UNIVERSITÉ CÔTE D'AZUR

MBDS

TPT Projet semestriel

MASLOW

Maison connectée

Nom des membres : Zakaria GASMI, Kamel BOUREK, William POITEVIN
Encadrant : Mr Grégory Galli

Table of Contents

Maslow est un projet domotique visant à concevoir une maison intelligente grâce aux outils technologiques à notre disposition, ces outils peuvent être des composants matériel (hardware) ou immatériel (logiciels, programmes, outils conceptions et de développement, nos compétences).

Lors de ce projet notre équipe a travaillé afin d'apporter de nouvelles dimensions et de nouveaux schémas à ce projet notamment l'aspect sécurité, couche d'automatisation, la gestion d'accès (ouverture serrures), une appli mobile (émulant un lecteur QR code de porte d'accès), ainsi qu'une application web pour le contrôle des objets connectés.

La méthodologie utilisée s'est basée sur la méthode AGILE avec des livraisons de code d'une manière plus ou moins régulière en un laps de temps très réduit afin d'avoir un travail coordonné avec tous les membres de l'équipe.

Grace au travail des membres, on a obtenu des résultats très satisfaisants, on a réussi a faire un contrôle d'accès avec QR code, la gestion des objet connectés, la possibilité d'intégration de différentes plateforme d'iot (Tuya dans notre projet mais aussi Openhab et le champs reste ouvert), et l'intégration de dispositifs électronique Arduino ESP32 avec ces composants de type thermomètre, relais de contrôle 220V, ainsi qu'un photo résistor.

Nous avons implémenté des technologies diverses allant des technologies web Angular, .NET jusqu'à l'Android et MQTT ainsi que du C++

Introduction générale

De nos jours la technologie est utilisée dans des domaines de plus en plus nombreux, allant des taches simples aux plus complexes, la domotique est devenue un vrai sujet de travail afin d'apporter plus de confort aux résidents et d'améliorer considérablement la qualité de vie.

La domotique peut notamment être utilisé à des buts divers comme par exemple le domaine d'assistance aux personnes et perte d'autonomies (ex-EHPAD), nous avons choisi ce projet car aujourd'hui se secteur représente des réelles opportunités car le besoin est grandissant, faire ce projet peut être une base a des projets de domotique dont les spectres sont très larges.

Nous avons donné une priorité a la domotique pour son aspect technologique pointu et qui renforce nos compétences techniques et managériales.

Présentation du Projet

Lors de ce projet on aborde la domotique comme base concrète pour réaliser des maisons intelligente (Smart Homes), la domotique est un domaine innovant de part les technologies utilisées ainsi que les méthodes appliquées pour sa conception et sa réalisation.

Concrètement le but est d'avoir un écosystème fonctionnel qui nous permet d'accéder aux résidences d'une manière sécurisé avec une ouverture automatique (avec QR code), du contrôle de lumière et de divers objets d'une manière digitale, ainsi que l'autogestion et l'optimisation en termes d'Energie et de confort de vie (contrôle de température automatique, extinction des lumières si n'as pas utilisés etc.)

Les principaux enjeux sont la grande valeur ajoutée de ce projet notamment la possibilité de l'utiliser pour but d'assistance aux personnes handicapés, fragiles ainsi que l'aspect environnemental et économique.

Concernant les risques qui sont bien évidemment comme toute solutions présentes comme des dysfonctionnements probables restent très mineurs par rapport aux gains apportés

Etat de l’art sur le sujet traité

Le marché de la domotique connaît un véritable essor, profitant même en plus ces derniers temps de l'attrait pour l'équipement de la maison avec le confinement.
Selon la prévision du Digital Market Outlook le chiffre d’affaires mondiale devrait atteindre 88 milliards d’euros l’année en cours

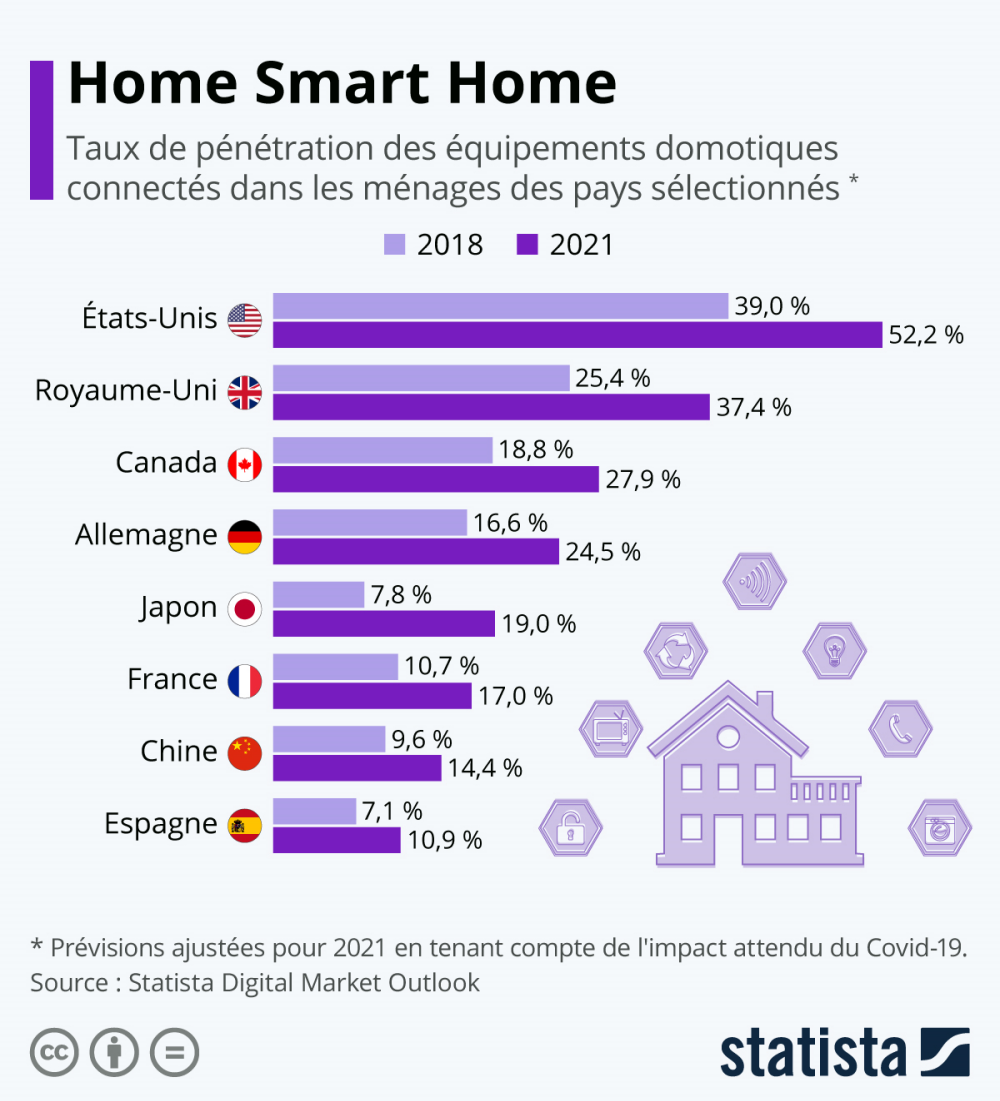


Figure 1

Sur le graph ci-dessus on note que le taux de pénétration des équipements (correspond à la vente d'équipements connectés) domitiques a progressé de 5 points et qu’il est assez bas pour pouvoir voir une opportunité et qui peut rapidement évoluer et dépasser le stade de marché de niche.

L’écosystème domotique regroupe quelques entreprises activant sur ce marché et dont le but est de mettre en place des maisons connectées comme l’entreprise immotik ou bien Nively qui

active dans une niche de la domotique médicale (ou M.GASMI a réalisé son stage alternance de M1 chez eux), ainsi que Netatmo.

Nous on offre dans l'ensemble des solutions plus ou moins similaires avec bien sur des manques du a la précocité de maturation du projet ainsi que le temps a disposition comparé aux professionnels du secteur.

Critères de comparaisons :
Afin d'avoir une comparaison homogène on définit les critères de comparaison pour ainsi identifier nos forces et faiblesses par rapport a nos concurrents, et pour cela on va essentiellement se baser sur les critères suivants :

- Type de clientèle
- Sécurité
- Environnement
- Offre

Nively : l'entreprise agit dans le secteur des EHPADS et c'est une entreprise de BtoB qui traite directement avec des personnes morales, leurs cibles sont les personnes âgées et de facto les entreprises de ce secteur.
La sécurité est leur valeur métier car ils communiquent sur cet aspect de sécurité des patients, concernant l'aspect environnemental il n'a clairement pas défini dans leur cahier des charges (économies énergie etc).
Leur offre est assez complète et ne représente pas de concurrence directe mais peut un jour le devenir (en tant que produit de substitution)

immotik: l'entreprise agit dans le secteur des maisons connectées et c'est une entreprise de BtoC qui traite directement avec des clients du grand publique, leurs cibles sont les personnes le grand publique de la génération y majoritairement qui sont assez a l'aise avec la technologie.

La sécurité est un point fort, ils disposent d'un système robuste de vidéosurveillance et anti-intrusion, concernant l'aspect environnemental ils ont la gestion de l'énergie et de l'électricité.
Leur offre est très complète et représente une concurrence directe.

	Cible	Sécurité	Environnement	Offre
Nively	Entreprises : EHPADS	Maintien a domicile, Monitoring , Alerte	N/A	Monitoring personnes agées
immotik	Public : Gen Y	Surveillance , Alarme, Anti intrusion	Energie, Lumiére	Smart Homes
Netatmo	Public : Gen Y	Surveillance , Alarme, Anti intrusion, Incendie	N/A	Maison connectés sécurisées

Maslow	Public et Entreprises	Electricité, Surchauffe, Acces QR code login pass sur application	Energie, Lumiére	Smart Homes, Bueaux, commerces

Etude de l'existant et solution envisagée

Etude de l'existant

Description externe du système logiciel existant

Maslow permet à un utilisateur de s'authentifier et d'effectuer des actions sur les serrures connectés The Keys déjà configurées et sur les objets connectés OpenHab. L'utilisateur administrateur peut configurer les comptes utilisateurs, les rôles et les privilèges sur les objets OpenHab.

L'accès à l'interface se fait via une page web responsive.

Description interne du système logiciel existant (vision développeur/conception)

Le projet que nous avons récupéré repose sur l'architecture suivante :

- Une base de données MongoDB (hébergée localement)
- Un serveur OpenHab 2.5.3 (hébergé localement)
- Une application serveur backend basée sur LoopBack 4
- Une application client frontend basée sur Ionic 4.11.13 avec React

Une partie de l'application est basée sur la serrure connectée The Keys. Elle est connectée à sa gateway qui communique avec l'API propriétaire (https://api.the-keys.fr).

Cette API permet de contrôler les serrures de la marque, et notamment l'ouverture et la fermeture.

Le backend repose sur le framwork Loopback 4 (https://loopback.io/) et la base de données MongoDB. Il s'appuie sur une architecture MVC.

Les accès aux différentes routes protégées sont bien configurés, sauf pour les serrures connectées qui sont accessibles sans identifiant.

La gestion des rôles et des utilisateurs a été implémentées. L'api utilise un JWT pour sécuriser les accès utilisateurs.

La gestion des objets connectés est séparée en deux. Il existe une route pour les objets connectés à OpenHab et une route pour les serrures de The Keys. Ces routes effectuent des appels aux API OpenHab et The Keys. L'API de MASLOW reprends la notion d'objet (ex : une lampe) et de choses (ex : une passerelle) pour l'implémentation du connecteur OpenHab.

L'implémentation des privilèges est faite pour les objets provenant d'OpenHab, mais elle n'est pas en place pour les serrures connectées The Keys. Elle applique les privilèges au niveau des choses qui permettent de gérer plusieurs objets en même temps.

Le frontend est basé sur Ionic, en vue de faire des applications mobiles à partir de ce développement. React est utilisé pour faire l'interface web du client. L'application s'appuie uniquement sur l'API de MASLOW.

L'interrface utilise la librairie Bootstrap-React qui gère notamment le responsive design, les thèmes claires et sombres et fournis des composant graphiques.

Cette interface gère les connexions et les droits utilisateurs. Elle permet d'afficher la liste des objets connectés (Serrures The Keys et objets OpenHab) et d'interagir avec eux (Ouverture/Fermeture, Allumage/Arrêt).

En cas de connexion avec un compte administrateur, l'interface permet de procéder à des paramétrages de comptes utilisateurs, de groupes d'utilisateurs et de privilèges sur les objets OpenHab.

Critique de l'existant

La configuration du projet frontend et backend (node_module) posait des problèmes pour la mise en route. Seul le gestionnaire Yarn (https://yarnpkg.com/) a permis de lancer le projet. Le docker compose fourni avec le projet ne nous a pas permis de le lancer.

Le choix d'un framework comme Loopback pour le backend à permit de structurer le projet avec un système MVC et une gestion des identifications via JWT, mais la gestion des dépendances et son système de montée en version pose des problèmes sur la reprise du projet. De plus, il apporte une complexité inutile à une API qui doit rester simple.

Le model du projet n'est pas correct pour la gestion des objets connectés. Il existe une implémentation d'API pour les serrures The Keys et une implémentation pour les objets OpenHab.

De ce fait, la gestion des objets n'est pas unifiée et ne permet pas d'envisager l'intégration d'une nouvelle source d'objets connectés. Dans le projet existant, la prise en charge des serrures connectés n'est pas sécurisée à cause d'une erreur de configuration de l'API. Le système de privilège ne prend pas en charges les serrures.

Le système de rôle est totalement paramétrable via API, mais les privilèges administrateurs sont configuré en dur dans le code. On peut facilement planter le système avec une fausse manipulation. Il n'est pas possible de configurer les droits pour le rôle « Urgence », car ses droits sont également appliqués en dur dans le code.

L'utilisation de Ionic pour l'interface apporte beaucoup d'avantage pour le développement d'application mobile, bien qu'ils ne soient pas exploités à ce stade du projet.

L'ergonomie du site n'est pas bonne sur ordinateur et l'interface réagit mal sur des petits écrans. L'outil de gestion de thème est mal géré, car sur un système sombre, les formulaires ne sont pas lisibles.

Certaines pages plantes (Ex : gestion des rôles) et la navigation est peu pratique, la touche retour du navigateur ramène systématiquement à l'écran d'accueil.

Solutions envisagées

Deux solutions sont possibles pour la suite du projet. La première option est de reprendre le développement existant avec une mise à niveau du projet et une refonte de l'architecture du backend. La seconde est de recréer une nouvelle solution à partir de zéro.

Nous avons privilégié la seconde piste, car la dette technique du projet est très importante et la mise à niveau du webservice a un coût très élevé.

Démarche projet

Principes de la démarche projet

Activités d'ingénierie logicielle

Lors de l'attribution de projet, on a réalisé des activités d'ingénierie notamment lors de la phase de d'étude de l'existant ou on a étudié ce qui était déjà en place, en relevant les points forts et faibles.

On a de par la suite pris connaissances des exigences déjà en place mais pas réalisés, on y a ajouté des exigences qui nous semblaient pertinents et avons pensé ce qui devait être fait pour que se projet soit plus ou moins complet.

Après ces 2 phases, on s'est penché sur l'aspect conception ou on devait mettre tout les outils en place et penser a une solution qui tiendras dans le temps, en concevant les diagrammes UML et le choix des langages à utiliser.

Puis survient la phase de réalisation, on a mis en pratique la conception réalisée et on c'est adapté aux réalités du terrain pour avoir une solution adéquate.

Les tests se faisait au fur et à mesure du développement, par un membre qui n'as pas fait le développement de la fonctionnalité.

Méthode de gestion de projet utilisée

La méthodologie utilisée s'est basée sur la méthode AGILE avec des livraisons de code d'une manière plus ou moins régulière en un laps de temps très réduit afin d'avoir un travail coordonné avec tous les membres de l'équipe.

Toute l'équipe disposait de réunions fréquentes, voir même un travail en conférence afin d'avoir une coordination optimale en prenant en compte les délais réduits dont nous disposions.

Chacun des membres se voyait affecté une partie métier qu'il devait faire évoluer par ex partie Web, partie IOT partie backend, et en même temps tout le monde s'entraidait sur les blocages de chacun afin d'avancer rapidement sur les taches.

Rôles et responsabilités

Chaque membre s'est vu attribuer les rôles suivants :

Kamel BOUREK : responsable des service web et de l'implémentation

Zakaria GASMI : responsable des objets connectées et embarquées
William POITEVIN: responsible backend .NET

Outils

Les outils pour la réalisation du projet sont présentés comme suite :

Outils de conception :

- UML : Diagramme de classe, Diagramme des séquences et use case ;
- IDE dev (Visual Studio pour .NET, Visual Studio code pour Angular et Nodejs, Android Studio pour l'appli android, Arduino IDE pour arduino)
- Langage de programmation : Type Script, C# , C++, JavaScript Nodejs

Ces outils ont été choisis après réunion des membres et pour faciliter l'atteinte des objectifs.

Gestion de la configuration

Toute l'équipe travaillait en simultanée sur les différents aspects du projet, les sauvegardes se faisaient en utilisant les outils de vesionning GIT notamment GitHub afin de préserver le travail.

Les livraisons se faisaient au fur et a mesure de l'atteinte des sous objectifs (ex : faire fonctionner l'api de tuya en développant une api locale en NodeJs).

Quant aux règles de nommage on a utilisé des noms explicite, et utilisant du PascalCase pour les classes ainsi que les camelCases pour les variables et fonctions.

Contraintes et risques sur le projet

N° Risque	Libelle du risque	Importance	Facteur contribuant	Solution proposé	Statut
1	Ne pas pouvoir faire le projet avec des objets connectés (donc ne pas réaliser pouvoir mener à bien le projet)	Critique	_ Manque d'objets _Objets reçu non adaptable à Openhab	_ Utiliser une solution Open source sur GitHub pour la gestion des objets tuya en local _ Utiliser Arduino et des objets connectés dessus comme un thermomètre, un relais 220V	Réalisé
2	Ne pas finir le projet dans les temps	Haute	_Reception d'objets en retard _Manque de temps (autre projets) _solution précédente ne démarre pas	_Travailler les nuits et weekends _ Diviser le travail entre les membres	Réalisé

Démarche projet mise en œuvre

Backlog initiale :

- Analyse du projet existant et de l'architecture (3j.h)
- Lancement du projet et correction de bugs pour mise en marche (3j.h)
- Analyse de la possibilité de reprendre des parties existantes (3j.h)

- Conception de la nouvelle architecture (diagrammes) (3j.h)
- Spécifications techniques (2j.h)

- Implémentation part 1 (backend) (3j.h)
- Implémentation part 2 (front end) (3j.h)
- Implémentation part 3 (IoT) (3j.h)

Sprint 1 : (23/02/2021 - 07/03/2021) :

- Analyse du projet existant et de l'architecture Part1 (1j.h) - William
- Analyse du projet existant et de l'architecture Part 2 (1j.h) - Kamel
- Analyse du projet existant et de l'architecture Part 3 (1j.h) - Zakaria

- Lancement du projet et correction de bugs pour mise en marche Part 1(1j.h) - William
- Lancement du projet et correction de bugs pour mise en marche Part 2 (1j.h) - Kamel
- Lancement du projet et correction de bugs pour mise en marche (1j.h) - Zakaria

- Analyse de la possibilité de reprendre des parties existantes (1j.h) - William
- Analyse de la possibilité de reprendre des parties existantes (1j.h) - Kamel
- Analyse de la possibilité de reprendre des parties existantes (1j.h) - Zakaria
- Réunion avec Grégory GALLI (1j.h)

Sprint 2 : (07/03/2021 - 20/03/2021) :

- Conception de la nouvelle architecture (diagrammes) (1j.h) - William
- Conception de la nouvelle architecture (diagrammes) (1j.h) - Kamel
- Conception de la nouvelle architecture (diagrammes) (1j.h) - Zakaria
- Réunion avec Grégory GALLI (1j.h)

- Spécifications techniques (1j.h) - William
- Spécifications techniques (0.5j.h) - Kamel
- Spécifications techniques (0.5j.h) - Zakaria

Sprint 3 : (20/03/2021 - 30/03/2021) :

- Implementation part 1 (backend) (3j.h) - William
- Implémentation part 2 (front end) (3j.h) - Kamel
- Implémentation part 3 (IoT) (3j.h) - Zakaria

Planification

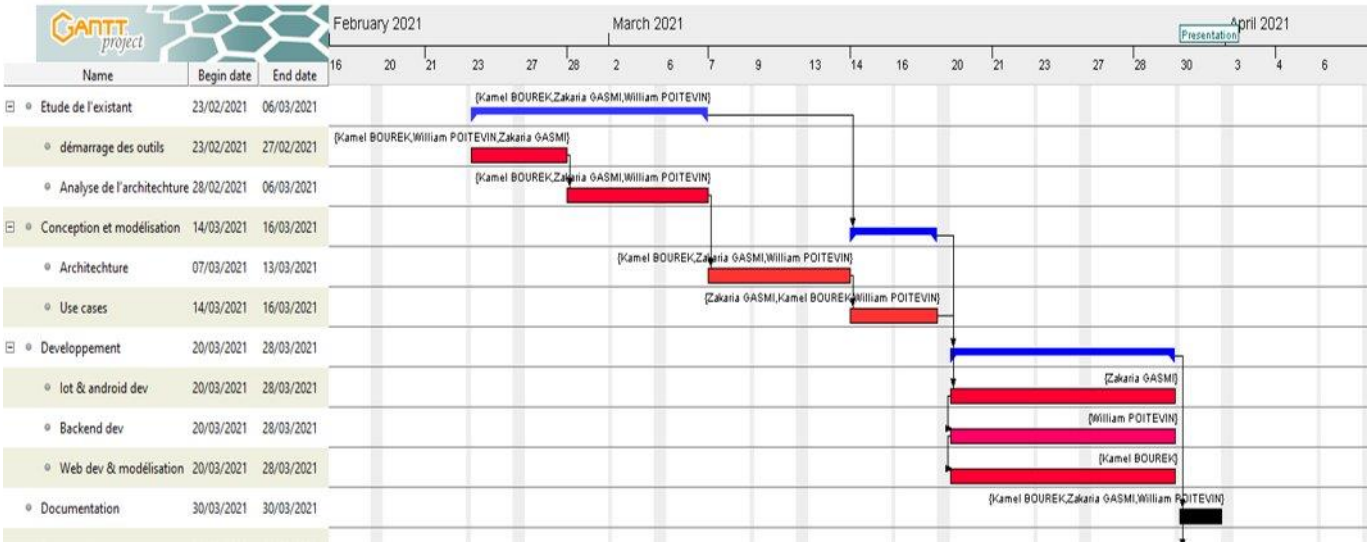


Figure 2

Budget du projet

Le projet Maslow a besoin de matériel pour reproduire une maison connectée :

- Une serrure connectée et sa passerelle : 200€
- Une enceinte connecté Alexa : 50€
- Deux prises connectées : 25€
- Deux lampes connectées : 25€
- Un thermomètre connecté : 20€
- Un Raspberry PI 3B complet : 70€
- Un routeur Wifi de voyage et câbles Ethernet : 40€
- Une carte programmable ESP32 avec capteurs : 20€
- Un relai 220v : 10€

Le budget total pour le matériel lié au projet est de 460€.
Il faut ajouter à cela les machines de travail utilisées pour le développement et le cout des ressources affectées au projet :

Charge de travail :

- Développeurs (21 j.h)
- Grégory GALLI (2 j.h)

Exigences réalisées dans le projet (vision externe/utilisateur)

Exigences fonctionnelles – Cas d'utilisation

Cas d'utilisation 1 : Scanner un code QR pour ouvrir la serrure

1. L'utilisateur scanne le code QR avec l'application Android
2. L'application Android s'authentifie auprès du Backend
3. L'application Android envoie une action vers le backend MASLOW
4. Le backend MASLOW envoie le Login à l'API The Keys
5. L'API The Keys renvoie une autorisation
6. Le Backend MASLOW envoie l'ordre à la serrure via The Keys
7. La Serrure exécute l'action
8. La Serrure renvoie son état vers l'application via l'API et le Backend.

Cas particulier :
L'utilisateur n'est pas authentifié, le système invite le client à présenter un code QR valide. Retour à l'étape 1.

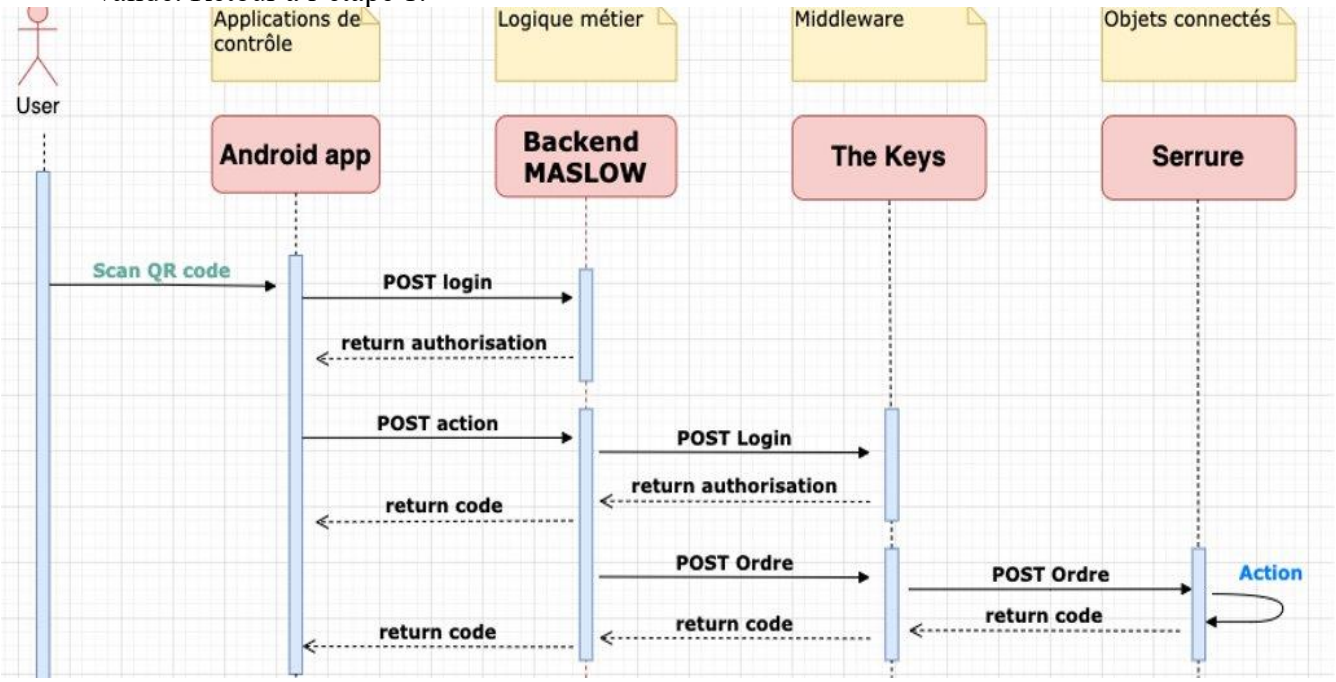


Figure 3

Cas d'utilisation 2 : Ouvrir/fermer la serrure avec la Web App

1. L'utilisateur se connecte à l'application web
2. Le système authentifie l'utilisateur
3. L'utilisateur clique sur le bouton dédié à la serrure
4. L'application envoie l'action au backend MASLOW
5. Le Backend demande l'autorisation à l'API The Keys
6. Le Backend est autorisé et envoie l'ordre d'ouverture/fermeture
7. La Serrure s'ouvre/ se ferme
8. L'API The Keys envoie l'état de la serrure à l'application WEB
9. L'application Web affiche l'état de la serrure

Cas particulier :
L'utilisateur n'est pas authentifié, le système invite le client à entrer des identifiants valides. Retour à l'étape 1.

Cas d'utilisation 3 : Allumer/ éteindre la lampe avec la Web App

1. L'utilisateur se connecte à l'application web
2. Le système authentifie l'utilisateur
3. L'utilisateur clique sur le bouton dédié à la lampe
4. L'application envoie l'action au backend MASLOW
5. Le backend envoie l'ordre à l'API Tuya
6. La Lampe s'allume/ s'éteint
7. L'API Tuya envoie l'état de la lampe à l'application WEB
8. L'application Web affiche l'état de la lampe

Cas particulier :
L'utilisateur n'est pas authentifié, le système invite le client à entrer des identifiants valides. Retour à l'étape 1.

Cas d'utilisation 4 : Allumer/ éteindre la prise connectée avec la Web App

9. L'utilisateur se connecte à l'application web
10. Le système authentifie l'utilisateur
11. L'utilisateur clique sur le bouton dédié à la prise
12. L'application envoie l'action au backend MASLOW
13. Le backend envoie une requête MQTT à l'Arduino
14. La Prise s'allume/ s'éteint
15. Le Backend envoie l'état de la prise à l'application WEB
16. L'application Web affiche l'état de la prise

Cas particulier :
L'utilisateur n'est pas authentifié, le système invite le client à entrer des identifiants valides. Retour à l'étape 1.

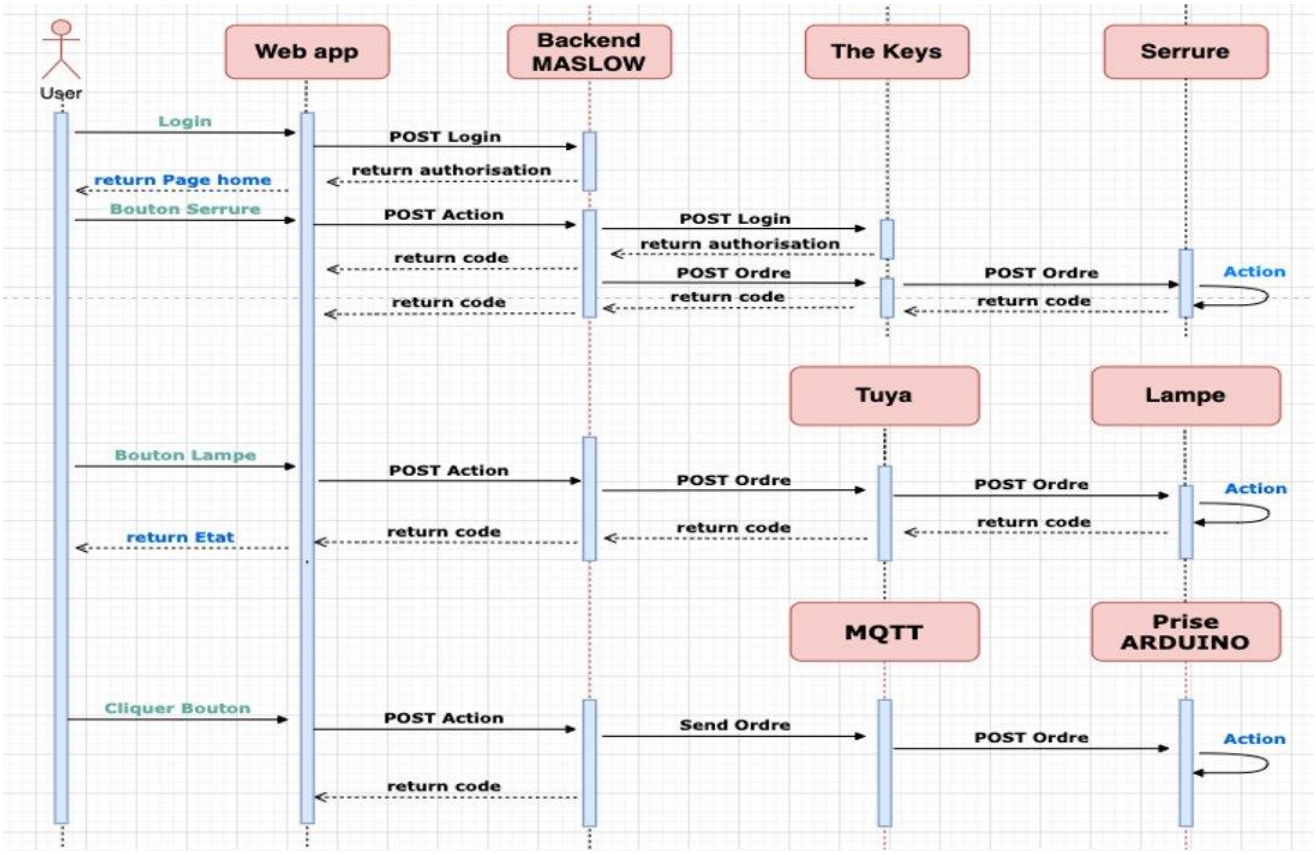


Figure 4

Exigences non fonctionnelles transverses

• **Utilisabilité** : Notre système est facile d'utilisation, il suffit d'ouvrir l'application Web, s'identifier et ensuite gérer la maison à l'aide des différents boutons dédiés à chaque fonctionnalité.

• **Performance** : Temps d'attente < 5s.
• **Fiabilité** : Système fiable et réponds toujours aux requêtes demandées.
• **Disponibilité** : 24/7
• **Sécurité** : Accès personnalisés, connexions sécurisées.
• **Maintenabilité** : Notre architecture est Modulaire, peu complexe, bien documentée.
• **Portabilité** : Utilisable avec plusieurs systèmes d'exploitation.

Interfaces détaillées

IHM

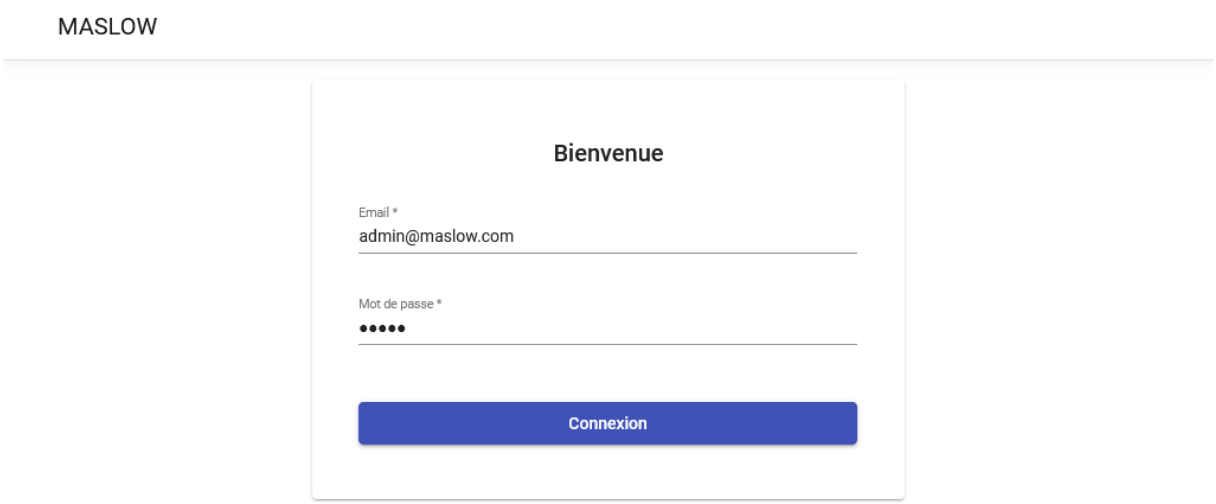


Figure 5

Interface de connexion

Figure 6

Interface de page d'accueil

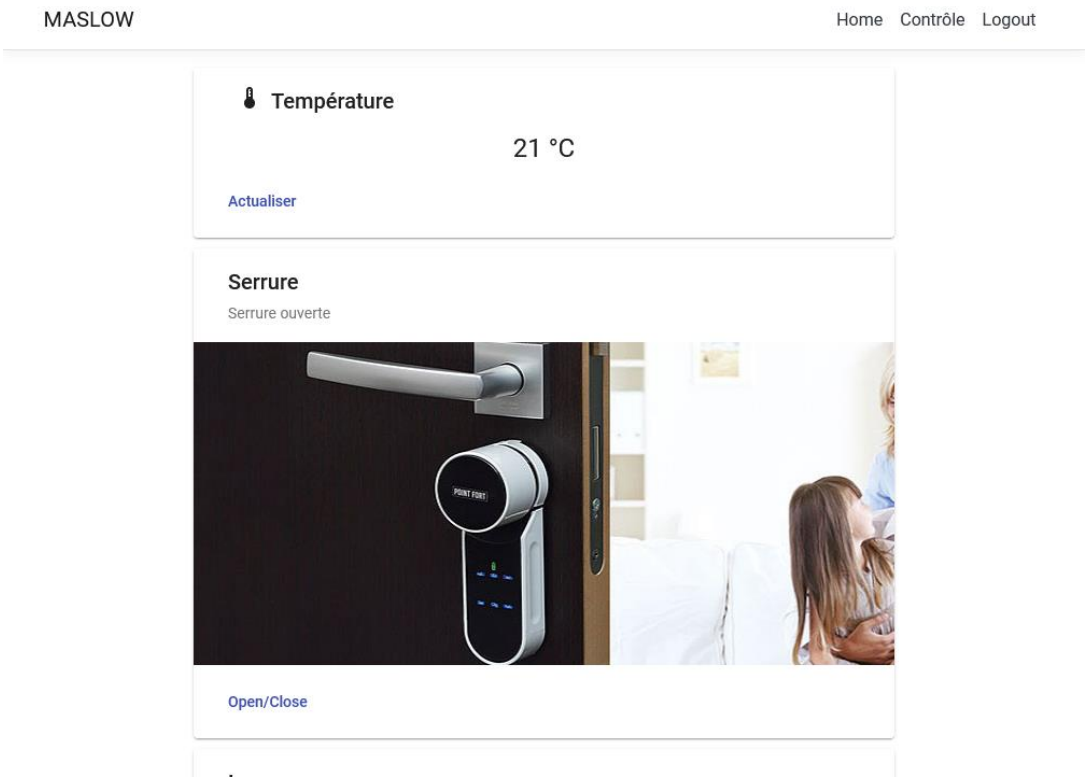


Figure 7

Interface de gestion des objets connectés

Interfaces avec d'autres systèmes

Le webservice Maslow est accessible par une API REST en JSON. L'authentification se fait via un token Bearer et l'API est entièrement testable grâce au SwaggerUI intégré au projet. Il est accessible à l'adresse « /swagger/index.html » :

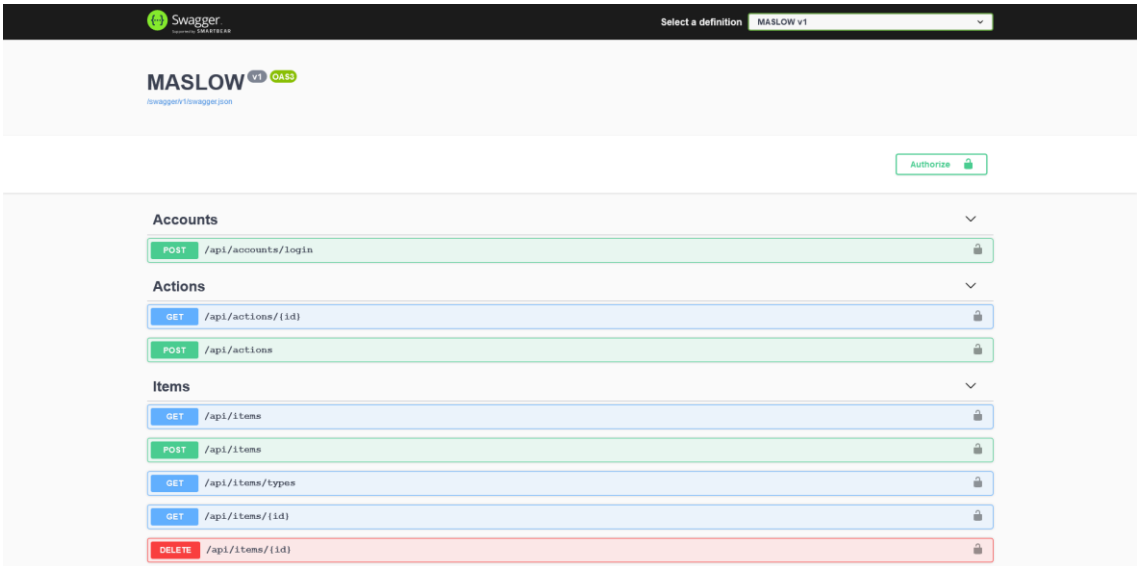


Figure 8

Voici les routes de l'API :

- Compte utilisateur
  - /api/accounts/login :

POST : Connexion au webservice. Retourne un token de connexion.

- Actions
  - /api/actions/{item}
    - GET : Liste les actions disponibles sur l'objet
  - /api/actions
    - POST : Permet de lancer une action sur un objet
- Objets
  - /api/items
    - GET : Liste des objets connectés
    - POST : Création d'un nouvel objet
  - /api/items/types
    - GET : Liste des types d'objets utiliser pour la création d'un nouvel objet dans l'API
  - /api/items/{id}
    - GET : Donne toutes les informations sur un objet connecté
    - DELETE : Supprime un objet connecté
- Capteurs
  - /api/sersors/{item}
    - GET : Liste des valeurs disponibles pour un objet connecté
  - /api/sersors/{item}/{value}
    - GET : Donne la valeur demandé

Architecture(s) système

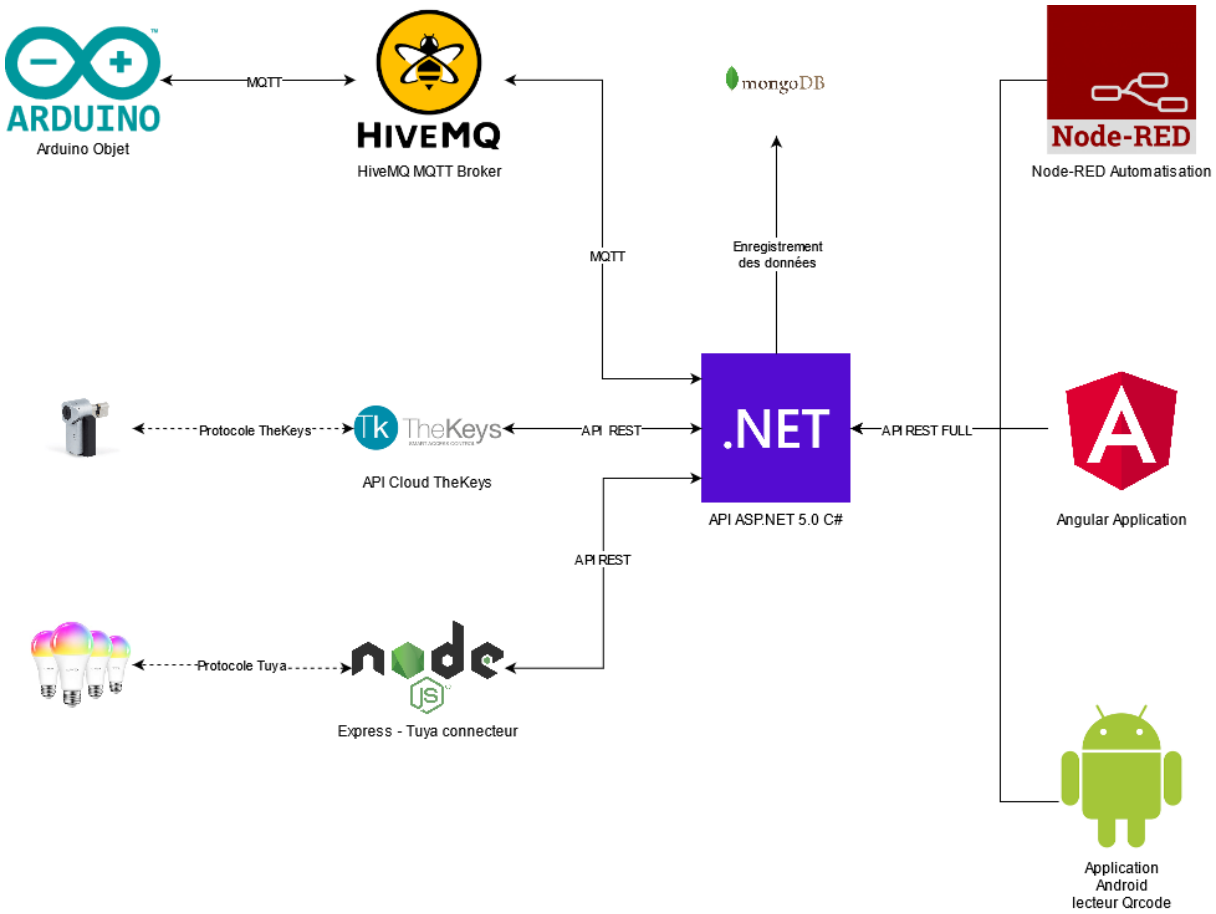


Figure 9

L'architecture de MASLOW s'appuie sur un serveur ASP.NET 5.0 en charge de communiquer avec les objets connectés d'une part et des applications qui interagissent avec les utilisateurs d'autre part. Ce webservice central sert de base au système pour fournir un service d'authentification, de gestion de rôles et permissions et permettre un accès unifié aux objets connectées.

La conception du webservice permet d'intégrer tous les objets connectés, quel que soit les méthodes de communication utilisée par les appareils. Cela est rendu possible par un système de plugins qui permet de développer un connecteur pour chaque protocole de communication et de les brancher au système noyau. Grace à ce principe, il est possible d'avoir une plateforme agnostique du matériel à contrôler.

La plateforme .NET fournit des outils pour mettre en place une API REST FULL avec une gestion des authentifications, ce qui permet de créer une plateforme qui peut être appelée par tous type de programmes comme des applications Web, des applications mobiles, des clients lourds ou mêmes des outils d'automatisation.

Le projet intègre une application web basée sur Angular 8, qui permet d'utiliser et de consulter les objets connectés. Elle permet de configurer le système et elle se base sur l'API

ASP.NET. L'application Android permet quant à elle de simuler un lecteur de QR code. Elle s'appuie sur l'API pour authentifier le QR code et ouvrir la porte.

Node-RED est une solution d'automatisation qui s'interface avec beaucoup de services existants (réseaux sociaux, courriel, logiciels, etc…) et qui est utilisée pour programmer des actions intelligentes aux données et aux actions fournies par le webservice ASP.

Conception du système logiciel réalisée dans le projet (vision interne/développeur)

Plate-forme technique

Nous avons choisi de refondre le projet en conservant le principe d'un webservice central. Nous avons choisi d'utiliser ASP.NET sur la dernière version 5.0 en lieu et place de Loopback 4. La plateforme .NET est pensée pour être multiplateforme et le projet générer permet de lancer un serveur web indépendant. Cela nous permet de développer, prototyper sur des machines Windows et MacOs et de déployer sur un Raspberry PI sous Linux.

Nous avons remplacé le frontend sous Ionic par une application Angular 11 qui est plus rapide à générer et qui s'intègre très bien sur le serveur offert par ASP.NET. De ce fait, c'est le serveur ASP qui distribue l'application Angular et nous permet de proposer une solution plus facile à déployer.

Nous avons ajouté l'application Android native codée en Java pour le lecteur de QR code. Cette solution permet de créer un lecteur à moindre coût et de concevoir un cas d'utilisation concret de bout en bout avec la serrure connectée. Elle nous a également permis d'exploiter les enseignements reçus.

Les développements liés à Arduino et aux objets Tuya sont de nouveaux ajouts nécessaires à l'ajout de nouveaux objets connectés mis à notre disposition.

Le code Arduino permet de contrôler le matériel connecté sur le microcontrôleur ESP32 et de communiquer grâce au protocole MQTT.

Le serveur Express pour Tuya permet de communiquer avec les objets en passant par une librairie « tuyapi » uniquement disponible sur NodeJS. Cette librairie permet de contrôler les objets Tuya sans passer par l'API Cloud payante.

Conception du logiciel développé

Conception du code source

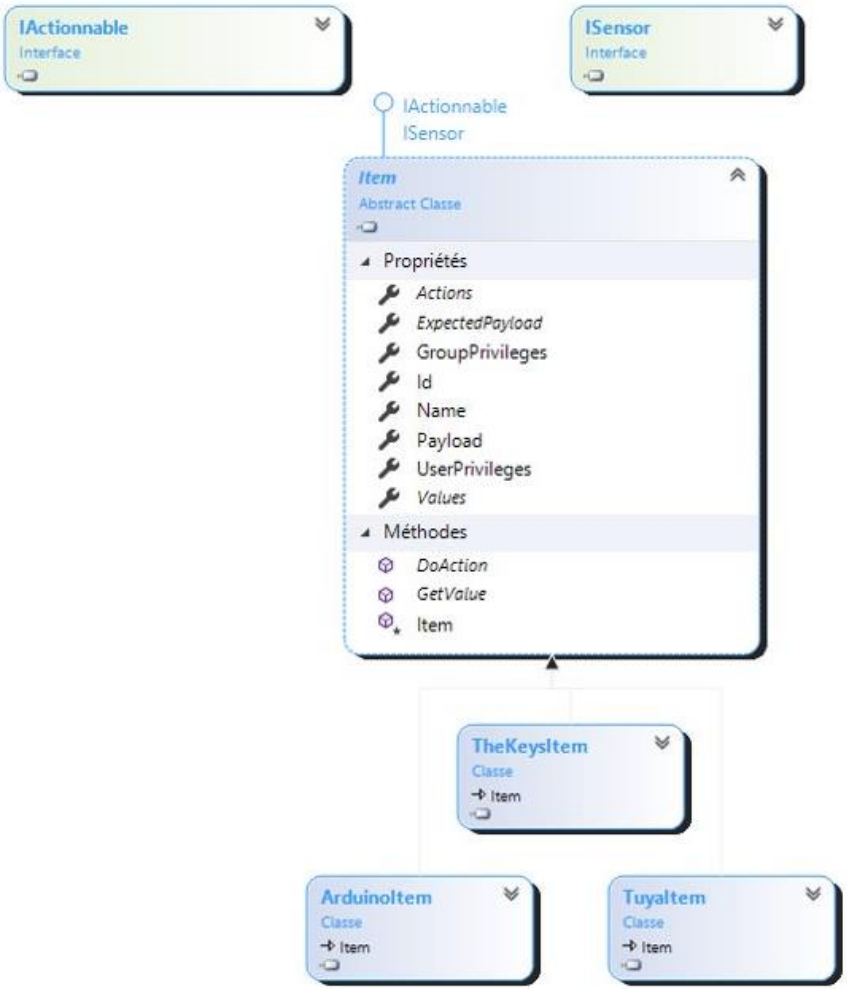
Nous avons découpé le webservice en plusieurs bibliothèque C#.NET pour permettre la mise en place d'un système de plugin. Les bibliothèques sont les suivantes :

- MASLOW : Projet ASP.NET
- MASLOW.Entities : Model de données. Utilisé dans toutes les librairies
- MASLOW.Arduino : Intégration des objets connectés Arduino.

- MASLOW.TheKeys : Intégration des objets connectés The Keys
- MASLOW.Tuya : Intégration des objets connectés Tuya

L'application MASLOW est basée sur le model MVC et intègre l'application Angular.

Modélisation de données



La modélisation de la base de données mongoDB liée au webservice ASP.NET est faite avec un ORM pour mongoDB.

Le système qui permet à MASLOW d'intégrer tous les types d'objets connecté repose sur les interfaces actionnables et les capteurs.

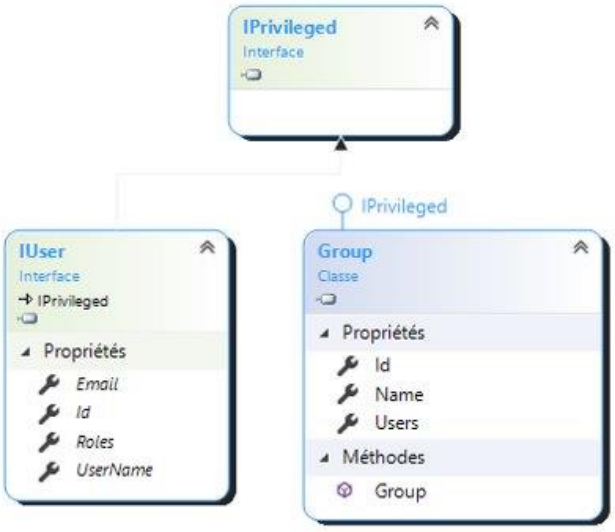
Nous avons créé une classe abstraite Item qui hérite des interfaces précédemment citées.

Cette classe Item doit être implémenté pour chaque nouveau type d'objet connecté à intégrer.

La gestion des droits sur les objets s'appuie sur des une classe privilège et permet de donner ou de refuser un droit à un groupe ou à une personne.

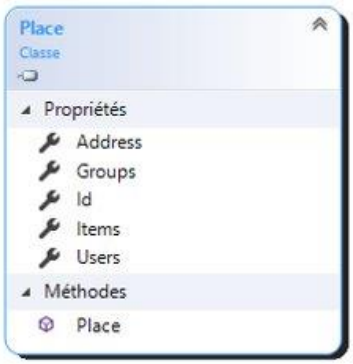
Le privilège peut porter sur un objet ou sur une action en particulier. Le privilège peut intégrer des capteurs pour mettre en place des conditions d'applications.

Cette partie n'est pas implémentée actuellement.



La gestion des utilisateurs s'appuie sur Identity Server qui permet d'intégrer la gestion l'authentification au sein d'ASP.NET. Elle utilise une classe utilisateur et une classe rôle.

Une importance est donnée au fait de ne pas permettre aux administrateurs de gérer la liste des rôles dans le système, car ils sont utilisés dans le code source. Une suppression ou une modification peut planter complétement la solution. Leur définition est faite au moment de la définition des règles métiers.



La place est la classe qui permet de gérer les objets connectés par lieu. Elle permet de créer des conditions valables sur un ensembles d'objets. Par exemple le propriétaire peut utiliser et gérer tous les objets de sa maison.

Les composants et leur déploiement

Le projet Webservice et Angular se compile grâce aux outils « dotnet » en version 5.0 et NPM. Le projet génère une DLL pour chaque bibliothèque, puis tout le projet de façon optimisé avec l'application Angular compilée et un exécutable qui permet de lancer le serveur web. Ce serveur sert l'API et l'application Angular sur la même adresse.

Tests du système logiciel

Nous avons effectué des tests fonctionnels :

Test	Résultats
Serrure avec QR Code	Ok, réponse instantanée
Serrure via Web App	Ok, après 5 secondes
Lampe via Web App	Ok, après 5 secondes
Prise Arduino	Ok, réponse instantanée
Login App	Ok
Log out App	Ok
Application Android	Ok

Conclusion générale

La conclusion est l'occasion pour vous de faire un bilan sur votre projet. Ce bilan doit comporter les parties suivantes :

- Le bilan des résultats obtenus pour l'entreprise
- Le bilan des problèmes rencontrés et des solutions apportés
- Les perspectives du projet
- Le bilan personnel

Le bilan des résultats obtenus pour l'entreprise

Nous avons changé l'architecture du projet afin qu'elle soit plus flexible est maintenable.
La solution actuelle fonctionne à l'aide de deux applications, une application Android pour ouverture de serrure grâce à un Code QR, une application Angular pour la gestion à distance des différents appareils connectés de la maison tels que la lumière, la gestion des accès et différents appareils branchés à une prise connectée.

Programmes réalisés :

- Application Angular
- Application Android
- Programme Arduino
- Serveur Tuya
- Système NodRed

A la fin du projet, nous avons réussi à faire marcher :

- 1 Lampe connectée Tuya
- 1 Prise connectée Tuya
- 1 Serrure connectée The Keys
- 1 Prise connectée sur Relai Arduino
- 1 Capteur de température
- 1 Capteur de luminosité

Le bilan des problèmes rencontrés et des solutions apportés

Nous avons rencontré quelques difficultés liées au manque de matériel ou à du matériel non compatible. Une fois que le problème de matériel a été résolu, nous nous sommes retrouvés avec un délai de réalisation très court, mais grâce à notre organisation et à notre détermination, nous avons pu nous dépasser et faire de notre mieux afin de réaliser un projet qui répond à la demande. Les fonctionnalités effectuées sont basiques, mais nous estimons qu'en terme de rapport temps /moyens, les résultats sont satisfaisants.

Les perspectives du projet

Différentes améliorations peuvent être apportées au projet :

- Gestion des droits utilisateurs
- Ajout d'appareils connectés
- Gestion de la température à distance
- Amélioration de l'interface
- Ajout dynamique d'objets
- Possibilité de création de compte utilisateur dans l'interface
- Hébergement de l'application

Le bilan personnel

Le projet nous a permis d'appliquer plusieurs technologies apprises pendant notre Cursus en MBDS et en Miage en générale telles que .Net et Angular.

Nous avons aussi appris à gérer la charge de travail en un temps court et à optimiser la production dans de pareilles circonstances.

Enfin, nous avons appris à manipuler des objets connectés et les différentes manières d'interagir avec ces derniers.

Table des figures :