



RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITÉ DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENE

Faculté d'Electronique et d'Informatique

Département Informatique

MÉMOIRE DE LICENCE

Spécialité

GÉNIE DES TÉLÉCOMMUNICATIONS ET RÉSEAUX

Thème

L'implémentation et la réalisation d'un système
générique d'Internet des objets
pour la surveillance et le contrôle à distance

Proposé et Dirigé par :

M.BOUALOUACHE
Abdelwahab

Réalisé et présenté par :

Mr. DJIBRILLA BOUKARY Yacine
Mr. KISSOUM Mohand Ouali

Devant le jury composé de :

M BELKHIR Abdelkader Président
Mme AISSAOUI Amel Membre

Soutenu le : 14/07/2019

Binôme N° : 47/2019

REMERCIEMENTS

Tout d'abord nous remercions **ALLAH** le tout puissant, le miséricordieux qui nous a donné la force, le courage et la volonté pour achever ce modeste travail.

Nous remercions nos parents qui sans leurs sacrifices et leurs efforts, nous ne saurons jamais arrivés où nous sommes aujourd'hui.

Nous tenons à exprimer notre profonde gratitude à notre encadreur **M.BOUALOUACHE Abdelwahab**, pour sa disponibilité, son aide précieuse et ses conseils judicieux.

Nos vifs remerciements à toute la **famille USTHB**, allant du recteur de l'université jusqu'au plus simple travailleur, particulièrement le corps enseignant et administratif pour la formation de qualité qu'ils nous ont délivré tout au long de notre cursus.

Enfin, un grand merci à tous **nos amis** et à toute personne ayant contribué de près ou de loin à la réalisation de ce travail.

KISSOUM Mohand Ouali

&

DJIBRILLA BOUKARY Yacine

DÉDICACE

Je tiens à dédier ce modeste travail à tous ceux qui m'ont encouragé durant toute la période de réalisation de ce travail. En particulier :

- A mes chers parents qui se sacrifient pour me voir réussir.
- A mes chers frères pour leurs conseils et leurs soutiens.
- A mes amis, merci à tous mes amis avec qui je partage des moments de ma vie au fil du temps.

DÉDICACE

Je dédie ce mémoire à :

- Mes très chers parents, qui m'ont accompagné depuis mes tout premiers pas jusqu'à la date d'aujourd'hui, pour leurs innombrables sacrifices et qui ont fait de moi ce que je suis présentement .
- A toute ma famille, pour leur soutien tout au long de mon parcours scolaire et universitaire.
- A tous mes amis pour leur accompagnement et leurs conseils.

DJIBRILLA Boukary Yacine

Table des matières

Table des figures	VI
Liste des tableaux	VII
Liste des abréviations	VIII
Introduction générale	1
I Généralités sur l'internet des objets	2
I.1 Introduction	3
I.2 Généralités sur l'internet des objets	3
I.2.1 Définition	3
I.2.2 Fonctionnement	3
I.2.3 Évolutions de l'écosystème de l'IoT	3
I.3 Architecture et protocoles de communications	4
I.3.1 Architecture	4
I.3.2 Technologies de communications	5
I.3.3 Les protocoles applicatifs	6
I.4 Domaines d'application	9
I.4.1 Santé et médecine	9
I.4.2 Transport et logistique	9
I.4.3 Industrie	9
I.4.4 Sport	9
I.4.5 Domotique	9
I.4.6 Energie	9
I.4.7 Agriculture	9
I.5 Contraintes liées à l'usage de l'IoT	10
I.5.1 Scalabilité	10
I.5.2 Hétérogénéité	10
I.5.3 Influence du monde physique	10
I.5.4 Sécurité et vie privée	10
I.6 Conclusion	10
II Conception	11
II.1 Introduction	12
II.2 Problématique et solution	12
II.2.1 Problématique	12
II.2.2 Solution	12
II.3 Architecture de la solution proposée	13
II.4 Partie matérielle et communication	13
II.4.1 Connexion au serveur MQTT	13
II.4.2 Communication MQTT	14

II.5	Partie logicielle	15
II.5.1	Fonctionnalités	15
II.5.2	Diagramme de cas d'utilisation	16
II.5.3	Diagramme de séquences	22
II.5.4	Diagramme de classes	26
II.5.5	Les règles de passage du diagramme de classe vers le modèle relationnel	27
II.5.6	Le modèle relationnel des données	28
II.6	Conclusion	29
III	Réalisation et tests	30
III.1	Introduction	31
III.2	Outils et environnement de travail	31
III.2.1	Partie matérielle	31
III.2.2	Partie logicielle	33
III.3	L'application web	35
III.3.1	Présentation de l'application web	35
III.3.2	Interfaces graphiques	35
III.4	Tests effectués	41
III.4.1	Schéma matériel	41
III.4.2	Scénarios testés	42
III.5	Conclusion :	45
Conclusion générale		46
A		I
Bibliographie		IV

Table des figures

Figure I.1	: L'évolution du nombre d'objets connectés	4
Figure I.2	: Architecture générale d'un système d'internet des objets	5
Figure I.3	: Fonctionnalité du protocole CoAP.	7
Figure I.4	: Publication/Souscription au niveau d'un serveur MQTT	8
Figure II.1	: Architecture du système	13
Figure II.2	: Connexion au serveur MQTT	14
Figure II.3	: Mecanisme de publication/souscription MQTT	15
Figure II.4	: Diagramme de cas d'utilisation d'un chef d'entreprise	20
Figure II.5	: Diagramme de cas d'utilisation d'un chef de projet	21
Figure II.6	: Diagramme de cas d'utilisation d'un technicien	22
Figure II.7	: Diagramme de cas d'utilisation d'un utilisateur ordinaire	22
Figure II.8	: Diagramme de séquence d'authentification	23
Figure II.9	: Diagramme de séquence d'ajout d'un utilisateur	24
Figure II.10	: Diagramme de séquence d'ajout d'un projet	24
Figure II.11	: Diagramme de séquence de la visualisation des données	25
Figure II.12	: Diagramme de séquence de commande d'actionneur	25
Figure II.13	: Diagramme de classe	26
Figure II.14	: Schémas relationnel des données	29
Figure III.1	: Le module NodeMCU Lolin ESP8266	31
Figure III.2	: Authentification	36
Figure III.3	: Ajout d'un utilisateur	37
Figure III.4	: Liste des projets	37
Figure III.5	: Modifier un utilisateur	38
Figure III.6	: Collecte des métriques en temps réel	39
Figure III.7	: Graphe de variation de l'humidité en fonction de temps	39
Figure III.8	: Graphe de variation de la température en fonction de temps	40
Figure III.9	: Graphe de variation de la luminosité en fonction de temps	40
Figure III.10	: Espace Actionneurs	41
Figure III.11	: Architecture matérielle	42
Figure III.12	: Montage collecte des métriques	43
Figure III.13	: Montage commande des actionneurs à distance	43
Figure III.14	: Montage détection de mouvement	44
Figure III.15	: Détection de mouvement	44
Figure III.16	: Montage allumage d'une lampe commandé par un seuil	45
Figure III.17	: Allumage d'une lampe commandé par un seuil	45

Liste des tableaux

Table II.1	: Description des fonctionnalités des acteurs de système	17
Table II.2	: Description du cas d'utilisation « Authentification »	18
Table II.3	: Description du cas d'utilisation gestion des projets	18
Table II.4	: Description du cas d'utilisation gestion des utilisateurs	19
Table II.5	: Description du cas d'utilisation « mise à ON ou à OFF des actionneurs	19
Table II.6	: Liste des attributs associés à chaque classe	27

Liste des abréviations

AMQP : Advanced Message Queuing Protocol

BLE : Bluetooth low energy

CSS : Cascading Style Sheets.

CoAP : Constrained Application Protocol

JSON : JavaScript Object Notation

IDE : Integrated Development Environment

IoT : INTERNET of THINGS

IdO : Internet des Objets

IP : Internet Protocol

HTML : Hypertext Markup Language

HTTP : HyperText Transfer Protocol

LAN : Local Area Network

LED : Light Emitting Diode

MySQL/SGBDR : système de gestion de bases de données relationnelles.

M2M : machine to machine

MQTT : Message Queuing Telemetry Transport

NFC : Near Field Communication

PHP : Pré Hypertext Processor.

REST : representational state transfer

RFID : Radio Frequency Identification

TCP : Transmission Control Protocol

UDP : User Datagram Protocol

UML : Unified Modeling Language

XMPP : Extensible Messaging and Presence Protocol

6LowPAN : IPv6 Low power Wireless Personal Area Networks

Introduction générale

L'avènement de l'internet est sans doute la chose qui a le plus bouleversé le monde et changer notre mode de vie quotidien. Au cours de son évolution, l'internet nous offre une multitude de services en allant des simples communications entre machines aux réseaux sociaux et du E-commerce au E-learning,etc....

Aujourd'hui l'internet est passé à un autre niveau grâce à l'avènement de l'internet des objets. Il est en effet possible grâce à cette nouvelle technologie de faire communiquer des objets de tous genres du monde physique et d'interagir avec eux à travers des applications web et embarquées ;

De nombreux spécialistes prévoient, pour les années à venir, une explosion du nombre des objets connectés dans les pays industrialisés, en particulier dans le contexte d'applications liées au confort, aux loisirs, à la qualité de vie et à la santé. Les pays émergents sont également susceptibles de profiter de ce bouleversement en raison de l'existence de nombreuses applications générant des besoins forts (agriculture raisonnée, énergies renouvelables...). Parmi ces objets connectés se trouvent des éléments fixes, monitorant l'environnement, et des éléments mobiles, portés par un humain, attachés à du bétail ou encore à des véhicules intelligents.

Ces différentes classes forment des réseaux séparés car elles utilisent des protocoles de communication et des technologies souvent non interopérables. L'objectif de l'IoT est de les faire collaborer en les dotant tous de la capacité de communiquer via Internet. Cette révolution se situe à plusieurs niveaux de compétences et ouvre de grandes opportunités de contributions. Elle est associée au concept de communications de Machine à Machine (M2M) où de nombreux travaux sont en cours.

Dans ce mémoire nous nous proposons de réaliser un système générique d'internet des objets permettant la surveillance et le contrôle à distance d'un ensemble de sites géographiques. Ce mémoire est divisé en trois chapitres organisés comme suit :

Un premier chapitre nous permettant de découvrir le monde de l'internet des objets en amont et en aval. Un deuxième chapitre consacré à l'analyse et la conception du système. Un dernier chapitre intitulé « réalisation et tests » destiné à l'implémentation du système et la réalisation des premiers tests.

Le présent mémoire se termine par une conclusion résumant les différentes connaissances acquises au travers du projet réalisé, ainsi que des propositions et perspectives pouvant permettre l'amélioration de notre système dans le futur.

Chapitre I

Généralités sur l'internet des objets

I.1 Introduction

Le *XXI^e* siècle fut caractérisé par d'importants progrès dans le domaine des télécommunications et des nouvelles technologies de l'information et de la communication. C'est ainsi que nous avons assisté à l'avènement d'un nouveau concept de l'internet . Ce nouveau concept de l'internet appelé internet des objets ou encore internet of things de par sa formulation anglaise est caractérisé par un nombre important d'objets connectés au réseau mondial internet.

Dans ce chapitre nous présenterons les généralités sur l'internet des objets, son architecture, les protocoles de communications utilisés ainsi que les domaines d'application de cette technologie.

I.2 Généralités sur l'internet des objets

I.2.1 Définition

L'internet of things est un phénomène assez complexe ne disposant cependant pas d'une réelle définition standard, chacun essaie de le définir à sa manière.

Certains surnomme ce phénomène par le concept de Web 3.0 considéré comme l'évolution direct de l'internet que nous connaissons (web 2.0 ou encore web social).

Néanmoins nous retiendrons de manière plus simple que l'IoT se défini comme étant une extension du monde virtuelle du réseau internet que nous connaissons aux objets du monde physique.[\[1\]](#).

I.2.2 Fonctionnement

Le principe de l'IdO est d'interconnecter les machines du réseau internet à des objets du monde physique (objets connectés) tel que des cafetières, des réfrigérateurs, des véhicules auxquels sont branché des multitudes de capteurs ayant pour rôles de collecter des données de l'environnement réel et de les envoyés à des serveurs (à travers le cloud) afin d'être traité. Une fois ces données traitées cela servirai à envoyer des commandes à des actuateurs (ou actionneurs) dans le but d'interagir avec le monde réel selon les données fournis par les capteurs.

Afin d'assurer la connectivité entre ces objets, ces derniers disposent d'un système d'adres-sage (RFID, code-barres, TCP/IP) afin d'être identifiable de manière unique sur le réseau et utilisant des protocoles de communications standard et interopérables dans le but de communiquer entre eux de manière efficace et autonome.

Nous verrons plus en détail le fonctionnement de l'IoT un peu plus loin dans ce chapitre [\[1\]](#).

I.2.3 Évolutions de l'écosystème de l'IoT

L'internet des objets est un néologisme utilisé pour la première fois lors d'une présentation en 1999 par Kevin Ashton, co-fondateur et directeur exécutif du Auto-ID Center qui fait référence à tous ces appareils (en dehors des ordinateurs et des smartphones) connectés à Internet. Qu'il s'agisse de voitures, de bracelets de fitness, de réfrigérateurs, de systèmes électriques ou d'autres choses encore, tous sont connectés à internet et nous permettent d'échanger des données grâce aux capteurs [\[2\]](#).

Pour CISCO l'internet des objets est le moment où le nombre d'objets connectés seraient supérieur au nombre de personnes connectés à internet (ce qui est actuellement le cas). Selon les rapports de CISCO la population mondiale s'élevait à 6.3 milliards d'habitants et 500 millions d'objets étaient connectés en 2003. Le nombre d'objets connectés aurait augmenté d'une manière exponentielle pour atteindre le seuil de 12,5 milliards contre 6,8 milliards d'habitant en 2010. Cette forte augmentation des objets connectés serait suite à une forte

commercialisation des smartphones et tablettes. Le nombre d'objets connectés est estimé à 50 milliards pour les années 2020 [3].

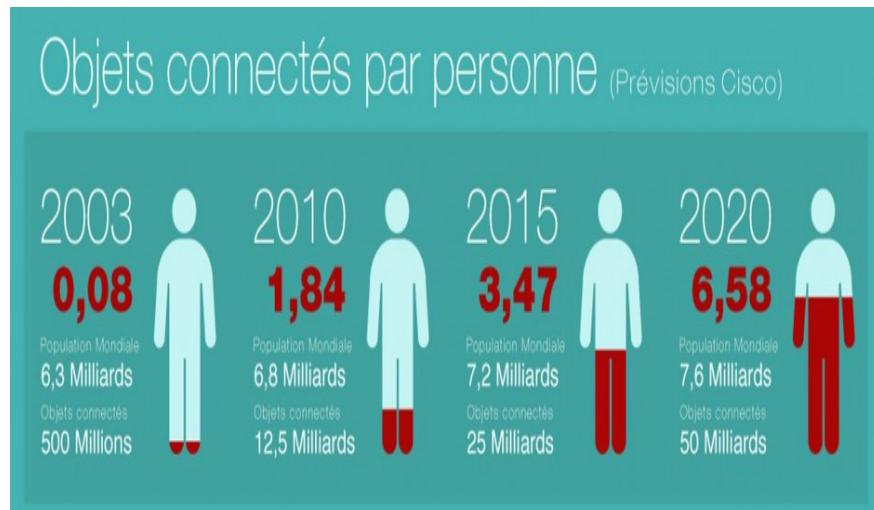


FIGURE I.1 – L'évolution du nombre d'objets connectés
[4]

I.3 Architecture et protocoles de communications

I.3.1 Architecture

L'IdO doit être en mesure d'interconnecter des milliards d'objets hétérogènes d'où la nécessité de disposer d'une bonne architecture adaptée aux besoins. L'architecture d'un système IoT est bâti autour de plusieurs niveaux qui communiquent entre eux pour assurer une liaison entre le monde physique et le monde virtuel d'internet. Il n'y a pas une architecture standard spécialement désigné pour les systèmes IoT mais il est cependant possible d'illustrer dans un schéma le parcours des données.

Un capteur mesure une grandeur physique analogique (qui peut être la température, l'hygrométrie, etc ...) et la convertie en une donnée numérique. Cette donnée est alors envoyée à une passerelle grâce à un premier protocole de communication. La passerelle a pour rôle de traduire les protocoles dans le but d'établir une communication entre les objets et le réseau (souvent le cloud). Parfois, une intelligence embarquée dans la passerelle permet le traitement et le stockage des données, ainsi que des fonctions de pilotage de l'objet. Pour les protocoles à courte portée (Zigbee, Z-Wave, Wi-Fi, BLE,), la passerelle est locale et se connecte souvent à la box du fournisseur d'accès à internet. Pour les protocoles à longue portée (Sigfox, LoRa, LTEM, NB-IoT, 3G/4G, ...), la passerelle se trouve sur le réseau de l'opérateur télécom.

Une plateforme IoT collecte les données, surveille, et contrôle les objets connectés. La plateforme logicielle IoT embarque une intelligence qui offre de multiples possibilités. Les mesures peuvent être stockées dans l'objectif de créer une historisation permettant la prédiction. La plateforme IoT communique avec différentes plateformes de services qui délivrent un service digital à l'utilisateur final restituant les données collectées des objets, permettant des actions de pilotage et apportant une couche d'intelligence (alertes, conseils, ...).

L'utilisateur final dispose d'une interface sur une tablette (ou tout autre terminal). Il peut alors piloter son objet, accéder à l'historique des mesures relevées, et utiliser les différentes fonctionnalités prévues par la solution IoT [5].

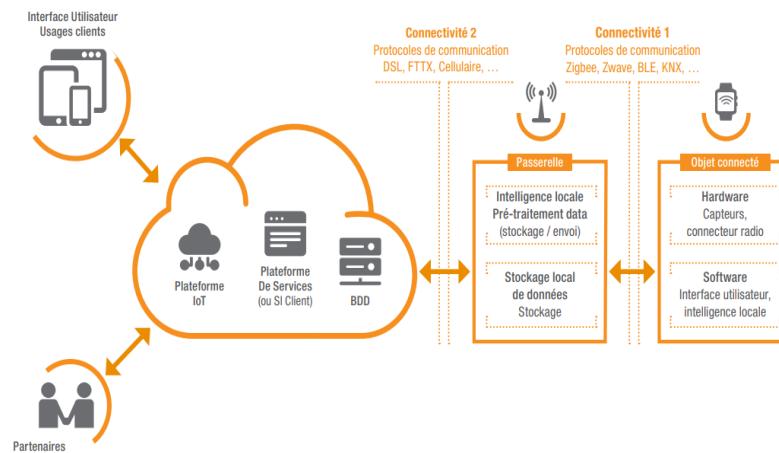


FIGURE I.2 – Architecture générale d'un système d'internet des objets
[5]

I.3.2 Technologies de communications

Pour mettre en place une bonne solution IoT il est nécessaire de choisir un réseau de communication le mieux adaptés aux systèmes que nous souhaitons mettre en place selon les besoins.

Le choix du réseau reste d'autant plus délicat qu'il existe plusieurs technologies différentes avec chacune une particularité bien définis en terme de portée, débit, consommation d'énergie, coût.

Nous distinguons deux grandes catégories de réseaux à savoir les réseaux longues portées et les réseaux courtes portées.

I.3.2.1 Les réseaux longues portée

Ils permettent de transférer des données d'un appareil à un autre sur une vaste étendue de surface. Ces réseaux sont utilisés dans le cadre d'un projet IoT de grande envergure tel que les smart city. Il existe plusieurs technologies réseau de ce type :

- **LoRa**

Les réseaux utilisant la technologie LoRa sont des LPWANs (bas débit, longue portée, faible consommation d'énergie). Ils permettent à des objets connectés de transférer des données vers internet pendant plusieurs mois sans nécessité d'être rechargés. LoRa sépare les communications dans des canaux fréquentiels de 125 kHz en utilisant une modulation à base de Chirps, assure une portée de 15km en milieu rural et 5km maximum en milieu urbain avec un débit variant de 0.3 à 50kbps. Les communications sont sécurisées par un chiffrement AES reposant sur une clé symétrique partagée.

- **Sigfox :**

Sigfox est un opérateur d'objets connectés créé en France en 2009. Les réseaux utilisant la technologie Sigfox utilisent des antennes basses fréquences dans la bande ISM 868MHz (en Europe) pour des services de communication peu énergivore, longue portée et bas débit. Basé sur des communications M2M et une modulation(DBPSK et GFSK), ce protocole permet d'envoyer environ 140 messages par jour avec un débit variant de 10 à 1000bit/s. Disponible dans les pays de l'Europe de l'Ouest et quelques villes des états unies, ce protocole est relativement bien adapté pour les systèmes M2M utilisant des objets tel que des capteurs d'environnement.

- **Les réseaux cellulaires (3G/4G)**

Ces technologies ont l'avantage d'avoir une très longue portée (jusqu'à 200km pour la 3G) et un débit relativement élevé (2Mbps pour la 3G et 10 Mbps pour la 4G). Cependant cette technologie implique des coûts et une consommation électrique excessive la rendant ainsi peu approprié pour certaines applications IoT.

I.3.2.2 Les réseaux courtes portée

Ces types de réseaux permettent d'interconnecter des appareils sur une courte distance. Ils sont très utilisés dans le domaine de la domotique. Nous distinguons également plusieurs technologies disponibles pour les réseaux courte portée :

- **Wifi** Le Wi-Fi est la technologie la plus utilisée dans la domotique et les réseaux locaux. Il dispose d'une faible portée (50 à 100 m) et d'un débit habituellement proche de 150Mbps. A l'heure actuelle la norme Wifi la plus utilisée est la 802.11n.
- **Z-Wave** C'est une technologie de communication pour des échanges à faible bande passante conçu pour le domaine de la domotique avec une faible latence et un débit de 100kbps. Conçu pour les communications dans le domaine de la domotique, il offre une totale résistance aux interférences causées par le Wi-Fi et les autres technologies sans fil.
- **ZigBee** Le ZigBee est une technologie de communication des réseaux de capteurs sans fil (Wireless Sensor Network) à courte distance (environ 100m) et à faible consommation d'énergie destiné pour des transmissions à faible débit (250kbps).
- **Bluetooth** Le Bluetooth est une norme de communication sans fil efficace à courte distance (environ 60m) et peu gourmande en énergie, compatible avec la majorité des appareils, utilisé pour connecter des appareils médicaux, application mobiles, oreillettes et montres intelligentes. Récemment une nouvelle version Bluetooth 4.0 (BLE) plus adapté pour l'IoT a été mis sur le marché.

I.3.3 Les protocoles applicatifs

I.3.3.1 CoAp (Constrained Application Protocol)

Le CoAp est un protocole de la couche application des applications IoT conçu par l'IETF (Internet Engineering Task Force) basé sur une architecture Request/Response. Ce protocole s'inspire du protocole HTTP dont il reprend en partie les concepts et la terminologie (méthodes GET, PUT, POST et DELETE) en allégeant considérablement les échanges, notamment en s'appuyant sur le protocole UDP plutôt que sur le protocole TCP.

Cette similitude avec le protocole HTTP permet une communication aisée entre des appareils utilisant les deux protocoles car le CoAp peut interagir facilement avec le HTTP via un proxy (passerelle).

Les paquets CoAp utilisent un format plus simple et petit que les paquets HTTP ; l'en-tête d'un message habituellement utilisé pour le HTTP est remplacé par un en-tête fixe de 4 octets plus adapté aux faibles ressources des équipements de l'IoT. La taille maximale d'un message CoAp est de 1024 octets.

Dans l'en-tête d'un paquet, quelques bits indiquent la version de CoAp utilisée, le type de message et la qualité de service souhaitée pour la transmission du message (CoAp utilise quatre types de message).

- Confirmable : Message envoyé avec une demande d'acquittement.
- Non-Confirmable : Message envoyé sans demande d'acquittement.
- Acknowledgment : Confirmation de la réception d'un message de type « confirmable ».
- Reset : Confirmation de la réception d'un message qui n'est pas exploitable[6].

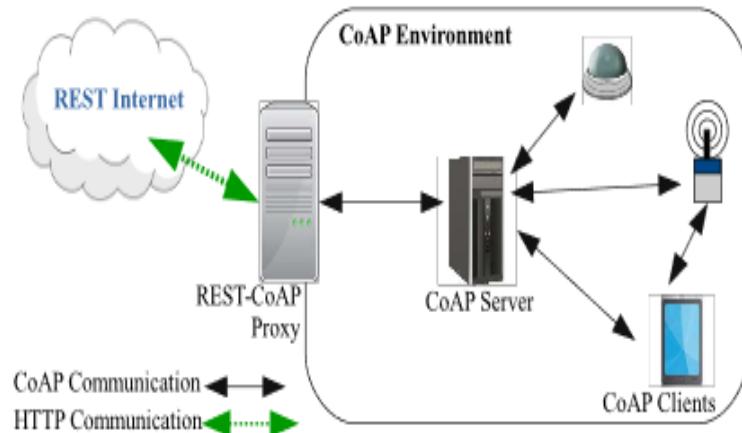


FIGURE I.3 – Fonctionnalité du protocole CoAP.

I.3.3.2 MQTT (Message Queue Telemetry Transport))

MQTT développé à l'origine par des chercheurs d'IBM en 1999 et normalisé à OASIS en 2003 est un protocole de messagerie simple et léger basé sur une architecture Publication/Souscription (Publish/Subscribe) spécifiquement conçu pour des applications M2M utilisé aussi dans les WSNs et des applications comme Messenger.

MQTT est construit au-dessus du protocole TCP et permet une notification en temps réel de la publication d'une nouvelle donnée sans interroger le serveur : cela apporte une forte réactivité aux applications [7].

a. Fonctionnement du protocole MQTT Contrairement au principe client/serveur utilisé sur le web, le MQTT se base sur un principe de Publication/Souscription. Dans cette architecture plusieurs clients se connectent à un serveur (appelé broker) pour publier des informations ou souscrire à leur réception. Pour effectuer une souscription les clients s'enregistrent auprès du broker sur une sorte de chemin d'accès appelé topics. Par exemple un client peut souscrire à un capteur de température d'une maison (Figure 1.3.)

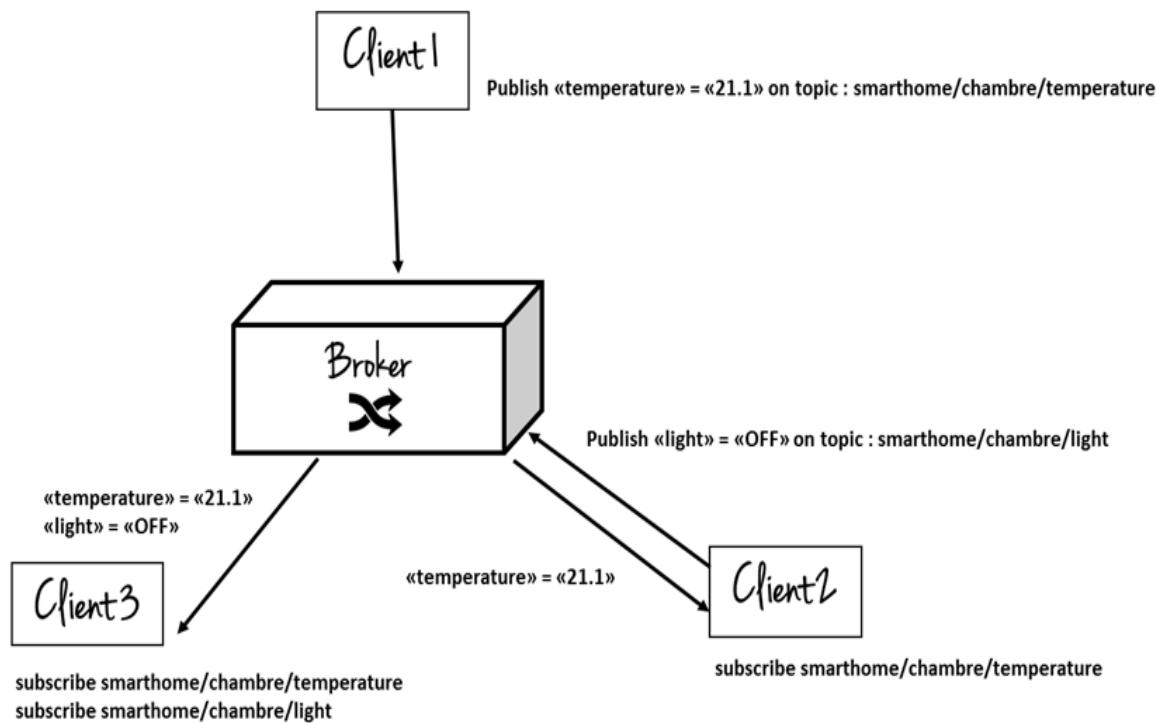


FIGURE I.4 – Publication/Souscription au niveau d'un serveur MQTT

[7]

b. Avantages et inconvénients

Le MQTT dispose de nombreux avantages dont entre autre sa légèreté, sa flexibilité, la sécurité offerte à travers l'authentification par certificats client et serveur, le chiffrement SSL/TLS.

Le MQTT offre également une bonne qualité de service (QoS) dont trois niveaux sont prévus pour la transmission des messages :

- – Fire and forget : le message est envoyé une fois et aucun acquittement n'est exigé.
- – Delivered at least once : le message est envoyé au moins une fois et un acquittement est exigé.
- – Delivered exactly once : un mécanisme en quatre temps assure la délivrance du message une seule et unique fois.

L'inconvénient du protocole MQTT est que le broker doit être un matériel suffisamment puissant pouvant supporter l'utilisation des protocoles de sécurité. Encore plus léger, une version optimisée du protocole MQTT, adaptée aux objets à très faibles ressources communiquant sans fil, est spécifiée sous le nom MQTT-SN (MQTT for Sensor Network) via UDP [7]

I.3.3.3 AMQP

AMQP est un protocole M2M léger, développé par John O'Hara en 2003. Il s'agit d'un protocole de messagerie d'entreprise conçu pour la fiabilité, la sécurité, le provisionnement et l'interopérabilité et récemment est utilisé dans l'iot.

AMQP prend en charge à la fois les requêtes / réponses et les interfaces de publication / souscription. Il offre un large éventail de fonctionnalités liées à la messagerie, telles que la mise en file d'attente fiable, la messagerie publication / souscription par sujet, le routage flexible et les transactions. AMQP utilise TCP comme protocole de transport par défaut et les stratégies de sécurité TLS / SSL et SASL. Ainsi, la communication entre client et broker

est orientée connexion. La fiabilité est l'une des fonctionnalités essentielles d'AMQP et offre deux niveaux préliminaires de qualité de service (QoS) pour la transmission des messages :[8]

I.4 Domaines d'application

On peut distinguer différents domaines d'applications de l'internet des objets :

I.4.1 Santé et médecine

Dans ce domaine l'IoT permet de suivre l'état d'un patient à distance (à travers l'utilisation de biocapteurs) en temps réel afin intervenir en cas d'urgence. Dans ce cas on peut utiliser des capteurs de pression artérielle, de glycémie, etc...

I.4.2 Transport et logistique

Dans ce secteur l'IoT facilite la récolte des données du trafic routier afin de résoudre les problèmes de congestion de la circulation et de stationnement ainsi améliorer la sécurité routière, la traçabilité et le suivi des véhicules de transport de marchandise. On utilise beaucoup le GPS et les accéléromètres dans le domaine des transports.

I.4.3 Industrie

La mise en place d'un système IoT dans une industrie permet d'améliorer le confort, la sécurité, la communication entre les différents appareils, la gestion de l'énergie, la surveillance de la chaîne de production et le déclenchement d'une alarme ou l'envoie d'alertes suite à une anomalie.

I.4.4 Sport

Dans le domaine du sport on peut utiliser des capteurs pour surveiller les performances d'un athlète (distance parcourue). Son emploi permet donc de mieux connaître les forces et les faiblesses d'une équipe ou d'un joueur en particulier lors d'un entraînement et aide aussi à améliorer les performances de l'athlète. On utilise les capteurs les plus couramment utilisés dont le cardiofréquencemètre et le podomètre.

I.4.5 Domotique

L'IoT intervient aussi beaucoup dans le domaine de la domotique. De nombreuses possibilités sont offertes par cette technologie, dans les domaines de la surveillance et la sécurité notamment en déclenchant une alarme en cas d'incident (feu, fuites, intrusion...), du confort (optimisation de la lumière, climatisations et autres appareils), ou encore de la gestion de la consommation énergétique.

I.4.6 Energie

L'ToT propose des possibilités de gestion en temps réel pour une distribution et une gestion efficaces de l'énergie de certaines machines conçus pour fonctionner dans des milieux difficiles, contrôle à distance plusieurs types d'installations énergétique : électriques, distribution d'eau et les réseaux électriques intelligents (smart grid).

I.4.7 Agriculture

Les nouveaux apports de l'IoT permettent d'améliorer considérablement le rendement des cultures à travers l'agriculture de précision. En effet l'IoT permet une supervision en temps réel de l'environnement et des cultures (température, hydrométrie, ...). On peut ainsi

gérer précisément l’arrosage, et la prévention des risques météo.

I.5 Constraintes liées à l’usage de l’IoT

I.5.1 Scalabilité

Dans un mode en perpétuel évolution technologique l’IoT se doit de faire face à des pics ponctuels de connexion et à la croissance continue d’utilisation du service et des augmentations des données lié à l’augmentation fulgurante des objets connectées[9].

I.5.2 Hétérogénéité

L’hétérogénéité des objets qui peuvent être connectés dans de tels environnements, rend difficile leur interopérabilité. En outre, les observations produites par ces objets sont générées avec différents vocabulaires et formats de données. Cette hétérogénéité de technologies dans le monde de l’IoT rend nécessaire l’adoption de solutions génériques à l’échelle mondiale. De plus, elle rend difficile le partage et la réutilisation des données

L’IoT se doit ainsi de surmonter les défis d’interopérabilité sémantiques introduits par la variété des capteurs potentiellement utilisés. [9].

I.5.3 Influence du monde physique

Nous avons pu voir qu’il y a déjà plus d’objets connectés que d’habitants sur la planète et cette tendance n’est pas près de s’arrêter, bien au contraire ! L’explosion du nombre d’objets connectés représente donc un enjeu énergétique important. Cependant l’on ne peut disposer de source énergétique dans n’importe quel emplacement physique d'où la nécessité de tenir compte de la topologie du monde physique et également des aléas climatiques pour ne pas trop exposé les objets connectés[9].

I.5.4 Sécurité et vie privée

Le problème majeur lié à l’usage de l’IoT est son impact sur la vie privée des personnes. En effet la forte collectivité avec l’internet pourrait entraîner des vols d’identité et d’informations personnelles. Une stratégie de sécurité doit donc être intégrée entièrement dans l’architecture pour éviter les failles matérielles et logicielles à tous les niveaux du système. Notons également que l’usage des objets connectés concourt à la saturation des adresses IPV4 d'où la nécessité d'un nouveau protocole d'adressage[9].

I.6 Conclusion

Tout au long de ce chapitre, nous avons choisi de définir les différents concepts liés à l’internet des objets et son impact sur la vie social et économique notamment à travers la surveillance et le monitoring.

Dans le chapitre qui suit nous entamons la partie conception de notre système où l’on va identifier les différents besoins et problèmes recensés auxquels notre système doit répondre ainsi que ses fonctionnalités, les acteurs et son architecture.

Chapitre II

Conception

II.1 Introduction

L'analyse et la conception sont les étapes incontournables qui nous permettront de bien mener notre projet. Ces étapes nous permettront de mieux cerner les contours de notre projet et de le conformiser avec les règles standards de conception.

Dans un premier temps nous verrons dans ce chapitre une présentation globale du projet(problématique et solution proposée), l'architecture du système, la partie matérielle et les modes de communications et en fin la partie logicielle dans laquelle nous développerons les aspects fonctionnels du système et établirons les diagrammes des cas d'utilisations, de séquences suivie du modèle relationnelle associé à notre système.

II.2 Problématique et solution

II.2.1 Problématique

Notre problématique peut être abordé sous différents angles dans divers environnements de surveillances. Prenons par exemple, le domaine de la santé où il y a une nécessité de surveiller les patients et les personnes âgées d'une façon permanente et en temps réel sans se rendre chez eux dans leurs maisons. Cela permettrait ainsi de réduire les frais d'hospitalisation et de surveillance qui ne cesse d'augmenter, surtout pour les personnes âgées ; améliorer le diagnostics et le traitements des maladies, maintenir les gens en vie et faciliter l'intervention en un minimum de temps ; pouvoir aussi analyser les flux donnés et les analyser avec profondeur.

Dans le domaine industriel, on peut réduire des dépenses et des frais de production ; un autre objectif peut être le gain de temps précieux, notamment celui d'assurer la maintenance prédictive des engins dont l'état de forme est constamment sous surveillance.

D'autre part comment assurer la sécurité et la surveillance permanente avec moindre coût. La surveillance des infrastructures, les personnes et leurs biens sans aucune intervention humaine, ainsi que d'autres préoccupation ; la protection de l'environnement et la prévention contre les catastrophes naturelles, agir et avertir les gens. Un autre souci est de trouver une solution pour réduire la consommation d'énergie et optimiser son utilisation dans les différents domaines. On rêve d'un système domotique qui pourrait surveiller l'état des appareils électriques de la maison pour anticiper les pannes de même qui nous avertirait de l'imminence du problème, on doit s'assurer aussi une bonne analyse des données collectées dans ses environnement et de les visualiser en temps réel.

Toutes ces préoccupations font que le nombre d'application IoT ne cesse de croître de jour en jour et cela de paire avec la demande sur le marché. Il devient alors nécessaire de réduire le coût de conception des applications .

II.2.2 Solution

Notre solution proposée consiste à assurer la coopération entre la supervision, la gestion et contrôle des plusieurs sites de surveillances distants via un système générique adaptables aux différents domaines de surveillances. Ce système sera accessible et contrôlable en temps réel via une interface web et extensible par des actionneurs et des capteurs composées de deux (2) parties :

Une partie matérielle consistant en un ensemble d'équipements physiques (microcontrôleurs, capteurs) qui vont servir à capter et mesurer les différentes métriques de l'environnement d'implémentation et contrôler les actionneurs qui à leurs tours vont recevoir des commandes à partir de l'interface web qui est l'intermédiaire entre l'utilisateur et la partie matériel de notre système . La partie logicielle consiste une application web de gestion pour le stockage, l'analyse et la visualisation des données collectées ainsi que le contrôle du système

à distance et cela en quasi temps réel.

Cela permettrait ainsi le suivi de toute opération qui se déroule dans les environnements d'implémentation. La communication entre les objets du système sera assurée par un protocole de communication ,le MQTT en l'occurrence. Ce dernier offre une sécurité des données aux utilisateurs et une bonne qualité de service

II.3 Architecture de la solution proposée

Notre solution est composée de deux parties comme le montre la figure ci-dessous.

Nous avons une partie matérielle (hardware) qui est essentiellement constituée de capteurs et de microcontrôleurs qui s'échangent des informations via le protocole MQTT. Les données récoltées sont alors transmises via internet à un serveur qui se chargera des traitements adéquats.

C'est la partie software qui se charge de traiter les données et permet ainsi leurs visualisations sur une interface web. C'est cette même interface qui permettra aux utilisateurs d'envoyer des commandes au site distant.

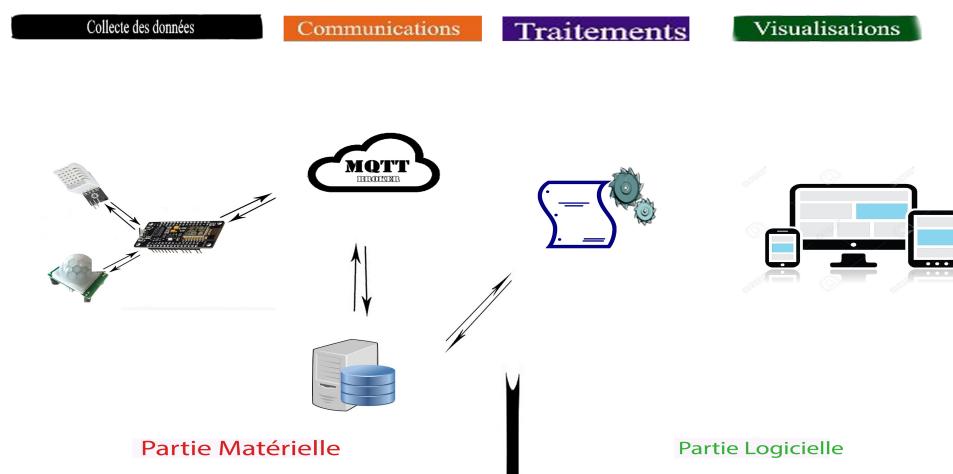


FIGURE II.1 – Architecture du système

II.4 Partie matérielle et communication

Dans cette section nous décrirons les principales composantes de la partie hardware de notre système ainsi que des différentes interactions entre elles. Comme décrit précédemment cette partie est constitué de microcontrôleurs auxquelles sont reliées des capteurs et des actionneurs. Pour assurer la communication entre les différentes cartes matérielles utilisés, le protocole MQTT s'avère être le mieux adaptés car très souple à mettre en œuvre.

II.4.1 Connexion au serveur MQTT

Les différentes cartes sont reliées à un serveur MQTT(broker) à travers une connexion WiFi ; un programme lance la procédure de connexion au serveur MQTT, si cette connexion échoue, une reconnexion est engagée, on patiente ensuite 5 secondes pour engager à nouveau la procédure de connexion comme le montre la figure ci-dessous.

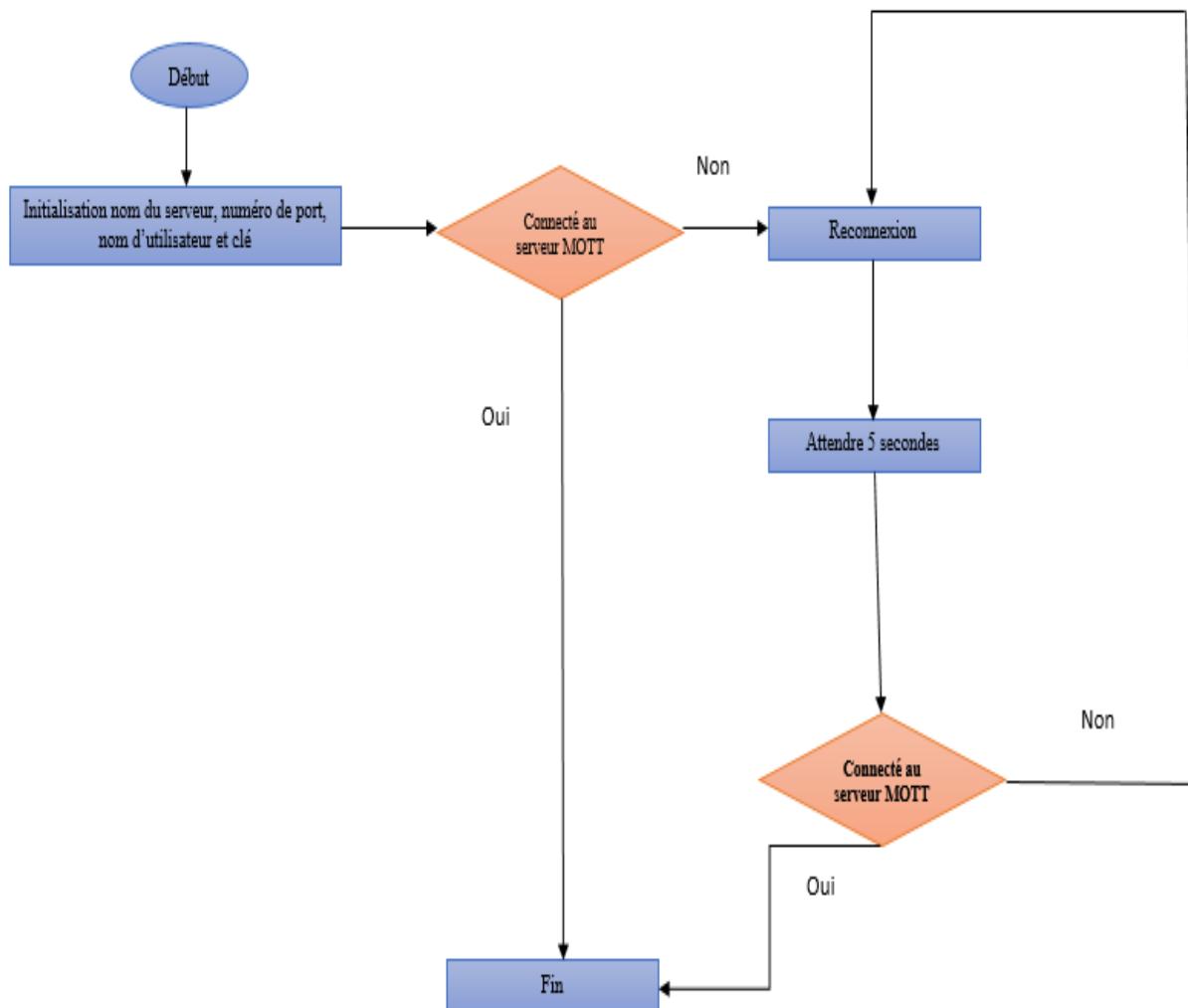


FIGURE II.2 – Connexion au serveur MQTT

II.4.2 Communication MQTT

Une fois la connexion établie, les échanges peuvent alors s'effectuer via le modèle Publish/Suscrib.

Le mécanisme de fonctionnement est assez simple, les cartes souscrivent aux topics au niveau du broker afin de recevoir les flux d'informations qui sont publiés par les autres cartes sur ces mêmes topics.

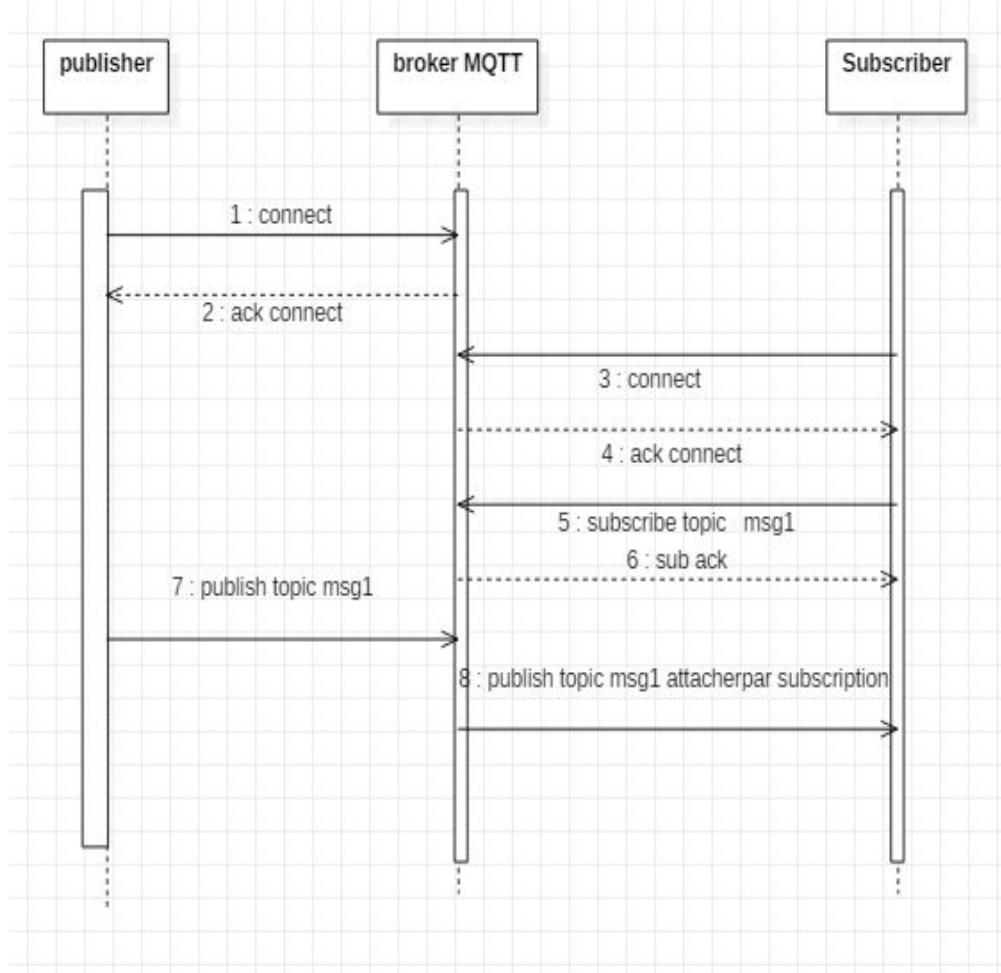


FIGURE II.3 – Mecanisme de publication/souscription MQTT

II.5 Partie logicielle

Cette section sera comme dit en amont consacrée à l'étude fonctionnelle du système où nous déterminerons les différents acteurs ainsi que les besoins fonctionnelles et non fonctionnelles du système. Une partie de la section sera donc consacrée à la modélisation du système (élaboration des diagrammes des cas d'utilisation, des diagrammes de séquence et le diagramme de classe du système ainsi que le schémas de la base de donnée)

II.5.1 Fonctionnalités

Le système devra permettre :

Un accès à distance : l'utilisateur du système une fois identifié, pourra avoir accès au système depuis internet.

D'ajouter des cartes, des capteurs et des actionneurs : l'administrateur principal ou un utilisateur délégué peut étendre les fonctionnalités du système de surveillance avec de nouveaux capteurs et actionneurs.

D'actionner des actuateurs : si un objet électrique est relié à une prise secteur et que cette dernière est équipée d'un relais qui lui-même est relié au microcontrôleur ; il devient alors tout à fait possible d'allumer ou d'éteindre cet actuateur.

Visualisation des métriques et des états de capteurs et actionneurs : il sera possible de consulter les valeurs transmises par les différents capteurs (humidité, température...) et les états de chaque actionneur via l'interface web.

Voir l'historique et les statistiques : Il doit être alors possible de remonter dans l'historique des valeurs et des états captés de chaque entité(capteurs/actionneurs) à temps régulier.

Attribuer des rôle spéciaux aux utilisateurs : cette fonction est réservée au super utilisateur.

II.5.2 Diagramme de cas d'utilisation

II.5.2.1 Identification des acteurs du système

Nous regrouperons les acteurs qui interviendront dans notre système dans le tableau suivant ainsi que les rôles qu'ils y joueront.

		Acteurs			
Rôles		Chef d'entreprise	Chef de projet	Technicien	Utilisateur
Gestion d'utilisateur	ajouter	✗			
	supprimer	✗			
	modifier	✗			
Gestion de projet	ajouter	✗			
	modifier	✗	✗		
	supprimer	✗			
	Assigner à un chef projet	✗			
	Assigner à un technicien	✗	✗		
	Assigner à un utilisateur	✗			
Gestion de site	ajouter		✗	✗	
	modifier		✗	✗	
	supprimer		✗	✗	
Gestion de carte	ajouter		✗	✗	
	modifier		✗	✗	
	supprimer		✗	✗	
Gestion des capteurs	ajouter		✗	✗	
	modifier		✗	✗	
	supprimer		✗	✗	
Gestion des actionneurs	ajouter		✗	✗	
	modifier		✗	✗	
	supprimer		✗	✗	
Visualisation des données et statistiques		✗	✗	✗	✗
Actionner les actionneurs		✗	✗	✗	✗
S'authentifier		✗	✗	✗	✗

TABLE II.1 – Description des fonctionnalités des acteurs de système

II.5.2.2 Description des cas d'utilisation

Description du cas d'utilisation « Authentification »

Cas d'utilisation N° 01	Authentification
Résumé	Donner l'accès pour exploiter le système.
Acteurs	Tous les utilisateurs
Scénario N°01	<p>[Début]</p> <ol style="list-style-type: none"> 1. L'utilisateur lance l'application ; 2. L'application renvoie le formulaire adéquat ; 3. L'utilisateur introduit les informations qui l'identifient ; 4. L'utilisateur soumet sa requête au système ; 5. Le système vérifie la conformité des informations ; 6. Le système donne accès à l'application web ; 7. L'application web affiche l'interface d'accueil adéquate ; <p>[Fin]</p>
Exception	<p>L'utilisateur remplit mal le formulaire (champ non rempli).</p> <p>Utilisateur non existant.</p>

TABLE II.2 – Description du cas d'utilisation « Authentification »

Description du cas d'utilisation gestion de projets

Cas d'utilisation N° 02	Gestion des projets.
Résumé	Ajouter, modifier ou supprimer un projet et assignation de droit.
Acteurs	Chef d'entreprise.
Scénario N°02	<p>[Début]</p> <ol style="list-style-type: none"> 1. Le chef d'entreprise passe à la section gestion des projets ; 2. Chef d'entreprise. clique sur l'une des icônes ajouter, modifier, ou supprimer un projet ; 3. L'application affiche le formulaire adéquat ; 4. Le chef d'entreprise effectue l'opération adéquate ; 5. Le système renvoie un message ; <p>[Fin]</p>
Exception	<p>Le chef d'entreprise remplit mal le formulaire (ajout, suppression, modification).Projet non existant</p>

TABLE II.3 – Description du cas d'utilisation gestion des projets

Description du cas d'utilisation gestion des utilisateurs

Cas d'utilisation N° 03	Gestion des utilisateurs.
Résumé	Ajouter, modifier ou supprimer un utilisateur.
Acteurs	Chef d'entreprise
Scénario N°03	<p>[Début]</p> <ol style="list-style-type: none"> 1. Le chef d'entreprise passe à la section gestion des utilisateurs ; 2. Le chef d'entreprise clique soit sur l'icône ajouter, modifier, ou supprimer un compte utilisateur ; 3. L'application affiche le formulaire correspondant ; 4. Le chef d'entreprise remplit le formulaire avec les informations adéquates ; 5. Le système renvoie un message de confirmation relativement à l'opération effectuée ; <p>[fin] .</p>
Exception	Le chef d'entreprise remplit mal le formulaire (des espaces dans l'identifiant de l'utilisateur, champ non rempli, caractères spéciaux).

TABLE II.4 – Description du cas d'utilisation gestion des utilisateurs

Description du cas d'utilisation mise à ON ou à OFF des actionneurs

Cas d'utilisation N° 04	Commander les actionneurs
Résumé	Envoyer des commandes aux actionneurs
Acteurs	Tous les Utilisateurs
Scénario N° 07	<p>[Début]</p> <ol style="list-style-type: none"> 1. L'utilisateur passe à la section visualisation des données et statistiques ; 2. L'utilisateur choisit le projet ; 3. L'utilisateur passe à la section des actionneurs ; 4. L'utilisateur commande l'actionneur désiré à ON ou OFF 5. Le système traite la requête ; <p>[Fin]</p>
Exception	Actionneur non connecté ou non existant.

TABLE II.5 – Description du cas d'utilisation « mise à ON ou à OFF des actionneurs

II.5.2.3 Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation représentent les fonctions du système de façon générale pour tous les types d'utilisateurs qui vont l'utiliser.

Diagramme de cas d'utilisation pour le chef d'entreprise

Le chef d'entreprise a un contrôle total sur le système, il assure la gestion des utilisateurs (ajout, suppression, modification), la gestion des projets (création, modification, suppression, assignation de chef de projet et technicien...).

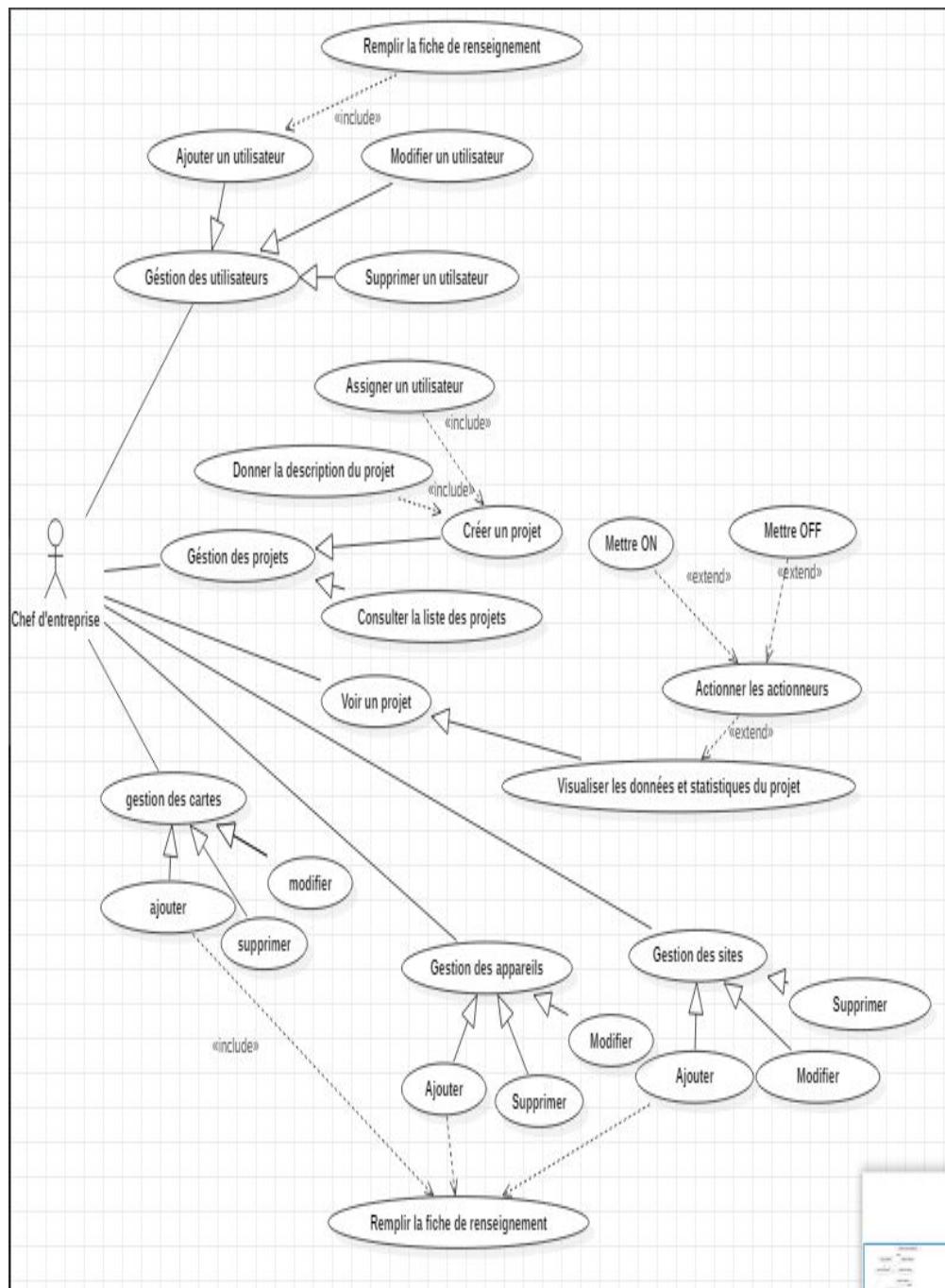


FIGURE II.4 – Diagramme de cas d'utilisation d'un chef d'entreprise

Diagramme de cas d'utilisation d'un chef de projet

Le chef de projet peut gérer les projets auxquels il est assigné, assigner des techniciens qui vont s'occuper de la gestion des sites, cartes et équipements, plus les fonctionnalités standards (ajout, modification, suppression, consultation des données).

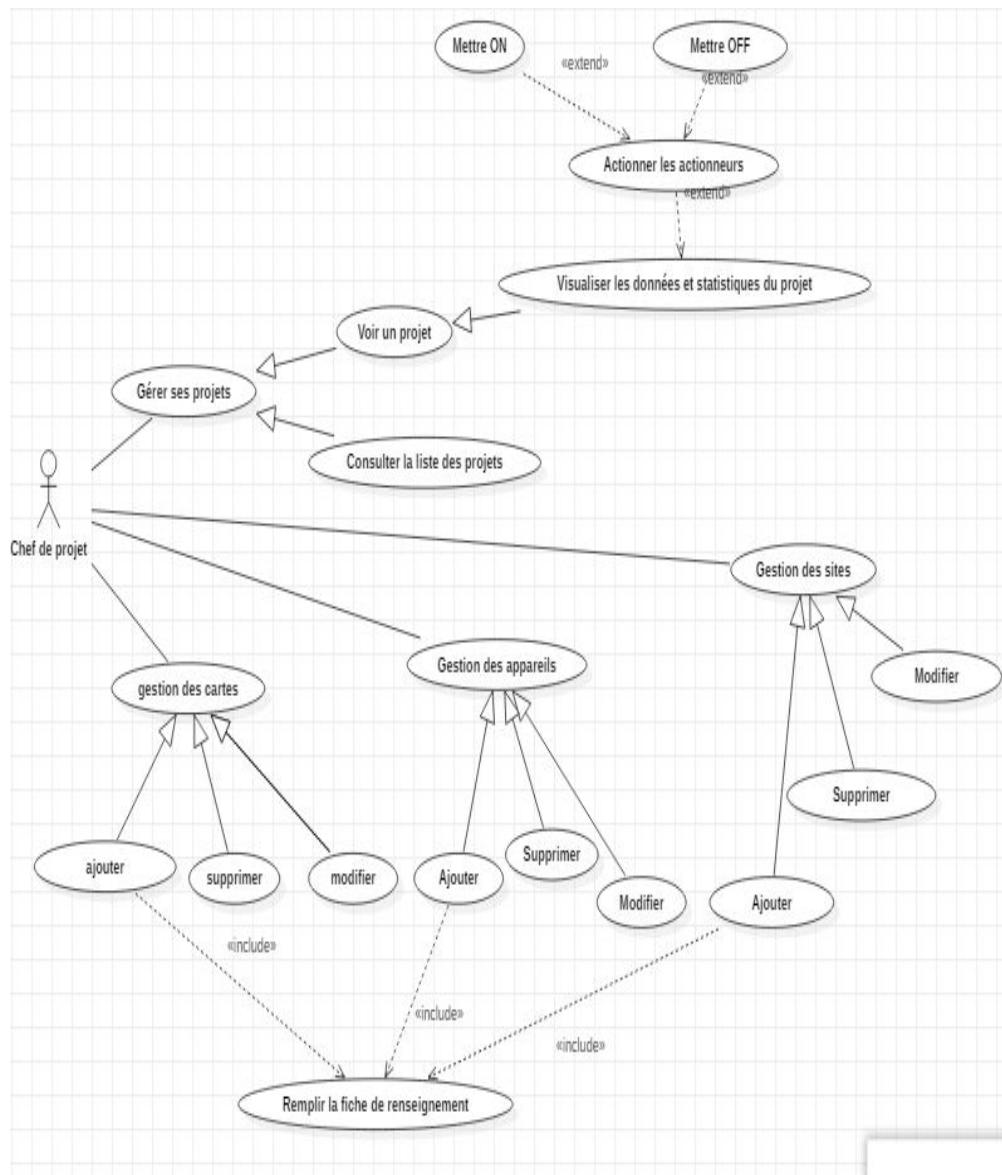


FIGURE II.5 – Diagramme de cas d'utilisation d'un chef de projet

Diagramme de cas d'utilisation d'un technicien

Le technicien possède l'option de consulter le projet qui lui est assigné , il pourra ajouter des sites ainsi que des cartes, des appareils (capteurs et actionneurs) ou bien les enlever. Il aura aussi la possibilité de modifier les équipements et ses sites

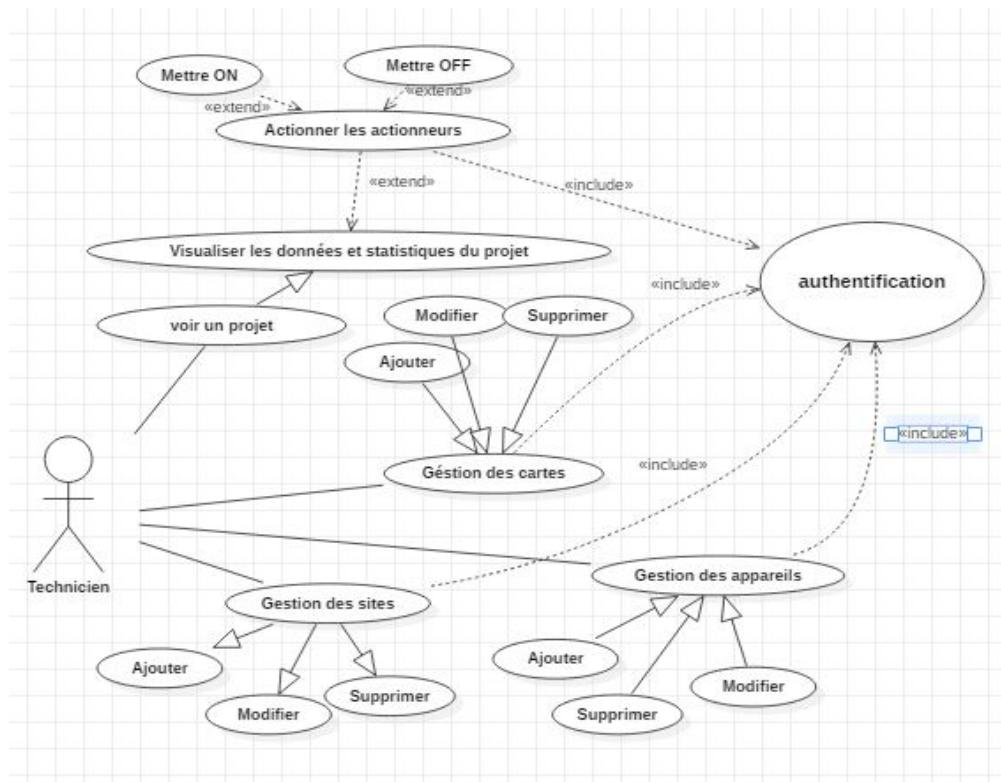


FIGURE II.6 – Diagramme de cas d'utilisation d'un technicien

Diagramme de cas d'utilisation pour un utilisateur ordinaire

L'utilisateur pourra consulter les informations liées à ses projets, voir les données collectées par les différents capteurs contenus dans son projet, il peut interagir avec le système pour déclencher des actions avec les différents actionneurs.

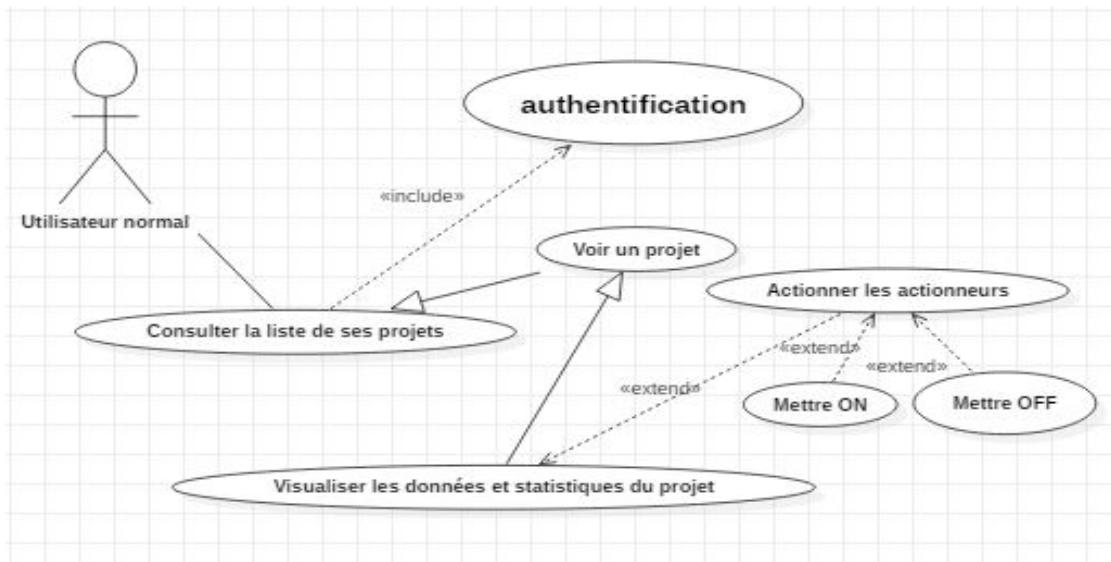


FIGURE II.7 – Diagramme de cas d'utilisation d'un utilisateur ordinaire

II.5.3 Diagramme de séquences

Les diagrammes de séquences permettent de décrire comment les éléments du système interagissent entre eux et avec les acteurs. Les objets au cœur d'un système interagissent

en s'échangent des messages. Les acteurs interagissent avec le système au moyen Interfaces Homme-Machine

II.5.3.1 Diagramme de séquence d'authentification

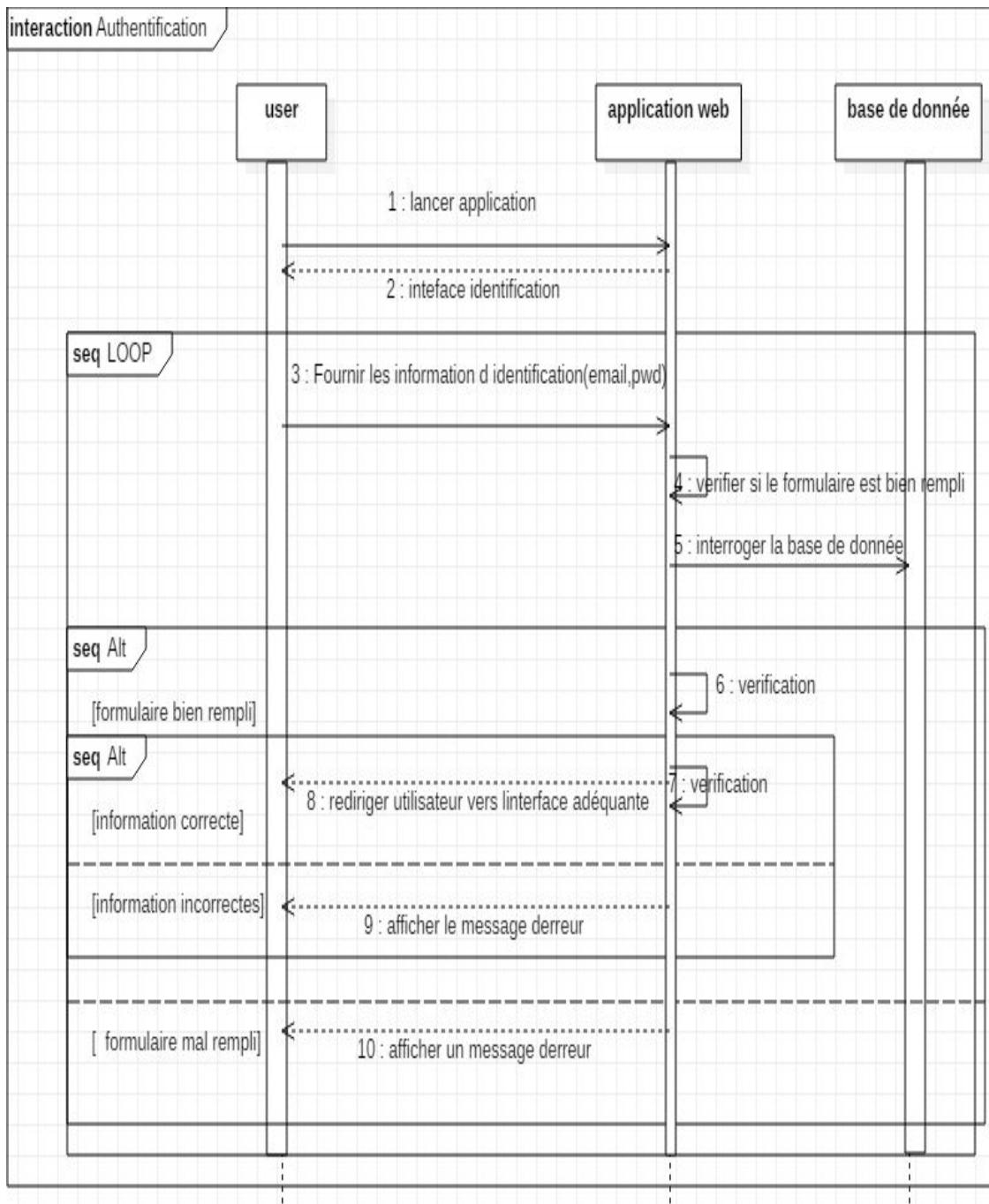


FIGURE II.8 – Diagramme de séquence d'authentification

II.5.3.2 Diagramme de séquence d'ajout d'un utilisateur

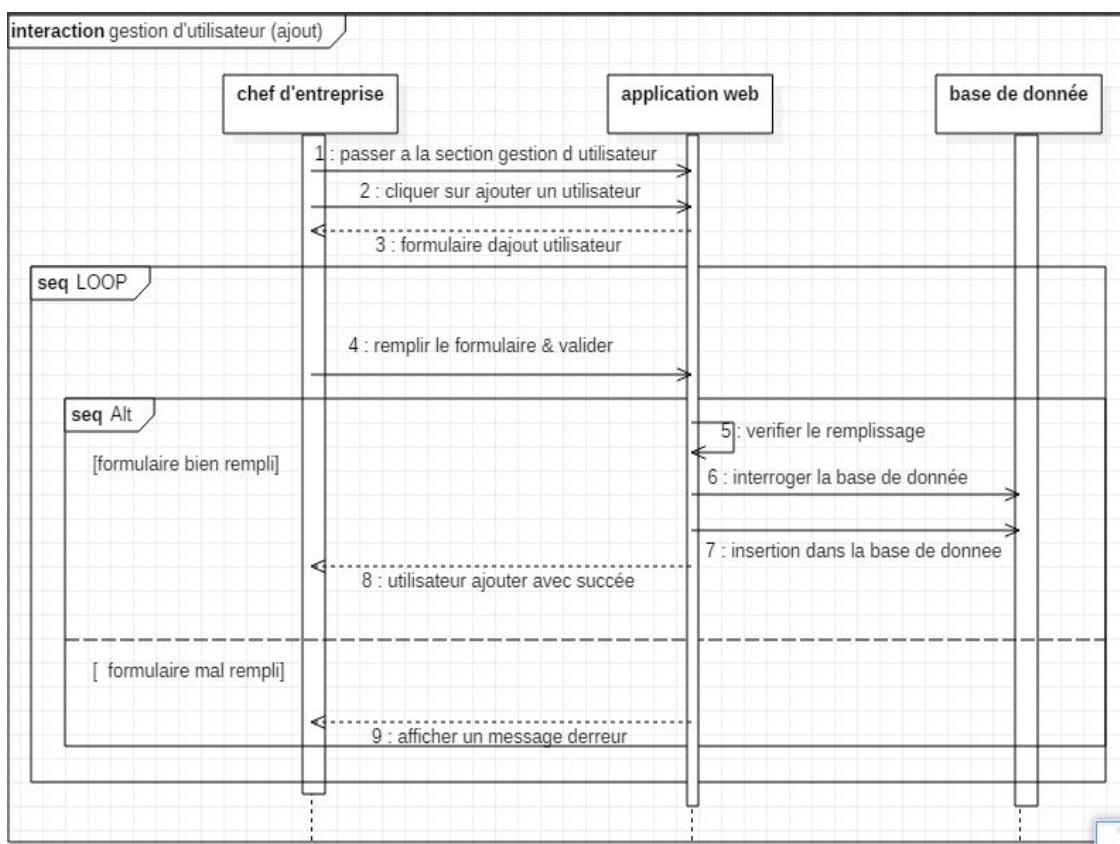


FIGURE II.9 – Diagramme de séquence d'ajout d'un utilisateur

II.5.3.3 Diagramme de séquence d'ajout d'un projet

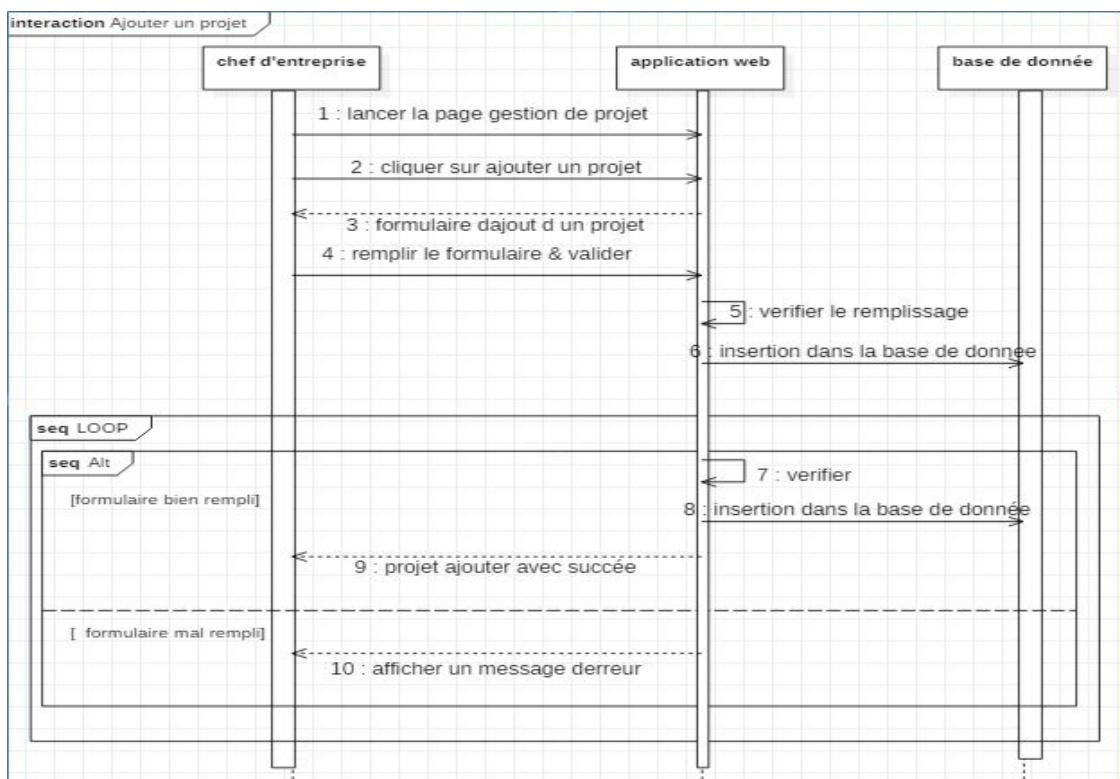


FIGURE II.10 – Diagramme de séquence d'ajout d'un projet

II.5.3.4 Diagramme de séquence de la visualisation des données

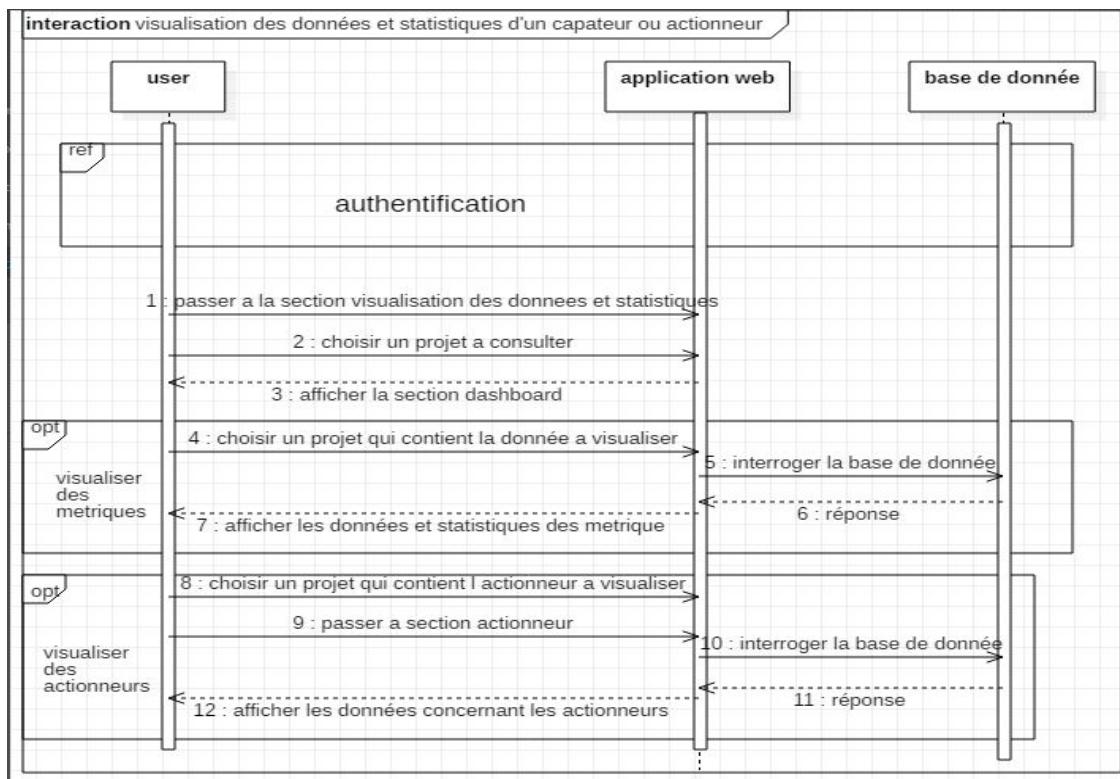


FIGURE II.11 – Diagramme de séquence de la visualisation des données

II.5.3.5 Diagramme de séquence de commande d'un actionneur

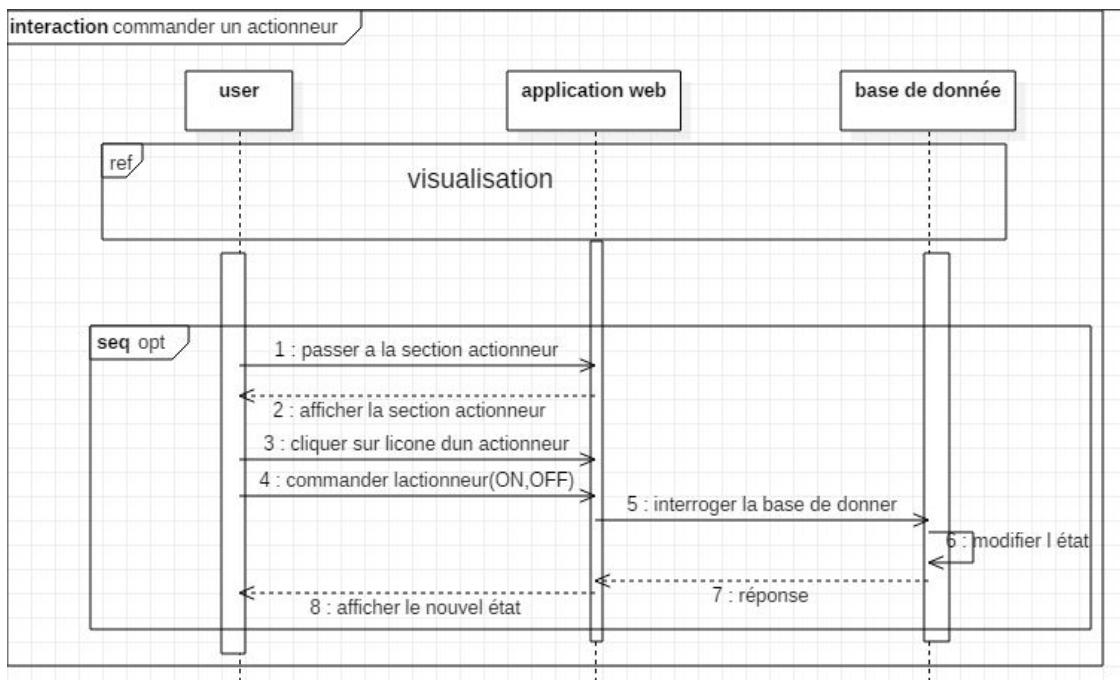


FIGURE II.12 – Diagramme de séquence de commande d'actionneur

II.5.4 Diagramme de classes

Le diagramme de classes représente les classes constituant le système et les associations entre elles. Les diagrammes de classes expriment de manière générale la structure statique d'un système, en termes de classe et de relations entre ces classes. De même qu'une classe décrit un ensemble d'objets, une association décrit un ensemble de liens ; les objets sont des instances de classes et les liens sont des instances de relations.

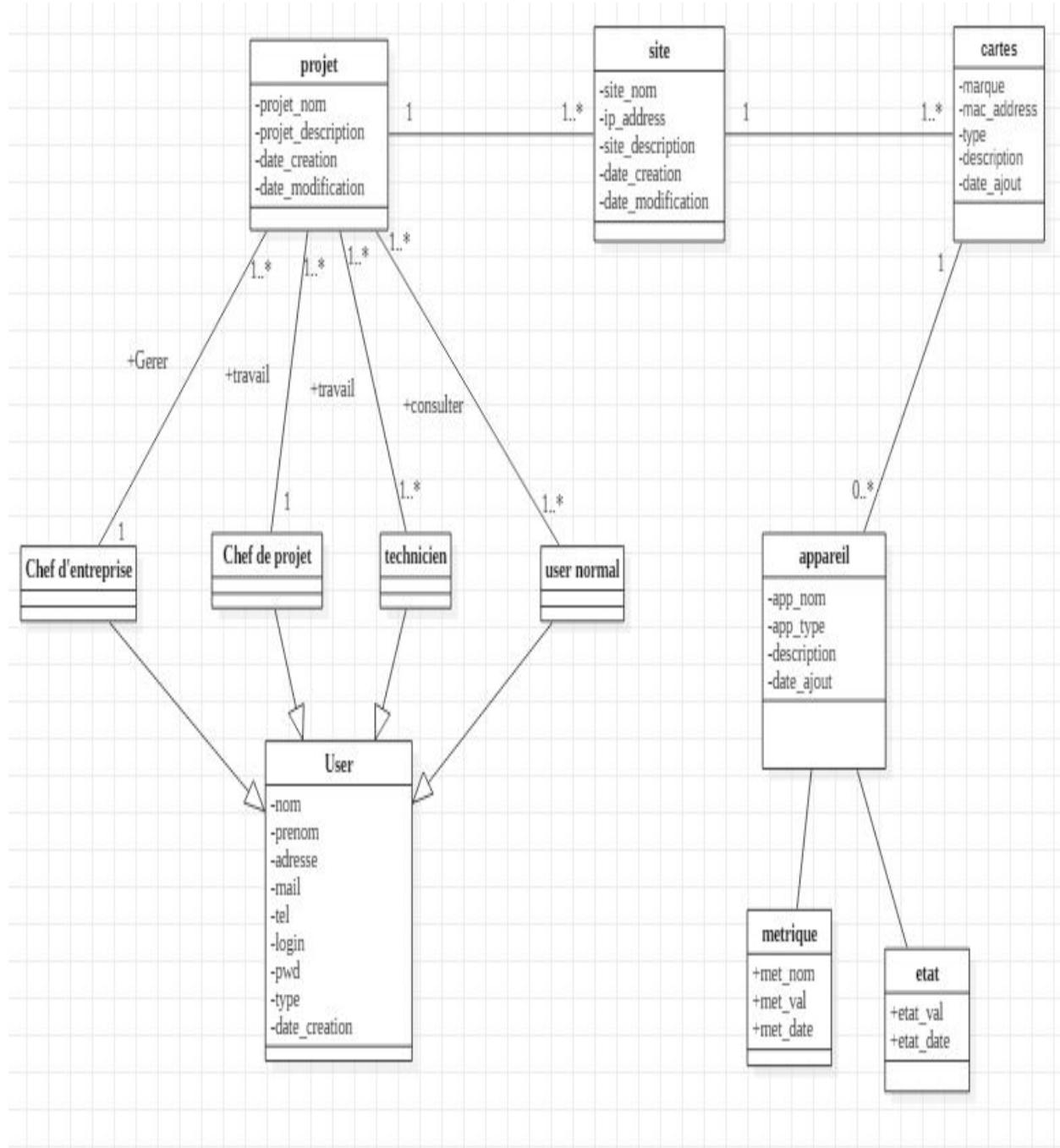


FIGURE II.13 – Diagramme de classe

II.5.4.1 Description des attributs des classes

Le tableau ci-dessous énumère l'ensemble des attributs associés à chaque classe, leurs types de données, ainsi que leurs brèves descriptions

classe	Attribut	Description de l'attribut
User	id	Identifiant du l'utilisateur
	nom	Nom d'Utilisateur.
	prenom	Prénom d'Utilisateur
	Adresse	Adresse d'Utilisateur
	mail	Mail d'Utilisateur
	tel	Numéro de Téléphone d'Utilisateur
	pwd	Mot de passe d'Utilisateur.
Projet	date_creation	Date de création du compte
	id	Identifiant du projet
	projet_nom	Nom du projet
	projet_description	Description du projet
	date_creation	Date de création du projet
Site	date_modification	Date de dernière modification du projet
	id	Identifiant du site
	site_nom	Nom du site.
	ip_address	Adresse Ip de site.
	site_description	Déscription du site
	date_creation	Date de création du site
Carte	date_modification	Date de dernière modification du site
	projet_id	Identifiant de projet
	id	Identifiant de la carte
	marque	Marque de la carte.
	mac_address	Adresse mac de la carte.
	type	Type de la carte
Appareil	description	Description de la carte
	date_ajout	Date d'ajout de la carte
	site_id	Identifiant de site
	id	Identifiant de la l'appareil
	app_nom	NOM de l'appareil.
Metrique	app_type	Type de l'appareil .
	description	Description de l'appareil
	app_date_ajout	Date d'ajout de l'appareil
	carte_id	Identifiant de la carte
	id	Identifiant de la metrique
Etat	met_nom	Nom de la métrique
	met_val	la valeur lu par le capteur.
	met_date	Date de recolte de la metrique
	app_id	Identifiant de la l'appareil
Etat	etat_id	Identifiant de l'état
	etat_val	La valeur lu par l'actionneur.
	etat_date	Date de recolte de l'état
	app_id	Identifiant de la l'appareil

TABLE II.6 – Liste des attributs associés à chaque classe

II.5.5 Les règles de passage du diagramme de classe vers le modèle relationnel

Règle 1 : Transformation des classes : Chaque classe devient une relation. L'identifiant (respectivement les attributs) de la classe devient la clé primaire (respectivement des attributs) de la relation.

Règle 2 : Association un à plusieurs : Il faut ajouter un attribut de type clé étrangère dans la relation fils de l'association. L'attribut porte le nom de la clé primaire de la relation père de l'association.

Règle 3 : Associations plusieurs à plusieurs ou classes-associations : L'association (classe-association) devient une relation dont la clé primaire est composée par la concaténation des identifiants des classes connectées à l'association (classe association). Les attributs de l'association (classe-association) doivent être ajoutés à la nouvelle relation. Ces attributs ne sont ni clé primaire, ni clé étrangère.

Règle 4 : Association un à un : Il faut ajouter un attribut clé étrangère, dans la relation dérivée de la classe ayant la multiplicité minimale égale à un. L'attribut porte le nom de la clé primaire de la relation dérivée de la classe connectée à l'association.

Règle 5 : Héritage

Décomposition par distinction : La clé primaire de la surclasse migre dans les relations issues des sous-classes et devient à la fois clé primaire et clé étrangère. [10].

II.5.6 Le modèle relationnel des données

Le modèle relationnel des données de l'application est représenté comme suit :

- Chef_entreprise (**id_chef_en**, #**user_id**)
- Chef_projet (**id_chef_pro**, #**user_id**)
- Technicien (**id_tec**, #**user_id**)
- User_normal(**id_normal**, #**user_id**)
- User (**id**, nom, prenom, adresse, mail, tel, password, date_creation)
- Projet (**id**, projet_nom, projet_description, date_creation, date_modification, #**user_id**)
- Site (**id**, site_nom, ip_address, site_description, date_creation, date_modification, , #**projet_id**)
- Carte (**id**, marque, mac_address, type, description, date ajout, #**site_id**)
- Appareil (**id**, app_nom, app_type, description, app_date ajout, #**carte_id**)
- metrique (**id**, met_nom, met_val, met_date, #**app_id**)
- etat (**id**, etat_val, etat_date, #**app_id**)

Notation :

- Les clés primaires sont en gras.
- Les clés étrangères sont en gras et précédées par #.

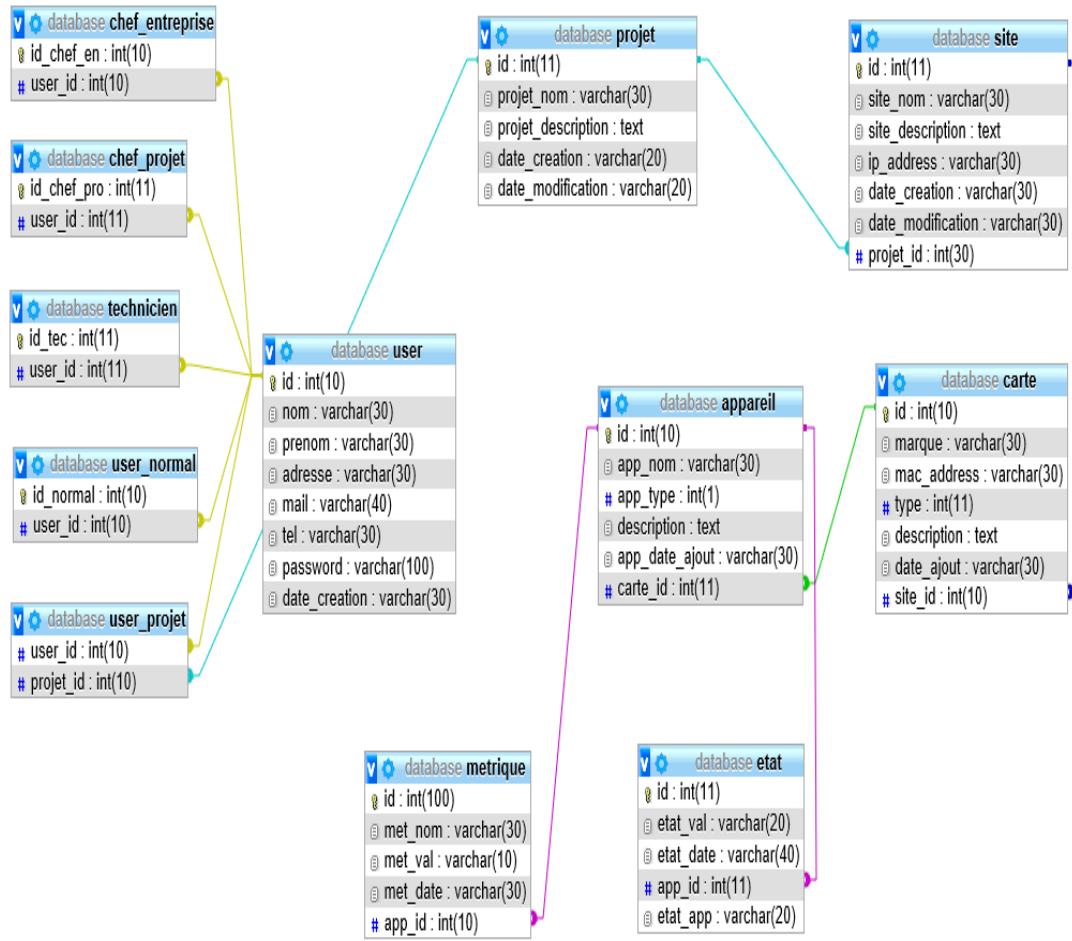


FIGURE II.14 – Schémas relationnel des données

II.6 Conclusion

Ce chapitre nous a permis de présenter le principe général de notre architecture et d'identifier les acteurs et les besoins fonctionnels et non fonctionnels de notre système tout en décrivant l'architecture de notre projet et détaillant les divers diagrammes de classe, de cas d'utilisation et de séquence qui va nous permettre d'entamer la phase de réalisation proprement dite. Le chapitre suivant sera consacré à la réalisation du système et à une synthèse des résultats obtenus.

Chapitre III

Réalisation et tests

III.1 Introduction

Dans ce qui suit nous allons décrire le processus de réalisation de notre système, en d'autres termes la mise en oeuvre de ce qui a été énoncé dans le chapitre précédent. Nous mettrons en évidence l'ensemble des environnements (logiciels et matériels) de développement, l'architecture matérielle du système, ainsi qu'un aperçu sur les interfaces de l'application WEB sans oublier les scénarios testés et les résultats obtenus.

III.2 Outils et environnement de travail

Nous présentons dans cette section l'environnement matériel mis à la disposition du présent projet, ainsi que l'environnement logiciel utilisé pour le développement et la mise en place de notre application sans oublier les différents outils mis à la disposition pour la réalisation de notre système.

III.2.1 Partie matérielle

III.2.1.1 Cartes et capteurs utilisés

- NodeMCU esp8266 :

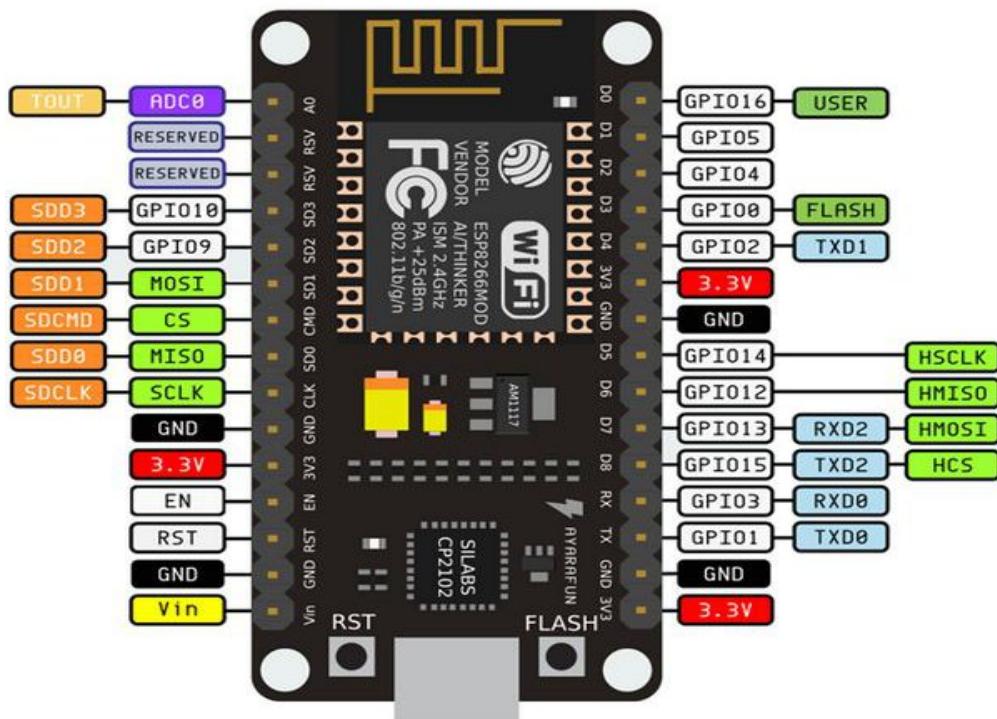


FIGURE III.1 – Le module NodeMCU Lolin ESP8266

[11]

L'ESP8266 est un circuit intégré à microcontrôleur (32-bit RISC CPU Tensilica Xtensa LX106 à 80 MHz) avec connexion WiFi développé par le fabricant chinois Espressif.

L'ESP8266 peut se programmer de plusieurs façons :

- Avec des scripts Lua avec le firmware NodeMCU
- En C++ avec l'IDE Arduino
- En MicroPython

- **DHT22**

Le capteur de température et d'humidité DHT22 communique avec un microcontrôleur pour lire la température et l'humidité de l'air. Le capteur est calibré et ne nécessite pas de composants supplémentaires pour pouvoir être utilisé.

- Alimentation : 3,3 à 6 Vcc
- Consommation max : 1,5 mA
- Consommation au repos : 50 μ A
- Plage de mesure :
- température : -40 à +80 °C.
- humidité : 0 à 100 % RH.
- Précision :
- température : \pm 0,5 °C.
- humidité : \pm 2 % RH.
- Dimensions : 25 x 15 x 9 mm[\[12\]](#).

- **Capteur de mouvement PIR :**

Les capteurs PIR, pour Passive Infrared Sensor (capteur à infrarouge passif) qui mesure les radiations IR émises par les objets se trouvant dans son champ de vision. Il est généralement utilisé dans divers systèmes de sécurité et détection d'intrusion. On en trouve à bas coût, et ils sont très simples à utiliser. Caractéristiques :

- Capteur de type HC SR501 ;
- Tension de fonctionnement : 5V à 20V continu (DC) ;
- Consommation statique : 65 A ;
- Niveaux de sortie : High 3.3 V, Low 0 V ;
- Temps de délai ajustable : de 0.3 secondes à 18 secondes ;
- Temps mort : 0.2 secondes.
- Déclenchement : L sans répétition , H répétition possible, valeur H par défaut.
- Angle de détection : angle de moins de 120°. Distance de détection : 7 mètres
- Température de fonctionnement de -15°C à +70 °C.[\[13\]](#).

- **Photorésistance :**

Les photorésistances (PhotoCells ou CdS en anglais) sont des capteurs qui permettent de mesurer le taux de luminosité. Elles sont petites, bon marché, économiques en énergie, faciles à utiliser [\[14\]](#), dispose d'une résistance qui varie en fonction de la lumière qu'elle capte, cette résistance diminue en fonction de la lumière captée.

L'utilisation est diverse et variée :

- Mesure de la lumière ambiante pour une station météo.
- DéTECTEUR de lumière dans une pièce.
- Suiveur de lumière dans un robot.

III.2.1.2 Outils de développement

- **Arduino IDE :** Le logiciel Arduino est un environnement de développement (IDE) open source, gratuit et téléchargeable sur le site officiel d'arduino. Cet IDE dédié aux cartes Arduino permet :

- d'éditer un programme ;
- de compiler ce programme dans le langage « machine » de l'Arduino,
- de téléverser le programme dans la mémoire de l'Arduino,
- de communiquer avec la carte NodeMCU grâce au terminal .

Le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C). Une fois le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information. [15].

- **Mostquitto :**

Eclipse Mosquitto est un serveur MQTT (Broker) open source (licence EPL / EDL) qui implémente les versions de protocole MQTT 5.0, 3.1.1 et 3.1. Mosquitto est léger et convient à une utilisation sur tous les appareils, des ordinateurs monocartes basse consommation aux serveurs complets.

Le projet Mosquitto fournit également une bibliothèque C pour la mise en œuvre des clients MQTT, ainsi que les très populaires clients : MQTT mosquitto_pub et mosquitto_sub.

Mosquitto fait partie de la fondation Eclipse et est un projet de iot.eclipse.org. [16]

III.2.2 Partie logicielle

III.2.2.1 Langages de programmation et Framework :

- **PHP** : PHP (pour pré HyperText Processor) est un langage de script open source disponible pour diverses plates-formes (Unix, Linux et Windows) comparable à ASP de Microsoft. Il prend en charge l'ensemble des protocoles du web (HTTP, SMTP, LDAP, etc) et offre un accès natif aux principales bases de données du marché. PHP offre toutes les fonctionnalités utiles pour construire des sites web dynamiques sophistiqués[17] Le langage PHP compte diverses manières d'utilisation :

- Pour une interface web : c'est l'utilisation la plus courante ;
- En ligne de commandes (CLI "Command Line Interface") ;
- Pour produire une interface desktop (GUI "Graphical User Interface ") ;[18]

- **JavaScript :**

JavaScript est un langage de script orienté objet principalement utilisé dans les pages HTML. A l'opposé des langages serveurs (qui s'exécutent sur le site), Javascript est exécuté sur l'ordinateur de l'internaute par le navigateur lui-même. Ainsi, ce langage permet une interaction avec l'utilisateur en fonction de ses actions (lors du passage de la souris au-dessus d'un élément, etc).[19]

- **sql (Structured Query Language) :**

SQL est un langage informatique normalisé qui sert à effectuer des opérations sur des bases de données. Parmi ces opérations nous pouvons citer : l'ajout, la modification et la suppression de données dans la base de données. C'est le langage utilisé par les SGBDR (système de gestion de base de données relationnels) [20].

- **Python :**

python est un langage de programmation interprété, multi-paradigme et multiplate-formes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions.

- **jQuery :**

jQuery est un framework JavaScript libre et Open Source, implanté côté client, qui porte sur l'interaction entre le DOM(Document Object Model), JavaScript, AJAX et Le Html. Cette librairie JavaScript a pour but de simplifier les commandes communes du JavaScript. La devise de jQuery est en effet, "Écrire moins pour faire plus" (write less do more).jQuery, du moins à l'origine, est l'œuvre d'un seul homme : John Resig. Ce jeune sur doué de JavaScript développa la première version de jQuery en janvier 2006.Les spécificités de jQuery sont nombreuses mais l'essentielle est assurément la souplesse qu'il apporte pour accéder à tous les éléments du document HTML grâce à la multitude de sélecteurs mis en place. Cette caractéristique fut d'ailleurs retenue pour donner un nom à ce framework :j pour JavaScript et Query pour chercher ouer aux éléments[21]

- **Bootstrap :**

Bootstrap est un framework CSS, mais pas seulement, puisqu'il embarque également des composants HTML et JavaScript. Il comporte un système de grille simple et efficace pour mettre en ordre l'aspect visuel d'une page web. Il apporte du style pour les boutons, les formulaires, la navigation...etc. Il permet ainsi de concevoir un site web rapidement et avec peu de lignes de code ajoutées. Le framework le plus proche de Bootstrap est sans doute Foundation qui est présenté comme « The most advanced responsive front-end framework in the world » [22].

III.2.2.2 Outils de développement

Le développement d'un tel système nécessite l'utilisation de quelques outils. Dans ce qui suit, nous citons les outils qui ont été utilisés.

- **WampServer** WampServer (anciennement WAMP5), tient son nom des trois logiciels « Windows Apache MySQL, PHP5.6 », est un environnement serveur tout en un, il nous permettra de développer nos applications PHP en toute tranquillité. Wampserver n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi qu'une administration pour deux bases SQL PHPMyAdmin et SQLite Manager[18].

- **phpMyAdmin**

L'outil phpMyAdmin est développé sous PHP en offrant une interface intuitive pour l'administrateur des bases de données du serveur. Cet outil permet de :

- Créer/modifier/supprimer des tables ;

- Afficher/ajouter/ modifier/ supprimer des tuples dans des tables ;
- Effectuer des sauvegardes de la structure et/ou des données ;
- Effectuer n'importe quelle requête ;
- Gérer les privilèges des utilisateurs [23]

III.3 L'application web

III.3.1 Présentation de l'application web

L'accès à l'application web se fait après authentification de l'utilisateur. Ce dernier fournit au système ses identifiants et après vérification par le système accède à son espace personnel.

Nous avons quatre profil d'utilisateur à savoir le chef d'entreprise, le chef de projet, le technicien ou l'utilisateur normal. Les fonctionnalités fournis par le système dépendent du profil de l'utilisateur connecté comme nous pouvons le voir ici dans les interfaces graphiques.

III.3.2 Interfaces graphiques

Nous présenterons les résultats de l'implémentation de l'application web à travers des captures d'écran qui expliquent quelques fonctionnalités principales du système et les tests effectués .

III.3.2.1 Authentification

Cette première capture présente l'interface d'authentification dans laquelle on doit saisir l'adresse mail de l'utilisateur et le mot de passe. Cette interface constitue la fenêtre d'accueil de notre application. A travers cette fenêtre l'utilisateur s'authentifie pour utiliser l'application. Cette étape met en valeur l'aspect sécurité : nous vérifions la disponibilité du compte utilisateur et nous lui attribuons les droits et privilèges nécessaires d'accès.

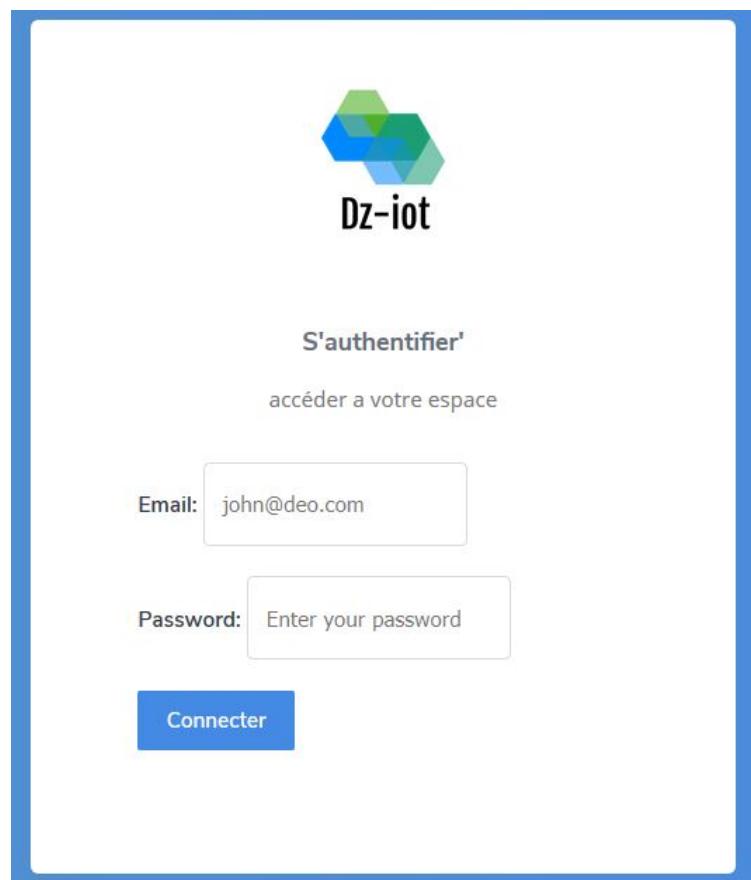


FIGURE III.2 – Authentification

III.3.2.2 Ajout d'un utilisateur

En cliquant sur le volet gestion d'utilisateur puis sur l'élément ajouter un utilisateur, un formulaire comportant plusieurs champs s'affiche pour l'entrée des données. L'application gère le contrôle de saisie ainsi que la sauvegarde dans les bases de données.



AJOUTER UN UTILISATEUR

Nom
Michael Zenaty

Prenom
Michael Zenaty

Adresse
Bab Ezzouar

Email address
john@deo.com

Tel
+213790226698

Type Utilisateur Chef d'entreprise

Password
Enter your password

Password
Confirm your password

Valider

FIGURE III.3 – Ajout d'un utilisateur

III.3.2.3 Liste des projets

Consulter permet de voir la liste des projets et leurs informations. On peut sélectionner un projet pour le modifier ou le supprimer.



A Propos

NAVIGATION

- Gestion Des Utilisateurs
- Gestion Des Projets
 - Liste de projet
 - Ajouter Un Projet
 - Modifier Un Projet
 - Supprimer Un Projet
 - Assigner Un Chef De Projet A Un Projet
 - Assigner Un Technicien A Un Projet
 - Assigner Un Utilisateur A Un Projet
- Gestion des sites
- Gestion Des Cartes

Id	Nom du projet	Description	Date de creation	Date de modification	
2	projet1	Projet de surveillance de l'université Houari Boumediene	2019-04-20 14:07:17	2019-05-08 14:31:01	 
4	sonelgaz	projet realiser dans le cadre de surveillance et contrôle de la centrale électrique de l'hamra	2019-07-04 14:25:13	2019-07-04 14:25:13	 

FIGURE III.4 – Liste des projets

III.3.2.4 Modifier un utilisateur

A l'aide de cette page, le chef d'entreprise peut modifier les données d'un utilisateur après avoir choisi un nom d'utilisateur dans la liste déroulante (Figure III.5,partie 1),un formulaire s'affiche contenant les informations concernant l'utilisateur choisi (Figure III.6,partie 2).

1

CHOISIR UN UTILISATEUR

Utilisateur: Djibrilla Boukary Yacine

modifier

2

MODIFIER UN UTILISATEUR

Nom: Djibrilla Boukary

Prenom: Yacine

Adresse: résidence universitaire

Email address: yacine@usthb.dz

Tel: +213790226698

Type Utilisateur: Chef de projet

Password: Enter your password

Confirm Password: Confirm your password

Valider

FIGURE III.5 – Modifier un utilisateur

III.3.2.5 Visualisation des métriques

Lorsque l'utilisateur clique sur la section « mes projets » dans le menu, une interface de liste des projets existant auxquels l'utilisateur a accès s'affichent ; après la sélection d'un projet l'interface ci-dessous s'expose (Figure III.6).Elle permet une visualisation des différents métriques dans des jauge ainsi que l'historique des métriques sous forme de graphes comme le montrent les figures :(Figure III.7),(Figure III.8)(Figure III.9).

CHAPITRE III. RÉALISATION ET TESTS

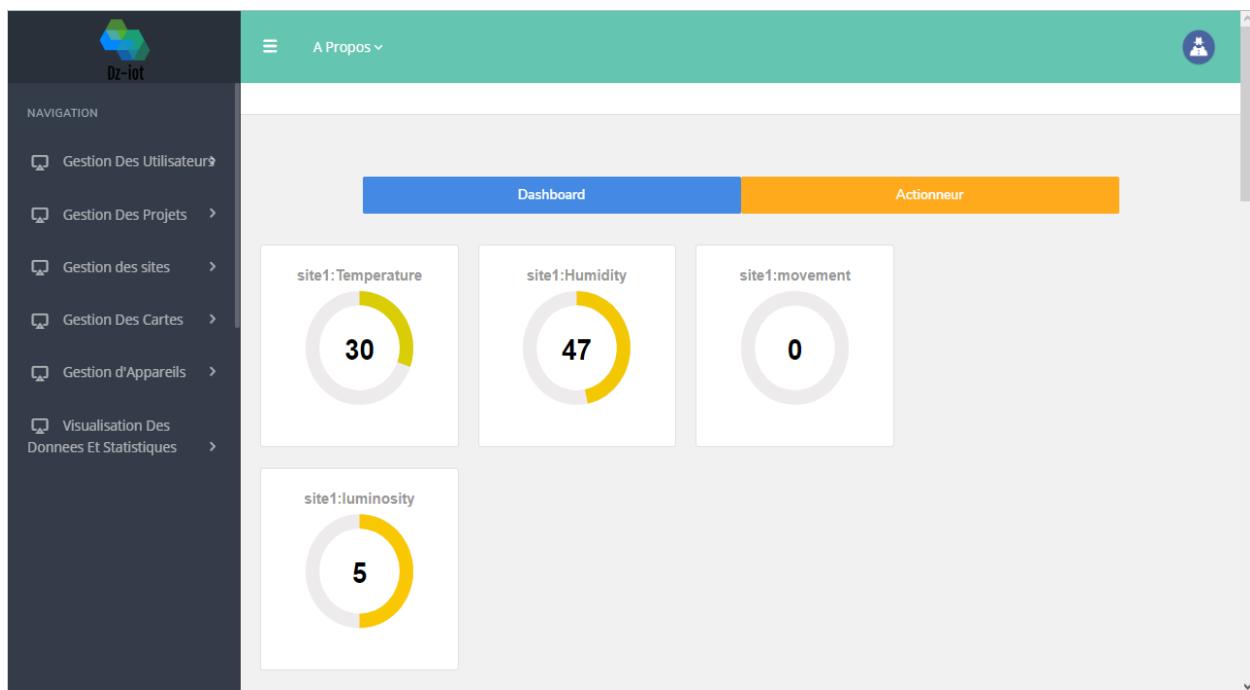


FIGURE III.6 – Collecte des métriques en temps réel

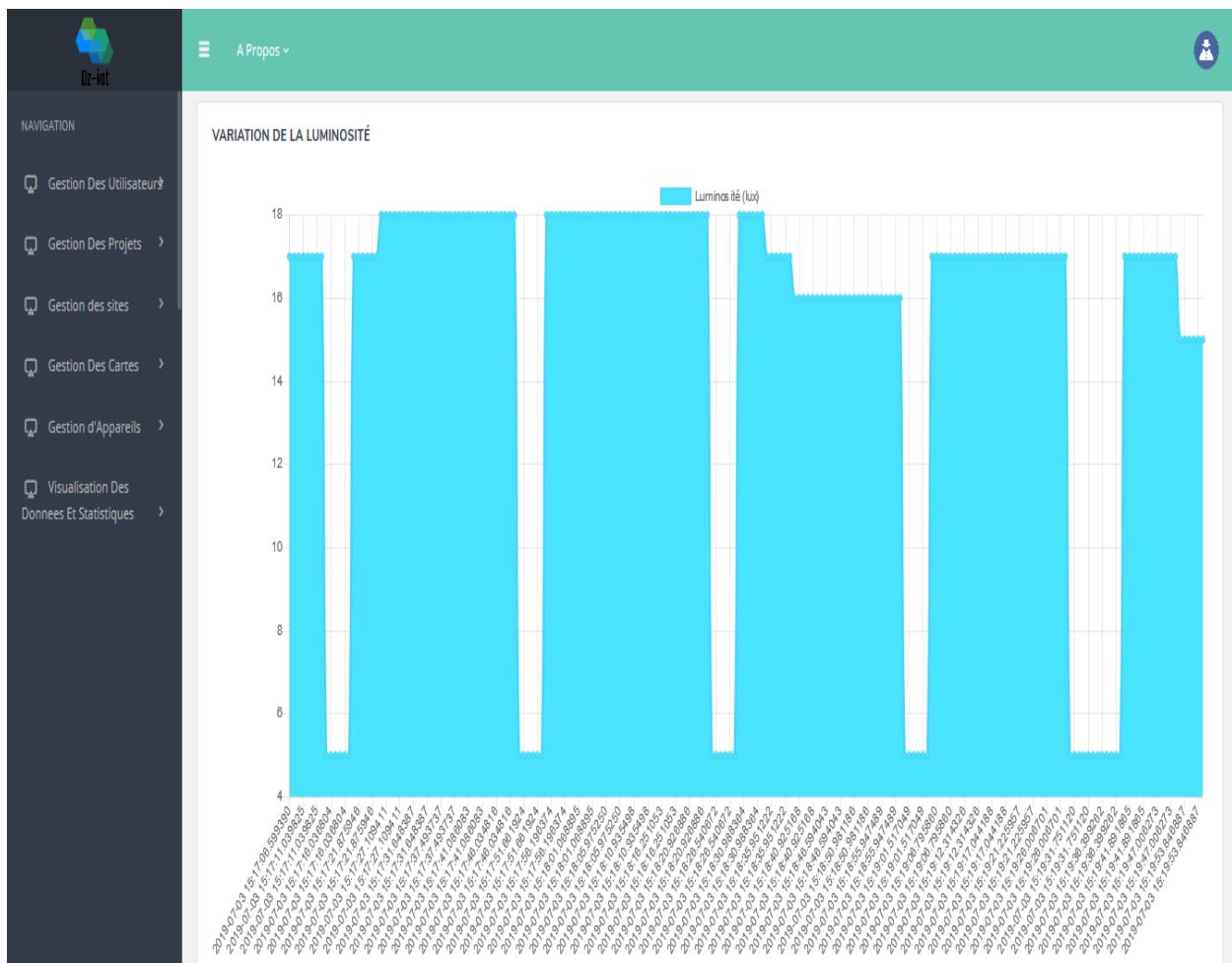


FIGURE III.7 – Graphe de variation de l'humidité en fonction de temps

CHAPITRE III. RÉALISATION ET TESTS

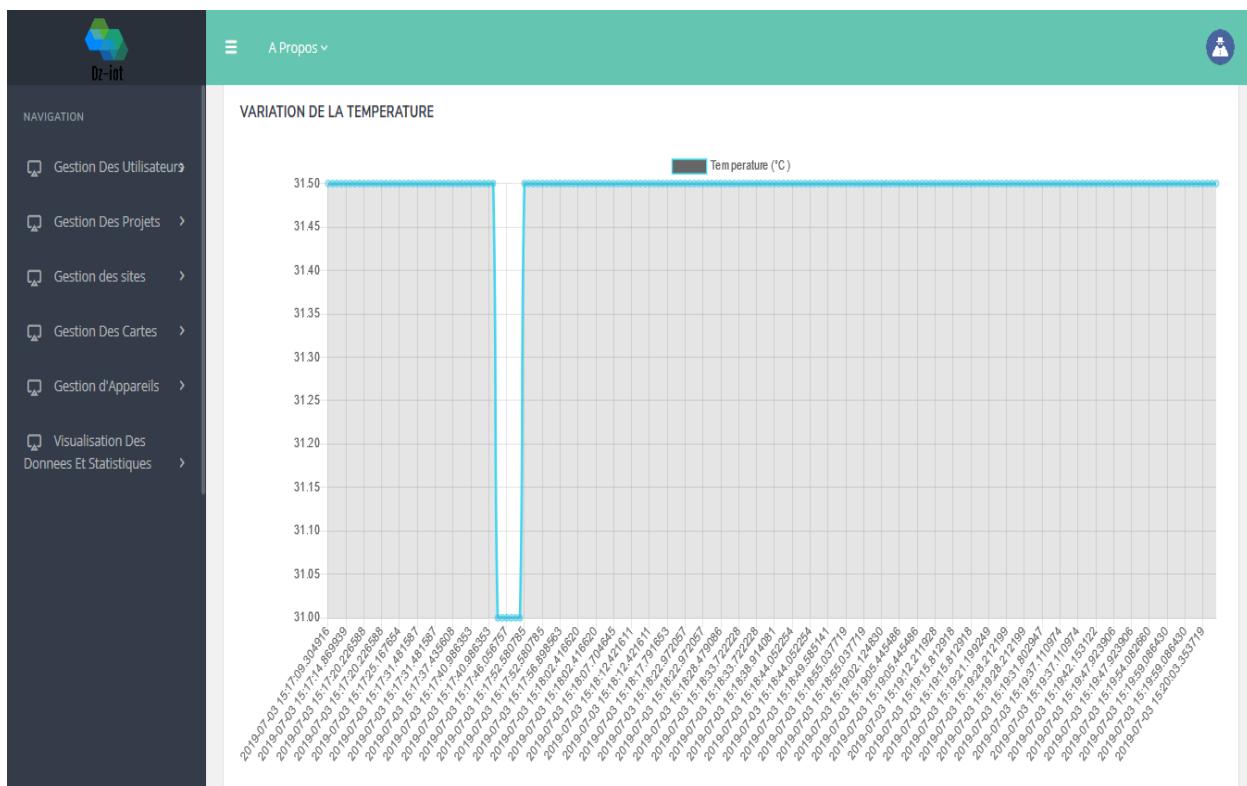


FIGURE III.8 – Graphe de variation de la température en fonction de temps

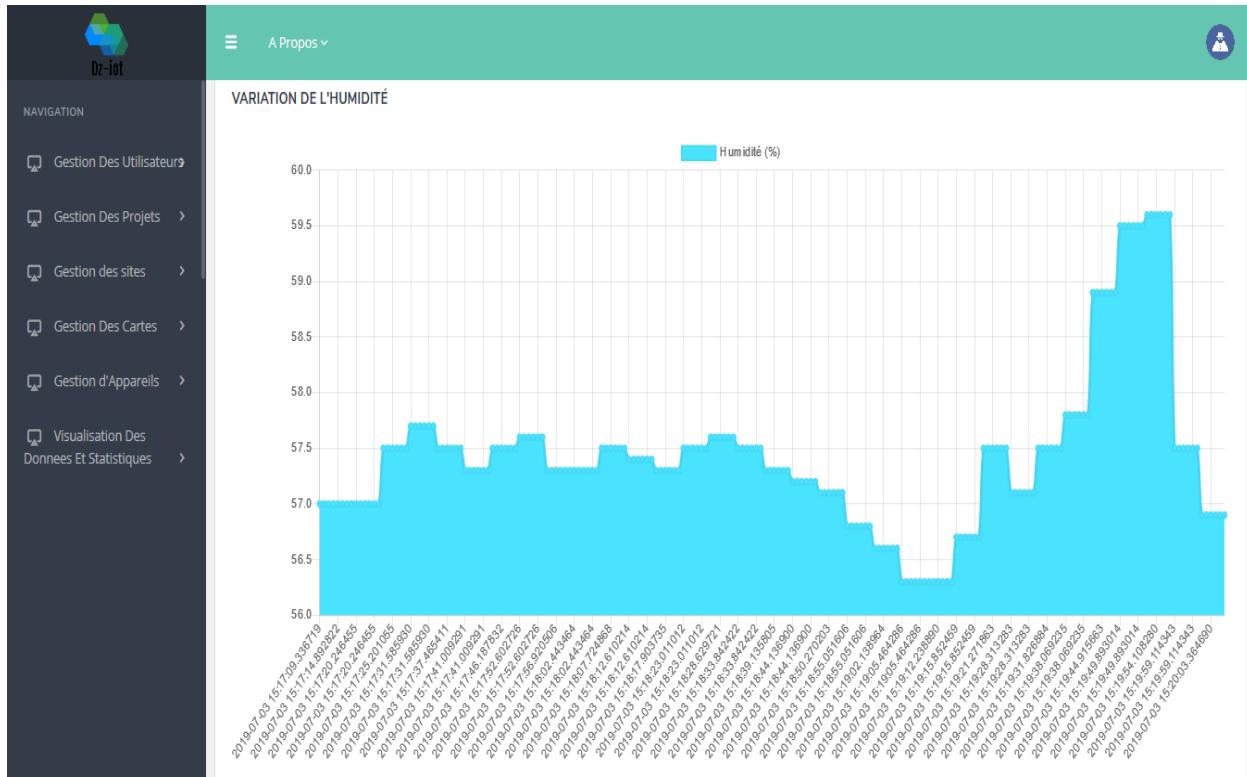


FIGURE III.9 – Graphe de variation de la luminosité en fonction de temps

III.3.2.6 Espace Actionneurs

Lorsque l'utilisateur clique sur la section « mes projets » dans le menu, une interface de liste des actionneurs existant à lesquels l'utilisateur à accès s'affichent ; L'utilisateur peut agir sur ces derniers en cliquant sur les buttons par exemple ON /OFF

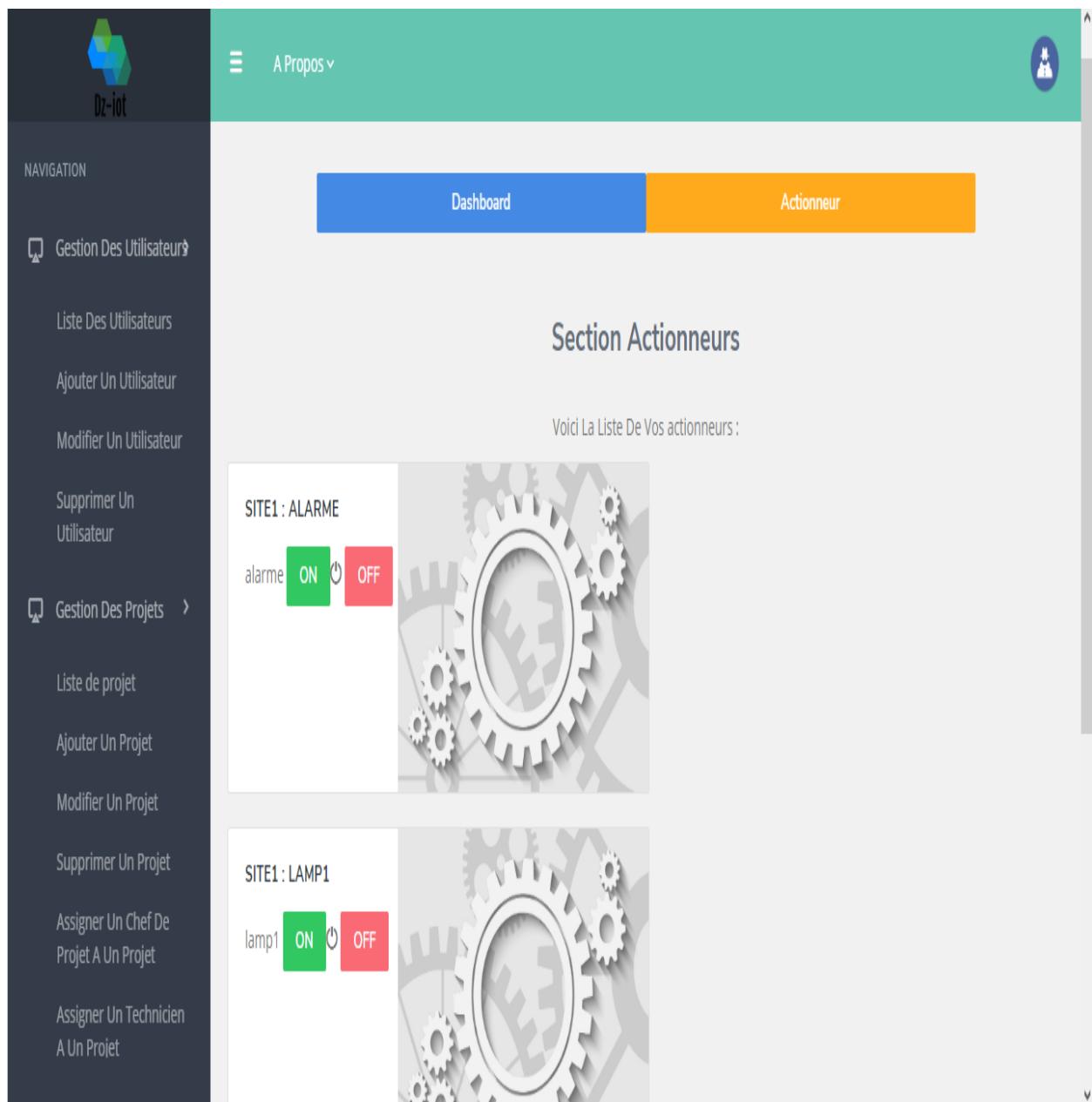


FIGURE III.10 – Espace Actionneurs

III.4 Tests effectués

III.4.1 Schéma matériel

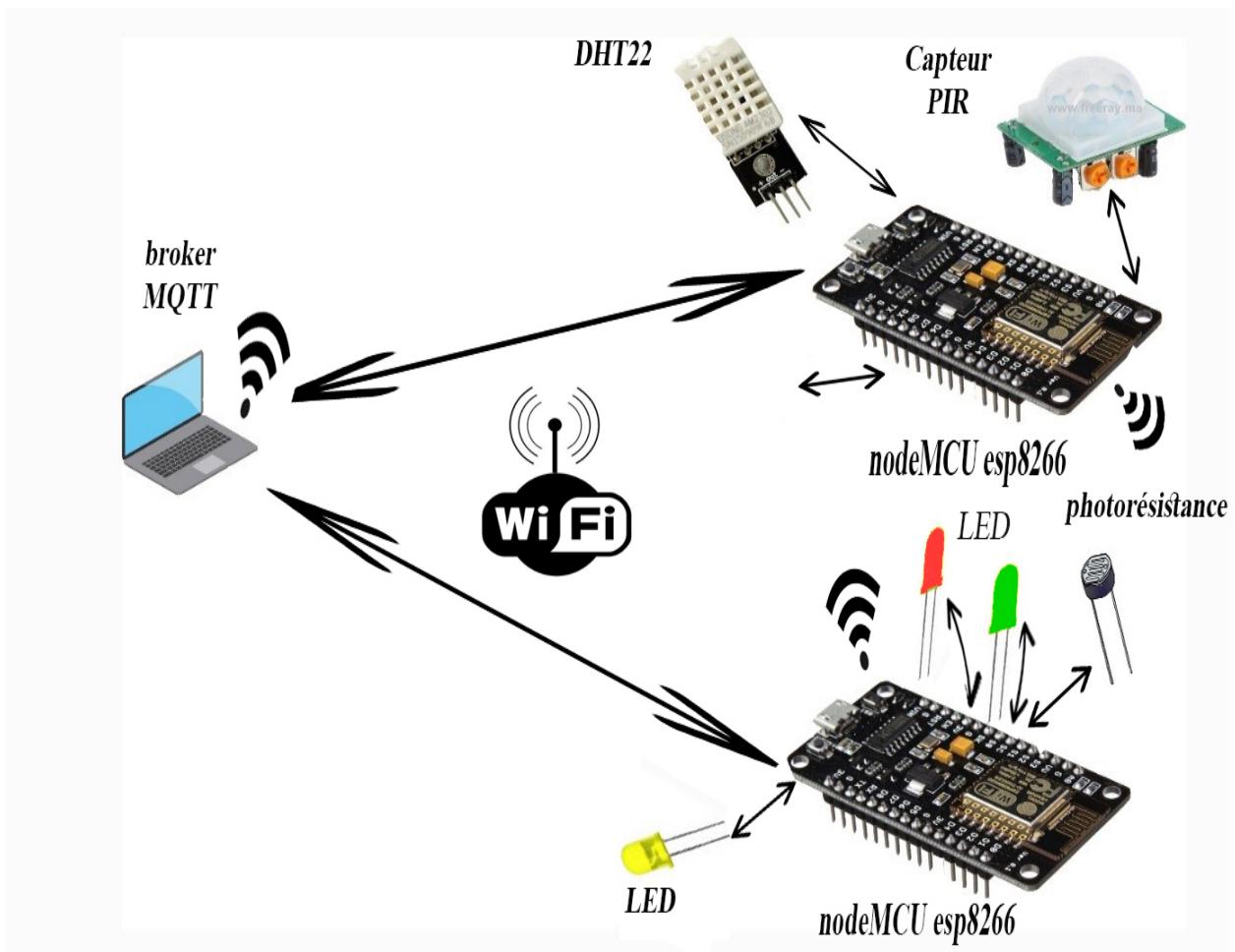


FIGURE III.11 – Architecture matérielle

III.4.2 Scénarios testés

III.4.2.1 Scénario1(Collecte des données)

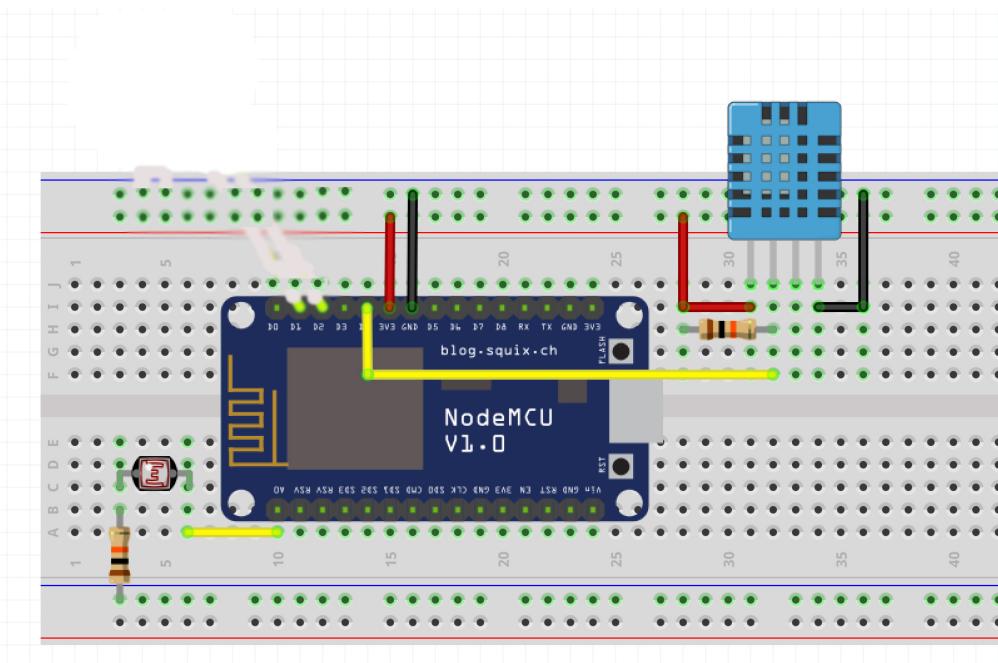


FIGURE III.12 – Montage collecte des métriques

Dans ce scénario , les données de l'environnement physique collectées par les capteurs sont stockés dans une base de donnée et peuvent être visualisées directement sur une page web comme nous le montre la(Figure III.6)

III.4.2.2 Scénario2(Commande à distance)

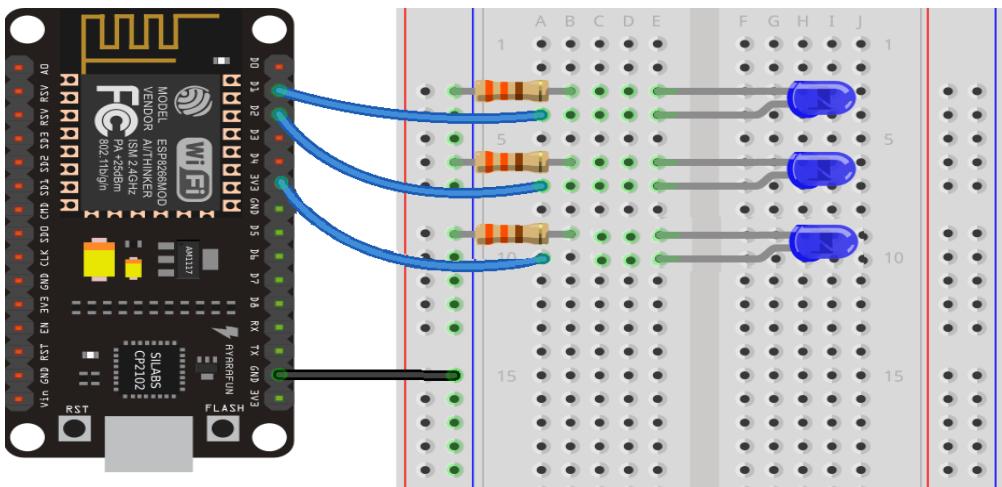


FIGURE III.13 – Montage commande des actionneurs à distance

Pour la mise en œuvres de notre cas commander un actionneur l'utilisateur choisit via l'interface web l'actionneur à actionner (FIGURE III.10). Celui ci peut être mis à ON ou à OFF en cliquant sur le bouton adéquat. Dans notre cas nous avons simuler un scénario en prenant des leds comme actionneurs . Une mise à ON équivaut à un allumage de la led et une mise à OFF à une extinction.

III.4.2.3 Scénario3(Détection de mouvement)

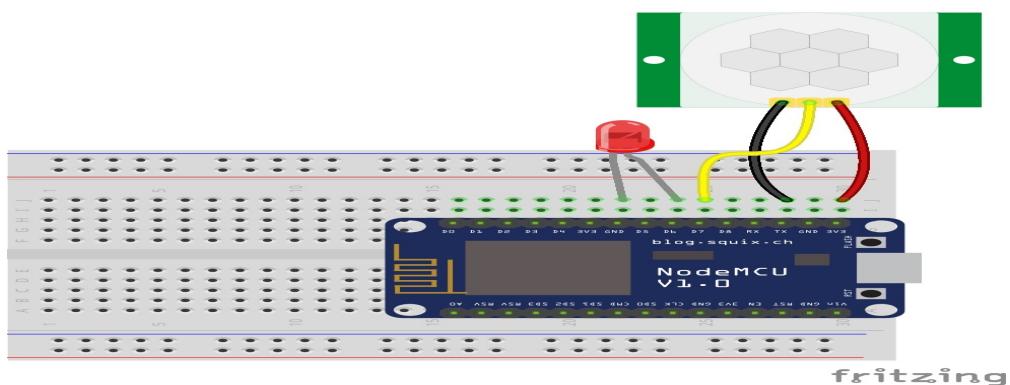


FIGURE III.14 – Montage détection de mouvement

Le système réalisé nous permet également de détecter des mouvements sur le lieu de surveillance, ce qui sert grandement pour prévenir des intrusions , notamment en déclenchant une alarme en cas de détection de mouvements. L'alarme est simulée ici par une LED comme le montre la figure III.14 .

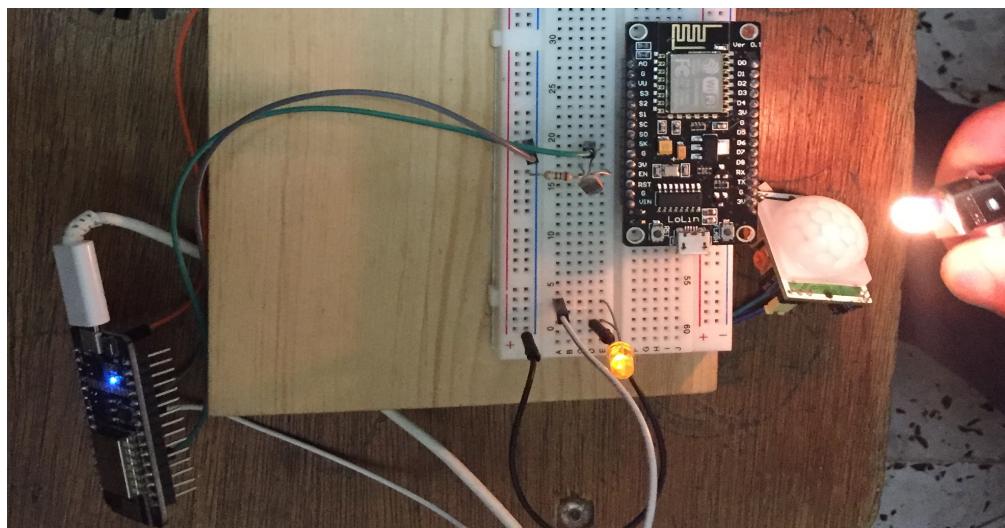


FIGURE III.15 – Détection de mouvement

III.4.2.4 Scénario4(Interactivité automatique avec l'environnement)

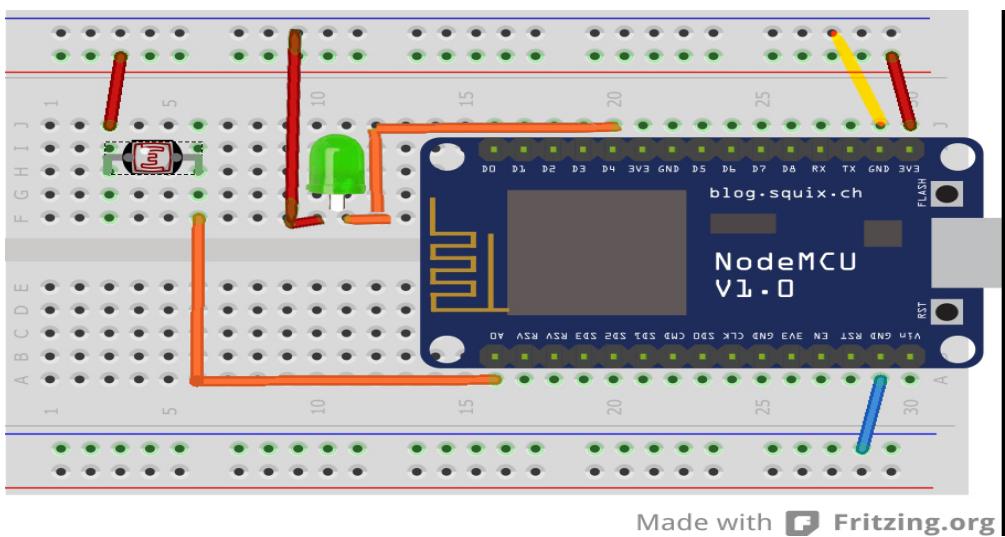


FIGURE III.16 – Montage allumage d'une lampe commandé par un seuil

Le système réalisé nous permet également la régulation automatique de l'éclairage d'un site distant. En effet sous un certains seuil de luminosité, l'éclairage automatique est déclenché. Nous simulons les lampadaires destinés à l'éclairage par des LEDs.

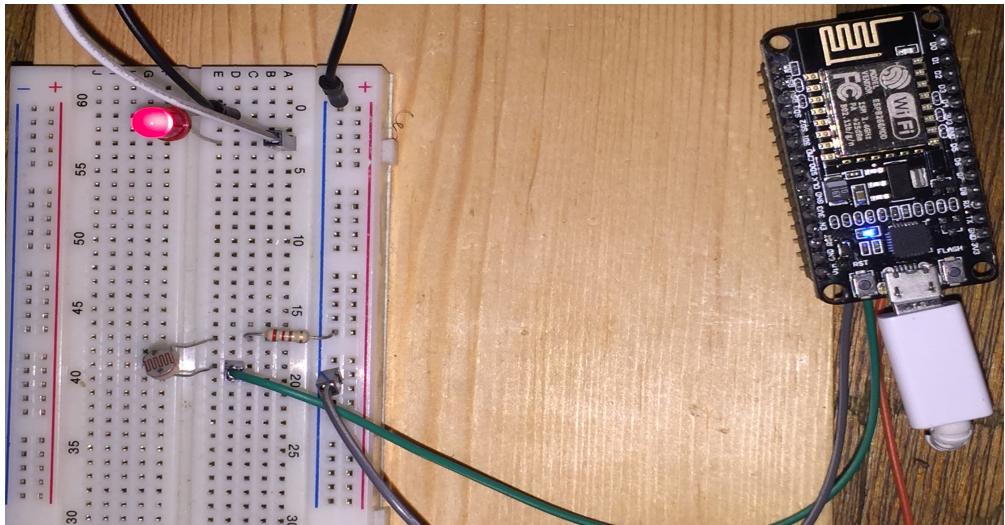


FIGURE III.17 – Allumage d'une lampe commandé par un seuil

III.5 Conclusion :

Ce dernier chapitre a été consacré à la présentation des résultats des tests d'évaluation de notre système selon les critères jugés pertinents. Nous avons également présenter notre application WEB permettant l'interfaçage entre l'utilisateur et le système.

Les résultats obtenus dans ce chapitre nous ont permis de conclure que le système répond aux objectifs fixés

Conclusion générale

Tout au long de cette année, ce travail nous a permis de découvrir le monde très prometteur de l'internet des objets, de mettre en pratique les connaissances acquises au cours de notre cursus universitaire en terme développement web (du frontend en allant au backend) et également d'acquérir de nouvelles connaissances en électronique et en programmation de microcontrôleurs.

La réalisation de notre travail s'est déroulée en trois étapes hiérarchique : la première étape consistait à faire une recherche d'une manière générale sur l'internet des objets et comprendre le concept pour en déduire les différents domaines d'applications . Une deuxième étape qui était la conception du système où nous avons pu mettre en pratique nos connaissances acquises en matière de modélisation UML et de communications M2M.

Le système réalisé n'est qu'un prototype générique d'un système de surveillance basée sur l'IoT et adaptable à tout environnement, il peut être améliorer pour prendre également en compte la gestion, de configuration des matérielles et s'adapter automatiquement et ce de manière dynamique à l'environnement d'implémentation et selon les besoins.

Au final étant donnée que nul ne peut se prétendre aborder un domaine dans son ensemble nous souhaiterons venir :

- Améliorer les interfaces pour quelles répondent aux critères ergonomique.
- Établir un système de sécurité des bases de données et limiter le nombre de tentatives d'authentification à l'application.
- Héberger l'application sur un serveur.
- Développer une application mobile
- Prendre en compte la question des Big Data.

ANNEXE

Annexe A

Code source

Programmation des cartes nodeMCU

Configuration de 1^{ère} carte

```
1 #include <ESP8266WiFi.h>
2 #include <PubSubClient.h>;
3 #include <PubSubClient.h>;
4 #include "DHT.h"; // Librairie des capteurs DHT
5 #define wifi_ssid "iPhone de yacine" // ssid du point d'accès
6 #define wifi_password "00000000" // mot de passe du point d'accès
7 WiFiClient espclient;
8
9 #define DHTPIN 4 // Pin sur lequelle le capteur DHT est branché
10 #define PIRPIN 2 // Pin sur lequelle le capteur de mouvement est branché
11
12 #define DHTTYPE DHT22 // Définition du type de capteur = DHT 22 (AM2302)
13 DHT dht(DHTPIN, DHTTYPE); // Initialisation
14
15
16 #define mqtt_serveur "172.20.10.9"
17 #define mqtt_user ""
18 #define mqtt_pwd ""
19 // format des topics projet/site/nom_de_l_appareil/nom_de_la_mesure
20 #define temperature_topic "cub3/N38/dht22/Temperature" // topic sur lequel est publié la température
21 #define humidity_topic "cub3/N38/dht22/Humidity" // topic sur lequel est publié l'Humidity
22 #define movement_topic "cub3/N38/pir/movement" // topic sur lequel est publié la présence de mouvement
23
24 PubSubClient client(espclient);
25
26 void setup() {
27   pinMode(PIRPIN, INPUT);
28   Serial.begin(9600);
29   dht.begin();
30   delay(10);
31   setup_wifi();
32   delay(5000);
33   client.setServer(mqtt_serveur, 1883);
34 }
35
36
37 // Connexion au réseau WiFi
38 void setup_wifi() {
39   delay(10);
40   Serial.println();
41   Serial.print("Connexion à ");
42   Serial.println(wifi_ssid);
43
44 WiFi.begin(wifi_ssid, wifi_password);
45
46 while (WiFi.status() != WL_CONNECTED) {
47   delay(500);
48   Serial.print(".");
49 }
50
51 Serial.println("");
52 Serial.println("Connexion WiFi établie ");
53 Serial.print("=> Adresse IP : ");
54 Serial.print(WiFi.localIP());
55
56
57 // Reconexion
58 void reconnect() {
59   // Si aucun signal obtient une reconexion
60   while (!client.connected()) {
61     Serial.print("Connexion au serveur MQTT...");
62     if (client.connect("ESP8266Client", mqtt_user, mqtt_pwd)) {
63       Serial.println("Connected");
64     } else {
65       Serial.print("KO, erreur : ");
66       Serial.print(client.state());
67       Serial.println(" On attend 5 secondes avant de recommencer");
68       delay(5000);
69     }
70   }
71 }
72
73
74 void loop() {
75   if (!client.connected()) {
76     reconnect();
77   }
78   char * sensorID="DHT0001";
79   float h = dht.readHumidity(); // Lecture du taux d'humidité
80   float t = dht.readTemperature(); // Lecture de la température en Celsius
81   float rh = dht.readHumidity(true);
82   if (isnan(h) || isnan(t)) {
83     Serial.println("Failed to read from DHT sensor!");
84     return;
85   }
86
87   Serial.print("Humidity: ");
88   Serial.print(h);
89   Serial.print(" %t");
90   Serial.print("Temperature: ");
91   Serial.print(t);
92   Serial.print(" °C ");
93 }
```

Script de la 1^{ère} carte

Configuration de 2^{ème} carte

```

6   #include <ESP8266WiFi.h>
7
8   #include <PubSubClient.h>
9
10  #define ledrouge 2
11  #define ledbleu 4
12  #define ledverte 12
13  #define photoresistence A0
14
15  WiFiClient espclient;
16  PubSubClient client(espclient);
17
18  void reconnectmqttserver() {
19    while (!client.connected()) {
20      Serial.print("Attempting MQTT connection...");
21      String clientId = "ESP8266Client-";
22      clientId += String(random(0xffff), HEX);
23      if (client.connect(clientId.c_str())) {
24        Serial.println("connected");
25        client.subscribe("cub3/N38/actionneur1/ledbleu");
26        client.subscribe("cub3/N38/actionneur2/ledrouge");
27        client.subscribe("cub3/N38/actionneur3/ledverte");
28      } else {
29        Serial.print("failed, rc=");
30        Serial.print(client.state());
31        Serial.println(" try again in 5 seconds");
32        delay(5000);
33      }
34    }
35  }
36
37  void callback(char* topic, byte* payload, unsigned int length) {
38
39    if (strcmp(topic,"cub3/N38/actionneur2/ledrouge")==0){
40      String MQTT_DATA = "";
41      for (int i=0;i<length;i++) {
42        MQTT_DATA += (char)payload[i];
43      if (MQTT_DATA == "ON") {
44        digitalWrite(ledrouge,HIGH);
45      }
46      if (MQTT_DATA == "OFF") {
47        digitalWrite(ledrouge,LOW);
48      }
49    }
50
51    if (strcmp(topic,"cub3/N38/actionneur2/ledbleu")==0) {
52      String MQTT_DATA = "";
53      for (int i=0;i<length;i++) {
54        MQTT_DATA += (char)payload[i];
55      if (MQTT_DATA == "ON") {
56        digitalWrite(ledbleu,HIGH);
57      }
58      if (MQTT_DATA == "OFF") {
59        digitalWrite(ledbleu,LOW);
60      }
61    }
62
63    if (strcmp(topic,"cub3/N38/actionneur3/ledverte")==0) {
64      String MQTT_DATA = "";
65      for (int i=0;i<length;i++) {
66        MQTT_DATA += (char)payload[i];
67      if (MQTT_DATA == "ON") {
68        digitalWrite(ledverte,HIGH);
69      }
70      if (MQTT_DATA == "OFF") {
71        digitalWrite(ledverte,LOW);
72      }
73    }
74  }
75
76  }
77}
78
79  void setup()
80  {
81    Serial.begin(9600);
82    WiFi.disconnect();
83    delay(3000);
84    Serial.println("START");
85    WiFi.begin("iPhone de yacine","00000000");
86    while ((!(WiFi.status() == WL_CONNECTED))){
87      delay(300);
88      Serial.print(".");
89    }
90    Serial.println("Connected");
91    Serial.println("Your IP is");
92    Serial.println((WiFi.localIP().toString()));
93    client.setServer("172.20.10.9", 1883);
94    client.setCallback(callback);
95
96    pinMode(ledrouge, OUTPUT);
97    pinMode(ledbleu, OUTPUT);
98    pinMode(ledverte, OUTPUT);
99  }
00

```

Script de la 2^{ème} carte

Script python

Connexion au broker MQTT et insertion dans la BDD

```

1  import paho.mqtt.client as mqttClient
2  import time
3  from pathlib import Path
4  from database import *
5
6  def on_connect(client, userdata, flags, rc):
7      if rc == 0:
8          print ("Connected to broker")
9          global connected #Use global variable
10         connected = True #Signal connection
11     else:
12
13         print ("Connection failed")
14
15    def on_message(client, userdata, message):
16        var=message.payload.decode("utf-8")
17        var=var.split("_")
18        print ("Message received: ",var)
19        print("Topic:",message.topic)
20        adddata(message.topic,var[0])
21
22    def on_publish(client,userdata,result):             #create function for callback
23        print("data published \n")
24
25
26
27    broker="localhost"
28    connected = False #global variable for the state of the connection
29    port = 1883
30    user = ""
31    password = ""
32    client = mqttClient.Client("Python") #create new instance
33    client.username_pw_set(user, password=password) #set username and password
34    client.on_connect = on_connect #attach function to callback
35    client.on_message = on_message #attach function to callback-
36    client.on_publish = on_publish
37    print("Connecting to broker", broker)
38    client.connect(broker, port=port) #connect to broker
39    client.loop_start()
40    while connected != True: #wait for connection
41        time.sleep(0.1)
42        client.subscribe("#")
43    try:
44        while True:
45            time.sleep(1)
46            fichier = open("C:/wamp64/www/iot/actionneur.txt", "r") #read topic and value in the file
47            data = fichier.readline()
48            temp=data.split("_")
49            topic=temp[0]
50            value=temp[1]
51            client.publish(topic,value) #publish the value in the topic
52
53    except KeyboardInterrupt:
54        print ("exiting")
55        client.disconnect()
56        client.loop_stop()
57

```

Connexion au broker MQTT et insertion dans la BDD

Bibliographie

- [1] Dorian Keuller, Le secteur de la santé face à l'émergence de l'Internet des Objets : développement d'un outil d'aide à la décision Mémoire-recherche présentée vue de l'obtention du Master en sciences de gestion (2016).
- [2] Jessica Murgia ,internet des objets,(fev.2017), <https://www.androidpit.fr/internet-des-objets-c-est-quoi> (consulté le 15/01/2019).
- [3] Rapport de CISCO Internet Business Solutions Group (IBSG) sur l'internet des objets 'Comment l'évolution actuelle d'Internet transforme-t-elle le monde ?' disponible sur : https://www.cisco.com/c/dam/global/en_ca/solutions/executive/assets/pdf/internetofthingsfr.pdf (consulté le 20/01/2019).
- [4] Frédéric Charles pour Green , Internet des objets : Lords of the (connected) rings <https://www.zdnet.fr/blogs/green-si/internet-des-objets-lords-of-the-connected-rings-39825158.htm> (consulté le 20/01/2019).
- [5] PwC france ,la bonne acrchiecture iot : <https://transformation-digitale.pwc.fr/data/la-bonne-architecture-iot> (consulté le 21/01/2019)
- [6] AlFuqaha, Ala, et al. "Internet of things : A survey on enabling technologies, protocols, and applications." IEEE Communications Surveys Tutorials 17.4, (2015) : 23472376.
- [7] Romain Tribout,protocoles applicatifs de l'iot . <https://www.frugalprototype.com/quels-protocoles-applicatifs-pour-l-internet-des-objets> (Consulté le 18/mars/2019).
- [8] Mr Hidjeb Ali ,dImplémentation d'un protocole d'élection d'un serveur d'authentification dans l'internet des objets Mémoire de fin de cycle de master en Administration et Sécurité des Réseaux de l'université de Bejaïa(2017).
- [9] Mle Chalal Lina et Mr Sirouakane Silimane. Gestion des clés dans l'internet des objets. Mémoire de fin de cycle de master en réseau et systèmes distribué de l'université de Bejaïa (2017).
- [10] AKOU Hamza, H ARKOUK Saïd, Mémoire de fin de cycle, Conception et Réalisation d'une Application Web de Gestion d'Hôtel Sous JAVA Cas : « Hôtel ROYAL » (2015).
- [11] T.Vaira,architecture ESP8266,
www.tvaira.free.fr/esp8266/nodemcu-lolin-esp8266.html consulter le (21/04/2019).
- [12] GOTRONIC, Composant electronique, <https://www.gotronic.fr/art-capteur-de-t-et-d-humidite-dht22-20719.html> (consulter le 21/04/2019).
- [13] Generation Robots,capteur PIR, <https://www.generationrobots.com/fr/capteur-pir-de-mouvement.html> (consulter le 23/04/2019).
- [14] Dominique Meurisse(2012),Photo-résistance,
[https://arduino103.blogspot.com/2012/05/photo-resistance-cds-photocells-en.html](http://arduino103.blogspot.com/2012/05/photo-resistance-cds-photocells-en.html) (consulter le 03/05/2019).
- [15] Intel IQ. IDEArduino. In <https://www.arduino.cc/en/Main/Education>, editor,booktitle, (consulter le 01/05/2019).
- [16] Projectsdiy(2016),Mosquitto, <https://projetsdiy.fr/mosquitto-broker-mqtt-raspberry-pi/> (consulter 20/04/2019).
- [17] J. MARIE. PHP/MYSQL avec Dreamweaver8, édition Eyrolles, 2011.

- [18] jean-m.Richer Développement Web PHP Avancé.disponible sur : <https://www.cours-gratuit.com/cours-php/cours-developpement-web-php-avance-pdf> (consulter 18/04/2019).
- [19] C. PORTENEUVE et T. NITOT. Bien d'evelopper pour le web 2.0 : Bonnes pratiques Ajax. Eyrolles, 2008.
- [20] R.Grin, Le langage SQL, version 2.3, Université de Nice SophiaAntipolis 2000.
- [21] L. Lancker. jQuery- le framwork JavaScript du web2.0 ,Eni,2014.
- [22] Bestmomo. Prenez en main Bootstrap,site du zero, 2012.
- [23] PH.Rigaux, Pratique de MySQL et PHP, édition DUNOD, 2009.