (RFC 7519) qui définit une méthode compacte et autonome de transmission sécurisée entre parties d'informations encodées sous forme d'objet JSON[21].

- **Implémentation**

Lorsqu'un utilisateur se connecte avec succès, le serveur crée un JWT contenant les informations de l'utilisateur cryptées ainsi qu'une signature pour garantir son intégrité. Ensuite, le jeton est envoyé au client et stocké pour être inclus dans les demandes ultérieures. À chaque demande, le serveur vérifie l'authenticité du JWT pour accorder l'accès aux ressources protégées et assurer l'authentification.

2. Hachage des Mots de Passe

Le hachage des mots de passe est une fonctionnalité de sécurité essentielle utilisée pour protéger les informations d'identification des utilisateurs stockées dans la base de données du site.

- **Implémentation**

Pour stocker les mots de passe des utilisateurs, l'application utilise un algorithme de hachage standard et robuste. Le mot de passe original n'est jamais conservé, seul le mot de passe haché est conservé dans la base de données.

3. Protection contre les Injections SQL

Injection SQL est une technique d'injection de code utilisée pour modifier en exécutant des requêtes SQL malveillantes ou récupérer des données dans des bases de données SQL[22]. Protéger les données sur le site nécessite de prévenir les injections SQL.

- **Implémentation**

L'application a mis en place des méthodes solides de validation des entrées pour renforcer les mesures de sécurité contre les menaces d'injection SQL en examinant attentivement les entrées utilisateur et en recherchant la présence de caractères potentiellement dangereux, en particulier les guillemets simples (').

3.4 Déploiement de l'application SAAS

3.4.1 Objectifs de déploiement

- Assure le lancement rapide de l'application SAAS sur le marché.
- Assurer la disponibilité et l'accessibilité de l'application pour les utilisateurs.
- Assure que l'application fonctionne avec des performances optimales.
- Assure que l'application est sécurisée contre les menaces et les vulnérabilités de données.
- Assure la satisfaction des clients en répondant rapidement à leurs besoins.

3.4.2 Description de l'Architecture de l'application SAAS

Cette architecture offre une base solide pour développer l'application SAAS sur le cloud. Voici les principaux composants :

1. **Interface Utilisateur (UI) :**

C'est la couche visible de l'application que les utilisateurs peuvent voir, et être accessible par un navigateur web ou application mobile.

2. **Serveurs d'Application :**

Ces serveurs reçoivent les demandes des utilisateurs, traitent les données et fournissent des réponses aux clients.

3. **Bases de Données :**

Les bases de données stockent les données de l'application et permettent la gestion de sauvegarde, disponibilité et la sécurité de données.

4. **Intégration :**

Cette couche facilite la communication entre les différentes parties de l'application (les serveurs d'application, les bases de données et l'interface utilisateur) avec l'intégration des services tiers.

5. **Authentification et Sécurité :**

Ce composant gère l'authentification et l'autorisations des accès pour les utilisateurs.

6. **L'infrastructure :**

Cette couche gère la partie technique de l'application nécessaire pour le fonctionnement de l'application.

7. **Gestion de surveillance et journalisme :**

Cette couche est responsable sur la surveillance des performances, la gestion de données et les mises à jour de l'application.

3.4.3 Comparaison entre infrastructure basé sur (VM) et une infrastructure basée sur des conteneurs

voici la comparaison entre l'utilisation d'une infrastructure basée sur des machines virtuelles (VM) et une infrastructure basée sur des conteneurs pour un SaaS :

- **Déploiement**

- **VM** : Les VM se déploient plus lentement, car il nécessite le provisionnement de l'ensemble de la machine virtuelle.
- **Conteneurs** : Le déploiement des conteneurs peut être rapide, car ils sont prêts à l'emploi avec tous les éléments nécessaires pour l'application.

- **Performance**

- **VM** : Les VM peuvent avoir une surcharge en termes de ressources en raison de l'émulation matérielle, mais elles sont adaptées aux charges de travail exigeantes en ressources.
- **Conteneurs** : Les conteneurs sont plus légers et ont une surcharge de performance minimale, ce qui les rend rapides et scalables pour les déploiements.

- **Isolation**

- **VM** : Chaque machine virtuelle fonctionne sur son propre système d'exploitation, ce qui rend Les machines virtuelles offrent une isolation complète.
- **Conteneurs** : Les conteneurs offrent une isolation plus légère, car ils partagent le même noyau d'OS. Ce qui les rend plus efficaces en termes de ressources.

- **Évolutivité**

- **VM** : L'évolutivité plus complexe que les conteneurs car elle nécessite le déploiement de nouvelles VM, ce qui peut être coûteux en termes de ressources.
- **Conteneurs** : Les conteneurs peuvent être rapidement, ce qui les rend idéaux pour les applications à forte demande.

- **Portabilité**

- **VM** : Les VM sont moins portables car elles dépendent souvent de configurations spécifiques à l'hyperviseur.
- **Conteneurs** : Les conteneurs sont très portables, car ils encapsulent toutes les dépendances de l'application, ce qui les rend faciles à déplacer entre différentes infrastructures.

Enfin, le choix entre VM et conteneurs dépend aux besoins spécifiques de notre SaaS, de nos ressources disponibles et de nos préférences en matière de gestion et de performance. De nombreux déploiements modernes utilisent une combinaison des deux pour tirer parti de leurs avantages respectifs.

3.4.4 Présentation le déploiement d'une application SaaS sur Google Cloud Platform (GCP)

1. L'environnement de déploiement de l'application SAAS sur le google cloud

- **Infrastructure Cloud utilisé**
 - **Compute Engine** : Google Cloud propose des machines virtuelles (VM) hautement évolutives pour exécuter les composants de l'application. Et on choisi les tailles de VM par rapport nos besoins.

- **Bases de données**
 - **Cloud SQL** : Il héberge des bases de données relationnelles telles que MySQL, PostgreSQL et SQL Server.
 - **Cloud Spanner** : Pour des bases de données globalement distribuées et hautement évolutives.

- **Stockage**
 - **Cloud Storage** : Pour stocker des fichiers et des données non structurées.
 - **Persistent Disk** : Utilisé pour le stockage persistant associé à nos VM.

- **Réseau**
 - **Virtual Private Cloud (VPC)** : C'est la création d'un réseau privé virtuel pour isoler l'environnement cloud et contrôler le trafic réseau.
 - **Cloud Load Balancing** : Il répartit la charge du trafic entre plusieurs instances de votre application pour garantir la disponibilité et la scalabilité.

- **Gestion des utilisateurs et de l'authentification**
 - **Identity and Access Management (IAM)** : Il est utilisé pour gérer les autorisations et les rôles des utilisateurs et des services au sein de projet Google Cloud.

- **Services de conteneur**
 - **Google Kubernetes Engine (GKE)** : Utilisé pour gérer et orchestrer les conteneurs Docker de manière automatisée.

- **Services de surveillance et de journalisation**

- **Stackdriver (devenu Google Cloud Operations)** : Pour surveiller les performances de l'application, collecter des journaux et générer des alertes en cas de problèmes.
- **Sécurité**
 - **Identity-Aware Proxy (IAP)** : Pour sécuriser l'accès à l'applications via Google Cloud Identity.
 - **Cloud Identity Platform** : Facilite l'authentification et la gestion des identités dans votre application SaaS.

Voici la figure de l'admin console dans (GCP) (Figure 3.1).

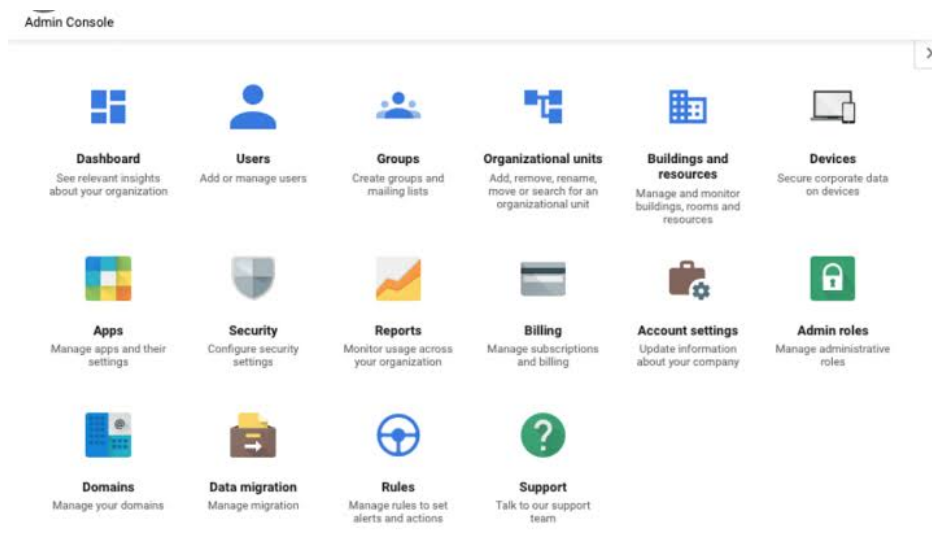


FIGURE 3.1 – Interface de l'admin console

2. **Shéma de l'architecture** la conception de l'architecture de déploiement sur Google Cloud Plateform (GCP) (3.2).

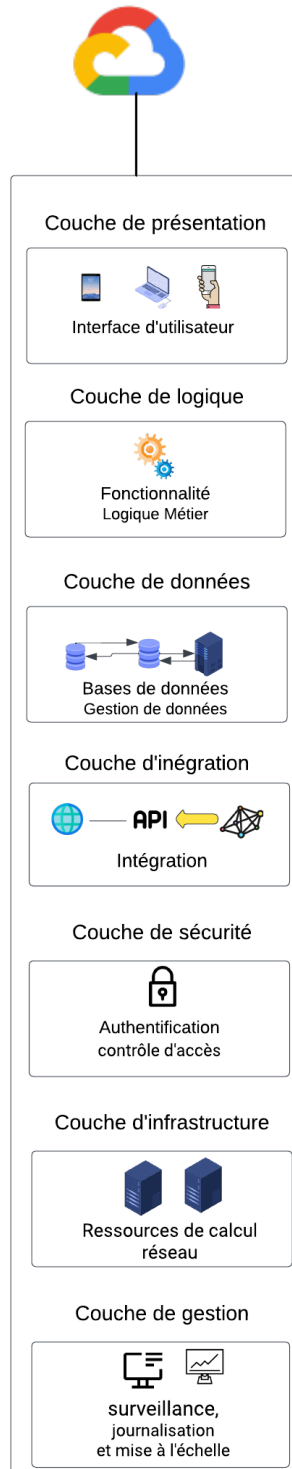


FIGURE 3.2 – Diagramme de l'architecture de l'application SAAS

3.4.5 Déploiement avec Docker

Le Docker est une technologie qui simplifie certaines étapes du processus de déploiement de l'application SAAS.

- **Présentation du Docker**

Docker est une technologie de virtualisation qui permet de créer, déployer et démarrer des applications facilement en utilisant des conteneurs de logiciels ou services.[30]. La plateforme Docker repose sur plusieurs technologies et composants. Voici les principaux éléments :

- **Docker Engine** : Est l'application à installer sur la machine hôte pour créer, exécuter et gérer des conteneurs Docker.
- **Docker Daemon** : Traite les requêtes API afin de gérer les différents aspects de l'installation tels que les images, les conteneurs ou les volumes de stockage.
- **Docker Client** : Est la principale interface permettant de communiquer avec le système Docker. Il reçoit les commandes par le biais de l'interface de ligne de commande et les transmet au Docker Daemon.
- **Dockerfile** : Chaque conteneur Docker débute avec un " Dockerfile ". Il s'agit d'un fichier texte rédigé dans une syntaxe compréhensible, comportant les instructions de création d'une image Docker.
- **Les images Docker** : Est un modèle en lecture seule, utilisé pour créer des conteneurs Docker.
- **Docker run** : Est la commande permettant de lancer un conteneur.
- **Les conteneurs Docker** : Est une instance d'image Docker exécutée sur un microservice individuel ou un stack d'application complet.
- **Le registre Docker** : Est un système de catalogage permettant l'hébergement et le " push and pull " des images Docker. Il Le Docker Hub est le registre officiel de Docker[29].

- **Étapes pour le Déploiement avec Docker**

Voici les étapes pour le déploiement de notre application Web Angular avec Node.js côté serveur et une base de données MySQL en utilisant Docker.

1. **Créer le Dockerfile** : Pour définir les dépendances et les configurations nécessaires pour exécuter votre application dans un conteneur Docker (Figure 3.3).

```

# Stage 1: Build the Angular app
FROM node:18 AS build
WORKDIR /app
COPY att/package*.json ./att/
COPY backend/package*.json ./backend/
RUN npm install -g @angular/cli && \
  cd att && npm install && \
  cd ../backend && npm install
COPY . .
RUN cd att && ng build --configuration production --output-path=../backend/public

# Stage 2: Serve the Angular app with NGINX
FROM nginx:alpine
COPY --from=build /app/backend/public /usr/share/nginx/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]

```

FIGURE 3.3 – Script de Dockerfile

2. **Créer une image Docker** : En utilisant la commande ‘**docker build**’ (Figure 3.4).

```

PS C:\Users\PC\Desktop\New folder\att project> docker build -t attendance-app:v1.
>>
[+] Building 107.2s (15/15) FINISHED

```

FIGURE 3.4 – La commande de docker build

3. **Tester l’image Docker localement** : En utilisant la commande ‘**docker run**’ (Figure 3.5).

```

PS C:\Users\PC\Desktop\New folder\att project> docker run -p 8080:80 attendance-a
>>

```

FIGURE 3.5 – La commande de docker run

4. **Hébergement sur un Registre Docker** : Pour publier les images sur un **Docker Hub** après la création d’un compte sur Docker Hub (Figure 3.6).

```

PS C:\Users\PC\Desktop\New folder\att project> docker login
>>
>>
Log in with your Docker ID or email address to push and pull images from Docker H

```

FIGURE 3.6 – La commande pour docker login

```

PS C:\Users\PC\Desktop\New folder\att project> docker tag attendance-app:v1.0 dal
>>
PS C:\Users\PC\Desktop\New folder\att project> docker push dal/attendance-ap
>>
The push refers to repository [docker.io/daliaray/attendance-app]
107137b76e75: Preparing
ef6182113153: Preparing
107137b76e75: Pushed
2cf6da0936ad: Mounted from library/nginx
ca660b07329b: Mounted from library/nginx
854831065e8f: Mounted from library/nginx
1adf0aa7b921: Mounted from library/nginx
4693057ce236: Mounted from library/nginx
v1.0: digest: sha256:a06d4d6bc3c8c7ccfaad347a749311e3f4342a4453711c7c0b5ff0fbadb7 size: 2199

```

FIGURE 3.7 – La commande pour push une image

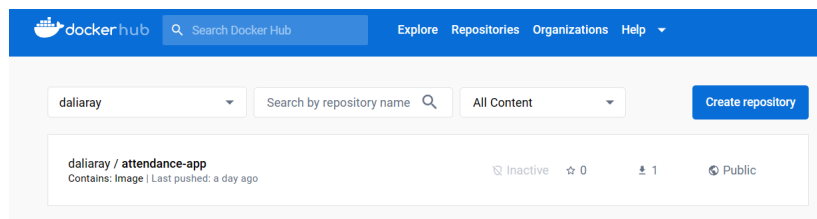


FIGURE 3.8 – Docker Hub

5. **Déploiement de l'Application** : Sur le serveur de production après que le Docker est installé et que les ports nécessaires sont ouverts, on utilise les commandes '**docker pull**' et '**docker run**' pour télécharger et exécuter les images Docker (Figure 3.9).

– Pull l'image

```
docker pull daliaray/attendance-app:v1.0
```

FIGURE 3.9 – La commande pour pull une image

– Run l'image

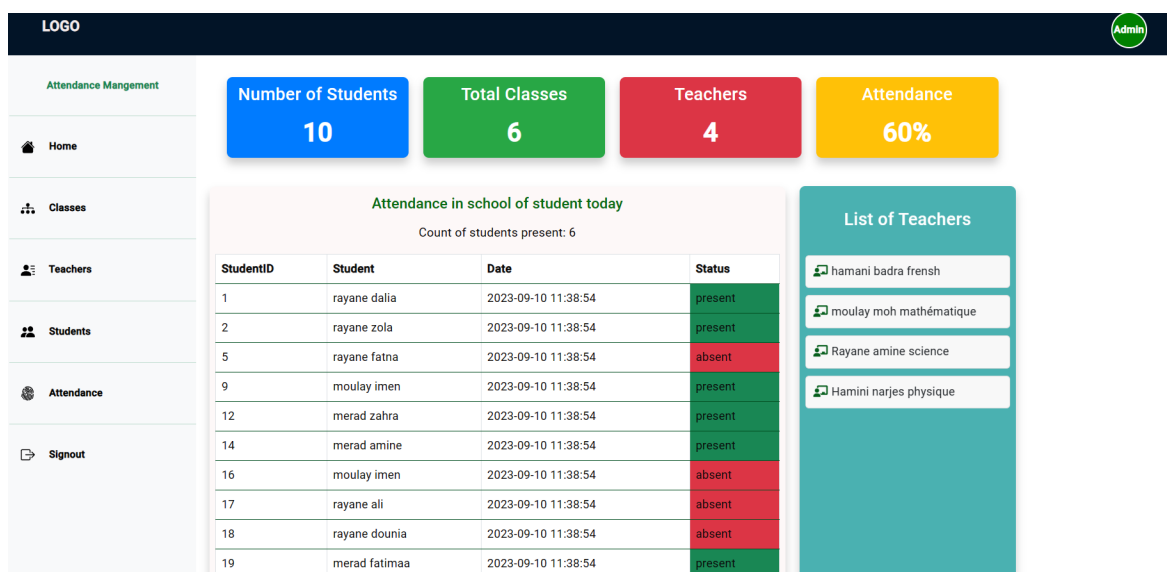


FIGURE 3.10 – La page Home après le déploiement

6. **Gestion des Conteneurs Docker** : On utilise les commandes Docker telles que `docker ps`, `docker logs`, `docker stop`, et `docker start` pour surveiller et gérer les conteneurs en cours d'exécution sur le serveur de production.

3.4.6 Comparaison entre la réalisation de l'auto-hébergement SaaS avec Docker et l'hébergement sur Google Cloud pour le SaaS

Le choix entre réaliser l'auto-hébergement de l'infrastructure avec Docker ou utiliser Google Cloud dépendra sur les compétences techniques, les ressources dis-

ponibles, de les besoin de contrôle, de la tolérance au risque et de la capacité à la gestion de la complexité de l'infrastructure. Voici la comparaison entre les deux utilisation.

Réalisation de l'auto-hébergement SaaS avec Docker

- Contrôle total sur propre infrastructure, avec la configuration, la sécurité et la maintenance.
- Selon les besoins spécifiques, on mis la personnalisation de chaque aspect de l'infrastructure.
- Les coûts initiaux sont généralement plus faibles car permet d'utiliser leurs propres ressources matérielles ou cloud.
- la gestion de toute la complexité de l'infrastructure, y compris la sécurité, la scalabilité et la mise en réseau.

Réalisation de l'hébergement sur Google Cloud pour le SaaS

- Google Cloud offre des services de déploiement faciles à utiliser.
- Google Cloud offre des outils de sécurité avancés et des certifications de conformité.
- La facilité de mettre à l'échelle les ressources en fonction de la demande sans problème de la gestion matérielle.
- Les coûts peuvent être basés sur l'utilisation réelle, ce qui peut être plus flexible.
- Google Cloud gère la majeure partie de l'infrastructure sous-jacente, qui permet de concentrer sur le développement de SaaS plutôt que sur la gestion de l'infrastructure.

3.5 Conclusion

En conclusion, le déploiement de notre SaaS est une étape cruciale dans la création et la gestion d'une application cloud moderne. Dans ce chapitre, nous avons d'abord discuté les besoins en termes d'exigences de logiciel, qu'elles soient fonctionnelles ou non fonctionnelles pour la conception. Ensuite, nous avons exposé l'environnement ainsi que les différents outils de développement et analyser les mesures de sécurité de l'application. Enfin, nous avons exploré deux approches différentes, la mise en place d'une infrastructure avec Docker et présentation de l'utilisation de Google Cloud. De plus, Docker nous offre une solution puissante pour la gestion de conteneurs, qui permettre une flexibilité, des coûts initiaux plus faibles et une personnalisation maximales de notre infrastructure. Cependant, cela implique également une plus grande responsabilité en matière de configuration, de sécurité et de gestion des ressources. D'autre part, Google Cloud fournit une plateforme cloud robuste avec des services simplifier la gestion de déploiement, la gestion et la scalabilité de SaaS. Le choix entre ces deux approches, nous avons utilisé Docker, car il dépendre à nos besoins spécifiques, et nos compétences techniques et de nos objectifs à long terme.

CHAPITRE 4

RÉALISATION DE L'APPAREIL DE DÉTECTION DE PRÉSENCE PAR L'EMPREINTE

4.1 Introduction

Ce chapitre est dédié à la conception et à la réalisation de l'appareil de détection de présence par l'empreinte digitale. Il sera d'abord consacré à la description en détail de l'apparence et l'identification des composants et les technologies utilisés. La conception matérielle et logicielle de l'appareil sera également traitée. Par la suite, le fonctionnement ainsi que les résultats des tests effectués seront présentés. Enfin, nous aborderons les mesures de sécurité qui permettraient de préserver la confidentialité des données biométriques des élèves.

4.2 Description de l'appareil

L'appareil de détection de présence par l'empreinte digitale dans notre système a un rôle de capturer, vérifier et enregistrer les empreintes digitales des individus afin de surveiller leur présence dans l'école. Cet appareil repose sur une technologie biométrique sophistiquée pour garantir une identification précise et fiable des élèves.

4.2.1 Présentation des composants de l'appareil



FIGURE 4.1 – Capteur D'empreinte Digitale DY50

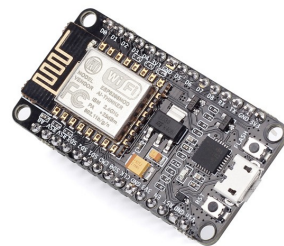


FIGURE 4.2 – Module WIFI ESP8266-E12 NodeMCU V2

1. Le module de capteur d'empreintes digitales

Ces modules qui sont basé sur une technologie optique, livrés avec une mémoire FLASH pour stocker les empreintes digitales et fonctionnent avec n'importe quel microcontrôleur ou système avec série TTL[23]. Dans notre modèle on utilise un capteur d'empreinte Digitale de type (DY50).

• Caractéristiques

- Alimentation en tension : DC 3,6 à 6,0 V
- Alimentation en courant : 120mA
- Couleur du rétroéclairage : vert
- Interface : UART
- Mauvais taux : 9600
- Niveau de sécurité : cinq (de faible à élevé : 1,2,3,4,5)
- Taux de fausses acceptations (FAR) : <0,001
- Taux de faux rejets (FRR) : <1,0
- Fichier de signature : 256 octets
- Capacité de stockage : 1000
- Capable de stocker 127 empreintes digitales différentes[24].

• Connexion matérielle

Le capteur a six broches qui sont étiquetées dans la figure ci-dessous.

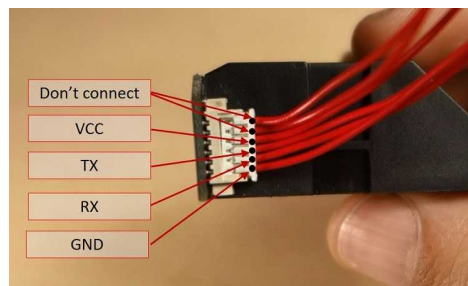


FIGURE 4.3 – Brochage du capteur

Le tableau ci-dessous montre comment câbler le capteur à l'Arduino (Tableau 4.1).

Détecteur d'empreintes digitales	Arduino
VCC	5V (ça marche aussi avec 3.3V)
Émission	Réception (broche numérique 2, logiciel série)
Réception	Émission (broche numérique 3, logiciel série)
GND	GND

TABLE 4.1 – Câblage de capteur avec l'Arduino

2. Le module wifi ESP8266-E12 NodeMCU V2

- Module basé sur un ESP8266 cadencé à 80 MHz et exécutant le firmware open source **NodeMCU**. Cette carte se programme via l'IDE Arduino et est compatible avec les scripts LUA.
- Ce microcontrôleur dispose d'une interface WiFi idéale pour les objets connectés. Des connecteurs latéraux mâles et femelles permettent d'enficher le module sur une plaque de montage rapide.
- L'interface sans fil Wifi permet la création de point d'accès sans fil, l'hébergement d'un serveur, la connexion à internet et le partage des données par exemple.
- Le module se programme directement à partir de l'IDE Arduino (installation d'une extension nécessaire) et nécessite un cordon mini USB (non inclus). Son implantation le rend compatible avec les plaques de connexions rapides[25].

• Caractéristiques

- Alimentation : 5 Vcc via micro-USB
- Microcontrôleur : ESP8266
- Microprocesseur : Tensilica LX106
- Fréquence : 80 MHz
- Mémoire RAM : 64 kB
- Mémoire Flash : 96 kB
- 10 E/S digitales compatibles PWM
- Interfaces : I2C, SPI, UART
- Interface Wifi 802.11 b/g/n 2,4 GHz
- Antenne intégrée
- Température de service : -40 à 125 °C
- Dimensions : 58 x 31 x 12 mm .

3. Connecteurs et ports

L'appareil peut comporter des connecteurs et des ports pour la connectivité aux périphériques externes, tels que des clés USB, des lecteurs de cartes ou des périphériques de sortie.

4.3 La conception matérielle et logicielle de l'appareil

La conception matérielle

La conception matérielle de l'appareil de détection de présence par l'empreinte digitale est un élément clé pour assurer son bon fonctionnement. Voici le schéma de câblage pour réaliser le circuit électronique (Figure 4.4).

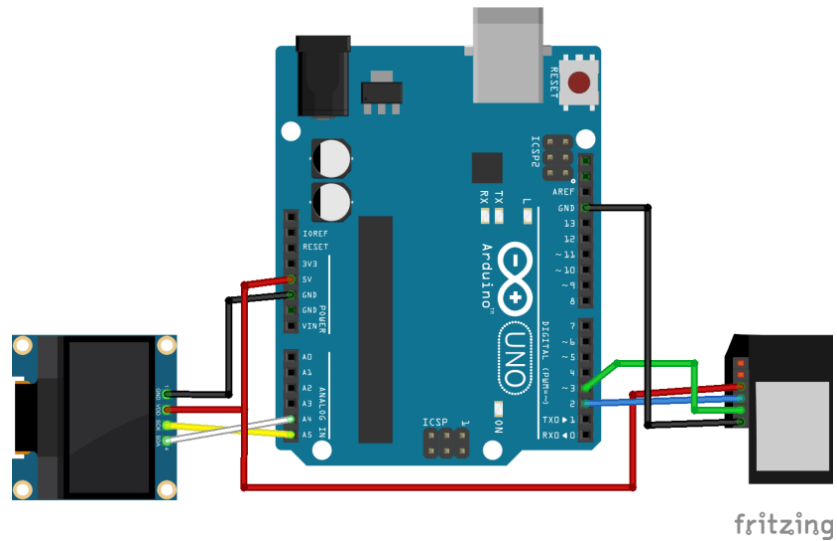


FIGURE 4.4 – Câblage des composants de l'appareil

La conception logicielle

La conception logicielle joue un rôle essentiel dans le bon fonctionnement de l'appareil. Voici les principaux éléments et aspects à considérer lors de la conception logicielle de cet appareil.

- **Firmware basé sur Arduino**

Notre appareil de détection de présence par l'empreinte digitale repose sur un firmware personnalisé développé en utilisant la plateforme Arduino.

- **IDE Arduino**

Le logiciel Arduino est un Environnement de Développement Intégré (IDE) open source et gratuit basés sur des microcontrôleurs Arduino. Qui fournit aux développeurs et aux amateurs une interface conviviale pour écrire, téléverser et déboguer le code destiné aux cartes Arduino[26].

- **Bibliothèque de Capteurs d'Empreintes Digitales utilisé**

Pour faciliter la communication avec le capteur d'empreintes digitales, nous avons intégré la bibliothèque **Adafuit**, qui offre des fonctions préprogrammées pour la capture et le traitement des empreintes digitales.

- **Bibliothèque Adafuit**

Une collection de bibliothèques logicielles open-source créées par **Adafuit Industries** pour faciliter le développement[27].

- **Algorithmes de Gestion des Empreintes Digitales**

- **Enregistrement d’Empreintes Digitales** : Cet algorithme guide les utilisateurs à travers le processus d’enregistrement de leurs empreintes digitales. Il capture l’empreinte, extrait ses caractéristiques distinctives et crée un modèle numérique unique qui est ensuite stocké de manière sécurisée dans la mémoire de l’appareil (Figure 4.5).

```
#include <Adafruit_Fingerprint.h>

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial);
int fingerprintID = 1; // Initialize fingerprint ID to 1

void setup() {
  Serial.begin(57600);
  if (!finger.begin()) while (true);
  finger.deleteAll();
}

void loop() {
  Serial.print("Place your finger on the sensor for fingerprint ");
  Serial.print(fingerprintID);
  Serial.println("...");

  while (!finger.getImage()) delay(500);
  if (finger.image2Tz() != FINGERPRINT_OK) continue;

  if (finger.createModel() == FINGERPRINT_OK) {
    Serial.print("Fingerprint ");
    Serial.print(fingerprintID);
    Serial.println(" enrolled successfully!");
    fingerprintID++; // Increment the fingerprint ID
  } else {
    Serial.println("Failed to enroll fingerprint. Please try again.");
  }
  delay(1000);
}
```

FIGURE 4.5 – Script d’enregistrement d’Empreintes Digitales

- **Reconnaissance d’Empreintes Digitales** : L’algorithme de reconnaissance d’empreintes digitales est chargé de comparer une empreinte capturée avec les modèles enregistrés. Il analyse minutieusement les caractéristiques de l’empreinte digitale pour déterminer si une correspondance est trouvée. Si tel est le cas, l’accès est autorisé (Figure 4.6).

```

#include <Adafruit_Fingerprint.h>

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial);

void setup() {
  Serial.begin(57600);
  if (!finger.begin()) while (true);
}

void loop() {
  Serial.println("Place your finger on the sensor...");

  if (finger.getImage()) {
    if (finger.image2Tz() == FINGERPRINT_OK) {
      int fingerprintID = finger.fingerFastSearch();
      if (fingerprintID != FINGERPRINT_NOID) {
        Serial.println("Fingerprint recognized!");
      } else {
        Serial.println("Fingerprint not recognized.");
      }
    }
  }
  delay(1000);
}

```

FIGURE 4.6 – Script de reconnaissance d’Empreintes Digitales

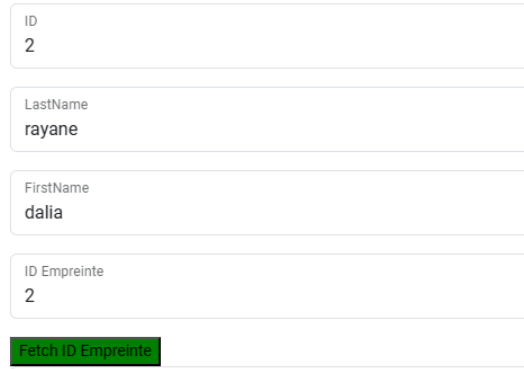
4.4 Fonctionnement de l’Appareil de Détection de Présence

- Dans l’inscription d’un élève dans l’établissement scolaire, son empreinte est capturée à l’aide de l’appareil. Cette empreinte digitale est convertie en un modèle numérique, sécurisé et associé à l’identifiant de l’élève dans la base de donnée.
- Lorsque l’élève se présente à l’école, il place son doigt sur le capteur d’empreintes digitales de l’appareil. Le capteur capture l’empreinte digitale et la compare au modèle enregistré dans la base de donnée.
- L’algorithme de vérification compare l’empreinte capturée avec le modèle enregistré. Si une correspondance est trouvée, cela signifie que l’élève est authentifié avec succès et sa présence est enregistrée automatiquement l’heure à laquelle l’élève a marqué sa présence. Cette information est stockée dans le système de suivi de présence de l’école. .
- Les données de présence de tous les élèves sont enregistrées dans une base de données centrale, ce qui permet au personnel de l’école de suivre facilement l’assiduité des élèves.

4.5 Résultats des tests

Enregistrement de l’empreinte digitale

- Interface de transmission le ID de l’élève pour enregistrer son empreinte (Figure 4.7).



ID
2

LastName
rayane

FirstName
dalia

ID Empreinte
2

Fetch ID Empreinte

FIGURE 4.7 – Interface d’enregistrer l’Empreinte

- Résultat d’identifier une valeur numérique à l’Empreinte dans ArduinoIDE (Figure 4.8).

```
ID 2
Stored!
[HTTP] begin...
[HTTP] PUT...
[HTTP] POST... code: 200
received payload:
<<
{"message":"Student id_empreinte updated"}
>>
```

FIGURE 4.8 – Résultat d’enregistrement d’Empreinte Digitale

Résultat de reconnaissance de l’Empreinte et marquer la présence

(Voir la Figure 4.9.)

```
Image converted
Found a print match!
Found ID #2 with confidence of 107
[HTTP] begin...
[HTTP] Post...
[HTTP] POST... code: 200
received payload:
<<
{"message":"Attendance submitted successfully",'
>>
```

FIGURE 4.9 – Résultat de reconnaissance pour l’Empreinte Digitale

4.6 La figure de l'appareil

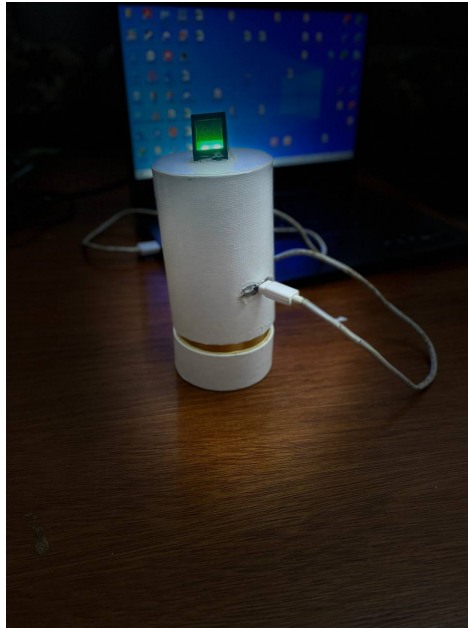


FIGURE 4.10 – L'appareil de l'empreinte

4.7 La sécurité et la protection des données biométriques

La sécurité et la protection des données jouent un rôle essentiel dans le fonctionnement de l'appareil de détection de présence par empreinte digitale. Telles que les modèles d'empreintes digitales, sont stockées de manière sécurisée dans la base de données, avec un accès seulement pour les personnes autorisées et transmises sous forme chiffrée.

4.8 conclusion

En conclusion, ce chapitre a exploré la conception et la réalisation de l'appareil de détection de présence par empreinte digitale. Nous avons décrit l'appareil et ses composants avec les technologies utilisées dans cette appareil. Ainsi que, nous avons expliqué la conception matérielle et logiciel. et le fonctionnement du dispositif du logiciel. Enfin, Nous avons présenté les résultats des test et mis en avant les mesures de sécurité pour préserver la confidentialité des données biométriques des élèves.

IMPLÉMENTATION DES RESEAUX NEURONNES ARTIFICIEL RNN

5.1 Introduction

Ce chapitre se penche sur l'application des avancées en matière d'intelligence artificielle à ce problème complexe. Plus précisément, nous explorerons l'utilisation de réseaux de neurones artificiels, en mettant l'accent sur les RNN avec mécanisme d'attention, pour prédire le taux d'absentéisme dans le cadre d'un système de gestion de présence dans les écoles.

5.2 Objectifs de travail

Dans ce projet, nous avons choisi d'utiliser un réseau de neurones (RNN) avec mécanisme d'attention pour prédire le taux d'absentéisme dans le cadre d'un système de gestion de présence dans les écoles. L'objectif principal de ce travail est de prédire les périodes de "semaines vertes" et "semaines rouges", ce qui peut aider à alerter les responsables de l'école pour prendre des mesures appropriées, comme une intervention précoce.

5.3 Présentations de quelques travaux antérieurs

Dans cette section, nous examinons quelques travaux de recherche précédents qui se sont penchés sur le phénomène de l'absentéisme scolaire. L'objectif est de fournir un contexte approfondi pour notre propre étude, qui se concentre sur l'utilisation de réseaux de neurones artificiels (RNN) avec mécanisme d'attention pour la prédiction du taux d'absentéisme dans le cadre d'un système de gestion de présence dans les écoles.

5.3.1 Article 1 : Predicting and Analyzing Student Absenteeism Using Machine Learning Algorithm

[31]

Résumé : L'objectif est d'analyser, évaluer les causes de l'absentéisme des étudiants en utilisant le machine learning. 26 critères ont été sélectionnés, relatifs à la famille, à la démographie, au social, à l'université et des critères personnels. L'étude a démontré. Que les familles de divorcés ou où un parent est alcoolique et que les

relations avec les autres étudiants jouent un rôle important dans l'absentéisme de l'étudiant.

5.3.2 Article2 : Factors Affecting Pupils' Absenteeism at Felicormfort Junior High School (JHS) in Cape Coast, Ghana

[32]

Résumé : L'objectif était de trouver les facteurs affectant l'absentéisme chez les élèves. L'étude a été réalisée sur une population de 145 élèves et enseignants en utilisant des questionnaires, des interviews. Et l'analyse se faisant avec les moindres carrés, les fréquences et les pourcentages. L'étude a mis en avant des facteurs tels que la concentration, l'intérêt, le niveau financier et la distance à l'école.

5.3.3 Article3 : Revealing underlying factors of absenteeism : A machine learning approach

[33]

Résumé : L'objectif était de savoir que peut-on apprendre à propos des facteurs d'absentéisme en utilisant le machine learning. L'étude a analysé et comparé plusieurs critères. Et a démontré que la relation avec le corps enseignant et le niveau scolaire jouaient un rôle important dans l'absentéisme.

Cependant. Ces études ne font pas de prédiction. Ils ne font que mettre en exergue les critères les plus importants affectant l'absentéisme des élèves. Mais le facteur étant présent ne veut pas dire que l'élève souffrira d'absentéisme. En plus les critères importants diffèrent d'une étude à une autre selon la région, le pays et le niveau scolaire. C'est pour cela que nous avons mis en point. Une méthode de prédiction de l'absentéisme hebdomadaire. Basé sur le machine learning.

5.4 Architecture du Modèle de Réseau de Neurones

Voici les étapes à suivre pour réaliser ce modèle :

1. **Couche d'Entrée :** Cette couche est responsable sur l'acceptation des données en entré ce modèle avec la spécification de la taille de cette couche en fonction du nombre de caractéristiques (features) que nous utiliserons dans cette prédication.
2. **Couche RNN :** Cette couche est le coeur de ce modèle pour capturer les dépendances temporelles à long terme dans les données de la présence à l'aide de cellules LSTM ou GRU.
3. **Mécanisme d'Attention :** En intégrons une couche d'attention pour donner une importance différente à diverses parties de la séquence la séquence de présence.
4. **Couche de Sortie :** L'utilisation de couche dense avec une activation appropriée qui permetre de prédire le taux d'absentéisme.
5. **Fonction de Perte :** On mesure la différence entre les prédictions du modèle et les valeurs réelles d'absentéisme à l'aide de la fonction de perte.

6. **Entraînement** : Cette phase est responsable de l'entraînement ce modèle en minimisant la fonction de perte avec un optimiseur tel que l'optimiseur Adam.
7. **Évaluation** : On va evaluer dans cette partie les performances du modèle à l'aide de métriques telles que l'exactitude, la précision, le rappel, le F-score et la courbe ROC.

5.5 Implémentation d'un modèle de prédiction basé sur RNN

L'objectif de ce code est de prédire les catégories d'absentéisme scolaire (semaines 'green' ou 'red') en utilisant un modèle de réseau de neurones LSTM (Long Short-Term Memory) avec une séquence d'entrée de 4 semaines passées.

Importation des bibliothèques :

Les bibliothèques nécessaires qui sont importées (Figure 5.1) .

- **pandas** pour la manipulation des données.
- **numpy** pour les calculs numériques.
- **TensorFlow/Keras** pour la création et l'entraînement du modèle.
- **scikit-learn** pour certaines opérations de prétraitement des données.
- **EarlyStopping** pour arrêter l'entraînement du modèle en cas de surapprentissage.

```
2 # Importation des bibliothèques nécessaires
3 import pandas as pd
4 import numpy as np
5 from tensorflow.keras.models import Sequential
6 from tensorflow.keras.layers import LSTM, Dense
7 from sklearn.preprocessing import LabelEncoder
8 from sklearn.model_selection import train_test_split
9 from tensorflow.keras.callbacks import EarlyStopping
```

FIGURE 5.1 – Bibliothèques importées dans script de prédiction

Chargement des données :

Les données d'assiduité sont chargées à partir d'un fichier CSV appelé 'attendancedata.csv' à l'aide de la bibliothèque **pandas** (Figure 5.2).

```
10
11 # Chargement des données à partir d'un fichier CSV
12 data = pd.read_csv('attendance_data.csv')
13
```

FIGURE 5.2 – Code de collecte de data

Définition de la longueur de la séquence :

La variable 'sequence-length' est définie pour spécifier le nombre de semaines passées à considérer pour la prédiction future (Figure 5.3).

```
14 # Définition de la longueur de la séquence
15 sequence_length = 4
```

FIGURE 5.3 – Code de définir la longueur de séquence

Prétraitement des données :

Les étiquettes de la colonne 'semaine' (qui peut prendre les valeurs 'green', 'red') sont converties en labels numériques à l'aide de `LabelEncoder` (Figure 5.4).

```
17 # Conversion des étiquettes de 'semaine' en labels numériques
18 label_encoder = LabelEncoder()
19 data['semaine_encoded'] = label_encoder.fit_transform(data['semaine'])
20
```

FIGURE 5.4 – Code de prétraitement des données

Préparation des séquences et des labels :

Les données sont préparées pour l'entraînement du modèle **LSTM**. Pour chaque étudiant, des séquences de longueur **sequence-length** sont créées à partir des données d'assiduité. Les séquences sont stockées dans la liste **X**, et les étiquettes correspondantes (la semaine suivante : le résultat) sont stockées dans la liste **Y** (Figure 5.5).

```
21 # Préparation des séquences et des étiquettes
22 X = []
23 y = []
24
25 # Boucle sur chaque étudiant
26 for student_id, student_data in data.groupby('student_id'):
27     student_data = student_data.sort_values('date')
28     for i in range(len(student_data) - sequence_length):
29         X.append(student_data['status'].values[i:i+sequence_length])
30         y.append(student_data['semaine_encoded'].values[i+sequence_length])
31
32 # Conversion des listes en tableaux numpy
33 X = np.array(X)
34 y = np.array(y)
35
```

FIGURE 5.5 – Code de préparation des séquences et des labels

Division des données :

Les données sont divisées en ensembles d'entraînement, de validation et de test à l'aide de `train-test-split` de `scikit-learn` (Figure 5.6).

```
36 # Division des données en ensembles d'entraînement, de validation et de test
37 X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.2, random_state=42)
38 X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=42)
39
```

FIGURE 5.6 – Code de division des données

Création du modèle LSTM :

Un modèle de réseau de neurones LSTM est créé à l'aide de `Keras Sequential`. Il comporte une couche LSTM avec 50 unités suivie d'une couche Dense avec 3 unités. La couche Dense utilise une fonction d'activation `softmax` pour la classification en trois catégories de semaines ('green', 'red', 'normal') (Figure 5.7).

```
40 # Création du modèle LSTM
41 model = Sequential()
42 model.add(LSTM(units=50, input_shape=(sequence_length, 1)))
43 model.add(Dense(3, activation='softmax')) # Couche Dense a
44
```

FIGURE 5.7 – Code de création du modèle LSTM

Compilation du modèle :

Le modèle est compilé en spécifiant la fonction de perte (`'sparse_categorical_crossentropy'`), l'optimiseur (`'adam'`), et les métriques à suivre (`'accuracy'`) (Figure 5.8).

```
44
45 # Compilation du modèle avec une fonction de perte, un optimiseur et des métriques
46 model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
47
```

FIGURE 5.8 – Code de compilation du modèle

Mise en œuvre de l'arrêt anticipé :

Un rappel d'arrêt anticipé est configuré pour surveiller la perte de validation et arrêter l'entraînement (Figure 5.9).

```
48 # Mise en place de l'arrêt anticipé pour éviter le surapprentissage
49 early_stopping = EarlyStopping(monitor='val_loss', patience=5)
50
```

FIGURE 5.9 – Code de mise en œuvre de l'arrêt anticipé

Entraînement du modèle :

Le modèle est entraîné sur les données d'entraînement avec un nombre d'époques limité (20) et une taille de lot de 32 (Figure 5.10).

```
51 # Entraînement du modèle avec un nombre d'époques limité
52 model.fit(X_train, y_train, epochs=20, batch_size=32, validation_data=(X_val, y_val), callbacks=[early_stopping])
53
```

FIGURE 5.10 – Code de l'entraînement du modèle

Évaluation du modèle :

Le modèle est évalué sur l'ensemble de test, et la perte de test et la précision de test qu'ils seront affichées (Figure 5.11).

```
54 # Évaluation du modèle sur l'ensemble de test
55 test_loss, test_accuracy = model.evaluate(X_test, y_test)
56 print(f'Test Loss: {test_loss:.4f}, Test Accuracy: {test_accuracy:.4f}')
57
```

FIGURE 5.11 – Code de l'évaluation du modèle

Utilisation du modèle pour les prédictions :

Enfin, le modèle entraîné est utilisé pour faire des prédictions sur une séquence future représentée par **future-sequence** (Figure 5.12).

```
58 # Utilisation du modèle pour faire des prédictions
59 future_sequence = np.array([[0, 1, 0, 1]]) # Séquence future (0 pour absent, 1 pour présent)
60 predicted_probabilities = model.predict(future_sequence)
61 predicted_label = np.argmax(predicted_probabilities, axis=-1)
62 predicted_category = label_encoder.inverse_transform(predicted_label)
63 print(f'Predicted Category for Future Week: {predicted_category[0]}')
64
```

FIGURE 5.12 – Code de l'utilisation du modèle pour les prédictions

5.5.1 Discussion des résultats

Ce modèle a prédit **"green"** comme catégorie d'absentéisme pour la séquence d'absentéisme future. Donc le modèle estime qu'il y aura peu d'absences lors de la semaine future, et cela montre que le modèle a effectué une prédiction avec succès (Figure 5.13).

```
Epoch 1/20
1/1 [=====] - 3s 3s/step - loss: 1.0904 - accuracy: 0.4400 - val_loss: 1.1113 - val_accuracy: 0.0000e+00
Epoch 2/20
1/1 [=====] - 0s 44ms/step - loss: 1.0880 - accuracy: 0.4400 - val_loss: 1.1161 - val_accuracy: 0.0000e+00
Epoch 3/20
1/1 [=====] - 0s 45ms/step - loss: 1.0857 - accuracy: 0.4400 - val_loss: 1.1209 - val_accuracy: 0.0000e+00
Epoch 4/20
1/1 [=====] - 0s 60ms/step - loss: 1.0835 - accuracy: 0.4400 - val_loss: 1.1259 - val_accuracy: 0.0000e+00
Epoch 5/20
1/1 [=====] - 0s 60ms/step - loss: 1.0813 - accuracy: 0.4400 - val_loss: 1.1310 - val_accuracy: 0.0000e+00
Epoch 6/20
1/1 [=====] - 0s 61ms/step - loss: 1.0791 - accuracy: 0.4400 - val_loss: 1.1362 - val_accuracy: 0.0000e+00
1/1 [=====] - 0s 36ms/step - loss: 1.1849 - accuracy: 0.0000e+00
Test Loss: 1.1849, Test Accuracy: 0.0000
1/1 [=====] - 0s 494ms/step
Predicted Category for Future Week: green
```

FIGURE 5.13 – Le résultat de prédiction

Voilà le résultat de test de prédiction pour les dix semaines suivante et se présenter en fromage graphique avec le poucentage des semaines rouges et les semaines vertes, et on consate que les semaines vetrtes sont plus que les semaines rouges dans la figure ci-dessous.

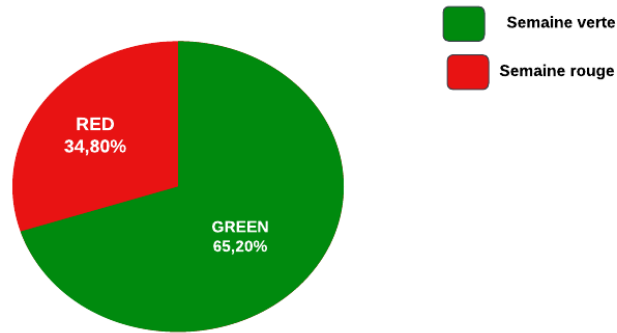


FIGURE 5.14 – Graphique en secteur de prédiction

Ce graphique ci-dessous représente l'apprentissage par rapport la précision et la perte (Figure 5.15).

- Plus l'apprentissage augmente, plus la précision augmente rapidement.
- Plus l'apprentissage augmente, plus la perte diminue rapidement.

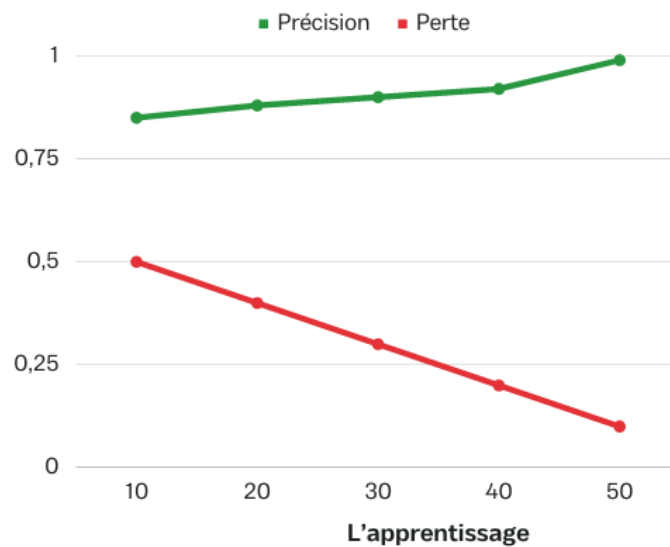


FIGURE 5.15 – Graphique en courbes de l'apprentissage

Ce graphique ci-dessous représente les nombres de neurones par rapport la précision et la perte (Figure 5.16).

- Plus le nombre de neurones augmente, plus la perte diminue lentement.
- Avec plus de nombres de neurones, la perte diminue très lentement

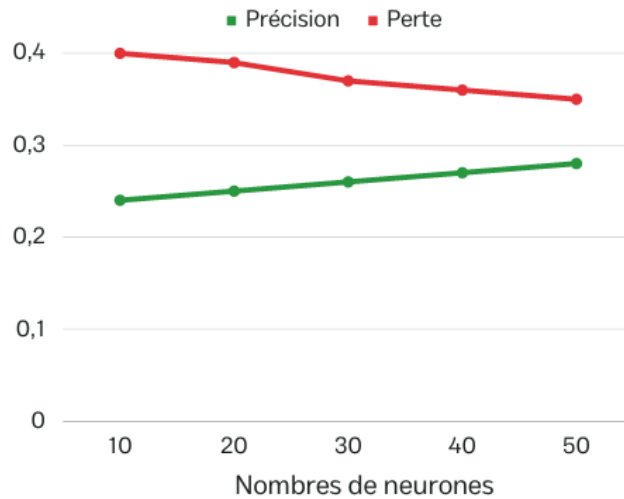


FIGURE 5.16 – Graphique en courbes de nombres de neurones

5.6 Conclusion

En conclusion, ce chapitre marque une avancée significative dans notre quête pour développer des outils avancés de gestion de l'absentéisme scolaire. L'utilisation de l'intelligence artificielle, et en particulier des RNN avec mécanisme d'attention, offre des perspectives prometteuses pour prédire avec précision les absences des élèves et pour mieux comprendre les facteurs qui y contribuent.

CONCLUSION GÉNÉRALE

6.1 Conclusion

Ce mémoire a exploré en profondeur l'optimisation de la gestion de l'assiduité des élèves grâce à un SAAS innovant basé sur des réseaux de neurones et une détection de présence IoT. Au fil de ce travail, nous avons franchi plusieurs étapes essentielles. La première partie, a consisté à explorer les concepts fondamentaux du SAAS, de l'IA basée sur les réseaux de neurones et de l'IoT. Cette étape nous a permis de comprendre comment ces technologies peuvent être appliquées de manière innovante pour résoudre un problème aussi crucial que la gestion de l'assiduité des élèves.

Ensuite, nous avons procédé à la conception et au développement d'une application dédiée à la gestion de l'assiduité des élèves. Nous avons utilisé des outils modernes tels que l'UML pour la conception de nos diagrammes aussi Angular, Node.js et MySQL pour créer une application robuste et flexible. Cette application permet de collecter, de stocker et de gérer efficacement les données de suivi de l'assiduité des élèves dans les établissements scolaires.

Nous avons aussi intégré l'IoT, cette étape qui a été essentielle de notre travail, avec la création d'un appareil de détection de présence par empreinte digitale. Cet appareil offre une précision inégalée dans la collecte des données de présence.

De plus, Nous avons déployé notre application en tant que service SaaS, grâce à Docker pour garantir une accessibilité étendue et une gestion efficace des ressources.

Enfin, nous avons franchi une étape majeure en implémentant un modèle basé sur les réseaux neuronaux RNN avec attention pour prédire le taux d'absentéisme des élèves et préciser les semaines vertes et les semaines rouges, ce qui permet de donner des informations aux administrateurs de les établissements afin de prendre des décisions éclairées en matière de gestion de l'assiduité.

Dans l'ensemble, ce mémoire a proposé une solution complète pour optimiser la gestion de l'assiduité des élèves. Il a combiné les avancées du SAAS, de l'IA et de l'IoT pour créer une solution innovante.

6.2 Futures améliorations

Cette thèse a ouvert des améliorations perspectives fort intéressantes, néanmoins, des améliorations pourraient être introduites à l'avenir ; dont :

- Renforcer et investir dans la sécurité car c'est une priorité fondamentale pour toutes les applications SAAS, elle assure la scalabilité et la crédibilité afin de garantir la confiance de nos clients.
- Ajouter dans l'application une option pour que les parents puissent recevoir des notifications en temps réel sur la présence de leurs enfants à l'école et on classe.
- Intégrer des notifications automatisées pour avertir les enseignants et les élèves en cas d'absence non autorisée.
- Améliorer le modèle de prédiction en temps réel pour une alerte plus rapide.

BIBLIOGRAPHIE

- [1] Akbi Khalil and Zehri Mohammed, *Etude et mise en place d'une solution cloud computing privé au sein de l'université de Ouargla*, 2013.
- [2] Djeriri Youcef, *Les réseaux de neurones artificiels de l'université sidi-bel-abbas*, septembre 2017.
- [3] IBM, *les réseaux de neurones*, 2021.
- [4] *Introduction de réseaux de neurones, bejia*, 2020.
- [5] Vonintsoa. *Apprentissage supervisé et non supervisé*, Mars 2022.
- [6] <https://www.lucidchart.com/pages/fr/langage-uml>, [consulté le 10/08/2023].
- [7] <https://www.lucidchart.com/blog/fr/types-de-diagrammes-UML>, [consulté le 15/08/2023].
- [8] <https://inf1410.teluq.ca/teluqDownload.php?file=2014/01/INF1410-PresentationStarUML.pdf>, [consulté le 15/08/23].
- [9] <https://www.lucidchart.com/pages/fr/diagramme-de-classes-uml>, [consulté le 15/08/2023].
- [10] <https://www.lucidchart.com/pages/fr/diagramme-de-sequence-uml>, [consulté le 15/08/2023].
- [11] <https://kinsta.com/fr/base-de-connaissances/qu-est-ce-que-mysql/>, [consulté le 20/08/2023].
- [12] <https://docs.phpmyadmin.net/>, [consulté le 20/08/2023].
- [13] <https://learn.microsoft.com/en-us/graph/use-postman>, [consulté le 20/08/2023].

- [14] <https://monpetitdev.fr/cest-quoi-angular-definition/>, [consulté le 22/08/2023].
- [15] https://developer.mozilla.org/fr/docs/Learn/HTML/Introduction_to_HTML, [consulté le 22/08/2023].
- [16] <https://www.ionos.fr/digitalguide/sites-internet/web-design/quest-ce-que-le-css/>, [consulté le 22/08/2023].
- [17] <https://kinsta.com/fr/base-de-connaissances/guide-complet-typescript/>, [consulté le 22/08/2023].
- [18] <https://www.journaldunet.com/web-tech/developpeur/1159810-bootstrap-definition-tutoriels-astuces-pratiques/>, [consulté le 22/08/2023].
- [19] <https://tutoriels.edu.lat/pub/angular-material/angular-material-overview/materiau-angularaire-apercu>, [consulté le 22/08/2023].
- [20] <https://fr.education-wiki.com/3958588-what-is-visual-studio-code>, [consulté le 22/08/2023].
- [21] <https://auth0.com/fr/learn/token-based-authentication-made-easy>, [consulté le 23/08/2023].
- [22] <https://www.cloudflare.com/fr-fr/learning/security/threats/sql-injection/>, [consulté le 23/08/2023].
- [23] <https://www.raspberryme.com/>, [consulté le 01/09/2023].
- [24] <https://binarytech-dz.com/produit/capteurs-robotique/capteurs/magnetique-capacitif-potentiometres/capteur-dempreinte-digitale-dy50/>, [consulté le 01/09/2023].
- [25] <https://binarytech-dz.com/produit/arduino-raspberry/cartes-diveres/module-wifi-esp8266-e12-nodemcu-v2/>, [consulté le 03/09/2023].
- [26] <https://arduino.blaisepascal.fr/presentation/logiciel/>, [consulté le 03/09/2023].
- [27] <https://arduino-france.site/adafruit>, [consulté le 03/09/2023].
- [28] <https://kinsta.com/fr/base-de-connaissances/qu-est-ce-que-node-js/>, [consulté le 06/09/2023].
- [29] <https://datascientest.com/docker-guide-complet>, [consulté le 10/09/2023].

- [30] <https://reactic.io/archives-blog/avantages-docker>, [consulté le 10/09/2023].
- [31] Bowen, F., Gentle-Genitty, C., Siegler, J., & Jackson, M. (2022). Revealing underlying factors of absenteeism : A machine learning approach. *Frontiers in Psychology*, 13. <https://doi.org/10.3389/fpsyg.2022.958748>
- [32] Mukli, L., Rista, A. (2022). Predicting and Analyzing Student Absenteeism Using Machine Learning Algorithm. University Aleksandër Moisiu Durres, Durres, Albania. Integration of education. DOI : 10.15507/1991-9468.107.026.202202.216-228.
- [33] Senyamator, F., Gyimah, E. K., & Minadzi, V. M. (2018). Factors Affecting Pupils' Absenteeism at Felicormfort Junior High School (JHS) in Cape Coast, Ghana. *Journal of Education and Learning*, 7(6). <https://doi.org/10.5539/jel.v7n6p138>
- [34] <https://www.redhat.com/fr/topics/cloud-computing/iaas-vs-paas-vs-saas>, [consulté le 20/09/2023].

DÉTAIL DE LA CONCEPTION ET LA RÉALISATION DE L'APPLICATION

Les diagrammes de la conception de l'application

Le diagramme de classe

Le diagramme de classes est l'un des types les plus populaires en langage UML. Il est un type de diagramme de structure, et il met, les classes, les objets, les attributs, les méthodes et les relations entre ces éléments[9] (Figure 6.1).

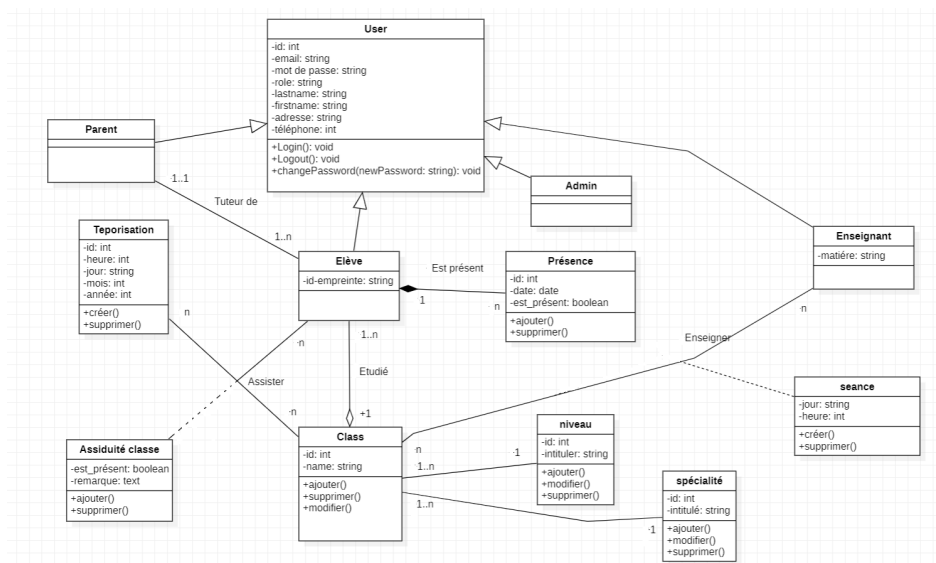


FIGURE 6.1 – Diagramme de classes

Le diagramme de cas d'utilisation

Le diagramme de cas d'utilisation illustre les personnes qui utilisent le système. Il offre une vue graphique des acteurs impliqués dans le système, des différentes fonctions nécessaires pour ces acteurs, ainsi que de la manière dont ces fonctions interagissent entre elles (Figure 6.2).

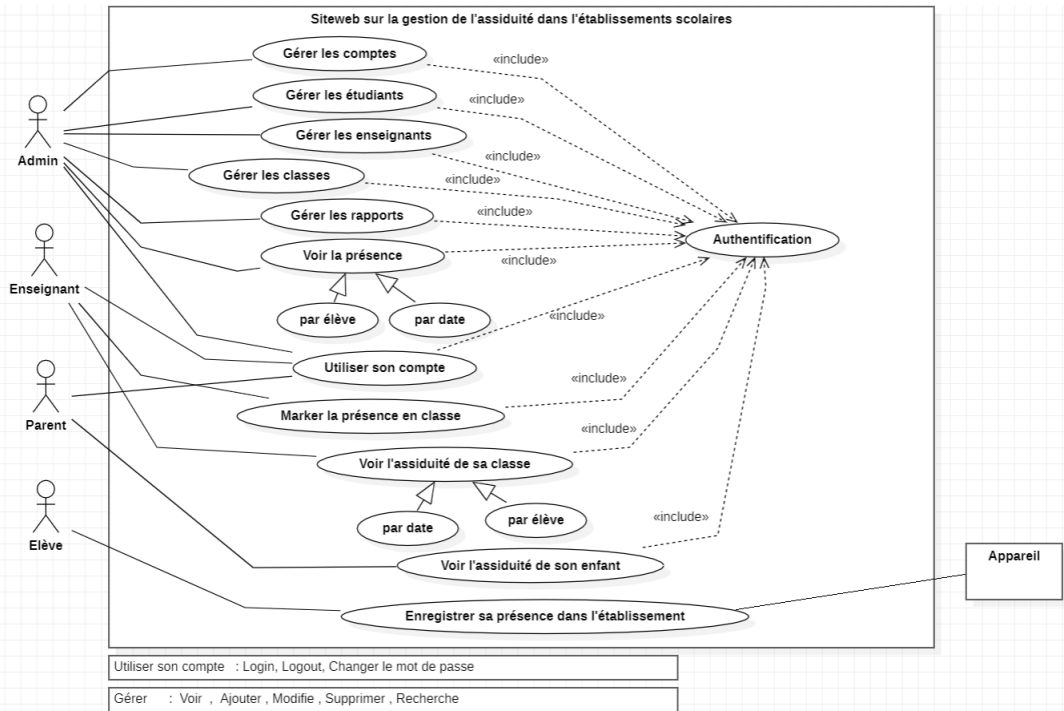


FIGURE 6.2 – Diagramme de cas d'utilisation

Le diagramme de séquence

Un diagramme de séquence est un type de diagramme d'interaction, car il décrit comment et dans quel ordre plusieurs objets fonctionnent ensemble. Ces diagrammes sont utilisés à la fois par les développeurs logiciels et les managers d'entreprises pour analyser les besoins d'un nouveau système [10].

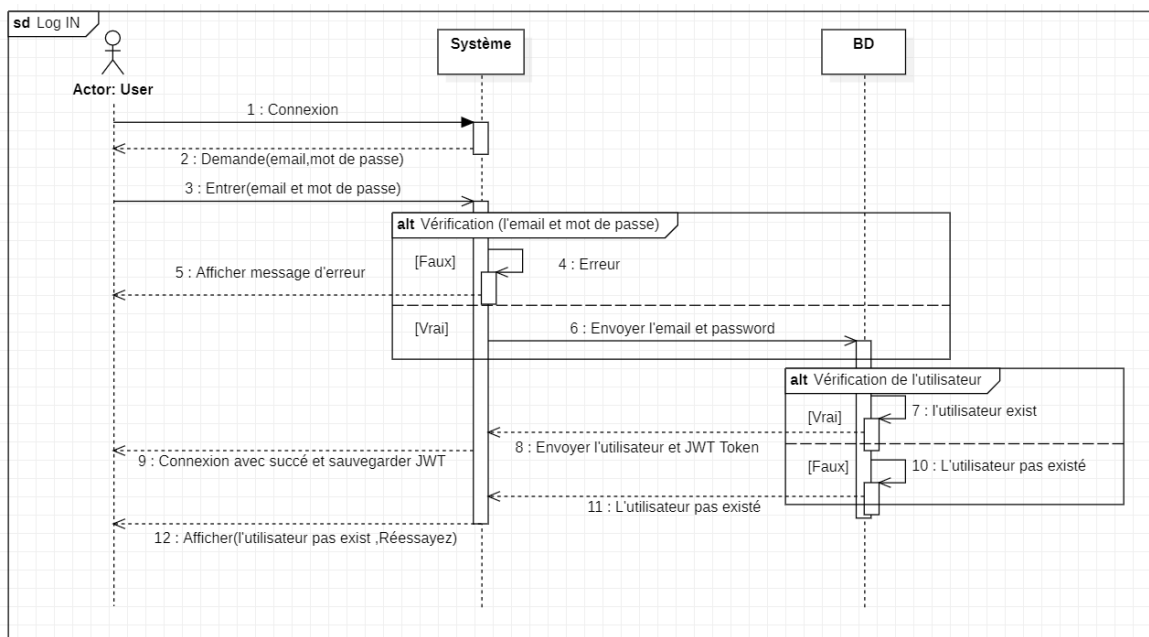


FIGURE 6.3 – Diagramme de séquence login

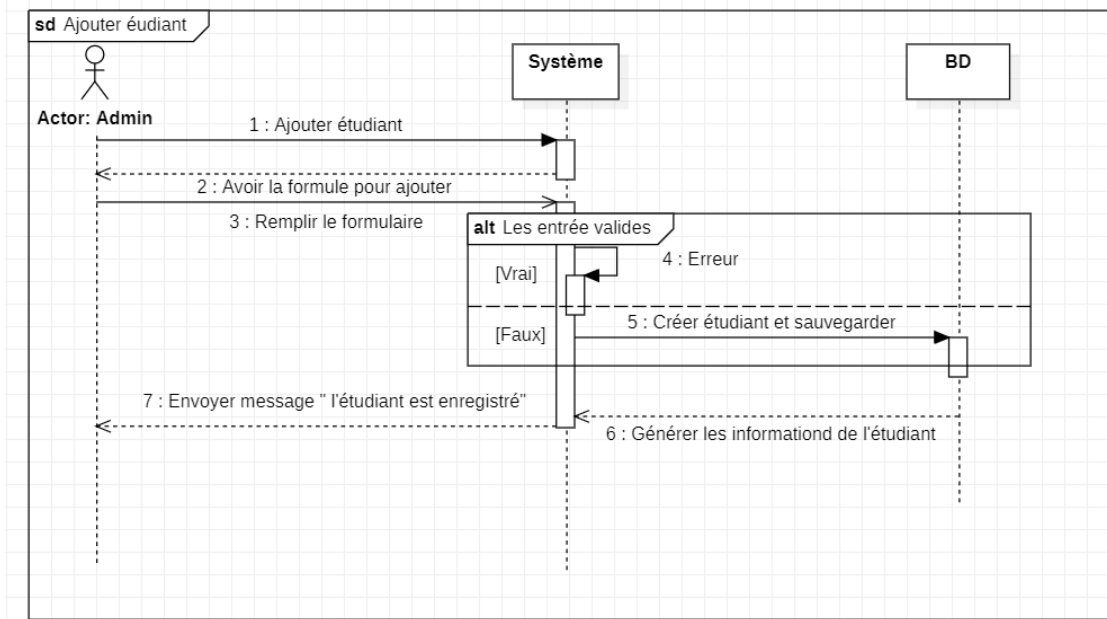


FIGURE 6.4 – Diagramme de séquence ajouter étudiant

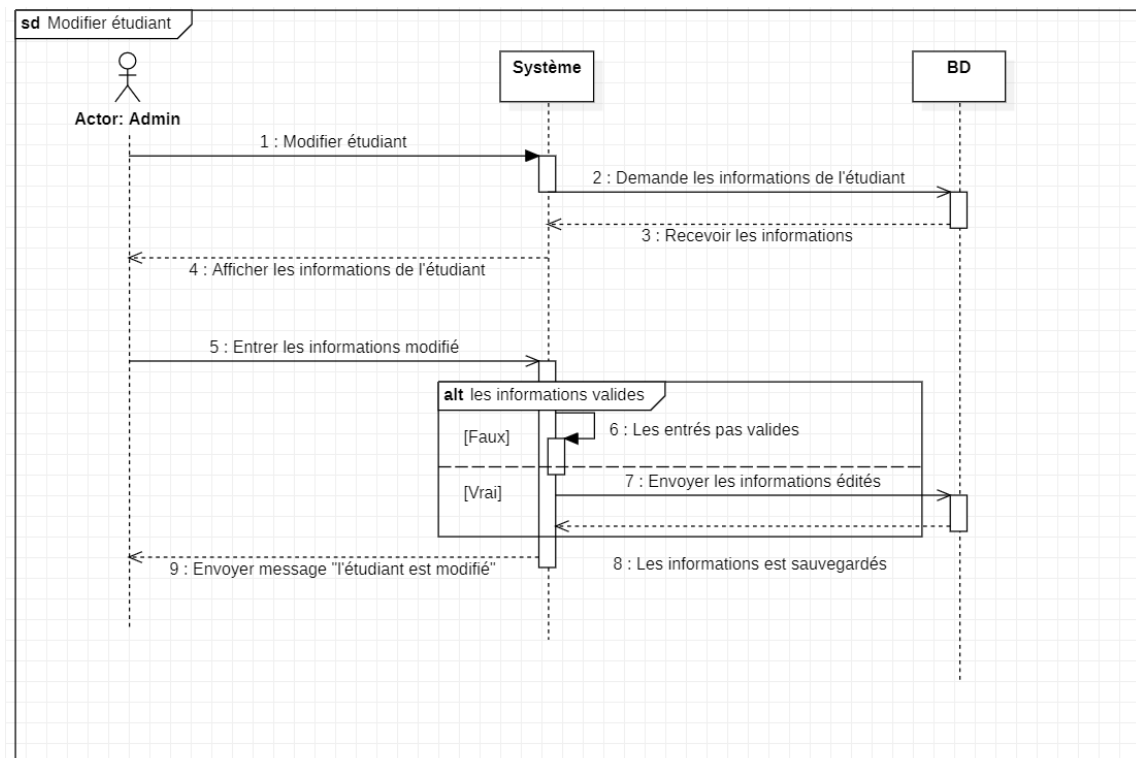


FIGURE 6.5 – Diagramme de séquence modifier étudiant

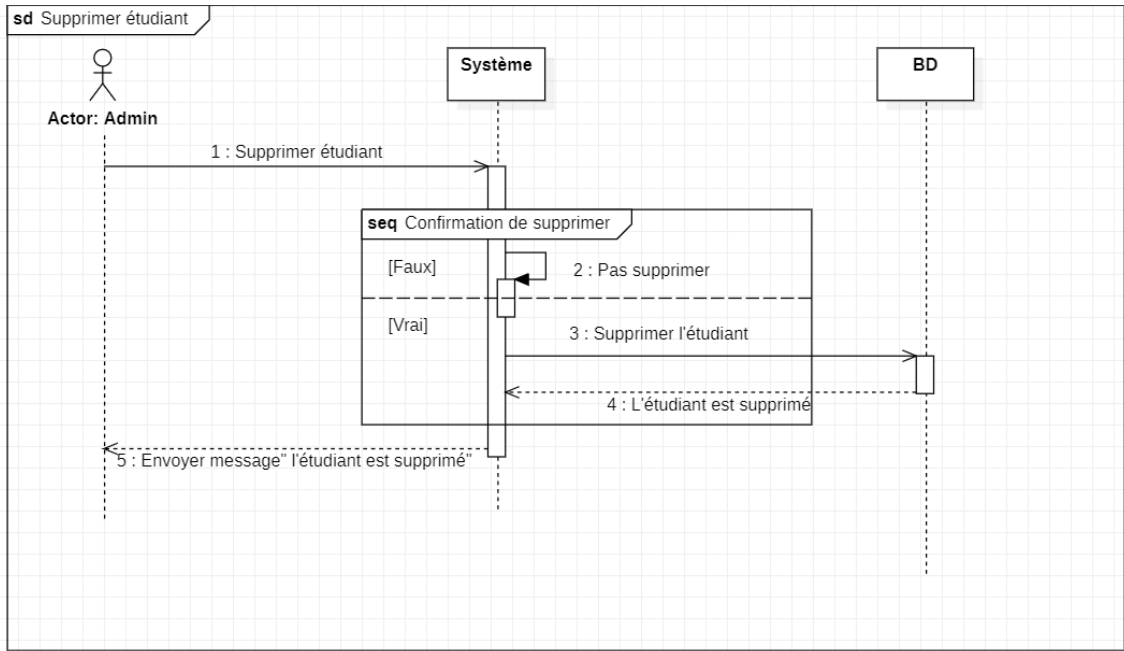


FIGURE 6.6 – Diagramme de séquence supprimer étudiant

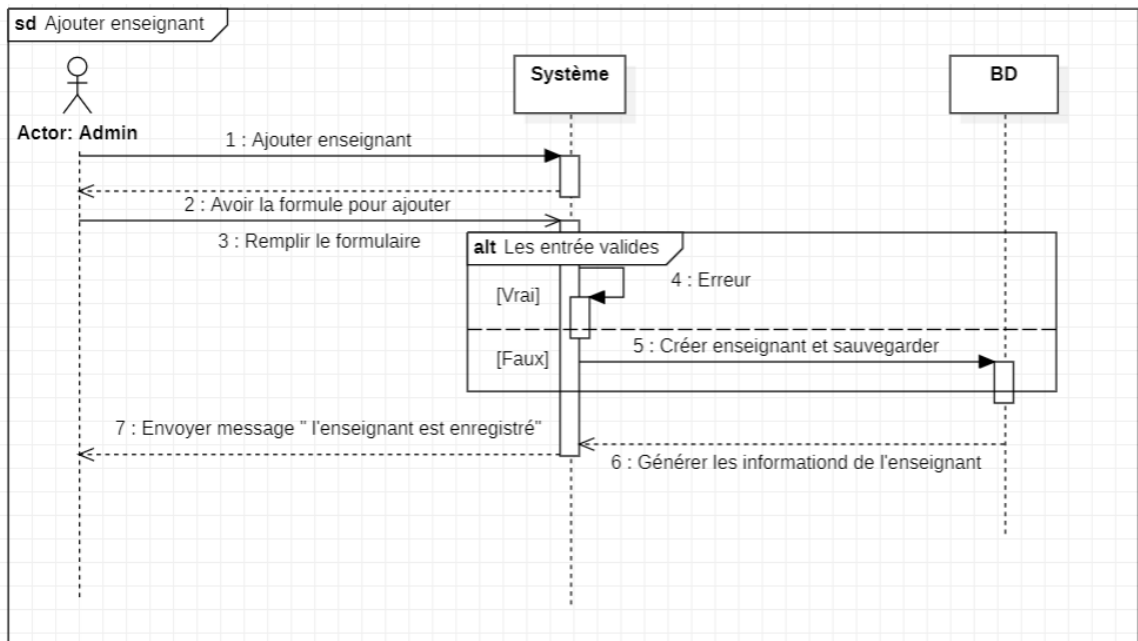


FIGURE 6.7 – Diagramme de séquence ajouter enseignant

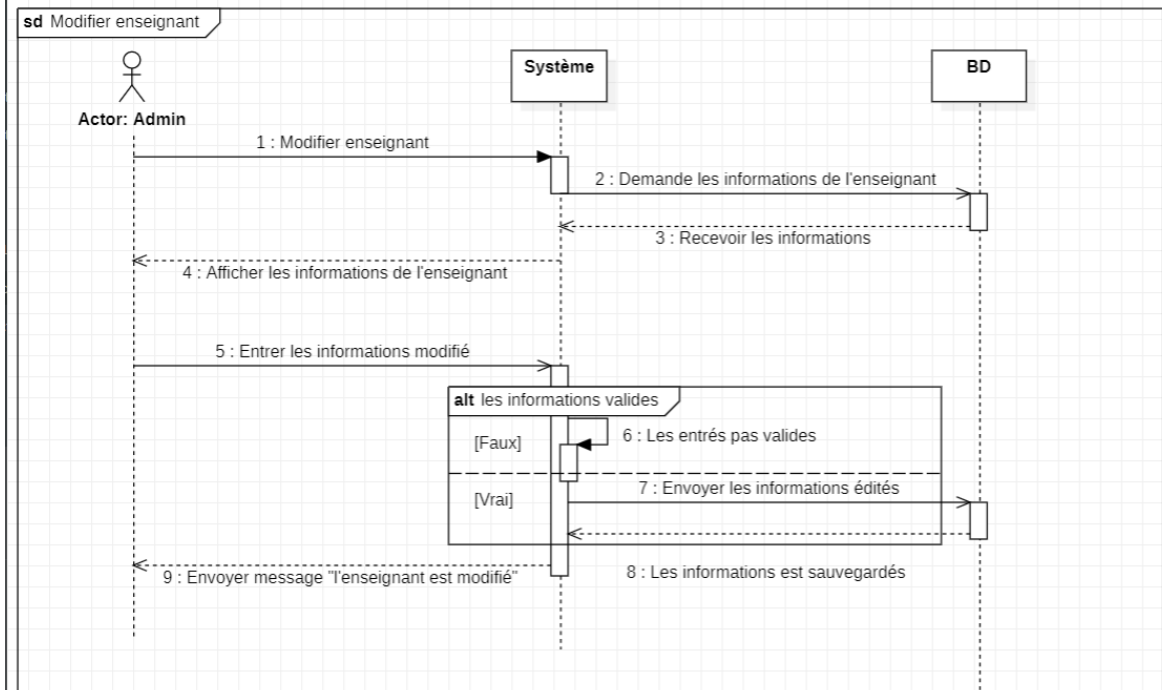


FIGURE 6.8 – Diagramme de s  quence modifier enseignant

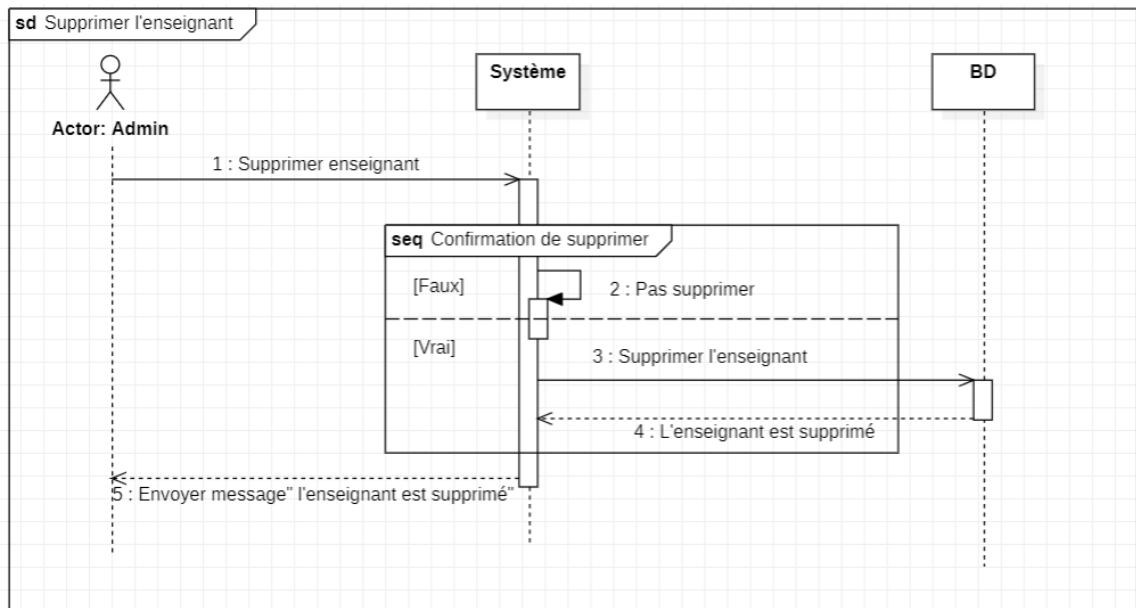


FIGURE 6.9 – Diagramme de s  quence supprimer enseignant

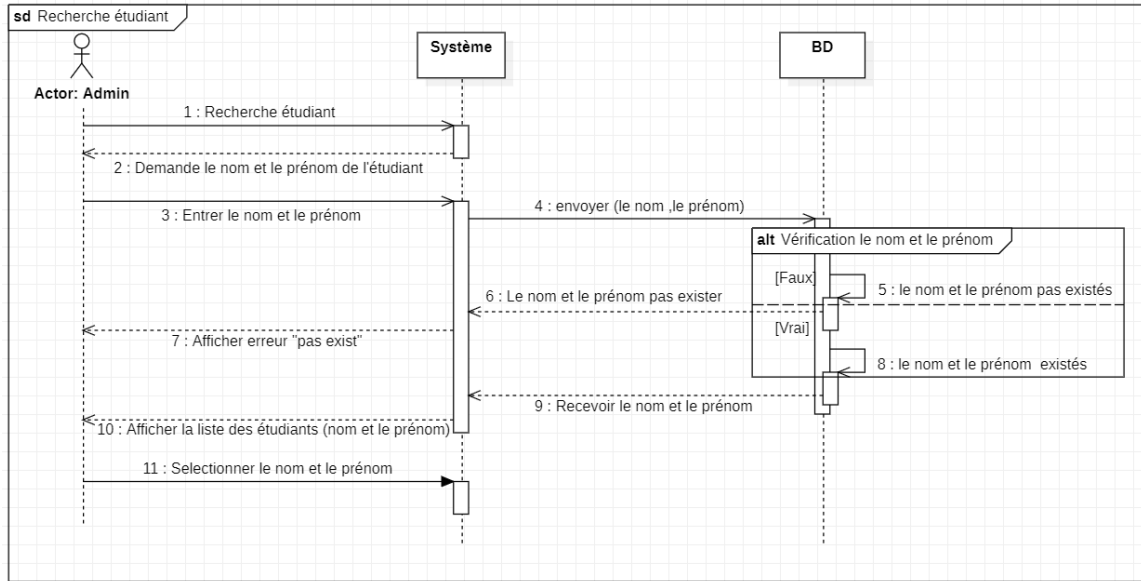


FIGURE 6.10 – Diagramme de séquence recherche étudiant

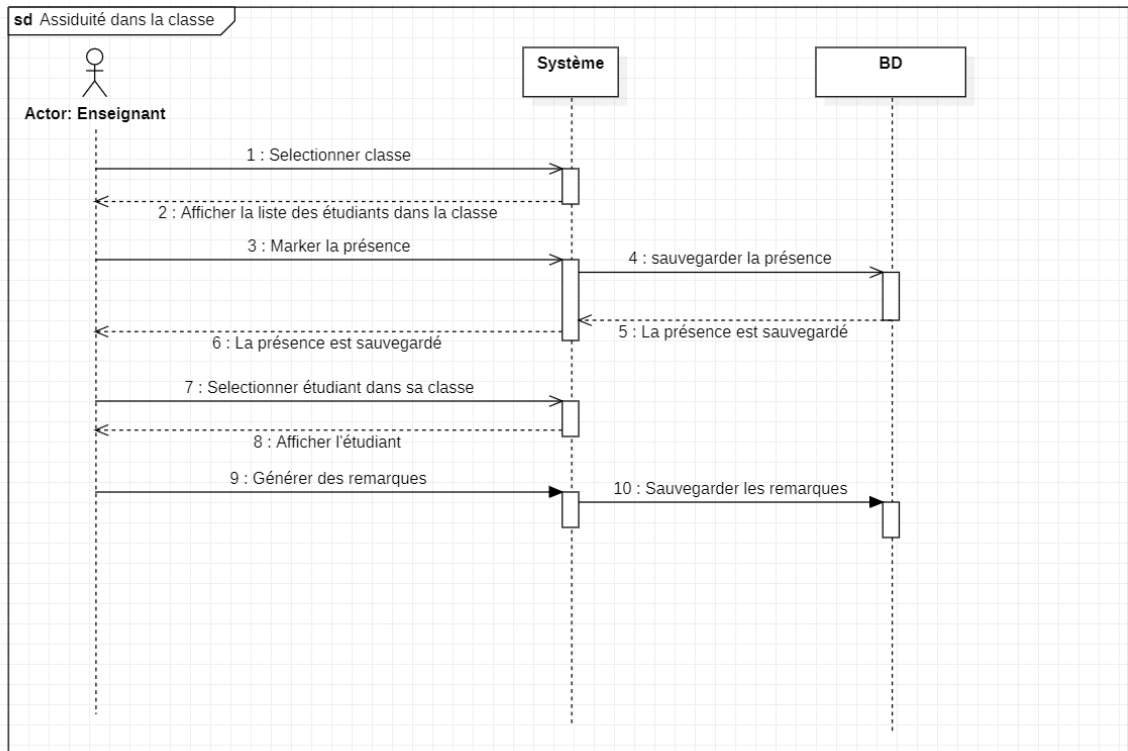


FIGURE 6.11 – Diagramme de séquence d'assiduité dans classe

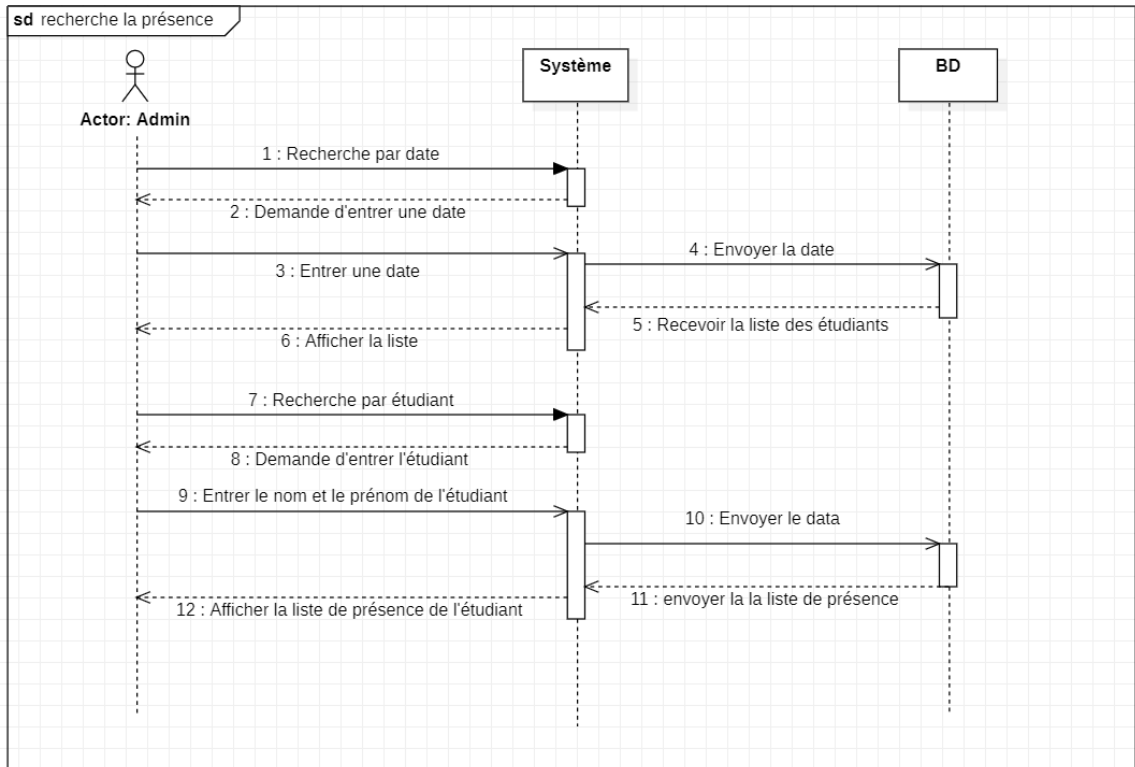


FIGURE 6.12 – Diagramme de séquence recherche de présence

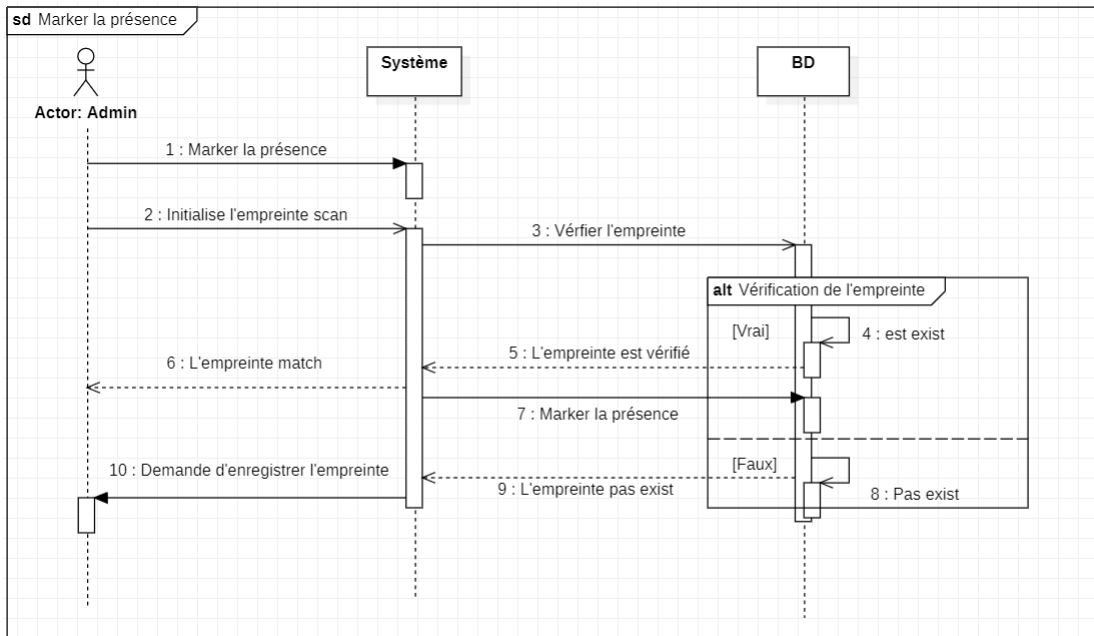


FIGURE 6.13 – Diagramme de séquence de marker la présence

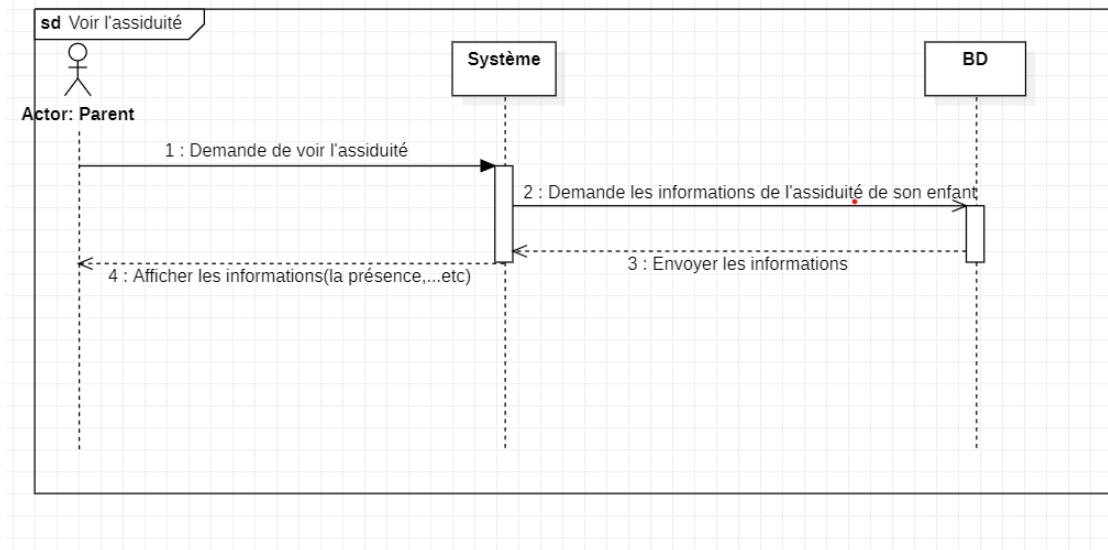


FIGURE 6.14 – Diagramme de séquence de voir la présence

Captures d'écran de la présentation de l'application

Connexion

Cette page offre un portail de connexion sécurisé pour les utilisateur de ce logiciel. Pour accéder, les utilisateurs doivent entrer leurs coordonnées(nom d'utilisateur et le mot de passe). De plus, le système effectue une vérification basée sur le rôle de l'utilisateur pour déterminer les fonctionnalités et les autorisations auxquelles ils ont accès (Figure 6.15).

User Login

Email address

Password

Log in

FIGURE 6.15 – Page de connexion

Ajouter un élève

Le Admin de l'école a l'accé d'ajouter l'élève et remplir la formulaire (Figure 6.16).

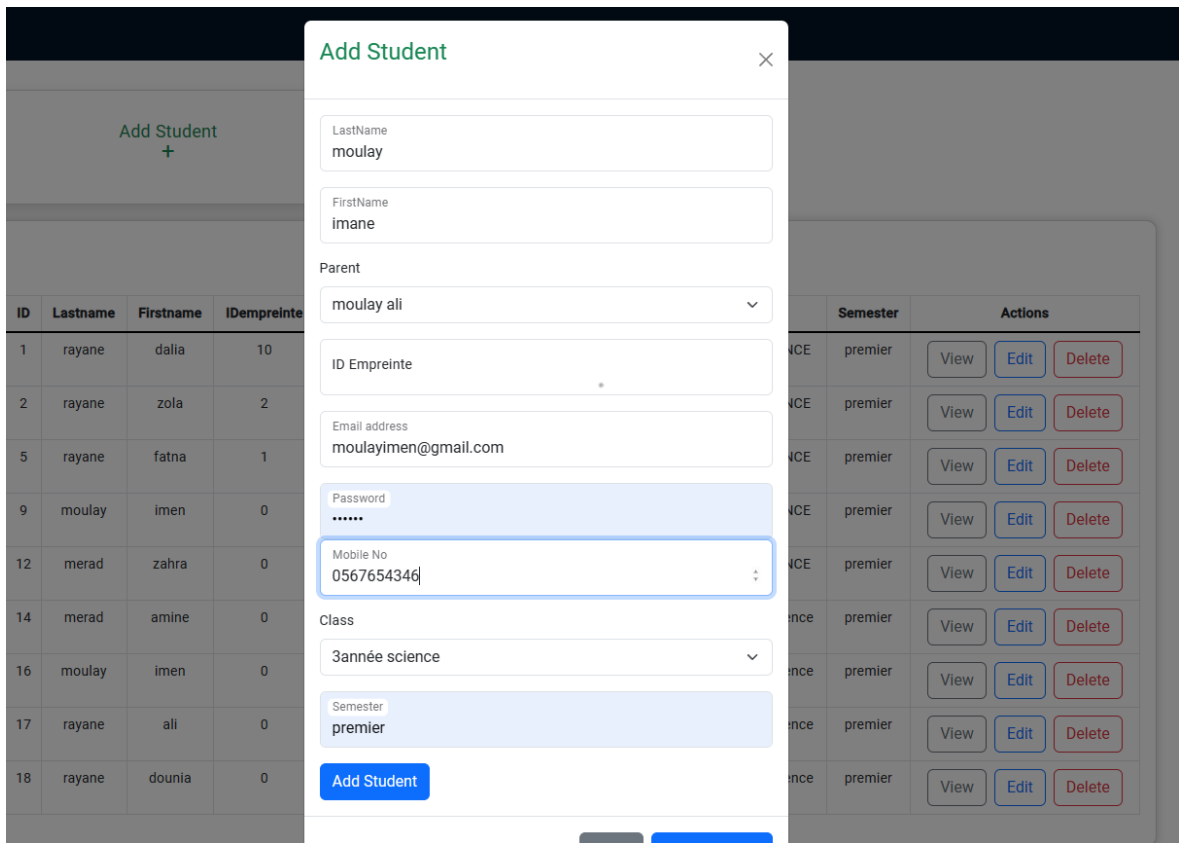


FIGURE 6.16 – Page d’ajouter l’élève

Modifier l’élève

Le Admin de l’école a l’accé de modifier l’élève et remplir la formulaire (Figure 6.17).

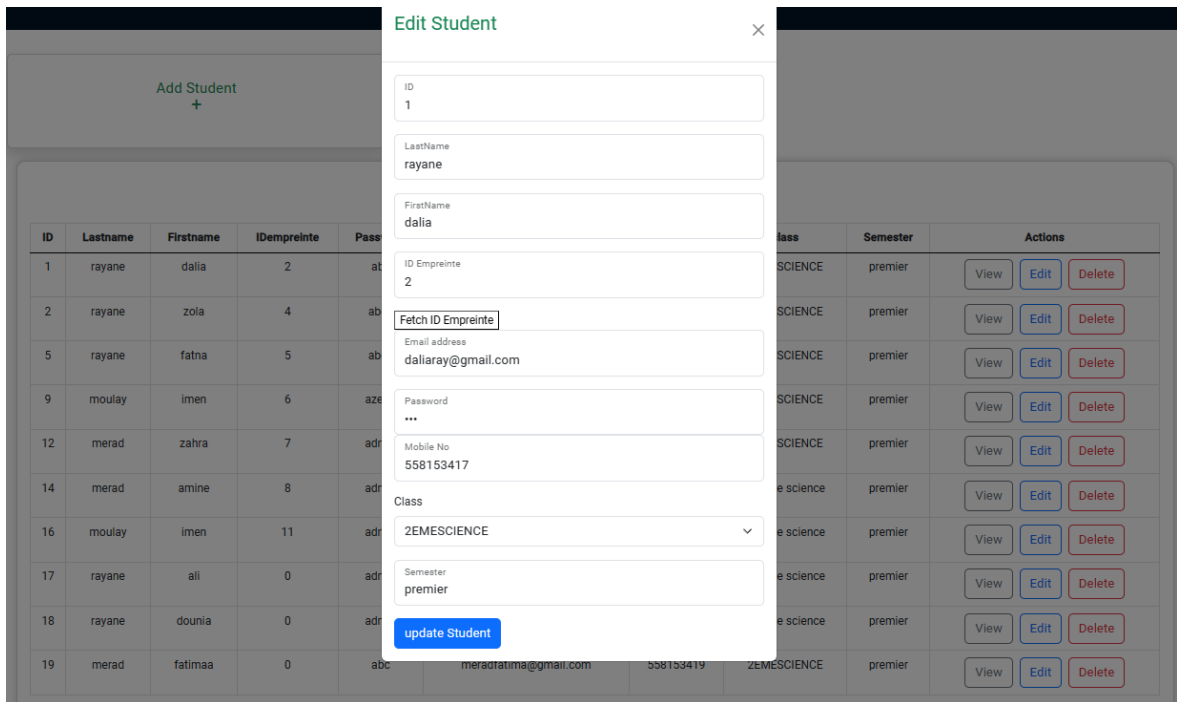


FIGURE 6.17 – Page de modifier l’élève

Voir un élève

Le Admin de l’école a l’accé de voir les informations l’élève (Figure 6.18).

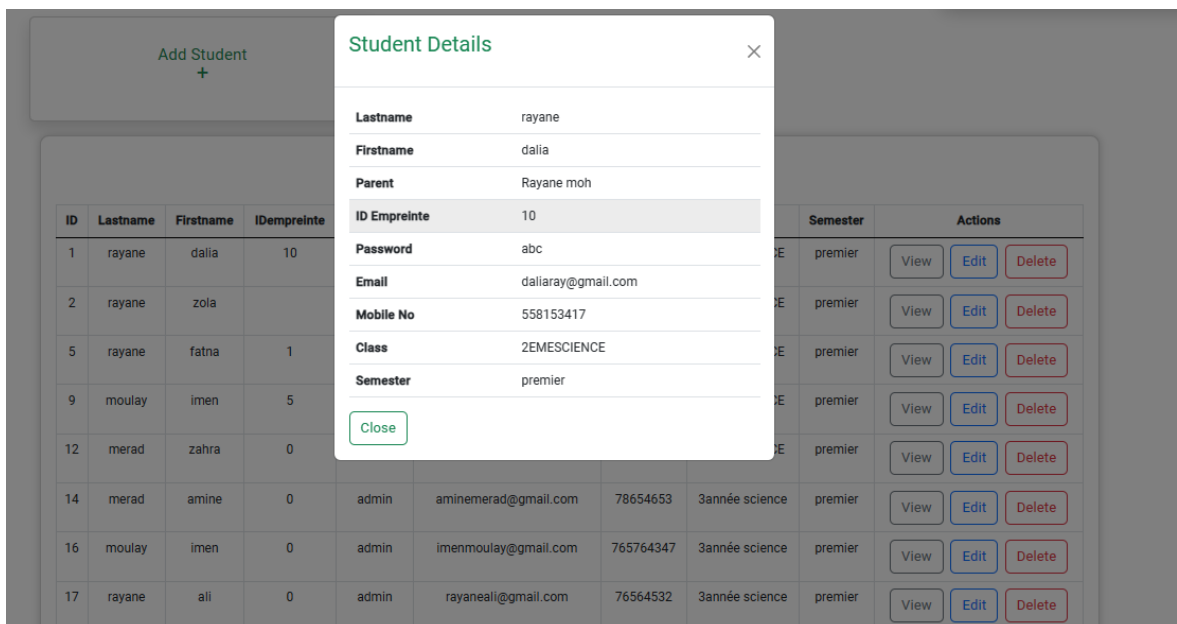


FIGURE 6.18 – Page de voir les informations de l’élève

Recherche un élève

Le Admin de l’école a l’accé de rechercher les élèves par classe (Figure 6.19).

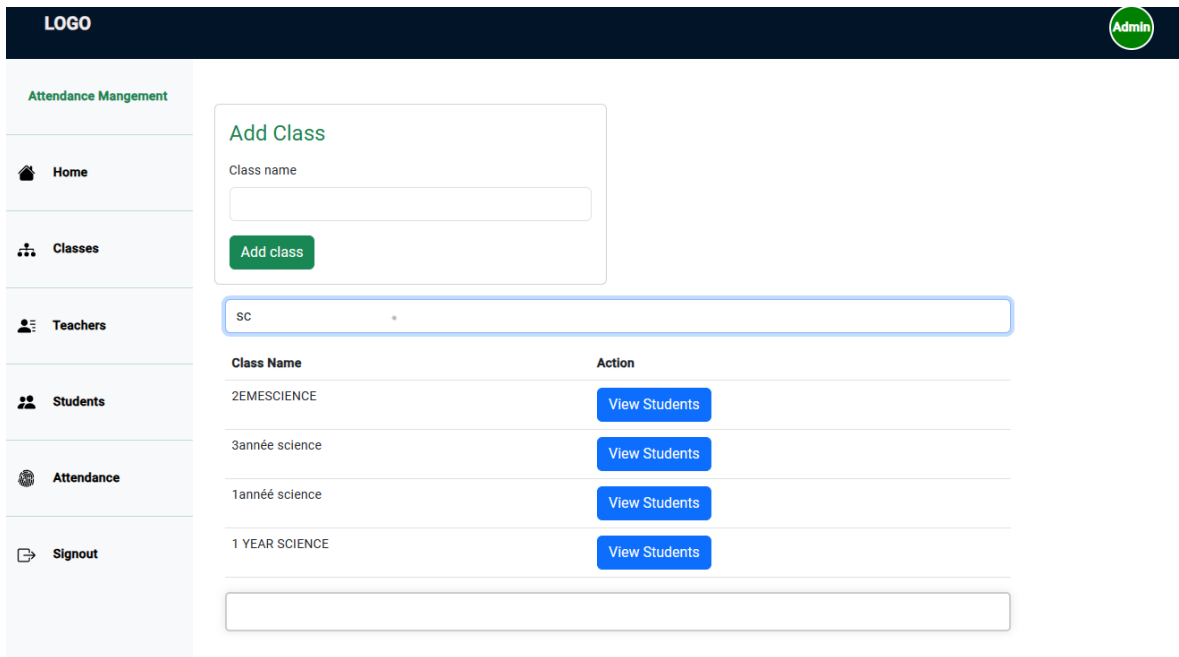


FIGURE 6.19 – Page de rechercher les élèves par classe

Ajouter enseignant

Le Admin de l'école a l'accé d'ajouter enseignant et remplir son formulaire (Figure 6.20).

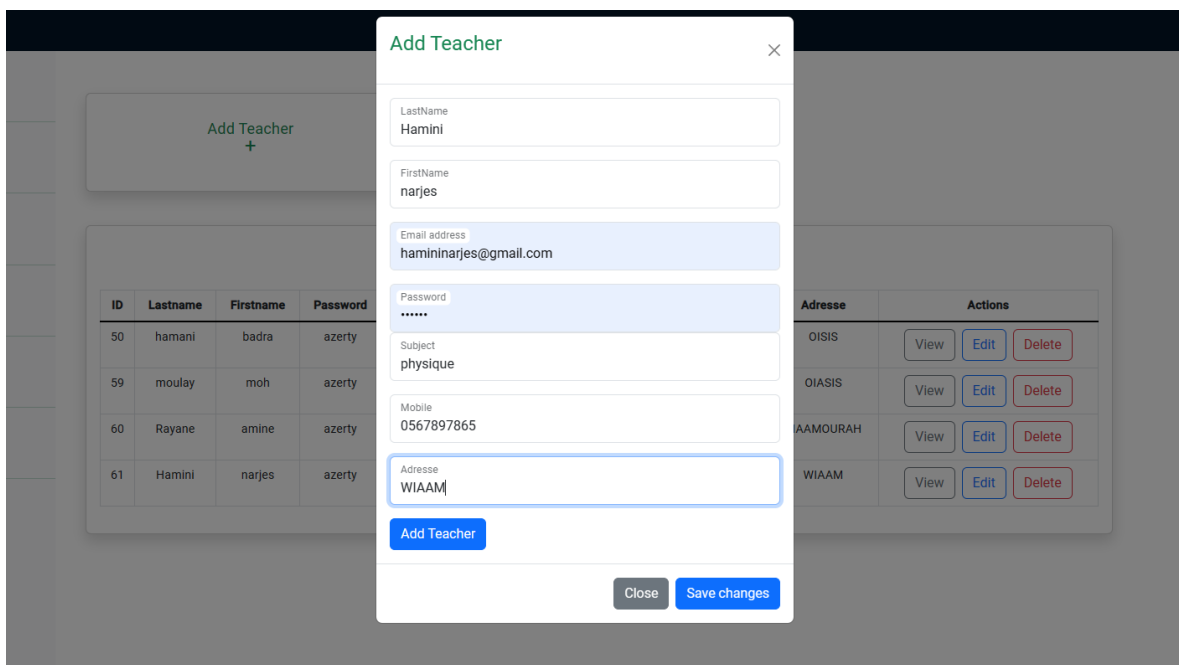


FIGURE 6.20 – Page d'ajouter enseignant

ID	Lastname	Firstname	Password	Email	Subject	Mobile No	Adresse	Actions
50	hamani	badra	azerty	hamanibadra@gmail.com	frensh	76587342	OISIS	View Edit Delete
59	moulay	moh	azerty	mohmoulay@gmail.com	mathématique	74893789	OIASIS	View Edit Delete
60	Rayane	amine	azerty	rayaneamine@gmail.com	science	67654567	MAAMOURAH	View Edit Delete
61	Hamini	narjes	azerty	hamininarjes@gmail.com	physique	674534567	WIAAM	View Edit Delete

FIGURE 6.21 – Page d’enseignant

Ajouter classe

Le Admin de l’école a l’accé d’ajouter les classes de l’établissement (Figure 6.22).

Class Name	Action

FIGURE 6.22 – Page d’ajouter les classes

Rechercher la présence des élèves dans classe

Le Admin de l’école a l’accé de rechercher la présence des élèves dans classe (Figure 6.23).

The screenshot shows the 'Attendance Management' interface. On the left is a sidebar with navigation options: Home, Classes, Teachers, Students, Attendance (selected), and Signout. The main content area is titled 'Attendance' and includes a 'Select Class:' dropdown menu with '2EMESCIENCE' selected. Below this is a table titled 'List of attendance by class' with the following data:

StudentID	Student	date	subject	status
1	rayane dalia	2023-09-01T21:24:48.525Z	physique	present
2	rayane zola	2023-09-01T21:24:48.525Z	physique	present
5	rayane fatna	2023-09-01T21:24:48.525Z	physique	present
9	moulay imen	2023-09-01T21:24:48.525Z	physique	present
12	merad zahra	2023-09-01T21:24:48.525Z	physique	absent
19	merad fatimaa	2023-09-01T21:24:48.525Z	physique	absent
1	rayane dalia	2023-09-01T23:22:09.157Z	frensh	absent
2	rayane zola	2023-09-01T23:22:09.157Z	frensh	present
5	rayane fatna	2023-09-01T23:22:09.157Z	frensh	absent
9	moulay imen	2023-09-01T23:22:09.157Z	frensh	absent
12	merad zahra	2023-09-01T23:22:09.157Z	frensh	absent
19	merad fatimaa	2023-09-01T23:22:09.157Z	frensh	present

FIGURE 6.23 – Page de rechercher la présence des élèves dans la classe

Voir la présence de l'élève dans l'école

Le Admin a l'accé de voir la présence de l'élèves dans l'école (Figure 6.24).

The screenshot shows the 'Attendance' page. The sidebar is the same as in Figure 6.23. The main content area is titled 'Attendance' and includes a 'Select Class:' dropdown menu with '- Select Class -' selected. Below this is a table titled 'List of attendance by class' with the following headers: StudentID, Student, date, subject, status. Below this table is a section titled 'Attendance in school by student' with an input field for 'Enter Student Name:' containing 'rayane dalia' and a 'Search' button. Below the search button is a table with the following data:

StudentID	Student	Date	Status
1	rayane dalia	2023-09-03 00:49:42	present

FIGURE 6.24 – Page de voir la présence de l'élève dans l'école

L'enseignant marque la présence de sa classe

Le Enseignant a l'accé de voir et marquer la présence des l'élèves de sa classe (Figure 6.25).

Mark Attendance

Class Subject: **mathématique**

Select Class:


2EMESCIENCE		
Last Name	First Name	Status
rayane	dalia	Present ✓ <input type="checkbox"/>
rayane	zola	<input type="checkbox"/> Absent ✗
rayane	fatna	Present ✓ <input type="checkbox"/>
moulay	imen	Present ✓ <input type="checkbox"/>
merad	zahra	<input type="checkbox"/> Absent ✗

[Submit Attendance](#)

FIGURE 6.25 – Page de l’enseignant marque la présence de sa classe

La page de parent

Le Parent a l’accé de voir l’assiuité de ses enfants (Figure 6.26).

LOGO


- merad zahra
- merad amine
- merad fatimaa

Child Details
First Name: amine
Last Name: merad

List of attendance by Child

StudentID	date	subject	classe	status
14	2023-08-31T23:02:19.163Z	anglais	3année science	absent
14	2023-09-01T19:59:57.477Z	mathématique	3année science	present
14	2023-09-06T02:12:36.061Z	frensh	3année science	present
14	2023-09-06T02:13:12.572Z	science	3année science	absent

List of attendance by Child in school

StudentID	Student	Date	status
14	merad amine	2023-09-03 00:51:26	present

FIGURE 6.26 – Page de voir l’assiduité de les enfants par leur parent