
République Algérienne Démocratique et populaire

Ministère de l'enseignement Supérieur et de la recherche scientifique

Université A/Mira de Bejaia, Faculté des sciences exactes, département d'informatique



Mémoire de fin d'étude

En vue de l'obtention du diplôme de master en informatique
spécialité : administration et sécurité des réseaux informatique.

Thème

Conception et réalisation d'une application mobile cross-
Platform pour l'Internet of things.

Réalisé par :

Mr ACHAT Asalas et Mr LAOUBI Lyes

Soutenu le 29 Juin 2017 devant le jury composé de :

Président :	MR ABBACHE Bournane	(Université A/MIRA Bejaia)
Examineur :	MR OUZEGANE Redouane	(Université A/MIRA Bejaia)
Examineur :	MR CHENNA ABDELBASSET	(Université A/MIRA Bejaia)
Encadreur :	MR NAFI Mohammed	(Université A/MIRA Bejaia)

Promotion 2016/2017

Remerciements

En premier lieu, nous remercions dieu le tout puissant de nous avoir donné la force et le courage pour réaliser ce modeste travail sans lequel nous n'aurions pas pu progresser.

Nous remercions chaleureusement notre promoteur Mr « Nafi Mchammed » de nous avoir fait confiance.

Nous tenons à citer dans ces remerciements les membres du jury qui ont bien voulu examiner et juger notre travail.

Nos profondes gratitude et salutation vont à tous ceux qui, de près ou de loin ont contribué à l'achèvement de ce travail.

Nous vous sommes très reconnaissantes.

Dédicaces

Nous dédions ce modeste travail à ceux qui nous ont donné la vie, nos chers parents qui se sont sacrifiés jour et nuit pour notre bonheur et notre réussite, et ont fait de sorte que rien ne nous manque. Que dieu les garde et les protège.

À nos frères et sœurs, camarades, et amis, à qui nous souhaitons plein de bonheur et de réussite dans leurs vies.

À tous ceux qui nous sont chers.

À tous ceux que nous connaissons de près ou de loin.

À tous ceux que nous aimons, et à tous ceux qui nous aiment.

Nous vous dédions ce travail.

Asalas et Lyes

Dedications

Table des matières

Table des matières	i
Liste des tableaux	iv
Table des figures	v
Liste des abréviations	vi
Introduction générale	1
Chapitre 01 : Généralités sur les applications mobiles, l'IOT, et les maisons connectées	2
1. Introduction	2
2. Application mobile	2
2.1. Evolution des applications mobiles (en nombre et en acquisition)	2
2.2. Objectifs, et perspectives des applications mobiles	3
3. Types d'applications et environnements de développement	3
3.1. Les applications natives	3
3.2. Les applications multiplateformes	4
4. Fonctionnement d'une application mobile	4
4.1. Architecture	4
4.2. Communication client-serveur	5
5. L'internet des objets (IOT)	5
5.1. Objet connecté	6
5.2. Domaines d'application de l'internet des objets	7
5.3. Composants d'un système IOT	8
5.4. Contraintes liées à l'IOT	9
6. Le web of things	10
7. Domotique et maison connectées	10
7.1. Domotique	10
7.2. Maison connectée	11
7.3. Différence entre domotique et maison connectée	11
7.4. Exemples d'usage dans une maison connectée	11
8. Techniques de mise en place d'une maison connectée	12
9. Solutions pour implanter une maison connectée	13
9.1. L'open hardware	13
9.2. L'open source	13
9.3. Système embarqué	13
9.4. Capteur	14
9.5. Actuateur (actionneur)	14

Table des matières

9.6. Solution à base de microcontrôleur (carte arduino)	14
9.7. Solution mixte (microcontrôleur + microprocesseur)	16
10. Conclusion	18
Chapitre 02 : Analyse et conception	19
1. Introduction	19
2. Présentation du projet	19
2.1. Problématique	19
2.2. Objectif	19
2.3. Fonctions du système	19
3. Modélisation des besoins	20
3.1. Identification des acteurs	20
3.2. Description des cas d'utilisation	20
3.3. Diagramme de cas d'utilisation	25
4. Diagramme de séquences	26
4.1. Le cas d'utilisation « authentification »	26
4.2. Le cas d'utilisation « gestion des utilisateurs »	28
4.3. Le cas d'utilisation « visualisation »	29
4.4. Le cas d'utilisation « consultation de l'historique et des statiques »	31
4.5. Le cas d'utilisation « mise à on ou à off des actionneurs »	32
4.6. Le d'utilisation « modifier le profil »	33
5. Conception	34
5.1. Règles de gestion	34
5.2. Description des attributs des classes	35
5.3. Diagramme de classe	36
5.4. Regles de passage	37
5.5. Le modèle relationnel	37
6. Conclusion	37
Chapitre 03 : Réalisation et implémentation	38
1. Introduction	38
2. Les outils de développement	38
2.1. Environnement matériel	38
2.2. Environnement logiciel	43
3. Description du système	47

Table des matières

3.1. Disposition des éléments du système	47
3.2. Architecture réseau	49
3.3. Schématisation électroniques du système	50
4. Description de l'application	53
4.1. Présentation du serveur virtuel	53
4.2. Arborescence de l'application	54
5. Présentation des interfaces	55
5.1. Interface d'authentification	55
5.2. Interface d'établissement de connexion	56
5.3. Interface d'accueil	57
5.4. Interface de contrôle de la lumière des chambres	59
5.5. Interface de contrôle des prises	61
5.6. Interface « plus d'information sur l'entité »	62
6. Conclusion	62
Conclusion générale	63
Webographie	64
Bibliographie	65

Liste des tableaux

1-1 Composants d'une solution IOT.....	09
2-1 Formalisme de description de cas d'utilisation.....	21
2-2 Description du cas d'utilisation « Authentification ».....	21
2-3 Description du cas d'utilisation « Gestion des utilisateurs ».....	22
2-4 Description du cas d'utilisation « Gestion des capteurs ».....	22
2-5 Description du cas d'utilisation « Gestion des actionneurs ».....	23
2-6 Description du cas d'utilisation « Visualisation ».....	23
2-7 Description du cas d'utilisation « consultation de l'historique et des statistiques ».....	23
2-8 Description du cas d'utilisation « Mise à ON ou à OFF des actionneurs ».....	24
2-9 Description du cas d'utilisation « modification de son profil ».....	24
2-10 Liste des attributs associés à chaque classe	35
3-1 Caractéristiques de la carte Arduino Mega 2560.....	39
3-2 Caractéristiques de la carte Arduino nano.....	40

Table des figures

1-1	Apparition de l'IOT selon Cisco IBSG	6
1-2	La lampe DAL (premier objet connecté)	7
1-3	Thermostat Qivivo	7
1-4	Système embarqué et ses dépendances	13
1-5	Microcontrôleur ATmega 328 sur une carte arduino	15
1-6	Connectique de la carte arduino Uno	16
1-7	Système embarqué et ses dépendances	17
2-1	Diagramme de cas d'utilisation du système	25
2-2	Diagramme de séquence du cas d'utilisation « Authentification »	26
2-3	Diagramme de séquence du cas d'utilisation « Gestion des utilisateurs »	29
2-4	Diagramme de séquence du cas d'utilisation « Visualisation »	30
2-5	Diagramme de séquence du cas d'utilisation « consultation de l'historique et des statistiques »	31
2-6	Diagramme de séquence du cas d'utilisation « Mise à ON ou à OFF des actionneurs »	32
2-7	Diagramme de séquence du cas d'utilisation « modification de profil »	33
2-8	Diagramme de classe du système	36
3-1	Carte Arduino Mega 2560	38
3-2	Ensemble des composants matériels impliqués dans ce projet	41
3-3	Environnement Arduino	45
3-4	Disposition des éléments du système	48
3-5	Architecture réseau	50
3-6	Schéma électrique de l'Arduino Mega et ses composants	51
3-7	Schéma électrique de l'Arduino Nano et ses composants	52
3-8	Description de l'application	53
3-9	Arborescence de l'application	54
3-10	Interface d'authentification	55
3-11	Interface d'établissement de connexion	56
3-12	Interface d'accueil pour l'administrateur	57
3-13	Interface d'accueil pour l'utilisateur simple	58
3-13	Interface de l'utilisateur simple pour le contrôle de la lumière	59
3-15	Interface de l'administrateur pour le contrôle de la lumière	60
3-16	Interface du simple utilisateur pour contrôler les prises	61
3-17	Interface « plus d'information » sur une entité	62

Liste des abréviations

API : Application Programming Interface

CSS : Cascading Style Sheets

CLI : Command line interface

DC : Direct Current

DHCP : Dynamic Host Configuration Protocol

EEPROM : Electrically-Erasable Programmable Read-Only Memory

FSF : Free Software Foundation

JSON : JavaScript Object Notation

IDE : Integrated Development Environment

IOS : iPhone Operating System

IOT : Internet Of Things

IP : Internet Protocol

I/O : Input/Output

GPS : Global Positioning System

GSM : Global System for Mobile

LAN : Local Area Network

LED : Light Emitting Diode

NDK : Native Development Kit

NFC : Near Field Communication

PGI : Progiciel de Gestion Intégré

PWM : Pulse width modulation

REST : REpresentational State Transfert

RFID : Radio frequency IDentification

RTS : Real Time Streaming

SASS : Syntactically Awesome StyleSheets

SDK : Software Development Kit

SMS : Short Message Service

Liste des abréviations

SOAP : Simple Object Access Protocol

SQL : Structured Query Language

SRAM : Static Random Access Memory

SSH : Secured Shell

TCP : Transmission control Protocol

UML : Unified Modeling Language

URI : Uniform Resource Identifier

URL : Uniform Resource Link

USB : Universal Serial Bus

XML : Extensible Markup Language

YAML : Yet Another Markup Language

Introduction générale

De nos jours les systèmes électronique et informatiques isolés (réduits à une zone géographique restreinte) tendent à disparaître, ils deviennent vite obsolètes et remplaçables par des objets connectés à internet. De ce fait est né l'Internet des objets, et ce phénomène ne fait que commencer.

Ces objets qualifiés de connectés sont des systèmes embarqués ou l'informatique et l'électronique s'entremêlent pour former un tout complexe pouvant être de taille d'une carte de crédit (exemple de la carte Arduino Uno pour des systèmes à base de microcontrôleur, et les cartes nano-computeur sous leurs divers nom pour des systèmes à base de microprocesseur).

Différents objets peuvent être impliqués dans l'internet des objets notamment les smartphones, grâce à une application installée sur ces derniers il est parfaitement possible d'effectuer différentes actions sur un autre objet connecté quel qu'il soit, où qu'il soit.

L'objectif de notre travail ; consiste alors à concevoir et à réaliser un système de maison connectée contrôlable via une application mobile multiplateforme en local ou par internet. Trois chapitres sont nécessaires pour aboutir à ce projet:

Le premier chapitre s'intitule « généralités », il a pour but de définir des concepts essentiels à l'internet des objets, aux applications mobiles, et aux maisons connectées.

Le second chapitre concerne « l'analyse et la conception » du projet. Dans ce chapitre, un établissement du diagramme de cas d'utilisation a été fait. Chaque cas d'utilisation a été détaillé par son diagramme de séquences. L'aboutissement de ce chapitre s'est fait avec un diagramme de classes associé à notre système et le modèle relationnel obtenu par l'application des règles de passage.

Dans le dernier chapitre, intitulé « réalisation et implémentation », les outils et environnements de travail ont été exposés. Le fonctionnement du système, ainsi que les interfaces de contrôle via l'application mobile font aussi partie de ce chapitre.

Ce présent mémoire se termine par une conclusion résumant les connaissances acquises au travers du projet réalisé, ainsi que des propositions pouvant venir améliorer notre système dans le futur.

Chapitre

01

Généralités

Systemes embarqués

Applications mobiles

Internet des objets

Maison connectée

Domotique

Arduino

...

1. Introduction

Actuellement en 2017 l'internet des objets est la tendance dans le monde de l'informatique, les medias parlent d'objets connectés, et les grands cabinets de consulting (Gartner et autres) avaient annoncé qu'il y aura plus d'objets connectés que d'êtres humains (ce qui est actuellement le cas). Ces objets sont hétérogènes, et remplissent des fonctionnalités diverses que nous verrons plus tard dans les pages qui suivent.

Ce chapitre est consacré à des généralités sur l'électronique embarquée, la domotique, les applications mobiles, l'internet des objets, et particulièrement les maisons connectées.

2. Application mobile

Une application mobile est un logiciel applicatif (exploitable directement par l'utilisateur) à des fins diverses, conçue pour être exécutée sur un terminal mobile (un téléphone, un smartphone, une tablette ...etc.) à des caractéristiques matérielles et logicielles bien définies. Souvent elles sont installables depuis une boutique (store) de téléchargement tel que l'App-store pour des téléphones sous ios, ou Google Play pour des téléphones sous Android, et cela de façon directe ou après le processus de paiement.

2.1. Évolution des applications (en nombre et en acquisition)

Les applications mobiles existent bien avant l'émergence des iPhones et des smartphones fonctionnants sous Android (Samsung, Sony, LG...etc.), il faut savoir que la première application mobile a été développée par Nokia au début des années 70, celle-ci était le fameux jeu du serpent (Snake). Du fait que Nokia et ses concurrents avaient des systèmes d'exploitation propriétaire, cela a induit au nombre restreint d'applications proposées, car autre que les développeurs internes aux entreprises personne ne pouvait concevoir des applications pour ce genre de téléphone.

Il a fallu attendre mi-2008 pour voir une croissance fulgurante des applications mobiles, le 10 juillet 2008 Apple ouvre sa plateforme de téléchargement App Store avec 500 applications téléchargeables, le 17 juillet de la même année le nombre de téléchargements a atteint les 10 millions, vers fin 2008 début 2009 le nombre d'applications dans la plateforme était de 10 000 et le nombre de téléchargements avait atteint le demi-milliard. Quant à Android Market (l'actuelle Play store), il ouvrit ses portes le 22 octobre 2008 avec cinquante applications en téléchargement, pour atteindre en 2010 les quatre milles applications dans la plateforme, et le 1 milliard de téléchargement en aout 2010. Actuellement il y a 1,5 milliard d'applications sur l'App-store, et 1,6 milliard d'application sur Google-Play, leurs nombres de téléchargements respectifs sont 25 milliards et 50 milliards [w1].

2.2. Objectifs, et prospective des applications mobiles

On recense 3 principaux objectifs pouvant être atteints grâce à une application mobile :

- Les applications mobiles visent d'abord à améliorer la productivité et à faciliter la récupération d'informations telles que le courrier électronique, calendrier électronique, contacts, marché boursier et informations météorologiques.
- Pour les professionnels ça permet de renforcer la relation et garder le lien avec ses interlocuteurs, d'atteindre une cible précise, et d'optimiser ses coûts.
- Disposer d'une solution métier: une application conçue dans cette perspective permet à l'utilisateur final de gérer ses stocks et commandes, et de piloter ou carrément d'automatiser ses interventions sur le terrain.

Actuellement elles nous permettent d'explorer internet, de jouer à des jeux, d'exploiter le GPS, l'infrarouge, et l'accéléromètre...etc. Leur potentiel est loin d'avoir été entièrement exploré et exploité ; de nombreuses innovations sont attendues, issues de la créativité des développeurs et des utilisateurs, ainsi que des évolutions technologiques. Les médias citent par exemple les applications liées aux Google Glass ou à des systèmes similaires, la réalité augmentée...etc [w2].

3. Types d'applications et environnements de développement

Autre que le classement des applications selon le service proposé (jeux, consultation de courrier électronique, écouter de la musique...etc.), il est aussi possible selon une perspective de développeur de distinguer les applications mobiles par leurs environnement d'exécution et les outils ayant contribué à leurs développement.

3.1. Les applications natives

Une application native est une application qui est développée spécifiquement pour une plateforme, grâce aux outils spécifiquement conçus pour la plateforme. Aussi, s'il est prévu de développer une application native pour Android et iOS, il convient en réalité de développer deux applications bien distinctes qui ne pourront malheureusement pas partager la moindre ligne de code.

Voici généralement les technologies utilisées lorsque l'on souhaite développer une application native :

- **Application sous Android**

Pour développer une application sur la plateforme de Google, le langage de programmation utilisé est le JAVA. Les applications sont alors développées à l'aide du SDK

Android. Suivant le besoin, on pourra également le mixer à l'utilisation du C++ et du NDK. L'emploi d'un IDE est aussi de mise, ça peut aussi bien être Eclipse ou Android studio. [B1]

- **Application sous iOS**

Pour développer une application sur la plateforme d'Apple, le langage de programmation utilisé historiquement est objective-c. Depuis quelques temps, il est également possible d'utiliser un nouveau langage : le Swift. X-Code est l'IDE par lequel jurent les développeurs d'iOS [B1].

- **Les applications sous Windows Phone**

Pour développer une application sur la plateforme de Microsoft, il convient d'utiliser un langage de la famille .NET ce qui offre aux développeurs plusieurs possibilités par exemple le VB.NET, le C# ou le C++. Il est également possible d'utiliser WinJS et ainsi développer une application à l'aide de technologies web. Développer une application Windows Phone ne fait pas exception à la règle de l'IDE. Aussi, il convient généralement d'utiliser Visual Studio [B1].

3.2. Les applications multiplateformes

Désignées aussi sous le nom d'applications hybrides. Les applications Cross-Platform s'opposent donc aux applications natives. Une application cross-Platform est une application qui une fois développée est capable de s'exécuter sur de nombreuses plates-formes.

Les applications cross-platforms sont souvent développées à l'aide de technologies web (html, css, javascript). Les frameworks les plus connus et utilisés étant probablement Phonegap, Ionic, Xamarin, et apache Cordova [B1].

L'avantage de ce type d'application est qu'un seul et unique code peut être développé sur différentes plateformes, ce qui permettra de réduire les couts de développement et d'homogénéiser l'expérience utilisateur.

4. Fonctionnement d'une application mobile

4.1. Architecture

A l'exception des jeux et des applications de bureautique diverses. Dans la plupart des cas les applications mobiles sont basées sur une architecture dite client-serveur. Deux actions sont récurrentes sur les terminaux mobiles ou ce genre d'applications est installé: afficher des informations et en recueillir. Au final une application mobile ne traite que très peu de donnée, elle délègue le travail à ce qu'on appelle un serveur.

Le serveur tient le rôle principal dans le cadre d'une application mobile. C'est lui qui alimente les applications en données et qui sauvegarde les données qui auraient besoin d'être persistées comme par exemple le compte d'un utilisateur (identifiant, mot de passe, données personnelles).

4.2. Communication client-serveur [B1]

La plupart du temps, les échanges entre un serveur et une application mobile se font à travers un web service, également appelé API. Bien que l'on distingue deux grandes familles –les web services REST et les web service SOAP– c'est bien les web services REST qui sont le plus souvent utilisés.

La communication est alors extrêmement simple. L'application mobile se contente de faire une requête HTTP au web service qui va alors s'exécuter, remplir son rôle (par exemple sauvegarder des données ou en récupérer dans une base de données) puis renvoyer une réponse à l'application mobile. Cette réponse, en plus de contenir les informations demandées contient souvent un certain nombre d'informations relatives au déroulement de la demande.

Les données renvoyées à l'application mobile sont formatées d'une manière très précise permettant à l'application de les traiter facilement. Souvent, les données sont formatées à l'aide du JSON car il a l'avantage d'être facile à écrire et léger à échanger à travers le web. Mais il est tout à fait possible d'utiliser une autre technologie come XML, ou encore YAML.

A noter que les web services peuvent être écrits dans n'importe quel langage de programmation capable de s'exécuter sur un serveur. Parmi les plus utilisés, on peut citer par exemple : PHP, JAVA, Python, Perl, Ruby, NodeJS, ou encore C#.

5. L'Internet des objets (IOT)

IOT est l'acronyme de « Internet Of Things », ou internet des objets en français. IOT est l'extension d'internet qui n'était qu'un monde virtuel et qui n'interagissait pas ou peu avec le monde physique à des entités et des emplacements existants sur terre. Les données générées par ces entités (objets) sont échangées via internet afin d'être exploitées dans divers domaine ; la santé, la domotique, l'agriculture...etc. Les géants de l'informatique parlent de maisons connectées, de villes intelligentes, et de véhicules autonomes.

Pour Cisco Internet Business Solutions Group (IBSG), l'IOT correspond simplement au moment où il y a plus de « choses ou d'objets » connectées à internet que de personnes (figure 1-1).

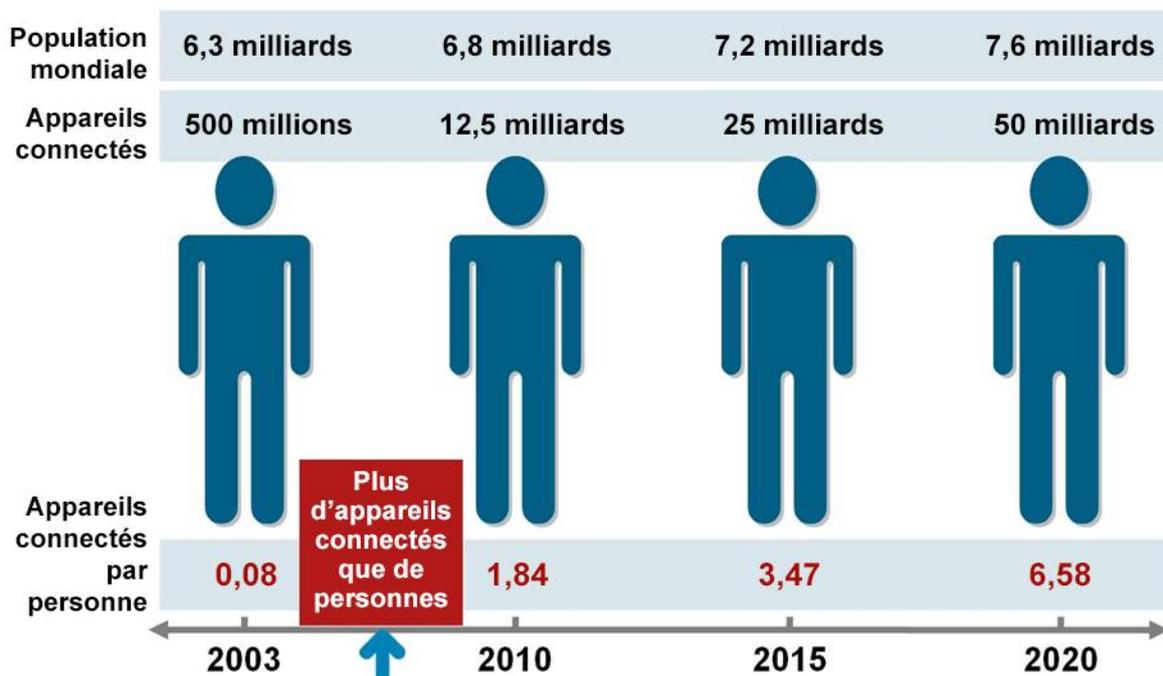


Figure 1-1 Apparition de l'IOT selon Cisco IBSG. [w3]

En raison de l'explosion des smartphones et des tablettes le nombre d'appareils connectés à internet a atteint 12.5 milliards en 2010, alors que la population mondiale était de 6.8 milliards. C'est ainsi que le nombre d'appareils connectés par personne est devenu supérieur à 1 pour la première fois dans l'histoire [w3].

5.1. objet connecté [B2]

Un objet connecté est un objet électronique relié à internet et capable de communiquer des informations, apportant ainsi un service ou une valeur ajoutée. Le premier objet connecté était la lampe DAL (figure 1-2), lancé en 2003 par Rafi Haladjan. Sensible au toucher et au bruit, cette lampe communiquait des informations sur la météo, la bourse, la pollution, des alertes Google et même des messages grâce à neuf LED de couleur.

Les fonctions proposées aujourd'hui vont beaucoup plus loin que la simple annonce de la météo. Prenons l'exemple du thermostat Qivivo (figure 1-3), qui permet non seulement de piloter son chauffage à distance, mais également d'obtenir un diagnostic de sa consommation d'énergie, des conseils d'optimisation et même d'être mis en relation avec des professionnels afin d'étudier les possibilités d'amélioration de son logement.

Les usages ont donc su se développer et, aujourd'hui, les objets connectés sont partout. Ils sont particulièrement appropriés dans certains domaines.



Figure 1-2 La lampe DAL (premier objet connecté).



Figure 1-3 Thermostat Qivivo.

5.2. Domaines d'application de l'internet des objets

On peut distinguer différentes catégories d'applications :

5.2.1. Transport et logistique

Voitures, trains, bus et vélos se voient de plus en plus dotés de capteurs, actuateurs et d'une logique de traitement des informations. Les routes aussi peuvent être munies de capteurs et tags (étiquettes) qui envoient des informations sur la circulation aux stations de contrôles mais aussi directement aux voyageurs pour mieux gérer le trafic, améliorer la sécurité routière et guider les touristes.

5.2.2. Soins et santé

Les objets connectés permettent de suivre et identifier en temps réel et à la demande, les outils, équipement et médicaments. Pour avoir des informations instantanément sur un patient peut souvent être déterminant.

5.2.3. Environnements intelligents

Capteurs et actuateurs distribués dans plusieurs maisons et bureaux peuvent augmenter le confort dans ces environnements : le chauffage peut s'adapter à la météo, l'éclairage suivant l'horaire et la position du soleil ; des incidents domestiques peuvent être évités avec des alarmes et beaucoup d'énergie pourrait être économisée.

Les environnements intelligents peuvent aussi améliorer l'automatisation en milieu industriel avec un déploiement massif de tags RFID associés aux différentes étapes de la

production. La ville intelligente est un exemple d'environnement intelligent. Le quartier d'affaires de Songdo en Corée du Sud est la première ville intelligente opérationnelle.

5.2.4. Signalement

Dans ce domaine on trouve des systèmes permettant à un opérateur ou à un utilisateur de signaler un dysfonctionnement.

Dans une copropriété ou immeuble, un boîtier permettant de signaler à la régie gestionnaire un dysfonctionnement de l'ascenseur, un problème de propreté ou la nécessité de tondre la pelouse, utilisera les technologies de l'IOT pour sa communication.

5.3. Composants d'un système IOT

L'internet des objets n'est pas une technologie à part-entière mais plutôt un système intégrant plusieurs autres systèmes. Lier un objet ou un lieu à Internet est un processus plus complexe que la liaison de deux pages Web. Divers composant sont de mise, L'IOT en exige sept : [w4]

1. Une étiquette physique ou virtuelle pour identifier les objets et les lieux.
2. Un moyen de lire les étiquettes physiques, ou de localiser les étiquettes virtuelles.
3. Un dispositif mobile (smartphone, tablette, ordinateur portable).
4. Un logiciel additionnel pour le dispositif mobile.
5. Un réseau sans fil de type 2G, 3G ou 4G.
6. L'information sur chaque objet lié.
7. Un affichage pour regarder l'information sur l'objet lié.

Le tableau suivant résume les principaux systèmes technologiques nécessaires à l'implantation d'une solution IOT.

Type de système	Enjeux	Technologies employées
Identification	Reconnaître chaque objet de façon unique et recueillir les données stockées au niveau de l'objet.	Code-barres, URI, GPS, radio-identification(RFID), ADN...etc.
Capteurs	Recueillir des informations présentes dans l'environnement pour enrichir les fonctionnalités du dispositif.	Luxmètre, capteur de proximité, thermomètre, hydromètre, accéléromètre, gyroscope...etc.
Connexion	Connecter les systèmes entre eux.	Câbles, fréquences radio, Bluetooth, Wi-Fi, ZigBee, Z-Wave, NFC...etc.
Intégration	Intégrer les systèmes pour que les données soient transmises d'une couche à l'autre.	Middleware (simple et complexe), analyse décisionnelle des systèmes complexe.
Traitement de données	Stocker et analyser les données pour lancer des actions ou pour aider à la prise de décisions.	Tableur, base de données, entrepôt de données, progiciel de gestion intégré(PGI).
Réseaux	Transférer les données dans les mondes physiques et virtuels.	Internet

Tableau 1-1 Composants d'une solution IOT

5.4. Contraintes liées à l'IOT

Les implications sociales et cognitives de ces applications et de celles à venir de l'IOT posent un certain nombre de problèmes quant à la protection de la vie privée, voire des façons de se comporter dans un environnement. L'interconnexion d'objets transmettant continuellement de l'information sur les personnes pourrait ainsi, selon certaines critiques, marquer la disparition complète du contrôle des individus sur les données les concernant [W5].

Il y a aussi la difficulté de trouver une norme de communication qui consomme peu de courant, pour augmenter l'autonomie de la batterie, et peu de donnée pour limiter la consommation de forfait Internet GSM [w6].

Des problèmes environnementaux peuvent aussi être soulevés, l'électronique moderne utilise une large variété de métaux lourds et de métaux rares, aussi bien que de produits chimiques synthétiques fortement toxiques, ce qui les rend difficilement recyclables [w6].

6. Le Web of Things

Le web of things (Web 3.0) est né après de nombreuses itérations du web, c'est un sous-jacent de l'Internet Of Things. Il est basé sur des technologies web tel que le HTTP pour le transfert de données, html5 pour la structuration, CSS3 pour l'interfaçage, et JavaScript pour les interactions. Un objet physique est alors vu comme un ensemble de services accessibles au travers du web (navigateur web ou application mobile hybride).

7. Domotique et Maison connectée

7.1. Domotique

Le mot domotique vient de domus qui signifie «domicile » et du suffixe -tique qui fait référence à la technique. la domotique est l'ensemble des techniques de l'électronique, de physique du bâtiment, d'automatisme, de l'informatique et des télécommunications utilisées dans le bâtiment, plus ou moins « interopérables » et permettant de centraliser le contrôle des différents systèmes et sous-systèmes de la maison et de l'entreprise (chauffage, volets roulants, porte de garage, portail d'entrée, prise électrique, etc.). la domotique vise à apporter des solutions techniques pour répondre aux besoins de confort (gestion d'énergie, optimisation de l'éclairage et du chauffage), de sécurité (alarme) et de communication (commandes à distance, signaux visuels ou sonores, etc.) que l'on peut trouver dans les maisons, les hôtels, les lieux publics...etc [B2].

A l'origine, la domotique avait donc pour but d'automatiser sa maison : ouverture et fermeture automatiques des volets, ouverture du portail électrique, gestion du chauffage, gestion de l'éclairage, etc. Ainsi avant l'ère des smartphones, il était par exemple possible d'activer son chauffage à distance en passant un coup de téléphone à sa maison, ou encore en lui envoyant un SMS. C'était tout à fait réalisable. Seulement une telle installation était relativement compliquée à mettre en place et, il faut bien l'avouer, coûteuse. Cette époque a malheureusement laissé des traces, puisque pour beaucoup encore aujourd'hui, domotique rime avec cher et compliqué. Pourtant, ce domaine a énormément évolué et il existe de

nombreuses solutions simples à mettre en place et tout à fait abordables pour le grand public [B2].

La domotique a surtout elle-même évolué, si bien que le terme est quelque peu dépassé. La domotique servait à automatiser sa maison ; aujourd'hui on parle de domotique 2.0, ou de « maison intelligente », pour bien marquer l'évolution de ce monde. Les différents domaines de la maison ne se contentent plus d'être automatisés et pilotables, ils communiquent ensemble, permettant à la maison de réagir selon différents événements [B2].

7.2. Maison connectée

C'est une maison dotée de capteurs en tous genres, quasiment invisibles mais permettant la connexion avec votre smartphone ou votre tablette via internet. Ces capteurs vous permettront de réaliser des économies sans contraintes et sans perte de temps. Grâce à votre smartphone, on peut recevoir des alertes lorsque des intrus tentent de faire une intrusion, ou lorsqu'une personne âgée maintenue à domicile est en situation de détresse. Gérez les accès internet, l'enregistrement de vos émissions tv, la température de la maison, les programmes des machines à laver sans modération, où que vous soyez et en toute simplicité.

7.3. Différence entre domotique et maison connectée

La domotique facilite le quotidien mais avec l'intervention de l'Homme (appuyer sur des boutons pour fermer les volets roulants, gérer votre climatiseur avec la télécommande.... etc.). La maison intelligente connaîtra déjà certaines de vos habitudes et s'adaptera pour vous, on parle alors de « machine learning » ou d'intelligence artificielle.

7.4. Exemples d'usage dans une maison connectée

La maison connectée trouve sa place dans trois domaines principaux en particulier :

- **Le confort**

Aujourd'hui une maison intelligente sait quand une personne rentre à la maison (grâce à son smartphone) et ouvre le portail avant même qu'elle n'arrive.

- **Les économies d'énergie**

En gérant le chauffage et les volets selon la saison, la maison intelligente vous fait économiser de l'énergie et donc de l'argent. Le suivi peut être très fin, qu'il s'agisse de la consommation d'électricité, d'eau, ou même de gaz.

- **La sécurité**

Les smart house peuvent contribuer à la sécurité de vos biens, en réalisant ce qu'on appelle une simulation de présence. Une détection de fuite peut aussi couper automatiquement l'arrivée d'eau afin d'éviter de gros dégâts.

Cependant, la sécurité concerne également les personnes : en cas de détection d'incendie, par exemple, il est tout à fait possible d'ouvrir automatiquement les volets, déverrouiller les portes et éclairer les chemins de sortie pour faciliter l'évacuation.

8. Techniques de mise en place d'une maison connectée [B2]

A quelques exceptions près, un système de maison intelligente est à peu près toujours constitué des mêmes équipements, quelle que soit la technologie utilisée.

- Un cerveau, que ce soit un automate, un ordinateur, ou une « box domotique ». c'est lui qui centralise toutes les informations de votre maison et déclenche des actions.
- Les capteurs. Un cerveau sans vue, sans ouïe, sans odorat ne sert pas à grand-chose. Les capteurs sont des périphériques qui donnent des sens à votre maison : relever la température, l'humidité, la luminosité, le niveau de CO2, le niveau de bruit, détecter une présence, de la fumée, une fuite de gaz, etc. Grâce à eux la maison saura tout ce qui se passe.
- Enfin, les actionneurs. Savoir ce qui se passe, c'est bien, mais pouvoir réaliser des actions en conséquence, c'est mieux ! Les actionneurs sont donc des périphériques qui pilotent des appareils (radiateurs, chaudière, télévision, machine à laver...etc.), des lampes, ou encore des automatismes (volets, porte de garage, store banne, etc.).

Le cerveau déclenche des actions en fonction des informations recueillies par les différents capteurs disséminés à travers la maison. Par exemple si aucun détecteur ne détecte de présence dans la maison, le cerveau demande aux radiateurs de passer en mode économique et aux lampes de s'éteindre.

Techniques de liaison

Les solutions filaires relient le cerveau, ses capteurs et ses actionneurs par des câbles à travers toute la maison. Autant dire que cela est difficilement envisageable pour beaucoup de monde, car le passage des câbles dans une habitation existante est une vraie contrainte. Ce type de solution s'adressera plutôt à des constructions neuves ou de grosses rénovations.

Il existe aujourd'hui de nombreuses solutions sans fil, tout aussi fiables, mais nettement plus simple à mettre en œuvre implémentant l'un des protocoles suivants : Z-Wave, Zigbee, RTS, EnOcean, et le Io-Homecontrol en sont les principaux exemples.

9. Solutions pour implanter une maison connectée

Dans cette partie nous exposerons différentes solutions de domotique pour implanter une maison connectée. Mais avant ça, quelques définitions s'imposent.

9.1. L'open hardware

Le matériel libre (en anglais open hardware ou hackable devices) désigne les technologies et produits physiques développés selon les principes des « ressources libres » (open source) [w7].

9.2. L'open source

La notion de logiciel libre a été inventée par la *Free Software Foundation* (FSF) fondée par Richard Stallmann en 1982 [w8].

Elle signifie que les utilisateurs d'un logiciel sont libres

- d'en étudier le fonctionnement, de l'améliorer, le modifier et de publier ces améliorations
- de redistribuer des copies, gratuitement ou non : toute entreprise ou particulier est donc autorisé à les commercialiser.

9.3. Système embarqué

C'est un système électronique et informatique autonome dédié à une tâche précise, souvent en temps réel, possédant une taille limitée et ayant une consommation énergétique restreinte. Il travaille en étroite collaboration avec deux entités : un capteur et un actuateur (figure 1-4) [w7].

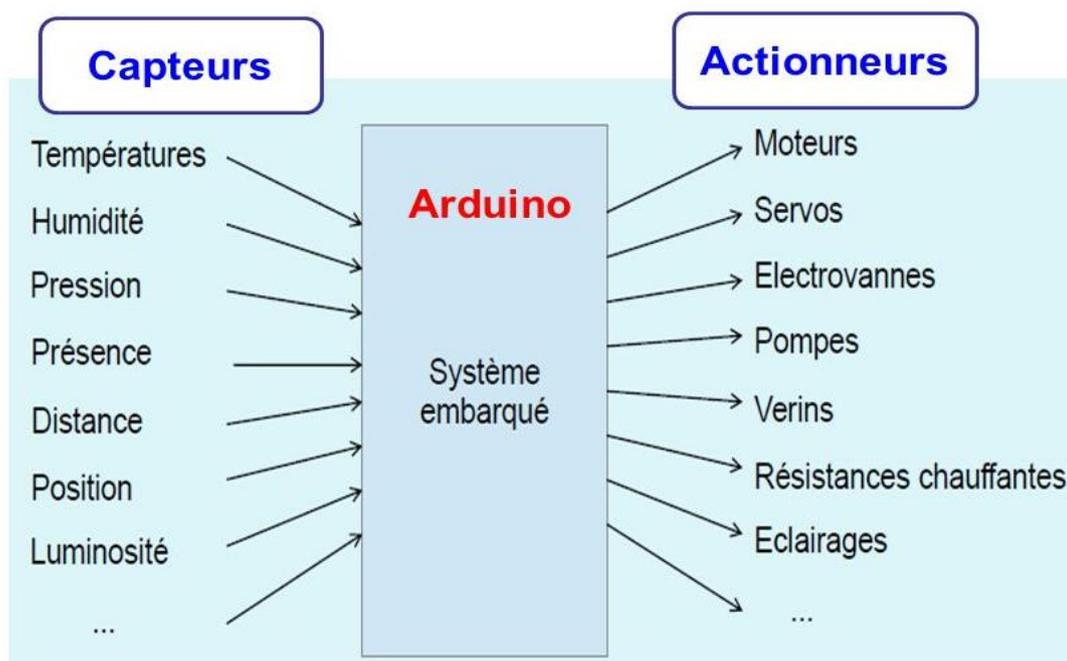


Figure 1-4 Système embarqué et ses dépendances

Pour concevoir un système embarqué, il faut généralement combiner des compétences en électronique, en informatique industrielle et en automatique. De tels systèmes sont nombreux dans des secteurs aussi variés que l'aéronautique, l'électroménager, le matériel médical, la téléphonie mobile, etc [w9].

9.4. Capteur

Les capteurs permettent de transformer des variables non-électriques en variables électriques. Selon leur mode de fonctionnement, on distingue les capteurs actifs et les capteurs passifs. La définition de ces deux qualificatifs n'est pas clairement définie et fait l'objet de discussion entre experts [B3]

- Les capteurs actifs sont des capteurs alimentés par une tension, qui contiennent des éléments d'amplification ou qui génèrent un signal. Le signal sort, par l'électronique intégrée dans les capteurs, sous forme de tension digitale.
- Les capteurs passifs sont des capteurs qui ne contiennent que des éléments passifs (bobine, résistance, condensateur). Le plus souvent les signaux sortent sous forme de tension analogique.

9.5. Actuateur (actionneur)

Dénommés de façon imagée les «muscles de la microélectronique». Ces derniers transforment les instructions électriques-numériques ou analogiques de l'appareil de commande en énergie mécanique. La transformation de l'énergie est réalisée par moteur, de façon pneumatique, hydraulique, magnétique [B3].

9.6. Solution à base de microcontrôleur (carte Arduino)

L'Arduino est une plateforme open source d'électronique programmée qui est basée sur une carte à microcontrôleur et un logiciel. Plus simplement, on peut dire que l'Arduino est un module électronique, doté d'un microcontrôleur programmable en langage C. Il peut être utilisé pour développer des objets interactifs, munis d'interrupteurs ou de capteurs, et peut contrôler une grande variété de lumières, moteurs ou toutes autres sorties matérielles.

Arduino est basé sur le concept de l'open hardware, soit du matériel libre de droits. Ainsi, toute la plateforme matérielle et logicielle est libre de droits. N'importe qui peut donc développer et commercialiser un clone d'Arduino. Néanmoins, le nom et le logo Arduino sont des marques déposées, afin d'identifier le matériel officiel. Il y a donc cinq catégories de matériel: le matériel officiel, les clones, les dérivés, les compatibles et les contrefaçons.

Le principe de fonctionnement est simple:

1. Réaliser le programme sur un ordinateur (grâce à l'ARDUINO IDE).
2. Connecter l'ordinateur à l'Arduino via une prise USB.
3. Envoyer le programme sur l'Arduino.
4. L'Arduino exécute enfin le programme de manière autonome.

9.6.1. Constitution de la carte

a) Le microcontrôleur

C'est le cerveau de la carte (figure 1-5). Il va recevoir le programme que nous allons créer et va le stocker dans sa mémoire avant de l'exécuter. Grâce à ce programme, il va savoir faire des choses, qui peuvent être : faire clignoter une LED, afficher des caractères sur un écran, envoyer des données à un ordinateur...etc [w10].

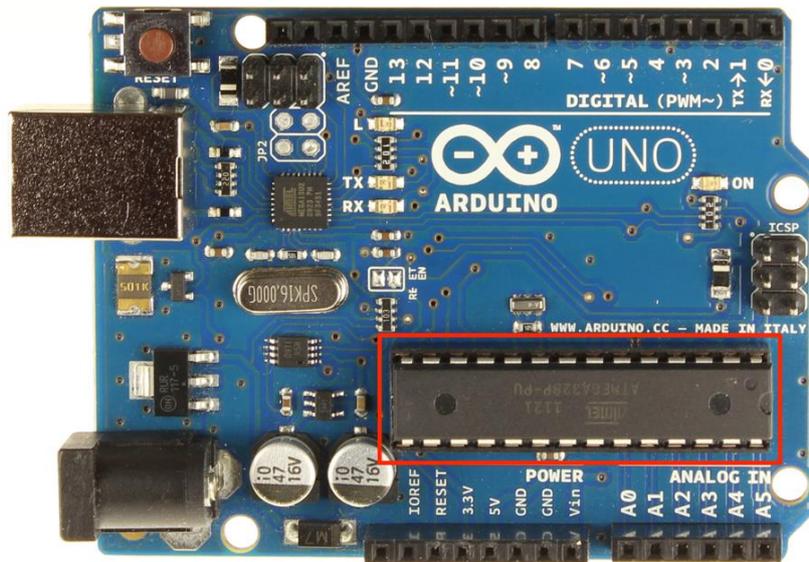


Figure 1-5 Microcontrôleur ATmega 328 sur une carte arduino.

b) L'alimentation

Pour fonctionner, une carte Arduino a besoin d'une alimentation. Le microcontrôleur fonctionnant sous 5V, la carte peut être alimentée en 5V par le port USB ou bien par une alimentation externe qui est comprise entre 7V et 12V. Un régulateur se charge ensuite de réduire la tension à 5V pour le bon fonctionnement de la carte [w10].

c) La connectique

La carte Arduino ne possédant pas de composants (résistances, diodes, moteurs...) qui peuvent être utilisés pour un programme, il est nécessaire de les rajouter. Mais pour ce faire, il faut les connecter à la carte. C'est là qu'interviennent les connecteurs de la carte (figure 1-6). Sur les Arduino et sur beaucoup de cartes compatibles Arduino, les connecteurs se

trouvent au même endroit. Cela permet de fixer des cartes d'extension, appelée shield en les empilant [w8].

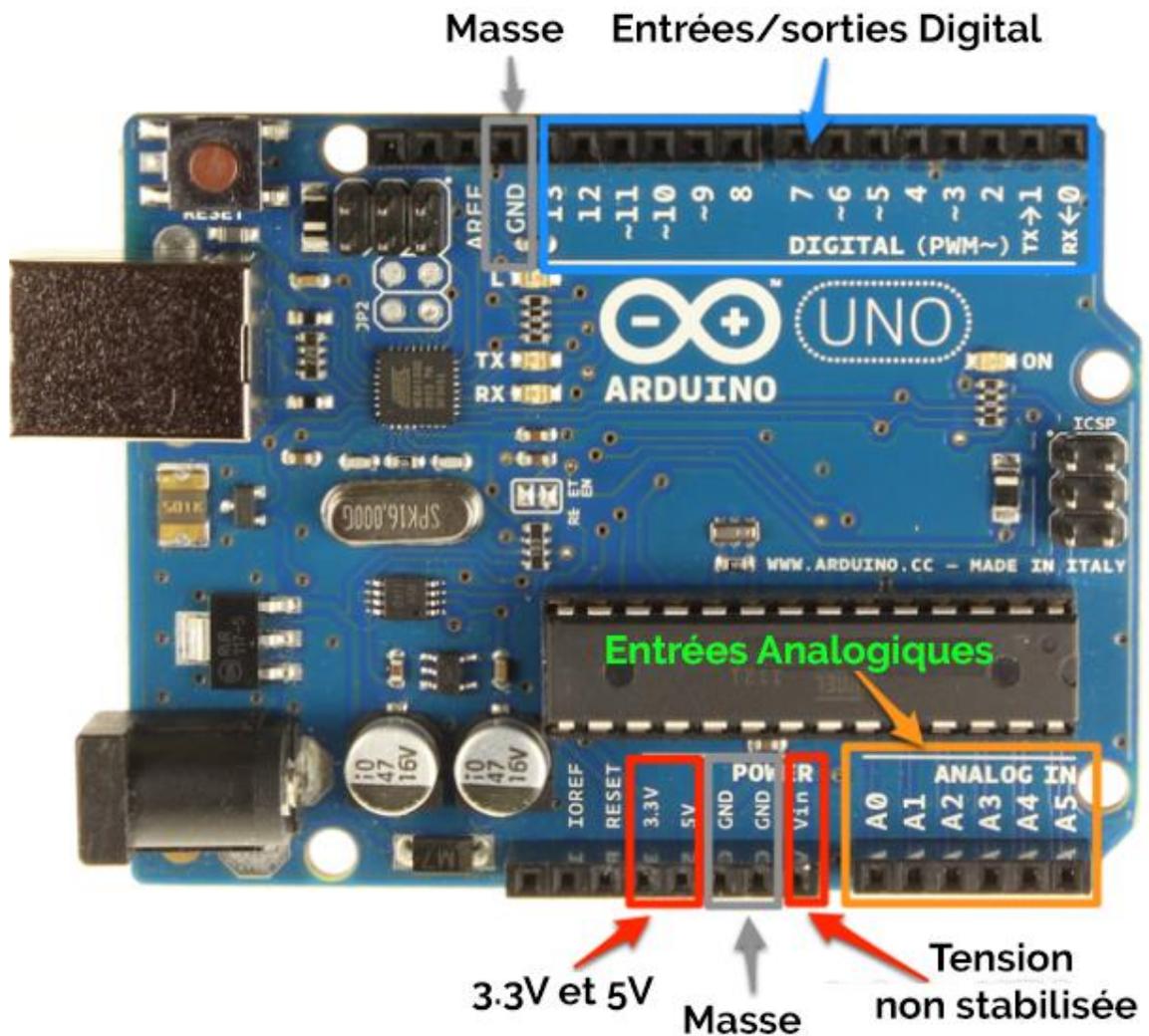


Figure 1-6 Connectique de la carte arduino Uno

9.7. Solutions mixte (microcontrôleur + microprocesseur)

L'Arduino Yun (figure 1-7) est une carte Arduino comme n'importe quelle autre. Il utilise le même processeur, l'Atmel ATmega32U4, il comporte aussi un processeur additionnel, un Atheros AR9331, sur lequel tourne la distribution Linux OpenWrt wireless stack. La programmation du 32U4 par le biais de l'interface USB.

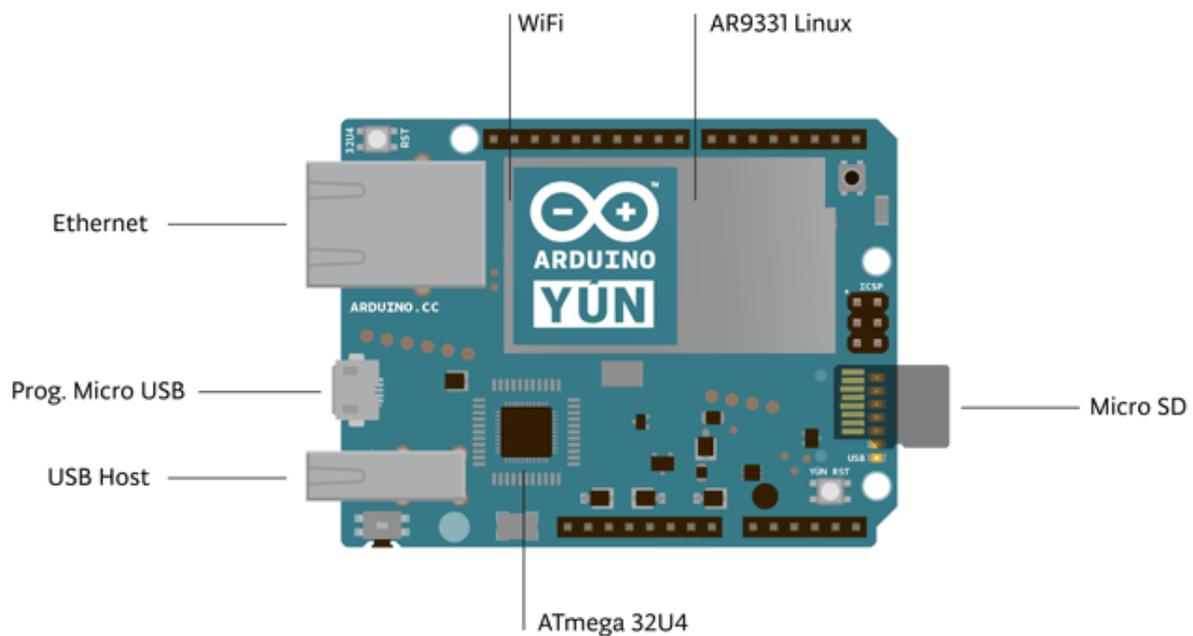


Figure 1-7 Connectique de la carte arduino Uno

9.7.1. Différences avec l'Arduino Uno [w11]

En plus du processeur 32U4, le Yún dispose d'un processeur additionnel, un Atheros AR9331, sur lequel tourne la distribution Linux pour systèmes embarqués appelée Linio basée sur OpenWrt wireless stack. Une installation complète de Python 2.7 est également incluse.

Le Yun a un nombre de caractéristiques physiques et de connecteurs différents du Uno. Il y a un connecteur pour cartes SD intégré, Une prise Ethernet RJ45 et un connecteur hôte USB de type A. Il n'y a pas d'embase d'alimentation sur le Yun ; il peut être alimenté par le biais du connecteur micro-USB de type B.

Le Yun a aussi un module wifi intégré à sa carte, ce qui lui permet de se connecter à un routeur sans fil, ou se comporter comme un point d'accès.

10. Conclusion

Tout au long de ce chapitre nous avons traité des applications mobiles, leurs types et environnements d'exécution. Ensuite nous avons introduit l'internet des objets, les objets connectés ainsi que les entités impliquées dans un système IOT. Enfin nous avons conclu ce présent chapitre avec les maisons connectées et les systèmes open-hardware permettant de les implanter.

Dans le chapitre suivant nous nous focaliserons sur l'analyse et la conception du système de maison connectée, ceci après une vue générale du projet.

Chapitre

02

Analyse et conception

Diagrammes de séquences

Modélisation des besoins

Présentation du projet

Diagramme de classe

...

1. Introduction

L'analyse et la conception sont les deux étapes primordiales pour mener à bien un projet informatique. Ils permettent de soulever des problèmes dans les brefs délais et de reconstituer le système de façon à être conforme aux règles établies.

Nous commencerons ce présent chapitre par une présentation générale de notre projet (problématique, objectif, fonction ...etc.), une fois ceci est mis en évidence, nous passerons à la modélisation du système illustré par le langage de modélisation UML.

2. Présentation du projet

2.1. Problématique

La majorité des cerveaux systèmes de maisons connectées appelés aussi box-domotiques sont propriétaires. Ils sont réservés à leur constructeurs, de ce fait découle un tas de problèmes, parmi eux nous pouvons citer :

- Incompatibilité entre protocoles propriétaires.
- Le cout d'un système domotique complet est onéreux.
- Choix limité au niveau des capteurs.
- Les box-domotiques sur le cloud, disparaissent avec leurs constructeurs.
- Les interfaces de contrôle ne sont pas ergonomiques.

2.2. Objectif

L'objectif dans ce projet est de créer un système de maison connectée en exploitant la technologie open-hardware Arduino, qui est sans doute le moyen le plus bas prix existant sur le marché.

Ce système sera accessible et contrôlable en temps réel via une application mobile cross-Platform, et extensible par des actionneurs et des capteurs que l'administrateur peut ajouter et configurer en rapport avec ses besoins.

2.3. Fonctions du système

- Accès à distance : l'utilisateur du système une fois identifié, pourra avoir accès au système depuis internet.
- Ajouter des capteurs et des actionneurs: l'administrateur du système peut étendre les fonctionnalités de la maison connectée avec de nouveaux capteurs et actionneurs.
- Actionner des actuateurs : si un objet électrique est relié à une prise secteur et que cette dernière est équipée d'un relai qui lui-même est relié au système Arduino alors il devient tout à fait possible d'allumer et d'éteindre cet actuateur.

- Visualisation des métriques et des états : il sera possible soit par internet ou en réseau local de consulter les valeurs transmises par les différents capteurs (humidité, gaz, niveau d'eau...etc.) et les états de chaque actionneur, tout cela via l'application mobile.
- Voir l'historique et les statistiques d'une entité : une entité peut désigner soit un actionneur, soit un capteur. Il est possible de remonter dans l'historique des valeurs et des états captés de chaque entité grâce au serveur qui interroge la carte Arduino à temps régulier.
- Attribuer des droits aux simples utilisateurs : cette fonction est réservée à l'administrateur du système pour décider de qui a droit à quoi.

3. Modélisation des besoins

3.1. Identification des acteurs

Un acteur représente une entité jouant un rôle défini par une ou plusieurs fonctions, cette entité est externe au système mais elle est en interaction avec celui-ci [B4].

Deux acteurs peuvent intervenir au sein du système :

- L'administrateur : Acteur possédant le contrôle total du fonctionnement du système. C'est lui qui décide des usagers du système et de leurs droits. Le contrôle des entités du système est aussi en sa possession.
- Utilisateur: Il a pour rôle de contrôler des actionneurs auxquels il a droit, et de consulter les métriques captées par les différentes entités internes au système.

3.2. Description des cas d'utilisation

La description des cas d'utilisation permet de clarifier le déroulement de la fonctionnalité, et de décrire la chronologie des actions qui devront être réalisées.

Des règles de structuration doivent être appliquées pour faciliter l'expression, la compréhension et la cohérence, pour illustrer les différents cas d'utilisation de notre projet, nous avons opté pour le formalisme suivant [B4] :

Numéro du cas d'utilisation	Nom du cas d'utilisation.
Résumé	But du cas d'utilisation.
Acteurs	Acteurs participant au cas d'utilisation.
Scénario	Séquence d'actions normales associées au cas d'utilisation.
Exception	Séquence d'actions conduisant à un échec.

Tableau 2-1 Formalisme de description de cas d'utilisation

3.2.1. Cas d'utilisation « authentification »

Cas d'utilisation N° 01	Authentification
Résumé	Donner l'accès pour exploiter le système.
Acteurs	utilisateur
Scénario N°01	<p>[Début]</p> <ol style="list-style-type: none"> 1. l'utilisateur fait son choix entre une connexion local ou via internet ; 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 mobile ; 7. L'application mobile affiche l'interface d'accueil adéquate ; <p>[Fin]</p>
Exception	L'utilisateur remplit mal le formulaire (des espaces dans son identifiant, champ non rempli).

Tableau 2-2 Description du cas d'utilisation « Authentification »

3.2.2. Cas d'utilisation « gestion des utilisateurs »

Cas d'utilisation N° 02	Gestion des utilisateurs.
Résumé	Ajouter, modifier ou supprimer un utilisateur.
Acteurs	Administrateur
Scénario N°02	<p>[Début]</p> <ol style="list-style-type: none"> 1. L'administrateur passe à la section gestion des comptes utilisateurs; 2. L'administrateur clique soit sur l'icône ajouter, modifier, ou supprimer un compte utilisateur ;

	<ol style="list-style-type: none"> 3. L'application affiche le formulaire correspondant ; 4. L'administrateur effectue l'opération adéquate ; 5. Le système renvoie un message; <p>[Fin]</p>
Exception	L'utilisateur remplit mal le formulaire (des espaces dans l'identifiant de l'utilisateur, champ non rempli, caractères spéciaux).

Tableau 2-3 Description du cas d'utilisation « Gestion des utilisateurs »

3.2.3. Cas d'utilisation « gestion des capteurs »

Cas d'utilisation N° 03	Gestion des capteurs.
Résumé	Ajouter, modifier ou supprimer un capteur.
Acteurs	Administrateur.
Scénario N°03	<p>[Début]</p> <ol style="list-style-type: none"> 1. L'administrateur passe à la section capteurs; 2. L'administrateur clique sur l'une des icônes ajouter, modifier, ou supprimer un capteur ; 3. L'application affiche le formulaire adéquat ; 4. L'administrateur effectue l'opération adéquate; 5. Le système renvoie un message; <p>[Fin]</p>
Exception	L'utilisateur remplit mal le formulaire (ajout, suppression, modification).

Tableau 2-4 Description du cas d'utilisation « Gestion des capteurs »

3.2.4. Cas d'utilisation « gestion des actionneurs »

Cas d'utilisation N° 04	Gestion des actionneurs.
Résumé	Ajouter, modifier ou supprimer un actionneur.
Acteurs	Administrateur
Scénario N°04	<p>[Début]</p> <ol style="list-style-type: none"> 1. L'administrateur passe à la section actionneurs; 2. L'administrateur clique sur l'une des icônes ajouter, modifier, ou supprimer un actionneur ; 3. L'application affiche le formulaire adéquat ; 4. L'administrateur effectue l'opération adéquate; 5. Le système renvoie un message;

	[Fin]
Exception	L'utilisateur remplit mal le formulaire (exemple : champ manquant)

Tableau 2-5 Description du cas d'utilisation « Gestion des actionneurs »

3.2.5. Cas d'utilisation « Visualisation »

Cas d'utilisation N° 05	Visualisation
Résumé	Consulter les valeurs des capteurs et les états des actionneurs.
Acteurs	Utilisateur
Scénario N° 05	[Début] <ol style="list-style-type: none"> 1. L'utilisateur clique soit sur l'icône de la section actionneurs, ou celle de la section capteurs; 2. L'application affiche la section appropriée; [Fin]
Exception	L'utilisateur oublie un champ dans le formulaire, ou introduit un caractère spécial.

Tableau 2-6 Description du cas d'utilisation « Visualisation »

3.2.6. Cas d'utilisation « Consultation de l'historique et des statistiques »

Cas d'utilisation N° 06	Consultation de l'historique et des statistiques d'une entité.
Résumé	Accéder à un capteur ou un actionneur et voir des détails le concernant.
Acteurs	Utilisateur
Scénario N° 06	[Début] <ol style="list-style-type: none"> 1. L'utilisateur passe à la section actionneurs, ou à la section capteurs; 2. Le système renvoie les informations appropriées à l'application ; 3. L'application s'occupe de l'affichage ; 4. L'utilisateur ou l'administrateur clique sur le bouton détails ; 5. L'application envoie une requête au système ; 6. Le système traite la requête et renvoie les données en question ; 7. L'application affiche les données ; [Fin]
Exception	Formulaire d'identification incomplet.

Tableau 2-7 Description du cas d'utilisation « consultation de l'historique et des statistiques »

3.2.7. Cas d'utilisation « Mise à ON ou à OFF des actionneurs»

Cas d'utilisation N° 07	Mise à ON ou à OFF des actionneurs.
Résumé	Accéder à un actionneur et le mettre à ON/OFF, mettre tous les actionneurs à ON/OFF
Acteurs	Utilisateur
Scénario N° 07	[Début] <ol style="list-style-type: none"> 1. L'utilisateur ou l'administrateur passe à la section actionneurs; 2. L'application affiche la section actionneurs; 3. L'utilisateur ou l'administrateur clique sur le bouton actionneur ; 4. Le système traite la requête et renvoie le résultat à l'application ; 5. L'application affiche le résultat; [Fin]
Alternative	Erreur lors de l'authentification.
Exception	Formulaire d'identification incomplet.

Tableau 2-8 Description du cas d'utilisation « mise à on ou à off des actionneurs »**3.2.8. Cas d'utilisation « modification de son profil »**

Cas d'utilisation N° 08	Modifier son profil.
Résumé	Accéder à son profil et le mettre à jour
Acteurs	Utilisateur
Scénario N° 08	[Début] <ol style="list-style-type: none"> 1. L'utilisateur passe à la section profil; 2. Le système renvoie les informations appropriées à l'application ; 3. L'application s'occupe de l'affichage ; 4. L'utilisateur modifie les informations le consternants et soumet le formulaire ; 5. L'application envoie la requête ; 6. Le système traite la requête et renvoie le résultat à l'application ; 7. L'application affiche le résultat; [Fin]
Exception	L'utilisateur oublie un champ dans le formulaire, ou introduit un caractère spécial non autorisé dans un champ donné.

Tableau 2-9 Description du cas d'utilisation « modification de son profil »

3.3. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est un diagramme UML utilisé pour donner une vision globale du comportement fonctionnel d'un système logiciel. Il permet de décrire l'interaction entre les acteurs et le système [B4]. Le diagramme auquel on a abouti dans ce projet est celui de la figure 2-1:

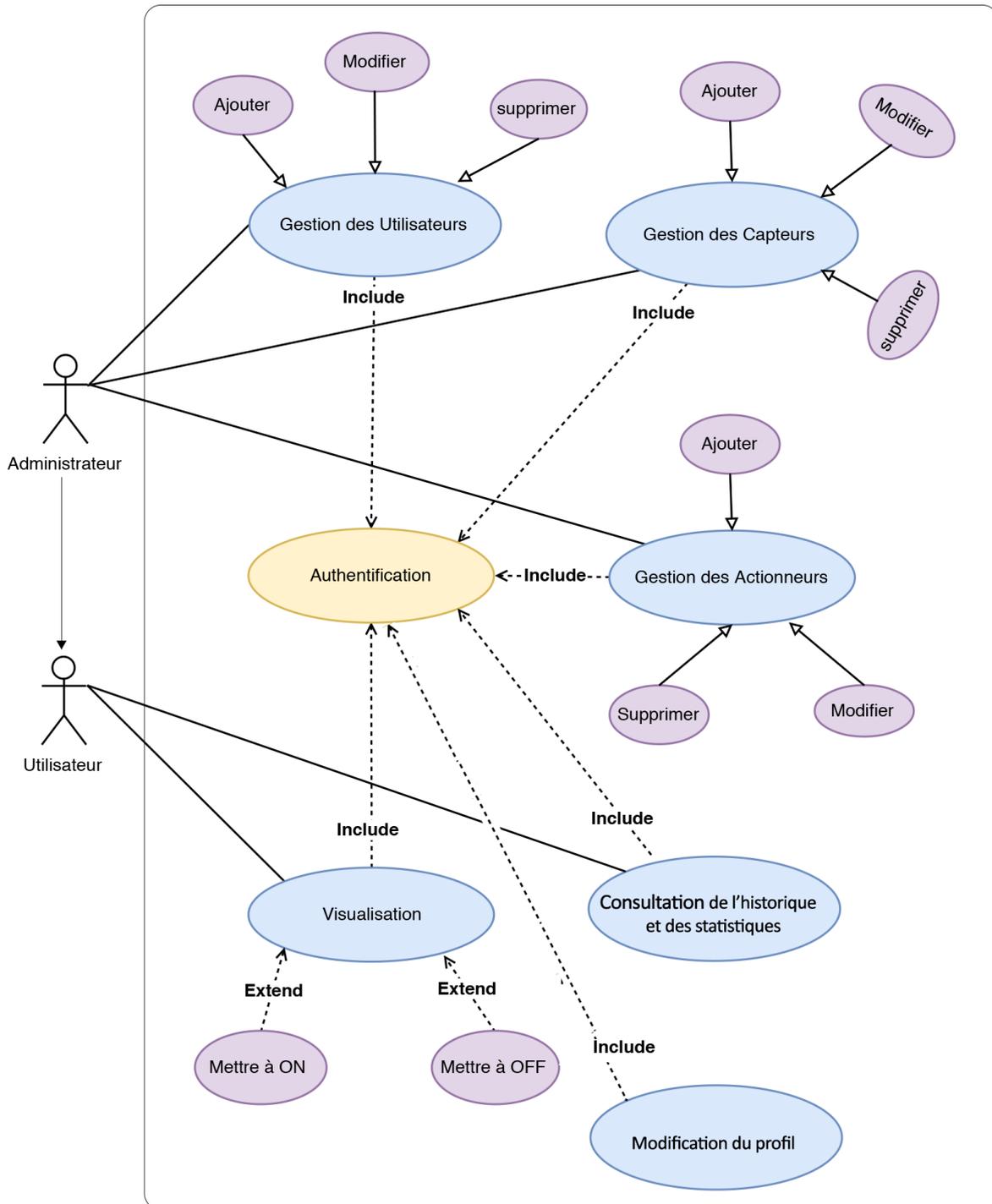


Figure 2-1 Diagramme de cas d'utilisation du système.

4. 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'échangeant des messages, les acteurs interagissent avec le système au moyen d'interface homme-machine [B4]. Chaque cas d'utilisation est distingué par un diagramme de séquences correspondant.

4.1. Diagramme de séquence du cas d'utilisation «Authentification»

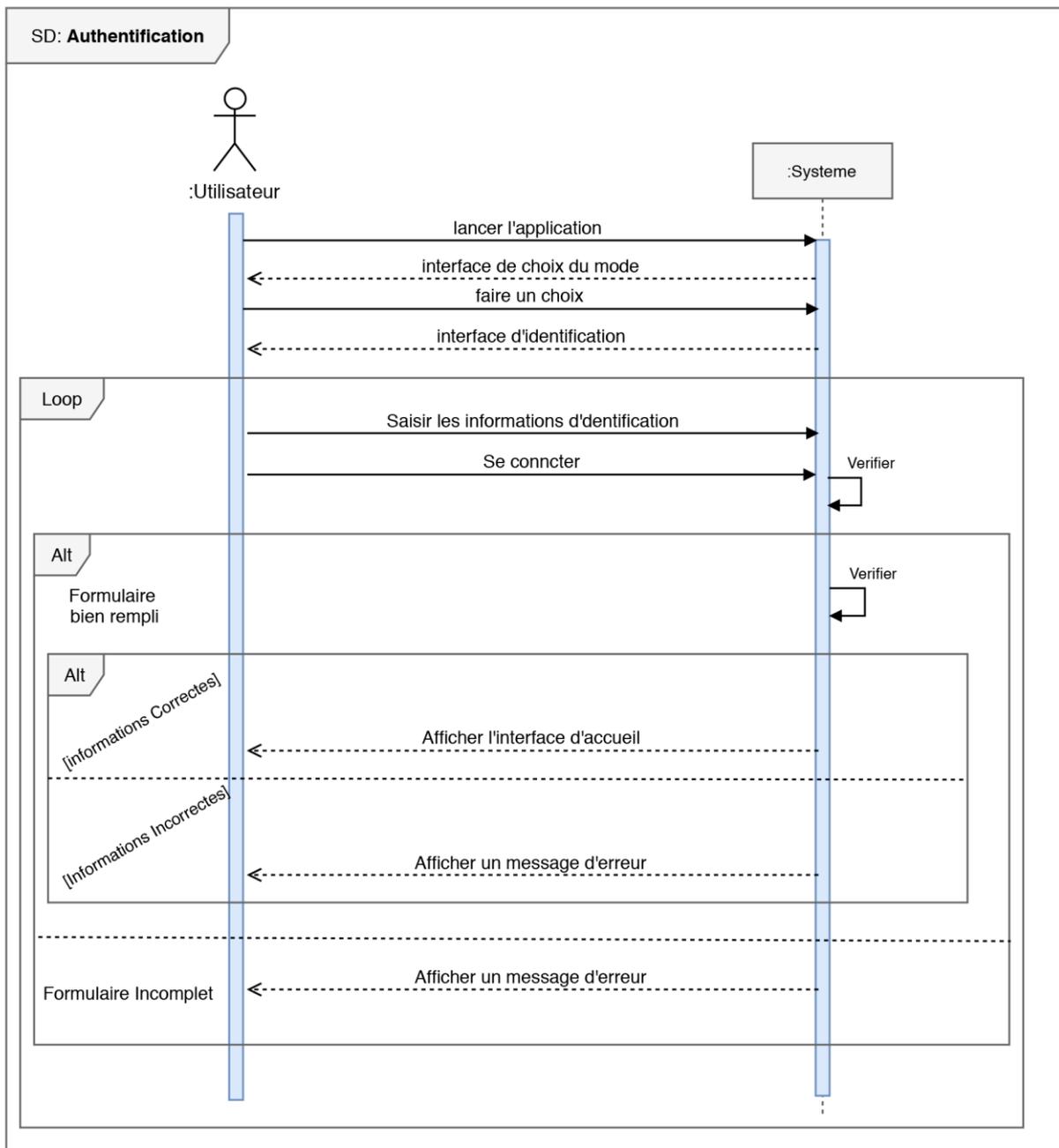


Figure 2-2 Diagramme de séquence du cas d'utilisation « Authentification »

Deux cas peuvent apparaître lors de l'authentification, ce qui justifie l'utilisation de l'opérateur «Alt», dans le cas où les informations présentées par l'utilisateur sont correctes le système accorde un accès à l'usage du système. Dans le second cas, le système refuse l'accès à l'utilisateur en lui affichant un message d'erreur d'authentification et l'invitant à réintroduire de nouveau ses identifiants.

Ce procédé est exécuté à chaque fois que l'utilisateur tente de s'authentifier, c'est pourquoi nous avons utilisé l'opérateur « loop ».

4.2. Diagramme de séquence du cas d'utilisation «Gestion des utilisateurs»

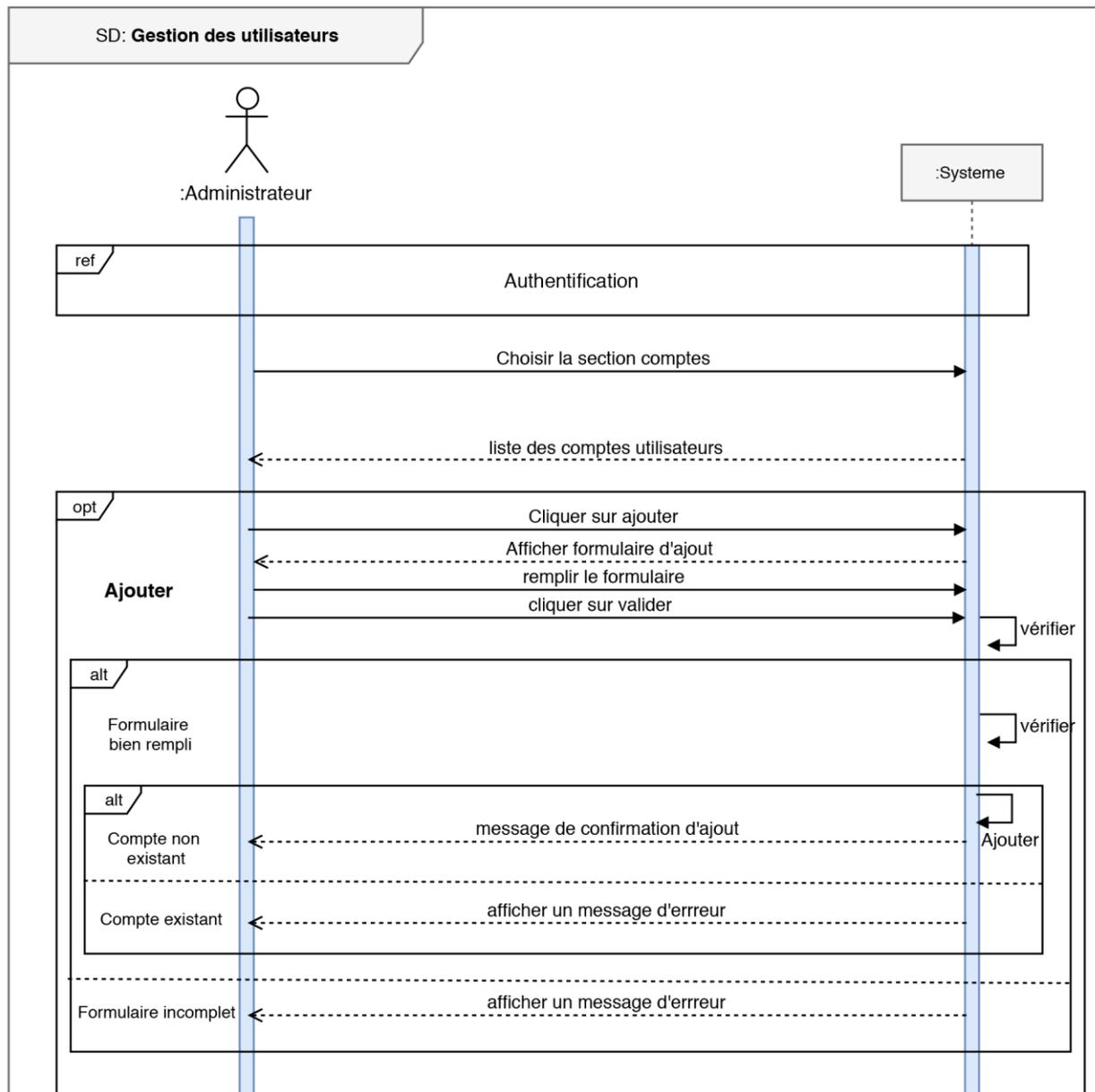
Après s'être authentifié comme administrateur. Ce dernier peut passer à la section gestion des utilisateurs. L'application affiche alors la page correspondante. Trois cas peuvent être distingués : ajouter un utilisateur, modifier un utilisateur, et supprimer un utilisateur.

- **Ajouter un utilisateur** : L'étape qui suit celle de l'affichage de l'interface de la section gestion des utilisateurs est celle de l'ajout d'un utilisateur. Après avoir cliqué sur ajouter un utilisateur, l'administrateur saisit les informations du nouvel utilisateur, puis valide son ajout à la liste des usagers.
- **Modifier un utilisateur** : Dans la liste des utilisateurs présents dans la section relative à la gestion des utilisateurs, une liste des usagers du système se présente à l'administrateur. Celui-ci peut apporter des modifications à un compte utilisateur via un formulaire, puis il valide sa demande.
- **Supprimer un utilisateur** : L'administrateur sélectionne l'utilisateur concerné, puis clique sur le bouton supprimer afin de valider son action.

Ci-dessous est la représentation du cas d'utilisation «gestion des utilisateurs», l'ajout, la suppression et la modification d'un utilisateur sont représentés par des fragments de type (opt) pour « optional » en anglais.

En utilisant le cadre « ref » on allège un autre diagramme de séquences faisant référence au diagramme de séquences « authentification ».

NB : pour la gestion des capteurs et des actionneurs, ce sont les mêmes étapes que dans la gestion des utilisateurs, à ajouter, modifier, ou supprimer un actionneur ou un capteur. Sauf qu'au lieu de passer la section gestion des utilisateurs, l'administrateur passera soit à la section actionneur pour gérer la liste des actionneurs, ou à la section capteurs pour gérer la liste des capteurs.



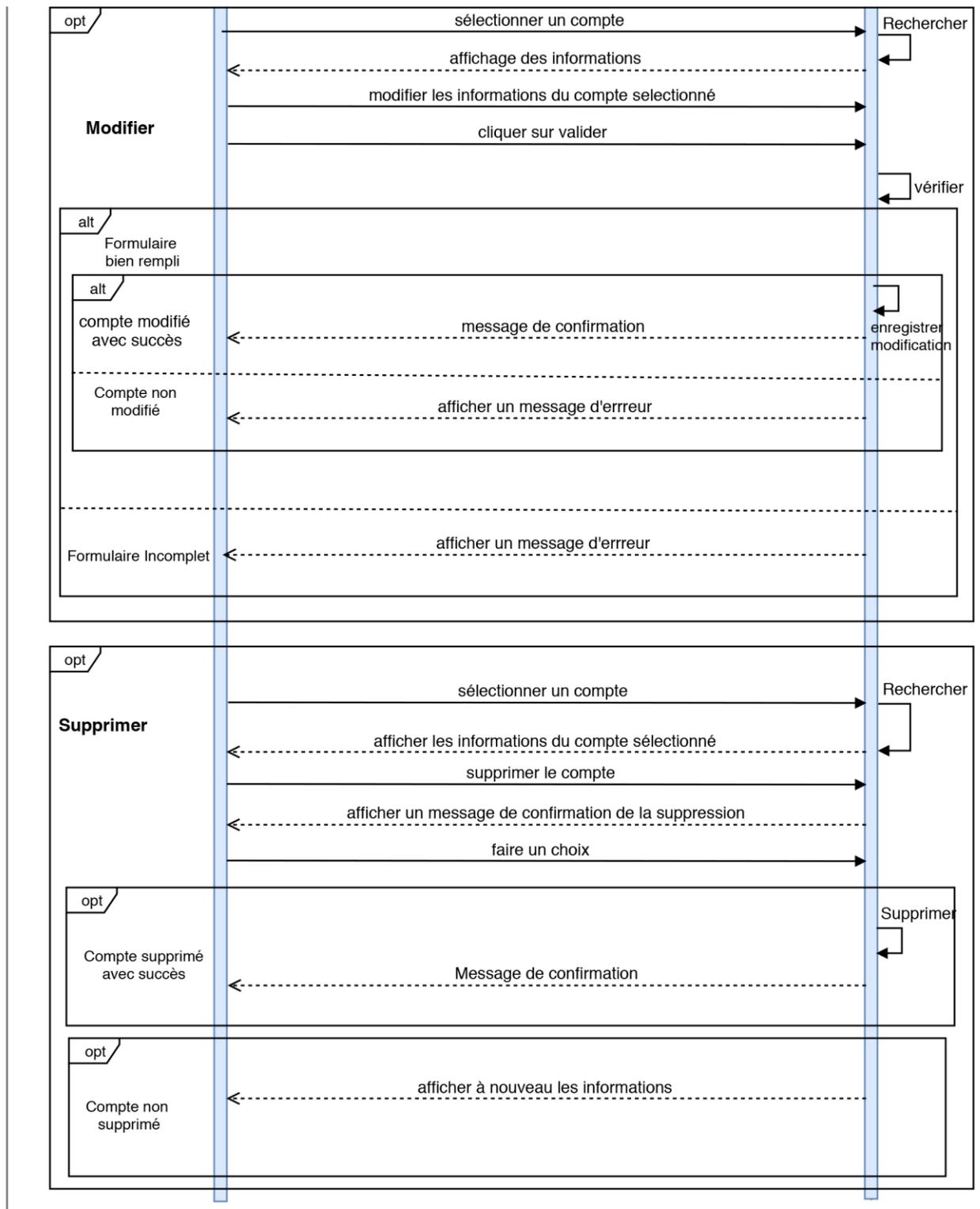


Figure 2-3 Diagramme de séquence du cas d'utilisation « Gestion des utilisateurs »

4.3. Diagramme de séquence du cas d'utilisation «Visualisation»

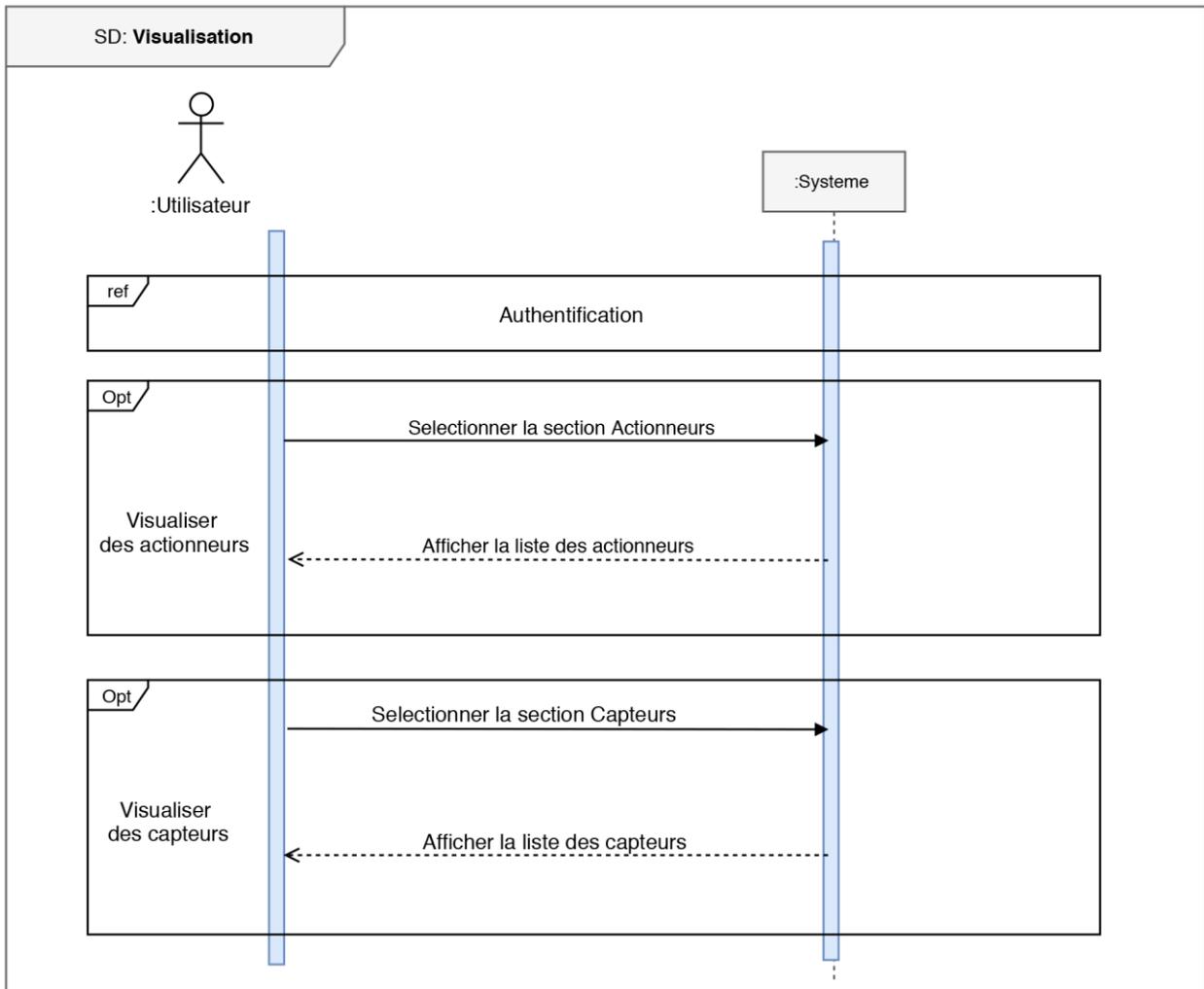


Figure 2-4 Diagramme de séquence du cas d'utilisation « Visualisation »

Après authentification, les deux acteurs (utilisateur, administrateur) peuvent soit voir l'état des actionneurs en appuyant sur l'icône représentative de cette section, ou rester dans la section d'accueil et voir les valeurs des capteurs.

Pour un simple utilisateur, celui-ci n'aura accès qu'aux capteurs et actionneurs qui lui sont autorisés. Les autorisations sont déterminées au préalable par l'administrateur du système, lors de la création d'un nouveau compte utilisateur, l'administrateur attribue des droits pour l'usage des entités présentes dans le système.

4.4. Diagramme de séquence du cas d'utilisation «consultation de l'historique et des statistiques»

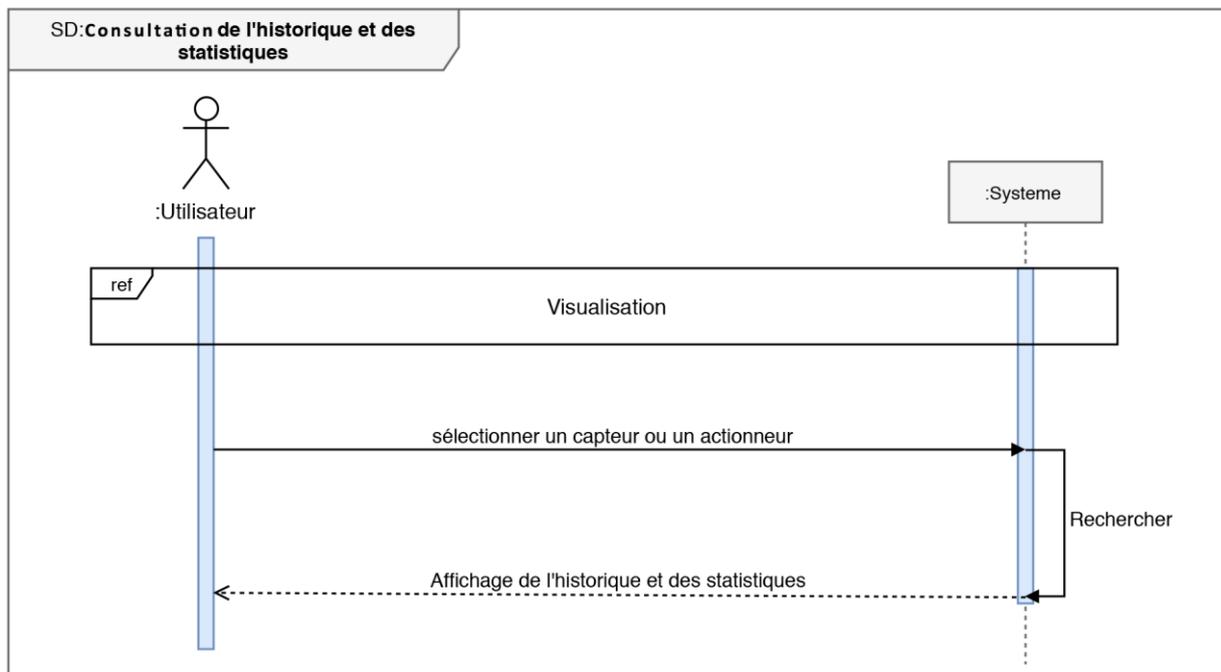


Figure 2-5 Diagramme de séquence du cas d'utilisation « consultation de l'historique et des statistiques »

Le passage par le cas d'utilisation « visualisation » illustré dans le diagramme ci-dessus par une référence est inévitable, une fois l'authentification et le choix de la section (actionneurs ou capteurs) sont faits, les acteurs pourraient voir l'historique et les statistiques d'un capteur ou d'un actionneur présent dans le système, et ce en cliquant sur le bouton « plus » de l'actionneur ou du capteur.

4.5. Diagramme de séquence du cas d'utilisation «Mise à ON ou à OFF des actionneurs»

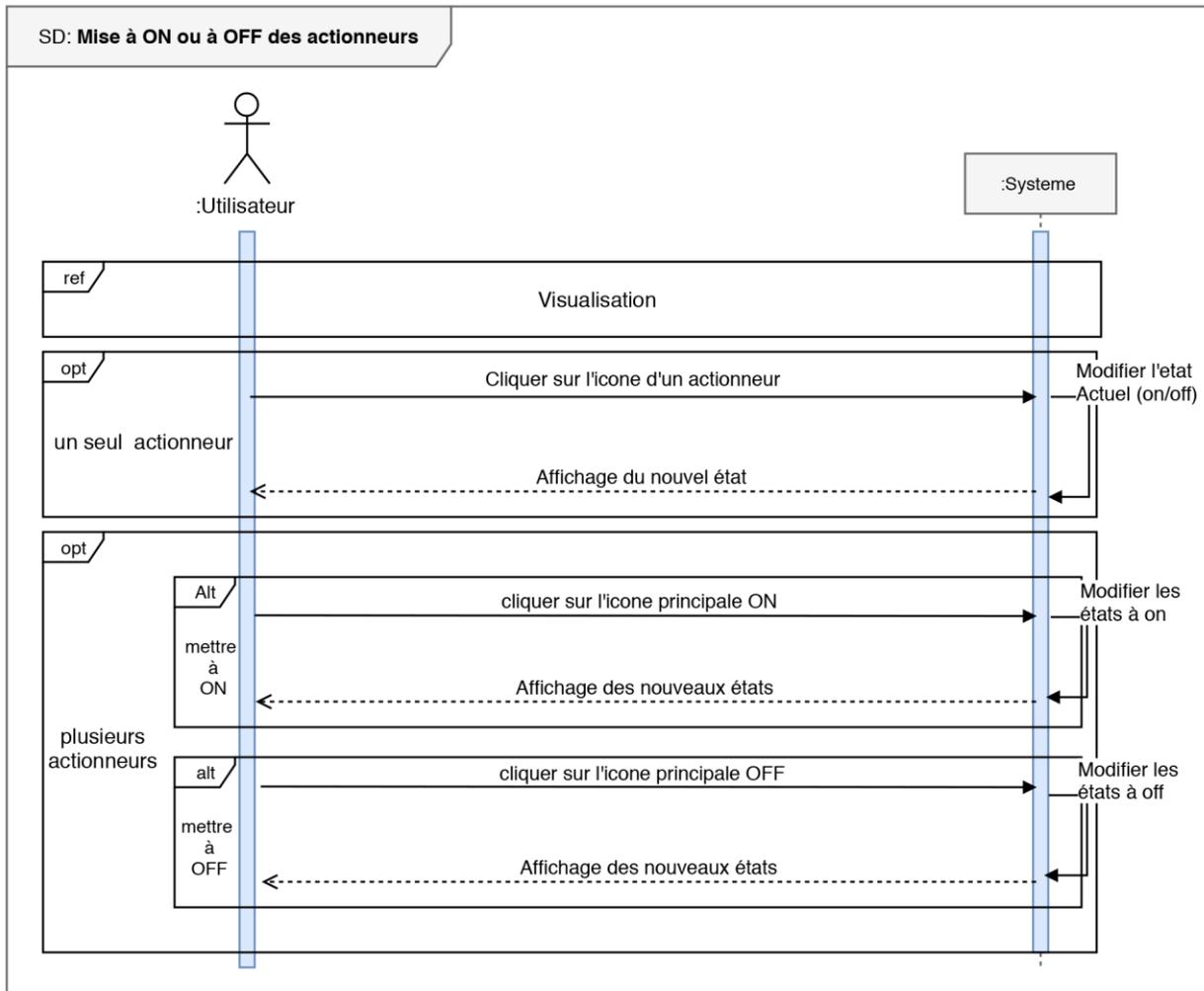


Figure 2-6 Diagramme de séquence du cas d'utilisation « Mise à ON ou à OFF des actionneurs »

Deux options peuvent se présenter lors de la mise à « ON/OFF » d'un actionneur, celle de la mise à « ON/OFF » d'un seul actionneur, et celle de la mise à « ON/OFF » de tous les actionneurs en un seul clic, dans le second cas deux boutons sont présent sur l'interface des actionneurs celui de la mise à « ON », et celui de la mise à « OFF », ce qui justifie l'emploi de l'opérateur « opt » dans le diagramme ci-dessus.

Dans ce cas aussi, il n'est possible d'actionner que les entités auxquelles l'utilisateur a droit, les autres elles sont cachées et non accessible.

4.6. Diagramme de séquence du cas d'utilisation « modification de profil »

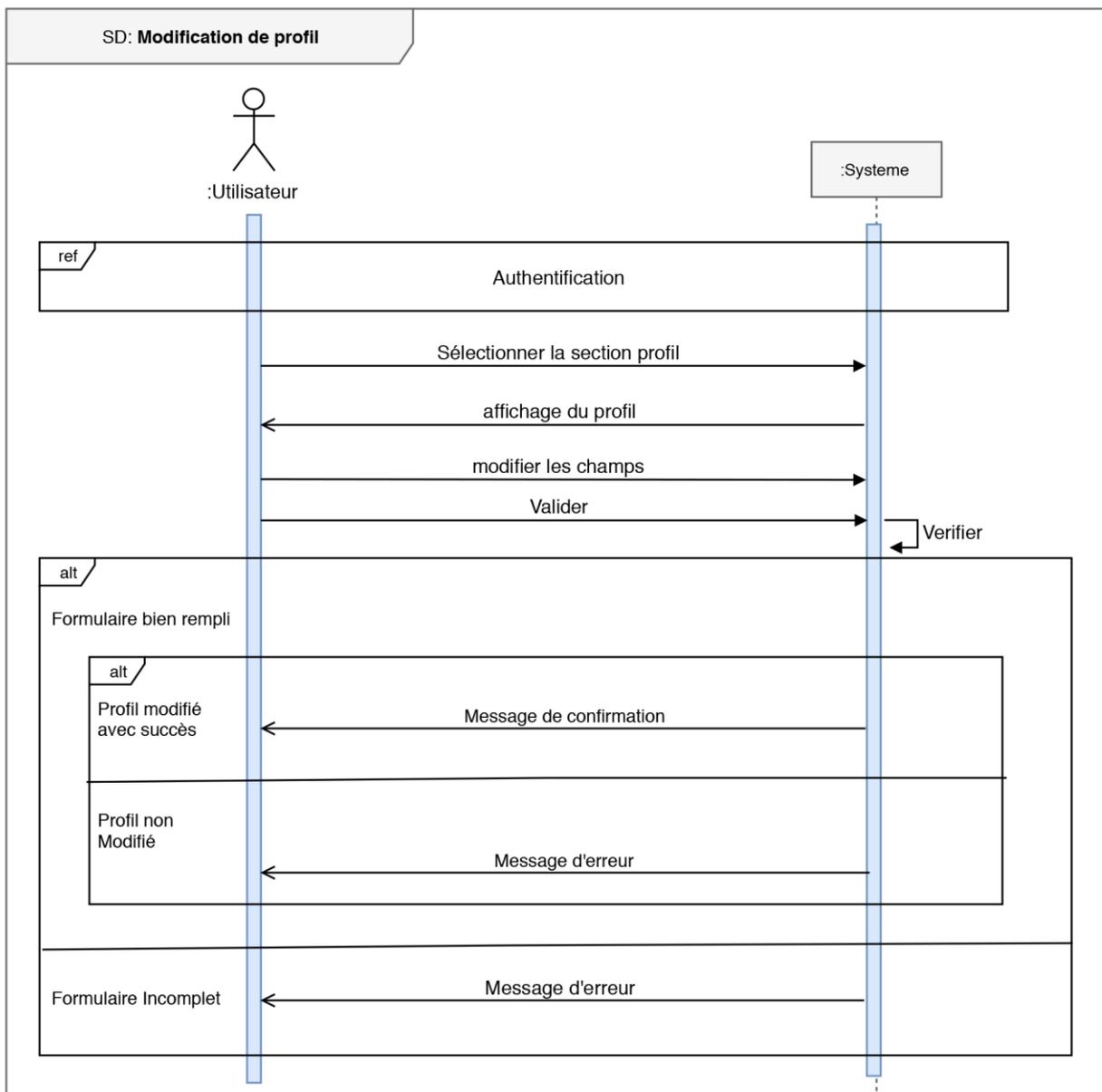


Figure 2-7 Diagramme de séquence du cas d'utilisation « modification de profil »

Dans le cas où le mot de passe de l'utilisateur est corrompu, ou pour une toute autre raison, celui-ci peut modifier son profil, et cela après authentification et le passage par la section profil.

5. Conception

Afin de réaliser un diagramme de classe, il est inévitable de se référer au dictionnaire de données et aux règles de gestion qui régissent le système. L'analyse des données du dictionnaire permet de les regrouper dans des entités à part. Ces entités sont reliées par des liens qui tiennent compte des règles de gestion.

5.1. Règles de gestion

Les règles de gestion ci-dessous régissent le système étudié :

1. Un utilisateur peut agir sur zéro ou plusieurs appareils.
2. Un appareil peut être actionné par zéro ou plusieurs utilisateurs.
3. Un utilisateur peut actionner zéro ou plusieurs appareils.
4. Un appareil peut être consulté par zéro ou plusieurs utilisateurs.
5. Un appareil peut avoir zéro ou plusieurs états.
6. Un appareil peut capter zéro ou plusieurs valeurs.
7. Une valeur avec un identifiant est captée par un seul appareil.
8. Un état avec un identifiant appartient à un seul appareil.

5.2. 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	Type
Utilisateur	util_id	Identifiant de l'utilisateur	Numérique
	util_nom	Nom de l'utilisateur	Alphabétique
	util_prenom	Prénom de l'utilisateur	Alphabétique
	util_ident	Identifiant de connexion de l'utilisateur	Alphanumérique
	util_password	Mot de passe de connexion de l'utilisateur	Alphanumérique
	util_date_insc	Date d'inscription de l'utilisateur (date et heure)	Alphanumérique
Appareil	app_id	Identifiant de l'actionneur	Numérique
	app_label	Nom d'affichage sur le terminal	Alphabétique
	app_date_creation	Date d'ajout de l'actionneur (date et heure)	Alphanumérique
	app_date_modification	Dernière date de modifications apportée sur l'actionneur	Alphanumérique
	app_statistique	Autorisation de garder trace des valeurs prises par un appareil (oui/non)	Alphabétique
	app_type	Variable identifiant l'entité (actionneur ou capteur)	Alphabétique
Etat	etat_id	Identifiant de l'état	Numérique
	etat_valeur	Valeur d'état (On/Off)	Alphabétique
	etat_date_deb	date (date/heure) de changement d'état de l'actionneur	Alphanumérique
Métrique	met_id	Identifiant de la valeur	Numérique
	met_valeur	Valeur captée par le capteur	Numérique
	met_date_deb	date (date/heure) de changement de valeur du capteur	Alphanumérique
Droits_agir	Dr_agir_date_atrib	Date d'attribution du droit	Alphanumérique
Droits_consultation	Dr_consult_date_atrib	Date d'attribution du droit	Alphanumérique

Tableau 2-10 Liste des attributs associés à chaque classe

5.3. Diagramme de classe

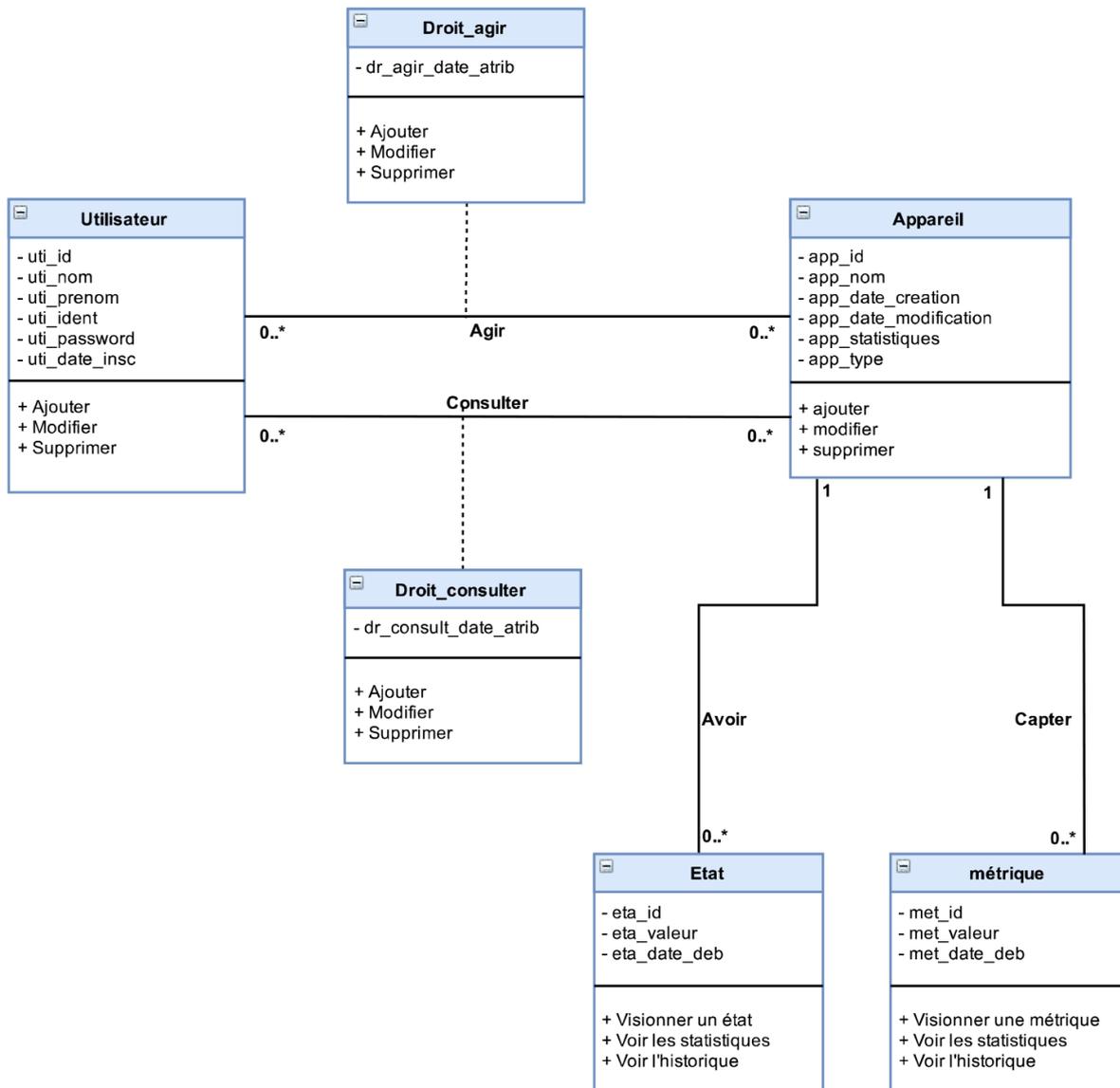


Figure 2-8 Diagramme de classe du système.

5.4. Règles de passage

Dans ce qui suit, nous allons présenter les différentes règles de passages, qui nous ont servis lors de l'élaboration du model logique de données [B5] :

- Affecter une table à chaque classe.
- Une association « un à plusieurs » engendre la migration de la clé primaire de la table mère à la table fille, qui deviendra par la suite une clé étrangère par rapport à la table fille.
- Une association « Plusieurs à plusieurs » est représentée par une table ayant pour clé primaire la concaténation des clés primaires des deux tables associées.

5.5. Le modèle relationnel

Le modèle logique de donnée ci-dessous a été dégagé après l'application des différentes règles de passages citées précédemment.

-Utilisateur (util_id, util_nom, util_prenom, util_ident, util_password, util_date_insc)

-Appareil (app_id, app_nom, app_date_creation, app_date_modification, app_statistique, app_type)

-Etat (eta_id, eta_v, eta_date_deb, #app_id)

-Métrique (met_id, met_valeur, met_date_deb, #app_id)

-Droit_consulter (#util_id, #app_id, dr_consulter_date_atrib)

-Droit_agir (#util_id, #app_id, dr_agir_date_atrib)

6. Conclusion

Dans ce chapitre nous avons pu exprimer clairement les objectifs du future système, ainsi que l'analyse associée à chaque cas d'utilisation afin d'élaborer un diagramme de classe et le model relationnel qui lui est associé.

Le chapitre suivant sera dédié à la réalisation du système de la maison connectée. Où nous présenterons son architecture logicielle et matérielle, les outils de développement utilisés, ainsi que les interfaces de notre application.

Chapitre

03

Réalisation et Implémentation

Fonctionnement de l'application

Interfaces de l'application

Environnement matériel

Environnement logiciel

...

1. Introduction

Dans ce chapitre, nous allons décrire le processus de réalisation de notre système qui n'est rien d'autre que la mise en application de ce qui a été énoncé dans le chapitre précédent. Ceci en mettant en évidence l'ensemble des environnements (logiciels et matériels) de développement, le fonctionnement du système, ainsi qu'un aperçu sur les interfaces de l'application mobile.

2. Les outils et environnements de travail

Dans cette partie, nous allons détailler l'ensemble des outils matériels et logiciels ayant permis la mise en place du système de la maison connectée. Nous parlerons des langages de programmation, d'IDEs, de logiciels, de plugins, de bibliothèque, de carte électronique, notamment d'Arduino.

2.1. Environnement matériel

Nous nous étalerons ci-dessous sur les cartes Arduino Mega et la carte Arduino Nano, qui sont les deux éléments majeurs, ainsi que l'ensemble des actionneurs et des capteurs que nous avons pu intégrer dedans.

2.1.1. Arduino Mega 2560

Cette carte (figure 3-1) est l'élément principal de notre projet, c'est le centre névralgique sur lequel viennent se greffer un tas de capteurs et d'actionneurs. Elle a pour but d'exécuter l'ensemble des instructions indiquées lors sa programmation. C'est elle qui va recevoir les valeurs envoyées par nos capteurs, et c'est elle aussi qui va commander nos actionneurs.



Figure 3-1 Carte Arduino Mega 2560.

Parmi les caractéristiques de la carte nous pouvons citer:

Microcontrôleur	ATmega2560
Poids	37 g
Dimension (mm)	101,52 X 53,3
Vitesse de l'horloge	16MHz
Tension de fonctionnement	5V
Tension d'entrée (recommandé)	7-12V
Tension d'entrée (limite)	6-20V
Digital I/O Pins	54 (dont 15 fournissent du PWM)
Broches d'entrée analogiques	16
DC Courant par I/O Pin	40 mA
Courant DC pour 3,3 Pin	50 mA
Mémoire Flash	256 Ko (ATmega2560) dont 8 Kb utilisé par Boot Loader
SRAM	8 Kb(ATmega2560)
EEPROM	4 Kb (ATmega2560)
Fréquence d'horloge	16 MHz

Tableau 3-1 Caractéristiques de la carte Arduino Mega 2560.

Les raisons et motivations du choix de la carte sont les suivantes :

- Le cout : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plateformes propriétaires (Somfy, LeGrand ...etc.).
- Open source et extensible : les cartes Arduino sont basées sur les microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA 328, ...etc. Les schémas des modules sont publiés sous une licence Creative Commons, et les concepteurs de prototypes peuvent réaliser leur propre version de cartes Arduino, en les complétant et en les améliorant et en les adaptant à leur contexte.
- La communauté : Arduino a de l'histoire, il existe depuis un bon moment déjà, du coup une grande communauté s'est formée autour de lui. Sur le site officiel d'Arduino il est possible de trouver des forums traitant de tous les problèmes rencontrés et de leurs solutions. Sur le web nous avons pu trouver facilement des articles et des sites web entiers qui lui sont dédiés. Des gens de différents profils (électronicien, informaticien, artiste...etc.) y font partie faisant ainsi d'Arduino la plateforme la plus convoitée.

- Le support logiciel : on peut trouver sur le site officiel d'Arduino des milliers de bibliothèques pour des usages divers, comme c'est le cas aussi sur d'autres sites tiers.
- La mémoire : ses 256KB de mémoire flash sont parfaitement adaptés au programme de ce projet.
- Versatilité : on peut le trouver dans des domaines divers, et utilisé par des profils variés (de l'artiste créatif jusqu'à l'électronicien ingénieur).
- Le nombre de pins : la raison majeure pour laquelle on a choisi la carte Arduino Mega est en rapport avec le nombre de pin imposant (54 I/O pins, 16 entrées analogiques) faisant d'elle la carte la mieux adaptée pour ce genre de projet.

2.1.2. Arduino Nano

La carte Arduino Nano est utilisée comme carte auxiliaire. Elle est en communication permanente avec la carte Arduino Mega via un module radio dont nous détaillerons les caractéristiques par la suite. Cette carte servira principalement à faire communiquer le niveau d'eau dans la citerne, la température et l'humidité externes, ainsi que l'état du relai contrôleur de la pompe à eau.

L'atout de cette carte est sa taille réduite (18 x 45 mm), sa compatibilité avec les autres cartes Arduino (en parfaite adéquation avec la carte Arduino Mega), ainsi que son nombre d'I/O qui est réduit à 22.

Le tableau 3.2 ci-dessous présente les caractéristiques de la carte Arduino Nano :

Microcontrôleur	ATmega328
Vitesse de l'horloge	16MHz
Tension de fonctionnement	5V
Tension d'entrée (recommandé)	7-12V
Digital I/O Pins	22 (dont 6 fournissent sortie PWM)
Broches d'entrée analogiques	8
DC Courant par I/O Pin	40 mA
Courant DC pour 3,3 Pin	50 mA
Mémoire Flash	32 Ko (ATmega328) dont 2 Kb utilisé par Boot Loader
SRAM	2 Kb(ATmega328)
EEPROM	1 Kb (ATmega328)

Tableau 3-2 Caractéristiques de la carte Arduino nano.

2.1.3. Organes

Des actionneurs, des actuators, des émetteurs/récepteurs radios, ainsi que des modules viennent se greffer sur nos cartes Arduino. La figure ci-dessous (figure 3-2) illustre bien ça, nos éléments principaux sont au centre (Arduino nano et Arduino Mega), et en leurs périphéries des organes hétérogènes.

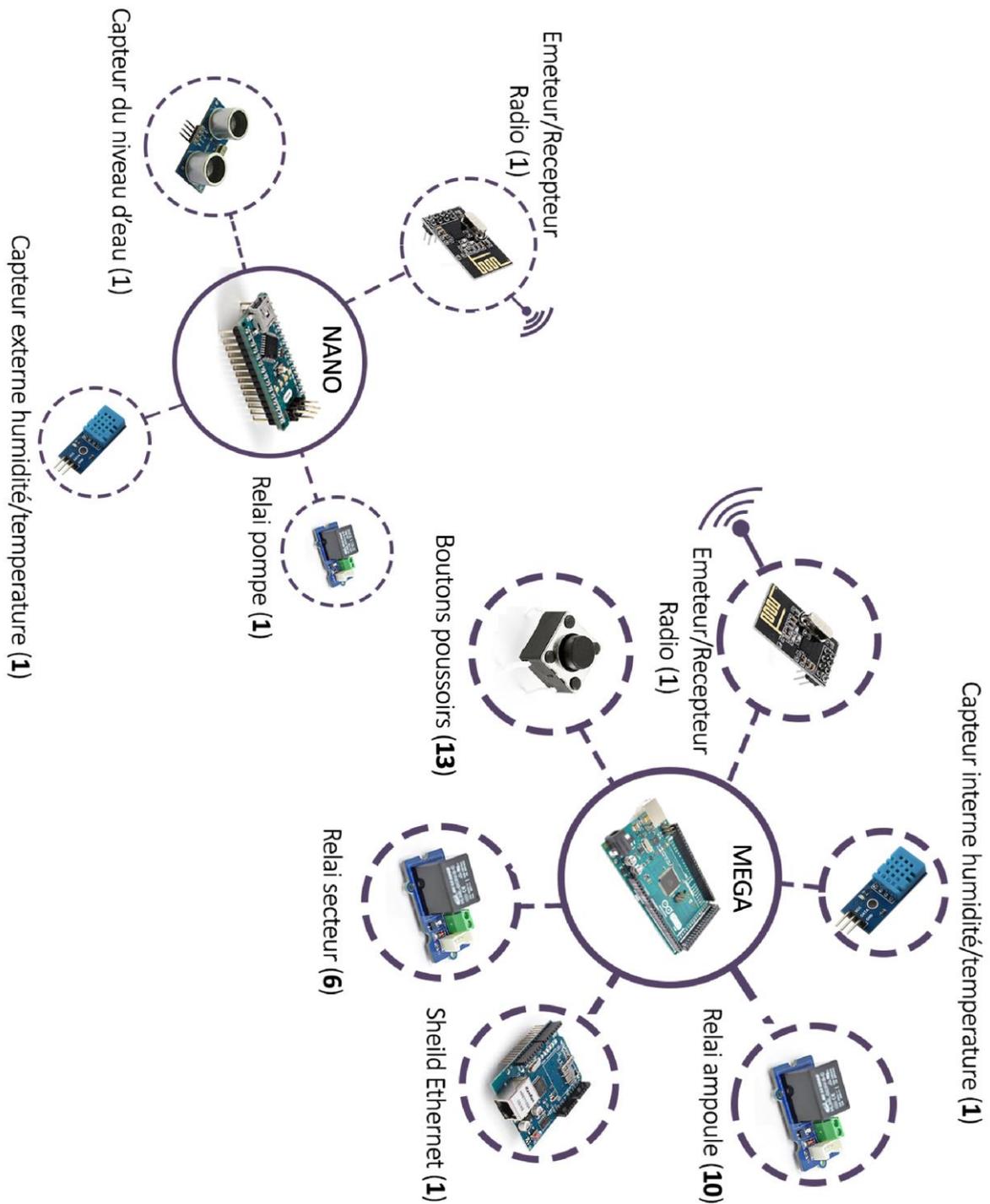


Figure 3-2 Ensemble des composants matériels impliqués dans ce projet.

2.1.3.1. Liste des capteurs

- Capteurs de température et d'humidité: DHT11 est un capteur de température et d'humidité "deux en un". Il est capable de mesurer des températures de 0 à +50°C avec une précision de +/- 2°C et des taux d'humidité relative de 20 à 80% avec une précision de +/- 5%. Une mesure peut être réalisée toutes les secondes [w12].

Deux capteurs DHT11 sont présents dans notre système, un relié à la carte Arduino Mega pour capter la température et l'humidité internes, et l'autre relié à la carte Arduino Nano pour capter la température et l'humidité externes.

- Capteurs à ultrason : le HC-SR04 est un capteur de distance très bon marché, il dispose de 4 broches (GND, VCC, TRIG, ECHO), dans ce projet nous l'avons utilisé comme capteur de niveau d'eau dans la citerne. Son fonctionnement est le suivant :

Il faut envoyer une impulsion niveau haut (à + 5v) pendant au moins 10 µs sur la broche 'Trig Input'; cela déclenche la mesure. En retour la sortie 'Output' ou 'Echo', va fournir une impulsion + 5v dont la durée est proportionnelle à la distance si le module détecte un objet. Afin de pouvoir calculer la distance en cm, on utilisera la formule suivante [w13] :

Distance = (durée de l'impulsion (en µs) / 58

2.1.3.2. Liste des actionneurs

- Relais : les relais que nous avons utilisés sont des 5v/220v, les broches 5 volts pour le circuit de commande, et les broches 220 volts pour le circuit de puissance.

Les relais sont utilisés très régulièrement, par exemple pour démarrer un moteur, commander une prise, une lumière, réguler le chauffage...etc. Dans notre cas nous les avons utilisés dans trois cas qui sont les suivant :

-un relai pour contrôler l'état de la pompe à eau.

-10 relais pour contrôler la lumière des pièces de la maison.

-6 relais pour des usages variés (humidificateur, climatiseur, chauffage...etc.)

- Boutons poussoirs : ils sont au nombre de 13, disposés dans des endroits différents, ils serviront d'interrupteurs pour chaque pièce de la maison. Il devient alors possible d'actionner les relais des ampoules soit via ces boutons soit via l'application mobile.

2.1.3.3. Shield

- Arduino Ethernet Shield : cette carte vient s'empiler sur notre carte arduino Mega. Le shield est basé sur une puce Wiznet W5100, qui permet de gérer les échanges de données avec le réseau. Elle contient 16 Ko de mémoire interne et permet jusqu'à 4 connexions simultanées (UDP et TCP) [w14].

Ainsi il devient facile d'interroger la carte via le protocole HTTP, soit par une requête GET afin de savoir les états de nos relais et les valeurs de nos capteurs, ou par des requêtes POST pour actionner des actuateurs.

2.1.3.4. Émetteur/récepteur radio [w15]

- nRF24L01 : le module nRF24L01 est un émetteur/récepteur qui utilise la bande de fréquence 2.4. Il comporte une antenne intégrée. Le protocole de transmission ShockBurst permet de garantir une transmission fiable sans s'occuper des détails. L'échange des données sur ce type de module est bidirectionnel, la vitesse de transmission maximale est de 2 Mbps, avec une faible consommation électrique, et une portée de 100 mètre en terrain dégagé. Deux modules nRF24L01 sont présents dans ce projet, l'un est relié à l'Arduino Mega et l'autre à l'Arduino Nano afin de garder une communication permanente, et de remonter les données capter par l'Arduino Nano.

2.2. Environnement logiciel

2.2.1. Langages de programmation et bibliothèques

Les langages de programmation qui nous ont permis d'implémenter ce projet sont tous des langages web, à l'exception du langage de programmation de nos cartes Arduino qui un c-like dont nous détaillerons les spécificités ci-dessous.

- PHP (PHP Hypertext Preprocessor) : langage spécialement conçu pour le développement d'application web qui peut être facilement intégré au HTML. Ce qui distingue PHP des langages de script comme le Javascript, est que le code est exécuté sur le serveur, générant ainsi le HTML, qui sera ensuite envoyé au client. Le client ne reçoit que le résultat du script, sans aucun moyen d'avoir accès au code qui a produit ce résultat. Ainsi, il n'y a aucun moyen de distinguer les pages qui sont produites dynamiquement des pages statiques [w16]. Le grand avantage de PHP est qu'il est extrêmement simple pour les néophytes, mais offre des fonctionnalités avancées pour les experts. Parmi ses finalités, PHP nous a permis de concevoir des scripts qui authentifieront les utilisateurs qui vont utiliser le système, ainsi que des scripts qui interrogeront à temps régulier la carte Arduino Mega pour persister l'information récupérée dans une base de données.
- 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) tel que Oracle, Mysql, Posgresql...etc.

- HTML (hypertext markup language) : il ne s'agit pas d'un langage de programmation mais plutôt d'un langage de description et de structuration de page. Etant simple à assimiler HTML permet d'écrire de l'hypertexte à base d'un système de balisage. Il a connu de nombreuses itérations dont la dernière en date est : HTML5.

Grace à des API intégrée dans HTML5 on peut :

- Utiliser le stockage local dans une application mobile,
- faire fonctionner les applications mobiles en mode hors ligne,
- Développer des nouvelles fonctionnalités visuelles,
- ...etc.

HTML5 nous a servi pour structurer le contenu de notre application mobile ainsi que les résultats retournés par notre carte Arduino une fois interrogée.

- CSS (Cascading Style Sheets) : C'est un langage qui permet de gérer la présentation (mise en forme de nos pages). Il permet de préciser les caractéristiques visuelles d'une page ou d'un élément : les polices de caractères et leur taille, les bordures, les couleurs...etc.
- Sass et Compass : Sass est une extension de CSS3, ajoutant de nouvelles règles dans notre façon d'intégrer un web design. Les grandes nouveautés sont : les variables, les mixins, et l'héritage.

Compass quand à lui est une bibliothèque basée sur Sass permettant d'accélérer le Workflow grâce à des mixins prédéfinies.

- Langage Arduino : le langage utilisé par le logiciel Arduino pour programmer le microcontrôleur est basé sur les langages C/C++.

Un programme Arduino est constitué principalement d'une entête faisant appel aux dépendances du programme, d'une fonction 'setup' qui initialise les variables impliquées, ainsi que d'une boucle 'loop' qui exécute en permanence les instructions de notre programme.

- JQuery : JQuery est une bibliothèque JavaScript libre d'usage. Elle intègre de nombreuses fonctionnalités, parmi elles on peut citer [B6] :
 - Accéder aux éléments d'un document,
 - Altérer le contenu d'un document,
 - Répondre aux actions de l'utilisateur,
 - Animer les modifications d'un document,
 - écrire plusieurs instructions sur une seule ligne grâce au chainage.

2.2.2. Outils et logiciels

- Photoshop : Ce logiciel qui est à notre goût un passe par tout, nous a permis de faire la conception des interfaces de notre application mobile : Pour avoir des prévisualisations, à choisir la palette des couleurs, à découper les icônes impliqué dans nos interfaces, à les redimensionner, et à les compresser pour alléger l'application.
- VirtualBox : c'est un logiciel de virtualisation de systèmes d'exploitation. Il permet la création d'un ou de plusieurs ordinateurs virtuels (*machines virtuelles*) dans lesquels s'installent d'autres systèmes d'exploitation (*systèmes invités*). Il servira dans ce projet à accueillir la distribution Ubuntu Server que nous détaillerons par la suite.
- SublimeText : c'est notre éditeur de code favori.
- Ripple : c'est une extension émulateur d'un environnement mobile sur navigateur. Elle nous a évité les allers retours sur smartphone et la répétition de commande de compilations pour un changement mineur dans l'application (des éléments visuel pour la pluparts).
- Arduino IDE : Le logiciel Arduino est un environnement de développement (IDE) open source et gratuit, téléchargeable sur le site officiel Arduino, cet IDE dédié aux cartes Arduino permet :
 - d'éditer un programme : des croquis (*sketch* en Anglais),
 - 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 Arduino grâce au terminal.

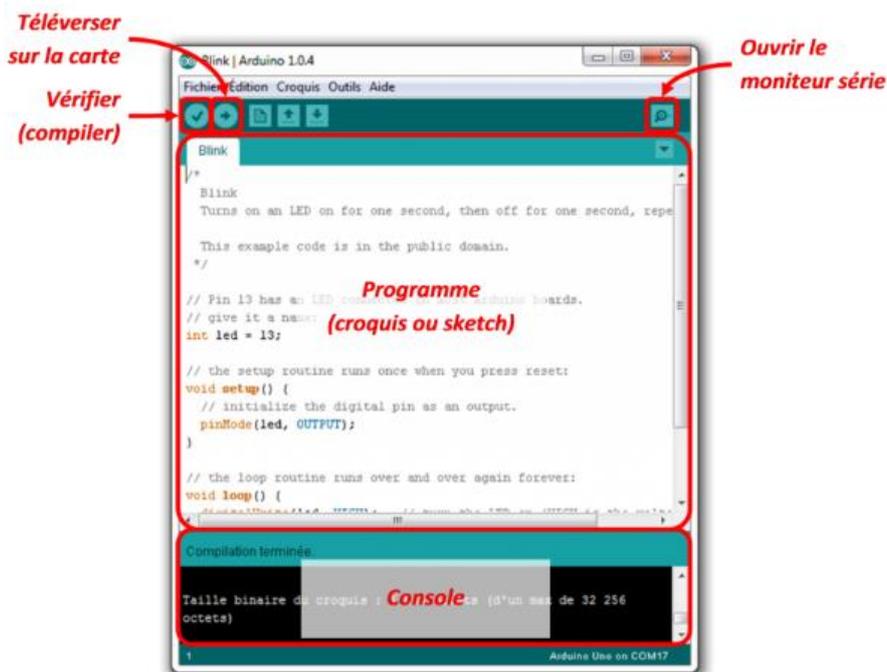


Figure 3-3 Environnement Arduino.

- Apache Cordova : Cordova permet de développer des applications pour des plateformes diverses avec les langages du web. Celles-ci contrairement aux webApp traditionnelles, ont la particularité de s'exécuter dans un contexte différent leur offrant davantage de liberté. On peut citer par exemple l'usage de la camera pour la prise de photos et ou de vidéos, ou encore l'accès au répertoire de l'appareil.

Cordova exploite une web-view qui n'est rien d'autre qu'un navigateur dépourvu de ses éléments visuels, des APIs HTML5 sont intégrées à celle-ci, lesquelles se chargeront alors de communiquer avec la machine en langage natif via Javascript.

Cordova est une sorte de traducteur JavaScript vers un langage natif (et vice versa) reposant sur les standards du Web, fonctionnant de façon extrêmement simple et quasi transparente. Une application web exécutée dans ce contexte, dialoguant avec du code natif, est nommée « application hybride » [B7].

- Ubuntu Server: Ubuntu Server est une variante de Ubuntu. Elle possède un noyau optimisé, et est dépourvue d'environnement graphique (toutes les configurations se font via la console), ce dernier est gourmand en ressources et superflu dans le cas d'un serveur amené à être manipulé assez rarement. Cette variante est donc la plus adaptée pour la mise en place d'un serveur utilisé de manière intensive [w17].

La mise en place d'un serveur se fait en deux étapes :

1. Installation (et configuration) du système d'exploitation.
2. Installation (et configuration) de l'une ou des application(s) adaptée(s) au(x) service(s) désiré(s) (dans notre cas le service désiré est celui d'un serveur HTTP).

Parmi les services installés sur cette distribution nous pouvons mentionner :

Le compromis PHP, MYSQL : des modules ont été ajoutés a PHP, à titre d'exemple : CURL, qui joue le rôle d'intermédiaire de requêtes.

Apache : c'est notre serveur HTTP, sa version est la 2.0

SSH : SSH est l'acronyme de Secure Shell, cet outil sert à faire des accès à un serveur distant pour des configurations et de la supervision.

VIM : éditeur de fichier sous la console très puissant une fois les raccourcis clavier sont assimilés.

3. Description du système

3.1. Disposition des éléments du système

La figure 3-4 illustrant la disposition des éléments de notre système sur le plan de la maison concernée. Nous pouvons constater qu'il y a deux types de liaisons, une liaison filaire reliant la carte Arduino Mega aux actionneurs, et une liaison radio entre la carte Arduino Mega et la carte Arduino Nano.

NB : Pour des raisons de simplicité nous avons regroupé ensemble la carte Arduino Mega et ses composants ainsi que le serveur virtuel et le point d'accès à internet pour finalement les représenter uniquement avec la carte Arduino Mega. Les icones symbolisant chaque pièce ont été reprises dans l'application mobile.

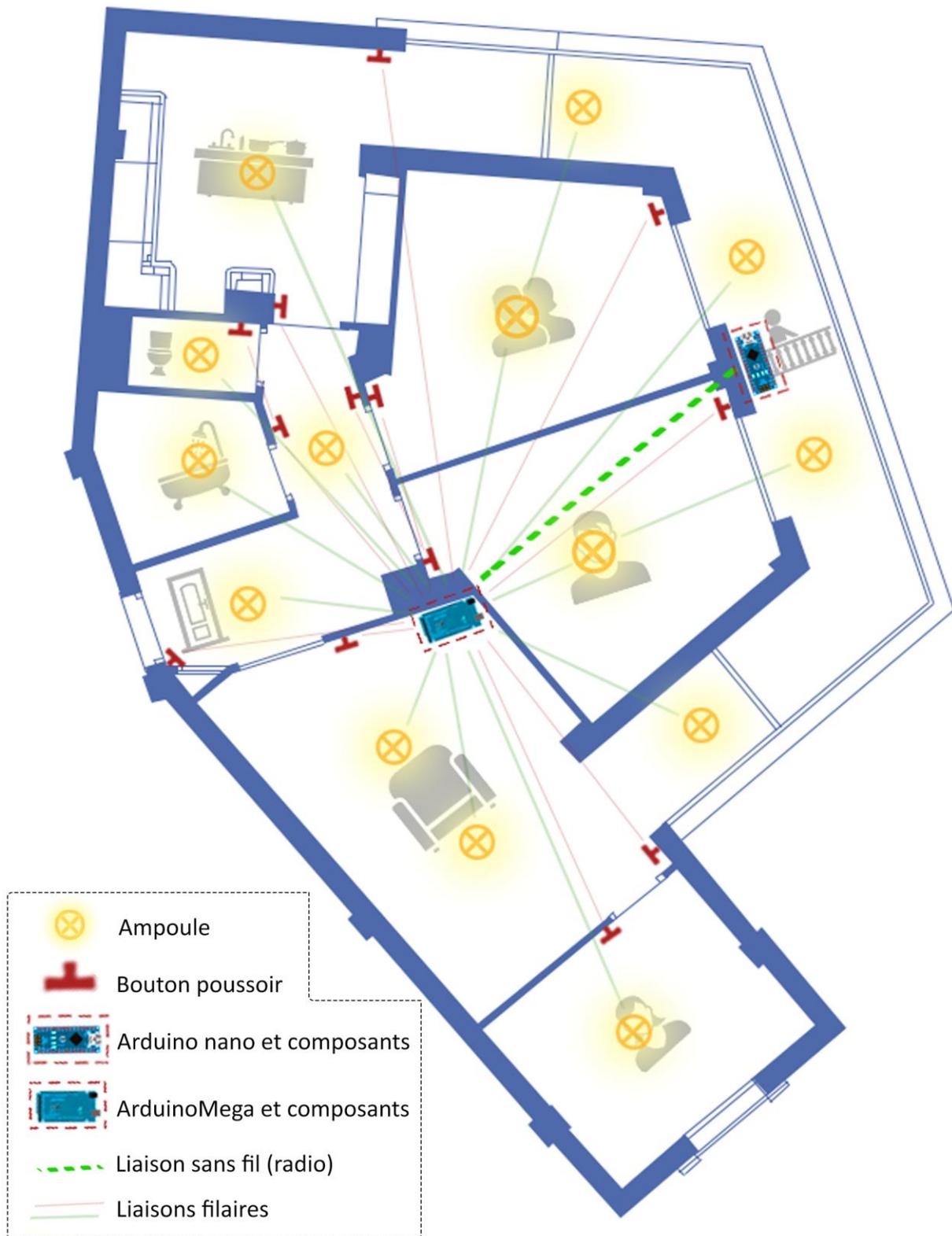


Figure 3-4 Disposition des éléments du système.

3.2. Architecture réseau

Afin que des éléments puissent communiquer avec notre système de maison connectée un point d'accès (figure 3-5) nous a été inévitable, dedans figurent deux liaisons, une filaire, et une autre en WIFI.

Nous pouvons distinguer deux réseaux dans le réseau global où est présent le système de maison connectée, un réseau privé non accessible de l'extérieur (via internet) regroupant tous les appareils du domicile qui sont utilisés principalement pour un accès à internet (navigation web, téléchargement, consultation de courrier...etc.), et une zone démilitarisé (DMZ) faisant en sorte que le système soit accessible depuis l'extérieur (par internet).

Deux types d'adressage y figurent dans ce système, un statique et l'autre dynamique. Le point d'accès a été configuré en premier de façon à attribuer des adresses IP dynamique aux équipements (PC, tablette, smartphone) du réseau, cela grâce au service DHCP intégré dans le routeur. En deuxième nous trouvons une configuration statique, dont laquelle nous attribuons des adresses IP fixes aux équipements de notre système de maison connectée, les adresses pour la carte Arduino Mega et le serveur virtuel sont respectivement comme suit : 192.168.1.177, 192.168.1.199.

Le réacheminement de port (port forwarding ou port mapping en anglais) consiste à rediriger des paquets réseaux reçus sur un port donne du routeur vers un autre ordinateur ou équipement réseau sur un port donné. Pour permettre aux machines externes d'accéder à notre système de maison connectée, une configuration de réacheminement de port est nécessaire. Ainsi toute personne ayant l'application mobile dédié au système de la maison connectée installée sur leurs téléphones serra en mesure de s'y connecter après identification.

Dans notre cas, lorsqu'un utilisateur veut accéder au système depuis l'extérieur en tapant l'adresse IP public du réseau où est connectée notre maison, un port forwarding est effectué au niveau du routeur pour rediriger les requêtes vers l'équipement concerné, en l'occurrence le serveur virtuel ayant comme adresse 192.168.1.199:80.

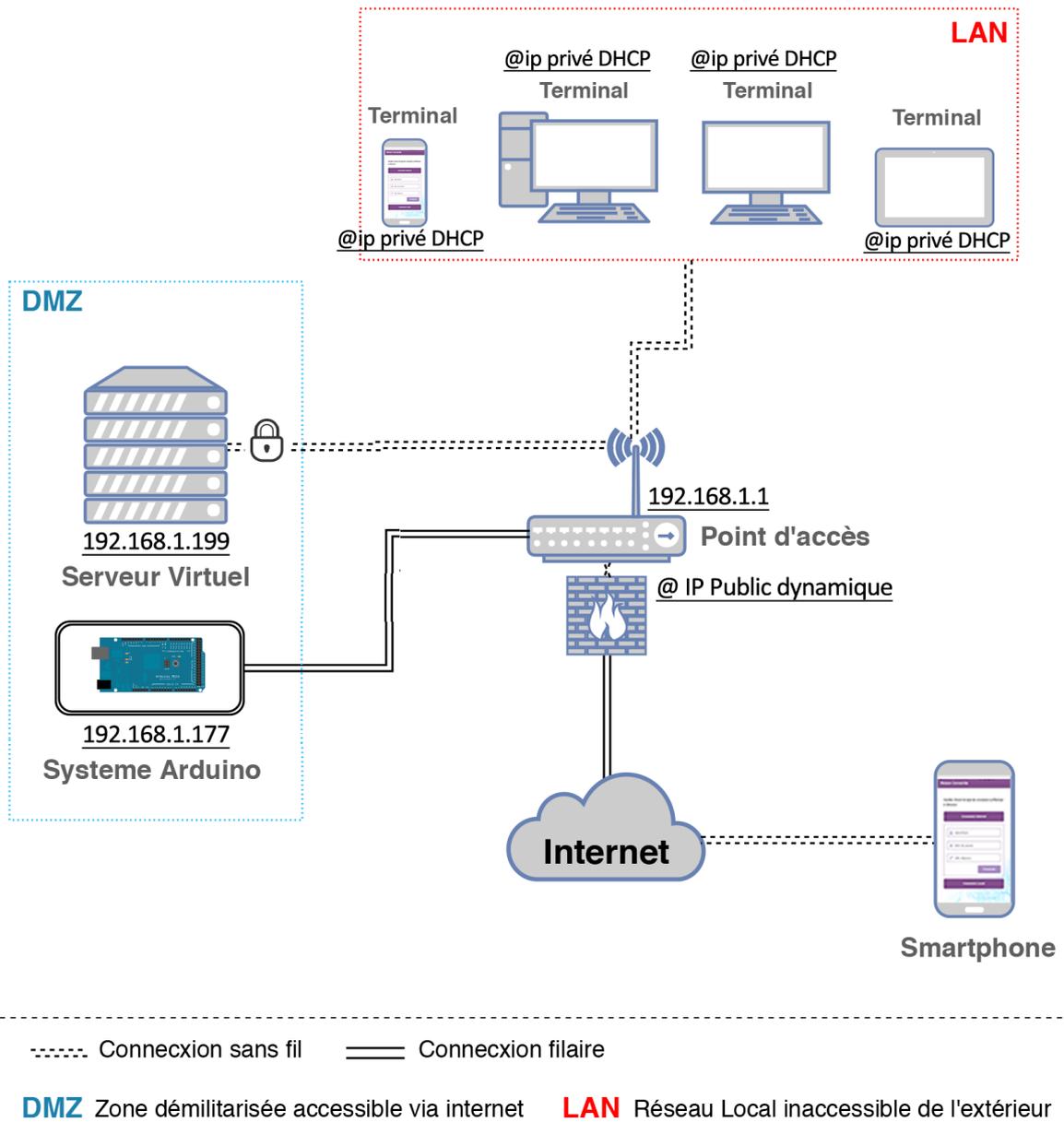


Figure 3-5 Architecture réseau.

3.3. Schématisation électronique du système

Les deux figures qui vont suivre (figure 3-6 et figure 3-7) sont les schémas électroniques de nos cartes Arduino Mega et Nano, reliées aux différents capteurs et actionneurs. Sur ces deux schémas nous pouvant distinguer deux types de circuits : un circuit de puissance qui alimente les actionneurs, et un circuit de commande relié aux cartes Arduino. Nous remarquons aussi la présence de flèches, celle qui sont de couleur bleue sont dites des signaux inputs (entrants), et celle qui sont de couleur rouge sont dites des signaux outputs (sortants).

3.3.1. L'Arduino Mega et ses composants

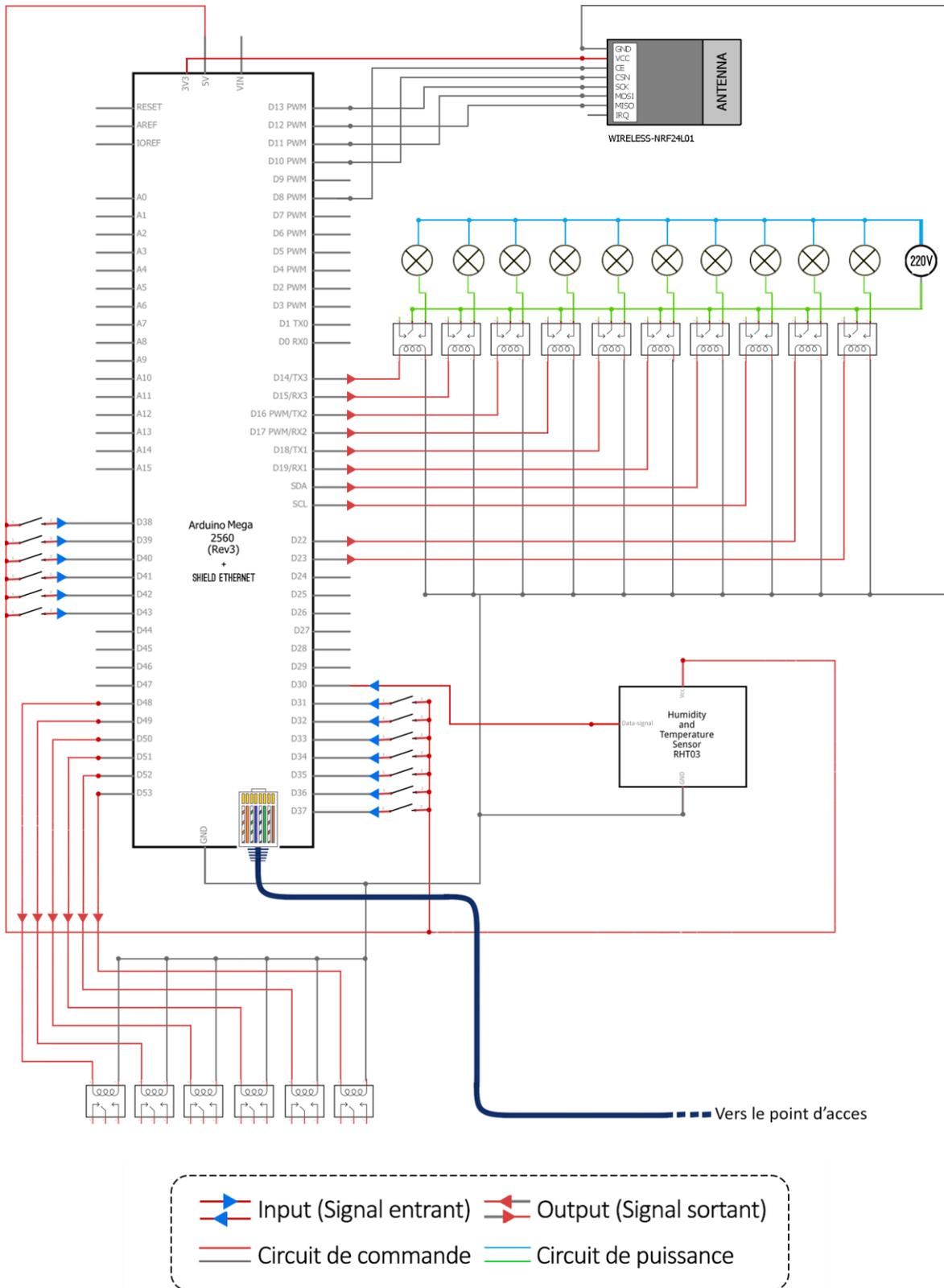


Figure 3-6 Schéma électrique de l'Arduino Mega et ses composants.

3.3.2. L'Arduino Nano et ses composants

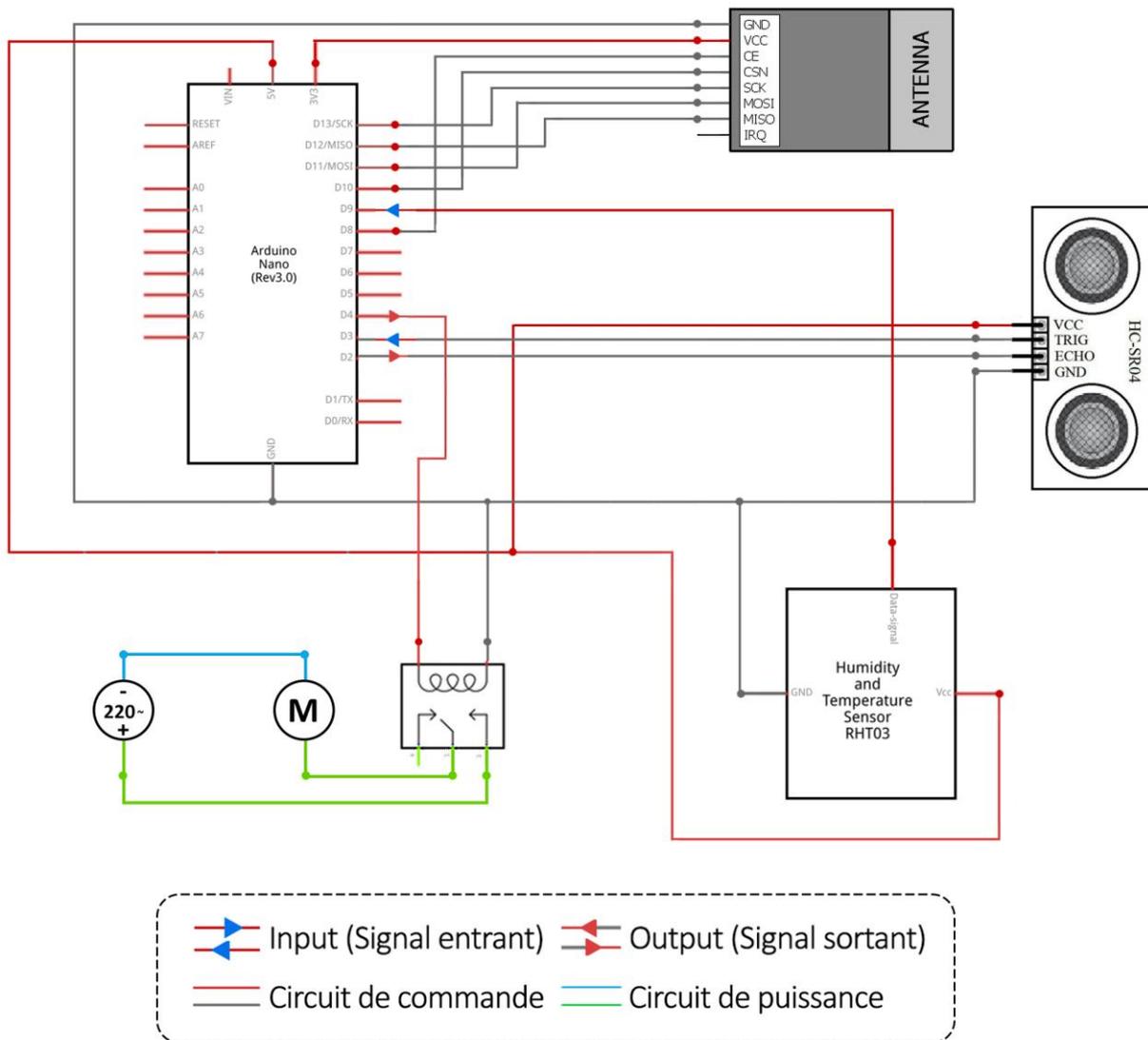


Figure 3-7 Schéma électrique de l'Arduino Nano et ses composants.

4. Description de l'application

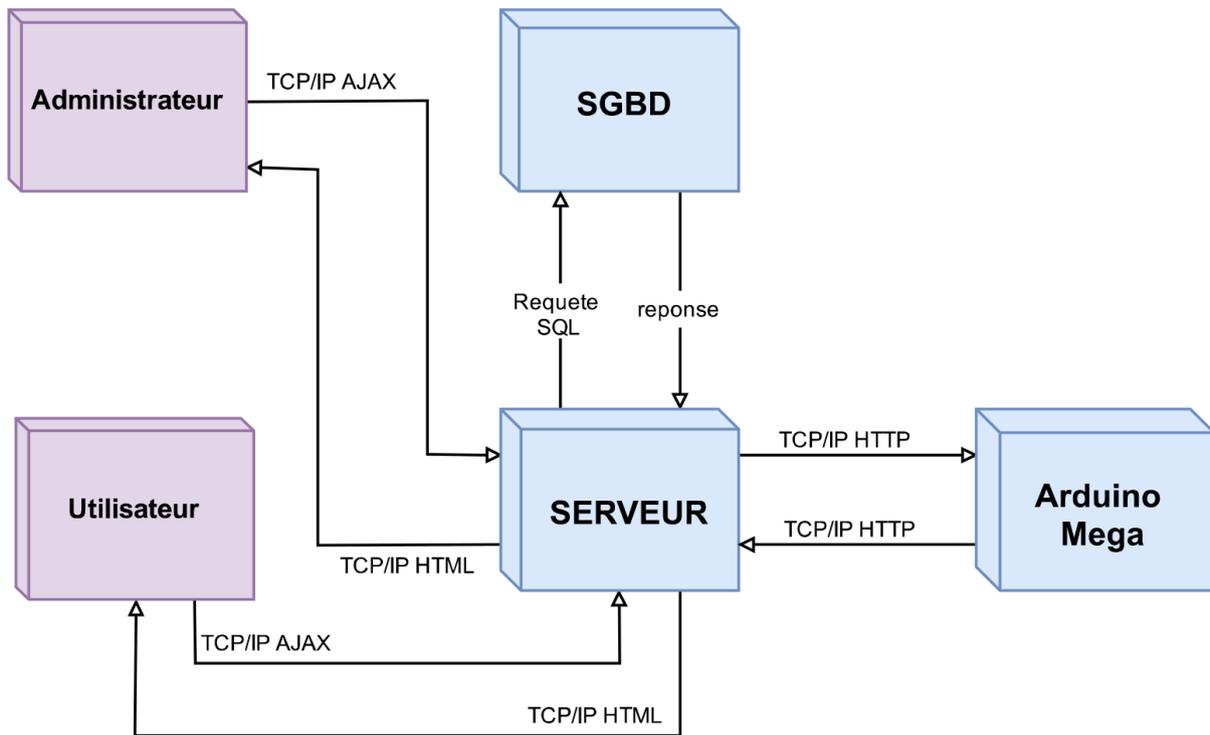


Figure 3-8 Description de l'application.

Notre système se base sur une architecture client/serveur. Chaque acteur a son propre terminal « smartphone » qui est en interaction permanente avec le serveur sur lequel est déployé une partie de l'application. Chacun des acteurs, partage une interface homme-machine qui utilise le service d'authentification contenu dans le serveur. Toutes les tables sont stockées dans une base de données qui est hébergée sur le serveur de base de données (SGBD).

4.1. Présentation du serveur virtuel

Notre application est basée sur le serveur qui joue le rôle d'un noyau. Tous les traitements effectués transitent par le serveur, c'est le centre du système de la maison connectée.

Ses principales fonctions sont les suivantes :

- **Authentification des utilisateurs** : c'est à lui d'authentifier les utilisateurs du système pour leur accorder l'accès au système ou pas. Si un utilisateur est authentifié alors ce dernier aura accès au contrôle de la maison connectée, dans le cas échéant le système lui renvoie un message d'erreur lui indiquant que les informations renseignées ne sont pas valides.

- Récupérer des données auprès de l'Arduino : un script en exécution permanente sur le serveur permet de récupérer des données sur les états des actionneurs et des capteurs à intervalle de temps prédéfini (1 minute pour les capteurs) et à changement d'état (pour les actionneurs).
- Médiateur entre Arduino et utilisateurs du système : du fait que l'Arduino ne possède pas de système d'exploitation, une surcouche logicielle ne peut être installée dessus. De ce fait une nécessité d'un serveur intermédiaire gérant les utilisateurs ayant une légitimité est nécessaire.
- Vérification et contrôle d'erreur : c'est à ce serveur de vérifier l'existence dans la base de données d'enregistrements entrant en conflit avec un enregistrement nouvellement saisi et d'en alerter l'utilisateur.

4.2. Arborecence de l'application

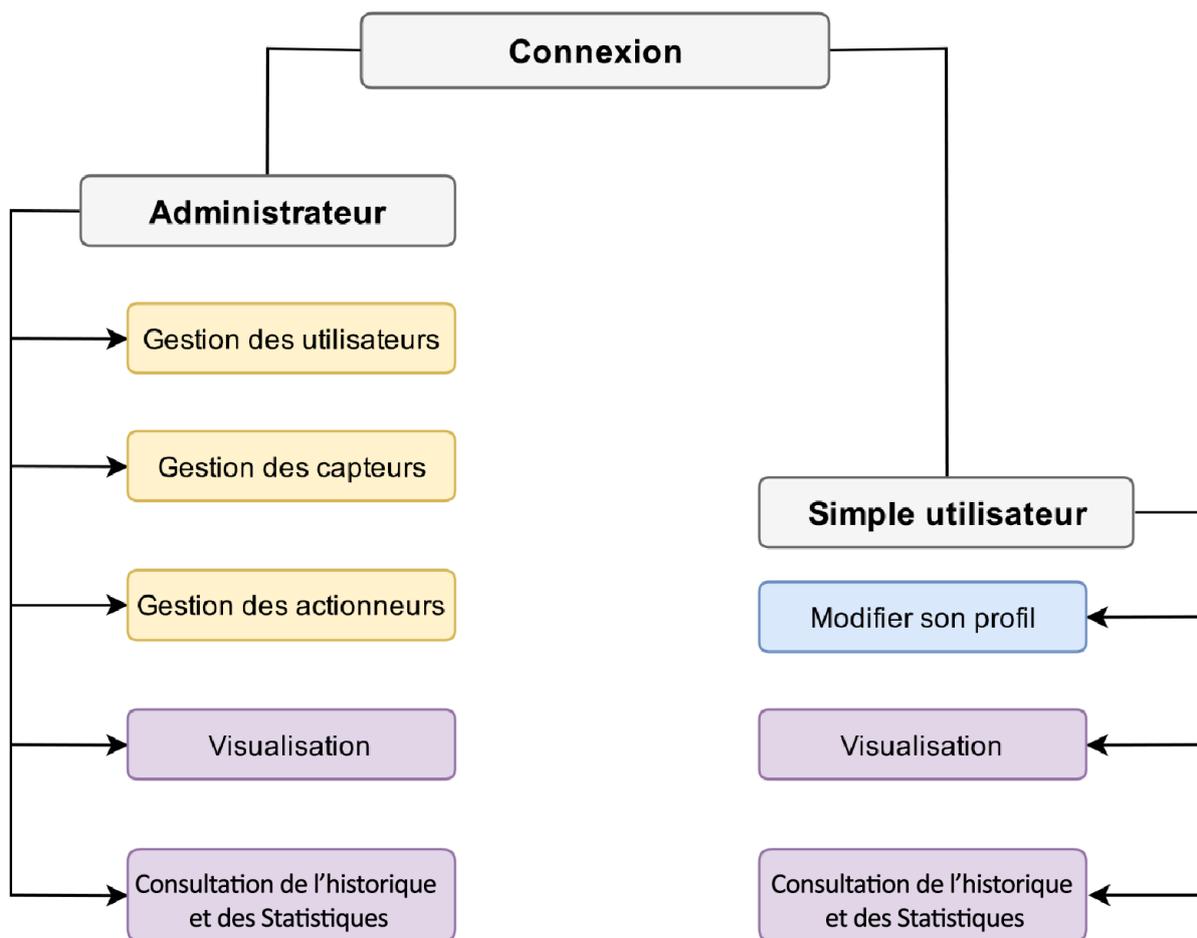
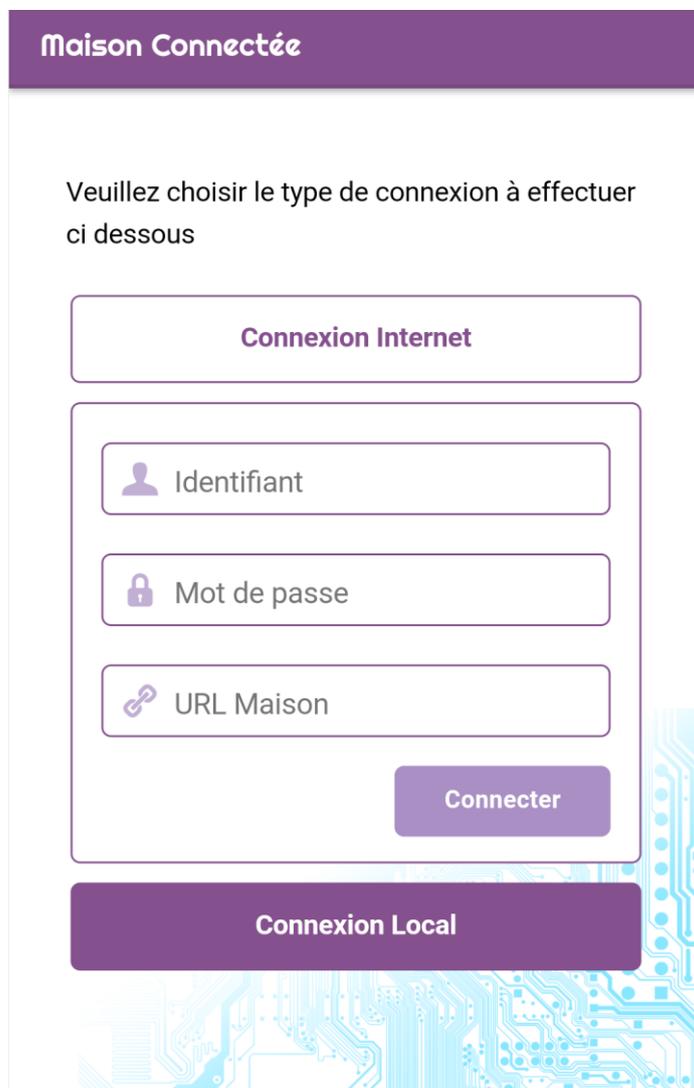


Figure 3-9 Arborecence de l'application.

5. Présentation des interfaces

5.1. Interface d'authentification



The screenshot shows a mobile application interface for 'Maison Connectée'. At the top, there is a purple header with the text 'Maison Connectée'. Below the header, the text reads 'Veuillez choisir le type de connexion à effectuer ci dessous'. There are two main options for connection: 'Connexion Internet' and 'Connexion Local'. The 'Connexion Internet' option is currently selected, indicated by a purple border and a purple background. This option contains three input fields: 'Identifiant' (with a person icon), 'Mot de passe' (with a lock icon), and 'URL Maison' (with a link icon). A purple 'Connecter' button is located below these fields. The 'Connexion Local' option is shown as a solid purple button below the internet option. The background of the interface features a light blue circuit board pattern.

Figure 3-10 Interface d'authentification.

Sur cette figure nous trouvons l'interface d'accueil de notre application invitant l'utilisateur à faire un choix entre deux méthodes de connexion, une via internet, et une autre en local. Ici on a développé celle de la connexion via internet dans laquelle figurent trois champs, l'identifiant de l'utilisateur, son mot de passe, ainsi que l'URL de la maison connectée. Pour la connexion en local deux champs sont uniquement présents, un identifiant et un mot de passe.

5.2. Interface d'établissement de connexion



Figure 3-11 Interface d'établissement de connexion.

Après authentification auprès du serveur, l'utilisateur est redirigé vers cette page (figure 3-11) lui indiquant que le système effectue un établissement de connexion avec la carte Arduino, d'où l'intérêt d'un spinner (barre rotative), dans le cas où la connexion n'a pas été établie un message d'information lui est affiché.

5.3. Interfaces d'accueil



Figure 3-12 Interface d'accueil pour l'administrateur.

Une fois l'établissement de connexion est fait, l'administrateur se trouve sur cette interface d'accueil, sur laquelle il peut : ajouter de nouveaux capteurs en cliquant sur le bouton plus en haut, consulter la valeur des capteurs ou les modifier, comme il peut aussi cliquer sur le bouton plus (à côté de chaque entité et avoir des détails concernant celle-ci).

Quatre icônes cliquables figurent dans La barre du menu principal, il y a (de gauche à droite) : l'icône vers la section profil, l'icône vers la section capteur, l'icône vers la section d'accueil, l'icône vers la section de contrôle de la lumière des chambres.

NB : l'interface d'accueil pour l'utilisateur simple est épurée de boutons de configuration des entités et celui de l'ajout de capteurs comme nous pouvons le voir sur la figure 3-13.



Figure 3-13 Interface d'accueil pour l'utilisateur simple.

5.4. Interfaces de contrôle de la lumière des chambres

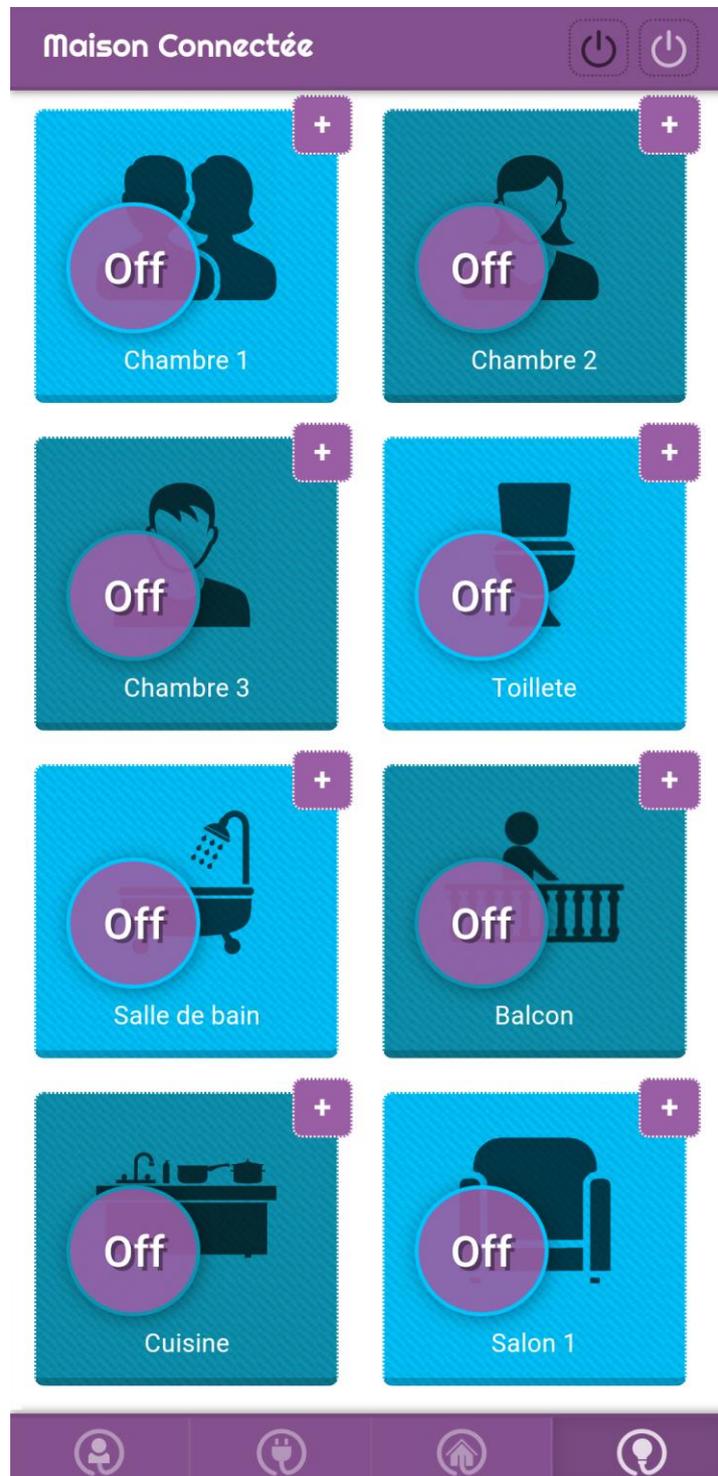


Figure 3-14 Interface de l'utilisateur simple pour le contrôle de la lumière.

La figure 3-14 représente la section permettant de contrôler la lumière de chaque pièce de la maison. Chaque pièce est illustré par une icône représentative, si la lumière d'une pièce donnée est éteinte, l'icône est alors de couleur noire, sinon elle devient blanche lorsque l'utilisateur clique sur le cercle violet au-dessus de l'icône de la pièce. Les deux boutons en haut (ceux de on/off) serviront à éteindre toutes les pièces d'un seul coup, et les boutons « plus » serviront à avoir des détails sur chaque pièces.

Sur l'interface d'administration de cette section, nous trouvons la même disposition, sauf qu'il y a des boutons en plus, comme nous pouvons le voir sur la figure 3-15, des boutons de configuration et un bouton d'ajout d'ampoule viennent s'ajoutés à l'interface.

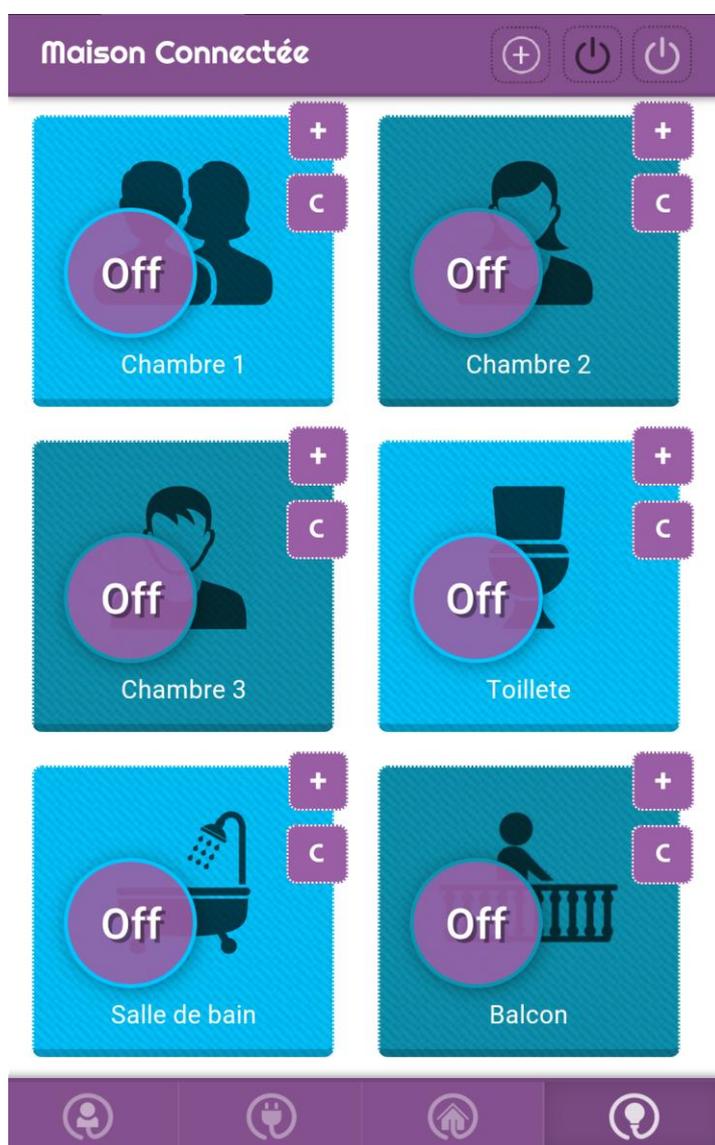


Figure 3-15 Interface de l'administrateur pour le contrôle de la lumière.

5.5. Interface de contrôle des prises

C'est dans cette section (figure 3-16) que l'utilisateur pourra contrôler les différents appareils, et ce en cliquant sur le cercle violet de l'appareil concerné. Les deux boutons de la barre du haut permettent respectivement de mettre à « on » ou à « off » tous les appareils connectée au système.

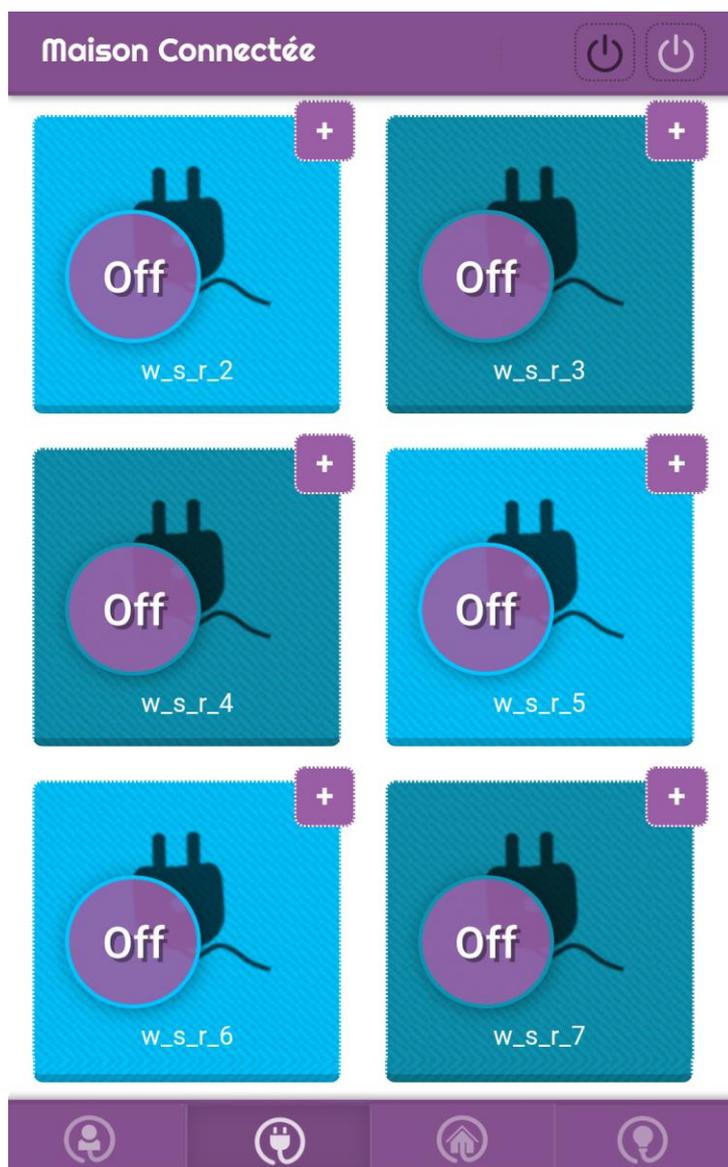


Figure 3-16 Interface du simple utilisateur pour contrôler les prises.

NB : comme dans les sections précédentes, l'interface d'administration diffère avec celle des usagés simples dans les boutons de configuration et celui de l'ajout d'entité.

5.6. Interface « Plus d'info sur l'entité »

Cette interface (figure 3-17) est dédiée aux détails à propos d'une entité. Dans ce cas c'est celle de la lumière de la cuisine dans laquelle nous pouvons trouver : la valeur actuelle, le nombre d'heures où c'est à « on », le nombre d'heures où c'est à « off », ainsi que les dix dernières valeurs.



Maison Connectée		
	Valeur Actuelle	Off
	Heurs à On (Tot)	625
	Heurs à Off (Tot)	1,772
	Dernieres Valeurs	
1	27-01-2017 à 15:01:52	Off
2	27-01-2017 à 15:01:43	On
3	27-01-2017 à 14:01:42	Off
4	27-01-2017 à 14:01:40	On
5	27-01-2017 à 14:01:55	Off

Figure 3-17 Interface « plus d'information » sur une entité.

6. Conclusion

Ce présent chapitre a été consacré aux différents outils matériels et logiciels ayant contribué à l'aboutissement de ce projet. Ensuite une vue détaillée du système (cartes et composants) et de l'application (architecture, hiérarchie, entités...etc.) a été mise en évidence. Enfin nous avons conclu par les interfaces permettant de contrôler différentes entités du système.

Conclusion générale

Notre étude des systèmes impliqués dans l'implémentation de maison connectée nous a permis de mieux comprendre les composants et technologies concernés par ce domaine de l'internet des objets. Elle nous a introduits à l'environnement mobile et aux applications multiplateformes ainsi qu'au monde de l'open-hardware, notamment celui de l'Arduino. La réalisation a été initiée par une mise en évidence de la problématique, l'objectif, et les fonctions du futur système, puis nous avons établi une analyse détaillée de notre application, en utilisant UML, qui est un standard incontournable dans la discipline de la conception logiciel. Que ce soit pour une maison connectée ou pour tout autre projet d'informatique embarquée, la connaissance de l'environnement Arduino est nécessaire pour implanter ce genre de projet. Nous avons par conséquent dans le dernier chapitre, présenté les différentes cartes électronique ainsi que leur environnement logiciel permettant de les programmer.

De ce travail, deux constats peuvent être tirés. Premièrement L'électronique commence à se démocratiser avec les concepts de l'open-hardware et de l'Arduino en essayant au maximum d'éviter les longs discours trop technique et en réduisant cela à des bibliothèques prêtes à l'emploi (tel a été le cas avec l'intégration des capteurs de température et d'humidité).

Le deuxième constat est que pour ce genre de projet, nous avons constaté que différentes disciplines peuvent y figurer : de l'administration système, de la programmation front-end, de la programmation back-end, ainsi que des compétences sommaires en électronique.

Bien évidemment, des améliorations futures peuvent être apportées à ce projet pour qu'il soit versatile, plus fiable, et adaptable. En perspective nous prévoyons :

- Différentes expériences utilisateurs : des interfaces utilisateurs adaptées à chaque type de terminal (tablette, smartphone, desktop, et pourquoi pas une Smartwatch) pour améliorer l'expérience utilisateur.
- Personnalisation de l'application : l'utilisateur pourra personnaliser son interface selon ses besoins.
- Unifier le serveur virtuel et l'Arduino : cela consiste à l'obtention d'une carte unificatrice intégrant de la programmation bas niveau (pour programmer les capteurs), et haut niveau (pour programmer un serveur http, une base de données...etc.). parmi les exemples: Arduino Yun, Raspberry Pi, Beaglebone Black, LinkItOne, et intel Galileo.
- Intégrer un système de notification : par exemple l'utilisateur sera alerté quand la température est trop basse, ou quand il y a de la fumée dans la maison.

Webographie

[w1] : http://readwrite.com/2012/02/07/infographic_history_of_mobile_app_stores-2/, Mars 2017.

[w2] : <http://fr.yeeply.com/blog/realite-augmentee-dans-applications-mobiles/>, Mars 2017.

[w3] : http://www.cisco.com/c/dam/global/en_ca/solutions/executive/assets/pdf/internet-of-things-fr.pdf, Avril 2017.

[w4] : <http://www.voxinternet.org/IMG/pdf/IdO.pdf/>, Avril 2017.

[w5] : <http://www.journaldunet.com/solutions/expert/64225/insecurite-des-objets-connectes---comment-conjuguer-l-iot-et-la-securite.shtml>, Avril 2017.

[w6] : <http://fing.tumblr.com/post/7359236108/les-d%C3%A9rives-des-villes-intelligentes>, Avril 2017.

[w7] : <http://www.lagrottedubarbu.com/2010/07/14/open-source-hardware-oshw-ebauche-de-definition-traduction/>, Avril 2017.

[w8] : <https://www.gralon.net/articles/internet-et-webmaster/logiciel/article-open-source---definition-et-histoire--1803.htm>, Avril 2017.

[w9] : <http://www.futura-sciences.com/tech/definitions/technologie-systeme-embarque-15282/>, Avril 2017.

[w10] : http://www.edurobot.ch/?wpfb_dl=139, Mai 2017.

[w11] : <https://www.arduino.cc/en/Main/ArduinoBoardYun>, Mai 2017.

[w12] : <https://www.carnetdumaker.net/articles/utiliser-un-capteur-de-temperature-et-dhumidite-dht11-dht22-avec-une-carte-arduino-genuino/>, Mai 2017.

[w13] : <https://itechnofrance.wordpress.com/2013/03/12/utilisation-du-module-ultrason-hc-sr04-avec-larduino/>, Mai 2017.

[w14] : <https://openclassrooms.com/courses/perfectionnez-vous-dans-la-programmation-arduino/decouvrez-le-shield-internet-et-le-lecteur-de-carte-sd>, Mai 2017.

[w15] : http://tiptopboards.free.fr/arduino_forum/viewtopic.php?f=2&t=22, Mai 2017.

[w16] : <http://php.net/manual/fr/intro-what-is.php>, Juin 2017.

[w17] : https://doc.ubuntu-fr.org/ubuntu_server, Juin 2017.

Bibliographie

[B1] : Christophe GILBERT, conception d'interface pour mobiles- graphisme et développement des applications natives, web et hybrides, Edition ENI, février 2017.

[B2] : Serge Darrieumerlou, le guide de la maison et des objets connectés, Edition Eyrolles, 2016.

[B3] : Pierre Mayé, edition Dunod, Aide-mémoire composants électronique, Dunod, janvier 2015.

[B4] : J.Gabay, Dunod 1ere edition, UML2 en action de l'analyse des besoins à la conception, 2016.

[B5] : Gilles Roy, conception de bases de données avec UML, presses de l'université du Québec 1ere édition, 2009.

[B6] : j.Chaffer et K.Swedberg , JQuery simplifiez et enrichissez vos developpements JavaScript, édition Pearson, 2009.

[B7] : Sébastien Pittion et Bastien Siebman, Applications mobiles avec Cordova et Phonegap, edition Eyrolles, mai 2015.

Résumé

Le travail réalisé dans ce présent mémoire s'articule autour de l'internet des objets, ses domaines d'application, ainsi que les technologies impliquées pour la mise en œuvre d'une solution IOT. Parmi les domaines d'application de l'internet des objets, il existe celui des maisons connectées, c'est ce cas particulier que nous avons traité tout au long de ce projet. Nous avons proposé une solution de maison connectée basé sur des technologies open-hardware (Arduino et compatibles). Afin de contrôler le système en local via un réseau LAN, ou à distance par le biais d'internet Le développement d'une application mobile multiplateforme (développée à l'aide des technologies du web) nous a été inévitable.

Mots clés: internet des objets, web des objets, maison connectée, matériel-libre, application mobile, capteurs, actionneurs, domotique, technologies web, contrôle à distance, électronique, multiplateforme.

Abstract

This project is about the internet of things, areas where we can find it, and some of the technologies implied in this field. Through this project we focused on smart houses, which is one of the fields of IOT. Our solution for smart houses is based on open-hardware technologies (Arduino and compatibles). In order to control the system locally or remotely, we developed a cross-platform mobile application based on web technologies.

Keywords: internet of things, web of things, smart house, open-hardware, mobile application, sensors, actuators, home automation, web technologies, remote control, electronics, cross-platform.