esprit
Se former autrement
HONORIS UNITED UNIVERSITIES

2023 / 2024

SPÉCIALITÉ :

Systèmes et logiciels embarqués ambiants et mobile

Système de gestion des absences des étudiants

Réalisé par : EZZAHI Sayf Eddine

Encadré par :

Encadrant ESPRIT :

Mr KHEDHIRI Imed

Encadrant Entreprise :

Mme REJEB Najla

esprit Tech

EUR-ACE® Délivrée par la Commission des Titres d'Ingénieur
CONFÉRENCE DES GRANDES ÉCOLES
cdio
AUF UNIVERSITAIRE FRANCOPHONIE IFGU

Encadrante professionnelle Mme REJEB Najla

Signature

Encadrant académique Mr KHEDHIRI Imed

Signature

Dédicaces

Je souhaite dédier ce travail à ma famille avec un profond sentiment de respect, d'amour et de reconnaissance pour avoir guidé mon éducation et créé les conditions propices à mon épanouissement académique.

Également, je dédie ce travail à mes amis, mes frères et ma sœur qui ont toujours été présents pour moi, me prodiguant réconfort et soutien inconditionnel tout au long de ce parcours.

Ezzahi Sayf Eddine.

Remerciements

Je tiens à exprimer ma profonde gratitude envers toutes les personnes qui ont contribué à la réalisation de ce travail et à qui je dédie ces lignes de remerciement.

Tout d'abord, je souhaite adresser mes plus sincères remerciements à toute l'équipe de la Recherche-Développement- Innovation RDI pour leur accueil chaleureux pendant mon stage, particulièrement à Mme REJEB Najla pour son encadrement exemplaire, son soutien constant, ses conseils avisés et sa disponibilité précieuse.

Je tiens également à exprimer ma reconnaissance envers M. KHEDHIRI Imed, mon encadrant à l'école Supérieure Privée d'Ingénierie et de Technologies (ESPRIT), pour sa présence attentive, son encadrement éclairé, ses conseils pertinents et ses remarques constructives qui ont enrichi mon travail.

Mes remerciements s'étendent également aux membres du jury qui ont accepté de consacrer leur temps et leurs compétences pour évaluer et enrichir mon projet.

Enfin, je souhaite exprimer ma gratitude envers mes proches, dont le soutien indéfectible et les encouragements ont été essentiels dans la réalisation de ce travail.

Résumé

Notre projet consiste à concevoir et déployer un système de gestion des présences des étudiants utilisant l'Internet des objets (IoT), l'intelligence artificielle (IA) et des applications Web. Nous intégrons des technologies telles que des puces RFID, des microcontrôleurs ESP32 et des nano ordinateurs Raspberry Pi pour créer des réseaux de capteurs dans les salles de classe. Chaque étudiant reçoit une carte RFID et enregistre sa présence via un lecteur de carte à l'entrée de la salle. Les données sont transmises en temps réel à une plateforme web centrale, permettant aux managers de suivre les présences et de générer des rapports. L'application web offre des fonctionnalités avancées telles que la gestion des plannings, la notification d'absence et la génération de statistiques.

Ce système répond au besoin de gestion des présences dans l'enseignement supérieur tout en offrant une expérience utilisateur intuitive.

.

Abstract

Our project consists of designing and deploying a student presence management system using the Internet of Things (IoT), artificial intelligence (AI) and web applications. We integrate technologies such as RFID chips, ESP32 microcontrollers and Raspberry Pi nano computers to create sensor networks in classrooms. Each student receives an RFID card and records his presence via a card reader at the entrance of the room. The data is transmitted in real time to a central web platform, allowing managers to track attendance and generate reports. The web application offers advanced features such as schedule management, absence notification and statistics generation.

This system meets the need for attendance management in higher education while providing an intuitive user experience.

Table des matières

Introduction générale…….. 1

Chapitre 1 : Présentation de l'organisme d'accueil et du contexte du stage……... 3

I. **Introduction…… 3**

II. **Présentation de l'organisme d'accueil…… 3**

1. **Structure…… 3**

2. **Valeurs…… 4**

III. **Présentation du projet…….. 5**

1. **Contexte initial…….. 5**

2. **Sujet de stage…… 6**

3. **Problématique……... 6**

IV. **Etude de l'existant…… 6**

1. **Description de l'existant……...……... 7**

1.1 **PowerSchool…….. 7**

1.2 **Class 123…… 8**

2. **Critique de l'existant…… 8**

V. **Solution proposée…….. 9**

VI. **Choix méthodologique……..10**

1. **Méthodologie de développement…… 10**

2. **Choix de la méthodologie…… .11**

3. **Méthodologie agile Scrum……12**

4. **Les Phases de Scrum…… 15**

4.1 **La planification de la release……... 15**

4.2 **La planification de sprint……..……15**

4.3 **Le sprint……..……16**

4.4 **La rétrospective de sprint……..……16**

4.5 **La revue du sprint……..……16**

VII. **Conclusion…… 16**

Chapitre 2 : Analyse et étude globale du projet……18

Sprint 0 : Etude et analyse globale du projet…… 18

I. **Introduction…….. 18**

II. **Analyse des besoins…….. 18**

1. **Les utilisateurs…….. 18**
2. **Besoins fonctionnels……..19**
3. **Besoins non fonctionnels…….. 20**
4. **Planification des sprints……... 20**

III. **Modélisation et architecture globale……... 21**

1. **Diagramme de cas d'utilisation globale……..22**
2. **Diagramme de classe globale……... 23**
3. **L'architecture globale du système……... 24**

IV. **Conclusion…… 24**

Chapitre 3 : Etude et analyse globale de la partie développement web et Mobile……. 26

Sprint 1 : Partie web……26

I. **Introduction……...26**

II. **Analyse des besoins……...26**

1. **Besoins fonctionnels……..26**
2. **Diagramme de cas d'utilisation……... 28**
3. **Diagramme de séquence pour la partie répondre aux réclamations……29**
4. **Backlog de la partie développement……30**

III. **Environnement de travail……30**

1. **Environnement de développement……..32**
2. **Langages de programmation……32**

IV. **Réalisation du sprint 1……..33**

1. **Les interfaces de la partie web……33**

V. **Conclusion……38**

Sprint 2 : Partie Mobile……..39

I. **Introduction……...39**

II. Analyse des besoins……...39

III. **Environnement de travail…….39**

IV. Réalisation du sprint 2……..40

V. **Conclusion……. 43**

Chapitre 4 : Etude et analyse globale de la partie embarquée……. 45

Sprint 3 : Partie embarquée…….45

I. **Introduction…….. 45**

II. **Analyse des besoins……...45**

1. **Diagramme de cas de transition…….. 45**

2. **Principe de fonctionnement des nœuds…….. 46**

III. **Environnement de travail…… 48**

1. **Environnement de développement matériel……...48**

1.1 **Etudes Comparatives des cartes……..48**

1.1.1 **Choix des cartes…… 48**

1.1.2 **Choix des capteurs et actionneurs……... 50**

2. **Environnement de développement logiciel……52**

2.1 **Choix des logiciels……52**

2.2 **Choix du système d'exploitation……..53**

IV. **Réalisation du sprint 3……..54**

1. **Montage nœud de capteurs……..……54**

2. **La maquette du projet……..55**

V. **Conclusion……..56**

Chapitre 5 : Etude et analyse globale de la partie Intelligence Artificielle……..58

Sprint 4 : partie Intelligence Artificielle……58

I. **Introduction……...58**

II. **Préparation et Prétraitement des Données……58**

1. **Les tâches …… 58**

2. **L'objectif……58**

III. **Environnement de travail……59**

1. **Environnement de développement logiciel……59**

2. **Implémentation des Modèles d'Apprentissage Automatique. ..……..…..59**

IV. **Optimisation des Modèles....……..……61**

V. **L'Intégration……..62**

1. **Création de l'API Flask…….62**

2. **Tests avec Flask……..…….62**

VI. **Conclusion……. 63**

Conclusion générale……..64

Nétographie……... 66

Table des figures

Figure 1 : PowerSchool……. 7
Figure 2 : Class 123…… 8
Figure 3 : Les 12 principes des méthodes agiles……... 10
Figure 4 : La méthode SCRUM…….. 12
Figure 5 : Principe de fonctionnement du système……19
Figure 6 : Diagramme des cas d'utilisation globale…… 22
Figure 7 : Diagramme de classe globale……23
Figure 8 : L'architecture globale du système…….. 24
Figure 9 : Diagramme de cas d'utilisation de la partie web…… 28
Figure 10 : Diagramme de séquence de la partie réponse sur les réclamations……. 29
Figure 11 : Logo Nest JS……31
Figure 12 : Logo React JS……...31
Figure 13 : Interface de l'authentification……... 33
Figure 14 : Interface de connexion……. 34
Figure 15 : Interface d'accueil de l'administrateur…….. 35
Figure 16 : Interface de la consultation de présence…….. 35
Figure 17 : Espace de modification de l'absence…… 36
Figure 18 : Interface d'accueil pour l'étudiant……36
Figure 19 : Interface de réclamation……... 37
Figure 20 : Interface de consultation des réclamations…….. 37
Figure 21 : Android Studio…… 40
Figure 22 : Interface de l'authentification……... 41
Figure 23 : Interface d'accueil pour l'étudiant……41
Figure 24 :Interface de réclamation…… 42
Figure 25 : Diagramme d'état de transition……45
Figure 26 : Communication MQTT ……...47
Figure 27 : Exemple d'une requête reçue……48
Figure 28 : Carte Raspberry pi 4……49
Figure 29 : Carte Esp32…….. 49
Figure 30 : Capteur RFID-RC522…….. 50
Figure 31 : Buzzer…….. 51
Figure 32 : Led……51
Figure 33 : Module SD……... 51
Figure 34 : Arduino IDE……52
Figure 35 : RealVNC Viewer……. 52

Figure 36 : Raspbian……..……..53
Figure 37 : Montage nœud de capteurs……...54
Figure 38 : La maquette du projet…….. 55
Figure 39 : Flask……..56

Table des tableaux

Tableau 1 : Tableau comparatif entre les méthodes SCRUM, Kanban et Scrumban……. 11
Tableau 2 : Backlog produit de la partie web……. 29
Tableau 3 : Tableau comparatif sur les cartes électroniques……...49

Introduction générale

Dans le cadre de la croissance technologique et de la numérisation de notre société, les usages de l'Internet des objets (IoT) et de l'intelligence artificielle (IA) occupent une place de plus en plus importante dans différents secteurs, y compris celui de l'éducation. Dans ce rapport, nous exposons le projet de création d'une application web et IoT visant à gérer la présence des étudiants dans une université. Ce projet fait partie intégrante de notre programme de formation en ingénierie embarquée à l'école supérieure privée d'ingénierie et de technologie ESPRIT Tunisie.

Dans les établissements d'enseignement supérieur, le suivi des présences des étudiants est un élément clé de la gestion académique. Cependant, les méthodes classiques de suivi des présences, comme les feuilles de présence papier, peuvent être sujettes à des erreurs et souvent inefficaces. L'utilisation des technologies IoT propose une solution novatrice pour automatiser et améliorer ce processus.

Le but principal de ce projet est de créer et de mettre au point une solution globale pour la gestion des présences des étudiants, en exploitant des technologies IoT comme les dispositifs RFID (Radio-Frequency Identification) et les microcontrôleurs ESP32, ainsi qu'une interface utilisateur conviviale via une application web. Cette solution offre une surveillance précise, en temps réel et fiable des présences.

La mise en place d'une solution AIoT pour gérer la présence des étudiants offre de nombreux bénéfices importants. Non seulement elle facilite l'automatisation du suivi des présences, mais elle fournit également des informations précieuses pour l'analyse et l'amélioration de la fréquentation des cours. En outre, cela permettra d'améliorer l'expérience des étudiants et des enseignants en proposant une solution contemporaine et performante.

Chapitre 1

Chapitre 1 : Présentation de l'organisme d'accueil et du contexte du stage

I. Introduction :

Dans ce chapitre, on débute par une introduction générale au projet, puis on examine et évalue de manière critique la situation actuelle. Finalement, nous aborderons les solutions suggérées.

II. Présentation de l'organisme d'accueil :

Créée en 2010, ESPRIT-Tech est l'organisation dédiée à la recherche, au développement et à l'innovation (RDI) au sein d'ESPRIT. Depuis la mise en oeuvre des programmes de formation, l'activité RDI chez ESPRIT est devenue une partie importante des décisions stratégiques prises par l'organisation. ESPRIT-Tech collabore avec les parties prenantes dans les domaines pertinents pour développer des stratégies et des orientations futures pour les activités de RDI. De plus, il se concentre sur la recherche appliquée et l'innovation, tout en reconnaissant l'importance de la recherche fondamentale[1].

1. Structure :

ESPRIT-Tech est une société de recherche avec un directeur qui gère toutes les dimensions stratégiques et opérationnelles.

Afin de réaliser sa mission, ESPRIT-Tech a instauré une structure adaptée. Elles sont désignées sous le nom d'équipes et sont formées afin de mener à bien des projets liés à la RDI sur des thématiques et des sujets de recherche. Les élèves peuvent prendre part à ce projet en concrétisant leurs projets de fin d'études.

En décembre 2015, ESPRIT-Tech a mis en place un comité spécialisé dans la recherche : le C-RDI (Comité RDI), qui était composé de tous les responsables d'équipe. Il a pour mission de développer la recherche et de

garantir la qualité de la production. En général, cela se produit en moyenne chaque mois.

Afin de rendre le comité plus efficace, il est équipé d'un secrétariat (SC-RDI) composé de 5 membres qui se réunissent régulièrement. Il vise à résoudre les problèmes en se basant sur les décisions prises par le C-RDI.[1]

2. Valeurs :

- Rigueur et intégrité scientifique
- Collaboration et partage des connaissances
- Respect
- Engagement et responsabilité sociale
- Inclusion
- Transparence
- Développement durable

[1] https://espit.tn/rdi/espri-tech

III. Présentation du projet :

1. Contexte initial :

À l'heure actuelle, les établissements scolaires font face à des difficultés dans la gestion manuelle des présences universitaires. Les approches classiques de suivi des absences peuvent être inefficaces et faibles, ce qui peut engendrer des lacunes dans la gestion des données. En outre, il est nécessaire que les enseignants et les administrateurs consacrent un temps et des efforts considérables à la tenue des registres de présence, ce qui peut engendrer de la frustration et une diminution de la productivité. Dans cette situation, l'emploi de technologies de pointe comme l'Internet des objets (IoT) et l'intelligence artificielle (IA) permet de moderniser et d'améliorer la gestion des absences des étudiants. En intégrant des appareils IoT dans les salles de classe et en élaborant des algorithmes d'intelligence artificielle pour analyser les données de présence, il est envisageable de concevoir un système automatisé qui repérera et enregistrera les absences en temps réel, tout en proposant des fonctionnalités avancées comme la notification automatique des absences non autorisées et l'intégration avec les systèmes d'information. de l'établissement. L'objectif de ce projet est de relever ces défis en offrant une solution novatrice et performante afin d'améliorer la gestion des absences des étudiants dans les collèges et lycées.[2]

.

[2]https://www.classter.com/fr/blog/edtech-fr/le-systeme-de-gestion-des-presences-rfid-et-ses-avantages-pour-lenvironnement-scolaire/

2. Sujet de stage :

Dans les établissements éducatifs, de nombreux temps sont consacrés aux procédures de notation manuelle.

Dans cette présentation, nous offrons un système qui offre la possibilité d'évaluer automatiquement l'assiduité à l'école en utilisant des cartes RFID. Effectivement, chaque élève autorisé recevra une carte RFID personnalisée contenant ses informations personnelles. Il s'agit d'un circuit intégré qui enregistre ces informations en modulant et démodulant les signaux radiofréquence émis. Ainsi, les informations enregistrées sur cette carte représentent les informations d'identification uniques de cette personne. Dès l'installation du tag devant le lecteur RFID, les informations qu'il renferme seront consultées et la présence de l'élève sera enregistrée.

Notre système comprend une solution AIoT ainsi qu'un site web qui permettra à la direction de suivre les absences des étudiants et de gérer les absences justifiées. De plus, nous avons pris en compte les besoins de l'enseignant et des étudiants pour qu'ils puissent consulter ces absences.

qui peuvent consulter ces absences.

3. Problématique :

Après des études approfondies, il est évident que ces dernières années ont été caractérisées par une forte hausse du taux d'absentéisme dans les écoles. La raison principale de cette augmentation est liée à des éléments humains, comme le manque de suivi efficace et la gestion manuelle des présences. Par conséquent, il est essentiel de trouver des solutions automatisées et à base de technologies avancées afin de résoudre cette problématique de plus en plus importante.

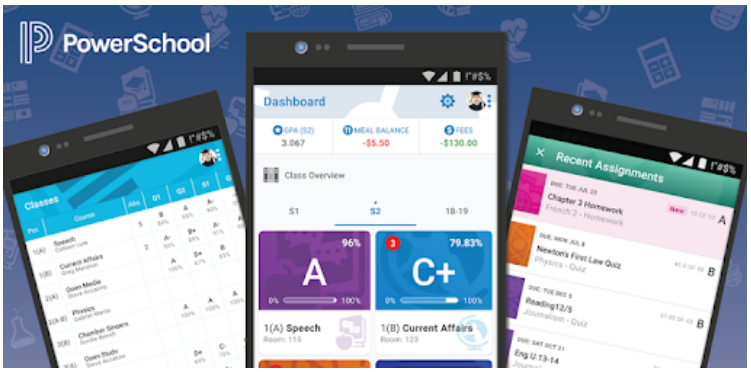
IV. Etude de l’existant :

Dans cette partie, je vais présenter les différentes options actuellement offertes pour gérer les absences des étudiants. Par la suite, j'effectuerai une comparaison des diverses technologies employées dans ces solutions.

1. Description de l'existant :

Un défi majeur se pose pour de nombreux établissements éducatifs en ce qui concerne la gestion des absences des étudiants, qui reposent souvent sur des méthodes manuelles ou semi-automatiques. Ces systèmes classiques consistent en général à rassembler les données de présence par les enseignants, puis à les saisir dans un système centralisé. Ces techniques peuvent avoir des performances médiocres, être vulnérables aux erreurs et souvent manquer de fonctionnalités avancées telles que l'analyse prédictive ou les notifications en temps réel.

1.1 PowerSchool:



(Figure 1 : PowerSchool)

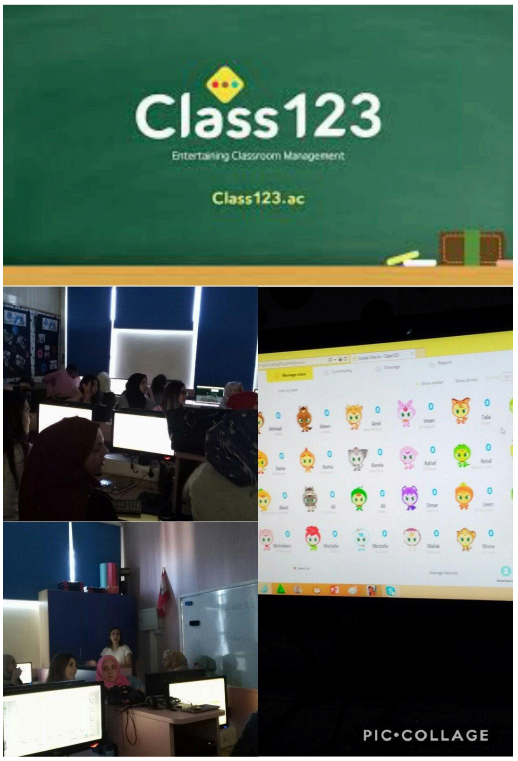
La plateforme cloud intégrée PowerSchool est utilisée par de nombreuses écoles et districts scolaires afin de gérer différents aspects de la scolarité, tels que la gestion des présences.

Grâce à elle, il est possible de surveiller les absences, d'envoyer des notifications aux parents et aux enseignants, et de produire des rapports sur les tendances d'absence.

PowerSchool propose également des modules pour évaluer, communiquer et gérer des comportements positifs.

[3] https://www.powerschool.com/company/

1.2 Class 123 :



(Figure 2: Class 123)

Class123 est une application mobile destinée à assister les enseignants dans le suivi de la présence des étudiants et la gestion de leur comportement en classe. L'application propose différentes options pour surveiller les absences, gérer les comportements et communiquer avec les parents.

De plus, elle offre la possibilité de représenter visuellement les données de présence et de produire des rapports détaillés.

2. Critique de l'existant :

Bien que des systèmes tels que PowerSchool et Class123 aient réalisé des progrès significatifs, il existe encore plusieurs désavantages dans les solutions actuelles :

- L'absence d'automatisation totale.

[4] https://www.radars-auto.com/info-radars/modele-radar/aximum-captor.php

- Étude de données restreinte.
- De nombreux systèmes ne sont pas complètement intégrés aux dispositifs IoT afin de permettre une détection automatique en temps réel.

V. Solution proposée :

Après une étude des diverses solutions disponibles sur le marché pour gérer les absences des étudiants, j'ai repéré les points faibles et les contraintes de ces solutions. En s'appuyant sur ces faiblesses, j'ai élaboré une solution novatrice qui exploite des technologies de pointe comme l'IoT et l'IA afin d'automatiser le suivi des présences et d'améliorer la gestion des absences dans les enseignants.

Je propose une solution qui permet de détecter automatiquement les absences en temps réel, en utilisant des capteurs IoT installés dans les salles de classe. La présence des étudiants est détectée de manière précise et non intrusive grâce à ces capteurs. Par la suite, des algorithmes d'intelligence artificielle étudient les données de présence afin de repérer les tendances d'absence et anticiper les absences à venir.

Outre la possibilité de détecter automatiquement les absences, ma solution propose une interface conviviale permettant de visualiser les informations de présence, des rapports et des statistiques détaillés pour une analyse approfondie. Il y a aussi un système de notification automatique qui permet d'alerter les enseignants et les étudiants en cas d'absence non autorisée.

En outre, ma solution offre la possibilité d'être intégrée au système d'information de l'établissement, garantissant ainsi une cohérence des données et une communication efficace entre les divers systèmes. Le système d'information de l'établissement peut synchroniser les données de présence, ce qui simplifie la gestion globale des étudiants.

VI. Choix méthodologique :

1. Méthodologie de développement :

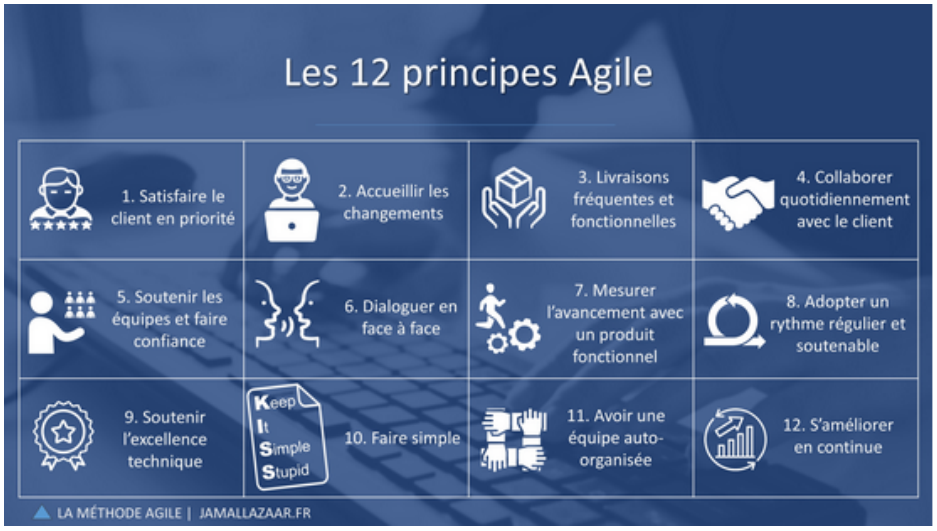
Pour faire face à la diversité des caractéristiques, il est primordial de choisir une approche de travail adaptée. Cela doit nous permettre d'observer de manière optimale l'avancement de notre projet, de proposer un projet qui répond aux exigences déjà établies tout en respectant les délais et la qualité du travail.

Méthodologie agile :

En organisant les projets en une série d'objectifs à atteindre, la méthode Agile simplifie la gestion des projets.

<u>Les principes :</u>

- Faire connaître la satisfaction du client..
- l'importance continue de l'excellence technique et d'une conception de qualité.
- Des versions fonctionnelles de l'application doivent être produites régulièrement..
- La facilité et la diminution des tâches superflues.
- Mettre en place des avancées constantes et pérennes dans le projet.
- L'optimisation et la réduction de la quantité de travail superflu.



(Figure 3 :Les 12 principes des méthodes agiles)

2. Choix de la méthodologie :

Une fois que j'ai choisi la méthodologie de travail agile, je me lance dans la recherche de la méthode appropriée pour mener à bien le projet.La facilité et la diminution des tâches superflues

La facilité et la diminution des tâches superflues J'ai effectué une étude comparative entre les méthodes SCRUM, Kanban, ShapeUp et Scrumban dans ce tableau.

(Tableau 1 : Tableau comparatif entre les méthodes SCRUM, Kanban et Scrumban)

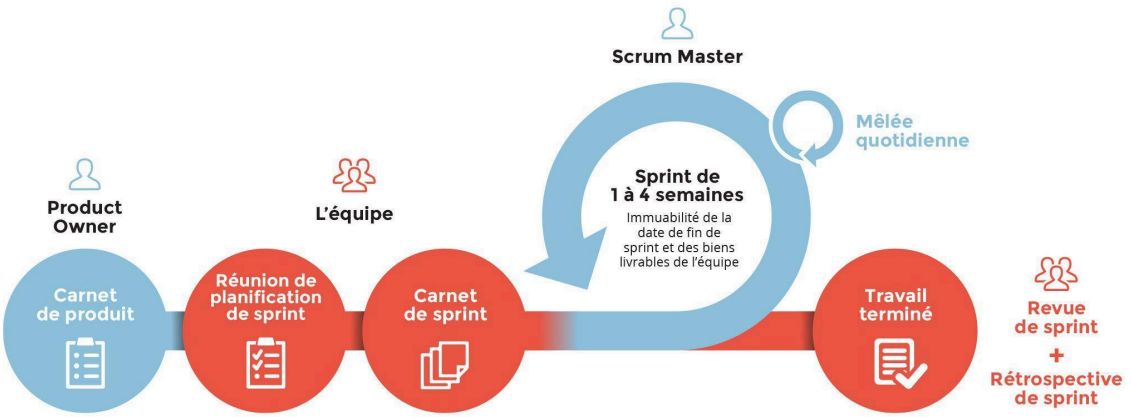
Caractéristiques	SCRUM	Kanban	Scrumban
Cadre de travail	Itératif et incrémental	Continu et évolutif	Hybride, combinant SCRUM et Kanban
Planification	Planification à court terme (sprints)	Aucune planification spécifique	Planification flexible
Rôles	Scrum Master, Product Owner, Équipe	Aucun rôle spécifique	Adaptation des rôles SCRUM
Itérations	Sprints	Aucune itération spécifique	Aucune itération spécifique
Priorisation des tâches	Basée sur la backlog et la planification des sprints	Limitée, basée sur les tâches en cours	Adaptée, basée sur les principes de Kanban
Gestion des tâches	Utilisation du backlog et des tâches assignées à chaque sprint	Utilisation d'un tableau Kanban avec des colonnes de flux de travail	Combinaison de backlog SCRUM et de flux de travail Kanban
Flexibilité	Moins flexible en raison de la structure de sprint	Très flexible, les tâches peuvent être ajoutées ou retirées à tout moment	Flexibilité moyenne, dépend de l'adaptation SCRUM-Kanban

Amélioration continue	Réunions de rétrospective à la fin de chaque sprint	Amélioration continue basée sur l'analyse des flux de travail	Amélioration continue intégrée dans la méthode
Visibilité	Clair, grâce aux réunions de planification et de rétrospective	Clair, grâce au tableau Kanban en temps réel	Clair, grâce à une combinaison de backlog SCRUM et de tableau Kanban
Complexité du processus	Moyenne à élevée en raison de la structure de sprint	Faible à moyenne, dépend du niveau de formalisme	Variable, dépend de l'adaptation SCRUM-Kanban
Adaptation aux changements	Élevée, les changements peuvent être pris en compte à chaque sprint	Élevée, les changements peuvent être intégrés en temps réel	Élevée, les changements peuvent être intégrés à tout moment

Pour mettre en œuvre mon projet, j'ai choisi la méthodologie Scrum car elle offre une solution qui peut s'ajuster aux évolutions et aux modifications des besoins. Elle réduit les risques de problèmes de conception.

3. Méthodologie agile Scrum :

Pendant la mise en œuvre de mon projet, j'ai choisi d'utiliser la méthodologie Scrum. Il s'agit d'une méthode de gestion de projet Agile. Il repose sur la division du projet en un ensemble d'itérations connues sous le nom



(Figure 4 : La méthode SCRUM)

Dans Scrum, la planification est réalisée en trois dimensions :

1. Planification du produit : La définition des fonctionnalités et des objectifs globaux du produit est une dimension importante de cette dimension. La création du Product Backlog consiste à établir une liste claire des fonctionnalités à développer, en fonction des besoins du client et des objectifs du projet..
2. Planification de la release : Après avoir établi le Product Backlog, les équipes Scrum élaborent des plans de livraison, c'est-à-dire des versions du produit qui seront livrées à des intervalles réguliers. Il est essentiel de définir les fonctionnalités du Product Backlog qui seront intégrées à chaque sortie et de définir un calendrier pour leur livraison.
3. Planification de l'itération : Finalement, la préparation de l'itération implique la sélection des éléments du Backlog de Produits pour être intégrés dans un Sprint particulier. Lors de la réunion de planification de Sprint, les équipes Scrum se réunissent afin de choisir les éléments les plus importants du Product Backlog et de déterminer les objectifs et les tâches spécifiques à accomplir pendant le Sprint.

- **Backlog de produit**

La liste des caractéristiques que l'application doit offrir est appelée le backlog du produit. Le nom, la description, un attribut qui évalue le budget de la fonctionnalité en points (les points peuvent être des jours ou des heures) et un attribut qui détermine la priorité de la fonctionnalité sont les caractéristiques qui identifient toutes les fonctionnalités.

- **Backlog du sprint**

On choisit les fonctionnalités du backlog du sprint parmi celles du backlog du produit qui seront mises en œuvre au cours d'un sprint. Par la suite, chaque fonctionnalité doit être divisée en une liste de tâches. Chaque tâche est accompagnée d'un titre, d'une description et d'une estimation du budget requis.

Les trois acteurs identifiés par Scrum :

- **Le Product Owner** (Directeur de Produit)
  Mme TEBER Faten, en tant que chef de projet, possède une expertise fonctionnelle et est capable de faire les choix nécessaires pour donner la priorité aux développements. Son rôle revêt une importance capitale, et le respect des règles du jeu est essentiel pour assurer le succès d'un projet agile.
- **Le Scrum Master**
  Mme REJEB Najla, encadrante de l'entreprise. En tant que membre de l'équipe, son rôle principal consiste à maximiser la productivité de l'équipe en l'encourageant à travailler de manière autonome et à s'améliorer en permanence. Il garantit également la mise en place adéquate de Scrum.
- **L'équipe** avec une taille réduite et qui assure le développement du produit sans se spécialiser dans les rôles. L'une des caractéristiques d'une équipe Scrum est d'être auto-organisée , ce qui signifie qu'il n'y a pas de hiérarchie.
  Il s'agit d'une rupture radicale avec les méthodes de gestion traditionnelles, qui mettent l'accent sur un contrôle centralisé, généralement représenté par le Chef de Projet (moi-même)..

4. Les Phases de Scrum :

4.1 La planification de la release :

Dans cette étape, il est nécessaire de déterminer quelles fonctionnalités du backlog du produit seront incluses dans la version suivante. La mise en place d'un calendrier pour la livraison de cette version est également requise.

4.2 La planification de sprint :

L'équipe sélectionne les éléments du backlog du produit à inclure dans le prochain sprint en se référant aux objectifs fixés pour le sprint et aux ressources disponibles lors de cette étape de planification de sprint.

4.3 Le sprint :

Cette étape de développement est la réelle, où l'équipe prend en charge les éléments sélectionnés lors de la planification du sprint. La durée du sprint est fixée, généralement entre deux et quatre semaines, pendant laquelle l'équipe développe un ajout qui pourrait être livré du produit.

4.4 La rétrospective de sprint :

Durant ce stade, une réunion d'équipe est organisée où les membres de l'équipe échangent sur les aspects qui ont été positifs lors du sprint, les aspects qui n'ont pas été positifs et les moyens de s'améliorer pour le prochain sprint. Cela permet à l'équipe de se faire une évaluation personnelle et de repérer les opportunités d'amélioration continu

4.5 La revue du sprint :

À l'issue de chaque sprint, l'équipe fait part aux parties prenantes du travail effectué lors de la réunion, communément connue sous le nom de revue de sprint. Cela donne aux acteurs impliqués la possibilité de faire part de leur opinion sur le travail réalisé et d'ajuster les priorités si nécessaire.[5]

VII. Conclusion :

Les méthodes agiles ne doivent pas être confrontées aux autres processus unifiés.

Chaque méthode est personnalisée en fonction de son propre contexte du projet.

La première condition de succès consiste à maîtriser et à intégrer la méthode dans le contexte du projet.

[5] https://scrumguides.org/scrum-guide.html

Chapitre 2

Chapitre 2 : Analyse et étude globale du projet

Sprint 0 : Etude et analyse globale du projet

I. Introduction :

Ce chapitre commencera par une analyse et une spécification des besoins, suivi de l'architecture globale du projet, puis de l'environnement de travail et d'une conclusion.

II. Analyse des besoins :

1. Les utilisateurs :

L'administrateur, cet acteur aura la possibilité de :

- Gérer les comptes et donner les rôles.
- surveiller et gérer les absences de tous les cours et de tous les étudiants..
- Gérer les absences justifiées.
- Visualiser les statistiques des absences.

L'enseignant, cet acteur aura la possibilité de :

- Consulter les absences de leurs étudiants .
- Réclamer à la direction.
- Visualiser les statistiques des absences.

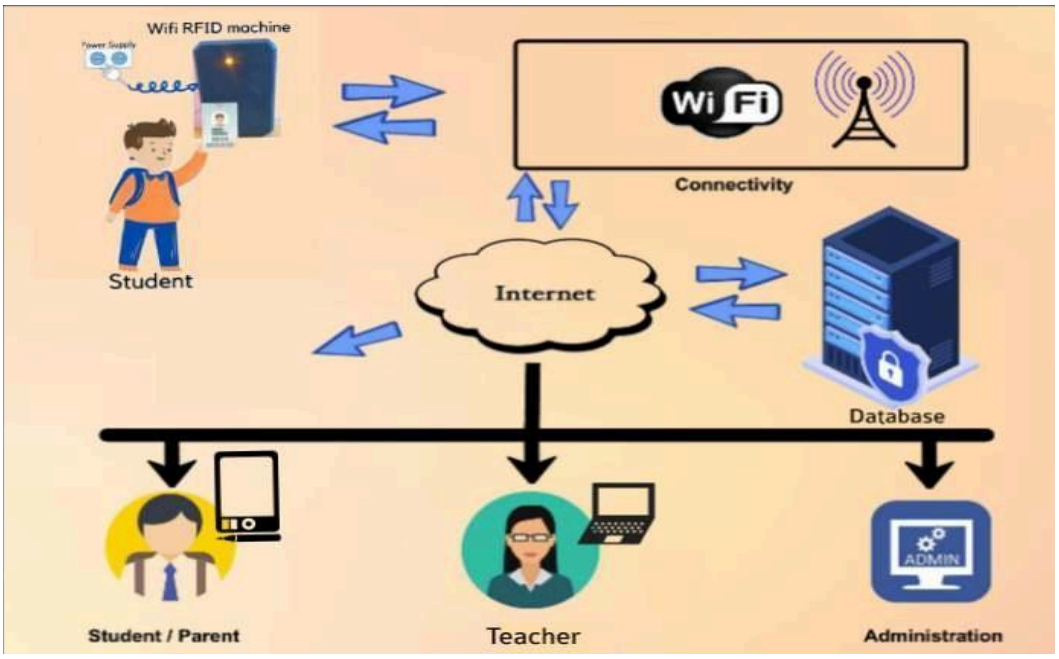
L'étudiant, cet acteur aura la possibilité de :

- Consulter ses absences.
- Réclamer à la direction.

2. Besoins fonctionnels :

Les besoins fonctionnels de notre système :

- S'authentifier.
- La détection automatique de la présence des étudiants dans les salles en utilisant des dispositifs IoT.
- collecter les données de présence des étudiants en temps réel et les stocker de manière sécurisée dans une base de données.
- Fournir un tableau de bord convivial permettant aux utilisateurs de visualiser facilement les données d'absence, ainsi que des rapports et des statistiques pour une analyse approfondie.
- Intégrer un système pour prendre compte des absences justifiées et permettre une intervention humaine en cas de circonstances particulières.
- Permettre la collecte de feedback des utilisateurs et planifier des mises à jour régulières.
- intégrer un système de notification automatique pour informer les enseignants et les étudiants en cas d'absence non autorisée.



(Figure 5 : Principe de fonctionnement du système)

3. Besoins non fonctionnels :

Les besoins non fonctionnels de notre projet de système AIoT de gestion des absences des étudiants incluent :

- **Sécurité des données :** Il est essentiel que l'application garantisse la confidentialité, l'intégrité et la disponibilité des données étudiées en mettant en place des mesures de sécurité solides.
- **Performances du système :** Il est primordial de garantir une expérience utilisateur fluide en assurant des temps de réponse rapides et une disponibilité élevée du système, même en cas de charge élevée ou de pics d'utilisation.
- **Facilité d'utilisation :**Faciliter l'adoption du système et améliorer l'efficacité opérationnelle en proposant une interface utilisateur conviviale pour les enseignants, les étudiants et les administrateurs.
- **Fiabilité et tolérance aux pannes :** Afin d'assurer la fiabilité et la résistance du système face aux pannes matérielles ou logicielles, il est nécessaire de mettre en place des dispositifs de sauvegarde et de récupération des informations.
- **Scalabilité :** Garantir la flexibilité du système pour faire face à une expansion future de l'établissement éducatif, en facilitant la gestion d'un nombre croissant d'étudiants, de salles de classe et de dispositifs IoT.

4. Planification des sprints :

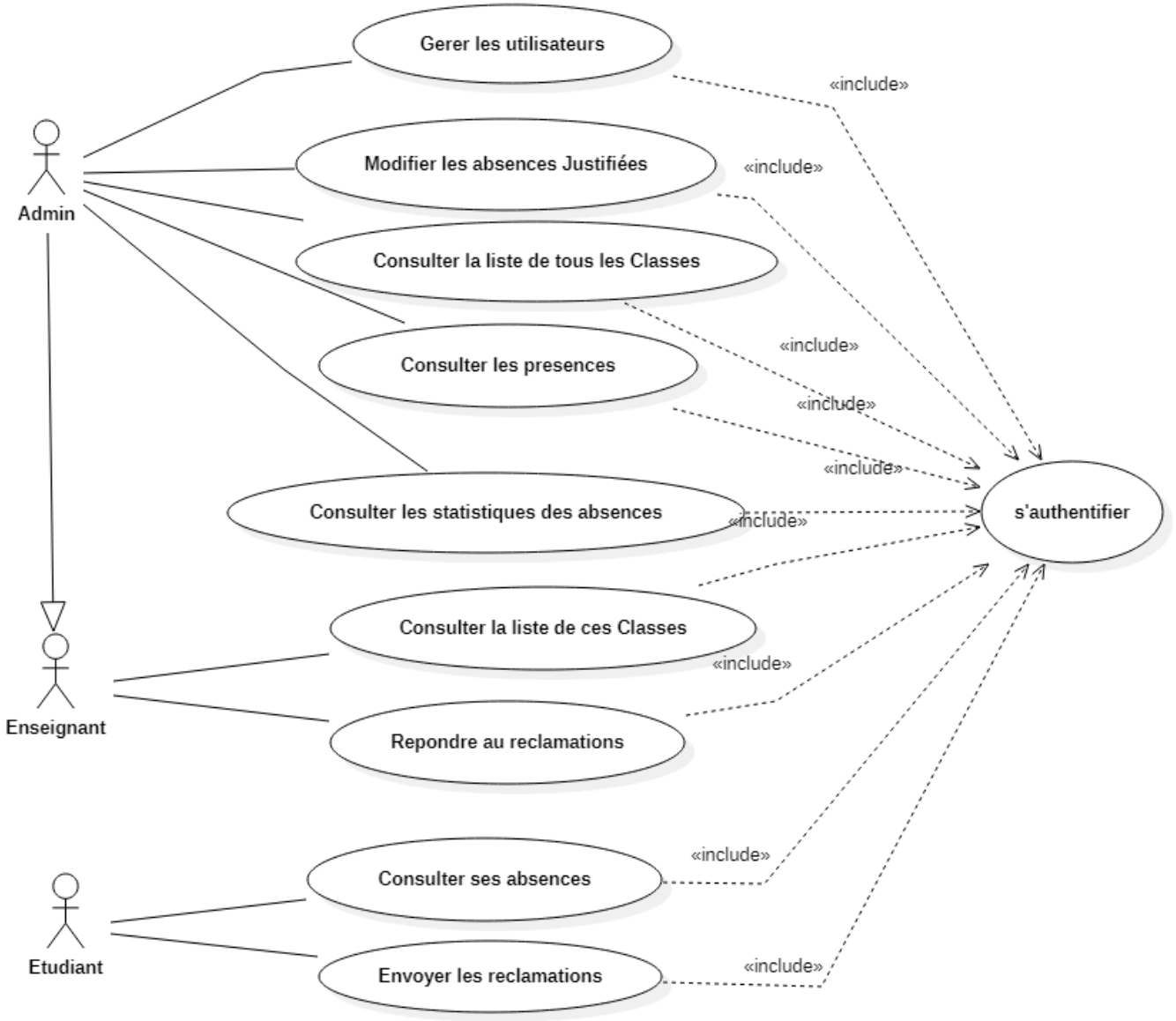
- **Sprint 1 :** partie web.
- **Sprint 2 :** partie Embarqué.
- **Sprint 3 :** partie Intelligence Artificielle.

III. Modélisation et architecture globale :

Ma solution est conçue en utilisant la notation UML (Language de Modélisation Unifié). Un langage standardisé de modélisation visuelle, appelé UML, est employé afin de concevoir, documenter et transmettre les divers aspects des systèmes logiciels.

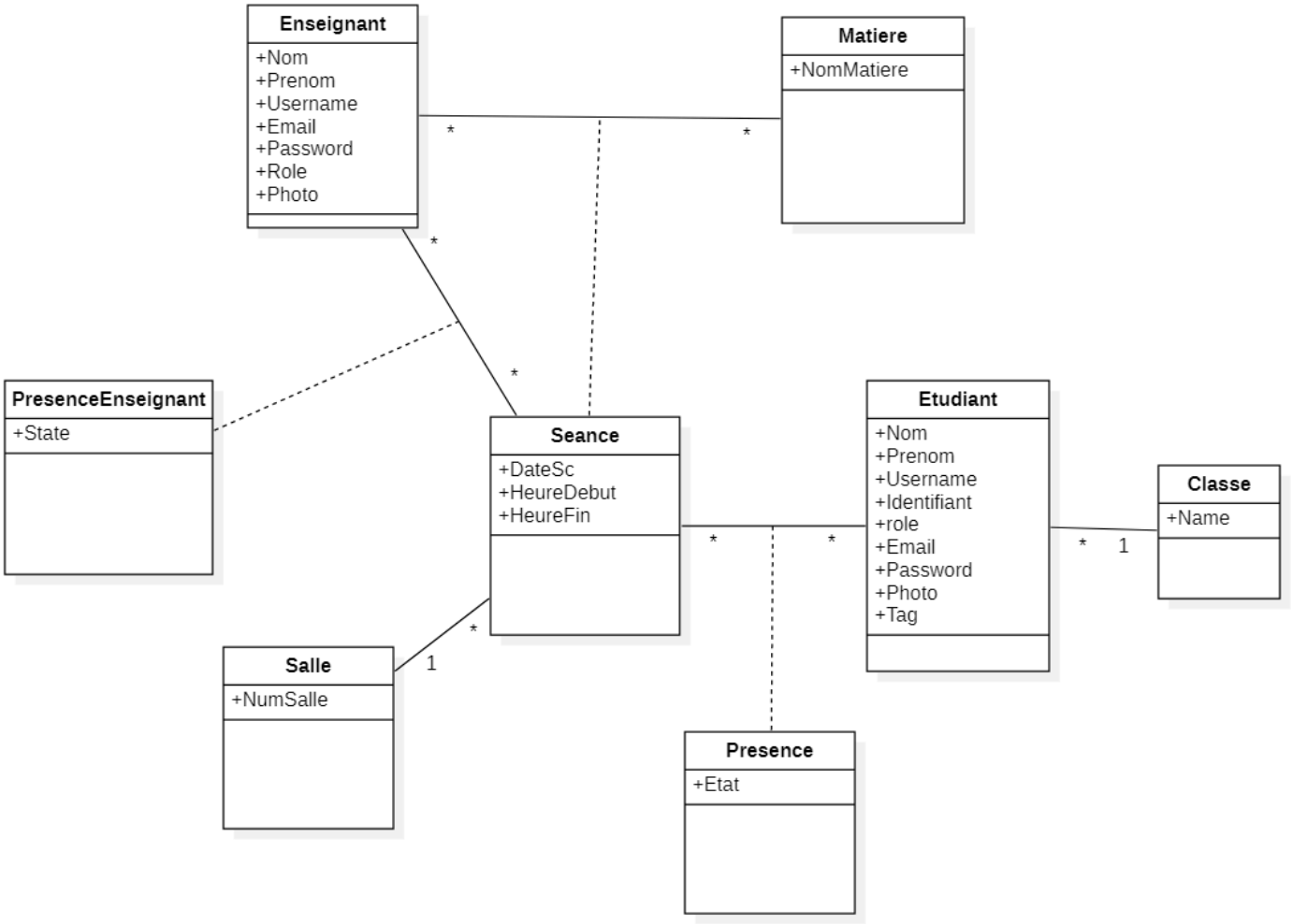
J'ai employé l'outil "StarUML" pour créer les divers diagrammes.

1. Diagramme de cas d'utilisation globale :



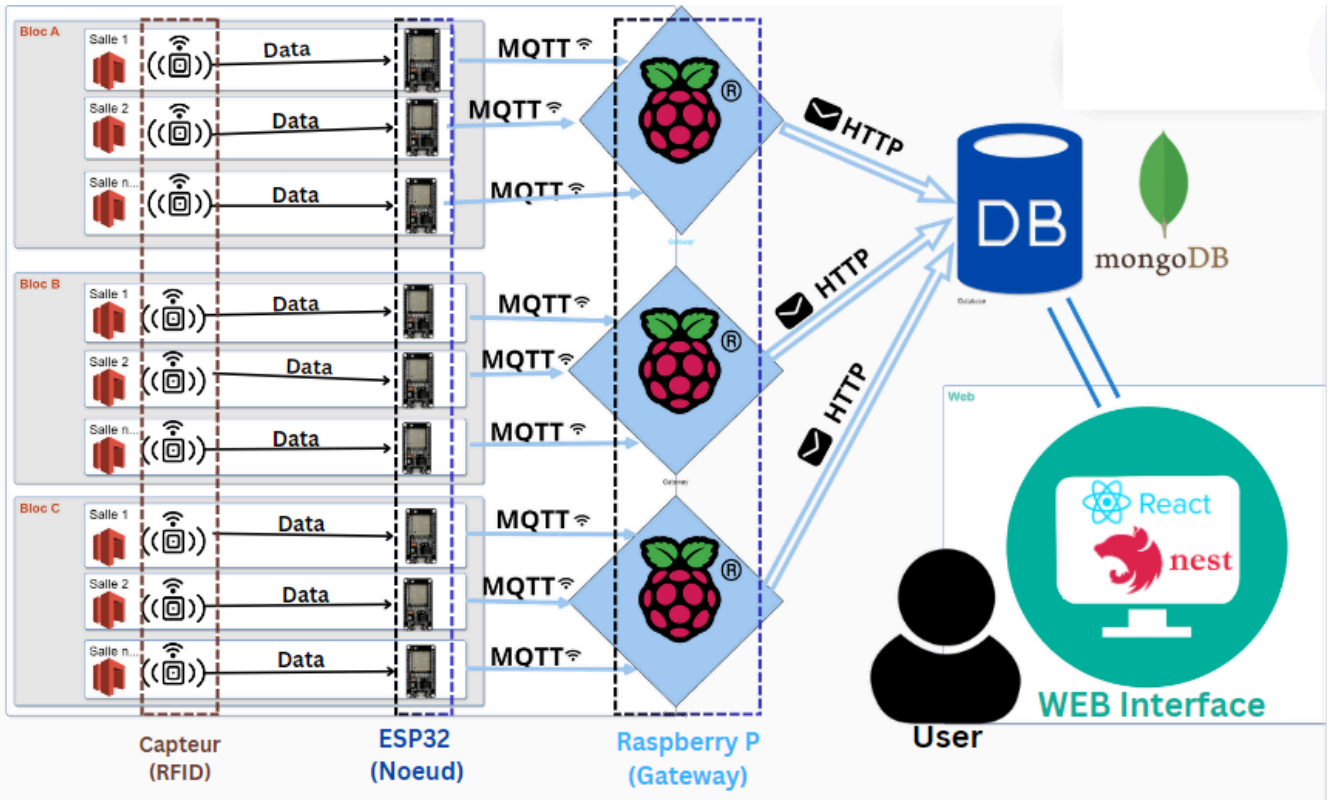
(Figure 6 : Diagramme des cas d'utilisation globale)

2. Diagramme de classe globale :



(Figure 7 : Diagramme de classe globale)

3. L'architecture globale du système :



(Figure 8 : L'architecture globale du système)

IV. Conclusion :

Dans ce chapitre, j'ai exposé les exigences fonctionnelles en utilisant le schéma des cas d'utilisation global.

Ensuite, j'ai exposé la représentation du diagramme de classe globale et j'ai finalement exposé l'architecture globale du système.

Chapitre 3

Chapitre 3 : Etude et analyse globale de la partie développement web et Mobile

Sprint 1 : Partie web

I. Introduction :

Dans ce chapitre, je vais exposer la première étape du sprint, à savoir la partie web. Je vais la décrire en abordant l'analyse des besoins, en incluant les spécifications, puis en présentant l'environnement de travail et en terminant.

II. Analyse des besoins :

L'élaboration de cette section nécessite tout d'abord de déterminer les divers rôles des utilisateurs qui sont impliqués dans notre application.

1. Besoins fonctionnels :

Les utilisateurs :

L'administrateur, cet acteur aura la possibilité de :

- Gérer les utilisateurs :
  - Ajouter des utilisateurs.
  - Supprimer des utilisateurs.
  - Afficher la liste des utilisateurs.
- Gérer les absences justifiées :
  - Consulter les absences.
  - Supprimer les absences.
  - Traiter les réclamations des étudiants.
- Visualiser les statistiques et les absences :
  - Les taux d'absences.
  - Les pourcentages d'absences par (Matières/ niveau / enseignants…).

- Le nombre des réclamations.

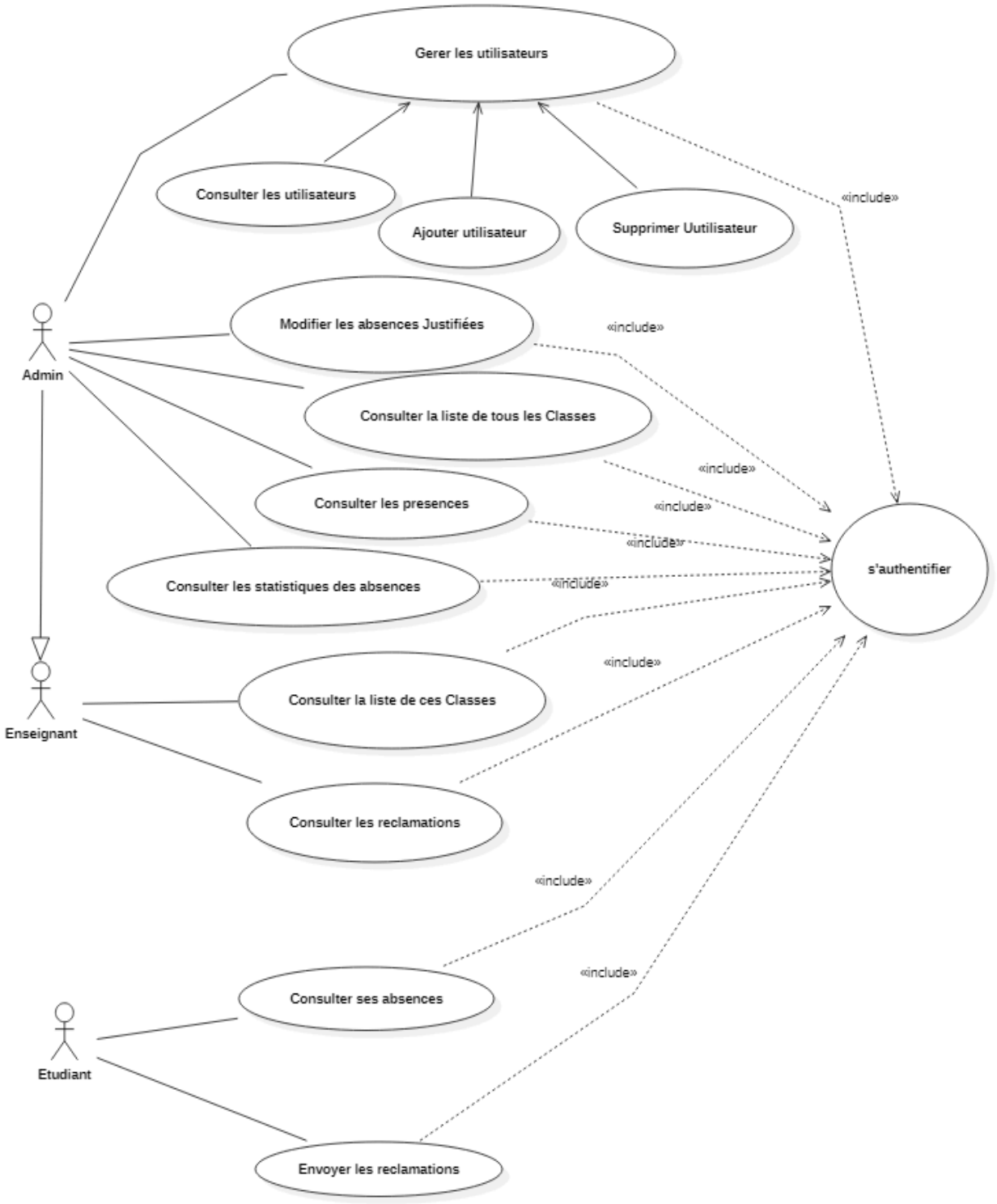
L'enseignant, cet acteur aura la possibilité de :

- Visualiser les statistiques et les absences :
  - Les absences de ses étudiants.
  - Les réclamations de ses étudiants.
  - Les pourcentages d'absences par (Matières/ niveau / enseignants…)

L'etudiant, cet acteur aura la possibilité de :

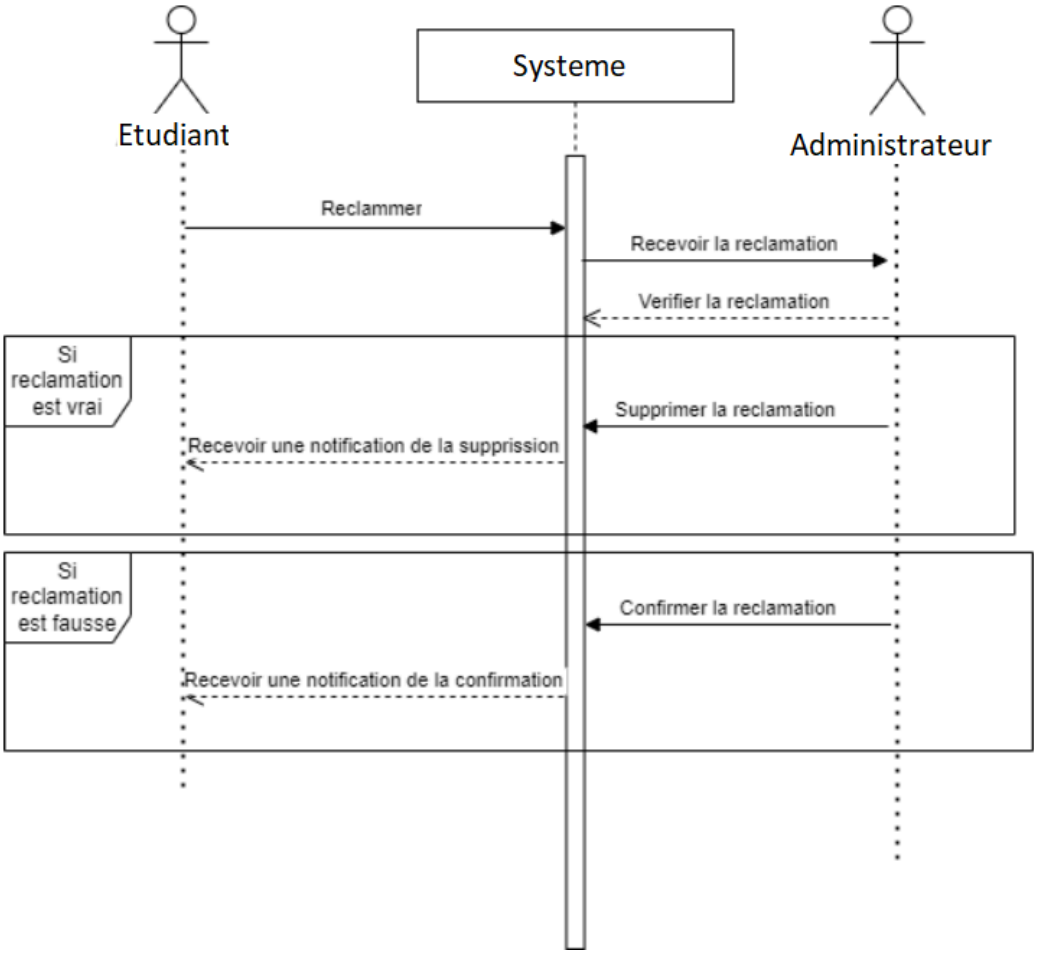
- Visualiser ses absences

2. Diagramme de cas d'utilisation :



(Figure 9 : Diagramme de cas d'utilisation de la partie web)

3. Diagramme de séquence pour la partie répondre aux réclamations :



(Figure 10 : Diagramme de séquence de la partie réponse sur les réclamations)

4. Backlog de la partie développement :

ID	Fonctionnalité	Description	Priorité	Estimation (points)
1	Intégration des capteurs RFID	Installer et configurer les capteurs RFID dans les salles de classe	Élevée	8
2	Collecte de données	Mettre en place une infrastructure pour la collecte de données des capteurs en temps réel	Élevée	10
3	Stockage sécurisé des données	Implémenter une base de données sécurisée pour stocker les données collectées	Élevée	6
4	Synchronisation avec le calendrier académique	Intégrer les données d'absence avec le calendrier académique	Moyenne	5

5	Développement du tableau de bord	Créer un tableau de bord convivial pour visualiser les données d'absence	Élevée	8
6	Génération de rapports et statistiques	Fournir des rapports et des statistiques pour une analyse approfondie	Moyenne	7
7	Analyse des tendances d'absence	Intégrer des modèles IA pour analyser les tendances et faire des prédictions	Moyenne	10
8	Interface d'analyse	Créer une interface permettant aux utilisateurs de comprendre les résultats de l'analyse	Moyenne	6
9	Gestion des cas d'absences justifiées	Intégrer un mécanisme pour prendre en compte les absences justifiées	Moyenne	5
10	Notifications automatiques	Mettre en place un système de notifications pour les absences non autorisées	Élevée	4
11	Intégration avec le système d'information de l'établissement	Connecter le système avec le système d'information existant pour synchroniser les données	Moyenne	8
12	Collecte de feedback des utilisateurs	Mettre en place des mécanismes pour recueillir les commentaires des utilisateurs	Basse	3
13	Mises à jour régulières	Planifier et effectuer des mises à jour régulières en fonction des retours et évolutions tech	Basse	5

(Tableau 2 : Backlog produit de la partie web)

III. Environnement de travail :

Nous allons présenter dans cette section les outils de développement employés pour mener à bien mon projet.

1. Environnement de développement :

- **Nest JS :**

(Figure 11 : Logo Nest JS)

Le framework NestJS offre la possibilité de concevoir des applications Node.js.

Il met à disposition une structure modulaire, des contrôleurs, des services et des dépendances afin de faciliter la création d'applications complexes et faciles à maintenir.

- **React Js :**

(Figure 12 : Logo React JS)

React.js est une extension JavaScript gratuite qui permet de concevoir des interfaces utilisateur interactives et dynamiques.

Elle offre la possibilité de fabriquer des éléments réutilisables qui gèrent leur propre état. Optimiser les mises à jour de l'interface en utilisant un DOM virtuel

2. Langages de programmation :

- **Javascript:**

La programmation interprétée est un langage de programmation principalement employé pour le développement Web via un serveur HTTP. Elle permet de rendre les pages web interactives et dynamiques en utilisant le modèle de l'objet documentaire (DOM).

- **HTML :**

Le langage de balisage HTML (HyperText Markup Language) est développé pour organiser les pages web. Il s'agit d'un langage d'écriture d'hypertexte qui offre la possibilité de définir la structure du contenu en utilisant des balises, de structurer le contenu des pages, d'intégrer des ressources multimédias et des programmes informatiques.

- **CSS :**

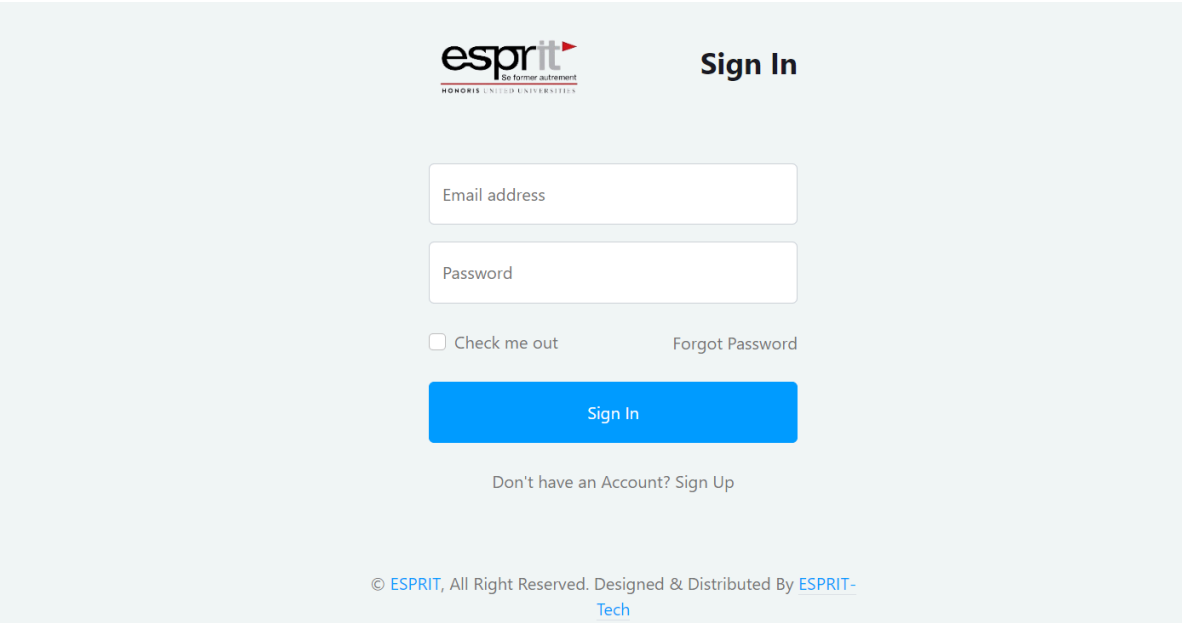
CSS est un langage de style qui permet de décrire la manière dont les documents HTML sont présentés et mis en forme.

IV. Réalisation du sprint 1 :

Après avoir détaillée la conception de la partie développement web, il est temps de passer à la mise en œuvre. Les interfaces développées sont les telles que :

1. Les interfaces de la partie web :

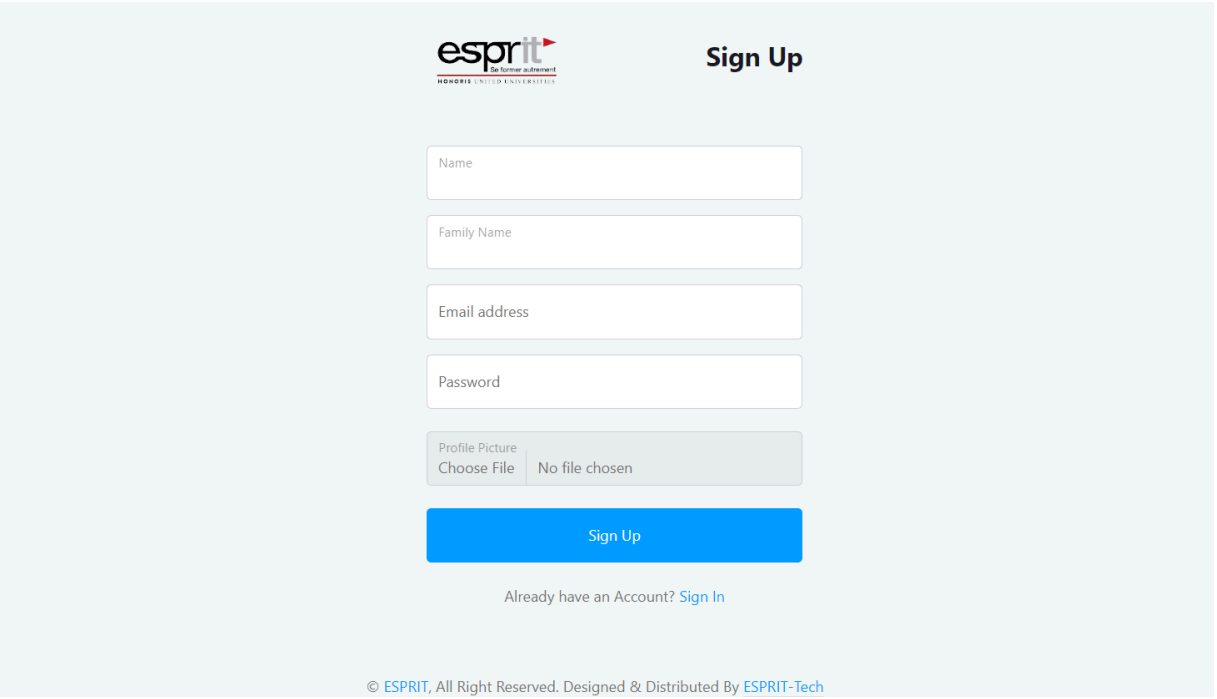
- **L'interface de l'authentification :**



(Figure 13 : Interface de l'authentification)

L'interface d'authentification représentée dans la figure permet aux utilisateurs d'accéder à l'application.

- **Interface de connexion :**

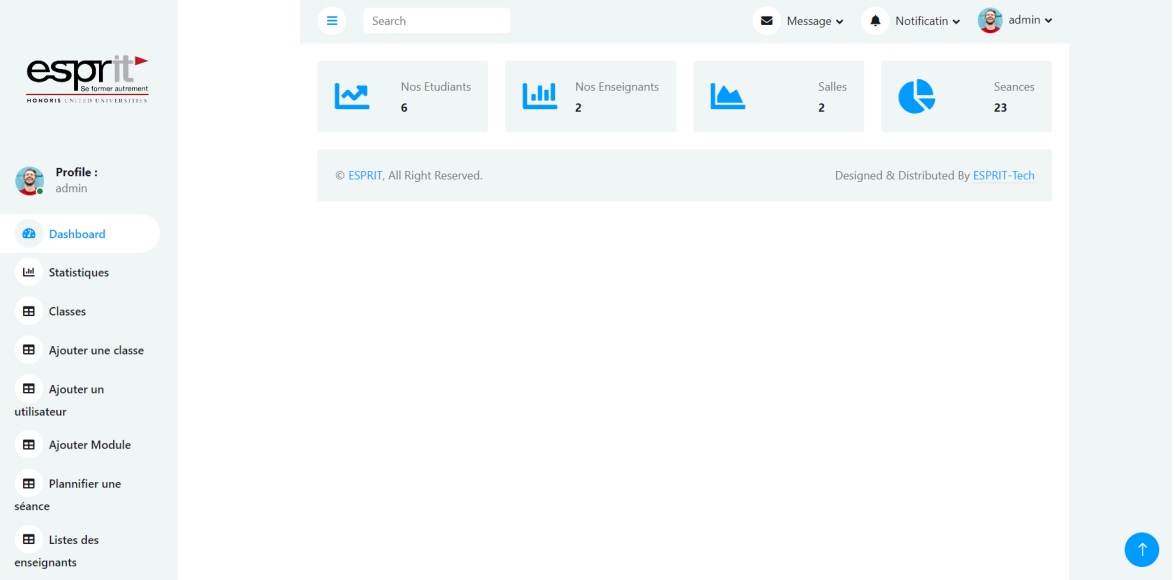


(Figure 14 : Interface de connexion)

La figure illustre l'interface Inscrire qui permet aux utilisateurs d'accéder au site afin de profiter des fonctionnalités proposées par la plateforme.

- **Session de l'administrateur et de l'enseignant :**

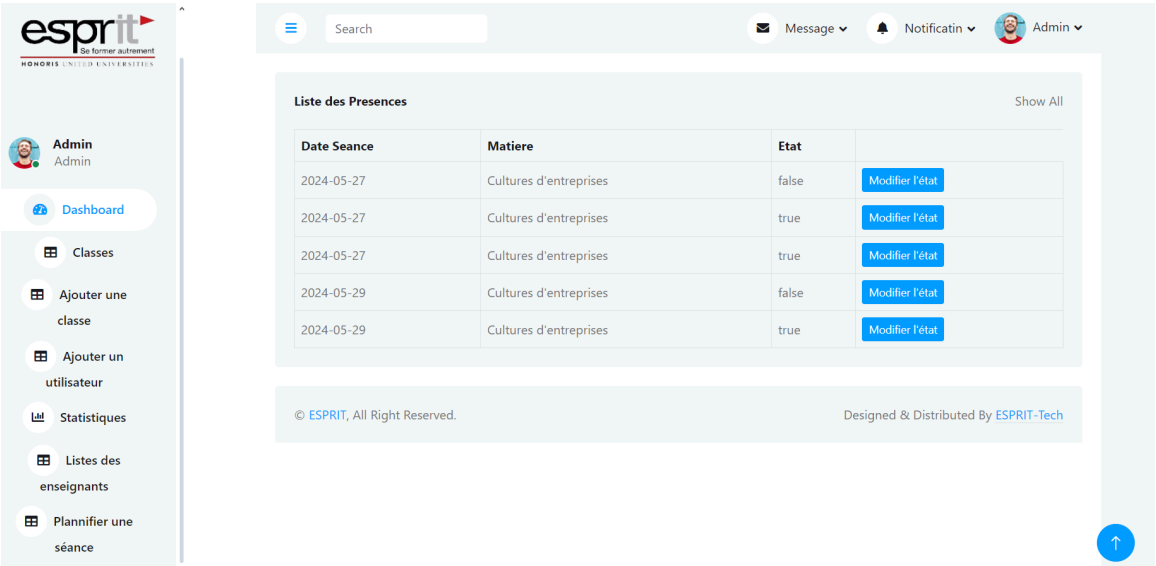
  - **Interface d'accueil :**



(Figure 15 : Interface d'accueil de l'administrateur)

La figure illustre l'interface d'accueil qui permet à l'administrateur et à l'agent de verbalisation de consulter les statistiques et les données liées aux infractions.

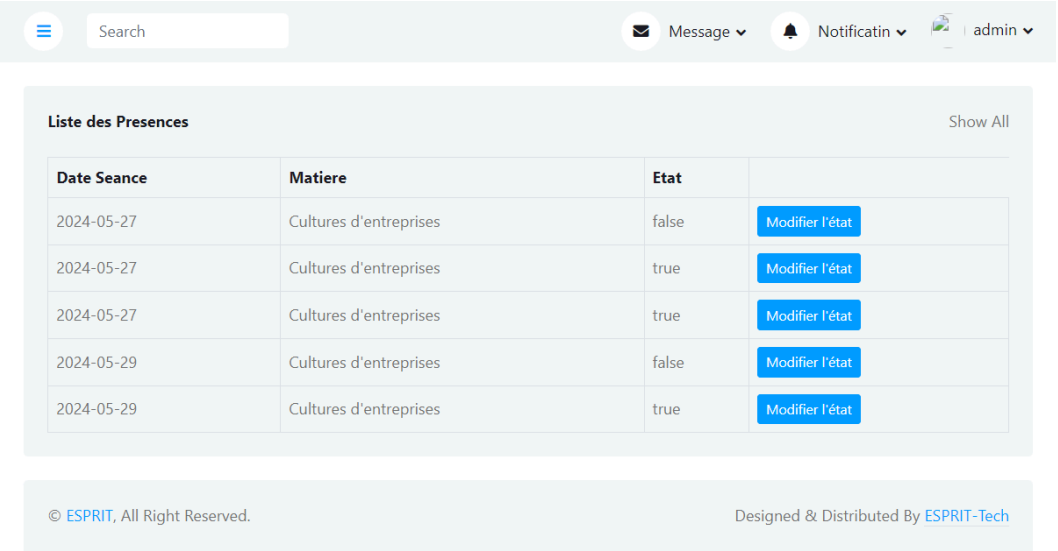
  - **Interface de la consultation de présence :**



(Figure 16 : Interface de la consultation de présence)

La figure illustre l'interface de la consultation de présence qui permet à l'administrateur et à l'enseignant de consulter les données relatives aux étudiants et à leurs absences.

- **Interface de la justification de l'absences :**

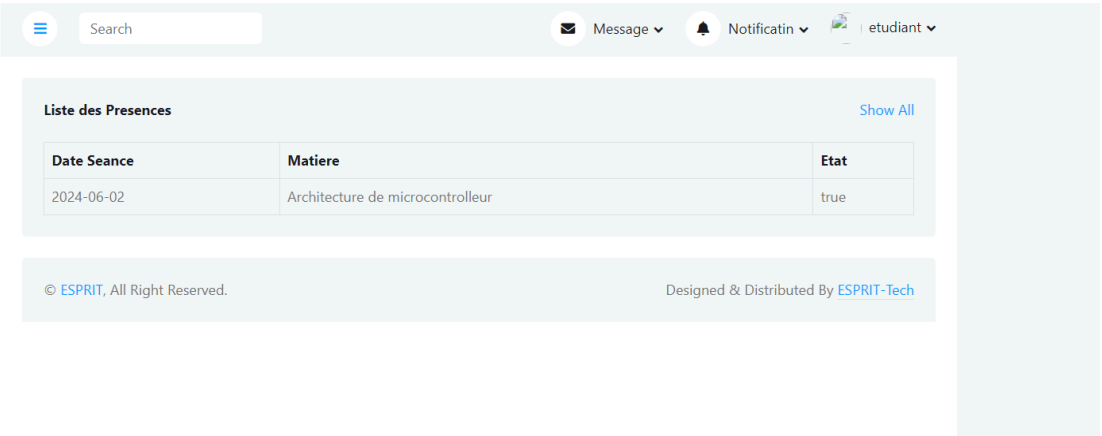


(Figure 17 : Espace de modification de l'absences)

La représentation graphique de l'interface de la justification de l'absence permet à l'administrateur de modifier l'état d'absence demandé par les étudiants en cas d'absences légitimes ou de problèmes techniques.

- **Session de l'étudiant :**

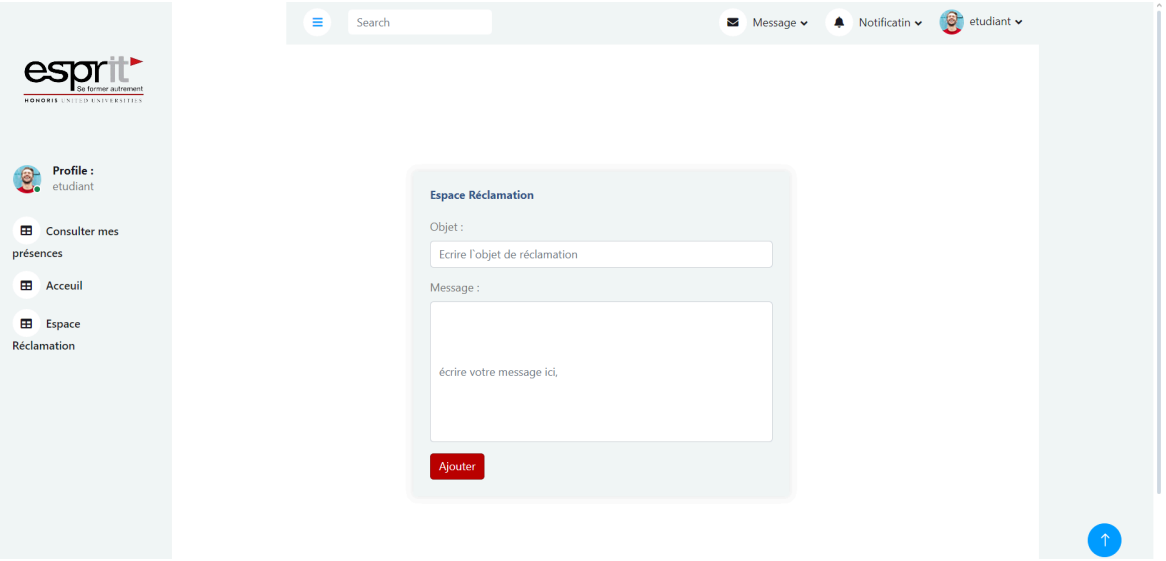
  - **Interface d'accueil :**



(Figure 18 : Interface d'accueil pour l'étudiant)

L'interface d'accueil représentée dans la figure permet à l'étudiant de consulter ses absences.

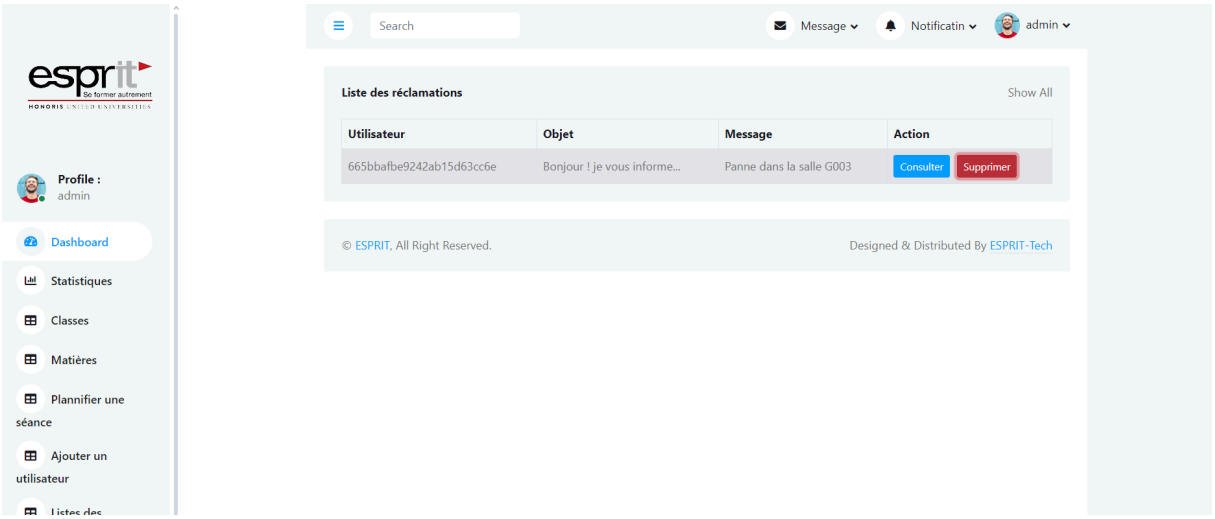
- **Espace de réclamation :**



(Figure 19 : Interface de réclamation)

L'interface de réclamation représentée dans la figure permet à l'étudiants d'envoyer des réclamations.

- **Espace de consultation des réclamations :**



(Figure 20 : Interface de consultation des réclamations)

Les étudiants peuvent consulter les réclamations des étudiants grâce à l'interface de consultation des réclamations illustrée dans la figure.

V. Conclusion :

Dans ce chapitre, j'ai exposé la description des exigences fonctionnelles en utilisant des illustrations de cas d'utilisation. Par la suite, j'ai sélectionné certaines fonctionnalités pour les détails, puis j'ai exposé quelques interfaces entre l'homme et la machine.

Sprint 2 : Partie Mobile

I. Introduction :

Pour ce projet, la deuxième étape du Sprint se focalise sur le développement mobile. Le but est de concevoir une application mobile qui permettra aux étudiants de vérifier leurs absences et de soumettre des réclamations directement depuis leur téléphone portable.

II. Analyse des besoins :

Afin de satisfaire les attentes des utilisateurs, il est nécessaire d'inclure plusieurs fonctionnalités essentielles dans l'application mobile.

1. Besoins fonctionnels :

Les utilisateurs :

L'étudiant, cet acteur aura la possibilité de :

- Consultation des absences
- Soumission de réclamations
- Sécurité et authentification

III. Environnement de travail :

Dans cette partie nous allons décrire l'outil de développement utilisé pour réaliser mon projet.

- **Android Studio :**

(Figure 21 : Android Studio)

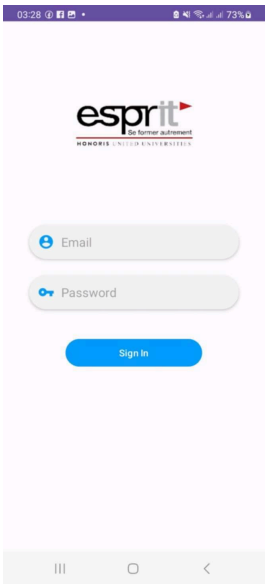
Android Studio est le logiciel officiel de développement intégré pour les applications Android. Il repose sur IntelliJ IDEA et offre des outils performants pour la création, le développement, le débogging et le déploiement d'applications Android. Il inclut un éditeur de code intelligent, un émulateur Android, des outils de mesure de performance, des modèles de projet et des intégrations avec des systèmes de gestion de version comme Git. Grâce à ces caractéristiques, il est plus facile de concevoir des applications solides et performantes pour la plateforme Android.

IV. Réalisation du sprint 2 :

Après avoir détaillée la conception de la partie développement web, il est temps de passer à la mise en œuvre. Les interfaces développées sont les telles que :

1. Les interfaces de la partie Mobile :

- **L'interface de l'authentification :**



(Figure 22 : Interface de l'authentification)

L'interface d'authentification représentée dans la figure permet aux utilisateurs d'accéder à l'application.

- **Interface d'accueil :**

(Figure 23 : Interface d'accueil pour l'étudiant)

L’interface d’accueil représentée dans la figure permet à l’étudiant de consulter ses absences.

- **Espace de réclamation :**

(Figure 24 : Interface de réclamation)

L’interface de réclamation représentée dans la figure permet à l'étudiants d’envoyer des réclamations.

V. Conclusion :

Dans ce chapitre, j'ai exposé les caractéristiques des exigences fonctionnelles. Par la suite, j'ai sélectionné quelques fonctionnalités afin de les préciser et enfin, j'ai exposé quelques interfaces entre l'homme et la machine.Une application mobile a été créée pour permettre aux étudiants de vérifier leurs absences et de soumettre des réclamations. Grâce à ce projet, j'ai pu acquérir une expertise dans diverses techniques de développement mobile et saisir les exigences particulières d'une application pour les étudiants.

Chapitre 4

Chapitre 4 : Etude et analyse globale de la partie embarquée

Sprint 3 : Partie embarquée

I. Introduction :

Je vais examiner la partie Embarqué dans ce sprint en me basant sur ces deux parties. :

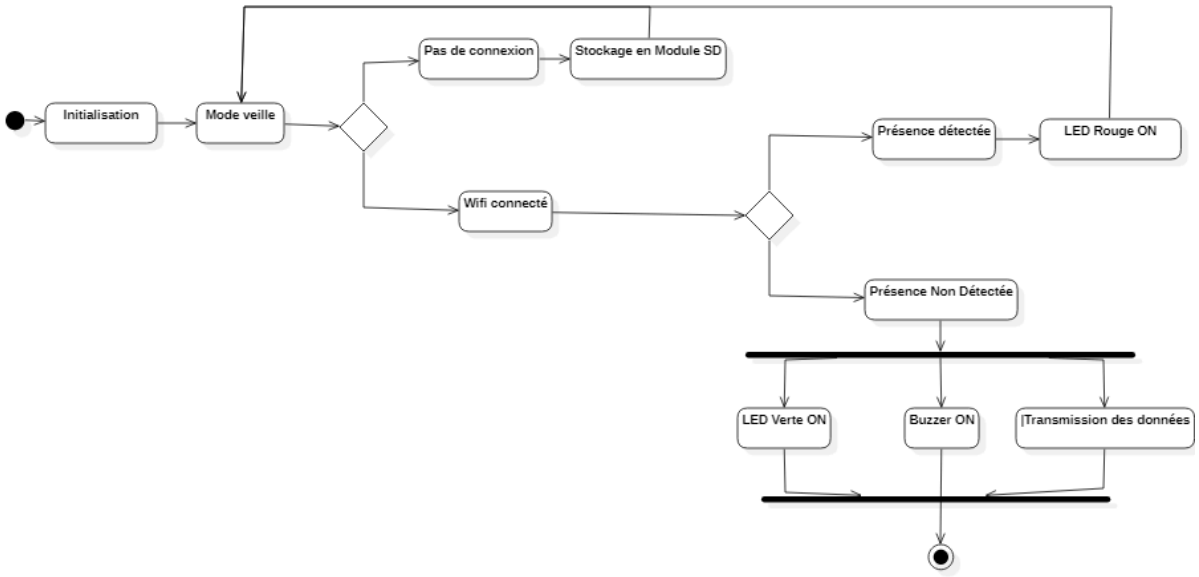
- Nœud de capteur.
- Passerelle (Gateway).

Une analyse comparative des différents éléments sera réalisée pour que je puisse prendre la décision qui répond à mes besoins.

II. Analyse des besoins :

1. Diagramme de cas de transition :

Le fonctionnement du système embarqué de gestion des absences des étudiants peut être expliqué à travers différents scénarios fréquents. Le diagramme d'état de transition de ce projet est présenté ci-dessous pour une meilleure compréhension de son fonctionnement :



(Figure 25 : Diagramme d'état de transition)

2. Principe de fonctionnement des nœuds :

Pour le nœud de capteur :

- Détection de Présence d'un Étudiant :

Le réseau Wi-Fi est connecté et en veille, un étudiant pénètre dans la salle et passe devant le capteur RFID.

- Le capteur RFID détecte la présence de la carte RFID de l'étudiant.
- Le microcontrôleur reçoit le signal de présence et vérifie l'identifiant de l'étudiant.
- La LED verte s'allume pour indiquer que la présence a été détectée.
- Le buzzer émet un bref son pour confirmer la détection.

- Non-Détection de Présence:

- Le système reste en mode veille, vérifiant régulièrement la présence d'un signal RFID.
- La LED rouge s'allume pour indiquer qu'aucune présence n'a été détectée.

- État Wi-Fi Non Connecté:

- Le microcontrôleur tente de reconnecter le Wi-Fi périodiquement.
- Les détections RFID sont stockées sur un module SD jusqu'à ce que la connexion soit rétablie.
- Une fois la connexion rétablie, les données stockées sont transférées vers la passerelle.

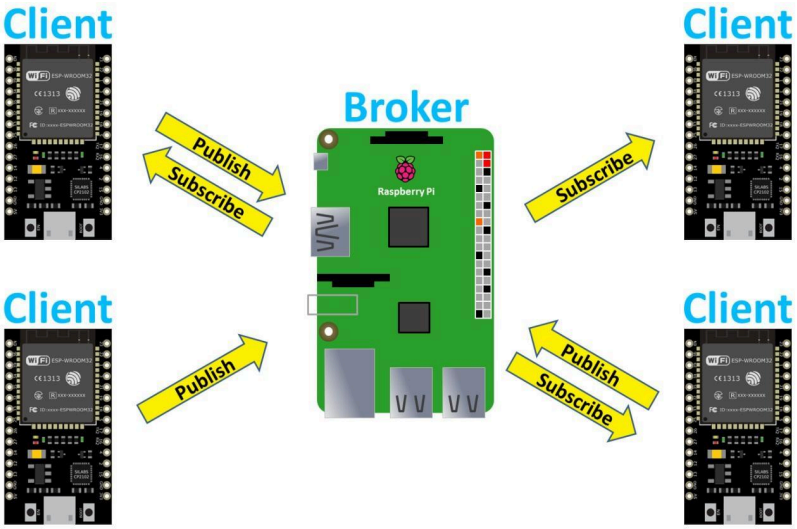
- État Wi-Fi Connecté :

- Le microcontrôleur maintient une connexion Wi-Fi active tout en étant en état de veille.

Pour la Passerelle (Gateway):

Une fois qu'une présence est détectée, les données sont transmises à la passerelle pour traitement.

- Le microcontrôleur compile les données de présence : identifiant de l'étudiant, heure de détection, etc.
- Les données sont envoyées via un module de communication via l'MQTT à la passerelle (Raspberry Pi).
- La passerelle reçoit les données, les stocke dans une base de données et met à jour le tableau de bord de l'application web.



(Figure 26 : Communication MQTT)

(Figure 27 : Exemple d'une requête reçue)

III. Environnement de travail :

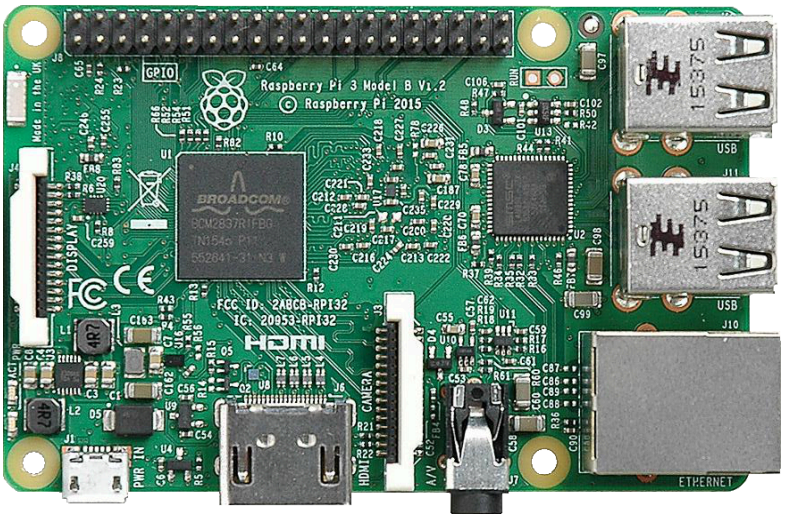
1. Environnement de développement matériel :

1.1 Etudes Comparatives des cartes :

1.1.1 Choix des cartes :

Caractéristique	ESP32	STM32	Arduino Uno	Raspberry Pi 4
Microcontrôleur	Xtensa Dual-Core 32-bit LX6	ARM Cortex-M4	ATmega328P	ARM Cortex-A72 Quad-Core
Fréquence d'horloge	240 MHz	180 MHz	16 MHz	1.5 GHz / 1.8 GHz
Mémoire RAM	520 KB	512 KB	2 KB	2GB, 4GB, ou 8GB LPDDR4
Mémoire Flash	4 MB	2 MB	32 KB	Carte microSD (jusqu'à 32 GB)
Connectivité	Wi-Fi, Bluetooth	pas de connectivité intégrée	pas de connectivité intégrée	Ethernet, Wi-Fi, Bluetooth
Ports GPIO	34 GPIO	114 GPIO	14 GPIO	40 GPIO
Interfaces	UART, SPI, I2C, PWM, ADC, DAC	UART, SPI, I2C, PWM, ADC, DAC	UART, SPI, I2C, PWM, ADC	UART, SPI, I2C, PWM, ADC, DAC, USB
Tension d'entrée	12v	3.3v	12v	5v
Consommation d'énergie	Faible	Faible à moyenne	Très faible	Moyenne à élevée
Prix	45.00 TND	130.00 TND	42.00 TND	200.00 TND

(Tableau 3 : Tableau comparatif sur les cartes électroniques)



(Figure 28 : Carte Raspberry pi 4)

(Figure 29 : Carte Esp32)

Pour la Passerelle (Gateway) :

J'ai opté pour le Raspberry Pi en tant que passerelle en raison de ses multiples bénéfices. La solution compacte, économique et puissante du Raspberry Pi est parfaite pour le traitement et la transmission des données. Il est capable de traiter de manière adéquate les tâches de collecte et d'analyse de données en temps réel, ainsi que de fonctionner avec des systèmes d'exploitation complets tels que Linux. De plus, le Raspberry Pi bénéficie d'une communauté active et d'un large soutien pour faciliter son déploiement et son intégration dans des systèmes complexes.

Pour le nœud de capteurs :

Les données de présence seront collectées à l'aide de la carte ESP32, qui est moins chère que l'STM32 et plus performante que l'Arduino Uno, en respectant les critères requis pour faire fonctionner ce nœud. C'est pourquoi j'ai décidé de choisir celle-ci.

1.1.2 Choix des capteurs et actionneurs :

- **Capteur RFID (RFID-RC522) :**

(Figure 30 : Capteur RFID-RC522)

Pour la détection de présence, j'ai besoin d'un capteur RFID qui va détecter les présences des étudiants à travers ces cartes étudiants.

Les caractéristiques techniques :

- Taux de transfert de données : Maximum 10 Mbit/s
- Fréquence de fonctionnement : 13,56 MHz
- Portée de lecture : 0 ~ 35 mm (carte mifare 1)

- **Buzzer :**

(Figure 31 : Buzzer)

J'ai besoin d'un amplificateur HX711 pour assurer la conversion analogique numérique des valeurs issues du capteur de poids.

- **2 Led :**

(Figure 32 : Led)

Pour le boîtier de collection du présences, j'ai besoin de 2 led, rouge et vert pour indiquer l'état de fonctionnement .

- **Module SD :**

(Figure 33 : Module SD)

Pour stocker les données en cas de rupture de connexion, j'ai besoin de ce module SD pour conserver les données.

2. Environnement de développement logiciel :

2.1 Choix des logiciels :

- **Arduino IDE :**

(Figure 34 : Arduino IDE)

C'est un logiciel open source, facile à utiliser. Ce logiciel peut être utilisé avec n'importe quelle carte Arduino.[7]

- **Real VNC Viewer :**

(Figure 35 : RealVNC Viewer)

RealVNC Viewer est un logiciel de visualisation à distance développé par RealVNC, qui permet aux utilisateurs de se connecter et de contrôler un ordinateur distant comme s'ils étaient physiquement présents.[8]

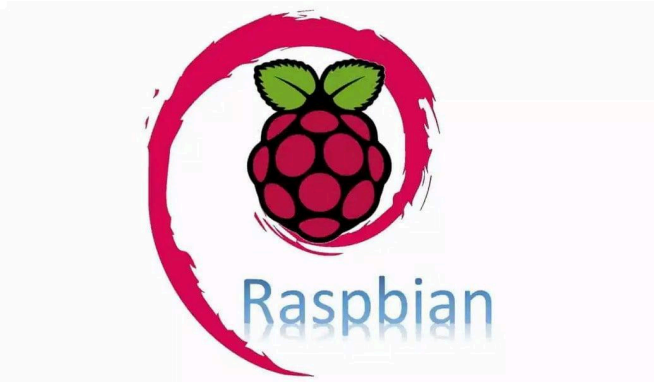
Parmi ces services, il offre :

- Gestion des sessions.
- Navigateur graphique
- Passerelle

[6] https://www.arduino.cc/en/software
[7] https://www.realvnc.com/en/company-profile/?__lai_s=0.1845238095238095&__lai_sr=15-19&__lai_sl=l#solutions

2.2 Choix du système d’exploitation :

- **Raspbian :**

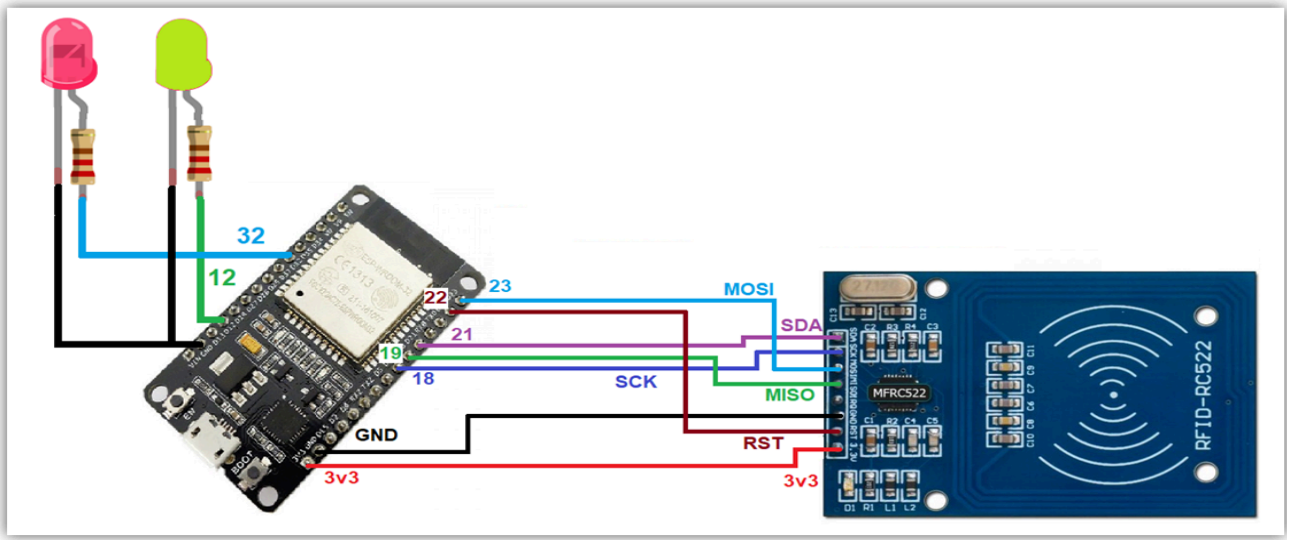


(Figure 36 : Rapsbian)

Les systèmes sur Raspberry Pi sont la plupart du temps des versions du système d'exploitation Linux, que les développeurs peuvent utiliser pour des fins particulières. Quant au Raspberry Pi, la légèreté du matériel nécessite un système compact et dépourvu d'applications superflues.

IV. Réalisation du sprint 3 :

1. Montage nœud de capteurs :



(Figure 37 : Montage nœud de capteurs)

Cette figure montre les différents composants qui construit le nœud de capteurs:

- 2 Led.
- Buzzer.
- Esp32- DevKit.
- Capteur RFID RC522.

2. La maquette du projet :

(Figure 38 : La maquette du projet)

V. Conclusion :

J'ai consacré ce chapitre au matériel et au logiciel que j'ai sélectionnés après une recherche approfondie. Cette étape revêt une importance capitale pour assurer le succès de la réalisation de ce sprint.

Chapitre 5

Chapitre 5 : Etude et analyse globale de la partie Intelligence Artificielle

Sprint 4 : partie Intelligence Artificielle

I. Introduction :

L'objectif de ce sprint était de concevoir, expérimenter et incorporer des modèles d'intelligence artificielle dans le projet. Les différents modèles sélectionnés étaient la Forêt Randome, les K-Nearest Neighbors (KNN) et un Réseau Nerveux. La collaboration avec Flask offre aux utilisateurs la possibilité d'interagir directement avec les modèles à travers une application web, ce qui rend les prédictions accessibles et dynamiques.

II. Préparation et Prétraitement des Données :

1. Les tâches :

La collecte des données a été réalisée en collectant des données provenant de sources pertinentes, en les nettoyant et en les standardisant afin de garantir une qualité optimale.

Au prétraitement, il était nécessaire de normaliser les données numériques, de traiter les valeurs manquantes et d'encoder les catégories.

Analyse de Données Exploratoires (EDA) : Une analyse des données dans Jupyter Notebook est réalisée afin de mieux appréhender leur structure, la répartition des variables et repérer les corrélations significatives. On a développé des représentations visuelles (diagrammes de dispersion, histogrammes, etc.) afin de confirmer les tendances présentes dans les données.

2. L’objectif :

Mettre à disposition des données de grande qualité prêtes à être exploitées dans les modèles d'intelligence artificielle et assurer une précision optimale lors de l'entraînement des modèles.

III. Environnement de travail :

1. Environnement de développement logiciel :

- **Flask :**

(Figure 39 : Flask)

Flask est un micro framework web écrit en Python. Il est classé comme un micro framework car il ne nécessite pas d'outils ou de bibliothèques particuliers. Il n'a pas de couche d'abstraction de base de données, de validation de formulaire ou de tout autre composant où des bibliothèques tierces préexistantes fournissent des fonctions communes.[8]

[8] https://en.wikipedia.org/wiki/Flask_(web_framework)

2. Implémentation des Modèles d'Apprentissage Automatique:

2.1 Modèle Random Forest :

- Le développement d'un modèle Random Forest dans Jupyter Notebook nécessite la définition du nombre d'arbres (n_estimators) et de la profondeur maximale (max_depth).

- Le modèle a subi des entraînements sur le jeu de données, puis a été évalué en ce qui concerne sa précision, son rappel et son score F1.

- Les résultats obtenus par le modèle sont rapides et solides, en particulier pour des données non linéaires, mais il présente des limites en termes de prédiction pour des données avec une corrélation multivariée élevée.

2.2 Modèle K-Nearest Neighbors (KNN) :

- Le modèle KNN a été configuré en utilisant diverses valeurs de k (nombre de voisins) afin d'améliorer les prédictions. Les métriques ont été les distances euclidiennes.
- L'entraînement et l'évaluation se combinent pour identifier la valeur de k qui offre les meilleures performances.
- Le modèle a démontré son efficacité pour des données de petite taille et bien réparties, mais a présenté une baisse des performances sur des ensembles de données volumineux.

2.3 Réseau de Neurones :

- La construction d'un modèle de réseau de neurones simple implique la création de trois couches distinctes : une couche d'entrée, une ou plusieurs couches cachées (avec activation ReLU) et une couche de sortie (avec activation softmax ou sigmoid).
- La formation et l'optimisation impliquent l'utilisation de backpropagation et de descente de gradient afin de réduire la fonction de perte. Essayer différentes

architectures (nombre de couches et de neurones) en utilisant Keras et Tensor Flow.

- Bien que ce modèle nécessite plus de temps pour être entraîné, il a montré une grande capacité d'apprentissage et de généralisation sur des ensembles de données plus complexes.

IV. Optimisation des Modèles:

- La validation croisée permet d'identifier les hyperparamètres optimaux pour chaque modèle en utilisant la validation croisée.
- Adaptation du nombre d'arbres et de la profondeur maximale dans la forêt Random.
- KNN : Évaluation de différentes valeurs pour k, en particulier entre 3 et 15.
- Modifications du taux d'apprentissage, du nombre de couches dissimulées et du nombre de neurones par couche dans le réseau de neurones.
- Sélection des modèles les plus efficaces : Évaluation de chaque modèle en fonction de différentes métriques (précision, rappel, score F1) pour comparer les performances.
- Le but est d'améliorer les performances et la précision des modèles en optimisant les hyperparamètres.
- La validation croisée permet d'identifier les hyperparamètres optimaux pour chaque modèle en utilisant la validation croisée.
- Adaptation du nombre d'arbres et de la profondeur maximale dans la forêt Random.
- KNN : Évaluation de différentes valeurs pour k, en particulier entre 3 et 15.
- Modifications du taux d'apprentissage, du nombre de couches dissimulées et du nombre de neurones par couche dans le réseau de neurones.
- Sélection des modèles les plus efficaces : Évaluation de chaque modèle en fonction de différentes métriques (précision, rappel, score F1) pour comparer les performances.
- Le but est d'améliorer les performances et la précision des modèles en optimisant les hyperparamètres.

V. L’Intégration:

1. Création de l’API Flask:

- Des routes API sont développées pour chaque modèle, comme /predict/random forest, /predict/knn, /predict/neural network.
- Les modèles entraînés sont exportés sous forme de fichiers pour être sérialisés.Pour les charger dans Flask et faciliter le déploiement, vous pouvez utiliser Joblib et Pickle.
- En temps réel, il est nécessaire de configurer les routes afin de recevoir des requêtes POST avec les données d'entrée et de renvoyer une prédiction sous forme de réponse JSON.

2. Tests avec Flask :

- Vérification des réponses API en utilisant des tests unitaires pour chaque route afin de garantir que les modèles fournissent des prédictions précises.
- Interface du consommateur Simple : Mise en place d'une interface basique (formulaire HTML) afin que les utilisateurs puissent saisir les données et obtenir des résultats directement à partir de Flask.
- Objectif : Faciliter l'accès aux modèles IA pour les utilisateurs grâce à une interface web dynamique et adaptative.

VI. Conclusion :

Grâce au sprint, une application IA fonctionnelle a été développée et mise en place, permettant de traiter en temps réel les données utilisateur grâce à Flask. Selon les scénarios de prédiction, chaque modèle (Random Forest, KNN, Réseau de Neurones) offre ses propres avantages, et l'application web Flask offre aux utilisateurs une expérience interactive et fluide.

Le déploiement a mis en évidence l'importance d'améliorer les hyperparamètres et le réglage de Flask afin d'assurer des performances satisfaisantes, ce qui ouvre des perspectives d'amélioration continue pour les prochaines itérations.

Conclusion générale

Le travail réalisé lors du stage de projet de fin d'études chez ESPRIT-Tech, ESPRIT, a été exposé dans ce rapport. J'ai exposé ma solution qui implique la mise en place d'un système de gestion des absences des étudiants en exploitant l'Internet des objets (IoT) et l'IA.

Grâce à ce stage, en tant qu'élève ingénieur, j'ai pu approfondir mes connaissances techniques et mes acquis lors de ma formation en travaillant sur ce projet. Il ne faut pas négliger le côté managérial et relationnel en s'intéressant de près au monde du travail. Mon projet consiste à concevoir un système automatisé qui repère en temps réel la présence des étudiants, analyse les données recueillies et envoie des notifications automatiques pour les absences non autorisées.

Au début de ce projet de fin d'études, j'ai effectué une analyse des systèmes de gestion des absences actuellement en place. Par la suite, j'ai exposé les technologies IoT et IA, ainsi que les méthodes techniques employées pour élaborer un système de détection utilisant des capteurs RFID, l'ESP32 pour l'intégration des capteurs, et le Raspberry Pi comme plateforme de traitement des données. En outre, la plateforme web conçue à l'aide de Nest.js et React propose une interface conviviale permettant de visualiser et de gérer les informations d'absence.

Je suis extrêmement satisfait d'avoir pu proposer une solution complète, pratique et novatrice, et d'avoir réussi à mener ce projet à bien. Grâce à ce projet, j'ai pu acquérir une grande expertise en maîtrisant diverses techniques et en utilisant des outils technologiques complexes.

Finalement, cette initiative m'a offert une opportunité exceptionnelle de mener à bien un projet concret, avec des objectifs clairement définis, et de me familiariser avec un environnement propice à mon développement professionnel.

Comme perspectives, je propose :

- Mieux comprendre et anticiper les tendances d'absence grâce à l'amélioration des algorithmes d'analyse prédictive.
- La mise en place de nouvelles technologies de détection, telles que des capteurs biométriques, afin d'améliorer la précision.
- L'amélioration de la compatibilité avec les systèmes d'information existants des établissements éducatifs pour une meilleure synchronisation des données.
- Le développement de fonctionnalités supplémentaires pour mieux gérer les cas d'absences justifiées et fournir des rapports encore plus détaillés.

Nétographie

[1] https://espit.tn/rdi/espri-tech

[2] https://www.classter.com/fr/blog/edtech-fr/le-systeme-de-gestion-des-presences-rfid-et-ses-avantages-pour-lenvironnement-scolaire/

[3] https://www.powerschool.com/company/

[4] https://helpdesk-class123.sosjt.com/serviceterms

[5] https://scrumguides.org/scrum-guide.html

[6] https://www.arduino.cc/en/software

[7] https://www.realvnc.com/en/company-profile/?__lai_s=0.1845238095238095&__lai_sr=15-19&__lai_sl=l#solutions

[8] https://en.wikipedia.org/wiki/Flask_(web_framework)

ESPRIT SCHOOL OF ENGINEERING

www.esprit.tn - E-mail : contact@esprit.tn

Siége Social : 18 rue de l'Usine - Charguia II - 2035 - Tél. : +216 71 941 541 - Fax. : +216 71 941 889

Annexe : 1-2 rue André Ampère - 2083 - Pôle Technologique - El Ghazala - Tél +216 70 250 000 - Fax +216 70 685454