

## Département Technologies de l'informatique

### Mémoire de Projet de Fin D'étude

Présenté en vue de l'obtention du diplôme de

Licence Appliquée en technologies de l'informatique

**Sujet :**

## Développement d'une Plateforme IoT

**Entreprise d'accueil :**

## DR Technologies

Réalisé par :

**Ghassen Ouertani**

**Encadreur(s) à l'ISET :**

Mme Basma Oueslati

**Encadreur Professionnel :**

Mr Ahmed Charef

Année universitaire : 2017/2018

# Table des matières

Table des matières .....	i
Liste des figures .....	iv
Liste des Tableaux.....	vi
Liste d'abréviations .....	vii
Remerciements .....	viii
Dédicaces .....	ix
Introduction Générale.....	x
Chapitre 1 .....	1
Présentation du cadre de projet .....	1
Introduction .....	2
I. Présentation de l'entreprise .....	2
1. Description de l'entreprise .....	2
2. Les Services de l'entreprise : .....	3
II. Cahier de charges.....	3
1. Problématique .....	3
2. Etude de l'existant.....	4
3. Solutions existantes.....	4
4. Solution proposée.....	7
5. Cible du projet.....	7
III. Etat de l'art .....	8
1. Internet des objets .....	8
2. Couches du modèle IoT .....	9
3. Protocole MQTT .....	11
4. Applications IoT .....	12
Conclusion .....	14

Chapitre 2 .....	15
Etude fonctionnelle et étude conceptuelle.....	15
Introduction .....	16
I. Analyse des besoins de notre projet .....	16
1. Analyse des Besoins Fonctionnels.....	16
2. Analyse des Besoins non Fonctionnels.....	16
II. Modélisation des besoins.....	17
1. Langage de modélisation .....	17
2. Identification des acteurs .....	18
3. Diagrammes de cas d'utilisation.....	18
III. Modèle de conception.....	21
IV. Les diagrammes de séquences .....	22
1. Diagramme de séquence de cas d'utilisation « s'authentifier » .....	22
2. Diagramme de séquence de cas d'utilisation « Consulter la liste des devices »....	24
V. Diagramme de classe .....	25
VI. Etude fonctionnelle.....	26
1. Choix des cartes embarquées .....	26
2. Capteurs .....	29
VII. Diagramme de déploiement.....	30
Conclusion .....	30
Chapitre 3 .....	31
Configuration et Implémentation .....	31
Introduction .....	32
I. Environnement de réalisation et de développement.....	32
1. Matériels de base.....	32
2. Choix des langages de développement .....	32
3. Choix de base de données .....	36

II.	Outils de développement .....	39
1.	Pycharm Community .....	39
2.	Visual Paradigm.....	40
	.....	40
3.	IDE Arduino.....	40
4.	Sublime Text.....	41
III.	Réalisation des prototypes .....	41
1.	Implémentation du prototype 1 .....	41
2.	Implémentation du Prototype 2.....	42
3.	Implémentation du Prototype 3.....	42
IV.	Implémentation .....	43
1.	Page de Bienvenue .....	43
2.	Tableau de bord.....	44
3.	Page de consultation des informations collectées en temps réel.....	47
	.....	47
4.	Application FrontOffice .....	47
	Conclusion .....	48
	Conclusion et perspectives .....	49
	Webographie .....	50
	Annexe A : Installation de Mosquitto broker .....	51
	Annexe B : Installation de python, Django, Restframework, Psycopg2 et l'environnement virtuel .....	52
	Annexe C : Exemple du code .....	51

# Liste des figures

Figure 1 Architecture de la plateforme IBM	5
Figure 2 Tableau de bord IBm waston	6
Figure 3 Tableau de bord things board	7
Figure 4 : Modèle IoT	9
Figure 5 Diagramme de d'utilisation générale.	19
Figure 6 Raffinement de cas d'utilisation « Consulter de la liste des devices »	20
Figure 7 Modèle MVC	22
Figure 8 Diagramme de séquence de cas d'utilisation « d'authentifier »	23
Figure 9 Diagramme de séquence consulter la liste des devices	25
Figure 10 : Diagramme de classe	26
Figure 11 : Arduino Uno	26
Figure 12 : Module WIFI ESP8266-1E	27
Figure 13 NodeMCU	28
Figure 14 : Capteur DHT11	29
Figure 15 : Capteur ADC	29
Figure 16 : Diagramme de déploiement	30
Figure 17: Logo mosquitto	33
Figure 18: Logo Python	33
Figure 19 Logo Django	34
Figure 20: SQL vs NoSQL	38
Figure 21: Logo Pycharm	39
Figure 22 Logo 1. Pycharm Community	40
Figure 23: Logo Visual paradigm	40
Figure 24: Logo arduinoIDE	41
Figure 25: Logo Sublime Text	41
Figure 26: Prototype 1	42
Figure 27: Prototype 2	42
Figure 28: Prototype 3	42
Figure 29: Page de Bienvenue	43

Figure 30: Page d'inscription	44
Figure 31: Page d'authentification	44
Figure 32 : Page Dashboard	45
Figure 33 Page liste des cartes disponibles	45
Figure 34 Page supervision du capteur Température	46
Figure 35 Page supervision de l'humidité	46
Figure 36: Monitoring page	47
Figure 37: Front-office	48
Figure 38: Installation de mosquito	51
Figure 39: Mosquitto Client	51
Figure 40: installer python	52
Figure 41: python version	52
Figure 42: installer pip	52
Figure 43: installer django	53
Figure 44: installer package	53
Figure 45: installer l'environnement virtuel	53
Figure 46: Installer RestFramework	54
Figure 47: Installer Psycpg2	54
Figure 48 Exemple du Code	51

# Liste des Tableaux

Tableau 1 : Description détaillée de cas d'utilisation " Consulter la liste des devices "	20
Tableau 2 Description Textuelle de diagramme de séquence de cas d'utilisation « s'authentifier ».....	22
Tableau 3 Description Textuelle de diagramme de séquence de cas d'utilisation « s'authentifier ».....	24
Tableau 4 : Etude comparative entre Arduino Uno et NodeMCU .....	28
Tableau 5: Matériels de base .....	32

# Liste d'abréviations

<b>IoT</b>	<b>I</b> nternet <b>O</b> f <b>T</b> hings
<b>IoE</b>	<b>I</b> nternet <b>O</b> f <b>E</b> verythings
<b>API</b>	<b>A</b> pplication <b>P</b> rogramming <b>I</b> nterface
<b>REST</b>	<b>R</b> epresentational <b>S</b> tate <b>T</b> ransfer
<b>MQTT</b>	<b>M</b> essage <b>Q</b> ueue <b>T</b> ransport <b>T</b> elemtry
<b>M2M</b>	<b>M</b> achine <b>T</b> o <b>M</b> achine



# Remerciements

Avant tout, je remercie dieu de m'avoir créé et me laisser la chance de vivre pour arriver à ce que je suis.

Il m'est agréablement honoré de remercier Monsieur **Ahmed CHAREF**, Président Directeur Général de la start-up **DR Technologies**, d'avoir accepté et m'accueillir dans sa société en qualité de stagiaire, ainsi en tant que mon encadrant de mon projet de fin d'études et partager des conseils et son soutien m'ont permis de développer une compréhension du sujet et d'élargir mes connaissances sur l'Internet des Objets.

Je remercie également mon encadrante interne Mme. **Basma OUESLATI**, pour l'aide et les conseils concernant les missions évoquées dans ce travail, qu'il m'a apporté lors des différents suivis.

Je tiens à remercier l'ensemble du personnel de l'entreprise pour leur accueil sympathique et chaleureux.

Le long de la réalisation de ce projet de fin d'études. Je tiens à remercier les membres du jury et l'honneur qu'ils m'ont offert d'être ceux qui évaluent mon humble travail.

# Dédicaces



Ma belle-mère pour ses prières inébranlables approfondissant ma carrière d'étude. Elle m'a appris à travailler dur pour les choses que j'aspire à réaliser.



Mon père pour ses encouragements incessants ce qui m'a donné la force d'aller à l'avant.



Mes sœurs et mes frères qui ont toujours cru en moi et ont occupé de moi.



Tous ces qui m'ont aidé de près ou de loin à l'élaboration de ce travail trouvent ici l'expression de mes sincères dédicaces

Ghassen Ouertani

# Introduction Générale

L'Internet des objets (IoT) apparaît comme la troisième vague du développement de l'internet. C'est le concept des objets du quotidien, des machines industrielles aux appareils portatifs et il implique l'environnement, les villes, les bâtiments, les véhicules, les vêtements, les appareils portatifs et autres objets qui leur sont associés de plus en plus d'informations pour communiquer et produire nouvelle information. Ainsi, l'Internet des objets est l'avenir de la technologie qui peut rendre nos vies plus efficaces et il va défendre sa position importante dans le contexte des technologies de l'information et de la communication et du développement de la société.

Des nouveaux types d'applications peuvent impliquer comme la maison intelligente, le bâtiment intelligent, l'usine intelligente et la ville intelligente dans lesquelles les appareils et services qui fournissent des notations, la sécurité, l'économie d'énergie, l'automatisation, les télécommunications, les ordinateurs et le divertissement sont intégrés dans un écosystème unique.

L'Internet des Objets n'est pas une technologie unique, c'est un concept dans lequel des nouvelles choses sont connectées comme des capteurs intégrés, la reconnaissance des images, l'intelligence artificielle dans un système de décision des machines intelligentes. Au fur et à mesure que l'IoT continuera de se développer, il s'agira d'une combinaison avec des approches et des concepts technologiques connexes tels que «Cloud Computing», «Internet», «Big Data», «Robotiques» et «Blockchain».

L'Internet des objets fournit des solutions basées sur l'intégration de la technologie de l'information, qui désigne le matériel et les logiciels utilisés pour stocker, extraire et traiter des données avec la technologie de communication machine à machine (M2M) à l'aide des réseaux sans fil (WSN). 2G / 3G / 4G, LoRa, RFID, BLE, etc.

D'ici 2020, plus de 50 milliards d'objets connectés seront déployés et nous entrerons dans la quatrième phase d'Internet, appelée Internet of Ev-erything (IoE). Le (IoE) est la connexion en réseau de personnes, de processus, de données et de choses. L'avantage de l'IoE est dérivé de l'impact composé de la connexion des gens, des processus, des données et des choses, et de la valeur que cette augmentation des objets connectés crée comme "tout" vient en ligne. Le concept IoE inclut, outre les communications M2M, les interactions machine-à-personne (M2P) et les interactions personne-à-personne assistées par technologie (P2P).

De gros volumes de données sont générés, et ces données sont transformées en actions utiles qui peuvent commander et contrôler les choses pour rendre notre vie beaucoup plus facile et plus sûre. La créativité de cette nouvelle ère est illimitée, avec un potentiel incroyable pour améliorer nos vies.

Mon projet de fin d'études s'inscrit dans ce contexte et consiste à développer un système intelligent pour les bâtiments basés sur les technologies IoT. Dans le présent projet, une enquête a été menée pour en apprendre davantage sur les technologies existantes et émergentes créées par les entreprises leaders sur le marché.

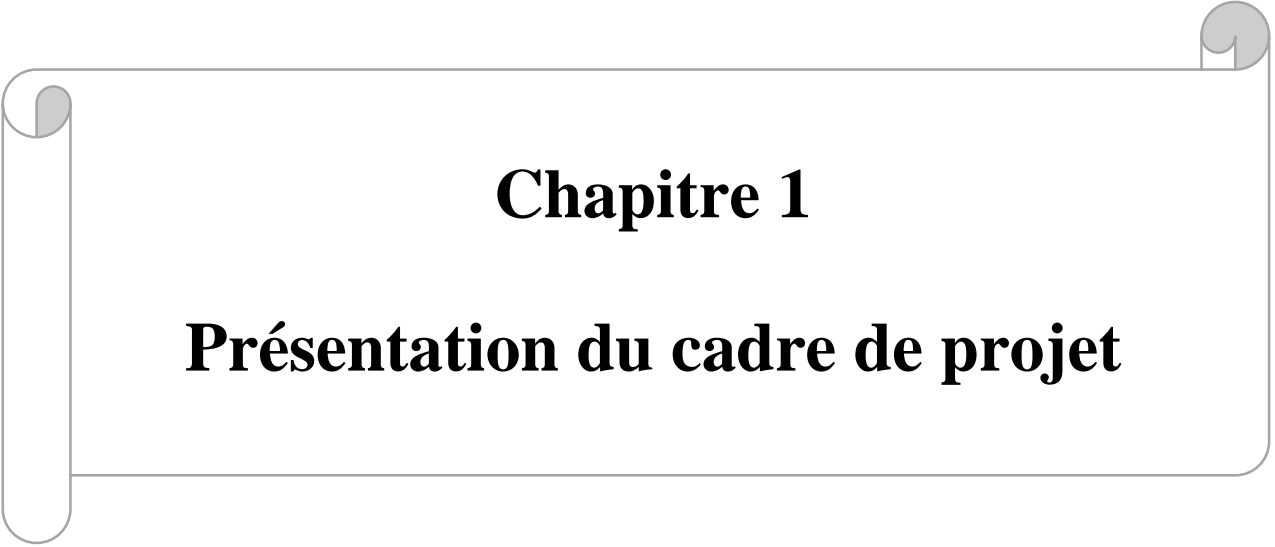
Cela était essentiel pour trouver une marge d'amélioration dans les applications et les technologies actuelles et pour formaliser le sujet du projet qui résout un problème restant à résoudre et proposer une nouvelle solution dans le domaine de la sûreté et de la sécurité des bâtiments.

Ce rapport est composé de trois chapitres comme suit:

Le premier chapitre « présentation du cadre de projet » présente le concept général du projet de fin d'études et l'institution d'accueil.

Le deuxième chapitre est consacré au choix de la méthodologie du travail et définir les différentes composantes du système proposé. Il donne des spécifications sur les exigences fonctionnelles et non-fonctionnelles du prototype.

Le troisième chapitre détaille la conception architecturale du système en présentant le diagramme de séquence des différents cas d'utilisation ainsi que la mise en œuvre et le développement du projet.

A decorative frame resembling a scroll, with a light gray background and a thin gray border. The frame has rounded corners and a vertical strip on the left side. The text is centered within the frame.

# **Chapitre 1**

## **Présentation du cadre de projet**

# Introduction

Afin d'obtenir plus de détails sur ce projet, ce chapitre présente une présentation générale sur la société d'accueil, puis précise le thème du projet et le problème à résoudre, ainsi que les objectifs. À la fin de ce chapitre, un état de l'art présentera le contexte général du projet.

## I. Présentation de l'entreprise

### 1. Description de l'entreprise

#### a. Technoparc ElGazella

Le Technoparc Elgazala est un environnement intégré pour le développement des petites et moyennes entreprises ainsi que pour les multinationales et les grands groupes du secteur des technologies de l'information et de la communication. Son objectif principal est d'accueillir et de soutenir le développement d'activités de haute technologie et de promouvoir la recherche et le développement ainsi que le transfert de technologie.

Sa mission s'inscrit dans la stratégie nationale de développement et de promotion de la recherche scientifique, de l'innovation et de la production à forte valeur ajoutée dans le domaine des technologies de l'information et de la communication. Il accueille des institutions de soutien telles que la formation et la recherche en plus des activités de production, dans le but de créer une synergie et des relations croisées entre ces acteurs.

Elgazala Technoparc accueille aujourd'hui plus de 100 entreprises, dont 12 filiales de grands groupes mondiaux (Microsoft, ST Microélectroniques, Ericsson, Alcatel Lucent), ainsi que des succès stories tunisiennes (Telnnet, Cynapsys, etc).

#### b. Technoparc Innovation Center

Mon projet a été réalisé au centre d'innovation sous le nom de Technoparc ElGazalla. Ce centre d'innovation est basé pour faciliter son intégration avec son environnement et la société. Il englobe les entités suivantes:

- **Un espace de travail collaboratif** où des équipes d'entrepreneurs seront hébergées pour des périodes variant entre 6 et 18 mois pour travailler sur des projets visant à résoudre des problèmes issus de l'environnement du Technoparc. Un accélérateur d'innovation qui contribuera, en étroite collaboration avec le Technoparc, au développement des prototypes développés dans le centre d'innovation.

- **Des laboratoires de tests et de prototypage** où les acteurs de l'écosystème Technoparc peuvent valider les prototypes et les solutions qu'ils développent en son sein.

En plus de ces composantes, le centre d'innovation s'appuiera sur un ensemble de partenaires qui vont notamment:

- Fournir le centre d'innovation avec les différentes composantes d'une plateforme de prototypage.
- Supervision technologique des équipes hébergées au centre d'innovation sur des sujets spécifiques et spécialisés.

L'avantage de ces composants est de créer un environnement où les équipes du projet (entrepreneurs, chercheurs, doctorants) peuvent travailler sur la résolution des problèmes concrets tout en préservant les intérêts de chacune des parties prenantes. Afin d'atteindre cet objectif, un modèle économique fondé sur le partage des intérêts et des avantages communs devrait être mis en œuvre. C'est ainsi que j'ai bénéficié de cet environnement qui prolonge la réalisation des projets.

## **2. Les Services de l'entreprise :**

Les services offerts par l'entreprise sont :

- Développement des solutions IoT
- Développement des applications web
- Développement des applications mobile
- Développement de Chat Bot

## **II. Cahier de charges**

### **1. Problématique**

La demande de sécurité est devenue de plus en plus parmi les occupations les plus intéressantes dans le monde, ce besoin se concrétise essentiellement en quittant notre lieu d'habitation pour aller au travail ou partir en vacance ou pour faire une autre occupation et aussi en quittant notre lieu de travail, les gens cherchent plus de sécurité, notre apport par rapport à cette problématique

est mettre en place une plateforme permettant d'avoir en temps réel tous les indicateurs nécessaires qui décrivent l'atmosphère de lieu visé à savoir par exemple le niveau du gaz dégagé, l'électricité, l'eau s'il y a un débordement et donc de pouvoir réagir afin d'éviter une telle catastrophe qui peut être déclencher. Les utilisateurs de cette plateforme vont avoir une information exacte en temps opportun et par conséquence avoir un niveau « level » de sécurité satisfaisant.

Cette plateforme va donner plus de sécurité non seulement pour les locaux à usage d'habitation mais aussi pour les usines, les magasins, les industries, etc.

Il est noté que l'IoT, malgré qu'elle est primordiale pour la sécurité des nations mais elle n'est pas encore concrétisée en Tunisie, ce projet d'étude est un premier pas pour sa réalisation et il est la première idée de recherche.

## **2. Etude de l'existant**

L'entreprise DR Technologies est une nouvelle start-up. Son domaine d'activité est le développement des sites web. Suite à l'émergence du domaine de l'IoT, ils ont pensé à développer une plateforme IoT pour ses clients.

Actuellement, ils n'ont pas une solution qui fait le même travail.

## **3. Solutions existantes**

A l'échelle internationale, il existe quelques solutions des plateformes IoT (IBM Bluemix, etc...) qui permettent d'avoir une information en temps réel à travers des capteurs installés dans une plaque électronique (Arduino uno, ESP8266, etc...)

### **a. IBM Watson IOT (closed source)**

La plate-forme IBM Watson <sup>TM</sup> IoT pour IBM Cloud(Bluemix) offre une boîte à outils polyvalente comprenant des périphériques de passerelle, une gestion des périphériques et un accès puissant aux applications. En utilisant la plate-forme Watson IoT, nous pouvons collecter des données de périphérique connectées et effectuer des analyses sur les données en temps réel de l'organisation.



### – Architecture de la plateforme Watson IoT

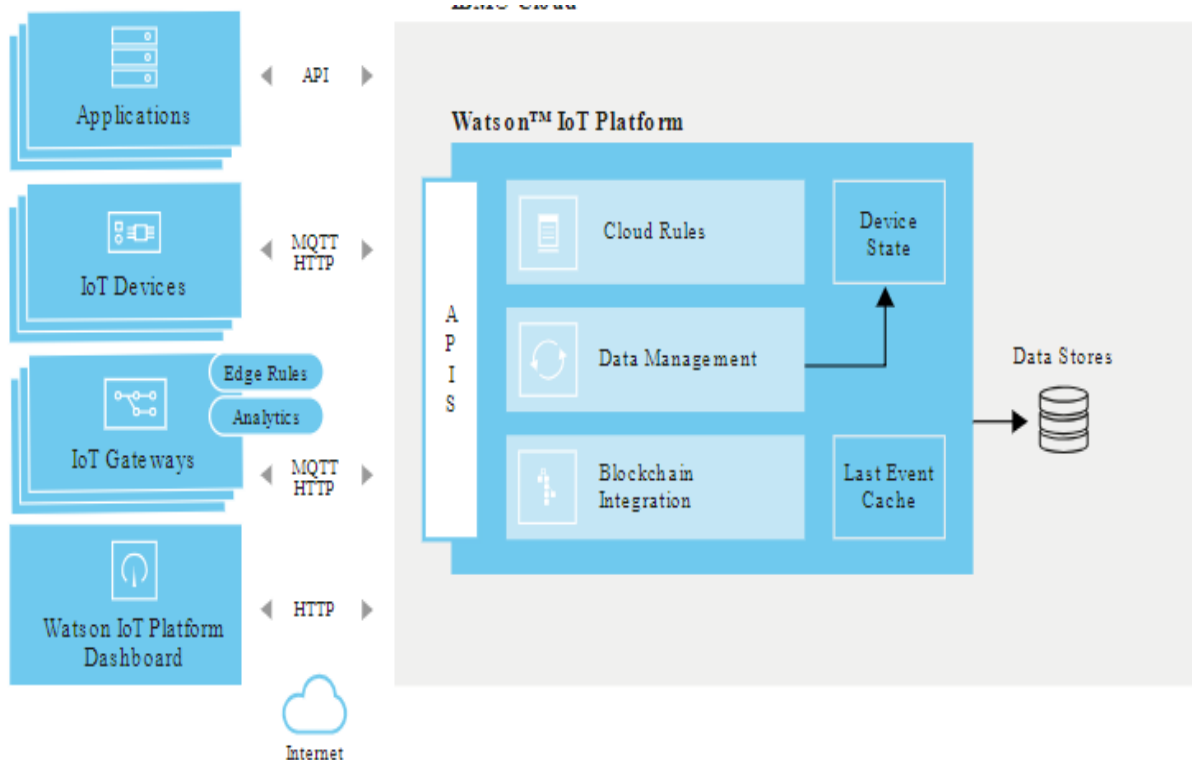


Figure 1 Architecture de la plateforme IBM

### – Les services offertes par Watson

- Connexion des appareils et développement des applications.
- Stockage, normalisation, transformation et examen des données de l'appareil et intégration de la plate-forme Watson IoT à d'autres services.
- Spécification des conditions de règles basées sur les données de périphérie en temps réel pour déclencher des alertes et des actions.
- Configuration de la connectivité et une architecture sécurisée avec un contrôle d'accès pour les utilisateurs et les applications.

## – Plateforme IoT de l'IBM

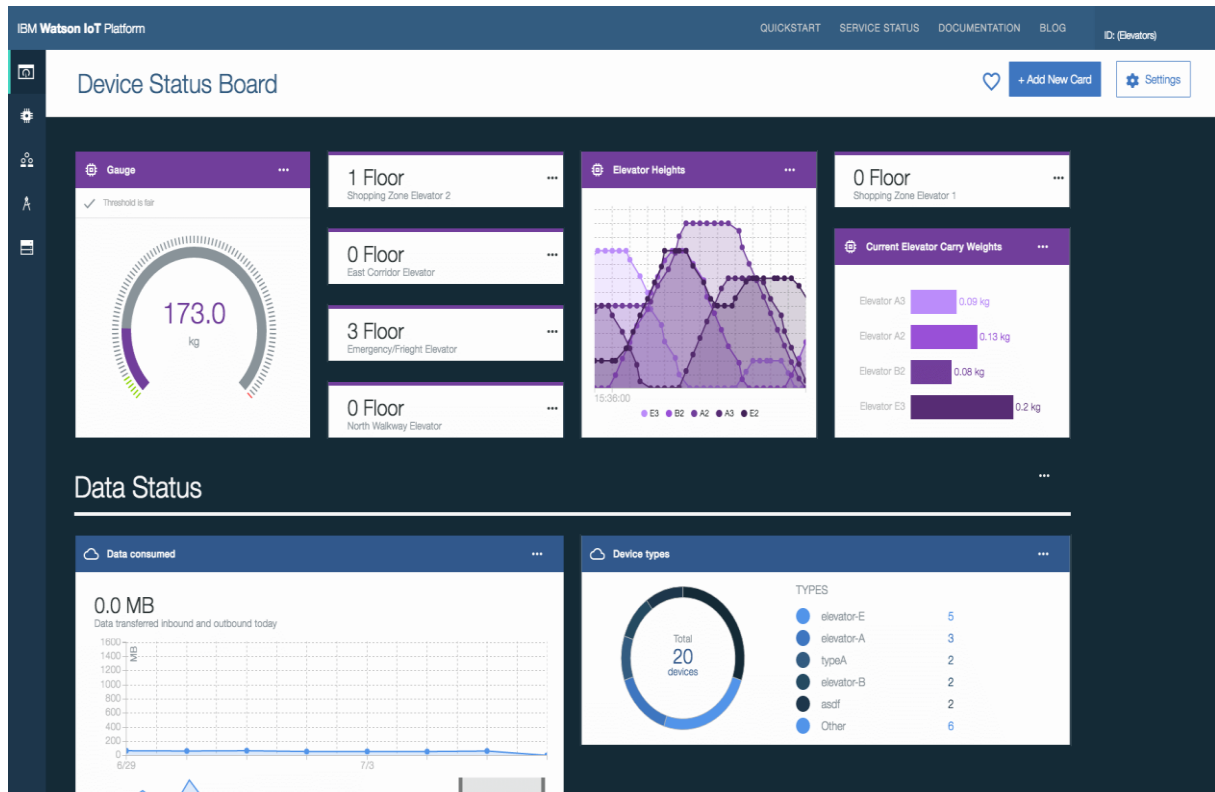


Figure 2 Tableau de bord IBM watson

## b. ThingsBoard : (opensource)

ThingsBoard est une plate-forme IoT open-source qui permet un développement, une gestion et une mise à l'échelle rapides des projets IoT.

### – Les services offerts par ThingsBoard :

- Dispositifs de provision et de contrôle.
- Collecte et visualisation des données de l'appareil.
- Analyse des données de l'appareil et déclenchement des alarmes avec un traitement d'événement complexe.
- Livraison des données de l'appareil à d'autres systèmes.
- Construction des workflows basés sur un événement de cycle de vie de périphérique, un événement d'API REST, une requête RPC, etc.
- Activation des fonctionnalités spécifiques au cas d'utilisation à l'aide de chaînes de règles personnalisables.

## – Plateforme ThingsBoard

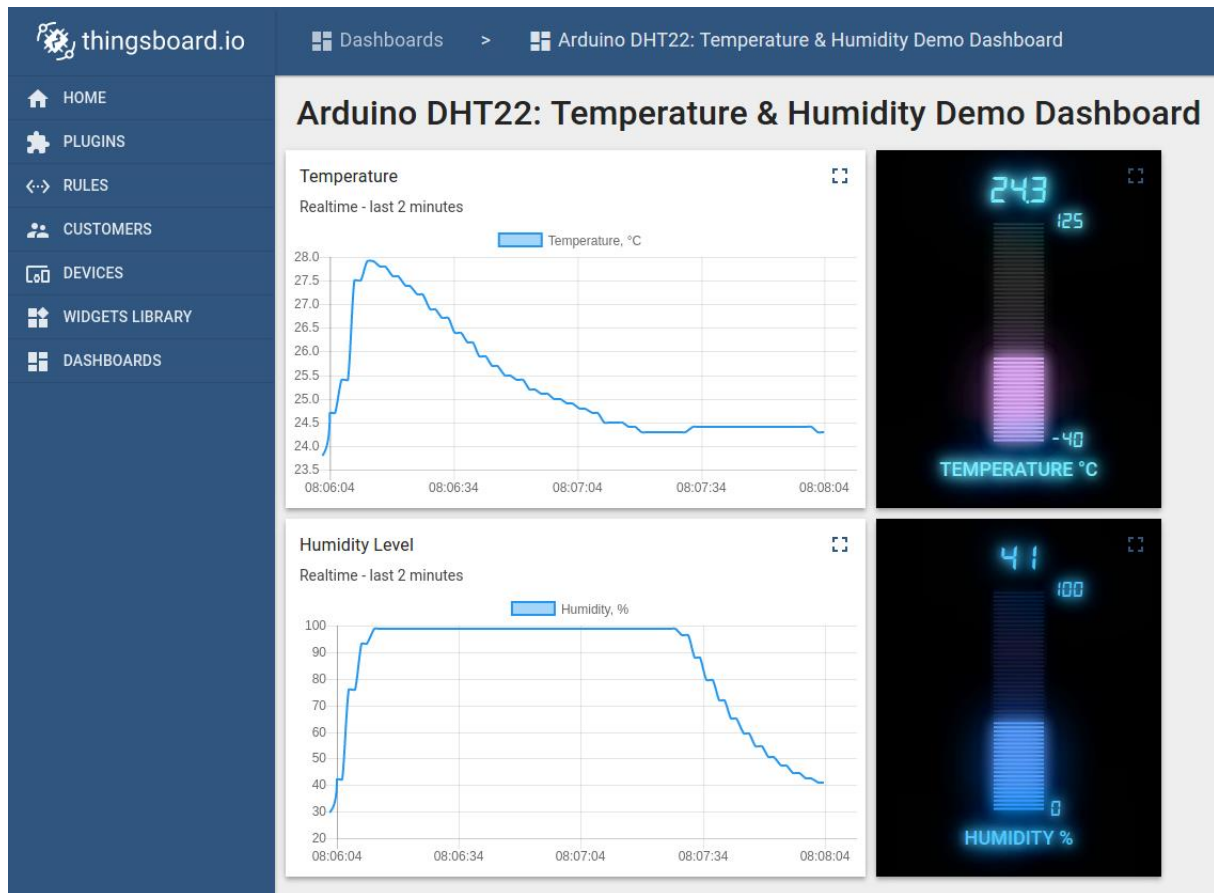


Figure 3 Tableau de bord things board

## 4. Solution proposée

La solution est de développer une plateforme IoT qui permet la supervision à distance le lieu visé des clients (la maison, l'usine, etc.) et d'avoir une information globale sur les capteurs installés en temps réel et de consulter l'historique des valeurs prises à travers ces capteurs sous forme d'une courbe.

## 5. Cible du projet

Mon travail se concentre principalement sur l'aspect de supervision de la domotique. En fait, les besoins de sécurité des personnes ont évolué. D'où l'utilité du développement des systèmes avancés de supervision à domicile utilisant un système de réseau de capteurs sans fil et une intelligence artificielle.

Ce projet de fin d'études vise à concevoir et mettre en œuvre un système automatisé et en temps réel qui être mis à jour sur la base des données des capteurs ou manuellement par le contrôle de l'utilisateur. Ce système est conçu pour être extensible, permettant de contrôler divers appareils aussi de protéger les bâtiments publics, les installations industrielles, les maisons privées contre le vandalisme, le vol, l'intrusion ou, le cas échéant, l'évasion illégale.

Le système proposé a une grande flexibilité en utilisant la technologie WI-FI pour interconnecter des capteurs distribués au serveur du bâtiment. Cela réduira les coûts de déploiement et augmentera la capacité de mise à niveau et de la configuration du système. En termes de réalisation d'une tâche prédéterminée, en fournissant un retour aux utilisateurs, et de répondre en conséquence aux situations. En d'autres termes, il permet simplement d'automatiser et de contrôler de nombreux aspects du système domestique tels que le contrôle de l'éclairage, le réseau et les communications, le système de divertissement, les systèmes d'intervention d'urgence et de surveillance de sécurité.

Dans le cas de l'activation tels que des radios, des systèmes stéréo ou des téléviseurs ou des signaux lumineux peuvent être transmis par le dispositif de lecture, spécifiquement pour simuler ou feindre la présence d'une personne dans un tel endroit comme la pièce, cuisine, etc., dans un appartement, maison, bureau, est occupé. Une simulation de ce type est utilisée comme moyen de dissuasion et vise à minimiser le risque que des personnes non autorisées ayant une intention criminelle aient accès à une telle pièce ou à de tels locaux.

### **III. Etat de l'art**

#### **1. Internet des objets**

Internet a joué un rôle fondamental dans notre vie en permettant aux gens de se connecter à travers les continents et nous a rapprochés en tant que race. La première vague Internet a permis de connecter 1 milliard d'utilisateurs via des ordinateurs personnels dans les années 1990. Cela a été suivi par la vague de l'Internet mobile qui a entraîné plus de 2 milliards d'utilisateurs connectés supplémentaires depuis le début de l'année 2000. La prochaine vague, qui a déjà commencé, connectera jusqu'à 50 milliards de "choses" à l'internet d'ici à 2020. 40 milliards qui sont déjà connectés aujourd'hui. L'Internet des objets (IoT) est le réseau d'objets physiques: appareils, véhicules, bâtiments et autres éléments embarqués d'électronique, de logiciels, de capteurs et de connectivité réseau permettant à ces objets de collecter et d'échanger des données.

Aujourd'hui, nous vivons au 21<sup>ème</sup> siècle où l'automatisation joue un rôle important dans la vie humaine. Les maisons intelligentes d'aujourd'hui se composent de nombreux appareils tels que plusieurs caméras, microphones, différents capteurs, actionneurs, contrôleurs de périphériques et bases de données domestiques, où les composants communiquent via un réseau local, ce qui indique l'automatisation des tâches quotidiennes pouvant être contrôlées. Éclairage, porte, ventilateurs, laveuses, sècheuses, réfrigérateurs, etc. Il fournit également la sécurité à la maison et le système d'urgence à activer comme la visualisation des intérieurs de la maison à des fins de surveillance, la détection des ouvertures des fenêtres et des portes. Fuite de gaz qui peut être accessible à distance pour la commodité de l'utilisateur. Également des équipements économes en énergie tels que les thermostats intelligents, les compteurs intelligents, la surveillance de la consommation d'eau et les soins de santé pour la surveillance à distance, les diagnostics et les services.

## 2. Couches du modèle IoT

Les réseaux traditionnels de communication de données ont de multiples fonctions, comme en témoigne le modèle de référence à 7 couches de l'Organisation internationale de normalisation (ISO). Cependant, un système IoT complet contient 7 niveaux en plus du réseau de communication (figure 1).



Figure 4 : Modèle IoT

**Niveau 1 : Périphériques physiques et contrôleurs**

Les périphériques physiques et les contrôleurs susceptibles de contrôler plusieurs périphériques. Ils comprennent une large gamme de périphériques de point de terminaison qui envoient et reçoivent.

**Niveau 2 : Connectivité**

La fonction la plus importante du niveau 2 est la transmission fiable et opportune de l'information.

Cela inclut les transmissions:

- Entre les appareils (niveau 1) et le réseau
- Vers les réseaux (est-ouest)
- Entre le réseau (niveau 2) et le traitement de l'information du bas niveau (niveau 3)

**Niveau 3 : Calcul Edge (Fog)**

Les fonctions du niveau 3 sont dictées par la nécessité de convertir les flux de données du réseau en informations qui conviennent au stockage et au traitement du niveau supérieur. Cela signifie que les activités du niveau 3 se concentrent sur l'analyse et la transformation de données à haut volume.

**Niveau 4 : Accumulation de données**

Les données circulent dans le réseau haut débit et à l'organisation déterminé par les dispositifs générant les données. La plupart des applications n'ont pas besoin de traiter les données sur le réseau. Les applications supposent généralement que les données sont en mémoire ou sur le disque. Ainsi, au niveau 4, Accumulation de données, les données en mouvement sont converties en données stockées physiquement.

**Niveau 5 : Abstraction de données**

Les fonctions d'abstraction des données de niveau 5 sont axées sur le rendu des données et leur stockage de manière à permettre un développement plus simple afin de combiner les données provenant de sources multiples et de simplifier l'application.

**Niveau 6 : Le niveau d'application**

L'interprétation d'informations se produit dans le niveau applicatif. Les logiciels de ce niveau interagissent avec le niveau 5 et les données au repos.

### **Niveau 7 : Collaboration et processus**

À ce niveau, les applications donnent aux professionnels les bonnes données, au bon moment, afin qu'ils puissent le travail demandé.

## **3. Protocole MQTT**

MQ Telemetry Transport (MQTT) est un protocole simple et léger utilisant le protocole de messagerie publish-subscribe. MQTT est conçu pour les réseaux à latence élevée, à faible bande passante ou peu fiables. MQTT est parfait pour les périphériques à ressources limitées, car il économise l'utilisation de la mémoire et la puissance. Les appareils connectés s'abonnent à un sujet hébergé sur un courtier MQTT. Chaque fois qu'un autre appareil / service publie des données sur un sujet, tous les appareils qui y sont abonnés reçoivent automatiquement les informations mises à jour.

Il existe plusieurs définitions clés: client, courtier et sujet:

**Le client MQTT** est un périphérique d'un microcontrôleur jusqu'à un serveur à part entière, qui possède une bibliothèque MQTT en cours de fonctionnement et qui se connecte à un courtier MQTT sur n'importe quel type de réseau.

**Le courtier** est principalement responsable de recevoir tous les messages, de les filtrer, de décider qui s'y intéresse et ensuite d'envoyer le message à tous les clients abonnés.

**Les messages** sont publiés dans une adresse appelée rubrique. Les clients s'abonnent aux topics et « écoutent » qu'ils reçoivent tous les messages publiés sur un sujet. Les sujets dans MQTT ont une structure hiérarchique. Mais, il est possible d'utiliser des caractères génériques "+" et "". "+" fait référence à un seul répertoire alors que "" correspond à un nombre quelconque d'entre eux.

Il y a un certain nombre de courtiers MQTT disponibles pour différentes machines. Pour ce projet, nous avons choisis Mosquitto comme courtier car c'est l'un des courtiers MQTT les plus populaires et les plus stables. Mosquitto est un service de courtier de messages open-source qui utilise le protocole MQTT pour envoyer et recevoir des messages, ce qui le rend approprié pour la messagerie « Internet des objets » comme les capteurs de faible puissance ou les microcontrôleurs comme l'Arduino.

Les bibliothèques clientes MQTT sont disponibles sur une grande variété de plateformes de langages de programmation comme C, C ++, C, Java, JavaScript, Android, iOS, etc.

MQTT définit des méthodes pour indiquer les actions souhaitées à effectuer sur les ressources identifiées. Les ressources peuvent être des périphériques, des fichiers ou les sorties d'un exécutable, trouvés sur un serveur.

Les méthodes dans le protocole MQTT sont:

- **Connecté:** Attend que la connexion soit établie avec le serveur.
- **Déconnecté:** Attend que le client MQTT termine tout travail qui doit être fait et que la session TCP / IP se déconnecte.
- **Abonnement:** demande au serveur de laisser le client s'abonner à un ou plusieurs sujets.
- **Désinscription:** demande au serveur de laisser le client se désabonner d'un ou de plusieurs sujets.
- **Publié:** Retourne immédiatement au thread d'application après avoir transmis la requête au client MQTT.

## 4. Applications IoT

Il est impossible d'envisager toutes les applications potentielles de l'IoT en ayant à l'esprit le développement de la technologie et les besoins divers des utilisateurs potentiels. Dans cette section, nous présentons quelques applications importantes. En fait, suite à l'émergence de l'IoT, plusieurs domaines d'activités se sont impliqués dans ce concept pour créer nombreux champs d'applications.

- **Militaire** : Le concept militaire des cycles de décision place le flux d'information au cœur de toutes les activités, de la logistique à l'intelligence. Dans le domaine militaire, les capteurs sont déployés dans des infrastructures critiques telles que les champs de bataille. Ils construisent un réseau pour des fins de surveillance dans les délais qui aideront les chefs militaires dans la prise de décision.
- **Transport** : L'IoT peut faciliter l'intégration des communications, du contrôle et du traitement de l'information dans divers systèmes de transport. L'application de l'IoT s'étend à tous les aspects des systèmes de transport (à savoir le véhicule, l'infrastructure

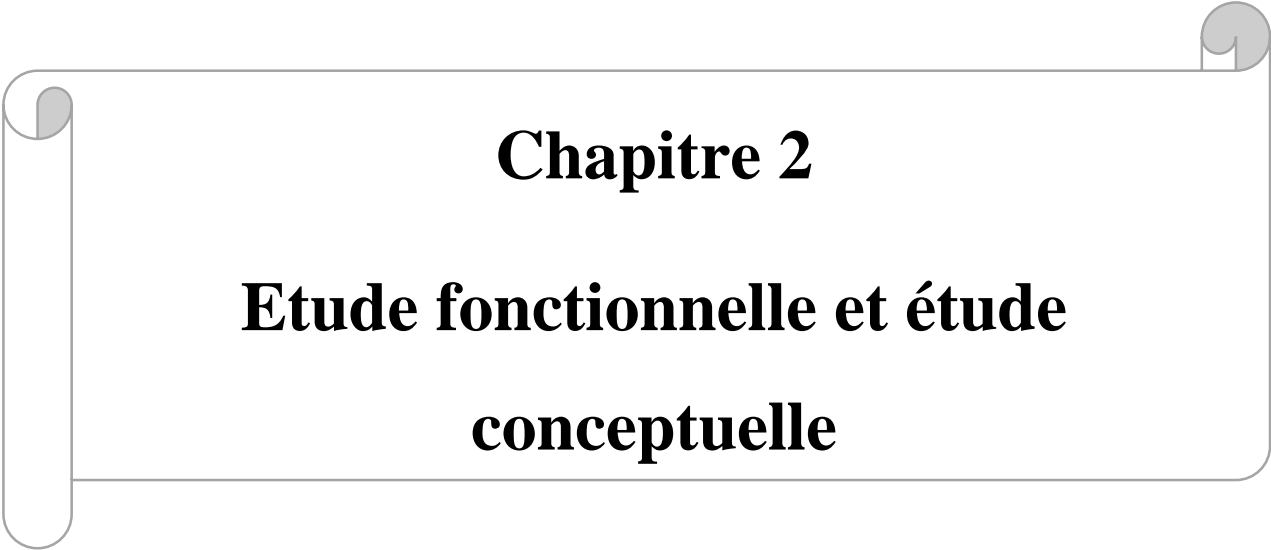


et le conducteur ou l'utilisateur). L'interaction dynamique entre ces composants d'un système de transport permet de nombreuses applications comme la communication véhiculaire, le contrôle intelligent du trafic, le stationnement intelligent, la gestion logistique, le contrôle des véhicules, la sécurité et l'assistance routière.

- **Soins de santé** : Les capteurs peuvent être déployés à un niveau de personne appelé Body Area Network (BAN). Ces capteurs pourraient également être appelés "portables". Ils sont très autonomes avec leur faible consommation d'énergie. Ils peuvent être utilisés pour assurer un suivi continu de l'activité des patients et mesurer leurs paramètres vitaux. Les informations mesurées, appelées aussi informations détectées, sont ensuite envoyées au patient médecin qui choisit la réaction adaptée au patient.
- **Smart Energie et Smart Grid** : Le public est de plus en plus conscient du changement de paradigme de la politique en matière d'approvisionnement, de consommation et d'infrastructures énergétiques. Pour plusieurs raisons, l'approvisionnement énergétique futur ne devrait plus être basé sur des ressources fossiles. En conséquence, l'approvisionnement énergétique futur doit reposer en grande partie sur diverses ressources renouvelables. Smart Energie fournit à ses utilisateurs un environnement viable, abordable, respectueux du climat et engageant qui répond aux besoins et aux intérêts de ses utilisateurs et qui repose sur une économie durable.
- **Maisons intelligentes** : Les applications résidentielles intelligentes ont pour but de contrôler les ampoules intelligentes, d'automatiser et de gérer l'éclairage, le chauffage, la climatisation, le divertissement, les portes et les fenêtres et les systèmes de sécurité d'une maison. En fait, un scénario de maison intelligente est celui où les appareils connectés à la maison sont interconnectés et partagent des informations les uns avec les autres. En utilisant les capteurs appropriés, comme les capteurs de cartes météo, les applications de maison intelligente visent également à surveiller les systèmes, les environnements et à envoyer des alertes en cas de problème. Il peut permettre à l'utilisateur de créer un groupe d'actions pour certaines situations particulières, telles que le réglage du système de sécurité d'une maison ou le réglage du chauffage et de l'éclairage. Il peut également utiliser des actionneurs et des dispositifs sensibles pour gérer un scénario de maison intelligente.

## Conclusion

Dans ce chapitre, nous avons présenté l'organisation d'accueil du projet de graduation. Cette étude a identifié la pertinence du problème et les approches pour le résoudre. Dans le prochain chapitre, nous allons présenter la méthodologie du travail adoptée pour les tâches requises. Ensuite, nous préciserons l'étude conceptuelle du projet. Enfin, nous présenterons l'étude fonctionnelle.

A decorative frame resembling a scroll, with a vertical strip on the left and a horizontal strip at the top, both featuring rounded ends and a slight shadow effect.

# **Chapitre 2**

## **Etude fonctionnelle et étude conceptuelle**

# Introduction

Dans ce deuxième chapitre, nous présenterons les objectifs de notre projet tout en identifiant les besoins fonctionnels et non fonctionnels aussi que les acteurs de notre système.

La modélisation de ces besoins est faite en utilisant le langage de modélisation UML. Ce qui nous amène à identifier les possibilités du système et les besoins des utilisateurs que nous les projetons dans des diagrammes de cas d'utilisation. Dans notre étude conceptuelle, nous présentons quelques diagrammes de séquences, le diagramme de classe et le diagramme de déploiement pour avoir une idée globale sur notre architecture.

Dans l'étude fonctionnelle, nous allons présenter une étude comparative pour dégager nos besoins fonctionnels en termes de solutions à adopter.

## I. Analyse des besoins de notre projet

Afin de bien analyser nos besoins, nous dégagerons les acteurs primordiaux, les besoins fonctionnels et les besoins non fonctionnels.

### 1. Analyse des Besoins Fonctionnels

Les exigences fonctionnelles sont les caractéristiques du système par lesquelles le système remplit la responsabilité fondamentale que nous nous attendons à accomplir. Un grand nombre de parties ou de parties prenantes peuvent être impliquées dans la fourniture de systèmes et de services de sécurité à domicile.

L'administrateur peut gérer les utilisateurs et les périphériques, visualiser les utilisateurs et les périphériques dans les tableaux.

La plate-forme permet à l'utilisateur de s'authentifier, d'obtenir le statut du système, de visualiser les périphériques disponibles, de visualiser en temps réel les valeurs des capteurs.

Un service de notification envoie également une notification à l'utilisateur.

### 2. Analyse des Besoins non Fonctionnels

Un besoin fonctionnel n'est pas une restriction ou une contrainte sur un service système, comme les contraintes environnementales et la mise en œuvre et les exigences de performance, de maintenabilité de plate-forme, d'extensibilité et de fiabilité.

Les exigences non fonctionnelles de l'application sont:

- **Sécurité:** Assurer la sécurité des informations personnelles des abonnés et de toutes les opérations administratives.
- **Performance:** Réduire le temps de réponse des pages de chargement et afficher les résultats et l'heure à actualiser.
- **Simplicité:** L'interface de la plate-forme doit être simple «User-friendly», conviviale et pratique afin que l'utilisateur puisse l'exploiter sans se référer à des connaissances spécifiques. En d'autres termes, l'information doit être lisible et facilement accessible par tout utilisateur.
- **Robustesse:** L'application doit pouvoir gérer un très grand volume de données sans se bloquer.
- **Disponibilité:** L'application doit être disponible à tout instant pour être utilisée par n'importe quel utilisateur et doit être facilement accessible via n'importe quel poste de travail.
- **Rapidité:** L'application doit fournir des résultats efficaces et cohérentes et dans un temps négligeable suite aux requêtes issues des utilisateurs.
- **Fiabilité:** Les résultats apportés par l'application doivent être fiables et reflètent effectivement l'état de la base au moment de son interrogation.

## II. Modélisation des besoins

Nous présentons dans cette section, les acteurs système, le diagramme de cas d'utilisation générale, le raffinement de quelques cas d'utilisation, quelques diagrammes de séquence et le diagramme de classe.

### 1. Langage de modélisation

Le Langage de Modélisation Unifié, en anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.

## 2. Identification des acteurs

Un acteur dans un élément externe qui peut être un homme, une machine ou un autre système qui interagit avec le système pour obtenir un résultat. Il/Elle prend des décisions et des initiatives. Dans le cas de notre application, nous distinguons deux acteurs principaux et dispositifs comme acteur secondaire:

### Acteurs principaux :

- **Administrateur:** Créer, lire, mettre à jour et supprimer les utilisateurs et appareils enregistrés.
- **Client:** Visualiser la liste des appareils, visualiser les valeurs des capteurs en temps réel, visualiser l'historique des valeurs.

### Acteurs secondaires :

- **Dispositif d'acteur secondaire:** l'appareil peut être un actionneur ou un capteur pour envoyer et recevoir des données de l'utilisateur.

## 3. Diagrammes de cas d'utilisation

La modélisation de cas d'utilisation est une technique de modélisation puissante qui peut être utilisée à différentes étapes du développement de systèmes et de logiciels. Sa compréhension par des personnes techniques et non techniques ainsi que sa simplicité, sa grande flexibilité et sa lisibilité a un grand rôle pour capturer et organiser les besoins. Il peut également être utilisé dans les processus d'analyse, de conception et de test. La modélisation de cas d'utilisation est également prise en charge par UML (**Unified Modeling Language**).

### a. Diagramme de Cas d'utilisation Générale

UML permet de créer plusieurs modèles d'un système: certains montrent le système du point de vue des utilisateurs, d'autres montrent sa structure interne, d'autres donnent une vue globale ou détaillée. Les modèles se complètent et peuvent être assemblés. La figure 5 illustre le diagramme de cas d'utilisation générale.

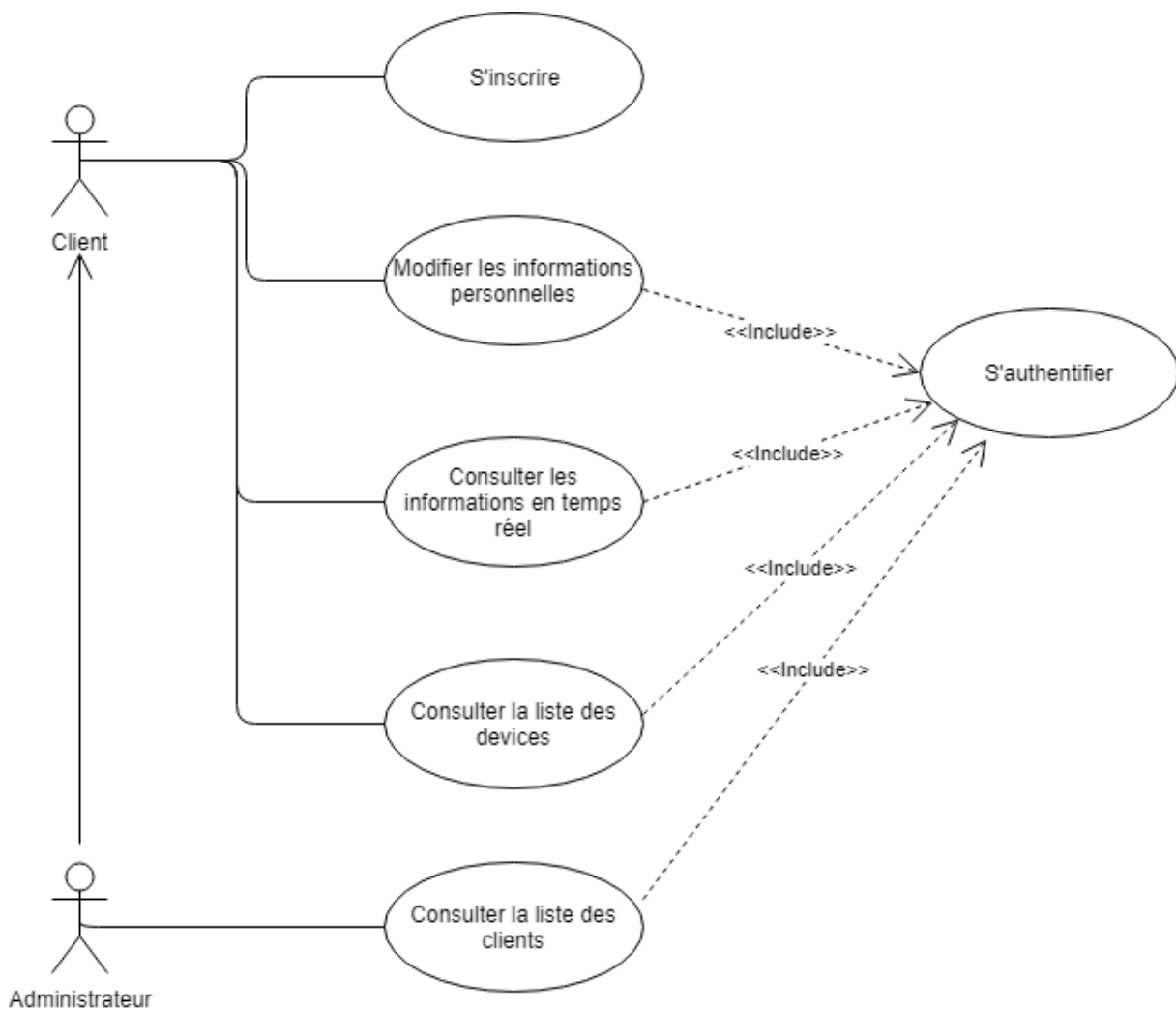


Figure 5 Diagramme de d'utilisation générale.

Le client peut s'inscrire, modifier ses informations personnelles, consulter les informations collectées des capteurs en temps réel et consulter la liste des devices. L'administrateur hérite les fonctionnalités du client et en plus peut consulter la liste des clients

### **b. Raffinement de diagramme de cas d'utilisation « Consulter la liste des devices »**

Le client peut définir différentes commandes sur la plate-forme et vérifier l'état du système et des périphériques. La figure 6 présente le raffinement de diagramme de cas d'utilisation « consulter la liste des devices ».

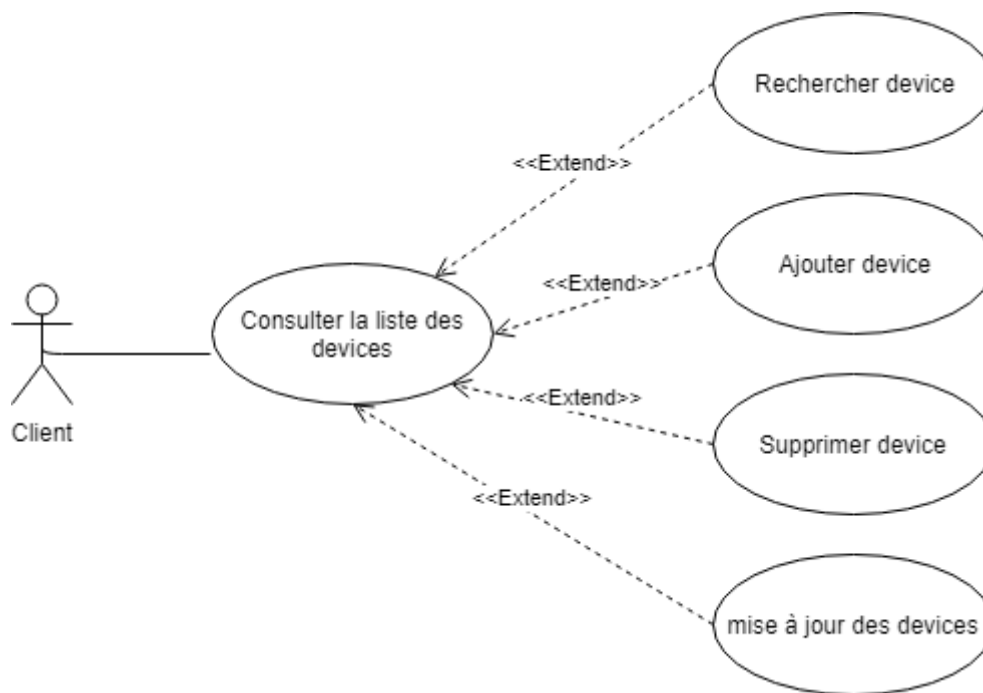


Figure 6 Raffinement de cas d'utilisation « Consulter de la liste des devices »

Le tableau 1 présente une description détaillée de cas d'utilisation « consulter la liste des devices »

Tableau 1 : Description détaillée de cas d'utilisation " Consulter la liste des devices "

CU: Consulter la liste des devices	
Résumé :	Ce cas d'utilisation permet à l'administrateur de gérer les appareils.
Acteur :	Client
Précondition :	Authentification
Post condition:	Modification effectuée
Scénario nominal :	<p>- Client : Après avoir consulter la liste des appareils, le client peut :</p> <ol style="list-style-type: none"> <li>1- Ajouter des données des appareils</li> <li>2- Mise à jour des données des appareils</li> <li>3- Recherche un appareil</li> <li>4- Supprimer un appareil</li> </ol> <p>-Système : 1- Effectuer les modifications</p>
Cas d'erreur :	<ol style="list-style-type: none"> <li>1- Problème de connexion</li> <li>2- Tentative d'authentification échouée</li> </ol>



### III. Modèle de conception

L'application est généralement représentée par trois niveaux, la présentation, l'application (logiques métiers) et la persistance. Les modèles de conception courants pour chacun de ceux-ci sont :

- **Présentation** : Modèle-Vue-Contrôleur (MVC)
- **Persistance** : modèle Entité Accès Objet (EAO), une version du modèle de Data Access Object, mise à jour pour JPA
- **Application** : Session Façade

Le but de MVC est justement de séparer la logique du code en trois parties que nous la retrouvons dans des fichiers distincts, comme l'explique la description qui suit :

- **Modèle** : cette partie gère les données du site. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. Nous y trouvons donc les requêtes SQL
- **Vue** : cette partie se concentre sur l'affichage. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. Nous y trouvons essentiellement du code HTML mais aussi quelques boucles et conditions PHP très simples, pour afficher par exemple la liste des messages des forums
- **Contrôleur** : cette partie gère la logique du code qui prend des décisions. C'est en quelque sorte l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue. Le contrôleur contient exclusivement du PHP. C'est notamment lui qui détermine si le visiteur a le droit de voir la page ou non (gestion des droits d'accès).

La figure 7 présente le mode de fonctionnement du MVC.

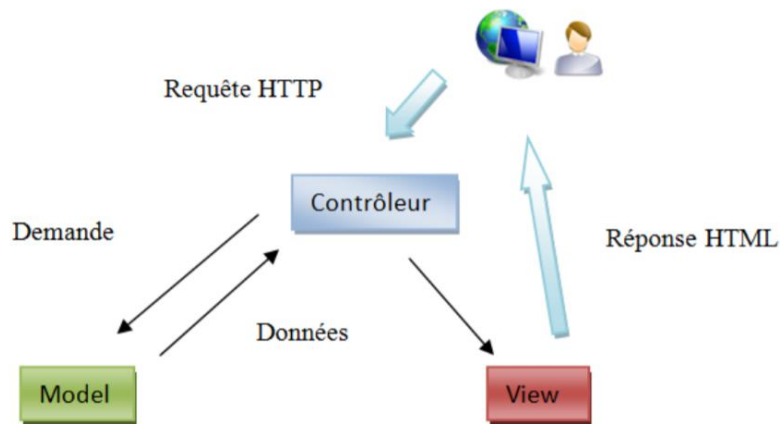


Figure 7 Modèle MVC

## IV. Les diagrammes de séquences

Les diagrammes de séquence représentent les interactions entre les objets en indiquant la chronologie des séquences. Les diagrammes de séquences ajoutent une dimension temporelle aux diagrammes de collaborations.

### 1. Diagramme de séquence de cas d'utilisation « s'authentifier »

#### a. Description textuelle

Tableau 2 Description Textuelle de diagramme de séquence de cas d'utilisation « s'authentifier »

<b>Acteur :</b> Administrateur, Client
<b>Résumé :</b> S'authentifier
<b>Pré-condition :</b> L'acteur doit avoir un email et un mot de passe
<b>Description du scénario nominal :</b> <ol style="list-style-type: none"> <li>1. Le client saisit son username et son mot de passe dans l'interface signin</li> <li>2. Le contrôleur vérifie que tous les champs sont remplis</li> <li>3. Le contrôleur envoie ces données à la table client pour les valider</li> <li>4. Un message de validation est envoyé au contrôleur</li> <li>5. Le contrôleur redirige l'utilisateur vers la page d'accueil</li> </ol>
<b>Description du scénario alternatif :</b> <ol style="list-style-type: none"> <li>1. Le contrôleur retourne un message d'erreur à l'interface pour informer le client de remplir tous les champs</li> </ol>

2. Le contrôleur retourne un message d'erreur à l'interface pour informer le client que son email ou mot de passe sont incorrects

**Post condition :** L'acteur est authentifié

## b. Diagramme de séquence

La figure 8 présente le scénario de diagramme de cas d'utilisation « s'authentifier ».

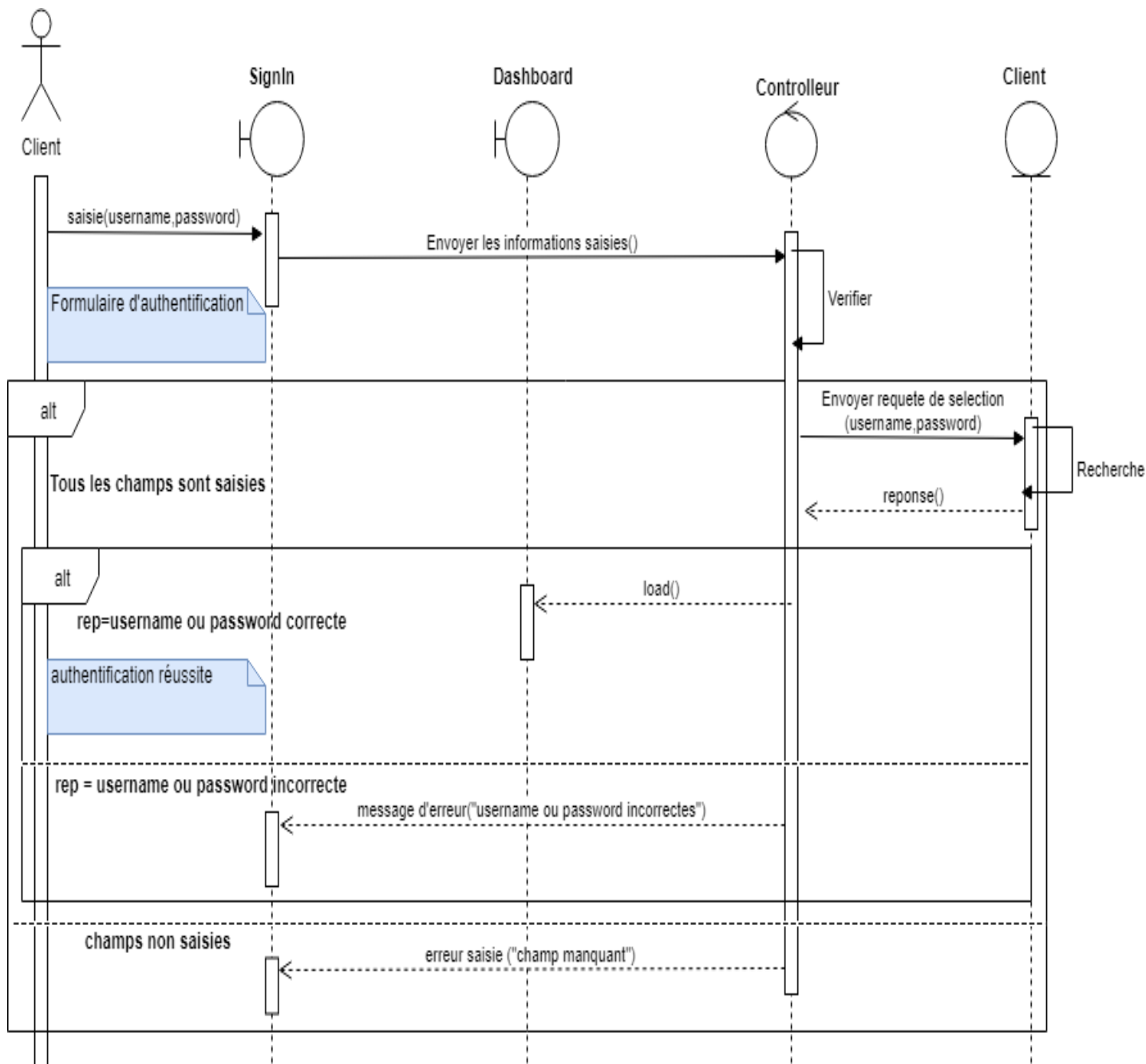


Figure 8 Diagramme de séquence de cas d'utilisation « d'authentifier »

## 2. Diagramme de séquence de cas d'utilisation

### « Consulter la liste des devices »

#### a. Description textuelle

L'utilisateur peut consulter les appareils enregistrés à partir de notre plateforme.

*Tableau 3 Description Textuelle de diagramme de séquence de cas d'utilisation « s'authentifier »*

<b>Acteur</b> : Administrateur, Client
<b>Résumé</b> : Consulter la liste des devices
<b>Pré-condition</b> : L'acteur authentifié
<b>Description du scénario nominal :</b> 1. Le client demande l'affichage de la liste des devices 2. Le contrôleur envoie à la base de données la requête de tous les devices 3. Le résultat de la requête est envoyé au contrôleur 4. Le contrôleur charge la page sensor 5. Le client demande la consultation des informations de l'un des devices retournés 6. La page sensor envoie la requête à la base de données pour la récupération des données
<b>Post condition</b> : Les informations du device seront affichées

#### b. Diagramme de séquence

La figure 9 présente le scénario de cas d'utilisation « consulter la liste des devices ».

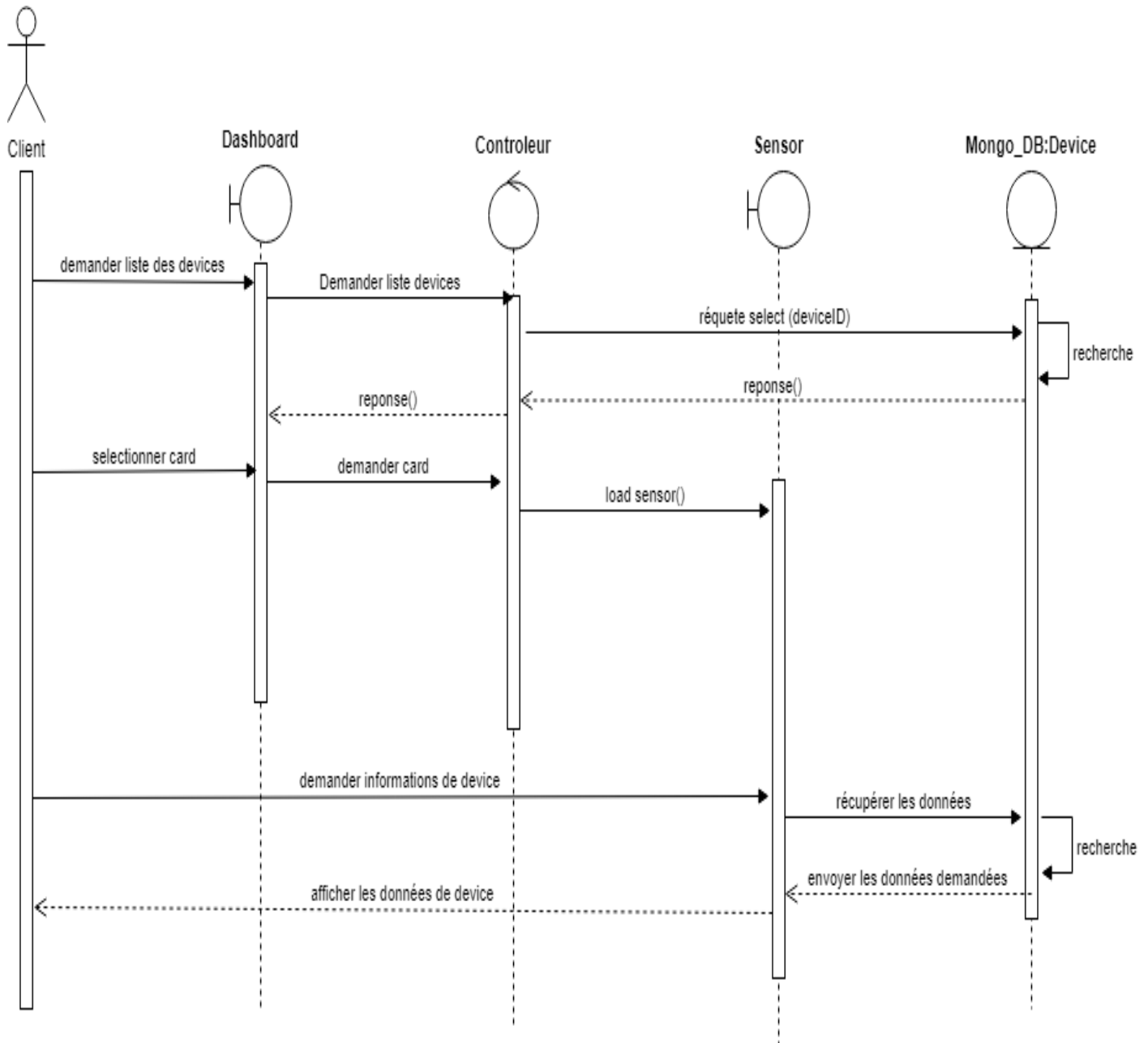


Figure 9 Diagramme de séquence consulter la liste des devices

## V. Diagramme de classe

Le diagramme de classes est la clé de la conception orientée objet. Ce diagramme représente la structure du code à développer. Certaines applications UML permettent même d'exporter du code à partir de diagrammes de classes. Cela permet d'unifier le travail de plusieurs programmeurs au sein d'une même équipe, en plus de sauver du temps.

La figure 10 présente le diagramme de classe de notre application.

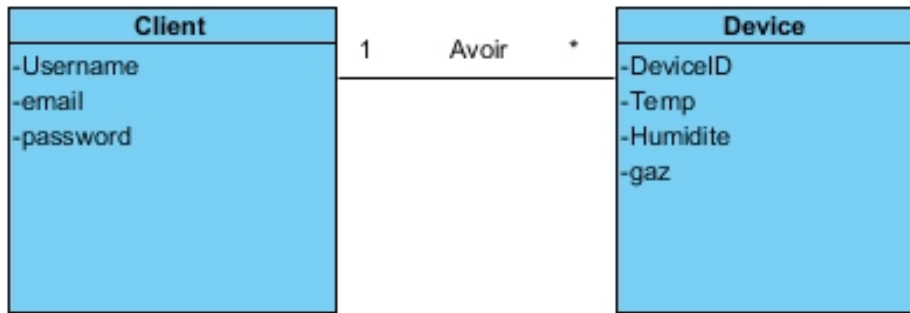


Figure 10 : Diagramme de classe

## VI. Etude fonctionnelle

Pour pouvoir tester le fonctionnement de notre plateforme, nous devons configurer des cartes embarquées et quelques capteurs. Dans cette section, nous allons présenter l'étude faite pour choisir les critères du choix de la carte embarquée et les capteurs choisis.

### 1. Choix des cartes embarquées

#### a. Carte Arduino UNO

Arduino est une plate-forme électronique open source accompagnée d'un matériel et de logiciels pour concevoir, développer et tester des prototypes et des produits électroniques complexes. Le matériel se compose d'un microcontrôleur avec d'autres composants électroniques qui peuvent être programmés en utilisant le logiciel pour faire presque n'importe quelle tâche. L'Arduino Uno est l'une des cartes les plus populaires de la famille Arduino et un excellent choix pour les débutants.

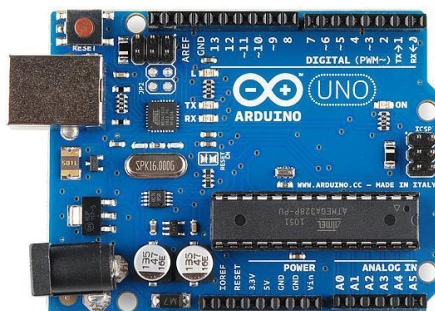


Figure 11 : Arduino Uno

## b. Module WIFI

ESP8266 est un module WIFI à faible coût permettant d'ajouter des fonctionnalités Wifi à un projet de microcontrôleur existant via une connexion série UART. Plus précisément, ESP8266 est un système sur puce (SoC) avec des capacités pour Wi-Fi de 2,4 GHz (802.11 b / g / n, supportant WPA / WPA2), entrée / sortie polyvalente (8 GPIO), Inter -Integrated Circuit (IC), modulation d'impulsions en largeur (PWM), protocole 802.11 b / g / n et protocole TCP / IP intégré STAC, 64 KB de RAM d'instruction, mémoire flash de 512 ko également un processeur RISC 32 bits intégré de faible puissance pourrait être utilisé comme processeur d'application. Le module peut même être reprogrammé pour agir comme un WiFi autonome connecté.

ESP8266 permet d'être intégré avec les capteurs et autres appareils spécifiques à l'application via ses GPIO.

C'est la première et la plus simple carte utilisant l'ESP8266. Il permet d'attacher des lignes série, et ne casse que deux broches GPIO pour un usage natif.

Comme le montre la figure 9, ESP8266 a 8 broches, 4 dans la rangée de 2. La première broche en haut à gauche est GND. Les deux broches à droite de la masse sont GPIO 2 et 0. La broche en haut à droite est la broche RX et la broche en bas à gauche est TX, ce sont les broches pour la communication sont également utilisés pour programmer le module. Les broches du milieu en bas sont CH\_PD (mise hors tension de la puce) et RST (réinitialisation).

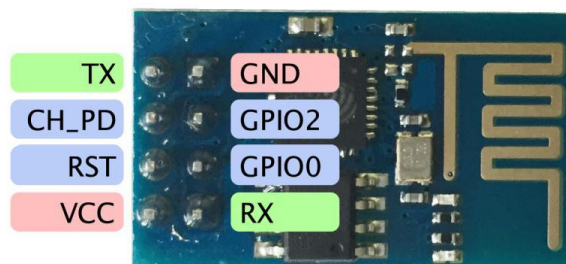


Figure 12 : Module WIFI ESP8266-1E

### Remarque

Nous avons utilisé la carte Arduino UNO en association avec un module wifi ESP8266 et deux capteurs (DHT11, ADC). Nous les avons programmés et nous avons réalisé un premier test mais nous avons remarqué que la synchronisation entre le module wifi et la carte Arduino UNO prend beaucoup du temps et beaucoup de lignes de code, en outre, les fils utilisés pour la connexion entre eux sont très fragiles. D'où, nous avons eu l'idée d'utiliser la carte NodeMCU

ESP8266-12E dans lequel il y a un module wifi intégré et beaucoup d'autres avantages (mentionnées dans le tableau 4).

### c. NodeMCU esp8266

NodeMCU est une plate-forme open source IoT. Il contient le firmware qui fonctionne sur la carte ESP8266 Wi-Fi **Soc** d'Espressif Systèmes et le matériel qui est basé sur le module ESP-12. Le terme "NodeMCU" se réfère par défaut au firmware plutôt qu'aux kits de développement. Le firmware utilise le **Lua** qui est un langage de script. Il est basé sur le projet eLua et construit sur l'Espressif **Non-OS SDK** pour ESP8266. Il utilise de nombreux projets open source comme **lua-cjson** et **spiffs**.



Figure 13 NodeMCU

### d. Etude comparative entre les cartes Arduino UNO et NodeMCU

Le tableau 4 montre une comparaison entre la carte ES8266 et la carte Arduino UNO :

Tableau 4 : Etude comparative entre Arduino Uno et NodeMCU

	<b>Arduino UNO</b>	<b>ESP8266 -12E</b>
<b>Microcontrôleur</b>	ATmel ATmega328p	Tensilica Xtensa LX106
<b>Espace d'adressage</b>	8bits	32 bits
<b>Fréquence d'horloge</b>	160MHZ	80MHZ
<b>Mémoire flash</b>	32kio	Up to 4Mb
<b>Mémoire RAM</b>	2kio	96kio



<b>E/S numérique</b>	18 (5V) dont 6 capables de PWN	16 (3,3V), toutes capables de PWN
<b>Module WIFI</b>	Non	Oui Support : 802.11b/g
<b>Entrées analogiques</b>	5 (10 bits)	1 (10 bits, 1V max (2))
<b>Prix</b>	60 DT	40 DT

## 2. Capteurs

Pour tester le fonctionnement de notre plateforme, nous avons utilisé les capteurs suivants : capteur température/humidité et capteur de gaz.

### a. Capteur Température / Humidité

Le **DHT11** permet, comme son nom l'indique, de mesurer l'**humidité** ambiante. Une mesure utile en matière de météorologie, mais aussi dans le secteur de l'habitat. Un air trop humide, ou trop sec, peut être source de désagréments, voire engendrer des risques sanitaires.

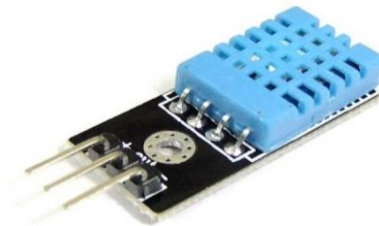


Figure 14 : Capteur DHT11

### b. Capteur de GAZ

Le détecteur de gaz est un équipement de protection individuelle. Le détecteur de gaz est un appareil pour surveiller et mesurer le pourcentage atmosphérique, piloté par un microcontrôleur (en l'occurrence celui d'une chambre souterraine ou d'une canalisation) en oxygène, gaz toxique et gaz explosif.



Figure 15 : Capteur ADC

## VII. Diagramme de déploiement

En UML, un diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux.

La figure 16 présente le diagramme de déploiement de notre plateforme.

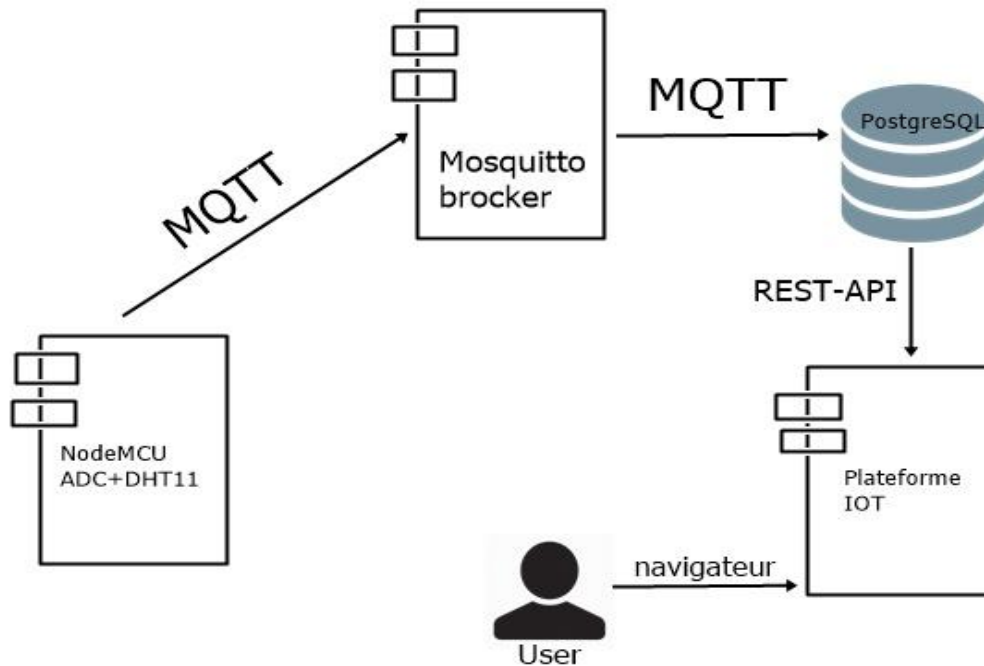


Figure 16 : Diagramme de déploiement

## Conclusion

Dans ce chapitre, nous avons présenté l'étude fonctionnelle et conceptuelle de l'application. Une modélisation est faite en utilisant le langage UML permettant de dégager le modèle logique des données qui sera exploité lors de l'implémentation. Ce modèle sera transformé en modèle physique de données. Les besoins fonctionnels sont bien décrits à travers le diagramme de cas d'utilisation générale et ses raffinements. Nous avons présenté une conception détaillée en donnant d'abord une vue sur le déroulement des événements à travers des diagrammes de séquences et enfin une vue globale du diagramme de classe. Nous avons achevé ce chapitre avec un diagramme de déploiement afin de décrire la disposition matérielle. Le chapitre suivant sera consacré à la phase de configuration du système embarqué et les résultats de la partie implémentation de la plateforme IoT.



# **Chapitre 3**

## **Configuration et Implémentation**

# Introduction

Le présent chapitre est consacré à la configuration de la partie matériels et la présentation de la partie implémentation et développement. Nous présenterons la configuration de la carte ESP8266-12E faite afin d'établir le test. La deuxième phase est l'implémentation considérée comme étant la concrétisation finale de toute la méthode de conception. Nous menons tout d'abord un environnement de réalisation où nous décrivons les ressources logicielles et matérielles utilisées dans le développement de notre projet. Aussi, nous présentons quelques interfaces réalisées pour illustrer le fonctionnement de quelques activités du système

## I. Environnement de réalisation et de développement

### 1. Matériels de base

Au cours des différentes phases du projet, nous avons utilisé un ordinateur portable avec les caractéristiques suivantes:

*Tableau 5: Matériels de base*

<b>Processeur</b>	Intel i7 -7700HQ 3.4 GHz
<b>Mémoire vive</b>	8 GO
<b>Mémoire graphique</b>	4 GO GTX950
<b>Disque dur</b>	1To HDD

### 2. Choix des langages de développement

#### a. Le protocole MQTT

MQTT est le protocole construit pour M2M et IoT qui est utilisé pour fournir des performances nouvelles et révolutionnaires. Il ouvre de nouvelles zones pour des cas d'utilisation de messagerie pour des milliards de choses connectées via Internet. Comme MQTT est spécialisé dans les environnements à faible bande passante et à latence élevée, il est considéré comme un protocole idéal pour la communication machine-to-machine (M2M).



Figure 17: Logo mosquitto

## b. Python

Python est un langage de programmation objet, multi-paradigme et multiplateformes. 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 ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.

Python est un langage qui peut s'utiliser dans de nombreux contextes et s'adapter à tout type d'utilisation grâce à des bibliothèques spécialisées. Il est cependant particulièrement utilisé comme langage de script pour automatiser des tâches simples mais fastidieuses, comme un script qui récupérerait la météo sur Internet ou qui s'intégrerait dans un logiciel de conception assistée par ordinateur afin d'automatiser certains enchaînements d'actions répétitives.



Figure 18: Logo Python

## c. Django

Django est un cadre de développement web source ouvert en Python. Il a pour but de rendre le développement web 2.0 simple et rapide. Pour cette raison, le projet a pour slogan « Le Framework web pour les perfectionnistes sous pression ».

Django est un cadre de développement qui s'inspire du principe MVC ou MTV (la vue est gérée par un gabarit) composé de trois parties distinctes :

- Un langage de gabarits flexible qui permet de générer du HTML, XML ou tout autre format texte ;
- Un contrôleur fourni sous la forme d'un « remapping » d'URL à base d'expressions rationnelles.
- Une API d'accès aux données est automatiquement générée par le cadre compatible CRUD. Inutile d'écrire des requêtes SQL associées à des formulaires, elles sont générées automatiquement par l'ORM.

En plus de l'API d'accès aux données, une interface d'administration fonctionnelle est générée depuis le modèle de données. Un système de validation des données entrées par l'utilisateur est également disponible et permet d'afficher des messages d'erreur automatiques.

Django peut être considéré comme une boîte à outils où chaque module peut fonctionner de façon indépendante.

Un exemple de cette modularité est la plate-forme de développement Google App Engine, où l'ORM de base a été remplacé par une API sur les BigTable de Google.



*Figure 19 Logo Django*

## **d. FLASK**

Flask est un Framework open-source de développement web en Python. Son but principal est d'être léger, afin de garder la souplesse de la programmation Python, associé à un système de Template. Il est distribué sous licence BSD.

Flask a été créé initialement par **Armin Ronacher** comme étant un poisson d'avril. Le souhait de Ronacher était de réaliser un Framework web contenu dans un seul fichier Python mais pouvant maintenir des applications très demandées.

- Contient un serveur de développement et un debugger

- Supporte les tests unitaires
- Utilise le moteur de Template Jinja2
- Supporte les cookies sécurisés (session)
- Entièrement compatible avec WSGI 1.0
- Se base sur l'Unicode
- Dispose d'une documentation complète
- Compatible avec Google App Engine
- Il est possible de créer des extensions

### **e. HTML**

HTML (HyperText Markup Language) est l'ensemble des symboles ou codes de balisage insérés dans un fichier destiné à être affiché sur une page de navigateur World Wide Web. Chaque page Web est en fait un fichier HTML. Chaque fichier HTML est simplement un fichier de texte brut, mais avec une extension de fichier .html et est constitué de plusieurs balises HTML ainsi que le contenu d'une page Web.

### **f. CSS**

Cascading Style Sheets (CSS) est un langage de feuille de style utilisé pour décrire la présentation d'un document écrit dans un langage de balisage. CSS gère l'apparence d'une page Web. En outre CSS, peut contrôler la couleur du texte, le style de polices, les modèles de mise en page, les variations d'affichage pour différents appareils et tailles d'écran ainsi qu'une variété d'autres effets.

### **g. JavaScript**

JavaScript est un langage de programmation informatique dynamique. Il est léger et le plus souvent utilisé en tant que partie de pages Web, dont les implémentations permettent au script côté client d'interagir avec l'utilisateur et de faire un appel côté serveur. C'est un langage de programmation interprète avec des capacités orientées objet.

### **h. JQuery**

JQuery est une bibliothèque JavaScript multiplateforme conçue pour simplifier l'écriture de scripts HTML côté client. JQuery simplifie les interactions HTML, la gestion des événements,

l'animation et les interactions Ajax pour le développement Web rapide. Nous avons utilisé jQuery pour faciliter l'utilisation de JavaScript sur notre interface web.

### **i. Bootstrap**

Bootstrap est le Framework HTML, CSS et JS le plus populaire pour développer des projets mobiles récurrents et mobiles sur le web. Bootstrap est entièrement gratuit à télécharger et à utiliser. Il comprend des modèles de conception HTML et CSS pour la typographie, les formulaires, les boutons, les tableaux, la navigation, les modaux, les carrousels d'images et bien d'autres. Il facilite la création d'un site Web qui peut s'ajuster automatiquement sur tous les appareils, des téléphones intelligents aux ordinateurs de bureau, etc.

### **j. VueJS**

Vue.js est un Framework JavaScript open-source permettant de créer des interfaces utilisateur. L'intégration dans des projets utilisant d'autres bibliothèques JavaScript est simplifiée avec Vue car elle est conçue pour être progressivement adoptée. Vue peut également fonctionner comme une structure d'application Web capable d'alimenter des applications d'une page avancée.

## **3. Choix de base de données**

Dans notre projet, nous aurons besoin de deux base de données : une base de données NoSQL Mongo DB et un SQBD à savoir Postgres

### **a. Mongo DB**

#### **– Les bases de données NoSQL**

L'Internet des Objets (IoT) est en constante expansion et les données qui y transitent grandissent immensément avec le réseau. C'est pourquoi les bases de données relationnelles et les méthodes régulières de gestion des données ne peuvent pas faire face aux besoins croissants.

NoSQL est une série de technologies, d'approches et de projets visant à la mise en œuvre de modèles de base de données avec des différences significatives par rapport au SGBD traditionnel fonctionnant avec le langage SQL.

NoSQL offre une grande stabilité et une grande résistance au partitionnement des données. C'est exactement ce dont les bases de données de l'Internet des objets ont besoin, la capacité de stocker différents types de données comme des données structurées, semi-structurées et non structurées et de croître dynamiquement avec des exigences changeantes d'application.



La seule chose que la plupart des bases de données NoSQL ont en commun est qu'elles ne suivent pas un modèle de données relationnel. Les bases de données NoSQL appartiennent généralement à l'une des quatre catégories suivantes:

Les magasins à valeur-clé sont les plus simples. Chaque élément de la base de données est stocké en tant que nom d'attribut (ou "clé") avec sa valeur.

Les bases de données basées sur des colonnes stockent les données ensembles en tant que colonnes au lieu de lignes et sont optimisées pour les requêtes sur de grands ensembles de données.

Les bases de données de documents associent chaque clé à une structure de données complexe appelée document. Les documents peuvent contenir plusieurs paires de paires valeur / clé, ou paires de tableaux de clés, ou même des documents imbriqués.

Certaines bases de données NoSQL populaires qui appartiennent respectivement aux catégories ci-dessus:

- **Bases de données de valeurs clés** : Membase, Redis, MemcacheDB, etc.
  - **Bases de données orientées document** : MongoDB, HBase, Cassandra, Amazon SimpleDB, Hypertable, etc.
  - **Bases de données graphiques** : Neo4j, OrientDB, Facebook Open Graph, FlockDB.
- Bases de données à base de colonnes - CouchDB, OrientDB, etc.

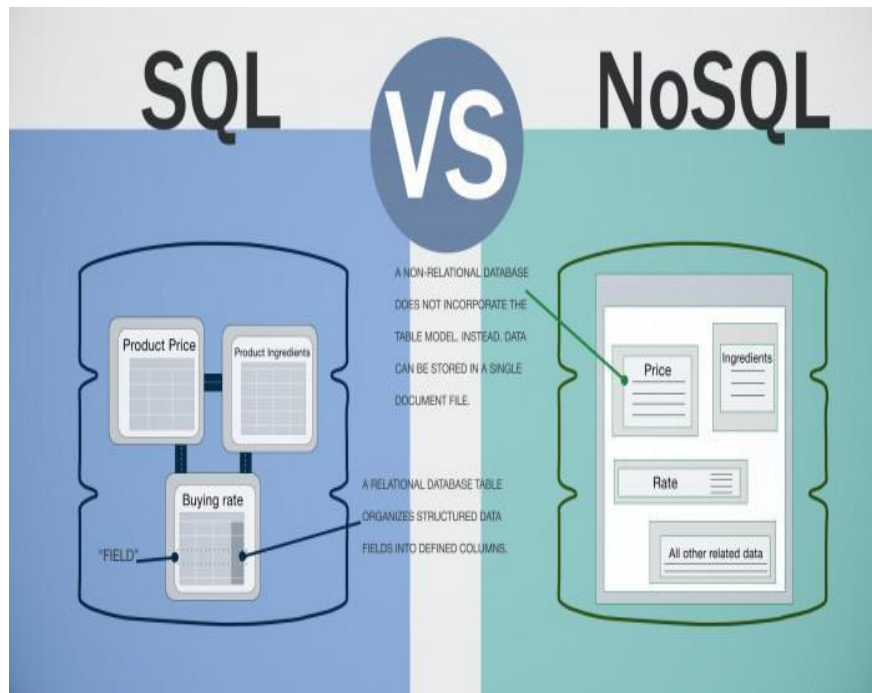


Figure 20: SQL vs NoSQL

#### – Base de données MongoDB

MongoDB est une base de données de documents open-source et une base de données NoSQL leader. MongoDB est écrit en C ++. MongoDB est une base de données orientée document, multiplateforme, offrant des performances élevées, une haute disponibilité et une évolutivité aisée. MongoDB travaille sur le concept de collection et de document.

Collection est un groupe de documents MongoDB. C'est l'équivalent d'une table RDBMS. Une collection existe dans une base de données unique. Les collections n'appliquent pas de schéma. Les documents d'une collection peuvent avoir des champs différents. Généralement, tous les documents d'une collection ont un but similaire ou connexe.

Un document est un ensemble de paires clé-valeur. Les documents ont un schéma dynamique. Le schéma dynamique signifie que les documents de la même collection n'ont pas besoin d'avoir le même ensemble de champs ou de structure et les champs communs dans les documents d'une collection peuvent contenir différents types de données.

Toute base de données relationnelle a une conception de schéma typique qui indique le nombre de tables et la relation entre ces tables. Alors que dans MongoDB, il n'y a pas de concept de relation.

– ***Avantages de MongoDB par rapport au SGBDR:***

- Une collection contient des documents différents. Aucune jointure complexe
- MongoDB est facile à mettre à l'échelle.
- Utilise la mémoire interne pour le stockage qui permet un accès plus rapide des données.
- Les données sont stockées sous la forme de documents de style JSON.
- MongoDB est la base de données NoSQL la plus populaire car elle préserve les meilleures fonctionnalités des bases de données relationnelles tout en intégrant les avantages de NoSQL.

## **b. PostgreSQL**

PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBDRO). C'est un outil libre disponible selon les termes d'une licence de type BSD.

Ce système est concurrent d'autres systèmes de gestion de base de données, qu'ils soient libres (comme MariaDB et Firebird), ou propriétaires (comme Oracle, MySQL, Sybase, DB2, Informix et Microsoft SQL Server). Comme les projets libres Apache et Linux, PostgreSQL n'est pas contrôlé par une seule entreprise, mais est fondé sur une communauté mondiale de développeurs et d'entreprises.

# **II. Outils de développement**

## **1. Pycharm Community**

PyCharm est un environnement de développement intégré (abrégé EDI en français ou en anglais: IDE (Integrated Development Environment)) utilisé pour programmer en Python.

Il offre l'analyse de code, un débogueur graphique, la gestion des tests unitaires, l'intégration des logiciels de gestion de versions et supporte le développement web avec Django.

Il est multiplateforme et fonctionne sous Windows, Mac OS X et Linux. Il est décliné en édition professionnelle, réalisé sous licence propriétaire et en édition communautaire réalisé sous licence Apache.



Figure 22 Logo 1. Pycharm Community

## 2. Visual Paradigm

Visual Paradigm est un logiciel qui permet aux étudiants, enseignants et professionnels de créer des organigrammes, des diagrammes UML, des diagrammes de flux de travail, des diagrammes d'ingénierie et bien d'autres types de diagrammes. Visual Paradigm permet de créer un large éventail de diagrammes des modèles, des formes et des outils de dessin tout en travaillant dans un environnement de bureau intuitif et familier.



Figure 23: Logo Visual paradigm

## 3. IDE Arduino

Arduino est une plate-forme open-source utilisée pour les projets de construction électronique. Arduino se compose à la fois d'une carte de circuit programmable physique (souvent appelée microcontrôleur) et d'un logiciel, ou IDE (Integrated Development Environment) qui fonctionne sur un ordinateur, utilisé pour écrire et télécharger du code informatique sur le disque physique. L'IDE Arduino est un logiciel open source et extensible, le logiciel Arduino est publié en tant qu'outil open source, disponible pour l'extension par des programmeurs expérimentés. Le langage peut être étendu grâce aux bibliothèques C ++, ce qui facilite l'apprentissage de la programmation.



Figure 24: Logo arduinoIDE

## 4. Sublime Text

Sublime Text est un éditeur de texte générique codé en C++ et Python, disponible sur Windows, Mac et Linux. Le logiciel a été conçu tout d'abord comme une extension pour Vim, riche en fonctionnalités<sup>1</sup>.

Depuis la version 2.0, l'éditeur prend en charge 44 langages de programmation majeurs, tandis que des plugins sont souvent disponibles pour les langages plus rares.



Figure 25: Logo Sublime Text

## III. Réalisation des prototypes

Au cours de notre travail, nous avons réalisé trois prototypes.

### 1. Implémentation du prototype 1

Comme nous l'avons déjà mentionné, nous avons utilisé l'IDE Arduino pour programmer la carte embarqué ESP8266. D'abord, nous avons ajouté l'extension ESP8266-1E à l'IDE Arduino (figure 26).

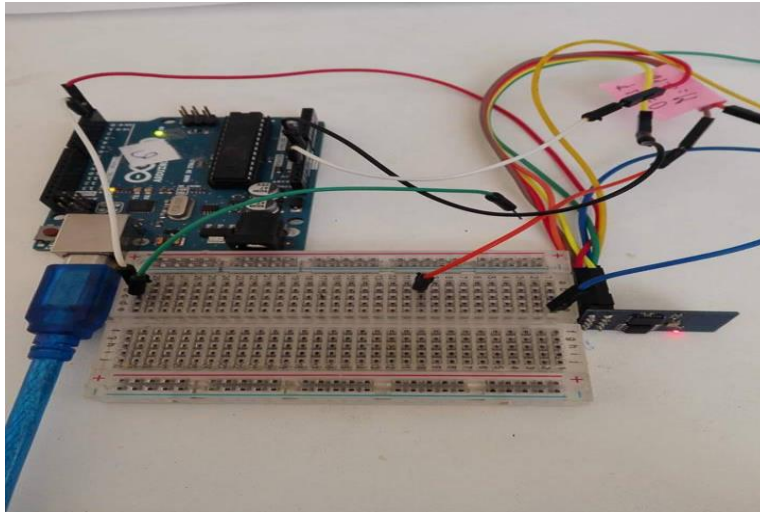


Figure 26: Prototype 1

## 2. Implémentation du Prototype 2

L'ESP8266 a seulement deux broches qui sont destinées à l'entrée / sortie. Pour développer le code, nous avons placé le capteur sur l'une des broches GPIO appropriées et nous avons pu utiliser le terminal série pendant le débogage.

La figure 27 présente le deuxième prototype avec le capteur.

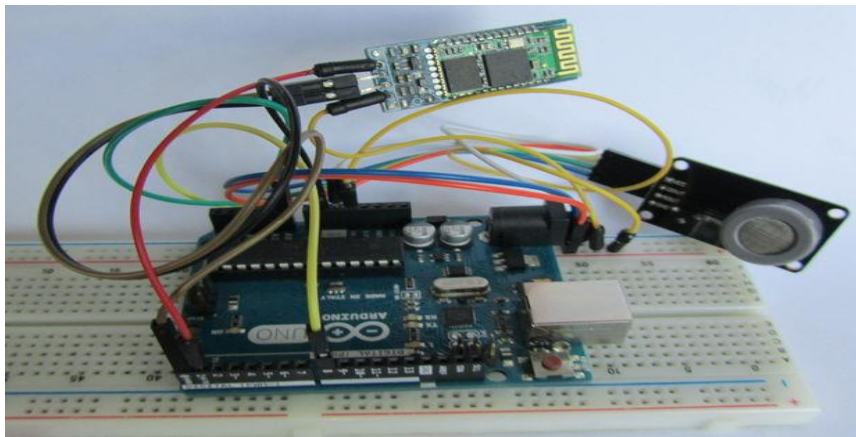


Figure 27: Prototype 2

## 3. Implémentation du Prototype 3

Nous avons placé le capteur DHT11 sur l'une des broches GPIO appropriées et nous avons pu utiliser le terminal série pendant le débogage.

La figure 28 illustre la carte embarquée avec le capteur DHT11.

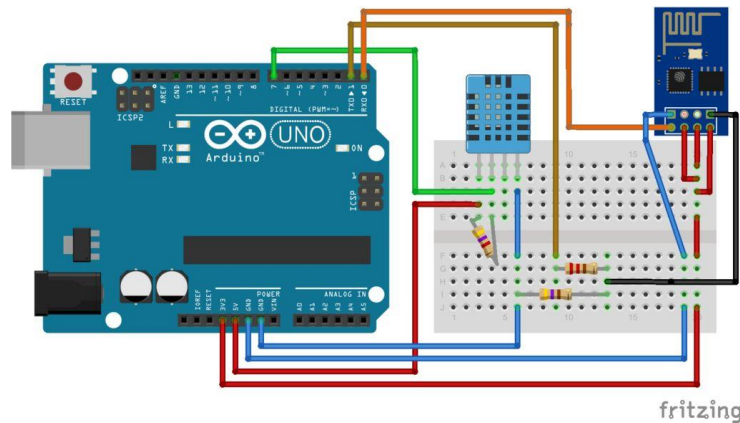


Figure 28 : Prototype 3

## iv. Implémentation

### 1. Page de Bienvenue

La figure 29 présente la première page où l'utilisateur peut se connecter pour accéder à la plateforme ou s'inscrire pour créer un compte.

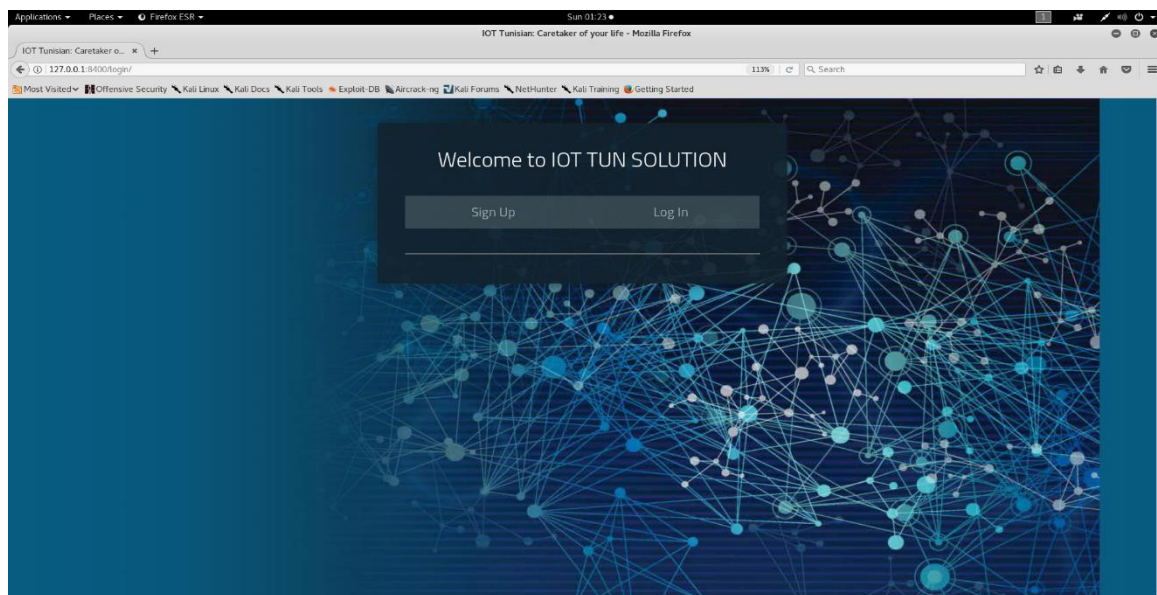


Figure 29: Page de Bienvenue

Il est composé de deux parties :

#### – Inscription « Sign up » :

L'utilisateur remplir le formulaire d'inscription pour s'inscrire et pouvoir s'authentifier par la suite (figure 30).



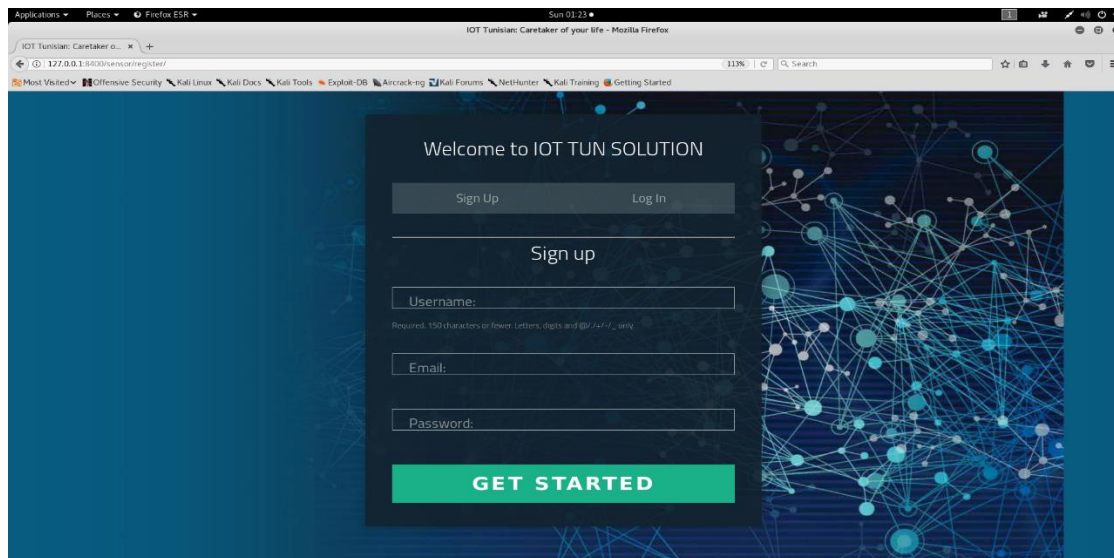


Figure 30: Page d'inscription

### – Authentification « Login »

L'utilisateur saisie son username et son mot de passe pour se connecter (figure 31).

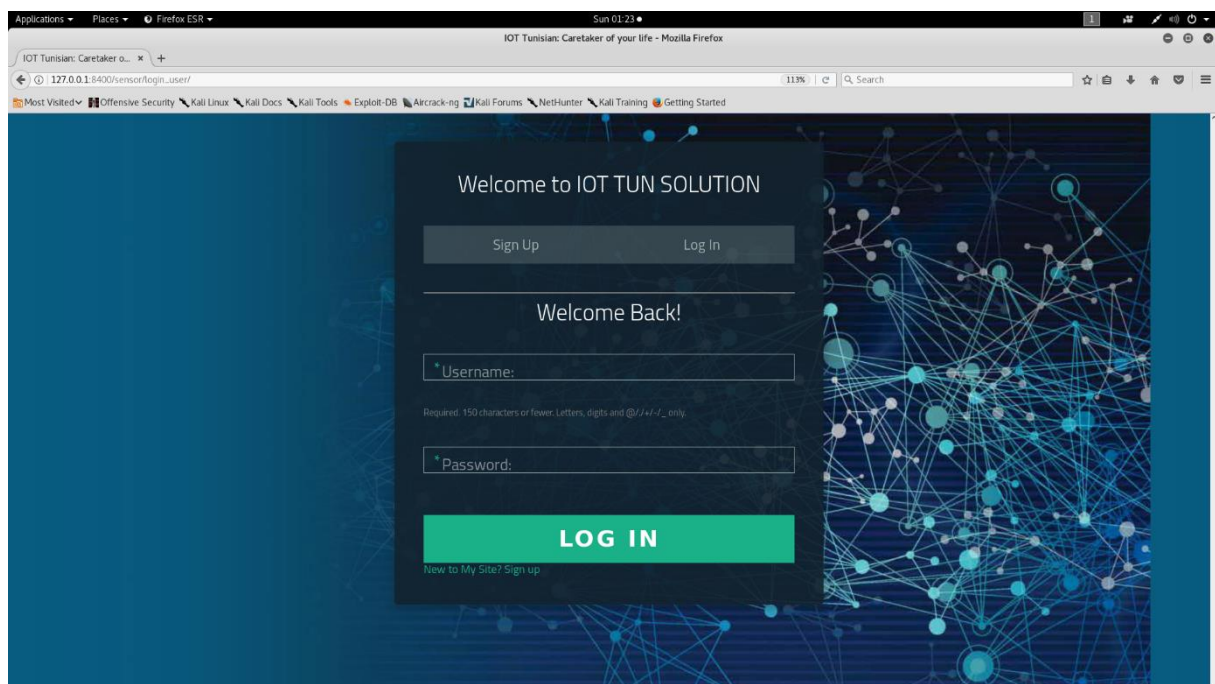


Figure 31: Page d'authentification

## 2. Tableau de bord

Lorsque l'utilisateur est authentifié, il va rediriger vers la page « Dashboard » (figure 32)



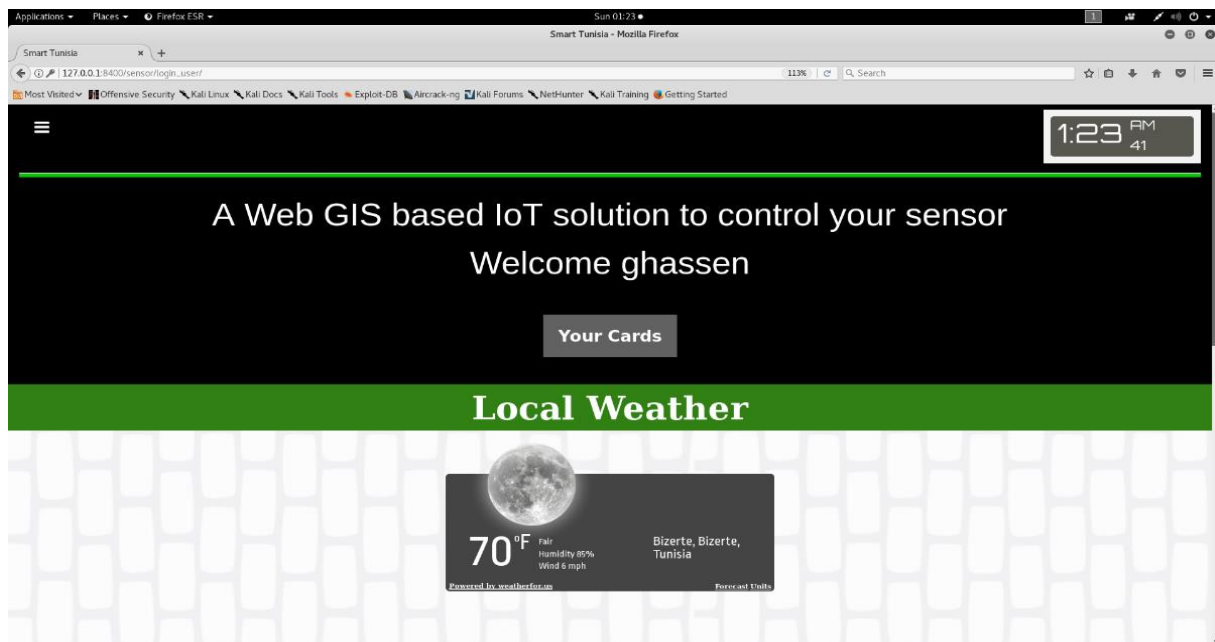


Figure 32 : Page Dashboard

Dans cette page, le client peut consulter la liste des appareils en cliquant sur « Your Cards ». La liste s'affiche comme le montre la figure 33. Dans notre cas, nous avons une seule carte.

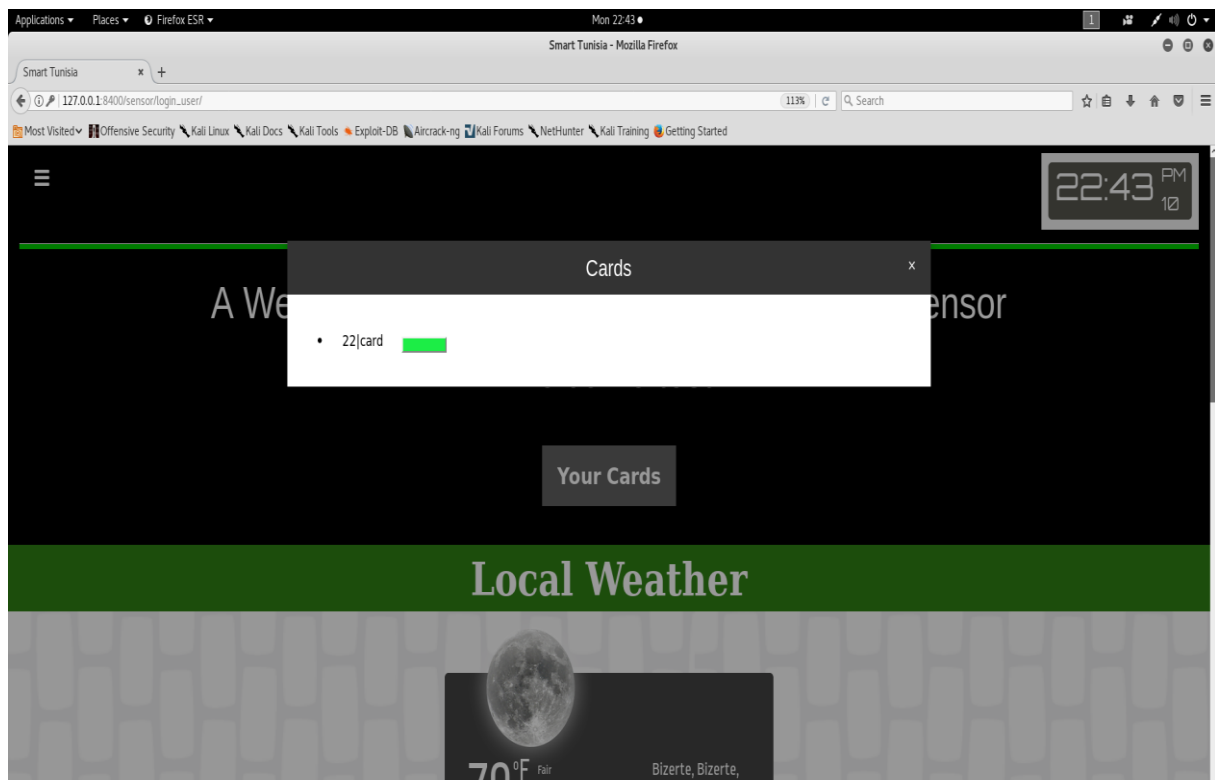


Figure 33 Page liste des cartes disponibles

En cliquant sur la carte souhaitée, nous pouvons choisir le paramètre à superviser : Température (figure 34) ou Humidité (figure 36) qui sont les deux paramètres que nous avons arrivé à superviser faite de temps.

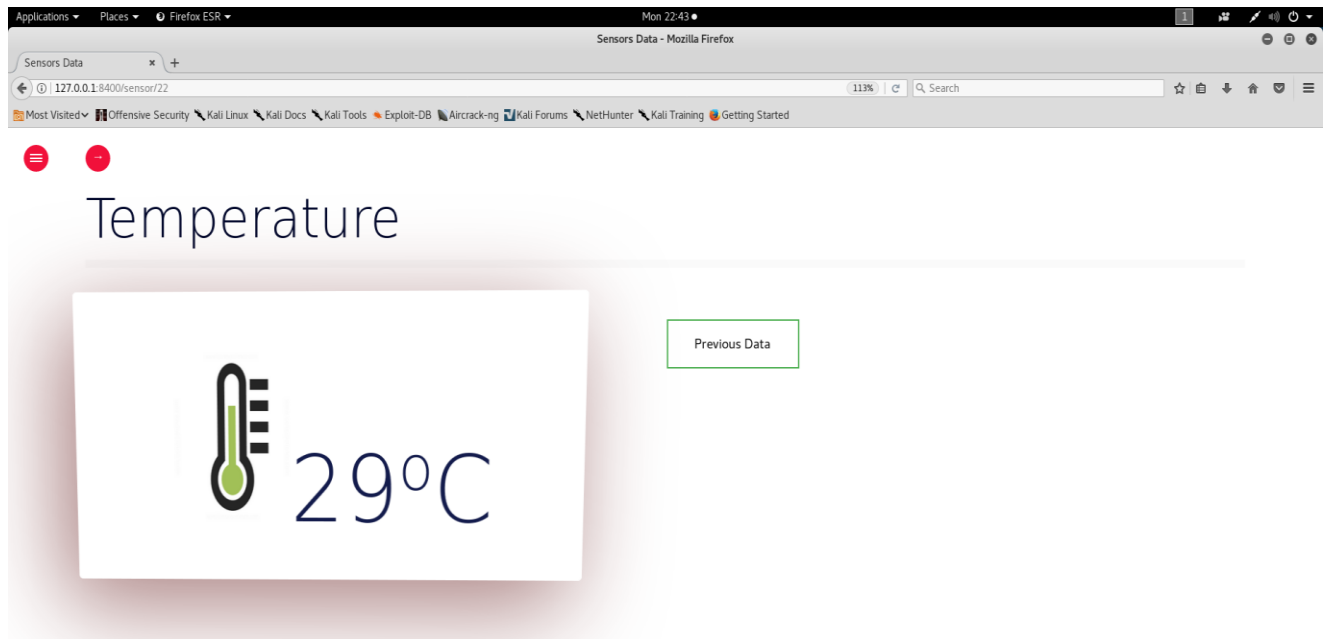


Figure 34 Page supervision du capteur Température

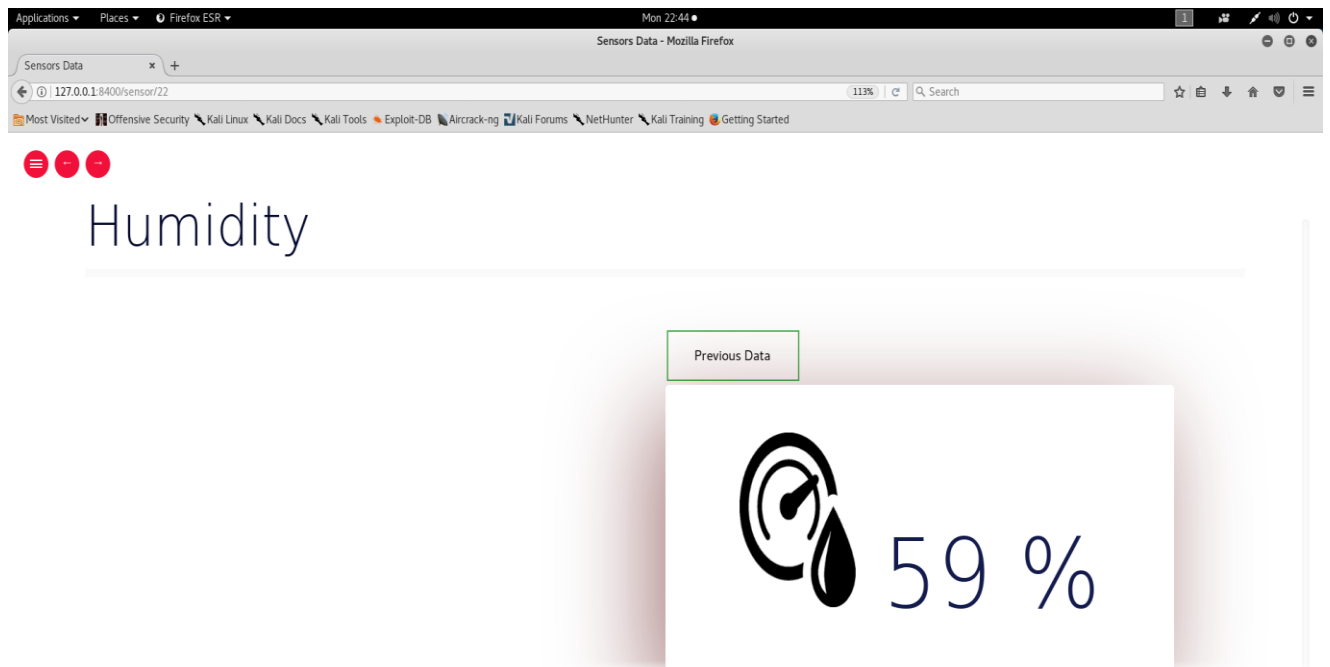


Figure 35 Page supervision de l'humidité

### 3. Page de consultation des informations collectées en temps réel

Grâce à notre plateforme, nous pouvons collecter tous les informations reçues des capteurs et les affichés en temps sous forme des graphes (figure 36).

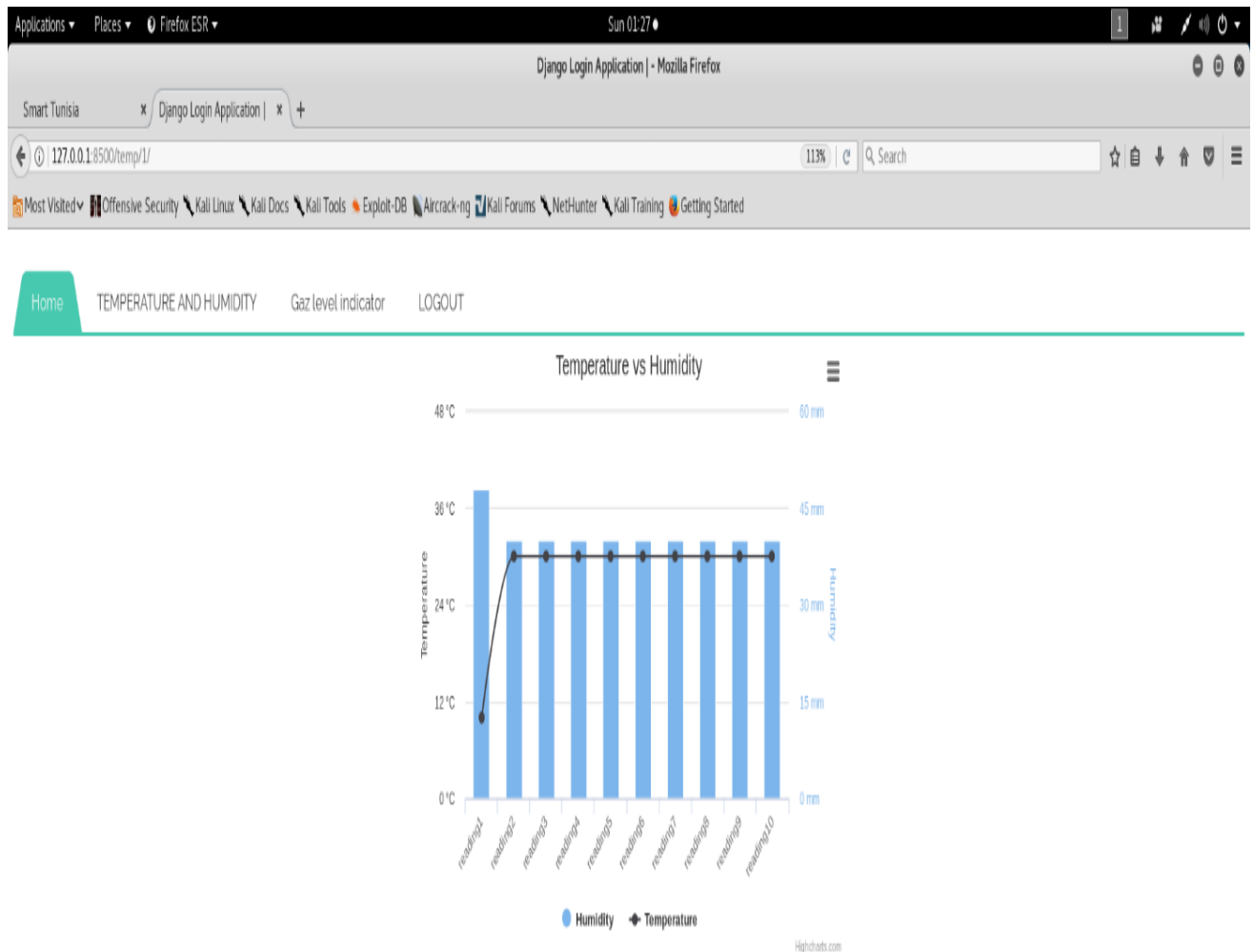


Figure 36: Monitoring page

### 4. Application FrontOffice

La figure 37 présente la partie FrontOffice.

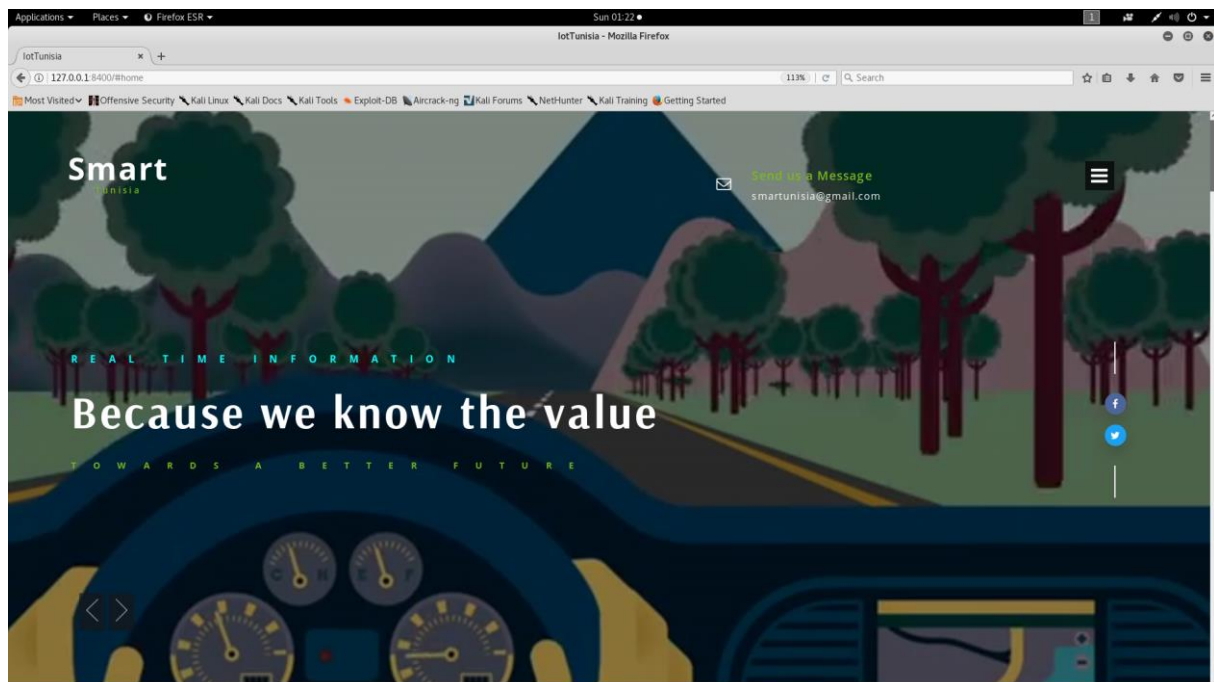


Figure 37: Front-office

## Conclusion

Dans le présent chapitre, nous avons décrit les étapes de l'implémentation de la partie embarquée dans laquelle nous avons utilisé NodeMCU ESP8266-12E. Nous avons conçu deux parties, une back-office pour la gérer les clients, cartes, capteurs, etc. Une autre partie est la Front-Office permettant l'accès aux valeurs prises des capteurs.

# Conclusion et perspectives

Ce projet ainsi que les différentes approches auxquelles j'ai opté lors des différentes phases du travail sont développés au Centre d'Innovation de Tunis au sein de l'entreprise DR Technologie,

Le rapport a commencé par un aperçu général du concept de l'Internet des objets et du modèle des maisons intelligentes. Le deuxième chapitre s'est focalisé sur les analyses et la spécification des exigences contenant une description de la méthodologie adoptée et une présentation des cas d'utilisation de ce projet et une étude conceptuelle. Dans le dernier chapitre, une brève terminologie a été illustrée pour donner un aperçu sur les connaissances techniques abordé dans notre projet et aussi nous avons présenté quelques interfaces de notre plateforme IoT.

En conclusion, l'objectif global de ce projet de fin d'études est de proposer un prototype qui aborde certains aspects des déficiences des solutions de marché existantes. Les objectifs particuliers de la mise en œuvre sont détaillés ici:

- Le choix des bons capteurs et dispositifs de communication pour ce projet. obtention des capteurs de mouvement et l'envoi de leurs valeurs via WIFI.
- Le développement d'une interface frontale pour les utilisateurs et l'administrateur
- Le développement d'une application Back end pour recevoir des appels de l'interface utilisateur.
- Le développement d'une API reposante.

Ce projet de fin d'études a été une riche opportunité de couvrir divers aspects de l'ingénierie sur les logiciels, le matériel et l'architecture IoT. J'ai fait des grands progrès dans la réalisation des résultats visés et je suis impatient de mettre en œuvre et de finaliser ce prototype pour atteindre un produit commercialisable. Ce projet ouvre des perspectives pour mener d'autres projets approfondis dans ce secteur prometteur des bâtiments intelligents basés sur les technologies IoT.

# Webographie

- [1] *django*. Récupéré sur wikipédia : <https://fr.wikipedia.org/wiki/django>
- [2] *flask*. Récupéré sur wikipédia : <https://fr.wikipedia.org/wiki/flask>
- [3] *css*. Récupéré sur wikipédia : <https://fr.wikipedia.org/wiki/css>
- [4] *python*. Récupéré sur wikipédia : <https://fr.wikipedia.org/wiki/python>
- [5] *Jquery*. Récupéré sur wikipédia : <https://fr.wikipedia.org/wiki/jquery>
- [6] *bootstrap*. Récupéré sur wikipédia : <https://fr.wikipedia.org/wiki/bootstrap>
- [7] *pycharm*. Récupéré sur wikipédia : [https://fr.wikipedia.org/wiki/pycharm\\_community](https://fr.wikipedia.org/wiki/pycharm_community)
- [8] *Django+Vujs*. Récupéré sur <https://medium.com/quick-code/crud-app-using-vue-js-and-django-516edf4e4217>
- [9] *Arduino+esp8266*. Récupéré sur <http://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html>

# Annexe A : Installation de Mosquitto broker

## 1. Pré-Installation

- Système d'exploitation Linux : Linux Mint
- Ouvrir le port 1883 dans le pare-feu

### Etape 1 : Installer Mosquitto Broker

Il faut mettre à jour notre système d'exploitation afin d'installer par la suite la dernière version de mosquitto.

```
sudo apt-get update  
sudo apt-get install mosquitto
```

*Figure 38: Installation de mosquito*

Le service Mosquitto commencera après l'installation.

### Etape 2 : Installer Mosquitto-client

Les clients de Mosquitto nous aident à tester facilement MQTT via un utilitaire de ligne de commande.

```
sudo apt-get install mosquitto-clients
```

*Figure 39: Mosquitto Client*

# Annexe B : Installation de python, Django, Restframework, Psycopg2 et l'environnement virtuel

## Etape 1 : Installer Python + Django

Linux Mint est livré avec Python 3 et Python 2 pré-installés. Pour s'assurer que nos versions sont à jour, nous devons mettre à jour notre système avec apt-get:

```
$ sudo apt-get update  
$ sudo apt-get -y upgrade
```

*Figure 40: installer python*

L'option **-y** confirmera que nous acceptons tous les éléments à installer.

Une fois que le processus d'installation est terminé, nous pouvons vérifier la version de Python 3 installée dans le système en tapant:

```
$ python3 -V
```

*Figure 41: python version*

Pour gérer les paquets logiciels pour Python, nous installons pip:

```
$ sudo apt-get install -y python3-pip
```

*Figure 42: installer pip*



En fait, Pip est un outil à utiliser avec Python. pip installe et gère les paquets de programmation que nous voulons utiliser dans notre projet de développement. Dans notre cas, nous allons installer le paquet Django :

```
$ pip3 install django==2.0.5
```

*Figure 43: installer django*

Pour s'assurer d'avoir une configuration robuste pour notre environnement de programmation, il faut installer quelques paquets et outils de développement.

```
$ sudo apt-get install build-essential libssl-dev libffi-dev python-dev
```

*Figure 44: installer package*

## **Etape 2 : Configuration d'un environnement virtuel**

Les environnements virtuels vous permettent d'avoir un espace isolé sur votre ordinateur pour les projets Python, garantissant que chacun de vos projets puisse avoir son propre ensemble de dépendances qui ne perturberont aucun de vos autres projets.

La mise en place d'un environnement de programmation nous donne un plus grand contrôle sur nos projets Python et sur la manière dont les différentes versions des paquets sont gérées. Ceci est particulièrement important lorsque nous travaillons avec des packages tiers.

Nous pouvons configurer autant d'environnements de programmation Python que nous le souhaitons. Chaque environnement est essentiellement un répertoire ou un dossier dans notre ordinateur qui contient quelques scripts pour le faire fonctionner comme un environnement.

Nous devons d'abord installer le module venv, qui fait partie de la bibliothèque Python 3 standard, afin que nous puissions créer des environnements virtuels.

```
$ sudo apt-get install -y python3-venv
```

*Figure 45: installer l'environnement virtuel*

### Etape 3 : Installer Django Restframework

Django est un cadre de développement web source ouverte en Python.

```
$ pip install djangorestframework
```

Figure 46: Installer RestFramework

### Etape 4 : Installer Psycopg2

Psycopg est un adaptateur de base de données PostgreSQL pour le langage de programmation Python. Il est conforme à la norme DB-API 2.0.

Il est actuellement à la version 2.x, qui est une réécriture complète du code 1.x d'origine pour fournir des classes de style nouveau pour les objets de connexion et de curseur et autres bonbons sucrés. Comme le psycopg1, psycopg2 a été écrit dans le but d'être très petit et rapide, et stable comme un roc.

Psycopg est conçu pour les applications fortement multi-thread qui créent et détruisent beaucoup de curseurs et font un nombre évident de INSERT ou UPDATE concurrents.

```
ghassen@ouertani:~$ sudo apt-get install python-psycopg2
[sudo] password for jon:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  python-psycopg2-doc
The following NEW packages will be installed
  python-psycopg2
0 upgraded, 1 newly installed, 0 to remove and 334 not upgraded.
Need to get 153 kB of archives.
```

Figure 47: Installer Psycopg2

# Annexe C : Exemple du code

La figure présente un exemple du code

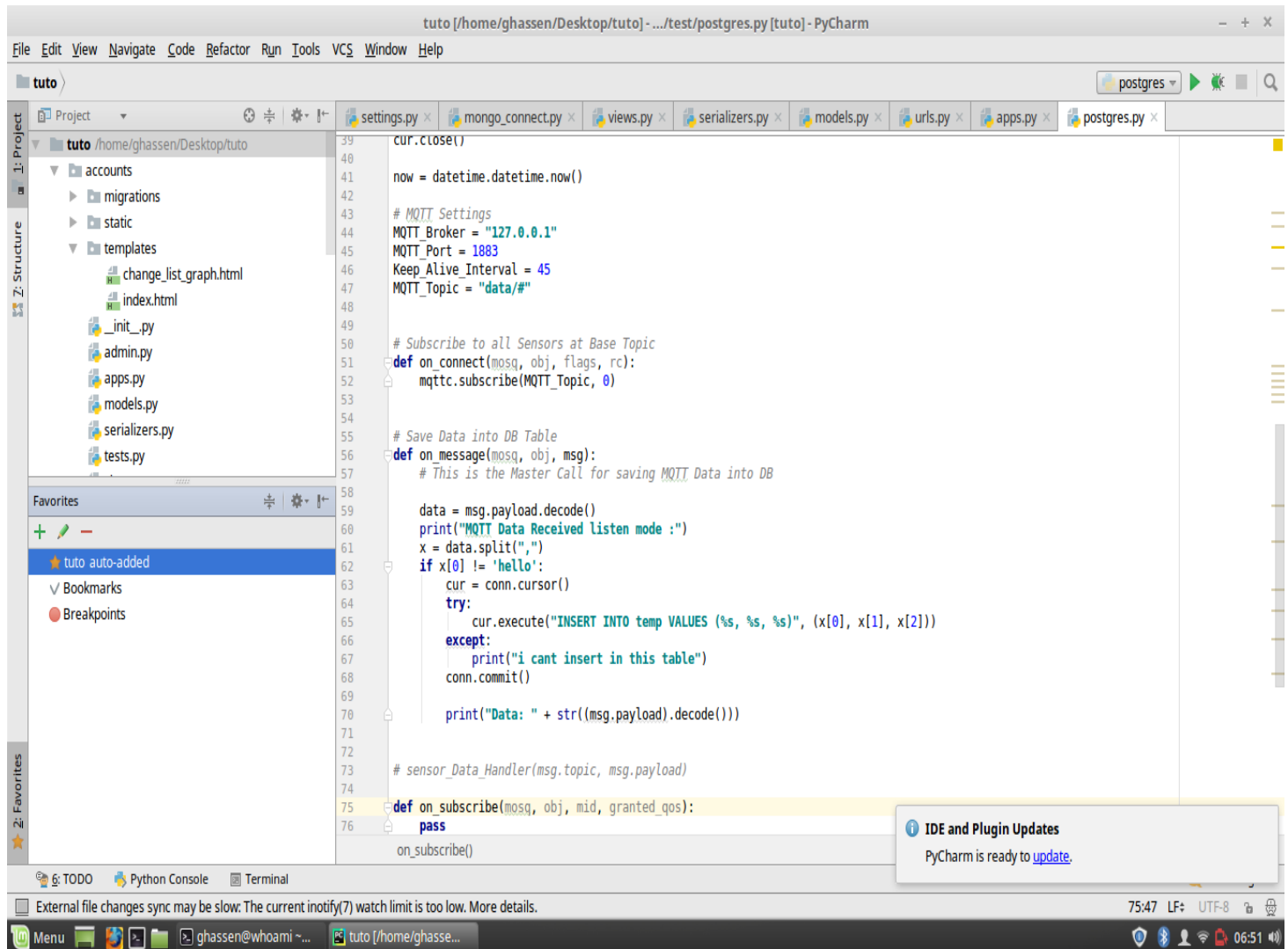


Figure 48 Exemple du Code