

Projet De Fin D'étude

Conception et réalisation d'une plateforme IoT et Mise en place d'un système smart light

Licence en Sciences Mathématiques et Informatique

Réalisé Par :

Abdelhakim El Ghayoubi abdo.elghayoubi@gmail.com

Ilyas Bazzi ily2002as@gmail.com

Examiné par :

Pr. Benlahmar Elhabib **Encadrant**

Pr. Ouahabi Sara **Examinateuse**

Mr. Elhassani Abdeljalil **Co-encadrant**

Dédicace

A nos très chers parents En témoignage de notre amour et de notre gratitude pour les sacrifices consentis à notre égard A nos frères et sœurs Pour leur soutien moral et leurs conseils précieux et leur encouragement à nos chères ami(e)s Pour leurs aides et supports dans les moments difficiles. A toute ma famille Pour avoir attendu avec patience les fruits de leur bonne éducation A tous ceux qui de

Remerciement

Nous tenons à présenter notre profond remerciement à toutes les personnes, qui de près ou de loin, nous ont apporté leur aide et leur soutien. Nous exprimons nos plus sincères gratitude à Monsieur le Doyen de Faculté des Sciences Ben M'sick pour ses efforts déployés au profit des étudiants.

Nous adressons également nos remerciements les plus sincères à **Mr. BENLAHMER Elhabib** et **Mme. EL FILALI Sanaa**, nos encadrants et mentors, pour leur soutien constant, leurs conseils précieux et leurs orientations tout au long de notre travail. Leur encouragement considérable et leur confiance en nous ont été d'une valeur inestimable.

Nous tenons également à remercier chaleureusement **Mr. TACE Youness** et **Mr. ELHASSANI Abdeljalil** pour leur contribution et leur aide dans notre projet. Leur soutien et leur expertise ont grandement enrichi notre travail.

Nous ne manquerons pas non plus de dire un grand merci aux membres du jury, les professeurs qui ont eu l'amabilité de discuter avec nous certains points clés de notre analyse, et de nous donner ses remarques pertinentes.

Résumé

L'objectif principal de ce travail de PFE consiste à mettre en œuvre notre projet de bureau intelligent avec un système d'éclairage, qui représente une solution efficace et innovante pour améliorer l'efficacité énergétique et le confort dans les environnements de travail. Grâce à l'utilisation de la plateforme IoT et à l'intégration des données de luminosité et de présence, notre système offre une gestion optimale de l'éclairage, répondant ainsi aux besoins spécifiques de chaque utilisateur. De plus, cette plateforme peut interroger chaque système IoT présent dans le bureau, ce qui permet une intégration harmonieuse avec d'autres dispositifs connectés pour une automatisation avancée des tâches et une meilleure optimisation des ressources disponibles.

Sommaire

Introduction Générale	10
Chapitre 1 : Présentation & Préparation de Projet	11
 Introduction	11
1. Cadre d'accueil :	12
1.1. Présentation de la faculté des sciences Ben M'sick:.....	12
1.2. Organigramme :	13
2. Cadre de projet :	13
2.1. Contexte :	13
2.2. Problématique :.....	13
2.3. Solution et Développement :	14
2.4. Objectifs :	14
2.5. Périmètre du projet :.....	14
3. Méthodologie et formalismes adoptés :	14
3.1. Méthode en V :.....	14
3.2. Digramme de Gantt :.....	15
 Conclusion	16
Chapitre 2 : Étude du Projet.....	17
 Introduction	17
1. Généralités sur l'IoT	18
1.1 Introduction sur l'IoT :.....	18
1.2 L'histoire de l'IoT :	18
1.3 L'objectif de l'IoT :.....	19
1.4 Architecture de l'IoT :	19
2. Smart Office	20
2.1 Benchmarking sur les smart offices :	21
2.2 Choix de partie : Smart light	22
3. Platform IoT	22
3.1 C'est quoi un platform IoT :	22
3.2 Critères clés pour des plateformes IoT :	22
3.3 Benchmarking :	23
4. Architecture générale :	24
4.1 Prérequis des systèmes IoT :	25
4.2 Communication :	25

4.3	Serveur :	26
4.4	Base de données :	26
4.5	Interface :	26
4.6	Notion de module MVC :	27
4.7	Smart light fonctionnement :	28
4.8	Mise sous tension des systèmes IoT :	28
5.	Étude des solutions matérielles	28
5.1	Le module Wifi ESP8266 NodeMCU :	29
5.2	Le choix des capteurs :	30
6.	Étude des solutions logicielles	31
6.1	Plateforme de programmation Arduino :	31
6.2	Le Simulateur circuito.io :	32
6.3	VS Code :	33
6.4	XAMPP :	33
6.5	StarUML :	33
6.6	Postman :	33
6.7	Git Bash & GitHub :	34
6.8	FileZilla Client :	34
6.9	Composer :	34
7.	Étude des solutions technologies	34
7.1	Html :	34
7.2	Bootstrap :	35
7.3	Chart.js :	35
7.4	Laravel :	35
7.5	APIs :	35
7.6	DBMS MySQL :	36
Conclusion		36
Chapitre 3 : Analyse & Conception		37
Introduction		37
1.	Présentation UML :	38
1.1.	Définition :	38
1.2.	Pourquoi UML ?	38
2.	Diagramme de cas d'utilisation :	38
2.1.	Définition :	38
2.2.	Cas d'utilisation générale :	38

2.3. Cas d'utilisation d'un utilisateur :.....	39
2.4. Cas d'utilisation d'un Administrateur :.....	40
2.5. Cas d'utilisation d'une machine :	41
3. Diagramme de séquence.....	42
3.1. Définition	42
3.2. L'ajout d'un système IoT :.....	42
3.3. Consultation des variables d'une machine :.....	43
3.4 Modification d'une variable de la machine :	44
4. Diagramme de classes :	44
4.1. Définitions :.....	44
4.2 Les Classes :	45
Conclusion	45
Chapitre 4 : Implémentation & Déploiement	46
Introduction	46
1. Smart light :.....	47
1.1. Présentation du système Smart Light :.....	47
1.2. Prototype du système :	47
2. Déploiement :	50
2.1. Le pour quoi ?.....	50
2.2. Le Comment ?	50
3. Les Interfaces de Platform IoT [Smart office] :	51
3.1. Accueil :.....	51
3.2. Authentification :	52
3.3. Profil :.....	53
3.4. Dashboard :.....	54
3.5. Les Rôles :	55
3.6. Les Utilisateurs :	55
3.7. Les Machines [Systèmes IoT] :.....	56
Conclusion	58
Conclusion et perspectives	59

Liste des Figures :

Figure 1: Organigramme de la faculté des sciences Ben M'Sik	13
Figure 2 : Méthode en V gestion de projet.....	15
Figure 3: Diagramme de Gantt.....	16
Figure 4 : les types des couches d'un Architecture iot.....	19
Figure 5 : Architecture générale de projet	25
Figure 6 : Structure de module MVC.....	27
Figure 7 : fonctionnement de PIR.....	28
Figure 8 : Module Wifi ESP8266 NodeMCU.....	29
Figure 9 : Schéma descriptif de l'ESP8266	29
Figure 10 : Example des Capteurs	30
Figure 11 : Capteur de luminosité Photorésistance (LDR).....	30
Figure 12 : Capteur LDR	30
Figure 13 : Capteur mouvement (PIR).....	31
Figure 14 : Interface de circuito.io	32
Figure 15 : Diagramme de cas d'utilisation générale	39
Figure 16 : Diagramme de cas d'utilisation d'un Utilisateur	39
Figure 17 : Diagramme de cas d'utilisation d'un administrateur	40
Figure 18 : Diagramme de cas d'utilisation d'une machine	41
Figure 19 : Diagramme de séquence d'ajout d'un système IoT	42
Figure 20 : Diagramme de Séquence : Consulter les machines	43
Figure 21: Diagramme de Séquence : modifier une variable de la machine.....	44
Figure 22 : Diagramme de Classe	45
Figure 23 : Matériels nécessaire pour le système Smart Light	47
Figure 24 : Simulation de Système Smart Light	48
Figure 28 : Description de Déploiement.....	51
Figure 29 : Page d'accueil	52
Figure 30 : Profil d'un Administrateur	53
Figure 31 : Profil d'un Utilisateur	53
Figure 32 : Dashboard d'administrateur	54
Figure 33 : Dashboard d'utilisateur	54
Figure 34 : List des Rôles.....	55
Figure 35 : Liste des utilisateurs.....	55
Figure 36 : Configuration des machines pour un administrateur	56
Figure 38 : Configuration des machines pour un utilisateur	57
Figure 39 : Affichage des Information d'une machine	57
Figure 40 : Modifie les données d'une machine	58

Liste des Tableaux :

Tableau 1 : Description de Cas d'utilisation d'un utilisateur	40
Tableau 2 : Description de Cas d'utilisation d'un Administrateur	41
Tableau 3 : Description de Cas d'utilisation d'un machine	41

Liste des Abréviations :

Abréviation	Description
IoT	Internet of Things (l'Internet des objets)
Wi-Fi	Technologie sans fil permettant la connexion à Internet
UTMS	Universal Mobile Telecommunications System (Système de télécommunications mobiles universel)
MQTT	Message Queuing Telemetry Transport (Protocole de messagerie léger pour les communications machine à machine)
REST	Representational State Transfer (Style d'architecture pour les systèmes web)
API	Application Programming Interface (Interface de programmation d'application)
LPWAN	Low-Power Wide Area Network (Réseau étendu à faible consommation d'énergie)
SDK	Software Development Kit (Kit de développement logiciel)
ESP	Module Wi-Fi produit par Espressif Systems
PIR	Passive Infrared Sensor (Capteur infrarouge passif)
LDR	Light Dependent Resistor (Résistance dépendante de la lumière)
IDE	Integrated Development Environment (Environnement de développement intégré)
MVC	Model-View-Controller (Modèle-Vue-Contrôleur, un design pattern)
JSON	JavaScript Object Notation (Format d'échange de données lisible par l'homme et par les machines)
UML	Unified Modeling Language (Langage de modélisation unifié)
HTTPS	Hypertext Transfer Protocol Secure (Protocole de transfert hypertexte sécurisé)
PHP	Hypertext Preprocessor (Langage de script utilisé pour le développement web)
FTP	File Transfer Protocol (Protocole de transfert de fichiers)
SGBDR	Système de Gestion de Base de Données Relationnelle
DBMS	Database Management System (Système de gestion de bases de données)
MAC	Media Access Control (Contrôle d'accès au support de transmission)

Introduction Générale

Le présent rapport décrit le travail réalisé dans le cadre de notre projet de fin d'études au sein de la Faculté des Sciences Ben M'sick, en tant qu'étudiants en Licence des Sciences Mathématiques et Informatique parcours base de données

Ces dernières années, une nouvelle technologie a émergé dans les domaines de l'électronique et des réseaux informatiques : l'Internet des objets (IoT). Cette technologie permet la création des systèmes connectés aux réseaux informatiques, ouvrant ainsi de nouvelles possibilités.

Dans ce projet, nous explorons le concept de l'Internet des objets (IoT) et nous nous concentrons sur la conception d'un prototype de bureau intelligent, appelé "Smart Office", basé sur cette technologie. L'objectif principal est d'optimiser la consommation d'électricité et de mettre en place une plateforme simple à utiliser.

Pour mener à bien ce projet, nous avons commencé par effectuer une recherche approfondie sur la technologie de l'IoT, suivie d'une phase d'analyse et de conception. Cette étude nous a permis d'identifier plusieurs axes importants, englobant les caractéristiques et les aspects techniques nécessaires à la réalisation du prototype.

Chapitre 1 : Présentation & Préparation de Projet



Introduction



Ce chapitre vise à situer notre projet dans son contexte global en présentant son cahier des charges. Tout d'abord, on commence par une présentation du contexte du projet, en mettant en évidence la problématique générale à résoudre et les solutions de développement envisagées. Ensuite, nous détaillerons les objectifs du projet, en veillant à mentionner son périmètre d'action. Enfin, nous aborderons la méthodologie et les formalismes adoptés pour mener à bien le projet.



1. Cadre d'accueil :

1.1. Présentation de la faculté des sciences Ben M'sick:

La Faculté des Sciences Ben M'sick, en continue mutation dans la Ville de Casablanca, est rattachée à l'Université Hassan II qui regroupe 18 établissements universitaires. Depuis sa création en 1984, elle n'a cessé de contribuer à la diversification des spécialités et des circuits de formation et à la consolidation de la recherche scientifique au profit de ses étudiants, ainsi qu'au développement socioéconomique de Casablanca.

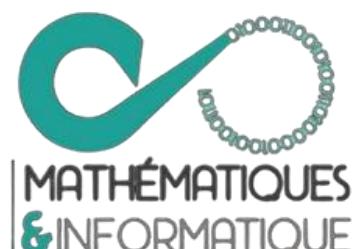


Dès son ouverture, la Faculté des Sciences Ben M'sik a accordé un intérêt particulier au développement de la recherche scientifique parallèlement à sa mission d'enseignement et de formation.

En effet, elle englobe actuellement 6 parcours de licences fondamentales et 18 Masters. La recherche scientifique connaît aujourd'hui une progression importante. 23 structures de recherche (2 centres de recherche, un observatoire « ordipu », une plateforme PINTECH, 19 laboratoires) ont vu le jour dans divers domaines tels que : les sciences et techniques de l'ingénieur, les matériaux, la biotechnologie, la géoscience

Depuis 2003, la faculté des sciences Ben M'sik dispense une formation modulaire et semestrielle dans le cadre de la réforme pédagogique de l'enseignement supérieur conformément au système LMD. Dans le cadre de la structuration de la recherche, que l'Université Hassan II – Casablanca, la Faculté des Sciences Ben M'sick a procédé à une nouvelle organisation et restructuration de ses équipes et laboratoires de recherche. C'est ainsi que la recherche à la faculté des sciences s'est organisée en 23 laboratoires et 2 équipes de recherches.

En 2008 suite à la réorganisation du cycle doctorat. La Faculté des Sciences Ben M'sick a mis en place le Centre d'Etude Doctoral (CED) : « Sciences et applications ». Ce centre est adossé à l'ensemble des structures de recherches accréditées par l'université.



La filière SMI donne une information de base à prédominance informatique avec un enseignement des modules de mathématiques et de physique. Les deux premiers semestres constituent un tronc commun avec la filière "SMA". Durant la deuxième et troisième année, on commence à introduire les notions de base (l'algorithme, programmations, structure des données, base de données, réseau informatique, système d'exploitation ...), sanctionné par un diplôme "Licence" permettant aux étudiants l'insertion dans la vie professionnelle ou la poursuite des études supérieures en Master.

Afin de renforcer la coopération, la Faculté des Sciences Ben M'sick a noué des relations de coopération internationale à travers la signature de conventions et accords avec des universités, des institutions universitaires, des centres et des laboratoires de recherche au Maroc et à l'étranger. Elle est aussi engagée dans divers projets de coopération universitaire internationale.

1.2. Organigramme :

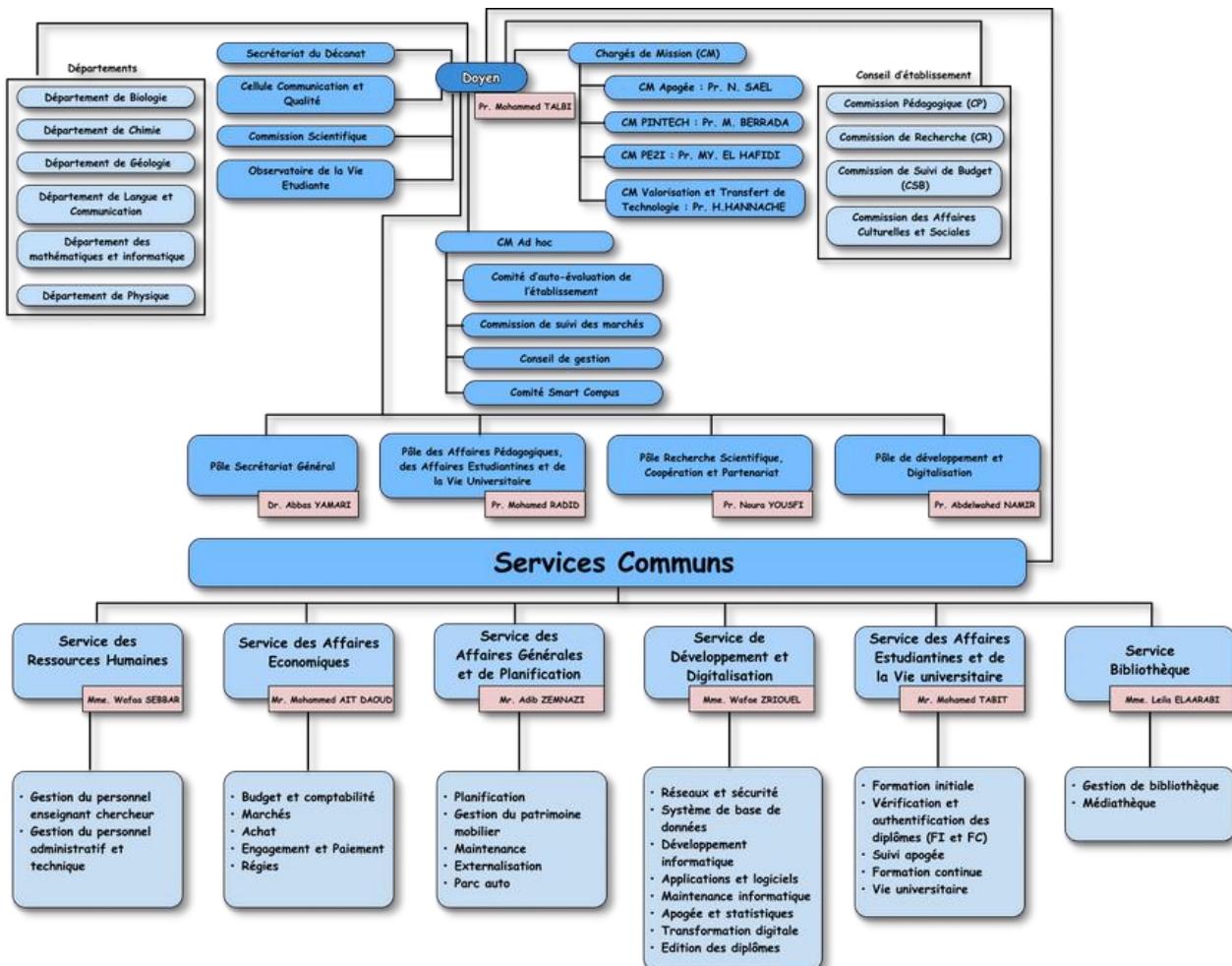


Figure 1: Organigramme de la faculté des sciences Ben M'Sik

2. Cadre de projet :

2.1. Contexte :

L'Internet des objets révolutionne les industries et transforme notre quotidien. Sa capacité à connecter les appareils, à collecter des données et à permettre une prise de décision intelligente offre un immense potentiel pour améliorer l'efficacité, la qualité de vie et créer un avenir plus durable. À mesure que l'IoT continue d'évoluer, son importance dans le monde réel ne fera que croître.

2.2. Problématique :

Les bureaux sont confrontés à de nombreux défis qui peuvent être résolus par l'IoT. Les systèmes de chauffage, de refroidissement et d'éclairage obsolètes peuvent gaspiller de l'énergie, tandis qu'il peut être difficile de contrôler et de surveiller l'accès à différentes parties du bâtiment. Une mauvaise qualité de l'air et une ventilation insuffisante peuvent entraîner

une diminution de la santé et de la productivité des employés, et il peut être difficile de suivre l'utilisation des équipements et des ressources. Enfin, une utilisation inefficace de l'espace peut entraîner une surpopulation ou une sous-utilisation de certaines zones. Donc, Comment les nouvelles technologies peuvent optimiser ce problème ?

2.3. Solution et Développement :

En utilisant l'IoT, il est possible de surveiller et d'analyser les données pour résoudre ces problèmes et améliorer continuellement les comportements et les flux de travail des employés exemple :

- Smart Offices (e.g Huawei offices)
- Smart speakers (e.g. Amazon Echo, Google Home)
- Smart lighting systems (e.g. Philips Hue, LIFX)
- Smart locks (e.g. August, Schlage)
- Smart conference systems (e.g. Cisco Webex, Zoom Rooms)
- Smart security systems (e.g. SimpliSafe, Ring)
- Smart sensors (e.g. temperature, humidity, vibration)

2.4. Objectifs :

Les objectifs du projet sont d'intégrer des solutions IoT pour résoudre les problèmes rencontrés dans les bureaux, d'améliorer l'efficacité énergétique et de réduire les coûts opérationnels. La plateforme IoT joue un rôle essentiel dans la réalisation de ces objectifs en permettant une gestion centralisée des différentes composantes du Smart Office.

2.5. Périmètre du projet :

Non projet consiste à :

- Allumer/éteindre la lumière [automatiquement ou manuellement]
- Détecter le mouvement et allumer/éteindre la lumière
- Régler la luminosité de la lumière
- Fait des opérations de gestion sur utilisateurs / machines
- Contrôler les machines inscrit
- Visualiser les données

3. Méthodologie et formalismes adoptés :

3.1. Méthode en V :

La méthode en V est une approche de gestion de projet qui suit une séquence linéaire et structurée. Elle est souvent utilisée dans des projets où les exigences sont bien définies dès le départ et où les changements sont peu probables comme se référer au schéma ce dessous (figure 2).

Nous avons opté pour la méthode V en raison de ses caractéristiques clés, notamment :

- Itératif.
- Centré sur l'architecture.
- Piloté par des cas d'utilisation.
- Orienté vers la diminution des risques.
- Structuration
- Séquentialité
- Validation et vérification
- Flexibilité

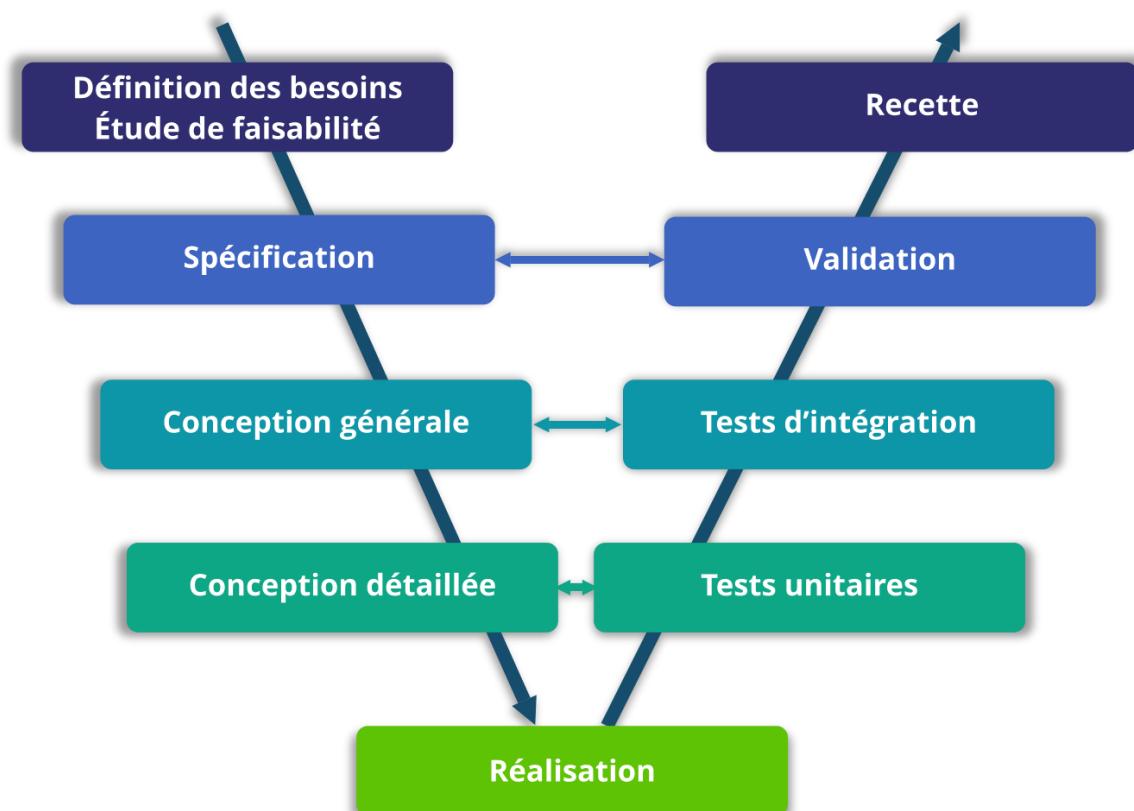


Figure 2 : Méthode en V gestion de projet

3.2. Diagramme de Gantt :

L'un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différentes activités (tâches) qui constituent un projet, couramment utilisé en gestion de projet.

On a fait une planification avec Gantt :



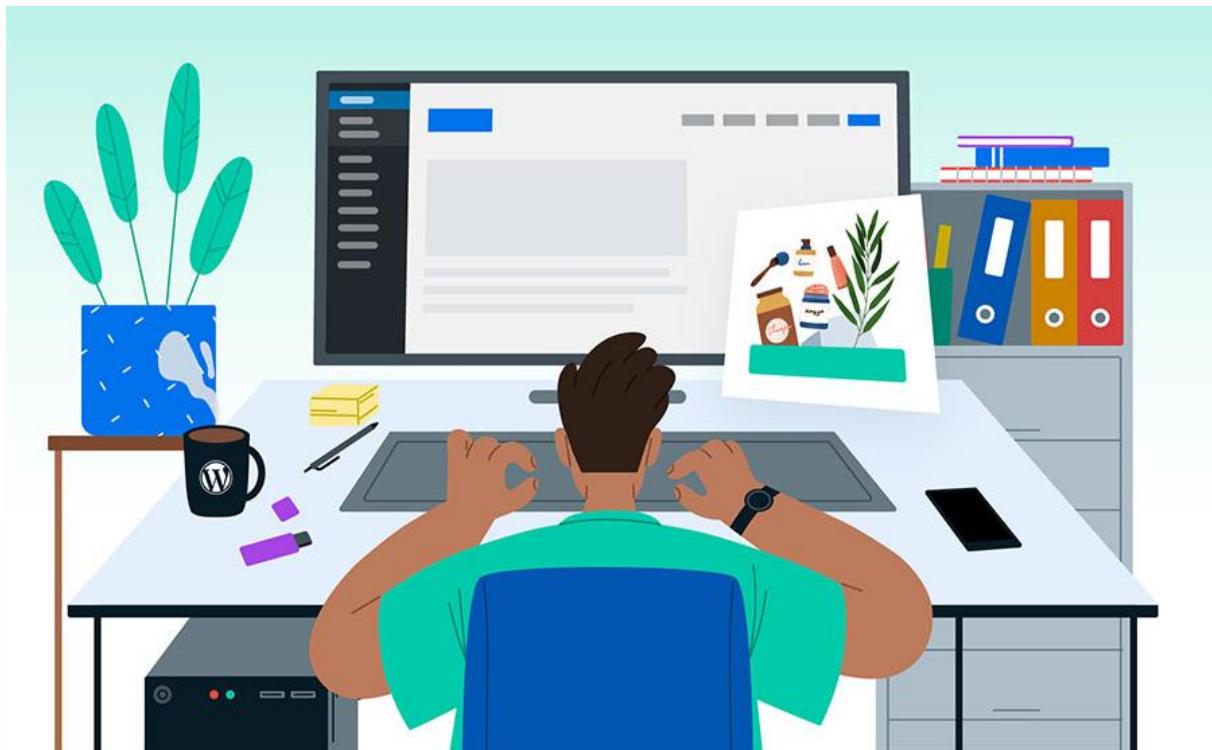
Figure 3: Diagramme de Gantt

L'étude préliminaire, elle consiste à déterminer et à ordonner les tâches du projet, à estimer leurs charges ce dessus.

Conclusion

Tout au long du chapitre, on a situé le contexte général du projet en élaborant un cahier des charges, la présentation des objectifs à atteindre, la méthodologie et les formalismes adoptés et la structure générale du système Smart office.

Chapitre 2 : Étude du Projet



Introduction



Dans ce chapitre, nous nous concentrerons sur l'étude approfondie du projet, mettant l'accent sur l'IoT et les solutions pour un smart office, avec une attention particulière sur la partie smart light. L'objectif principal de cette étude est de fournir une compréhension approfondie des aspects techniques et conceptuels liés à la réalisation d'un système de smart light, nous examinerons différentes solutions technologiques, y compris l'utilisation d'une plateforme IoT. La plateforme IoT sélectionnée jouera un rôle crucial dans la mise en œuvre et la gestion efficace du système de smart light.



1. Généralités sur l'IoT

Nous commencerons par introduire les concepts fondamentaux de l'IoT (Internet des objets), après nous explorerons l'histoire de l'IoT. Enfin nous discuterons également des objectifs & architecture des systèmes IoT.

1.1 Introduction sur l'IoT :



L'Internet des objets (IoT) se réfère à la connectivité des objets physiques tels que les appareils électroniques, les véhicules, les machines industrielles et les capteurs à Internet, permettant une communication bidirectionnelle entre ces objets et les systèmes informatiques. Les objets IoT sont équipés de capteurs, de logiciels et de matériel qui leur permettent de collecter des données et de les transmettre à des systèmes centraux pour l'analyse et l'automatisation. Les données collectées par les objets IoT peuvent être utilisées pour améliorer l'efficacité, la sécurité et la durabilité des processus et des produits, ainsi que pour créer de nouveaux produits et services innovants.

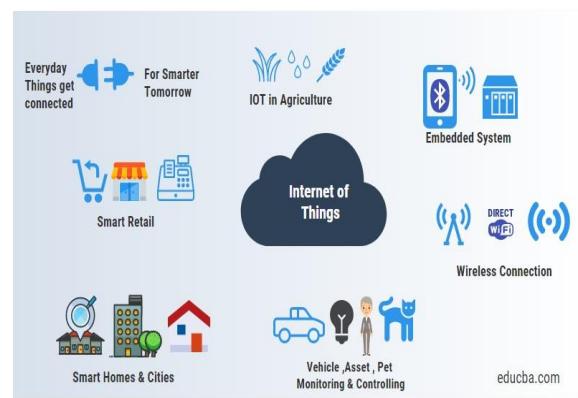
1.2 L'histoire de l'IoT :



L'Internet des objets (IoT) a commencé à prendre de l'importance au début des années 2000 avec le développement de la connectivité sans fil et de la technologie des capteurs. Cependant, l'idée de connecter des objets à Internet remonte aux années 1980, lorsque des appareils comme les distributeurs automatiques ont commencé à être connectés en réseau. Depuis lors, l'IoT a évolué pour inclure une gamme de technologies et d'applications, allant des appareils domestiques intelligents aux systèmes industriels avancés.

1.3 L'objectif de l'IoT :

L'Internet des objets a pour objectif de connecter des objets physiques à Internet, permettant ainsi une communication et une collaboration améliorées entre les objets, les personnes et les processus. Les objectifs spécifiques de l'IoT varient en fonction de l'application, mais ils peuvent inclure des améliorations de l'efficacité énergétique, de la productivité et de la sécurité, ainsi que des avantages pour la santé, l'environnement et la qualité de vie.



1.4 Architecture de l'IoT :

L'architecture de l'Internet des Objets (IoT) est la structure globale qui définit comment les différents composants de l'IoT interagissent entre eux pour permettre la collecte, le traitement et la transmission des données. L'architecture de l'IoT est généralement composée de plusieurs couches distinctes comme se référer au diagramme ce dessous :

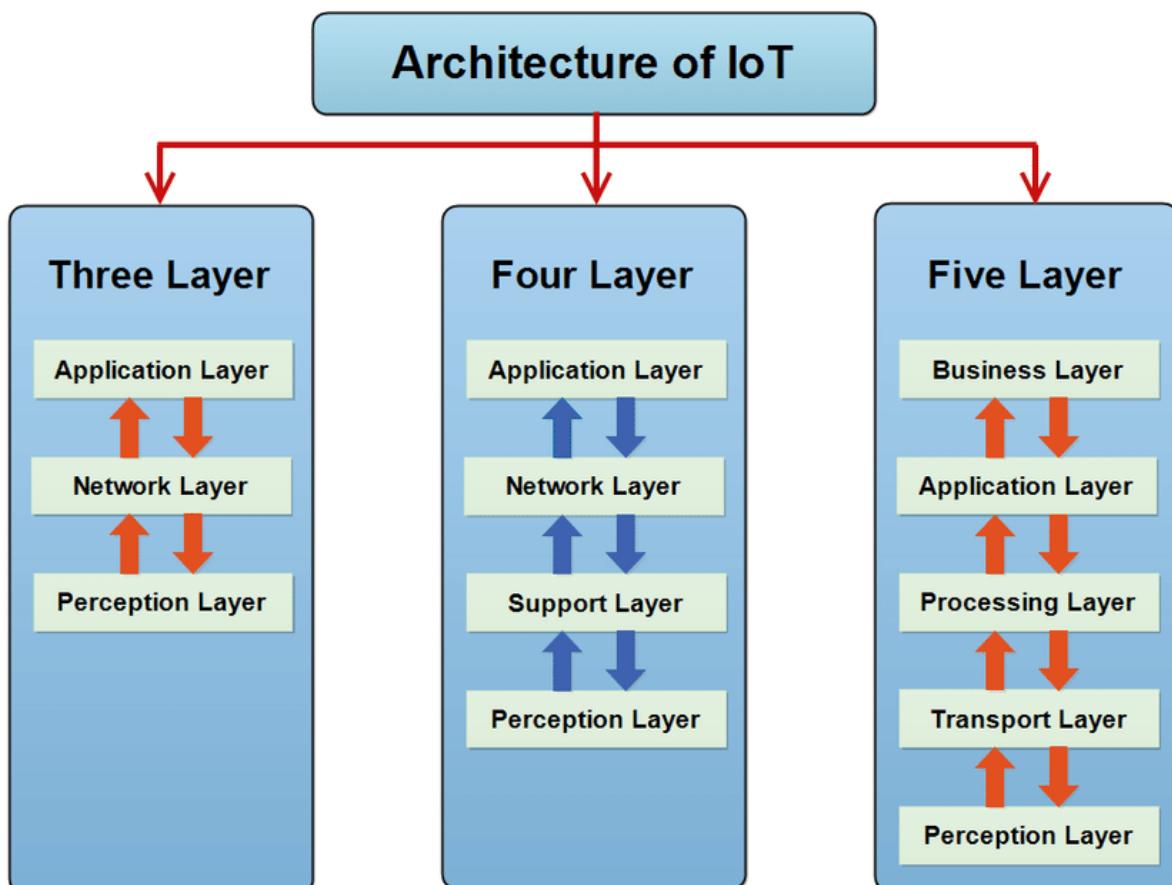


Figure 4 : les types des couches d'un Architecture iot

- **Couche de perception :**

Il s'agit de la première couche de l'architecture IoT. Dans la couche de perception, un certain nombre de **capteurs** et **d'actionneurs** sont utilisés pour recueillir des informations utiles telles que la température, la teneur en humidité, la détection d'intrus, les sons, etc. La fonction principale de cette couche est d'obtenir des informations de l'environnement et de transmettre des données à une autre couche afin que certaines actions peuvent être effectuées sur la base de ces informations.

- **Couche réseau :**

Comme son nom l'indique, il s'agit de la couche de liaison entre la perception et la couche middleware. Il obtient les données de la couche de perception et transmet les données à la couche middleware à l'aide de technologies de réseau telles que **3G**, **4G**, **UTMS**, **WiFi**, infrarouge, etc. Ceci est également appelé couche de communication car il est responsable de la communication entre la perception et la couche middleware. Tous les transferts de données effectués en toute sécurité en gardant les données confidentielles.

- **Couche middleware :**

Middleware Layer possède des fonctionnalités avancées telles que le **stockage**, le **calcul**, le **traitement** et les capacités de prise de mesures. Il stocke tous les ensembles de données et, en fonction de l'adresse et du nom de l'appareil, il donne les données appropriées à cet appareil. Il peut également prendre des décisions basées sur des calculs effectués sur des ensembles de données obtenus à partir de capteurs.

- **Couche d'application :**

La couche application gère tous les processus d'application en fonction des informations obtenues à partir de la couche middleware. Cette application consiste à envoyer des e-mails, activer une alarme, un système de sécurité, allumer ou éteindre un appareil, une smartwatch, une agriculture intelligente, etc.

- **Couche métier :**

Le succès de tout appareil ne dépend pas seulement des technologies utilisées, mais aussi de la manière dont il est livré à ses consommateurs. La couche métier effectue ces tâches pour l'appareil. Cela implique de créer des **organigrammes**, des **graphiques**, une **analyse des résultats**, et comment l'appareil peut être amélioré, etc.

2. Smart Office

Dans cette section, nous réaliserons une étude comparative sur les smart offices existants. Nous analyserons les technologies, les dispositifs et les solutions mises en œuvre pour rendre les bureaux intelligents. Nous étudierons les leçons apprises de ces cas d'utilisation, ce qui nous permettra de tirer des enseignements pour notre propre projet de smart office.

2.1 Benchmarking sur les smart offices :

Ces entreprises ont démontré les avantages de l'implémentation de l'IoT dans leurs bureaux, notamment une meilleure efficacité énergétique, des économies de coûts et un meilleur confort et productivité des employés. À mesure que la technologie de l'IoT continue d'évoluer, nous pouvons nous attendre à ce que de plus en plus d'entreprises adoptent des systèmes de bureau intelligents pour créer des espaces de travail plus connectés, efficaces et productifs.



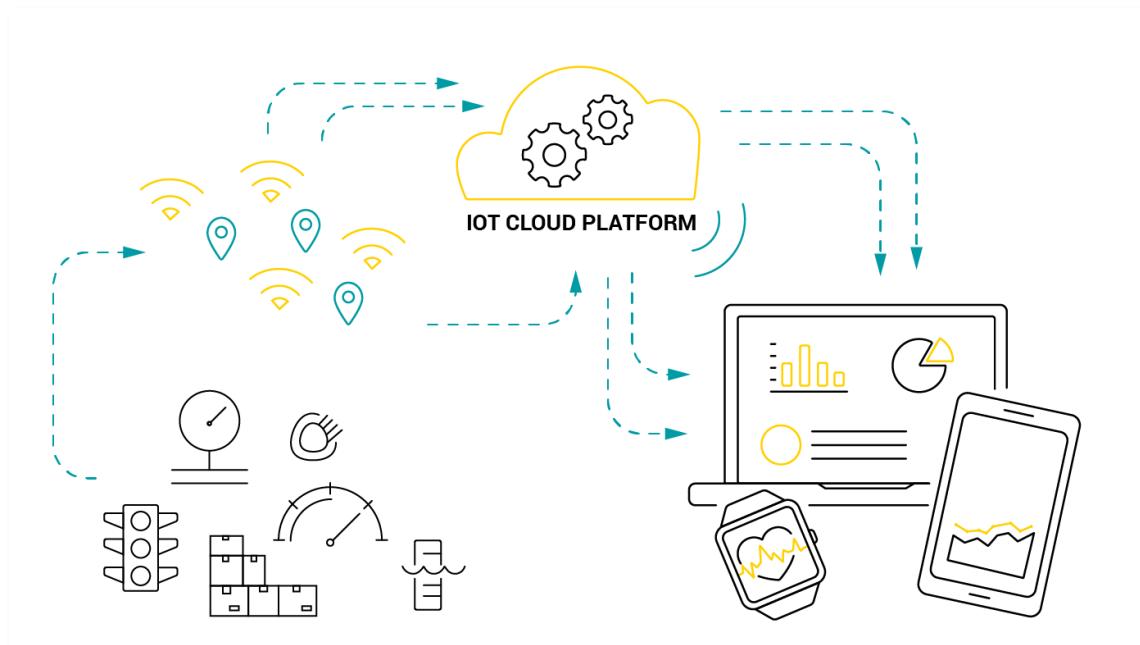
- **Huawei** : Le système de bureau intelligent de Huawei comprend des fonctionnalités telles que l'éclairage intelligent, la climatisation et la surveillance de l'environnement. Le système utilise des capteurs pour collecter des données sur la température, l'humidité et les niveaux de lumière afin de créer un espace de travail confortable et économique en énergie pour les employés.
- **Microsoft** : Le système de bureau intelligent de Microsoft comprend des fonctionnalités telles que l'éclairage intelligent, le contrôle du climat et les capteurs d'occupation. Le système utilise les données collectées par les capteurs pour optimiser la consommation d'énergie, réduire les coûts et créer un environnement plus confortable pour les employés.
- **Cisco** : Le système de bureau intelligent de Cisco comprend des fonctionnalités telles que l'éclairage intelligent, la planification de salles et la visioconférence. Le système utilise l'analyse de données pour optimiser la consommation d'énergie, réduire les coûts et améliorer la collaboration entre les employés.
- **Orange** : Orange a mis en place un système de bureau intelligent appelé "Smart Workplace". Le système utilise l'IoT pour surveiller l'utilisation des salles de réunion, contrôler la température et la lumière et fournir des informations en temps réel aux employés sur la disponibilité des salles de réunion et des espaces de travail.

2.2 Choix de partie : Smart light

Smart lighting, désignent les systèmes d'éclairage connectés à internet et pouvant être contrôlés à distance. Ces systèmes pouvant être programmés pour s'allumer ou s'éteindre, changer de couleurs et ajouter les niveaux de luminosité. Les éclairages intelligents deviennent de plus en plus populaires les bureaux et les espaces publics, car ils améliorent l'efficacité énergétique, réduire les coûts de maintenance.

3. Platform IoT

3.1 C'est quoi un platform IoT :



Une plateforme IoT est un ensemble de services logiciels et matériels qui permettent la connexion, la gestion et le contrôle des appareils connectés à Internet. Elle fournit une infrastructure pour faciliter le développement, le déploiement et la gestion d'applications IoT. Une plateforme IoT offre généralement des fonctionnalités telles que la connectivité des appareils, la collecte et le stockage des données, la gestion des appareils, l'analyse des données, la sécurité, la visualisation des données et les capacités d'intégration avec d'autres systèmes ou services. Elle vise à simplifier et à centraliser la gestion des appareils IoT, à faciliter la communication entre les appareils et les applications, et à permettre l'exploitation des données générées par les appareils connectés.

3.2 Critères clés pour des plateformes IoT :

Le benchmarking des plateformes IoT peut être une tâche complexe en raison de la grande variété de plateformes disponibles et des exigences spécifiques des différents cas d'utilisation.

Cependant, certains critères clés peuvent être pris en compte lors de l'évaluation et de la comparaison des plateformes IoT. Voici quelques facteurs importants à considérer :

- **Connectivité** : Évaluez la prise en charge de la plateforme pour différents protocoles de communication et technologies réseau tels que le **Wi-Fi**, le **Bluetooth**, les réseaux cellulaires et les réseaux à large bande basse consommation (**LPWAN**). Considérez la flexibilité de la plateforme pour intégrer différents appareils et capteurs.
- **Scalabilité** : Évaluez la capacité de la plateforme à gérer un grand nombre d'appareils connectés et de flux de données. Recherchez des fonctionnalités telles que la gestion des appareils, l'ingestion et le traitement des données qui peuvent gérer des volumes croissants de données à mesure que votre déploiement IoT se développe.
- **Sécurité** : La sécurité est primordiale dans les déploiements IoT. Évaluez les fonctionnalités de sécurité de la plateforme, telles que la transmission sécurisée des données, l'authentification des appareils, le contrôle d'accès et le chiffrement des données. Vérifiez si la plateforme est conforme aux normes de sécurité et réglementations de l'industrie.
- **Analyse et gestion des données** : Tenez compte des capacités de la plateforme en matière de stockage, d'analyse et de visualisation des données. Recherchez des fonctionnalités telles que **l'analyse en temps réel**, les capacités d'apprentissage automatique et l'intégration avec des outils d'analyse tiers. Évaluez la capacité de la plateforme à extraire des informations exploitables des données collectées.
- **Intégration et interopérabilité** : Évaluez la capacité de la plateforme à s'intégrer avec les systèmes, bases de données et applications existantes. Considérez la disponibilité des **API**, des **SDK** et la prise en charge de protocoles d'intégration courants tels que **REST** et **MQTT**. La compatibilité avec d'autres normes et **frameworks IoT** peut également être un avantage.
- **Convivialité pour les développeurs** : Évaluez les outils de développement de la plateforme, la documentation et la prise en charge pour le prototypage rapide et le développement d'applications. Considérez des fonctionnalités telles que les modèles préconstruits, les bibliothèques de code et les outils de débogage qui facilitent le processus de développement.
- **Coût et modèle de tarification** : Tenez compte de la structure de tarification de la plateforme, y compris les coûts initiaux, les frais d'abonnement, les coûts de stockage des données et les frais supplémentaires pour des fonctionnalités avancées ou la scalabilité. Comparez le modèle de tarification avec votre budget et vos besoins à long terme.
- **Support du fournisseur et écosystème** : Évaluez la réputation du fournisseur, sa fiabilité et les options de support. Considérez des facteurs tels que la taille de l'écosystème de partenaires

3.3 Benchmarking :

Arduino Cloud : Arduino Cloud est une plateforme basée sur le cloud spécialement conçue pour la gestion et le contrôle des appareils de l'Internet des objets (IoT) alimentés par Arduino. Elle offre un hub centralisé pour la connexion, la surveillance et le contrôle à distance des appareils basés sur Arduino. Arduino Cloud propose des fonctionnalités telles que la gestion

des appareils, la collecte et le stockage des données, la surveillance et le contrôle à distance, les analyses et les intégrations.

Blynk IoT : Blynk IoT est une plateforme IoT axée sur la facilité de développement d'applications et de projets IoT. Elle fournit une interface de glisser-déposer pour créer des interfaces utilisateur (UI) personnalisées permettant de contrôler et de surveiller les appareils connectés. Blynk IoT prend en charge une large gamme de plateformes matérielles et de protocoles de communication, ce qui la rend polyvalente pour diverses applications IoT.

Tuya : Tuya est une plateforme IoT qui propose des solutions complètes pour les applications d'habitat intelligent et d'IoT commercial. Tuya offre un écosystème complet comprenant des kits de développement logiciel (SDK), des services cloud et une gamme d'appareils intelligents compatibles. Elle permet aux développeurs et aux fabricants de construire et de déployer rapidement des appareils connectés avec des fonctionnalités telles que le contrôle à distance, le contrôle vocal et l'interopérabilité.

Google Cloud : Google Cloud est une plateforme de cloud computing proposée par Google qui offre une large gamme de services et d'outils pour la création et le déploiement d'applications IoT. Elle propose une infrastructure évolutive, des capacités de stockage et de traitement des données, des services d'apprentissage automatique et des outils d'analyse. Google Cloud fournit des services spécifiques à l'IoT tels que Cloud IoT Core pour la gestion des appareils et Pub/Sub pour l'ingestion des données et la messagerie.

Thingsboard: Thingsboard est une plateforme IoT open-source qui permet la gestion, la visualisation et l'analyse des appareils connectés et de leurs données. Elle offre des fonctionnalités telles que la gestion des appareils, la collecte des données, des tableaux de bord en temps réel, un moteur de règles et des capacités d'intégration. Thingsboard prend en charge différents protocoles de communication et offre une architecture extensible pour la personnalisation et l'intégration avec d'autres systèmes.

4. Architecture générale :

Dans ce diagramme (figure 5), Nous nous concentrons sur l'architecture générale du système, la communication via des API, l'architecture MVC, la gestion des données dans une base de données & la présentation conviviale des données.

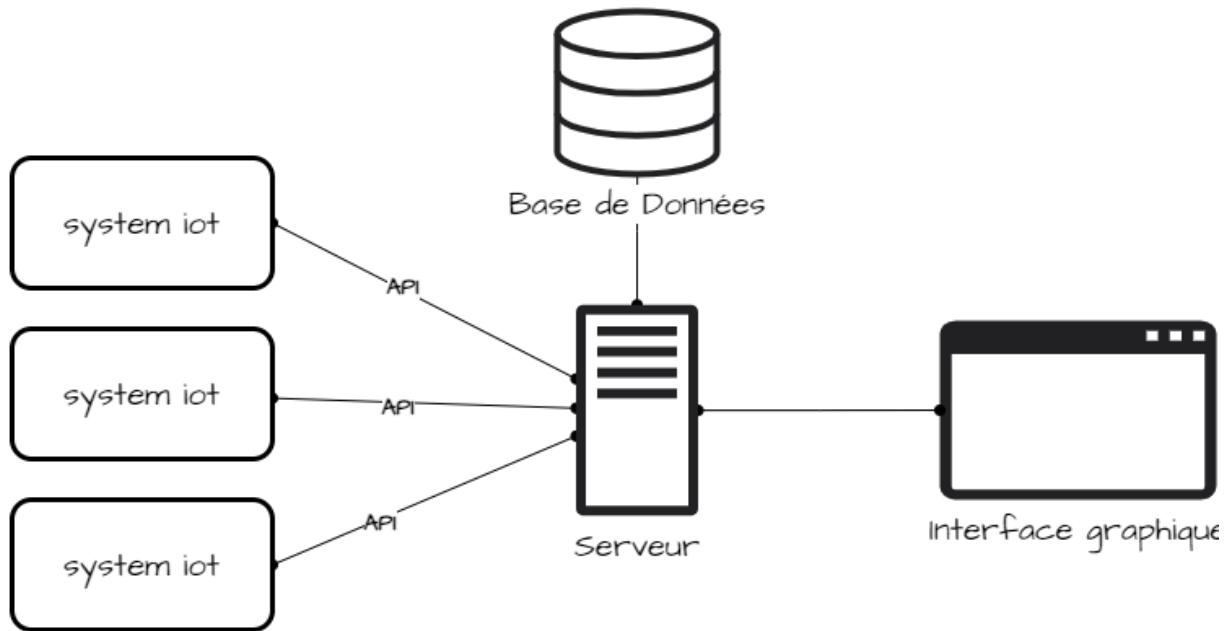
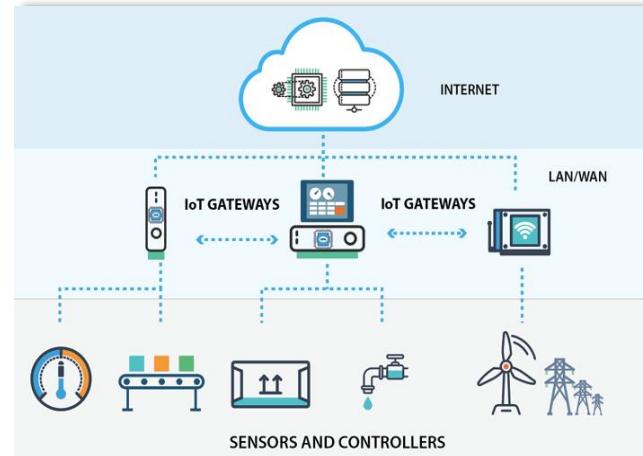


Figure 5 : Architecture générale de projet

4.1 Prérequis des systèmes IoT :

Pour permettre la communication et le contrôle des systèmes IoT par une plateforme IoT, il est essentiel d'avoir une connectivité réseau via des technologies sans fil telles que le **Wi-Fi**, ainsi que des microcontrôleurs comme les modules **ESP** pour établir une connexion à Internet. De plus, l'utilisation **d'API** facilite la communication structurée entre les systèmes IoT et la plateforme IoT, permettant ainsi l'échange de données et le contrôle à distance des systèmes IoT.



4.2 Communication :

La communication entre les cartes **ESP** (ESP8266, ESP32, etc.) et les plateformes IoT se fait généralement via des **API** (Interfaces de Programmation Applicatives). Les API fournissent un ensemble de règles et de protocoles qui permettent aux dispositifs ESP de se connecter et d'interagir avec les fonctionnalités offertes par la plateforme IoT.



Lors de la mise en place de la communication entre un **ESP** et une **plateforme IoT** via des **API**, certaines étapes sont généralement suivies. Tout d'abord, l'ESP doit être configuré pour établir une **connexion réseau**, comme le Wi-Fi, et se **connecter à Internet**. Ensuite, il doit s'authentifier auprès de la plateforme IoT en fournissant les informations d'identification appropriées, telles qu'une clé API ou un jeton d'accès.

Une fois la connexion établie et l'authentification réussie, l'ESP peut utiliser les API fournies par la plateforme IoT pour effectuer des actions spécifiques. Par exemple, il peut utiliser une API pour envoyer des mesures de capteurs à la plateforme, récupérer des informations d'état, activer des actions à distance, ou encore recevoir des mises à jour de configuration.

Les API fournissent une interface standardisée qui permet à l'ESP et à la plateforme IoT de communiquer de manière cohérente et fiable. Elles définissent les méthodes, les formats de données et les protocoles de communication à utiliser. Les données échangées entre l'ESP et la plateforme IoT peuvent être encodées dans différents formats, tels que **JSON** (JavaScript Object Notation) ou **XML** (eXtensible Markup Language), en fonction des conventions de l'API.

4.3 Serveur :

Le serveur joue un rôle central dans l'architecture IoT en facilitant la communication entre les systèmes IoT, les API, la base de données et l'interface utilisateur. Il agit comme un pont entre les différents composants, assurant le transfert fluide des données, la gestion des requêtes et la présentation des informations de manière conviviale. Le serveur est un élément essentiel pour assurer le bon fonctionnement et l'interopérabilité des différents aspects d'un système IoT.

4.4 Base de données :

La base de données dans une plateforme IoT sert de système de stockage et de gestion central pour les données générées par les dispositifs IoT. Elle permet une manipulation efficace des données, un traitement en temps réel, une évolutivité, des analyses avancées et des mesures de sécurité pour répondre aux besoins divers des applications IoT et faciliter la prise de décisions précieuses.

4.5 Interface :

Une interface bien conçue dans une plateforme IoT facilite la surveillance, le contrôle et la gestion des dispositifs connectés. Elle permet aux utilisateurs de tirer le meilleur parti des fonctionnalités de la plateforme, d'interagir avec les données IoT de manière efficace et d'automatiser les actions en fonction des besoins spécifiques de leur application.

4.6 Notion de module MVC :

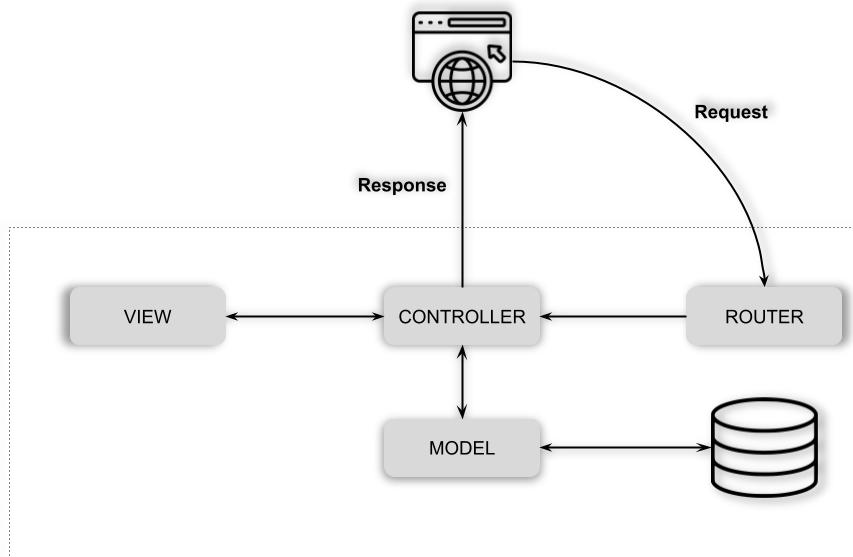


Figure 6 : Structure de module MVC

Le Modèle-Vue-Contrôleur (**MVC**) est un pattern architectural qui permet de structurer une application en séparant ses différentes responsabilités en trois composants principaux : le **modèle**, la **vue** et le **contrôleur**. Chaque composant joue un rôle spécifique dans le développement de l'application comme se référer au figure 6. Le MVC est largement utilisé dans l'industrie pour créer des projets web évolutifs et facilement maintenables.

- **Modèle** : Le composant Modèle est responsable de la gestion des données de l'application et de la logique métier associée. Il représente les données avec lesquelles les utilisateurs interagissent. Par exemple, dans une application de gestion de clients, le modèle peut inclure des classes pour accéder à la base de données, manipuler les données des clients et effectuer des opérations telles que l'ajout, la modification ou la suppression de clients.
- **Vue** : Le composant Vue est chargé de l'interface utilisateur de l'application. Il affiche les données du modèle à l'utilisateur et collecte les interactions de l'utilisateur. Il peut inclure des éléments tels que des formulaires, des tableaux, des boutons, etc. La vue est généralement définie à l'aide de langages de balisage tels que HTML, XML ou des outils de conception graphique.
- **Contrôleur** : Le composant Contrôleur agit comme un intermédiaire entre le modèle et la vue. Il gère la logique métier de l'application et traite les actions de l'utilisateur. Lorsqu'un utilisateur interagit avec l'interface utilisateur, le contrôleur reçoit les demandes, effectue les opérations nécessaires sur le modèle, puis met à jour la vue en conséquence. Par exemple, si un utilisateur demande d'ajouter un nouveau client, le contrôleur reçoit cette demande, crée un nouvel objet client dans le modèle, puis met à jour la vue pour afficher les changements.

4.7 Smart light fonctionnement :

a. Déetecter les mouvements :

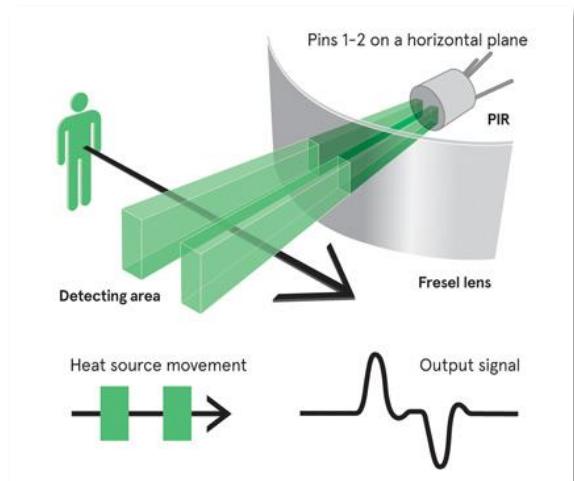


Figure 7 : fonctionnement de PIR

Les systèmes d'éclairage intelligent utilisent des capteurs de mouvement pour détecter la présence d'une personne dans une pièce et ainsi allumer ou éteindre automatiquement les lumières. Ces capteurs peuvent être équipés de technologies telles que les infrarouges passifs (**PIR**) ou les ultrasons pour détecter les mouvements avec précision, comme se référer au figure 7.

b. Régler la luminosité de la lumière

Utilisent des lampes connectées qui permettent de régler la luminosité des lumières. Certains systèmes d'éclairage intelligent sont équipés de capteurs de luminosité ambiante qui mesurent la quantité de lumière naturelle dans la pièce, permettant ainsi de régler automatiquement la luminosité des lumières en fonction des besoins Conception

4.8 Mise sous tension des systèmes IoT :

- Utiliser un **câble USB** : Vous pouvez alimenter votre Arduino Mega avec un câble USB qui se connecte à un ordinateur ou à un adaptateur mural USB.
- Utiliser une **batterie** : Vous pouvez alimenter votre système avec une batterie. Selon les besoins en tension et en courant de vos composants, vous devrez peut-être utiliser une ou plusieurs batteries en série ou en parallèle.
- Utiliser une **alimentation** : Vous pouvez utiliser une alimentation pour fournir la tension et le courant nécessaires à vos composants. Assurez-vous de sélectionner une alimentation capable de répondre aux exigences de votre système.
- Utiliser un **panneau solaire** : vous pouvez envisager d'utiliser un panneau solaire pour alimenter votre système. Cette option nécessite des composants supplémentaires, tels qu'un régulateur de charge et une batterie.

5. Étude des solutions matérielles

Nous nous concentrerons ensuite sur l'étude des solutions matérielles nécessaires pour la réalisation de notre système IoT. Nous présenterons le module Wifi ESP8266, en expliquant ses fonctionnalités et en décrivant la programmation de la carte ESP8266. Nous aborderons également le choix des capteurs, en mettant l'accent sur la définition d'un capteur, ainsi que

sur la photorésistance pour la détection de luminosité et le motion Sensor pour la détection de mouvement.

5.1 Le module Wifi ESP8266 NodeMCU :



Figure 8 : Module Wifi ESP8266 NodeMCU

Le **module Wi-Fi ESP8266 NodeMCU** est une carte de développement basée sur le microcontrôleur ESP8266. Il offre une connectivité Wi-Fi intégrée, ce qui en fait un choix populaire pour les projets d'Internet des objets (**IoT**). Le module **NodeMCU** facilite la programmation et la communication avec des services cloud, permettant aux développeurs de créer des applications connectées à Internet.

Ci-dessous vous trouverez le schéma montrant les composants de l'**esp8266 NodeMCU** :

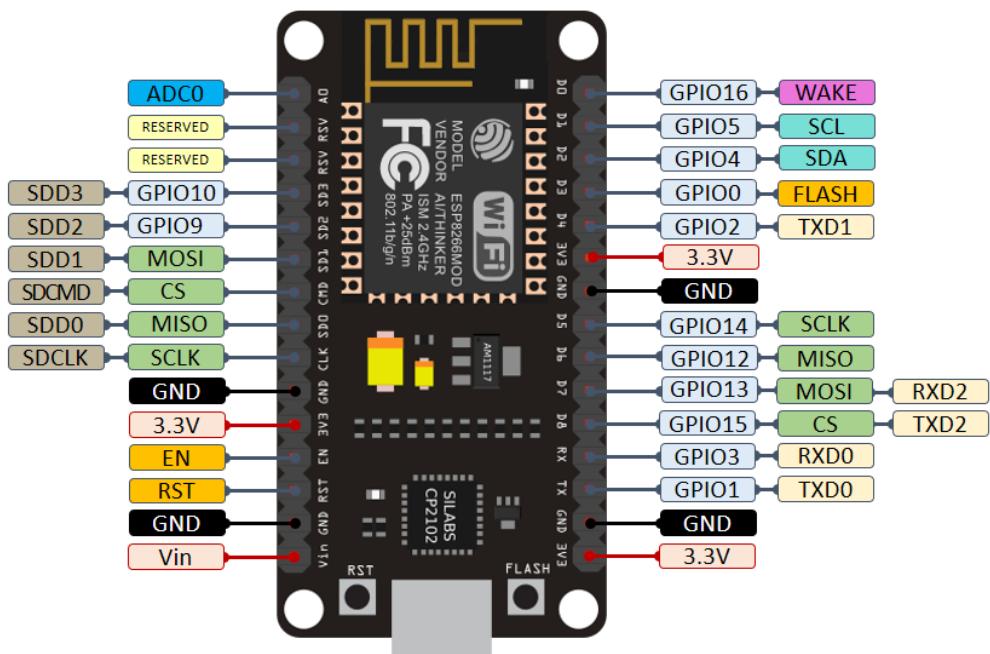


Figure 9 : Schéma descriptif de l'ESP8266

Dans notre projet on va utiliser essentiellement **Arduino IDE** pour programmer la carte **ESP8266 NodeMCU**.

5.2 Le choix des capteurs :

a. Définition d'un capteur :

Un capteur est un dispositif qui permet de mesurer ou de détecter un phénomène physique ou une grandeur spécifique, telle que la lumière, la température, la pression, le mouvement, l'humidité, etc. Il convertit ces informations en signaux électriques ou analogiques compréhensibles par les systèmes électroniques.

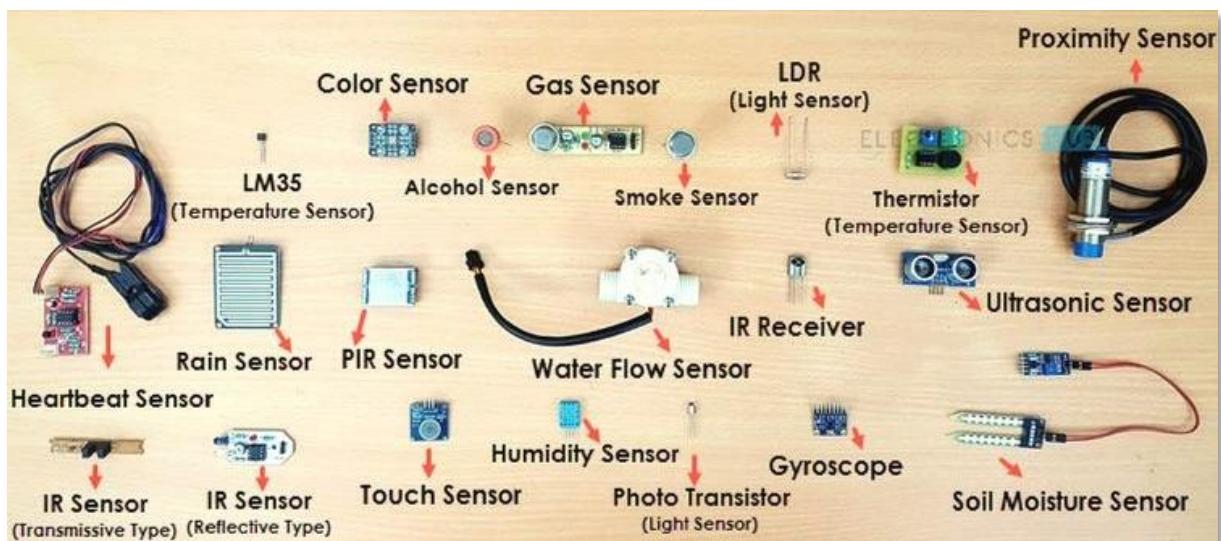


Figure 10 : Example des Capteurs

Plusieurs types des capteur pour différents fonctionnement

b. Capteur de luminosité [Photorésistance] :



Figure 11 : Capteur de luminosité Photorésistance (LDR)

Un capteur de photorésistance, également connu sous le nom de capteur de lumière, est un composant électronique sensible à la lumière ambiante. Il varie sa résistance en fonction de l'intensité lumineuse qu'il détecte.

Ce type de capteur est couramment utilisé pour des applications telles que les systèmes d'éclairage automatique, la mesure de la luminosité ambiante.

c. Capteur de mouvement Motion Sensor :



Figure 13 : Capteur mouvement (PIR)

Un capteur de mouvement est un composant électronique capable de détecter les changements dans l'environnement qui indiquent la présence ou le mouvement d'objets. Il peut être utilisé pour des applications telles que les systèmes d'alarme, les éclairages automatiques, les contrôles de sécurité, etc. Lorsqu'un mouvement est détecté, le capteur envoie un signal électrique qui peut être utilisé pour déclencher des actions ou des notifications

6. Étude des solutions logicielles

Cette section sera consacrée à l'étude des solutions logicielles utilisées dans le cadre de notre projet IoT. Nous présenterons la plateforme de programmation **Arduino**, en expliquant comment elle est utilisée pour le développement de notre système. Nous examinerons également le simulateur **circuito.io**, en mettant en évidence ses avantages et son utilité dans la conception et la simulation de circuits électriques. De plus, nous aborderons d'autres outils logiciels tels que **VS Code**, **XAMPP**, **StarUML**, **Postman**, **Git Bash & GitHub**, **FileZilla Client** et **Composer**, qui jouent un rôle clé dans notre développement.

6.1 Plateforme de programmation Arduino :



Arduino IDE : L'IDE Arduino est un environnement de développement intégré spécialement conçu pour programmer les cartes Arduino. Il fournit un éditeur de code, des outils de compilation et de téléchargement, ainsi que des bibliothèques et des exemples de code pour faciliter le développement d'applications pour Arduino. L'IDE Arduino prend en charge le langage de Programmation **C/C++** et permet aux utilisateurs d'écrire et de télécharger leur code sur les cartes Programmable.

6.2 Le Simulateur circuito.io :



Figure 14 : Interface de circuito.io

Interface simple et clair avec les diffèrent onglets (design, code, guide, composants.)

a. Présentation du Simulateur :



Circuito.io: Circuito.io est une plateforme en ligne qui facilite la conception et la configuration de circuits électroniques. Elle fournit des outils conviviaux pour sélectionner des composants électroniques, les connecter correctement et générer le schéma électrique et le code correspondant. Circuito.io est utile pour les débutants en électronique qui souhaitent créer rapidement des prototypes de circuits sans nécessiter de connaissances approfondies.

b. Les avantages de circuito.io :

- **Facilité d'utilisation :** circuito.io est conçu pour être convivial et accessible aux débutants ainsi qu'aux experts en électronique. Il propose une interface intuitive et conviviale qui permet de créer et de simuler des circuits électroniques sans nécessiter de connaissances approfondies en programmation ou en électronique.
- **Génération automatique du code :** l'un des avantages majeurs de circuito.io est sa capacité à générer automatiquement le code source nécessaire pour contrôler les composants de votre circuit. Il prend en compte les spécifications et les configurations de votre circuit, puis génère le code correspondant dans différents langages de programmation, tels que C++, Arduino, etc. Cela permet d'accélérer le processus de programmation et facilite l'intégration des composants dans votre projet.

6.3 VS Code :



VS Code : VS Code, abréviation de Visual Studio Code, est un éditeur de code source léger et polyvalent développé par Microsoft. Il est largement utilisé par les développeurs pour écrire du code dans différents langages de programmation. VS Code offre des fonctionnalités avancées telles que la coloration syntaxique, l'autocomplétion, le débogage, l'intégration avec des outils de contrôle de version, et une grande variété d'extensions pour personnaliser l'environnement de développement.

6.4 XAMPP :



XAMPP : XAMPP est un ensemble d'outils logiciels open source qui facilite la configuration d'un environnement de développement web local. Il inclut des composants tels que le serveur web Apache, la base de données MySQL et l'interpréteur de script PHP. XAMPP permet aux développeurs de créer et de tester des applications web sur leur propre ordinateur avant de les déployer sur un serveur distant.

6.5 StarUML :



StarUML: StarUML est un outil de modélisation UML (Unified Modeling Language) utilisé pour concevoir et visualiser des diagrammes de structure et de comportement dans le développement logiciel. Il permet de créer des diagrammes de classes, de séquence, d'activité, de composants, de déploiement, etc. StarUML facilite la communication et la conception de logiciels en fournissant une représentation graphique visuelle des différents aspects d'un système.

6.6 Postman :



Postman : Postman est un outil de collaboration et de test d'API populaire. Il permet aux développeurs de créer, tester et documenter facilement des API. Postman offre une interface conviviale pour envoyer des requêtes HTTP et HTTPS, inspecter les réponses, effectuer des tests automatisés, gérer des collections d'API et partager des informations avec des équipes de développement.

6.7 Git Bash & GitHub :



Git Bash & GitHub : Git Bash est une interface en ligne de commande (CLI) qui fournit un environnement Git complet pour la gestion des versions et le contrôle de code source. Git est un système de contrôle de version distribué largement utilisé pour suivre les modifications du code et faciliter la collaboration entre les développeurs. GitHub est une plateforme de gestion de code source basée sur Git, offrant des fonctionnalités telles que le stockage de dépôts, la collaboration, le suivi des problèmes et le déploiement continu.



6.8 FileZilla Client :



FileZilla Client : FileZilla Client est un logiciel de transfert de fichiers (FTP) gratuit et open source. Il permet aux utilisateurs de se connecter à des serveurs distants et de transférer des fichiers de manière sécurisée entre leur ordinateur et le serveur. FileZilla Client offre une interface conviviale avec des fonctionnalités telles que la gestion de sites, le glisser-déposer des fichiers et la reprise des transferts interrompus.

6.9 Composer :



Composer : Composer est un gestionnaire de dépendances pour les projets PHP. Il facilite l'installation, la mise à jour et la gestion des bibliothèques et des packages externes nécessaires à un projet PHP. Composer permet aux développeurs de spécifier les dépendances requises dans un fichier de configuration et de les télécharger automatiquement à partir du référentiel en ligne lors de la création d'un projet ou de l'ajout de nouvelles fonctionnalités.

7. Étude des solutions technologies

7.1 Html :



HTML : HTML, pour HyperText Markup Language, est le langage de balisage fondamental utilisé pour créer des pages web. Il fournit un format structuré pour organiser et présenter du contenu sur Internet. HTML utilise des balises pour définir la structure et la mise en page des éléments web tels que les entêtes, les paragraphes, les images et les liens, permettant aux navigateurs d'interpréter et d'afficher correctement le contenu web.

7.2 Bootstrap :



Bootstrap : Bootstrap est une populaire bibliothèque frontale qui facilite la création de sites web réactifs et adaptés aux mobiles. Elle fournit une collection de styles CSS pré-conçus, de composants JavaScript et de systèmes de grille adaptatifs, permettant aux développeurs de créer des interfaces web attrayantes et cohérentes sur différents appareils.

7.3 Chart.js :



Chart.js

Chart.js : Chart.js est une bibliothèque JavaScript qui simplifie la création de graphiques et de diagrammes interactifs et esthétiques sur les pages web. Elle offre une API simple et intuitive pour générer différents types de graphiques, tels que des diagrammes à barres, des graphiques linéaires, des graphiques circulaires, et bien plus encore. Avec Chart.js, les développeurs peuvent facilement visualiser et présenter des données de manière significative, améliorant ainsi l'expérience utilisateur de leurs applications web.

7.4 Laravel :



Laravel : Laravel est un framework web **PHP** réputé pour sa syntaxe élégante, ses fonctionnalités robustes et son approche conviviale pour les développeurs. Il suit le modèle d'architecture Modèle-Vue-Contrôleur (**MVC**), ce qui facilite l'organisation du code et la séparation des préoccupations. Laravel offre une large gamme d'outils et de fonctionnalités pour des tâches telles que le routage, la gestion de bases de données, l'authentification des utilisateurs et la mise en cache, permettant aux développeurs de créer des applications web évolutives et faciles à maintenir.

7.5 APIs :



APIs : Les API (Interfaces de Programmation d'Application) sont des ensembles de règles et de protocoles qui permettent à différentes applications logicielles de communiquer et d'interagir entre elles. Les API définissent les méthodes et les formats de données que les applications peuvent utiliser pour accéder et échanger des informations ou effectuer des tâches spécifiques. Elles permettent aux développeurs d'intégrer des services externes, d'accéder aux bases de données, de récupérer des données à partir de serveurs distants et de construire des systèmes logiciels modulaires et interconnectés.

7.6 DBMS MySQL :



DBMS MySQL : MySQL est un système de gestion de base de données relationnelle (SGBDR) open source très populaire. Il est utilisé pour stocker, gérer et organiser de grandes quantités de données de manière efficace et sécurisée. MySQL offre des performances élevées, une sécurité avancée, une scalabilité, une flexibilité et une intégration avec de nombreux langages de programmation. Il est largement utilisé dans les applications web, les systèmes d'information et les solutions d'entreprise.

Conclusion

Dans ce chapitre nous a permis d'approfondir notre compréhension du projet en étudiant les différents aspects liés à l'IoT, aux solutions matérielles, aux solutions logicielles et aux technologies associées. Ces connaissances constitueront une base solide pour la mise en œuvre du système Smart office.

Chapitre 3 : Analyse & Conception



Introduction

La phase d'analyse et spécification des besoins présente une étape importante dans le cycle de développement d'un projet. En effet elle permet de mieux comprendre le travail demandé et Les besoins des différents utilisateurs que le système doit accomplir

1. Présentation UML :

1.1. Définition :

Le Langage de Modélisation Unifié, de l'anglais **Unified Modeling Language (UML)**, est un **langage** de modélisation graphique à base de **pictogrammes** conçu comme une méthode normalisée de visualisation dans les domaines du **développement logiciel** et en **conception orientée objet**.

1.2. Pourquoi UML ?

De la même façon qu'il vaut mieux dessiner une maison avant de la construire, il vaut mieux modéliser un système avant de le réaliser.

- Obtenir une modélisation de très haut niveau indépendante des langages et des environnements.
- Faire collaborer des participants de tous horizons autour d'un même document de synthèse.
- Faire des simulations avant de construire un système.
- Exprimer dans un seul modèle tous les aspects statiques, dynamiques, juridiques, spécifications, etc...
- Documenter un projet.
- Générer automatiquement la partie logicielle d'un système.

2. Diagramme de cas d'utilisation :

Cette section présente une phase très importante pour le bon déroulement de notre plateforme c'est la phase d'analyse qui nous a permis de dégager les tâches principales fournies par notre plateforme.

2.1. Définition :

Les diagrammes de cas d'utilisation (DCU) sont des **diagrammes UML** utilisés pour une représentation du comportement fonctionnel d'un système **logiciel**. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet,

Cas d'utilisation : un cas d'utilisation (**use cases**) représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système.

2.2. Cas d'utilisation générale :

Description : Dans ce diagramme on a trois acteurs, chaque acteur à son propre Rôle par lequel interagit avec le système.

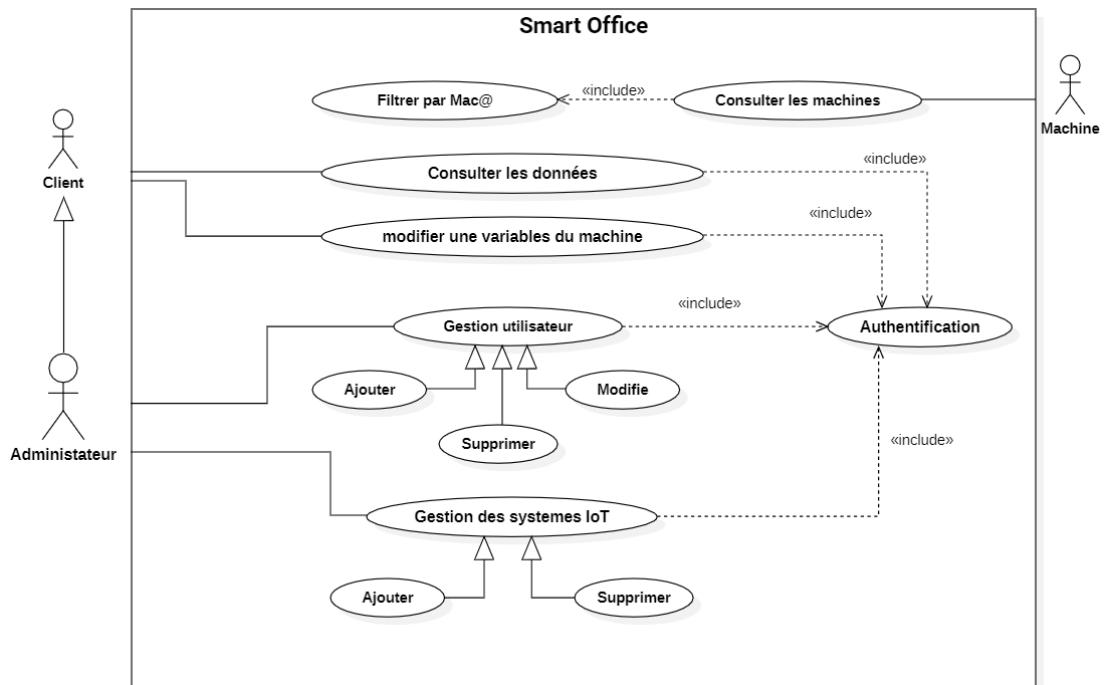


Figure 15 : Diagramme de cas d'utilisation générale

2.3. Cas d'utilisation d'un utilisateur :

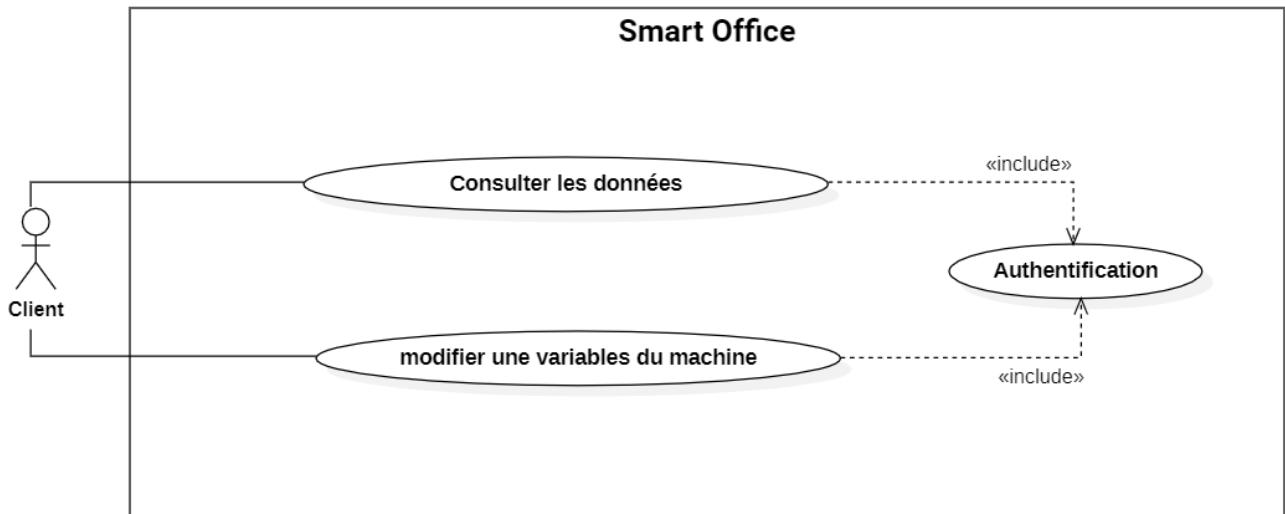


Figure 16 : Diagramme de cas d'utilisation d'un Utilisateur

Le client peut consulter les informations de ses machines telles que (le nom, l'adresse MAC, la date d'ajout, la date de modification, etc.), et les modifier après s'être authentifié.

Description générale des diagrammes de cas d'utilisation			
Utilisateur	Description	But	Précondition
	L'acteur "client" représente un utilisateur qui interagit avec la plateforme de gestion de systèmes IoT. Cet acteur a la capacité de modifier une variable de la machine et de consulter les données associées.	Exploiter les capacités de l'IoT pour répondre à ses besoins spécifiques en matière de gestion et de contrôle des objets connectés.	L'utilisateur doit premièrement s'authentifier.

Tableau 1 : Description de Cas d'utilisation d'un utilisateur

2.4. Cas d'utilisation d'un Administrateur :

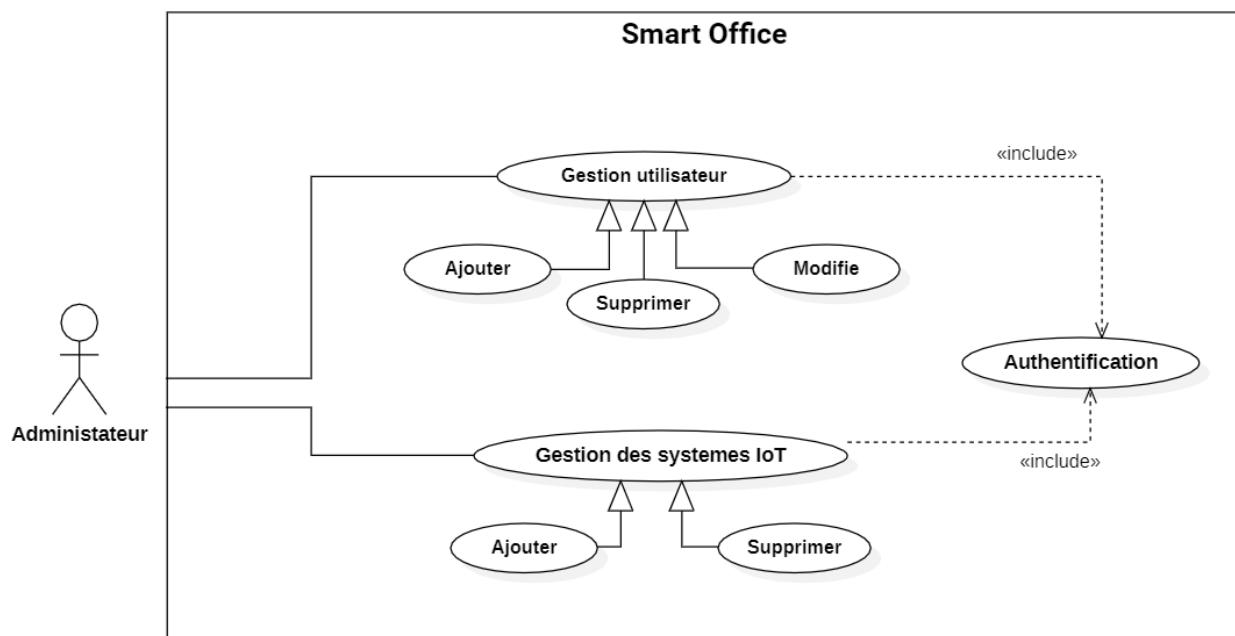


Figure 17 : Diagramme de cas d'utilisation d'un administrateur

La responsabilité de la gestion des utilisateurs et des machines incombe à l'administrateur, qui peut consulter les données une fois authentifié.

Description générale des diagrammes de cas d'utilisation			
	Description	But	Précondition
Administrateur	<p>L'acteur "Administrateur" représente un utilisateur ayant des privilèges étendus sur la plateforme. Il est responsable de la gestion des utilisateurs, des systèmes IoT, du contrôle des systèmes IoT et de la consultation des données associées.</p>	<p>Garantir la gestion efficace des utilisateurs, des systèmes IoT, du contrôle du système IoT et de la consultation des données sur la plateforme.</p>	<p>L'administrateur doit tout d'abord s'authentifier.</p>

Tableau 2 : Description de Cas d'utilisation d'un Administrateur

2.5. Cas d'utilisation d'une machine :

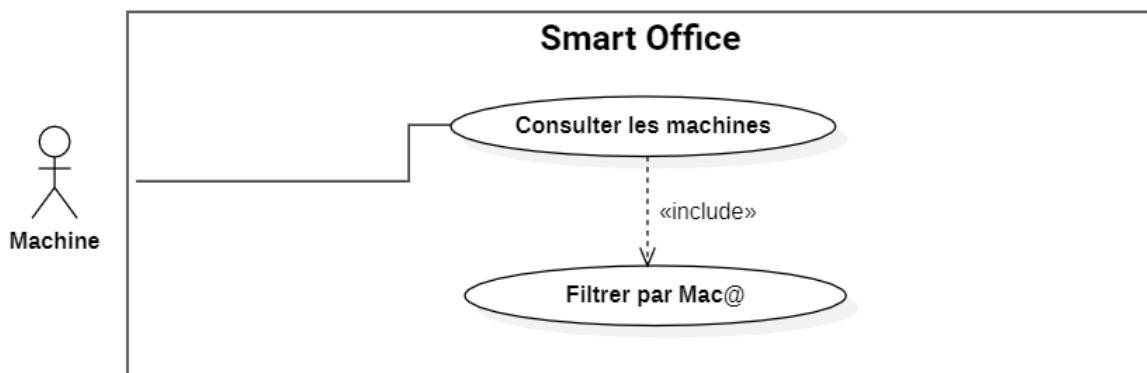


Figure 18 : Diagramme de cas d'utilisation d'une machine

La collecte des informations du système IoT spécifié par son adresse MAC

Description générale des diagrammes de cas d'utilisation			
	Description	But	Précondition
Machine	<p>L'acteur "Machine" dans la plateforme de gestion des systèmes IoT est responsable de la consultation, du filtrage et de l'obtention des variables associées aux machines spécifiques.</p>	<p>Il joue un rôle essentiel dans le suivi, la surveillance et le contrôle des machines connectées, offrant ainsi une gestion efficace et optimisée du système IoT dans son ensemble.</p>	<p>Disponibilité de l'API.</p>

Tableau 3 : Description de Cas d'utilisation d'une machine

3. Diagramme de séquence

Dans cette section on présente les diagrammes de séquence qui permettent de mieux comprendre le scénario de communication et d'interaction des utilisateurs avec le système.

3.1. Définition

Les diagrammes de séquences sont la représentation graphique des **interactions** entre les **acteurs** et le système selon un ordre chronologique dans la formulation **Unified Modeling Language**.

3.2. L'ajout d'un système IoT :

L'administrateur remplit les variables, l'adresse **MAC** et le **nom** du système via le **formulaire**, puis les données sont **stockées** dans la **base de données** pour une utilisation ultérieure. Cela permet à l'administrateur de gérer et de surveiller efficacement les systèmes IoT connectés à la plateforme.

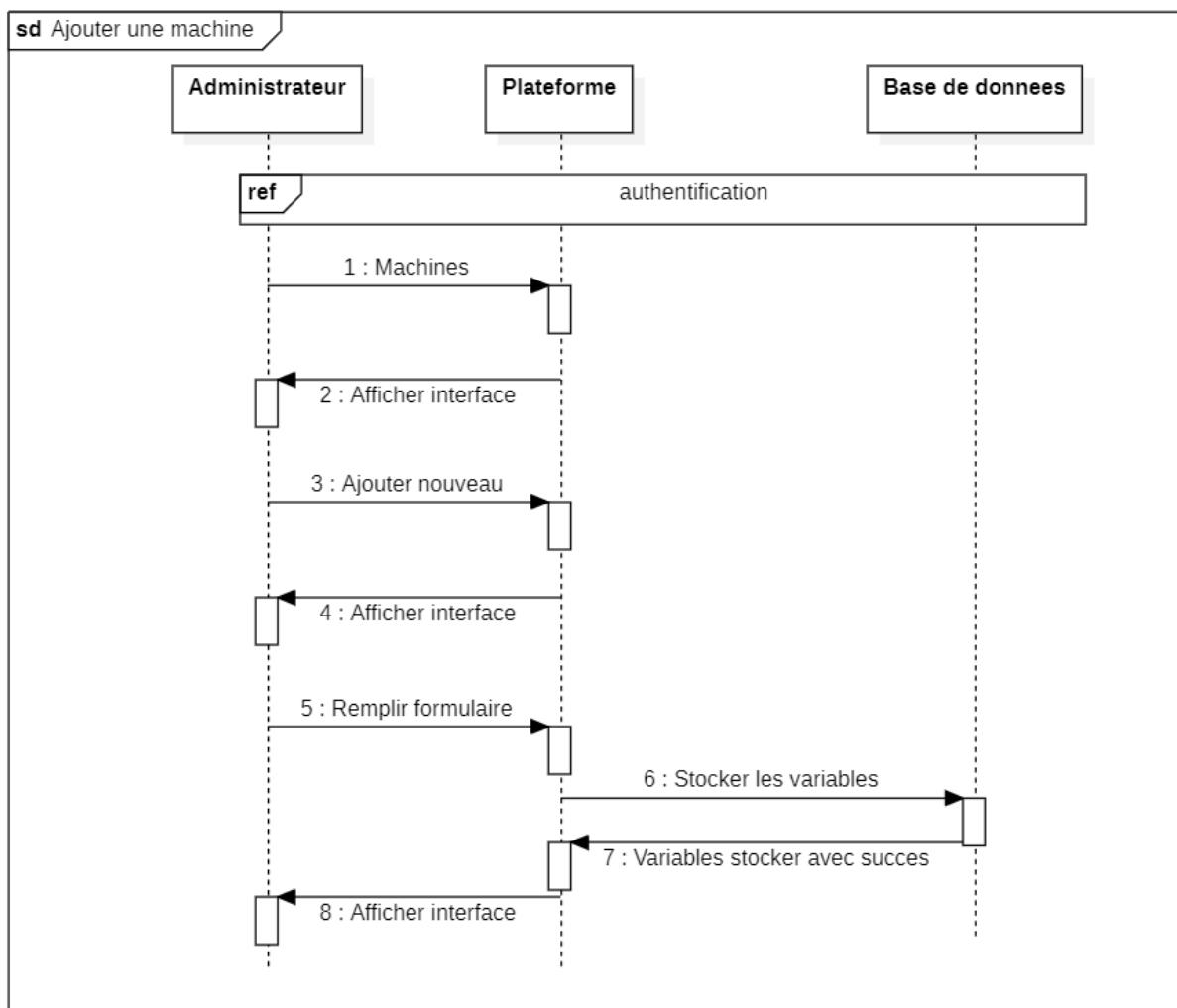


Figure 19 : Diagramme de séquence d'ajout d'un système IoT

Le diagramme de séquence illustre ainsi l'interaction entre l'utilisateur, l'interface de la plateforme et la base de données, montrant comment les modifications des variables de la machine sont capturées, stockées et reflétées dans une nouvelle interface pour l'utilisateur.

3.3. Consultation des variables d'une machine :

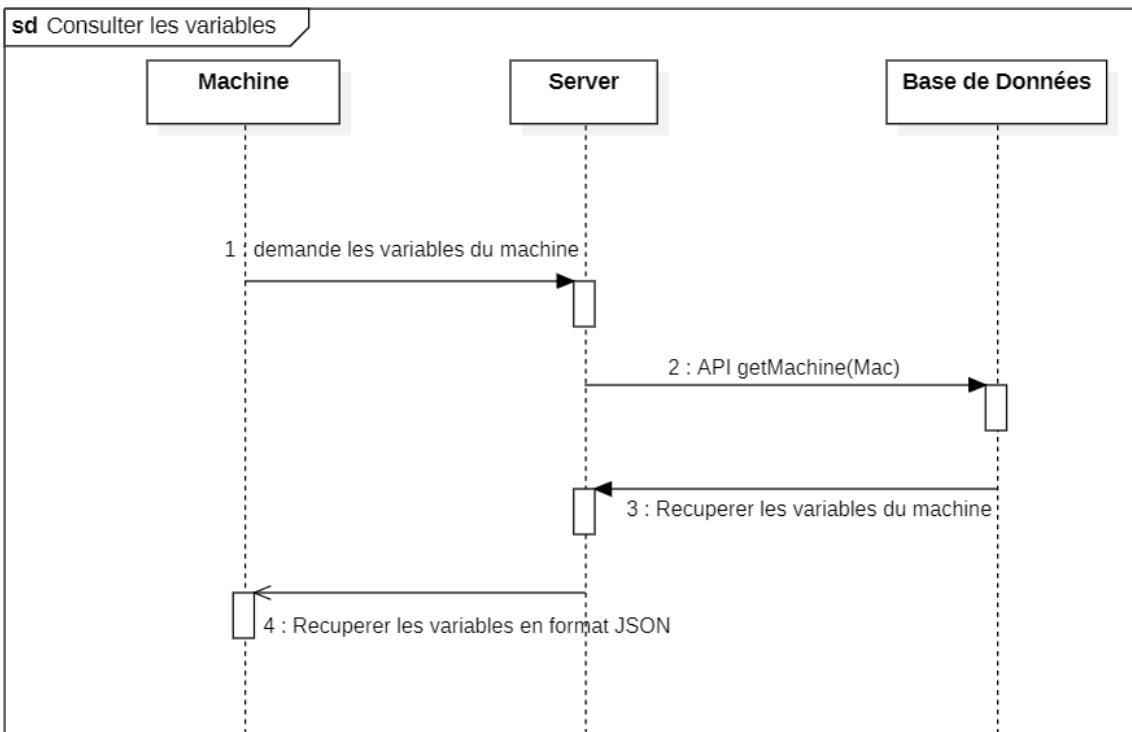


Figure 20 : Diagramme de Séquence : Consulter les machines

Ce diagramme de séquence illustre l'interaction entre la machine, le serveur et la base de données. La machine envoie une demande, le serveur interroge la base de données via une API, puis en convertissant le format JSON retourné par le serveur en un format utilisable par le système IoT grâce à un code Arduino spécifique.

3.4 Modification d'une variable de la machine :

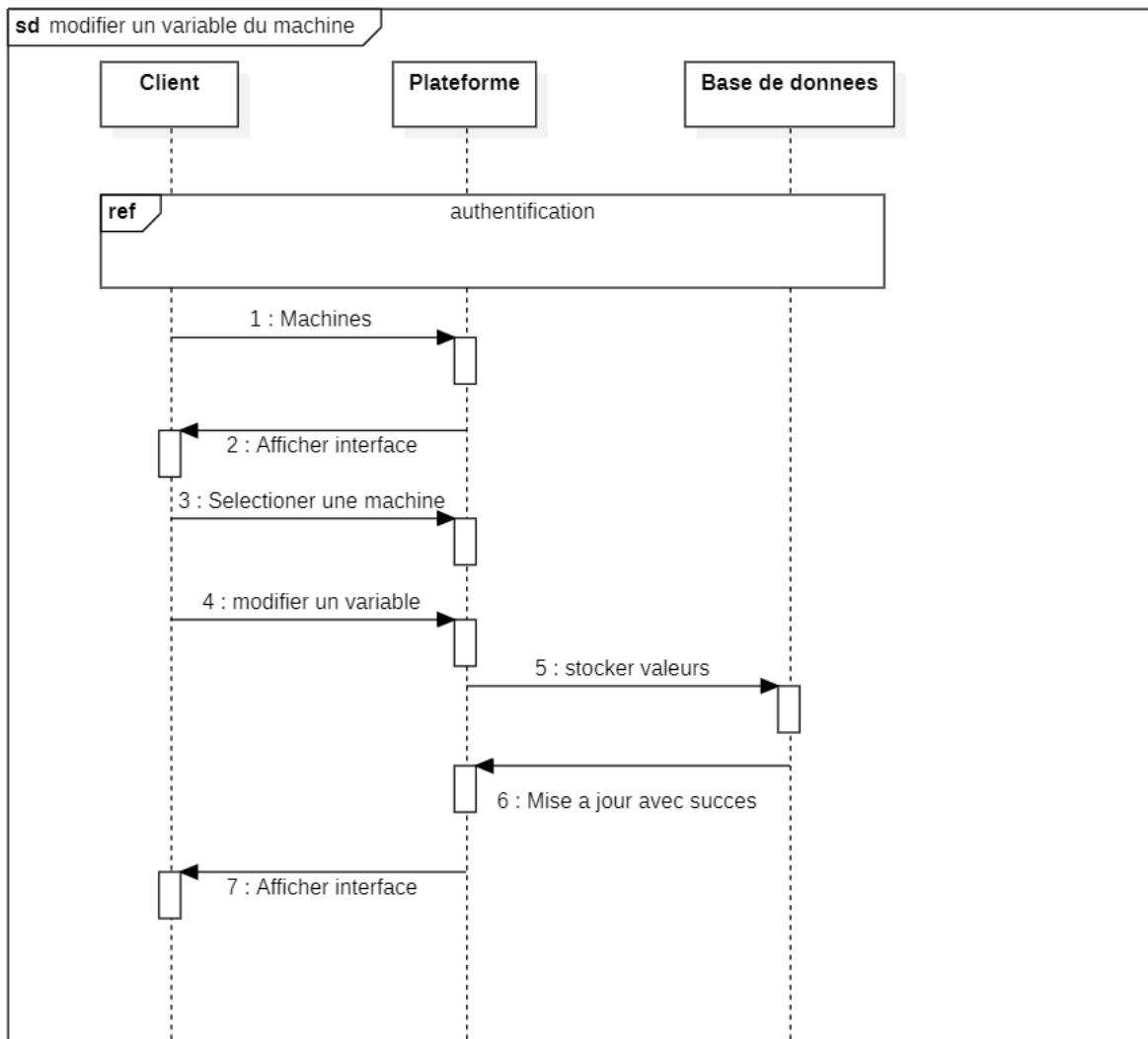


Figure 21: Diagramme de Séquence : modifier une variable de la machine

Le diagramme de séquence illustre ainsi l'interaction entre l'utilisateur, l'interface de la plateforme et la base de données, montrant comment les modifications des variables de la machine sont capturées, stockées et reflétées dans une nouvelle interface pour l'utilisateur.

4. Diagramme de classes :

4.1. Définitions :

Une **classe** est un ensemble de **fonctions** et de données (attributs) qui sont liées ensemble par un champ sémantique. Les classes sont utilisées dans la **programmation orientée objet**.

Elles permettent de modéliser un **programme** et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

4.2 Les Classes :

Les principales classes de notre platform sont :

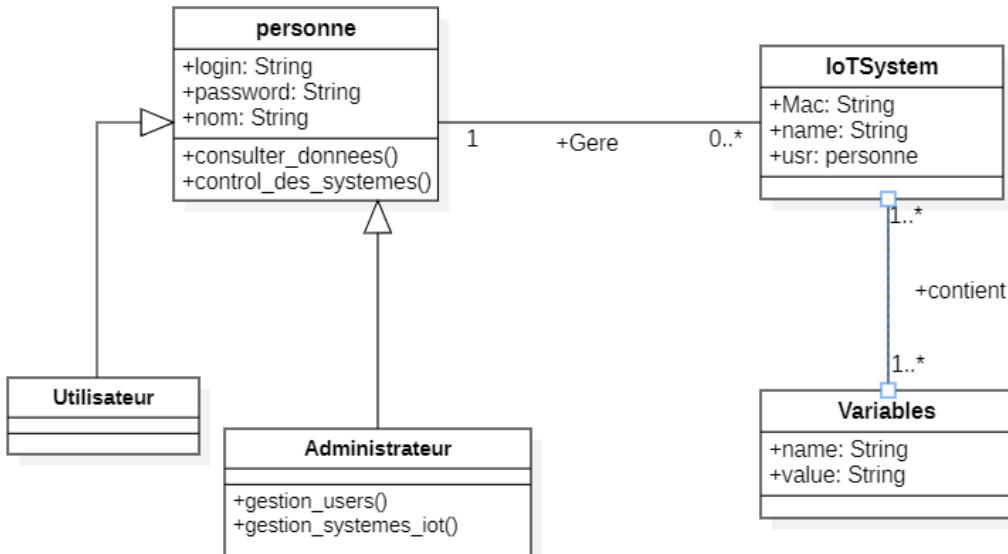


Figure 22 : Diagramme de Classe

- **Classe personne** : C'est la classe qui regroupe les attributs et les méthodes commun entre les classes.
- **Classe Utilisateur** : elle contient toutes les informations sur l'utilisateur.
- **Classe administrateur** : C'est la classe qui contient toutes les actions prises en charge par l'administrateur :
 - Gérer utilisateurs
 - Gérer systèmes IoT.
- **Classe IOT Système** : contient toutes les informations sur système iot développé.
- **Classe variable** : contient toutes les informations sur les variables du système iot développé.

Conclusion

Dans cette partie, on a présenté l'analyse et la conception de notre système, notamment les interactions entre l'utilisateur et les différents composants du système pour mieux comprendre le déroulement de notre projet. Cependant, Cette modélisation sera exploitée lors de la phase d'implémentation & déploiement.

Chapitre 4 : Implémentation & Déploiement



Introduction

Ce chapitre se concentre sur l'implémentation et le déploiement du projet. Nous présenterons le système Smart Light, son prototype, et aborderons le processus de déploiement. De plus, nous explorerons les interfaces de la plateforme IoT Smart Office, telles que l'accueil, l'authentification, le profil et le tableau de bord etc.

1. Smart light :

1.1. Présentation du système Smart Light :

Le système "Smart Lightning" est conçu pour offrir un éclairage intelligent et adaptatif en fonction de la luminosité ambiante. Cette partie du projet met en œuvre un circuit de commande ESP8266, qui est relié à quatre lampes, une photorésistance et un capteur de mouvement, tous câblés sur l'ESP8266. L'objectif est de réguler l'intensité des lampes en fonction des besoins. Le système offre deux modes : automatique et manuel.

- Dans le **mode automatique**, si une personne entre dans la pièce et que la luminosité ambiante est suffisante, toutes les lampes sont éteintes pour économiser de l'énergie. Lorsqu'il y a une luminosité modérée, deux lampes sont éteintes tandis que les deux autres sont allumées, offrant un éclairage doux et agréable. Enfin, lorsque l'environnement est sombre, toutes les lampes sont allumées pour assurer une visibilité optimale.
- Dans le **mode manuel**, l'utilisateur a la possibilité de choisir l'état des lampes via une plateforme. Si l'utilisateur choisit d'allumer les lampes, il peut également sélectionner le nombre de lampes à allumer.

Grâce à cette approche intelligente, le système "Smart Lightning" offre un éclairage adapté à chaque situation, tout en favorisant l'efficacité énergétique.

1.2. Prototype du système :

a. Matériels nécessaires :

Matériels	
Motion sensor	
Photorésistance	
4 * LEDs	
Résistances	
Breadboard électrique	
Wires	
ESP8266 NodeMCU	

Figure 23 : Matériels nécessaire pour le système Smart Light

b. Simulation avec circuito.io :

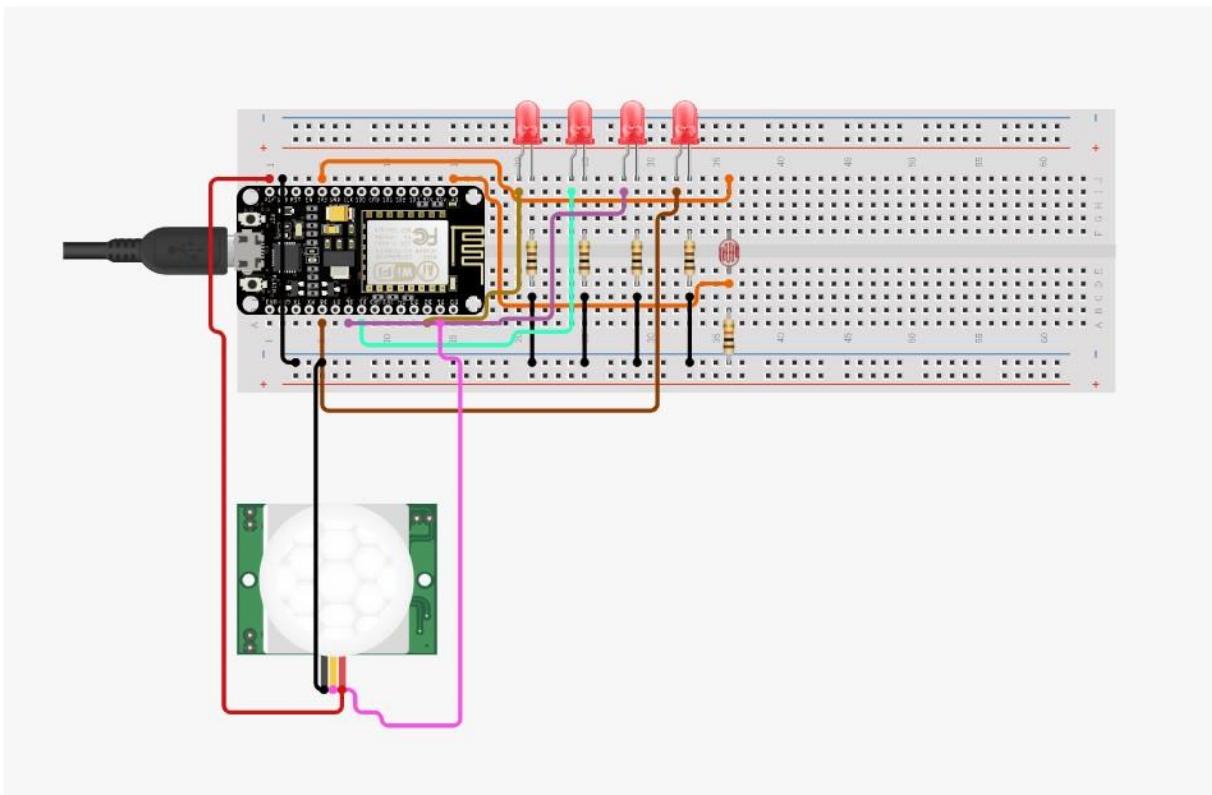


Figure 24 : Simulation de Système Smart Light

Utilisant [circuito.io](#), nous construisons un circuit d'éclairage intelligent avec les matériaux que nous avons définis dans **figure 23**.

c. Résultat :

- 1 er Cas : **Mode automatique** : détection de mouvement dans un environnement de faible luminosité



Figure 25 : En mode automatique, détection de mouvement dans un environnement sombre.

- 2ème Cas : **Mode automatique** : détection de mouvement dans un environnement de luminosité modérée.

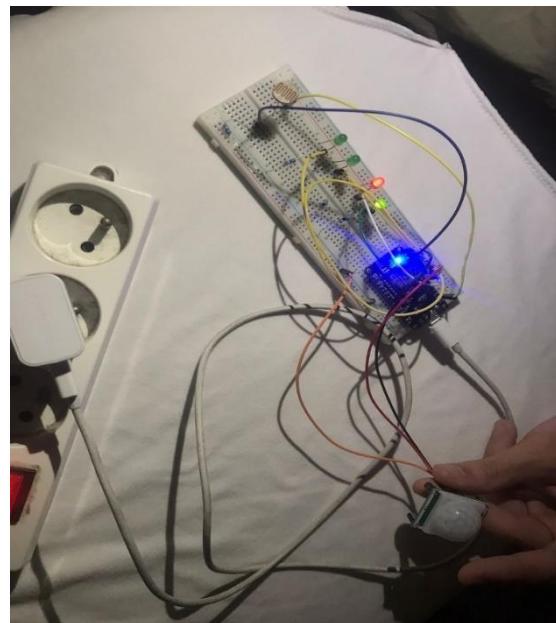


Figure 26 : Mode automatique - Détection de mouvement dans un environnement de luminosité modérée.

- 3ème Cas : **Mode manuel** : sélectionnez le nombre de lampes à allumer.

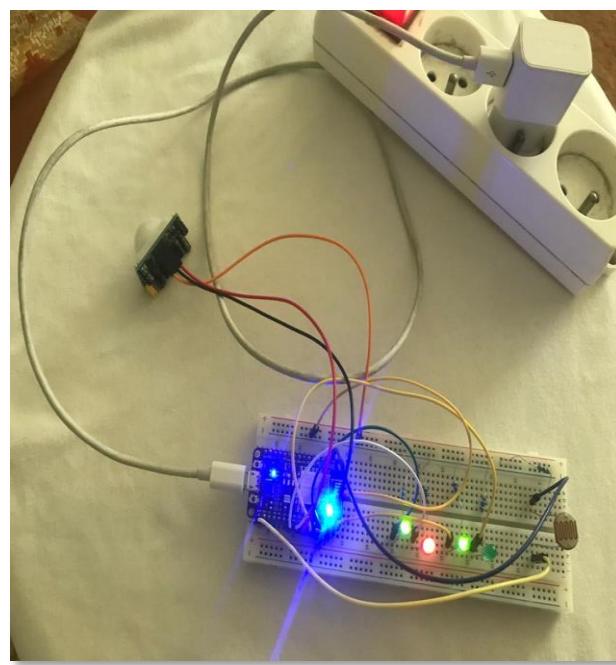


Figure 27 : Mode manuel - Sélection du nombre de lampes à allumer.

2. Déploiement :

2.1. Le pour quoi ?

- **Connectivité** : établir une connexion avec les systèmes IoT utilisant le module WiFi ESP8266, ce qui permet la communication et l'échange de données entre les deux. Cette connectivité est essentielle pour obtenir les fonctionnalités souhaitées.
- **Contrôle centralisé** : En déployant votre projet, vous pouvez centraliser le contrôle et la gestion des différents appareils IoT dans votre bureau intelligent. Cela garantit une surveillance et une coordination efficaces, permettant des opérations rationalisées et une productivité améliorée.
- **Échange de données** : Le déploiement facilite l'échange de données en temps réel entre la plateforme et le module ESP8266. Cela permet des mises à jour instantanées, des notifications et une synchronisation des informations, contribuant à une automatisation accrue et à une réactivité améliorée au sein de votre environnement de bureau intelligent.
- **Accès à distance** : accessibilité à distance, permettant aux utilisateurs autorisés de contrôler et de surveiller le système de bureau intelligent depuis n'importe où. Cette flexibilité est particulièrement bénéfique pour la gestion à distance, la résolution de problèmes et la fourniture d'assistance à distance aux utilisateurs.
- **Scalabilité** : ajouter facilement plus d'appareils ou d'étendre les fonctionnalités de votre système de bureau intelligent. À mesure que votre projet se développe, le déploiement facilite l'intégration de nouveaux appareils, API et fonctionnalités, fournissant une base pour les améliorations et les adaptations futures.

2.2. Le Comment ?

Utilisant un **serveur mutualisé**, nous louons un espace sur un serveur qui est utilisé par plusieurs utilisateurs. La logique derrière un serveur partagé repose sur le partage des ressources matérielles et logicielles entre ces différents utilisateurs. Chaque utilisateur dispose d'un espace dédié pour stocker ses fichiers et exécuter ses applications, en veillant à ce que les ressources nécessaires soient disponibles pour répondre aux besoins du projet. Bien que le serveur partagé puisse offrir une puissance de calcul et une bande passante limitées, il convient parfaitement à un projet à petite échelle. En surveillant attentivement les performances et en optimisant les ressources utilisées par notre projet, nous pouvons nous assurer que le serveur partagé répondra aux exigences de notre application sans problème.



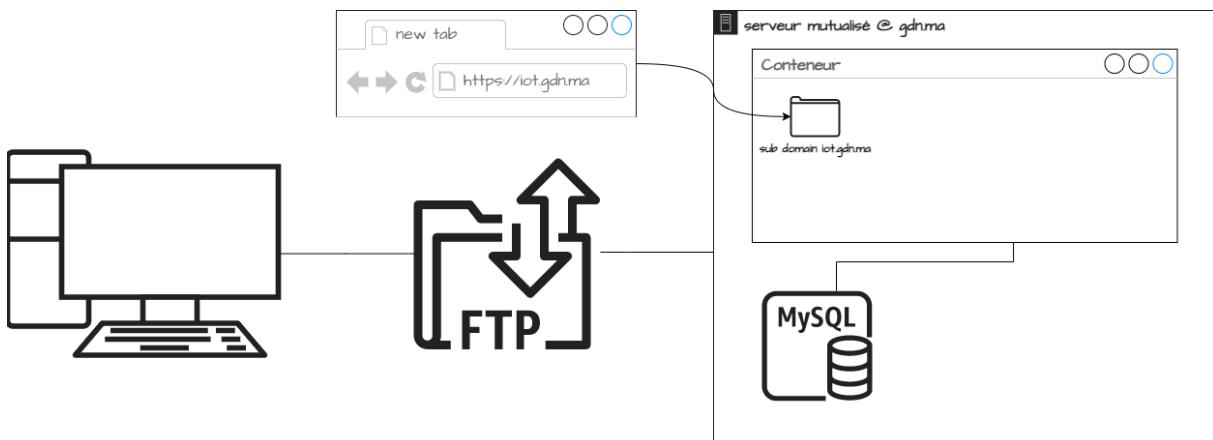


Figure 25 : Description de Déploiement

Pour déployer notre projet, nous utilisons FTP pour transférer les fichiers du projet du PC vers le serveur partagé via logiciel de transfert de fichiers (**FTP**) **FileZilla Client** comme se référer au schéma (**figure 28**). Le serveur partagé, associé au domaine "gdn.ma", héberge plusieurs conteneurs pour différents projets, y compris le nôtre. À l'intérieur du serveur partagé, nous avons un conteneur spécifiquement dédié à notre projet.

De plus, nous avons configuré un sous-domaine appelé "iot.gdn.ma" qui pointe vers le dossier de notre projet à l'intérieur du conteneur du serveur partagé. Ce sous-domaine agit comme une adresse spécialisée pour accéder à nos fonctionnalités liées à l'IoT.

En outre, le serveur partagé comprend également une base de données MySQL, utilisée pour stocker et gérer les données de notre projet de bureau intelligent IoT. L'intégration de la base de données MySQL permet un stockage et une récupération efficaces des données au sein de notre application.

3. Les Interfaces de Platform IoT [Smart office] :

3.1. Accueil :

Home – La page d'accueil du rapport présente un aperçu succinct des principaux objectifs de la plateforme Authentification

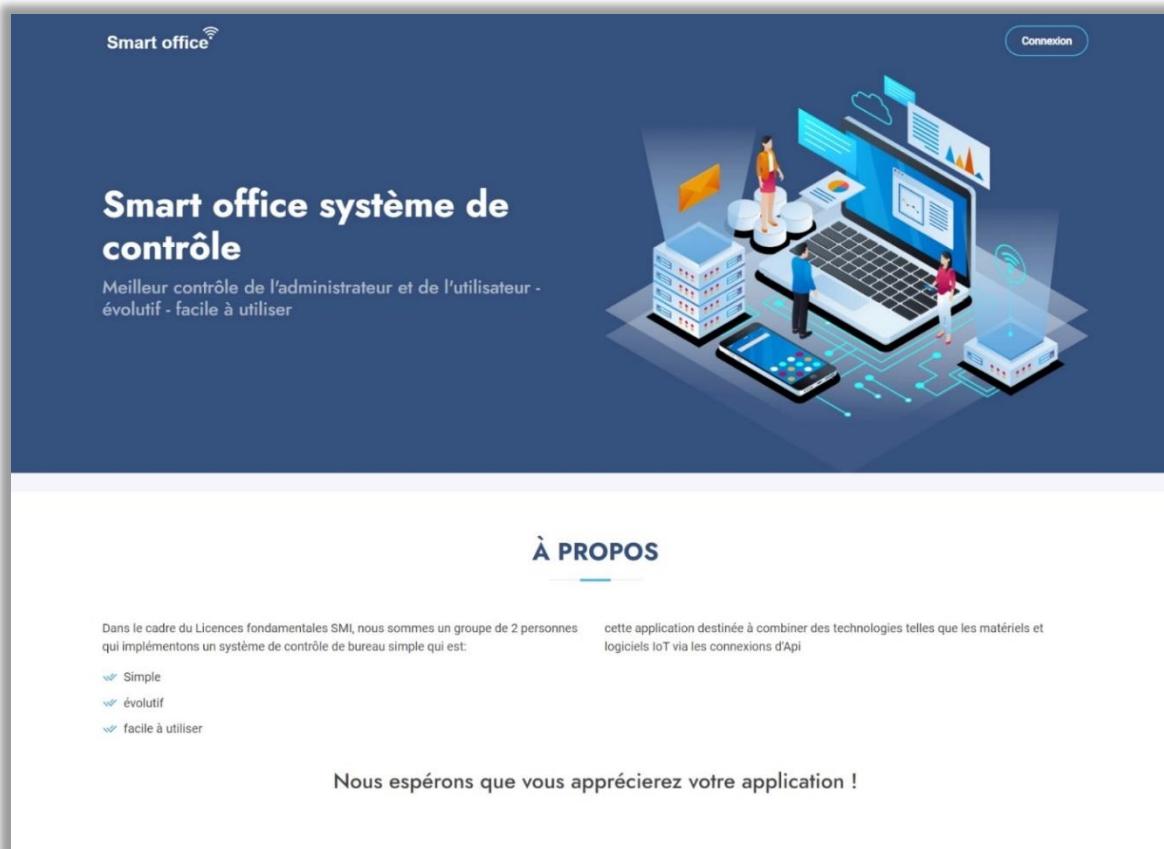


Figure 26 : Page d'accueil

3.2. Authentication :

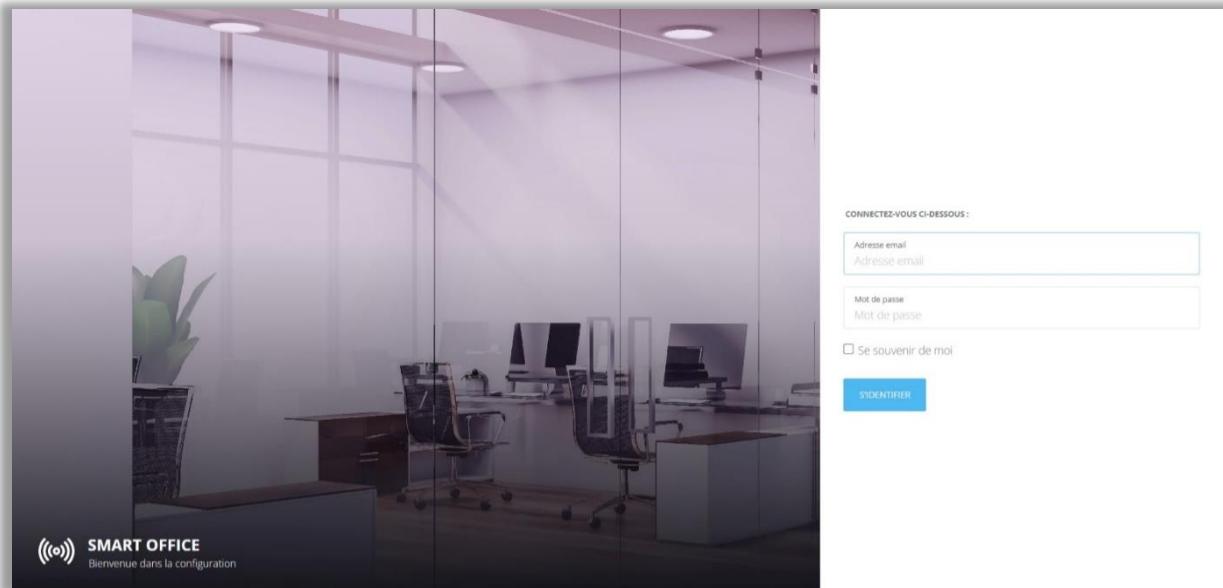


Figure 30 : Page d'authentification

L'authentification de chaque utilisateur avant la connexion, permettant ainsi de distinguer les administrateurs des utilisateurs normaux. Une fois authentifiés, les utilisateurs seront redirigés vers leur espace respectif (espace administrateur ou espace utilisateur normal)

3.3. Profil :

a. Admin :

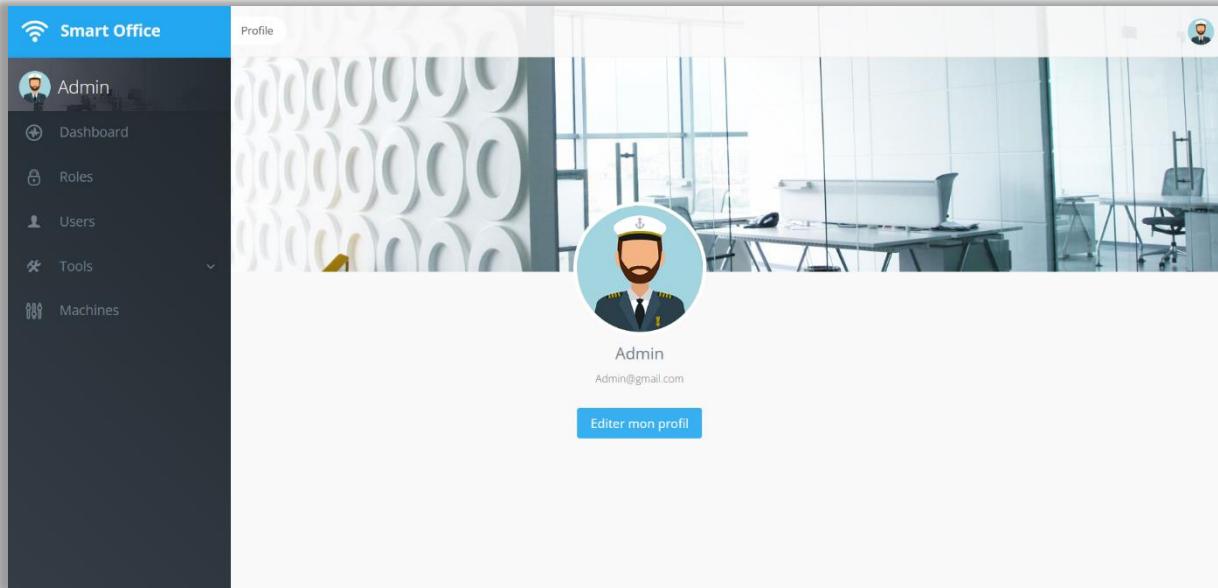


Figure 27 : Profil d'un Administrateur

Profil d'administrateur a l'accès à toutes les fonctionnalités de platform et d'interrogé onglets comme (tableau de board – rôles – utilisateurs – machines etc.)

b. Utilisateur :

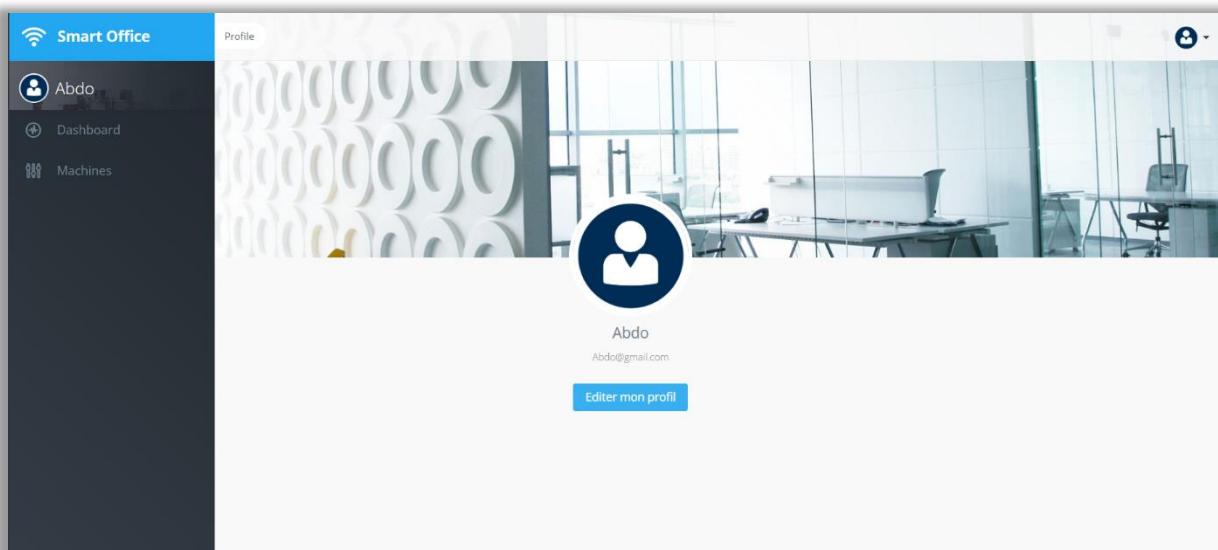


Figure 28 : Profil d'un Utilisateur

Profil d'un utilisateur a l'accès juste pour interroge son tableau de board & ses propre machines

3.4. Dashboard :

a. Admin :

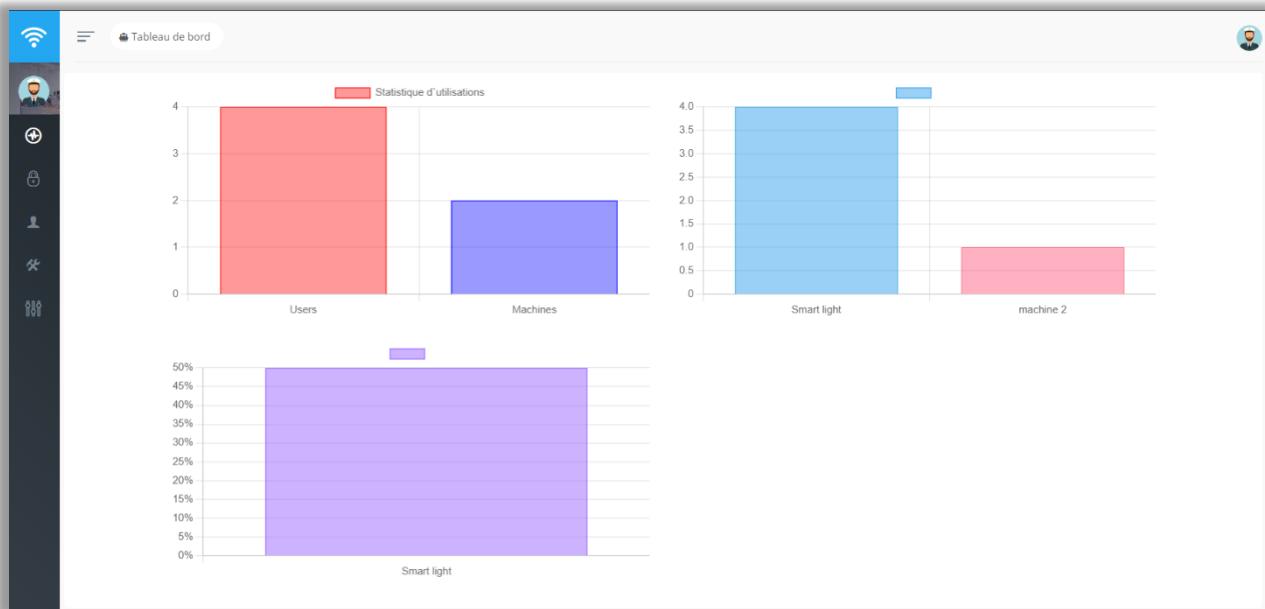


Figure 29 : Dashboard d'administrateur

L'administrateur consulter l'état courant des machines et nombre des utilisateurs et systèmes IoT inscrit

b. Utilisateur :

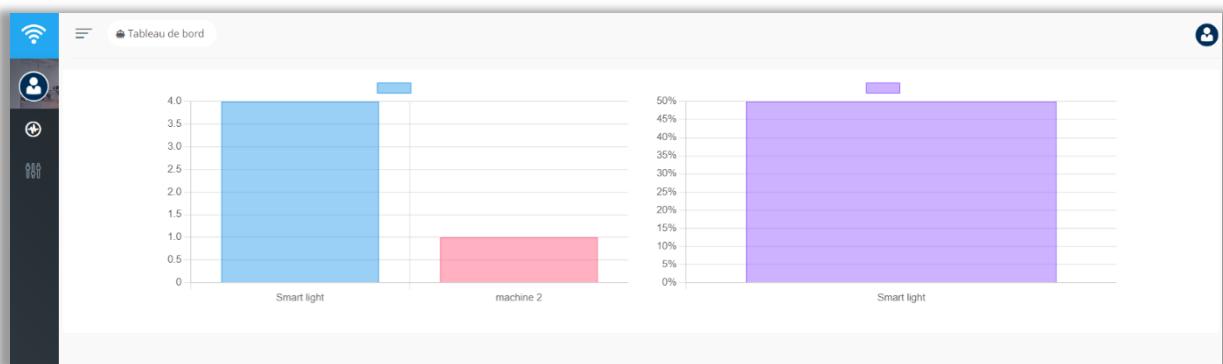


Figure 30 : Dashboard d'utilisateur

L'utilisateur peut consulter l'état des machines dans des graphes

3.5. Les Rôles :

Name	Display Name	Actions
admin	Administrator	Vue Editer Supprimer
user	Normal User	Vue Editer Supprimer

Figure 31 : List des Rôles

L'administrateur a le droit de consulter, ajouter, supprimer, modifier les différents rôles et personnaliser les rôles

3.6. Les Utilisateurs :

Name	Email	Created At	Email Verified At	Avatar	Role	Roles	Actions
abdo	abdo@gmail.com	2023-05-10 18:51:02			Normal User	Normal User	Vue Editer Supprimer
karim	karim@gmail.com	2023-05-02 14:44:08			Normal User	Normal User	Vue Editer Supprimer

Figure 32 : Liste des utilisateurs

L'administrateur a la possibilité de consulter la liste des utilisateurs et de gérer l'ensemble des utilisateurs du système.

3.7. Les Machines [Systèmes IoT] :

a. Admin :

	Name	Mac	utilisateur	Created At	Updated At	Auto	State	Slider	Value	String	Actions
<input type="checkbox"/>	Smart light	FF:FF:FF:00:00:00	abdo	2023-05-10 20:11:02	2023-05-10 22:45:58	Automatique	true	10%	80		<button>Vue</button> <button>Editer</button> <button>Supprimer</button>

Figure 33 : Configuration des machines pour un administrateur

L'administrateur a la possibilité d'ajouter ou de supprimer des machines, ainsi que de modifier les propriétés d'une machine. Pour supprimer une machine, il suffit de cocher la machine correspondante et d'appuyer sur le bouton "Supprimer machine".

Ajouter Machine

Name	<input type="text" value="Name"/>
Mac	<input type="text" value="Mac"/>
utilisateur	<input type="text" value="Aucun"/> Aucun Admin karim abdo

Figure 37 : Insère une machine

Les données essentielles à saisir sont l'adresse MAC et les propriétés de la machine afin d'établir une connexion et déterminer la portée de la machine.

b. Utilisateur :

The screenshot shows a user interface for managing IoT machines. At the top, there's a navigation bar with icons for Wi-Fi, user profile, and other system status. Below it, a sidebar has a 'Machines' icon. The main area is titled 'Machines' and displays a table of machine configurations. The columns are: Name, Mac, utilisateur, Created At, Updated At, Auto, State, Slider, Value, String, and Actions. A single row is shown for a 'Smart light' machine, with values: FF:FF:FF:00:00:00, abdo, 2023-05-10 20:11:02, 2023-05-10 22:45:58, Automatique, true, 10%, 80. Buttons for 'Vue' and 'Editer' are at the bottom right of the table. Below the table, a message says 'Affichage de l'élément 1 à 1 sur 1 éléments'. At the bottom right, there are navigation buttons for 'Précédent', a page number '1', and 'Suivant'.

Figure 34 : Configuration des machines pour un utilisateur

Consulter la liste des machines [systèmes IoT] est voire l'états courant, modifier les variables

This screenshot shows a detailed view of a machine configuration. The top navigation bar and sidebar are identical to Figure 34. The main area is titled 'Affichage Machine' and contains a form with the following fields and values:

Name	Smart light
Mac	FF:FF:FF:00:00:00
utilisateur	abdo
Created At	2023-05-10 20:11:02
Updated At	2023-05-10 22:45:58
Auto	Automatique
State	true
Slider	10%
Value	80

At the top right of the form, there are 'Editor' and 'Retourner à la liste' buttons.

Figure 35 : Affichage des Information d'une machine

Utilisateur a consulté les données de propre machine

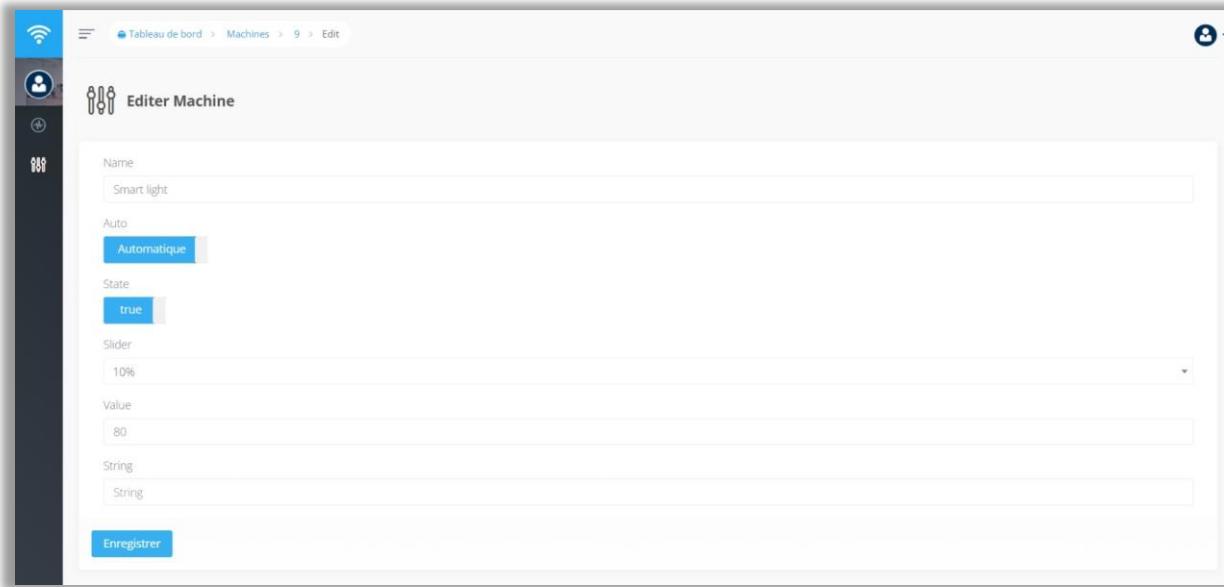


Figure 36 : Modifie les données d'une machine

L'utilisateur a possibilité de modifier les valeurs des variables du mesure ou nom de machine

Conclusion

Dans cette partie, nous avons abordé le système Smart Light en présentant les différents matériaux utilisés, ainsi qu'une description du montage et du fonctionnement. De plus, conformément à la section de modélisation, nous avons conçu une interface conviviale au sein de notre plateforme qui permet une interaction fluide avec les systèmes IoT. L'accent sur une expérience utilisateur simple, permettant aux utilisateurs de surveiller et de contrôler les composants du bureau intelligent.

Conclusion et perspectives

En conclusion, ce projet de bureau intelligent avec un système d'éclairage représente une solution efficace et innovante pour améliorer l'efficacité énergétique et le confort dans les environnements de travail.

Grâce à l'utilisation des technologies de l'Internet des objets et à l'intégration des données, notre système offre une gestion optimale de l'éclairage, répondant aux besoins des utilisateurs.

Au cours de ce projet, nous avons acquis des connaissances clés dans le domaine de l'IoT, notamment en ce qui concerne l'intégration des données, la connectivité des appareils et la gestion des utilisateurs.

Finalement, à mesure que la technologie continue d'évoluer, cela ouvre également de nouvelles perspectives pour l'avenir du bureau intelligent et ses applications potentielles dans les environnements de travail modernes en intégrant de nouvelles technologies à notre projet. Par exemple, l'intégration d'algorithmes d'apprentissage automatique ML, permettrait des analyses prédictives et une automatisation améliorée. De plus, une autre option à considérer serait l'utilisation d'une base de données NoSQL, qui offrirait une plus grande flexibilité et évolutivité pour gérer les données générées par les dispositifs IoT.

Webographie

Guide des capteurs, technologies liées à l'IoT :

<https://youtube.com/@AndreasSpiess>

Tutoriel Arduino :

<https://youtube.com/@paulmcwhorter>

<https://www.tutorialspoint.com/arduino/index.htm>

<https://www.arduino.cc/>

Fiches techniques des composants :

<https://drive.google.com/drive/folders/1peMmWkyiqBSVM5xcDzLPa8S79UmA7mlN?usp=sharing>

ChatGPT :

<https://chat.openai.com/>

Tutoriel Laravel :

<https://laravel.com/docs/10.x>

<https://youtu.be/MYyJ4PuL4pY>

Tutoriel Bootstrap :

<https://getbootstrap.com/docs/5.3/getting-started/introduction/>

Api fictive pour les tests :

<https://jsonplaceholder.typicode.com/>

Débogage et test du module Wifi ESP :

<https://room-15.github.io/blog/2015/03/26/esp8266-at-command-reference/>