

Université Hassan II de Casablanca

Faculté des sciences et techniques de Mohammedia

Département d'Informatique

Projet de Fin d'Etudes

**LICENCE SCIENCES ET TECHNIQUES
INFORMATIQUE, RÉSEAUX ET MULTIMÉDIA**

Sujet :

**Conception et développement d'un modèle des Smart
Homes**

Présenté par :

ESSEMRHOUNI Oumayma

ADIL Ayoub

CHBIRA Imane

Devant Les membres du jury :

Pr. LEGHRIS

Encadrant (FSTM)

Pr. M.KHALIL

Rapporteur (FSTM)

Pr. N.MOUMKINE

Examineur(FSTM)

Année universitaire 2019-2020

Remerciements :

Nous tenons à remercier notre entourage et toute personne ayant contribué au succès de ce stage.

Tout d'abord, nous adressons nos remerciements à notre professeur et encadrant **Pr. LEGHRIS**, qui nous a été d'une aide énorme durant cette période de stage grâce à sa disponibilité et ces bienveillants conseils, ainsi qu'au **corps professoral** de la Licence « IRM » pour leurs efforts.

Nous tenons à remercier **nos familles**, qui malgré cette période difficile du confinement, ont su nous fournir un bon environnement de travail nécessaire pour la réalisation de ce projet, aussi bien que des encouragements au quotidien.

Enfin, un grand merci envers nos **amis** qui nous ont apporté leur soutien moral et intellectuel tout au long de notre stage, et surtout ceux qui avec leurs motivation, amour et confiance, nous ont toujours poussé à devenir la meilleure version de nous-même.

Sans oublier nos **collègues** qui ont fait de cette année une belle aventure pleine de souvenir et d'accomplissement.

Résumé :

Ce document représente le rapport qui résume notre travail dans le cadre de notre projet de fin d'études en licence science et technique en « **Informatique, Réseau et Multimédia** » à la Faculté des Sciences et Techniques de Mohammedia.

Le but de ce projet est la conception et développement d'un modèle de smart home avec au moins cinq services, ainsi la mise en place d'une application web et mobile permettant de contrôler à distance ces derniers.

Ce rapport incarne le travail fourni dans projet, et nous introduit aux outils et technologies employées en plus du résultat obtenu.

Abstract :

For our end of study project, we -bachelors of **Computer Science, Network and Multimedia** at Faculty of Sciences and Techniques of Mohammedia- we thought first of our space, the place where we spent the last 3 months in quarantine. Our homes.

Not only we constructed a smart home model you can manage its services remotely, from your smart device, But also to give the user access to understand deeply the new to Morocco concept of a smart home, through an easy to use web app model for a pleasant user experience with 5 developed services the user can manage by a swipe of an index.

Table des matières

Remerciements :	II
Résumé :	III
Abstract :	III
Liste des figures :	VII
Liste des abréviations :	IX
Liste des tableaux :	X
INTRODUCTION GENERALE	XI
CHAPITRE 1 : Contexte général du projet	1
I-Organisme d'accueil :	2
II-Besoin actuel en IoT :	3
III-Cahier de charge :	3
1-Benchmarking :	3
2-Descriptions des besoins :	5
3-Objectif du projet :	5
IV-Planification et gestion de projet :	6
CHAPITRE 2 : Analyse et conception	8
I.Contenu du projet	9
1.Choix des services	9
2.Simulation des objets connectés	9
3.Espace SuperAdmin	9
4.Applications Web & Mobile	9
5.Système de notification	10
II.Etude conceptuelle :	10
1.Introduction	10
2.Les acteurs du système	10
3.Diagramme de cas d'utilisations :	11
4.Descriptions textuelles des cas d'utilisations :	12

5.Diagrammes de séquences :	16
6.Diagramme de classe :	19
CHAPITRE 3 : Etude technique et environnement de travail	20
I-Framework et technologies :	21
1-React :	21
2-React Native :	21
3-Node Js :	22
4-Flask :	22
5-React Navigation :	22
6-Firebase :	23
II-Environnement et outils :	23
1.GIT et GitHub :	23
2.VisualStudio Code :	24
3-Yarn :	25
4.Android Studio :	26
5-Star Uml :	26
6-Adobe Xd :	26
7-Ant Design :	27
CHAPITRE 4 : Réalisation	28
I.Simulation :	29
1.Modèle graphique des services de la maison Intelligente :	29
2.Comportement des objets connectés :	30
3.Flask API	32
II.Console : SuperAdmin :	34
1.Authentification :	34
2.Ajouter un Admin :	34
III.Application web responsive :	36
1.Système d'authentification :	36
2.Espace Admin :	38

IV.Application mobile : Android :	42
1.Introduction :	42
2.Scénario de l'expérience utilisateur :	42
Conclusion	45
Références	46

Liste des figures :

Figure 1. logo FSTM	2
Figure 2. Capture d'écran de l'application Trello	7
Figure 3. Diagramme de Gantt	7
Figure 4. Diagramme de cas d'utilisation	11
Figure 5. Diagramme de séquence « Authentification »	16
Figure 6. Diagramme de séquence « Chambre » :	17
Figure 7. Diagramme de séquence « Paramètres »	17
Figure 8. Diagramme de séquence « Main »	18
Figure 9. Diagramme de classes.	19
Figure 10. Logo React.	21
Figure 11. Logo React Native	21
Figure 12. Logo Node.JS	22
Figure 13. Flask.	22
Figure 14. Logo React navigation	22
Figure 15. Firebase logo	23
Figure 16. Logo Git	23
Figure 17. Screen d'un projet GitHub	24
Figure 18. Logo GitHub	24
Figure 19. Capture de Visual Studio Code.	25
Figure 20. Logo VS Code	25
Figure 21. Logo Yarn	25
Figure 22. Capture du fichier Yarn dans Vs code	25
Figure 23. Android Studio logo	26
Figure 24. Logo Star Uml	26
Figure 25. Logo Adobe Xd	27
Figure 26. Capture d'Adobe Xd	27
Figure 27. Logo Ant Design	27
Figure 28. Modèle graphique des services de la maison intelligente	29
Figure 29. Screen du code OutsideTemperature	30
Figure 30. Screen du code IsRain	30
Figure 31. Screen du code isMvt	31
Figure 32. Screen du code de l'état de lampe et Fenêtres	31
Figure 33. Screen d'une réponse sous forme JSON	32
Figure 34. Schéma d'architecture du systeme	33

Figure 35. Interface du login SuperAdmin _____	34
Figure 36. Interface d'ajout d'admin _____	35
Figure 37. Interface de Dashboard _____	36
Figure 38. Interface Landing page _____	37
Figure 39. Interface d'authentification _____	37
Figure 40. Interface d'accueil Responsive _____	38
Figure 41. Interface de paramètres _____	39
Figure 42. Interface responsive des chambres « User 1 » _____	39
Figure 43. Interface responsive des chambres « User 2 » _____	40
Figure 44. Interface Bedroom _____	40
Figure 45. Interface Living room _____	41
Figure 46. Interface Family _____	41
Figure 47. Interface About _____	42
Figure 48. Scénario de l'application mobile (part 1) _____	43
Figure 49. Scénario de l'application mobile (part 2) _____	44

Liste des abréviations :

1. IoT : Internet of things.
2. UML : Unified Modeling Language.
3. IOS : iPhone Operating System.
4. API : Application Programming Interface.

Liste des tableaux :

Tableau 1 :Acteur du système	11
------------------------------	----

INTRODUCTION GENERALE :

Pourriez-vous imaginer le contrôle de votre chez soi sans devoir se déplacer ou mettre fin à votre confortabilité ? Un clic, toutes vos fenêtres seront fermées, la porte clôturée, et les lumières éteintes ?

Et bien tout ceci est possible grâce aux objets connectés!

Que ce soit dans notre univers public, personnel ou professionnel, les objets deviennent de plus en plus connectés pour nous faciliter la vie, et nous les retrouvons partout dans notre quotidien.

Que ce soit pour réduire du trafic routier, la consommation d'énergie, faciliter l'accès instantané aux informations, jusqu'à la plus simple tâche d'ouvrir la porte de votre maison. Le monde devient accessible à travers votre téléphone mobile, et avec un seul clic, vous avez tout ce que vous désirez entre vos mains.

Cependant, cela veut bien dire qu'une application mobile et web soient indispensables pour le contrôle à distance de votre domicile, et c'est dans ce cadre que s'inscrit ce projet, qui a pour objectif la mise en place d'une solution mobile permettant essentiellement de contrôler à distance, un modèle de maison intelligente, où nous allons bien étudier le comportement de ces objets. Un projet de fin d'études qui sera notre station finale pour décrocher notre licence en « informatiques, Réseaux et Multimédias » à la Faculté des Sciences et Techniques de Mohammedia durant une période de deux mois prolongeable à compter du 10 Mars.

Ce rapport est constitué de quatre chapitres, Le premier chapitre est consacré au contexte général du projet et à la méthodologie de travail et planification. Le second chapitre est dédié au contenu du projet et à la conception de l'application et sa modélisation avec un diagramme de cas d'utilisation, un diagramme de classes et des diagrammes de séquences.

Le troisième chapitre est dédié aux outils de développement, choix de technologies et environnement de travail. Et le dernier chapitre présente nos réalisations.

CHAPITRE 1 :

Contexte général du projet.

Ce chapitre vise à présenter le contexte général du projet, introduire l'organisme et à la méthodologie de travail et planification.

I- Organisme d'accueil :

La Faculté des Sciences et Techniques de Mohammedia (FSTM) est un établissement public de l'enseignement supérieur à accès réglé, implantée au nord de la ville de Mohammedia et constitue une composante importante de l'université Hassan II de Casablanca, qui vise à assurer des formations scientifiques et techniques de qualité en adéquation avec le marché de l'emploi.

La mission de la FSTM s'articule autour de l'étudiant de telle manière à former non seulement un lauréat excellent sur le plan scientifique et technique, mais également un étudiant doté du sens de la citoyenneté et d'entrepreneuriat capable de saisir l'opportunité autour de lui et contribuer à l'engagement social.

La FSTM est un établissement à accès régulé où l'entrée se fait après une sélection basée sur les notes au baccalauréat de matières définies en fonction du parcours choisi (MIP ou BCG). Elle compte 8 départements et 13 laboratoires. Les formations de la recherche au sein de la FSTM couvrent différents champs disciplinaires en Biologie, Mathématiques et Physiques. La FSTM propose une carte de formation riche et relativement complète en Licence, Master Sciences et Techniques et Filières d'ingénieurs.



Figure 1. logo FSTM

Ainsi dans le cadre du développement du projet de la préparation à des établissements intelligents, la faculté des sciences et techniques de Mohammedia nous propose un stage dont l'objectif est la modélisation d'un smart Home et la conception et développement d'une application Web et mobile, de manière à permettre l'observation des objets connectés (IoT) et leur configuration à l'intérieur d'une maison intelligente.

Cette approche va permettre par la suite de se préparer à l'immigration vers l'IoT pour la gestion quotidienne interne de la FSTM, et grâce à ce projet, une visibilité plus claire sera faite sur les objets intelligents à prévoir pour ménager différents services.

II- Besoin actuel en IoT :

Internet des objets ou bien IoT (Internet Of Things) caractérise principalement les objets dans un système connecté à internet.

De la même façon que les ordinateurs ont un réseau de communication entre eux, l'IoT définit un même réseau entre les objets, un pont entre le virtuel et le physique.

L'internet des objets a un grand impact sur tous les domaines (santé, transport, industrie, immobilier ...) et bientôt, d'après une étude effectuée par CISCO (une entreprise informatique d'origine américaine spécialisée dans le matériel réseau), près de 50 milliards d'objets connectés seront utilisés sur la planète en 2020.

D'ailleurs grâce à l'IoT, on a connu l'évolution de la maison intelligente, une révolution technologique qui améliore le sentiment de la sécurité et le confort au sein de notre foyer.

L'internet des objets est au cœur de cette transformation qui nous facilite la vie dans notre habitation quotidienne.

III- Cahier de charge :

1-Benchmarking :

Ceci est une phase primordiale dans la réalisation de notre projet, une phase d'analyse et de recherche qui nourrit l'esprit et favorise la créativité.

Pour une maison intelligente, au vrai sens du terme, des services alléchants et la meilleure expérience qui soit, en bénéficiant de confort et en se sentant en toute sécurité, les solutions sont variées.

a. Solution TaHoma :

Somfy, qui est le leader mondial de l'ouverture et fermeture des maisons et bâtiments, propose la solution TaHoma. C'est une interface qui vous permet le contrôle de tous les équipements qui sont connectés chez vous, elle offre des services intelligents tout ceci grâce à une box ouverte et évolutive :

- Un pilotage de votre PC, tablette ou bien de votre application (Android/iOS);
- Fermer les volets ou rideau tout ça à distance pour une meilleure sécurité;
- Des capteurs d'ensoleillement, pour une optimisation d'énergie et une température ambiante et fraîche à l'intérieur de la maison;
- SomFy avec ces plusieurs partenaires, élargit ainsi la zone de compatibilité des équipements (éclairage, chauffage, ouvrants) avec leurs solutions.

b. Solution Smart Home de IAM :

Maroc Telecom, qui est l'une des sociétés de télécommunication les plus connus dans notre pays, le Maroc, propose cette solution smart home, qui offre plusieurs fonctionnalités de bien-être et une économie d'énergie qui se représentent dans les services suivants :

- Une box centrale ;
- Une caméra IP WIFI, réglable en cas d'intrusion pour prendre des photos par exemple, ainsi que des détecteurs de mouvements sans fil pour une sécurité garantie ;
- Une gestion de l'éclairage et de l'économie à distance.
- Un système de notification en cas de problème, soit par mail ou pas SMS sans frais supplémentaire et en illimité ;
- Tous ces services proposés sont accessibles sur une interface simple d'usage, sur ordinateur ou sur votre Smartphone, pour une bonne exploitation de ces fonctionnalités, en plus d'un service client d'assistance joignable par téléphone.

c. Solution FIBARO :

Une solution avancée par MAdomotique, un distributeur de solutions domotique et sonorisation au Maroc, assure confort, sécurité et économie d'énergie dans leur offre. Indépendante du Cloud, contrôle local ou à distance ainsi qu'un nombre d'utilisateur illimité s'adaptant à tous types de logement, elle englobe les services suivant:

- Commandes des rideaux soit par oubli, soit en cas de mauvais temps, programmable lors de votre sortie ou arrivé ;
- Commande de chauffage et climatisation, il est nécessaire de contrôler le temps de fonctionnement pour les équipements carnivores ;

- Bien sûr pour le domaine de sécurité les alarmes sont indispensables, un détecteur d'introduction, une excellente surveillance de votre maison ;
- De massif mode d'éclairage selon l'ambiance voulu : mode lecture, mode relaxation, mode cinéma...

MA domotique, comme pour toutes ces autres solutions, met en avant la compatibilité de ces équipements avec le contrôle vocal (Google, Amazon), un plus qui en effet fait beaucoup de différence et facilite la vie.

Devant une abondance de solutions, on déduit une similitude flagrante entre les services offerts dans le marché, due à une concurrence accrue qui fait qu'à peu près tous les fournisseurs ont un niveau de développement des technologies égal.

2-Descriptions des besoins :

L'évolution technologique de l'IOT mène à la nécessité de transformer notre habitat en maison moderne, intelligente et sécurisée, cette dernière devient de plus en plus répandue et présente dans notre société, et elle se doit répondre à certains besoins qui touchent l'être humain de tout âge.

Ces besoins de notre vie quotidienne se représentent principalement sur les aspects suivants :

- **Confort:** l'un des aspects principaux et le plus demandé dans le marché qui se caractérise en moins d'effort et plus d'aisance et de somptuosité, comme le fait d'ajuster la température ambiante de notre chez soi peu importe le climat extérieur ;
- **Sécurité:** la protection contre n'importe quel potentiel danger et un réseau qui veille sur les habitants de la maison ;
- **Santé:** un contact avec l'extérieur ou un système d'alerte lors d'une détection de fumée par exemple, on se doit de penser à toutes sortes de scénarios même les plus amers ;
- **Communication:** un accès instantané aux informations, et une visibilité sur l'état de votre habitation même à distance.

3-Objectif du projet :

Notre sujet comporte deux volets :

- 1) Concevoir et configurer un modèle de smart home avec au moins cinq services qui répondent exactement aux besoins qu'on a traités précédemment: confort, sécurité, santé et communication.

Plusieurs objets intelligents sont à prévoir pour assurer ces services, comme les lampes, les capteurs de températures, de fumée etc., et donc nous allons développer un modèle en exploitant le langage python, qui va simuler les comportements de ces objets en utilisant la programmation orientée objet pour capter les données de valeur.

- 2) Concevoir et développer deux applications Web et Mobile pour pouvoir gérer ces services à distance. Dans ce contexte, nous allons concevoir nos applications en utilisant les Framework ReactJS en front-end pour le web et React Native pour le mobile, et ceci pour réaliser de belles interfaces mobile, et Flask en back end qui va nous permettre de récupérer, stocker et traiter les données de ces objets, ensuite en va utiliser le service Firebase pour la gestion de la base de données des utilisateurs.

IV- Planification et gestion de projet :

Plusieurs méthodes de gestion de projet et d'outils informatiques correspondants existent de nos jours, et la chose la plus importante pour la réussite d'un projet est le management de délais et la bonne répartition des tâches entre les ressources humaines qui travaillent sur le projet.

- Afin de travailler de manière efficace sur notre projet à distance dans cette période du confinement sanitaire, nous avons choisi **Trello**, qui est un outil de gestion de projet en ligne, lancé en septembre 2011 et inspiré par la méthode Kanban de Toyota. Il repose sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre.

Dans notre cas, Trello nous aidera à organiser notre travail et notre temps et ainsi assurer une visibilité de nos tâches, leurs délais et leurs avancements.

Cette application nous aide aussi à garder les documents et les liens avec lesquels nous travaillons dans un seul workspace, ce qui nous facilite l'accès à l'information sans perdre de temps.

La figure 2 illustre une capture d'écran de notre espace de travail sur Trello dans un instant T :

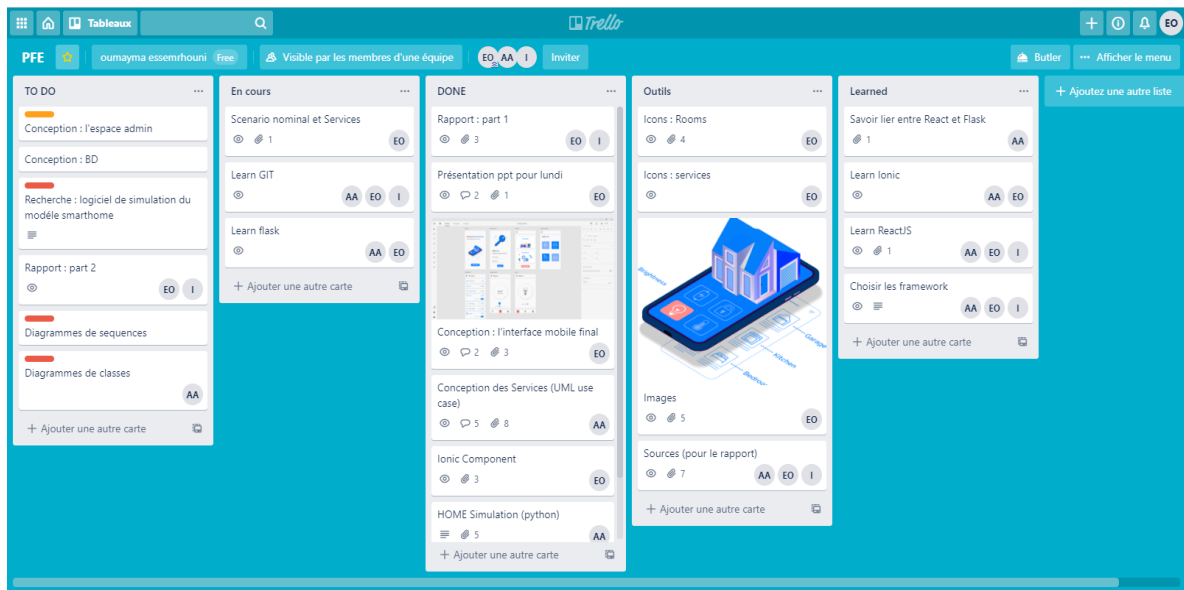


Figure 2. Capture d'écran de l'application Trello

Pour une bonne conduite de projet, nous avons besoin d'un plan à respecter, des différentes tâches à faire durant toute la période de travail sur notre projet de fin d'étude. Nous avons choisi pour cela le **diagramme de Gantt**, qui est un outil utilisé en ordonnancement et en gestion de projet et permettant de visualiser dans le temps les diverses tâches composant un projet et ainsi de représenter graphiquement l'avancement du projet.

Le diagramme de la figure 3 représente les tâches principales à réaliser dans notre projet :

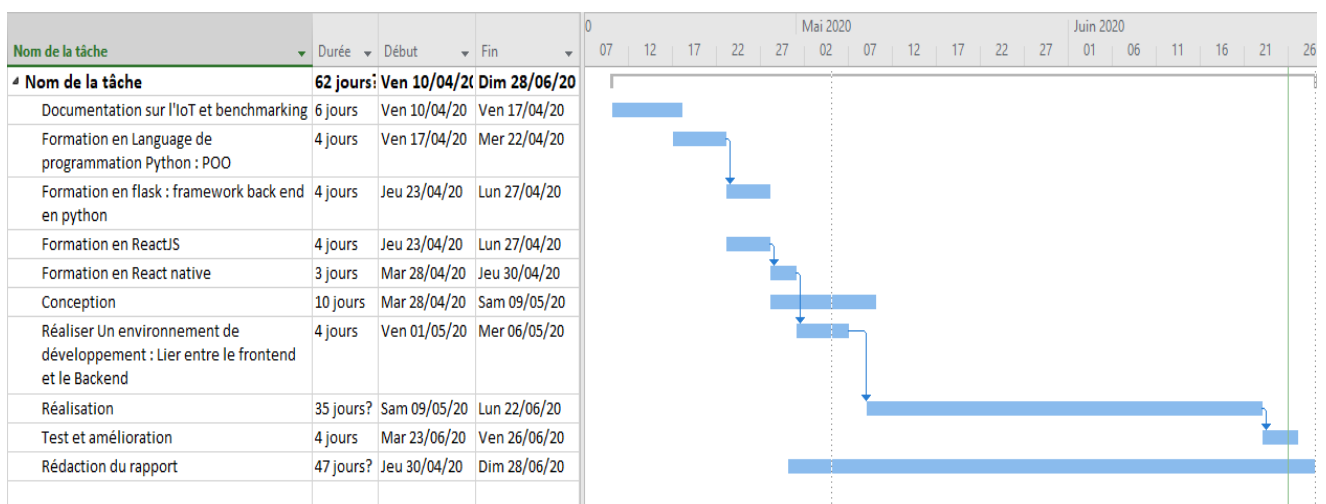


Figure 3. Diagramme de Gantt

CHAPITRE 2 :

Analyse et conception

Ce chapitre traite principalement les buts et objectifs du projet ainsi que son étude conceptuelle à travers des diagrammes de cas d'utilisations, séquences et classes.

I. Contenu du projet

1. Choix des services

Afin de répondre aux exigences citées auparavant, nous avons choisie de travailler sur les services suivant :

- Allumer / éteindre une lampe dans une chambre ;
- Détecter le mouvement pour allumer les lampes des couloirs ;
- Ajuster la température à l'aide du climatiseur ;
- Ouvrir / fermer la porte du garage ;
- Détecter la fumée et envoyer une alerte ;
- Ouvrir / fermer la porte maison ;
- Ouvrir / fermer les fenêtres automatiquement ;
- Visualiser la température d'extérieur ;
- Détecter l'état de la Pluit ;

2. Simulation des objets connectés

Pour concevoir ces services, il faudra bien sûr des thermomètres, des avertisseurs de fumée, des capteurs d'état de lampes etc.

Afin de générer des valeurs à partir de ces objets en absence de matériel électronique, nous allons devoir simuler leurs comportements en langage Python, et ceci en créant des classes et méthodes qui nous retournent des valeurs que nous allons traiter par après.

3. Espace SuperAdmin

Nous allons développer une application web pour l'administrateur, et son rôle consiste à faire la gestion des comptes des Admin et leurs donner les informations de connexion, qui leurs permettra d'accéder à l'application de contrôle à distance de leurs maisons intelligentes, et ainsi le droit d'ajouter des nouveaux comptes utilisateurs.

4. Applications Web & Mobile

Après avoir récupérer ses informations de connexion auprès du SuperAdmin, l'utilisateur va accéder aux applications Web & Mobile qui lui permettent de contrôler sa maison à distance.

Ces applications vont contenir deux principaux boutons :

- Un bouton « Rooms » pour accéder aux services de chaque chambre individuellement comme l'ajustement de la température du salon.

- Un bouton « Setting » pour accéder aux paramètres des services de la maison en général, comme celui de la fermeture de la porte du garage.

L'utilisateur de type « Admin » a une fonctionnalité de plus, à travers le bouton « Family » qui lui permet d'ajouter d'autre utilisateur à l'application.

5. Système de notification

Cette fonctionnalité consiste à envoyer une alerte à l'utilisateur (l'habitant) sous forme de notification dans l'application une fois le capteur de fumée détecte un déclenchement de feu.

II. Etude conceptuelle :

1. Introduction

L'étude conceptuelle est une étape très importante dans chaque projet, qui a pour but d'expliquer d'une manière claire et détaillée l'ensemble des fonctionnalités du système en phase de création, et définissant l'architecture générale de la solution proposée à travers les différents diagrammes que vous allez voir par la suite.

2. Les acteurs du système

Un acteur est une « entité » externe au système qui interagit avec le système.

Les acteurs apparaissent dans les diagrammes de cas d'utilisation.

Acteur	Rôle
Super Admin	Le SuperAdmin a pour rôle de gérer les comptes des admin de l'application et leurs donner les informations d'accès à l'application autant qu'administrateur.
Admin	L'admin a pour rôle de gérer les comptes des utilisateurs de l'application et leurs donner les informations d'accès à l'application autant qu'utilisateur. L'admin est aussi un utilisateur qui peut contrôler les paramètres de chaque service dans la maison.
Utilisateur	L'utilisateur peut consultez l'état de sa maison à n'importe quel moment à partir de l'application, et peut modifier les paramètres de chaque service par chambre ou globalement dans toute la maison à la fois.

4. Descriptions textuelles des cas d'utilisations :

a) Cas d'utilisation : Créer un compte Admin :

- 1) **Acteur** : SuperAdmin.
- 2) **Pré condition** : Le système fonctionnel / Connexion internet ...
- 3) **Scénario Nominal** :
 1. Le système affiche de formulaire de connexion ;
 2. Le SuperAdmin remplit le login et de mot de passe prédéfinie ;
 3. Le SuperAdmin soumet les informations au système ;
 4. Le système vérifie les informations envoyées ;
 5. Le système affiche le tableau des utilisateurs de notre application ;
 6. L'admin click sur "Add an Admin" ;
 7. Le système affiche le formulaire ;
 8. L'admin remplit tous les informations du nouveau Admin ;
 9. L'admin click sur "Add Admin" ;
 10. Le système ajoute le nouveau contact au tableau des Admin et à la base de donnée.

Scénario d'exception :

1. Nom d'utilisateur ou mot de passe incorrecte.

a) Cas d'utilisation : Authentification :

- 1) **Acteur** : Utilisateur (habitant).
- 2) **Pré condition** : L'application est installée et ouverte / Connexion internet / installation du système (les objets intelligents).
- 3) **Scénario Nominal** :
 1. Le système affiche de formulaire de connexion ;
 2. L'utilisateur remplit le nom d'utilisateur et de mot de passe ;
 3. L'utilisateur soumet les informations au système ;
 4. Le système vérifie les informations envoyées ;
 5. Le système affiche la page d'accueil de l'application.

Scénario d'exception :

2. L'utilisateur n'est pas enregistré au système ;
3. Nom d'utilisateur ou mot de passe incorrecte.

b) Cas d'utilisation : Ajouter un utilisateur :

Acteur : Admin.

Pré condition : Le système fonctionnel / L'application est installée et ouverte / Connexion internet.

Scénario Nominal :

1. Le système affiche la page d'accueil ;
2. L'admin click sur « Family » ;
3. Le système affiche la page du profil ;
4. L'admin click sur « Add Member » ;
5. Le système affiche le formulaire ;
6. L'admin remplit toutes les informations du nouvel utilisateur ;
7. L'admin click sur « Add » ;
8. Le système ajoute le nouvel utilisateur à la base de données.

c) Cas d'utilisation : Allumer / éteindre une lampe :

Acteur : Utilisateur (habitant).

Pré condition : Être connecté à l'application, maison connectée au serveur.

Scénario Nominal :

1. L'utilisateur clique sur "Rooms" ;
2. Le système affiche la liste des chambres qui existent ;
3. L'utilisateur accède à une chambre X ;
4. Le système affiche l'état de la lampe dans la chambre X ;
5. L'utilisateur peut changer l'état de la lampe en cliquant sur ON/OFF ;
6. Le système affiche l'état de lampe après changement d'état.

d) Cas d'utilisation : Lampes détecteurs de mouvement :

- 1) **Acteur :** Utilisateur (habitant).

2) **Pré condition** : Être connecté à l'application et le capteur de mouvement installé et connecté au serveur.

1. L'utilisateur click sur paramètre ;
2. Le système affiche le mode de détection de mouvement (ON ou OFF) ;
3. L'utilisateur click sur ON pour activé le service ;
4. Le système affiche l'état du service après modification.

e) Cas d'utilisation : Ajuster la température :

1) **Acteur** : Utilisateur (habitant).

2) **Pré condition** : Être connecté à l'application et le capteur de température installé.

3) **Scénario nominal** :

1. L'utilisateur click sur une chambre X ;
2. Le système affiche les services de cette chambre ;
3. Le système affiche la température actuelle de la chambre ;
4. L'utilisateur active le climatiseur au cas où il était désactivé;
5. L'utilisateur ajuste la température ;
6. Le système affiche la température après son ajustement.

f) Cas d'utilisation : Safety alert (Détection de fumée) :

1) **Acteur** : Utilisateur (habitant).

2) **Pré condition** : Être connecté à l'application et le capteur de fumée installé.

3) **Scénario nominal** :

1. L'utilisateur click sur "Setting" ;
2. Le système affiche de mode du service de l'alert et de l'arrosoir (ON/OFF) ;
3. L'utilisateur click sur ON pour activer le service de l'alert et de l'arrosoir ;
4. Le système détecte que la fumée est supérieure au seuil X ;
5. Le système déclenche l'arrosoir ;
6. Le système active une alarme dans la maison ;
7. L'utilisateur clic sur arrêter pour arrêter l'alarme.

g) Cas d'utilisation : Ouvrir/fermer la porte de garage :

- 1) **Acteur** : Utilisateur (habitant)
- 2) **Pré condition** : Être connecté à l'application
- 3) **Scénario nominal** :
 1. L'utilisateur click sur "Setting" ;
 2. Le système affiche l'état de la porte du garage ;
 3. L'utilisateur consulte l'état de la porte ;
 4. L'utilisateur click (ON/OFF) pour changer l'état de la porte ;
 5. Le système affiche l'état de la porte après la modification.

h) Cas d'utilisation : Ouvrir/fermer la porte de la maison :

- 1) **Acteur** : Utilisateur (habitant)
- 2) **Pré condition** : Être connecté à l'application
- 3) **Scénario nominal** :
 1. L'utilisateur click sur "paramètre" ;
 2. Le système affiche l'état de la porte ;
 3. L'utilisateur consulte l'état de la porte ;
 4. L'utilisateur click (ON/OFF) pour changer l'état de la porte ;
 5. Le système affiche l'état de la porte après la modification.

i) Cas d'utilisation : Ouvrir / fermer la fenêtre d'une chambre :

- 1) **Acteur** : Utilisateur (habitant).
- 2) **Pré condition** : Être connecté à l'application.
- 3) **Scénario nominal** :
 1. L'utilisateur click sur "Rooms" ;
 2. Le système affiche la liste de chambre ;
 3. L'utilisateur click sur la chambre X ;
 4. Le système affiche l'état de la fenêtre ;
 5. L'utilisateur consulte l'état de la fenêtre ;

6. L'utilisateur peut (Ouvrir/fermer) la fenêtre automatiquement ;
7. Le système affiche l'état de la fenêtre après la modification.

5. Diagrammes de séquences :

Un **diagramme de séquence** est un diagramme qui expose graphiquement les interactions et les opérations effectuées entre les acteurs et le système en détail tout en représentant la chronologie des actions.

La figure suivante représente le diagramme de séquence de l'action de l'authentification.

Lorsque l'admin ou l'utilisateur accède à leurs pages de connexion et saisit le login et le mot de passe, le système vérifie si ces informations sont valides. S'il se trouve que les champs saisis sont corrects, l'acteur réussit à accéder à son espace selon son rôle, sinon, un message d'erreur sera affiché :

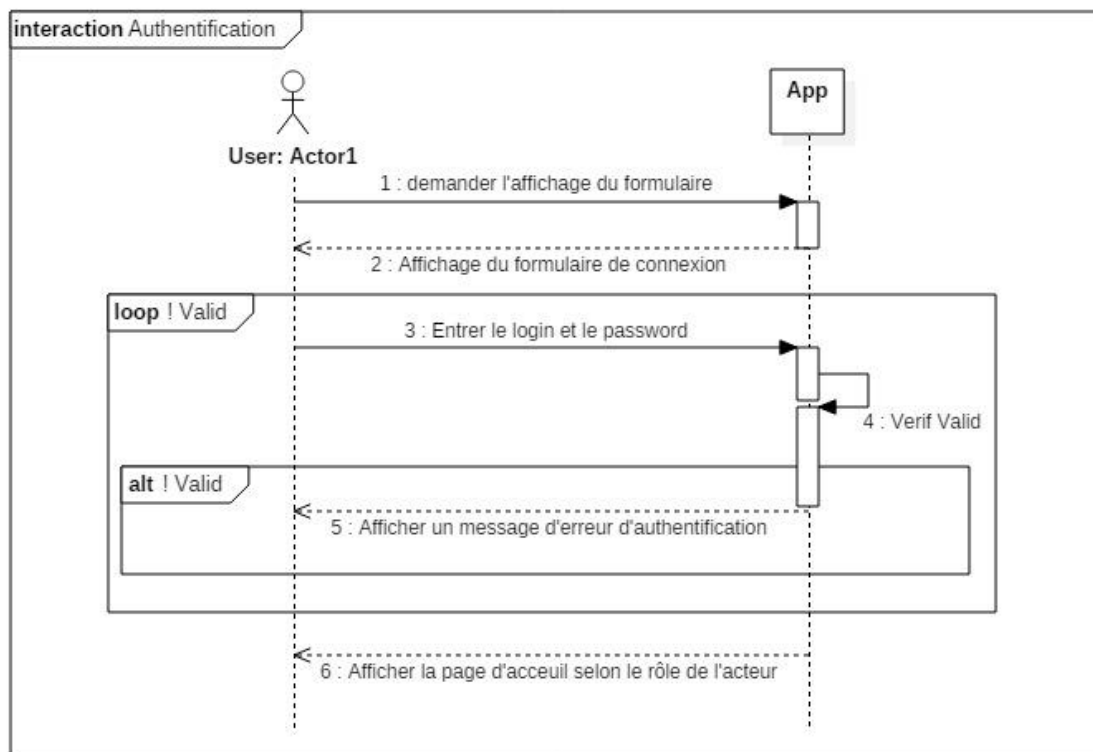


Figure 5. Diagramme de séquence « Authentification »

Après l'authentification, l'utilisateur/Admin choisit l'option « chambre », qui lui permet de choisir la chambre désirée, et consulter l'état de ces services ou changer ces états.

La figure suivante représente le diagramme de séquence de cette action :

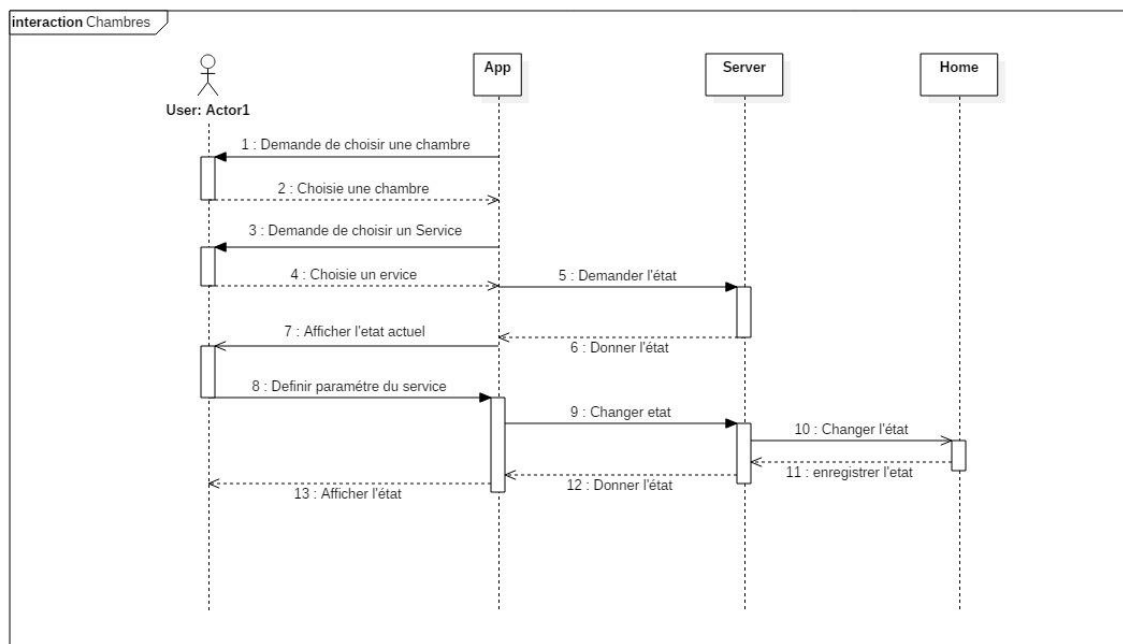


Figure 6. Diagramme de séquence « Chambre » :

Après l'authentification, l'utilisateur/Admin choisit l'option « paramètres », qui lui permet de consulter l'état des services globaux de la maison ou changer leurs états.

La figure suivant représente le diagramme de séquence de cette action :

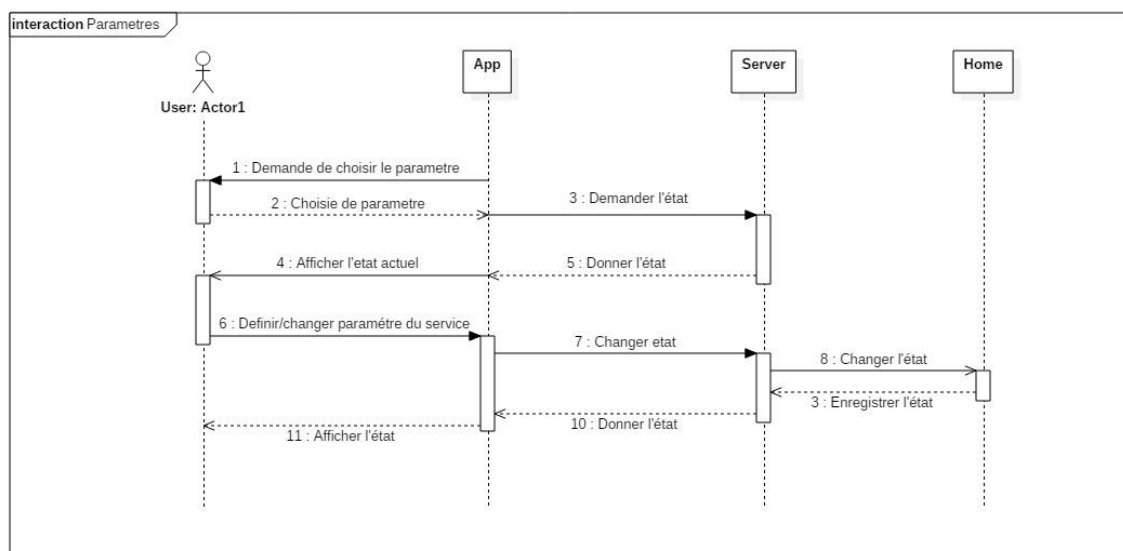


Figure 7. Diagramme de séquence « Paramètres »

La figure suivant représente le diagramme de séquence « Main » global des différentes actions.

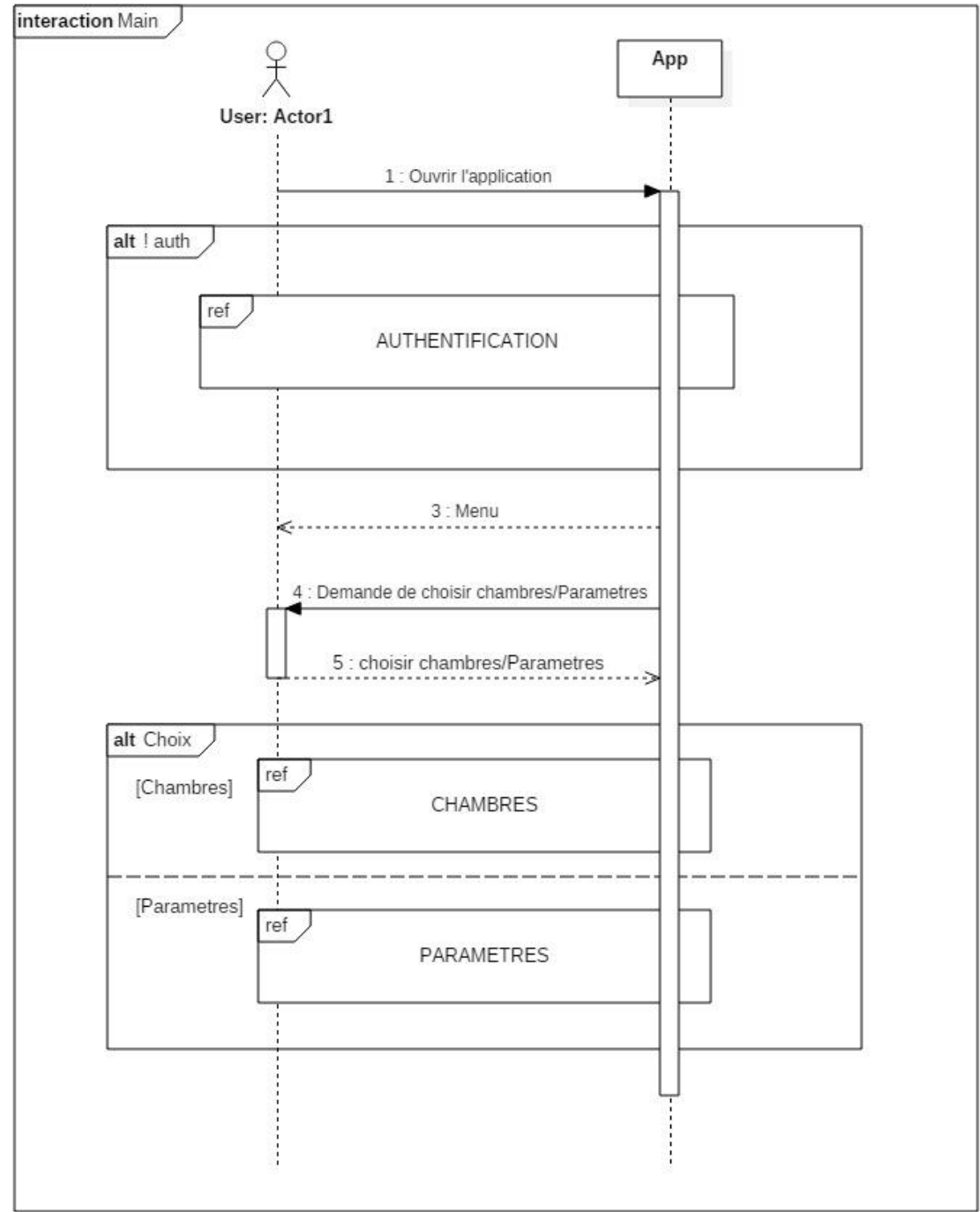


Figure 8. Diagramme de séquence « Main »

6. Diagramme de classe :

Le **diagramme de classe** schématise les classes et interfaces du système ainsi que les relations entre elles.

Ci-dessous, le diagramme de classe qui rassemble toutes les classes utilisées dans notre projet :

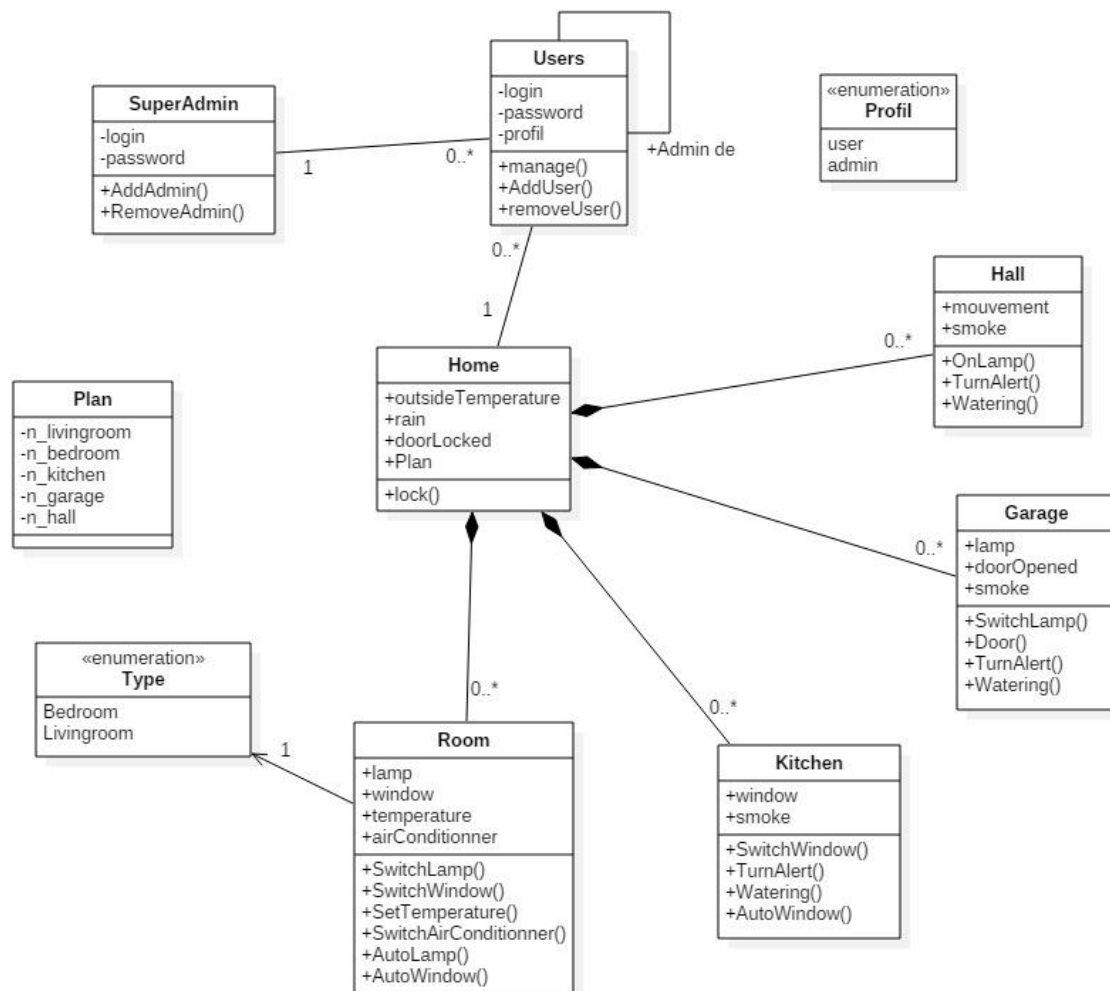


Figure 9. Diagramme de classes.

CHAPITRE 3 :

Etude technique et environnement de travail.

Dans ce chapitre, nous allons présenter les différentes technologies utilisées lors de la réalisation du projet, les outils ainsi que l'environnement de développement

I- Framework et technologies :

Au tout début de chaque projet, le choix des outils adéquats est considéré comme une étape cruciale. Une comparaison s'imposait, entre les différents Frameworks et technologies du marché ayant les meilleures recommandations et qui utilisaient les langages de programmation les plus exploités.

Après une étude précise et adaptés au but du projet, voici les technologies sur lesquelles notre choix s'est porté.

1- React :



Figure 10. Logo React.

React, réalisé en 2013, est un Framework qui utilise le langage JavaScript, très courant pour concevoir les applications web, en toute facilité grâce à la création des composants dépendants d'un état et générant une page HTML à chaque changement d'état.

Ces composants contiennent des éléments en JSX, une extension syntaxique de JavaScript, très près de l'HTML, avec du style en CSS.

L'ancienneté de React est un point très fort qui le rend stable vu le nombre de projet réalisé et maintenu par ce dernier.

2- React Native :

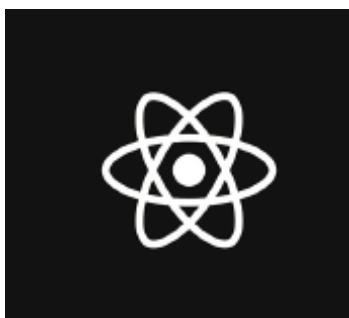


Figure 11. Logo React Native

React Native est un Framework d'applications mobiles open source créé par Facebook et basé sur React. Il est utilisé pour développer des applications pour Android, IOS et UWP en permettant aux développeurs d'utiliser React avec les fonctionnalités natives de ces plateformes.

3- Node Js :



Figure 12. Logo NodeJS

Node.Js est une plateforme logicielle de développement, orientée vers les applications réseau événementielles hautement concurrentes, il nous permet de générer du code JavaScript côté serveur.

Nous avons utilisé Node.js pour travailler avec les Frameworks React et React Native qui nécessite un environnement Node (côté serveur) pour fonctionner.

4- Flask :



Figure 13. Flask.

Flask est un Framework léger pour le développement des applications en Python. Il nous procure des bibliothèques et de outils pour aider dans la réalisation de l'application. Nous l'avons utilisé en back-end pour les exécutions côté serveur et créer une API pour les différents services et fonctionnalités des deux applications Web et Mobile.

5- React Navigation



Figure 14. Logo React navigation

React Navigation, compatible avec IOS et Android, est une librairie employée pour gérer la navigation de nos pages et la transition d'une vue à une autre ainsi que le retour en arrière

6- Firebase :

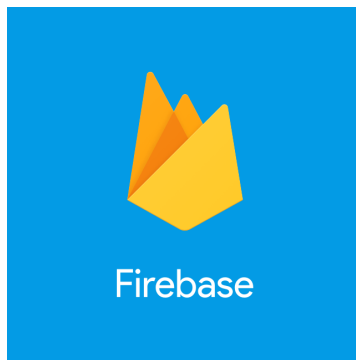


Figure 15. Firebase logo

Firebase est un ensemble de services d'hébergement pour n'importe quel type d'application (Android, iOS, Node.js, Java...).

Il propose d'héberger en NoSQL des bases de données. Et nous l'avons utilisé pour le service de Cloud Firestore pour le système d'authentification, ainsi que la gestion de l'espace SuperAdmin, ajout d'admin et d'utilisateurs.

II- Environnement et outils :

1. GIT et GitHub :

Git est un logiciel de contrôle de version utilisé pour la gestion des codes source, son utilisation s'est rendu indispensable d au changement fréquent du code et l'évolution du projet.



Figure 16. Logo Git

GitHub est une plateforme cloud sur laquelle les développeurs peuvent héberger et vérifier leur code, gérer des projets et créer des logiciels de façon collaborative. Les

équipes peuvent ainsi créer de nouvelles versions de code, les vérifier, y ajouter des commentaires et les mettre en œuvre dans le cadre de projets.

Ce dernier est un excellent outil collaboratif pour optimiser le travail d'équipe et ainsi améliorer notre productivité.

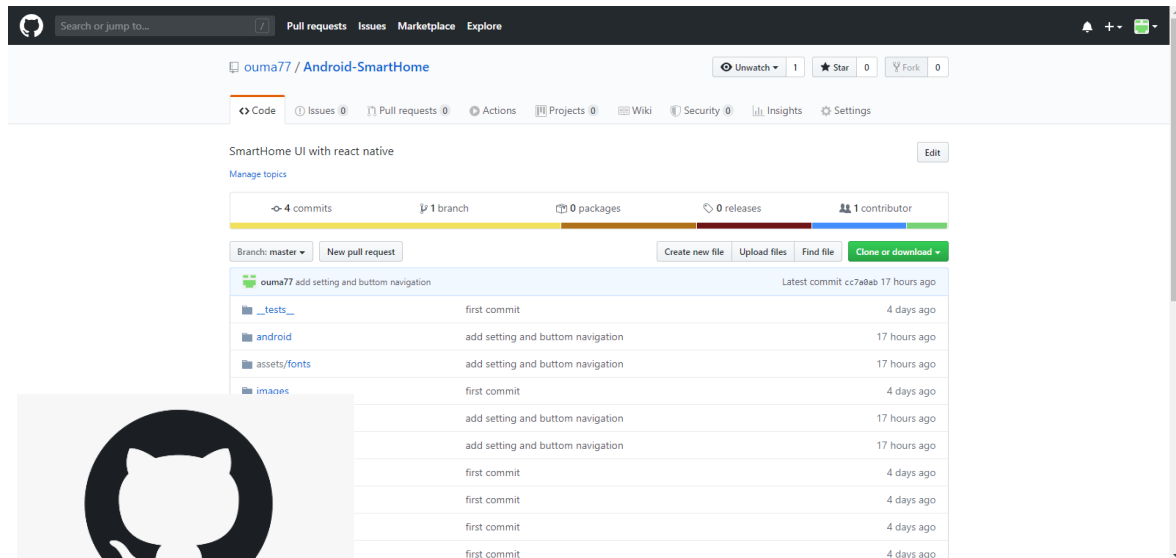


Figure 17. Screen d'un projet GitHub



Figure 18. Logo GitHub

2. Visual Studio Code :

Visual Studio Code est un éditeur de Code multiplateforme développé par Microsoft, il supporte plusieurs langages de programmation et facilite la saisie du code.

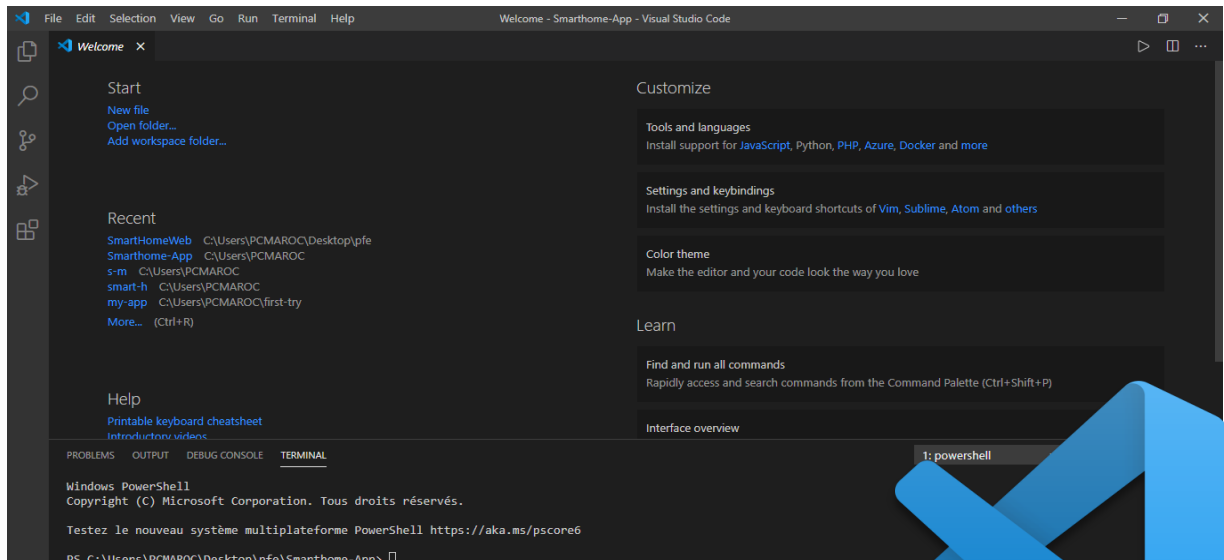


Figure 19. Capture de Visual Studio Code.

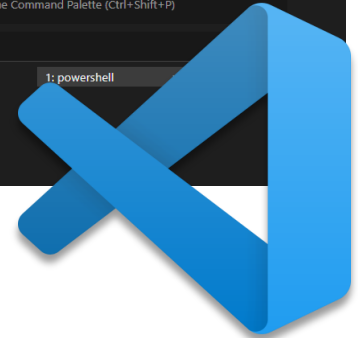


Figure 20. Logo VS Code

3- Yarn :

Yarn est un gestionnaire de paquets qui facilite au développeur la gérance des différents paquets installés dans un projet JavaScript, aussi il garde une trace de toutes les versions utilisées.

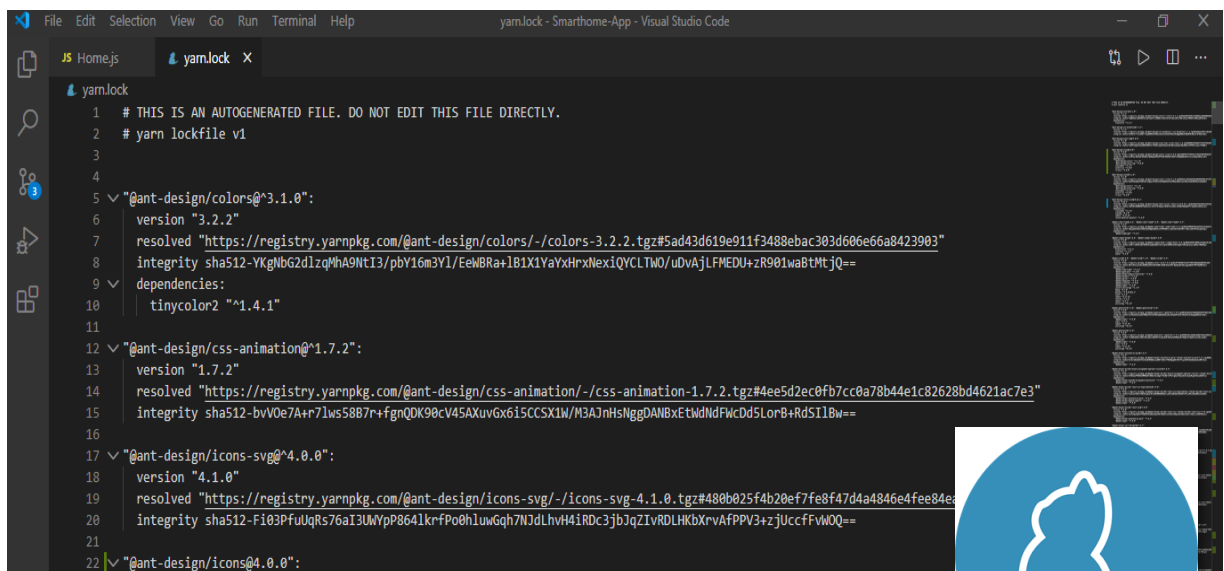


Figure 22. Capture du fichier Yarn dans Vs code

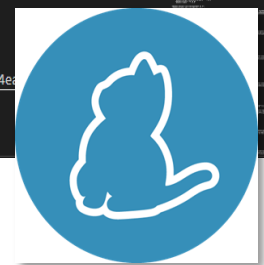


Figure 21. Logo Yarn

4. Android Studio :



Figure 23. Android Studio logo

Android Studio est un environnement de développement conçu pour le développement des applications mobiles Android.

Nous l'avons utilisé car il inclut un émulateur interne permettant de faire tourner un système Android virtuel sur nos ordinateurs.

5- Star Uml :



Figure 24. Logo Star Uml

Un logiciel pour la modélisation UML, employé pour l'illustration de tous les diagrammes dans la partie de l'étude conceptuel du projet.

6- Adobe Xd :

Adobe Xd est un outil vectoriel permettant le design et le prototypage, est utilisé pour la conception des interfaces de l'application web et mobile.

La figure suivante représente un design construit à partir de zéro par nous sur Adobe XD :

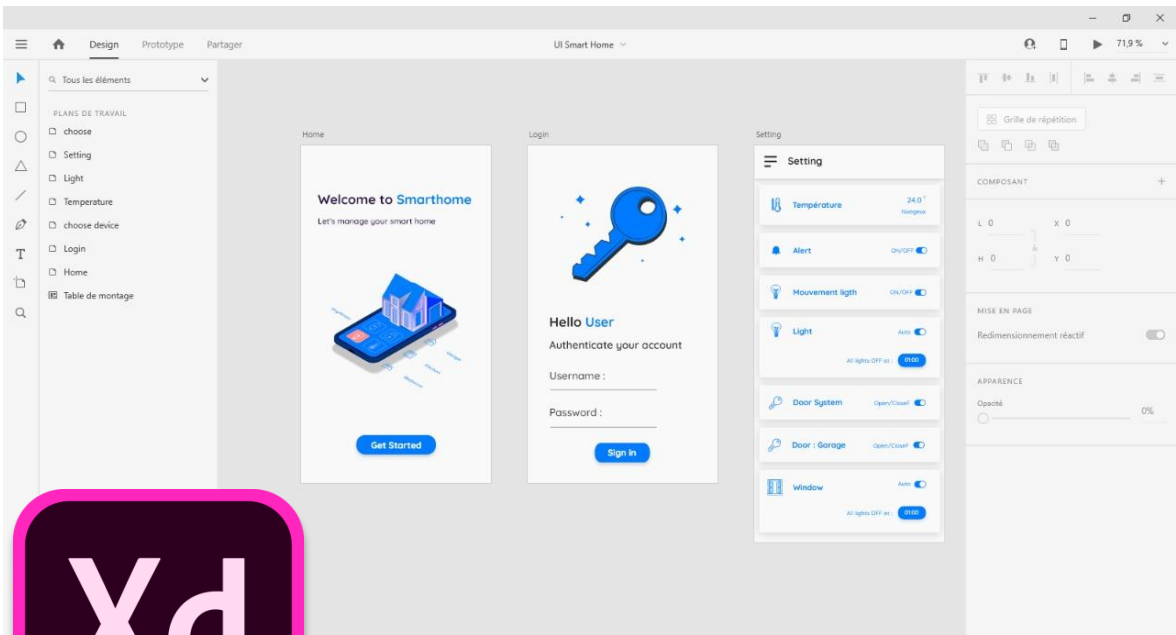


Figure 26. Capture d'Adobe Xd



Figure 25. Logo Adobe Xd

7- Ant Design :



Figure 27. Logo Ant Design

Antd est une bibliothèque de React contenant plusieurs composants qui aide à la construction de riches interfaces interactives, pour une meilleure expérience en développement.

CHAPITRE 4 :

Réalisation.

Ce chapitre vise à présenter la simulation des objets connectés à la maison en langage Python, ainsi les interfaces web et mobile de notre application avec un scénario de l'expérience utilisateur.

I. Simulation :

1. Modèle graphique des services de la maison Intelligente :

Le modèle sur lequel nous avons travaillé est une maison qui peut comporter un maximum de chambres à coucher, salons, cuisines et couloirs, ainsi qu'un garage.

La totalité des chambres représente le « plan » de la maison d'un habitant. Dans chaque chambre existent un ou plusieurs services différents qui transforment notre maison en un parfait habitat intelligent.

La figure suivante présente un modèle graphique qui illustre l'architecture désirée de la maison et la distribution de services sur lesquels nous avons travaillé pour les différentes chambres de cette maison intelligente :

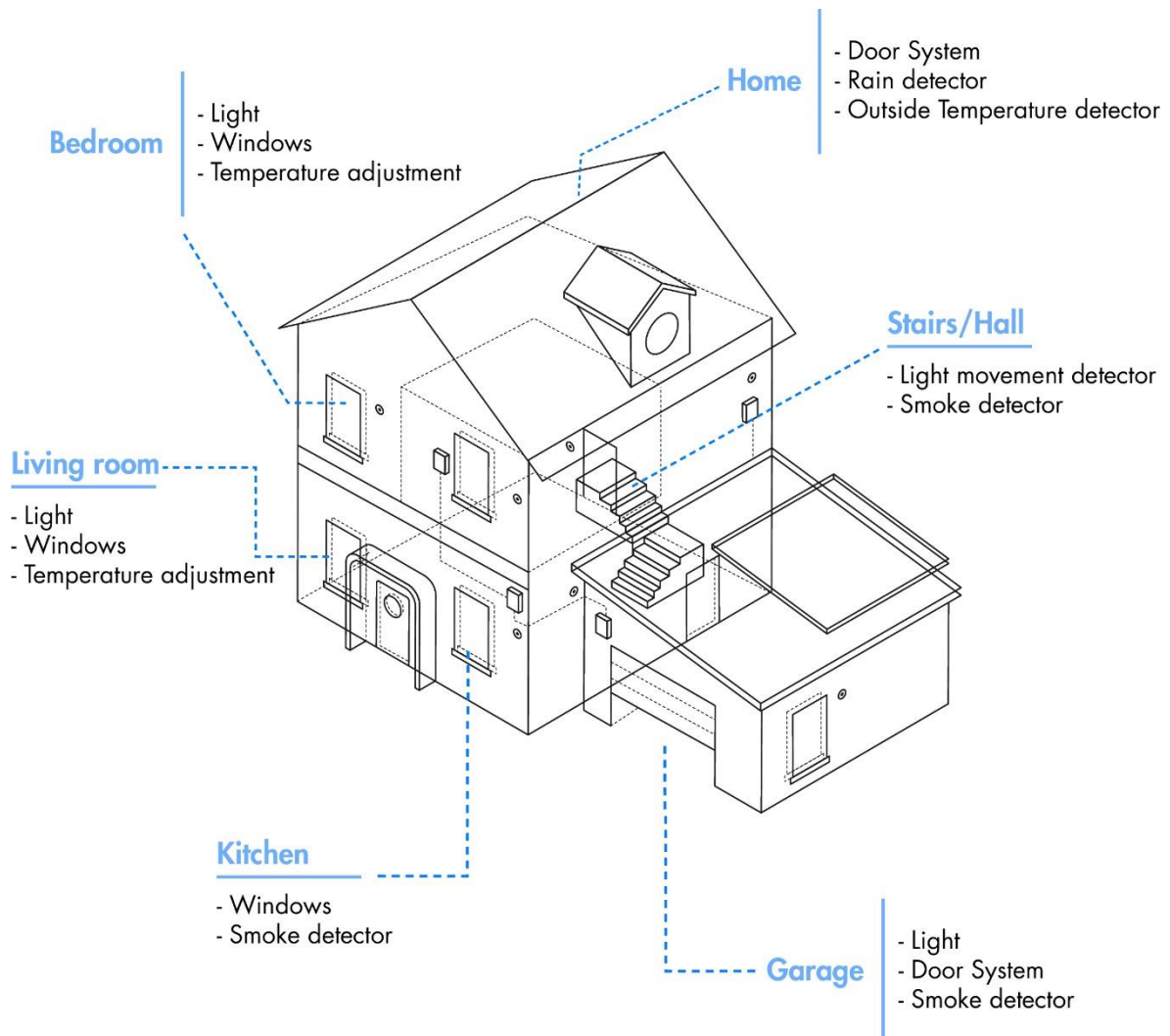


Figure 28. Modèle graphique des services de la maison intelligente

2. Comportement des objets connectés :

Un objet connecté est doté d'un certain nombre d'éléments de type émetteur-récepteur, et possède des capacités lui permettant de dialoguer à distance avec un autre objet.

Dans l'absence des matériels électroniques, nous avons décidé d'étudier le comportement de tous les capteurs dont nous avons besoin, pour connaître l'intervalle des valeurs, les poids des valeurs les plus fréquentes et ensuite nous avons simulé, en Language Python et à l'aide de la programmation orienté objet, le fonctionnement de tous ces objets pour qu'ils nous génèrent les valeurs les plus proches de la réalité.

Nous avons cité, dans cette partie, quelques exemples de nos simulations dans plusieurs parties de la maison.

a. Détecteur de la température d'extérieur :

Pour donner la possibilité de visualiser la température de l'extérieur de la maison, nous avons réalisé une fonction qui nous renvoie à chaque fois une valeur aléatoire entre 15 et 50 degré.

La figure suivante présente la fonction de cette simulation en Language python:

```
25
26     def outsideTemperature(self):
27         Temperatures=range(15,51)
28         self.outsideT=random.choice(Temperatures)
29         return self.outsideT
30
```

Figure 29. Screen du code OutsideTemperature

b. Détecteur de Pluit :

L'habitant peut aussi visualiser, à l'aide de son application, s'il pleut à l'extérieur ou pas, pour ainsi le prévenir de fermer les fenêtres, automatiquement, à travers l'application au cas où elles étaient ouvertes. Le capteur simulé détecte une fois sur huit l'existence d'une pluie.

La figure suivante représente la fonction du capteur de Pluit en Language python:

```
30
31     def isRain(self):
32         states=[True,False]
33         proba=[1,7]
34         self.rain=random.choices(states,weights=proba)
35         return self.rain
36
```

Figure 30. Screen du code IsRain

c. Lampes détectrices de mouvement :

Dans les couloirs, rien n'est plus pratique que des lampes qui s'allument immédiatement lors de votre passage, le temps de quelques secondes, et sans le besoin de les allumer et les éteindre à chaque fois.

La figure suivante représente la fonction de détecteur de mouvement en Language python :

```
7  # Services
8  def isMvt(self):
9      states=[True,False]
10     proba=[3,7]
11     self.mvt=random.choices(states,weights=proba)
12     return self.mvt
13
```

Figure 31. Screen du code isMvt

d. Lampes et Fenêtres :

Ces objets nous renvoient la valeur actuelle, mais l'utilisateur a la possibilité de la modifier selon son choix, dans chaque chambre individuellement.

La figure suivante représente la partie du code qui nous renvoie les valeurs actuelles de l'état de la lampe et de la fenêtre en Language python :

```
10     def lampState(self):
11         states=['on','off']
12         self.lamp=random.choice(states)
13         return self.lamp
14
15     def windowState(self):
16         states=['closed','opened']
17         self.window=random.choice(states)
18         return self.window
19
```

Figure 32. Screen du code de l'état de lampe et Fenêtres

Et pour lier la totalité des fonctions qui simulent les différents capteurs dont nous avons besoin dans la maison, nous avons créé une fonction principale qui crée des chambres selon le nombre qu'on veut, et distribue les services de lampes, températures et fenêtre sur chacune d'elles.

3. Flask API

Afin de lier les fonctions que nous avons réalisé à l'aide du langage Python avec notre application, on a besoin de travailler avec un serveur, et c'est la raison pour laquelle nous avons fait recours à Flask, qui nous a aidé à transformer notre travail à une API (interface de programmation applicative) facile à utiliser par nos interfaces, web et mobile.

Notre API nous fournit une méthode bien faite pour accéder à des éléments particuliers dans nos différents services et données.

L'interaction entre les données de l'API, l'application web, mobile, ainsi que la base de données est donc cruciale. Il s'agit bien de plusieurs requêtes GET et POST lancées à l'API, et des réponses sous forme de fichier JSON comme présenté dans la figure suivante :



Figure 33. Screen d'une réponse sous forme JSON

La figure suivante est un schéma qui explique l'échange de requêtes entre le client, le serveur et les capteurs :

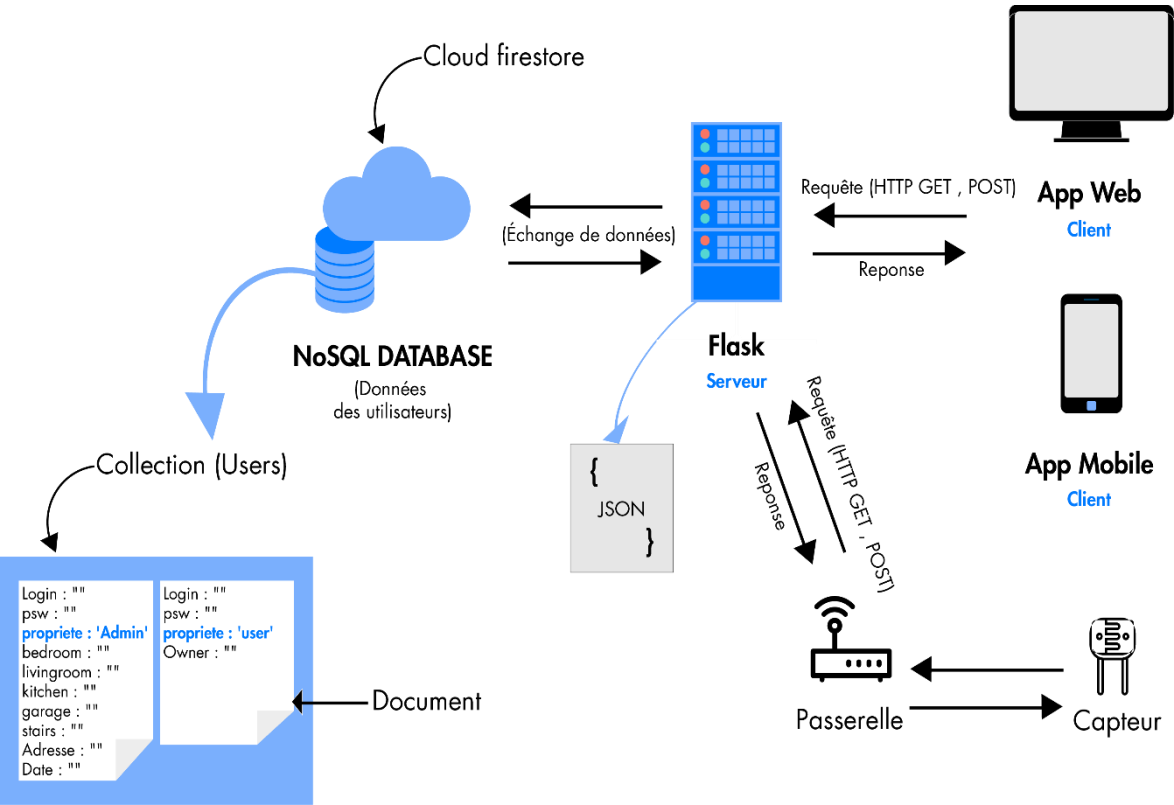


Figure 34. Schéma d'architecture du systeme

II. Console : SuperAdmin :

Pour toutes solutions mobiles, il doit y avoir un acteur externe qui se chargera de l'installation des différents capteurs, lampes et tous autres objets connectés à l'application, et qui donne par la suite l'accès à l'utilisateur pour pouvoir utiliser l'application. Cet acteur est appelé le SuperAdmin.

Nous avons donc réalisé une application web pour le SuperAdmin afin de gérer les comptes des utilisateurs de type « admin ».

1. Authentification :

Notre application possède un système d'authentification qui permet à un seul SuperAdmin d'accéder à la console réservée seul à lui.

Pour faire ceci, nous avons utilisé des variables de l'API que nous avons réalisées pour vérifier si le login et le mot de passe sont corrects. Une fois le SuperAdmin réussit à s'authentifier, l'application renvoie automatiquement l'interface d'accueil qui correspond à son rôle, sinon un message d'erreur sera affiché.

Ces figures représentent l'interface du système d'authentification du SuperAdmin :

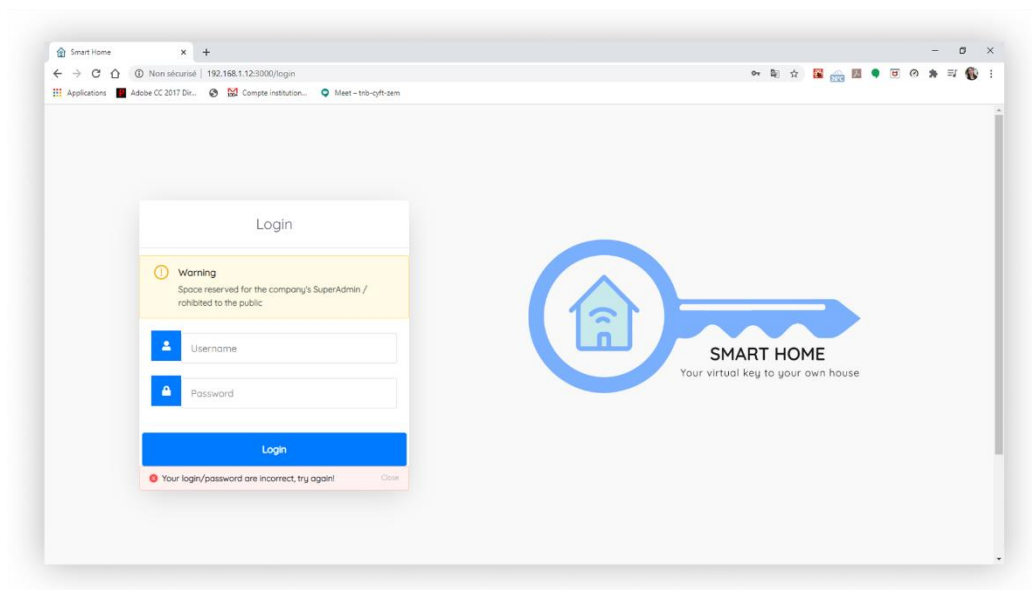


Figure 35. Interface du login SuperAdmin

2. Ajouter un Admin :

Lorsque le SuperAdmin est connecté à l'application, la page d'accueil de la console s'affiche. Cette dernière lui permet d'effectuer deux actions principales :

1. Visualiser les comptes des utilisateurs de type admin et les informations concernant le plan de leurs maisons à travers le champ « Dashboard » ;

2. Ajouter un autre utilisateur de type admin à travers le champ « Add an admin » ;

Les informations ajoutées sont toutes stockées dans une Base de donnée NoSQL en utilisant l'outil Cloud Firestore de Firebase, sous forme d'une collection appelée « users », qui comporte plusieurs documents, chacun spécifie un utilisateur de type admin bien défini et identifié par son login. Au cas où le login existe déjà, le système affiche un message d'erreur et demande d'en choisir un autre.

Les variables qui définissent le nombre de chaque type de chambre sont aussi envoyés à l'API réalisé avec Flask pour adapter l'application de l'utilisateur selon le plan de sa maison. Ceci permet aussi de personnaliser l'application selon chaque utilisateur.

La figure suivante présente l'interface du champ d'ajout d'admin sous forme de formulaire :

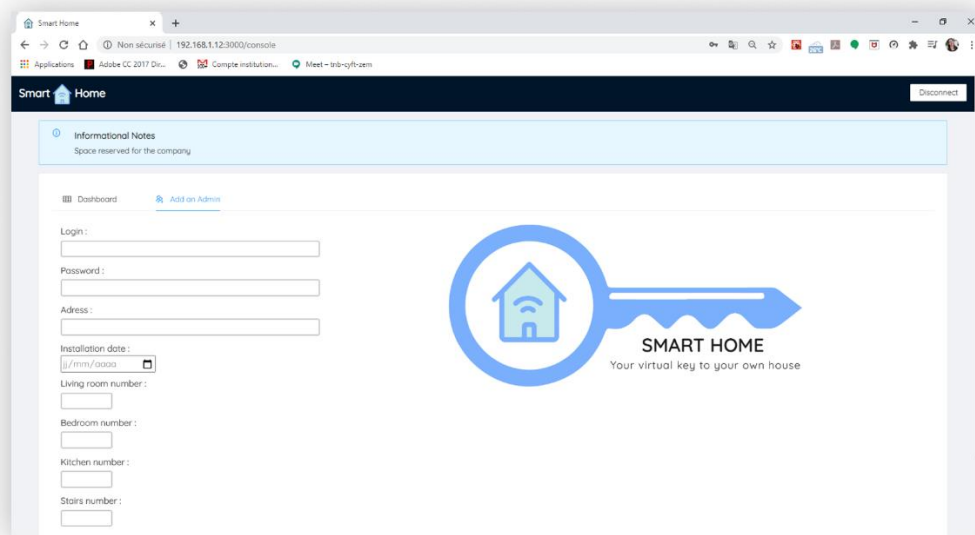
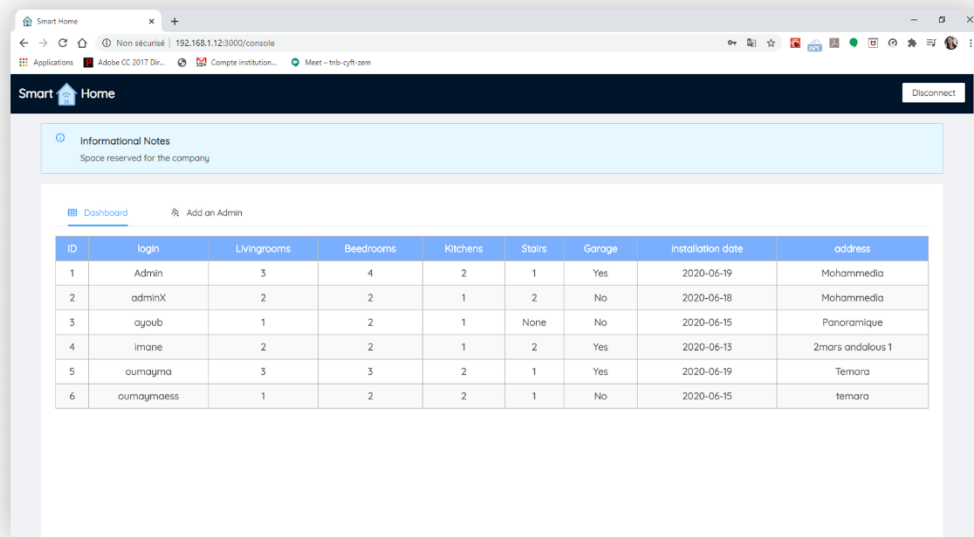
The image shows a web browser window displaying the 'Smart Home' application. The page title is 'Smart Home' and the URL is '192.168.1.12:3000/console'. The page has a dark blue header with the 'Smart Home' logo and a 'Disconnect' button. Below the header, there is a light blue box labeled 'Informational Notes' with the text 'Space reserved for the company'. The main content area has a sidebar with 'Dashboard' and 'Add an Admin' (the latter is highlighted). The 'Add an Admin' form includes fields for 'Login:', 'Password:', 'Address:', 'Installation date:' (with a date picker), 'Living room number:', 'Bedroom number:', 'Kitchen number:', and 'Stairs number:'. To the right of the form is a large blue key icon with a house inside the head, and the text 'SMART HOME' and 'Your virtual key to your own house'.

Figure 36. Interface d'ajout d'admin

La figure suivante présente l'interface du « Dashboard » dans laquelle sont affichées toutes les informations des utilisateurs de type « Admin » ajouté par le SuperAdmin :



The screenshot shows a web browser window with the URL 192.168.1.12:3000/console. The page title is 'Smart Home' and it includes a 'Disconnect' button. Below a blue header, there is a section for 'Informational Notes' and a 'Dashboard' tab. A link 'Add an Admin' is visible. The main content is a table with 9 columns: ID, login, Livingrooms, Bedrooms, Kitchens, Stairs, Garage, Installation date, and address. The table contains 6 rows of data for different admin users.

ID	login	Livingrooms	Bedrooms	Kitchens	Stairs	Garage	Installation date	address
1	Admin	3	4	2	1	Yes	2020-06-19	Mohammedia
2	adminX	2	2	1	2	No	2020-06-18	Mohammedia
3	ayoub	1	2	1	None	No	2020-06-15	Panoramique
4	Imane	2	2	1	2	Yes	2020-06-13	2mars andalous 1
5	ourayma	3	3	2	1	Yes	2020-06-19	Temara
6	ouraymaess	1	2	2	1	No	2020-06-15	temara

Figure 37. Interface de Dashboard

III. Application web responsive :

Après avoir parlé de la simulation, il est temps d'illustrer ce travail sous forme de belles interfaces faciles à utiliser.

Dans cette partie, nous allons décrire le scénario de l'expérience d'utilisation de notre application depuis un navigateur web quelconque, étape par étape.

1. Système d'authentification :

Tout d'abord, une fois l'application est ouverte sur le navigateur, l'utilisateur trouve directement la « Landing page » qui lui fait découvrir le rôle de cette application.

Cette dernière comporte deux actions principales :

- Le bouton « Get started », destiné aux utilisateurs de l'application; soit disant les habitants de la maison intelligente, leur permet de se diriger à l'interface d'authentification ;
- Le bouton « Espace société », destiné au SuperAdmin, et qui lui permet de se diriger à sa console réservée seul à la société.

La figure suivante présente l'interface de la « Landing page » :

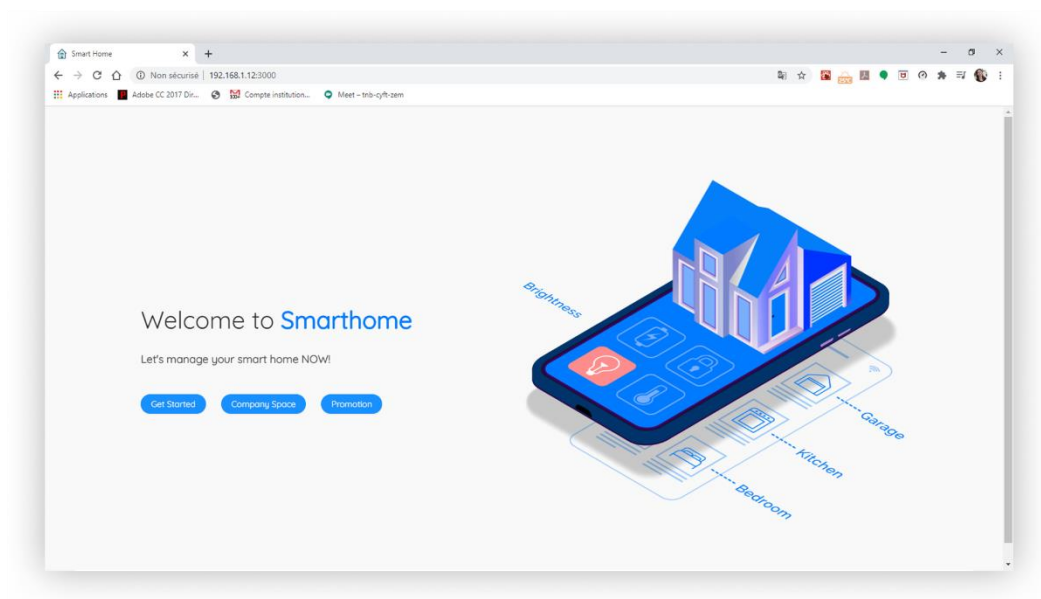


Figure 38. Interface Landing page

L'utilisateur clique sur le bouton « Get started » pour se diriger à la page d'authentification.

La figure suivante présente l'interface d'authentification :

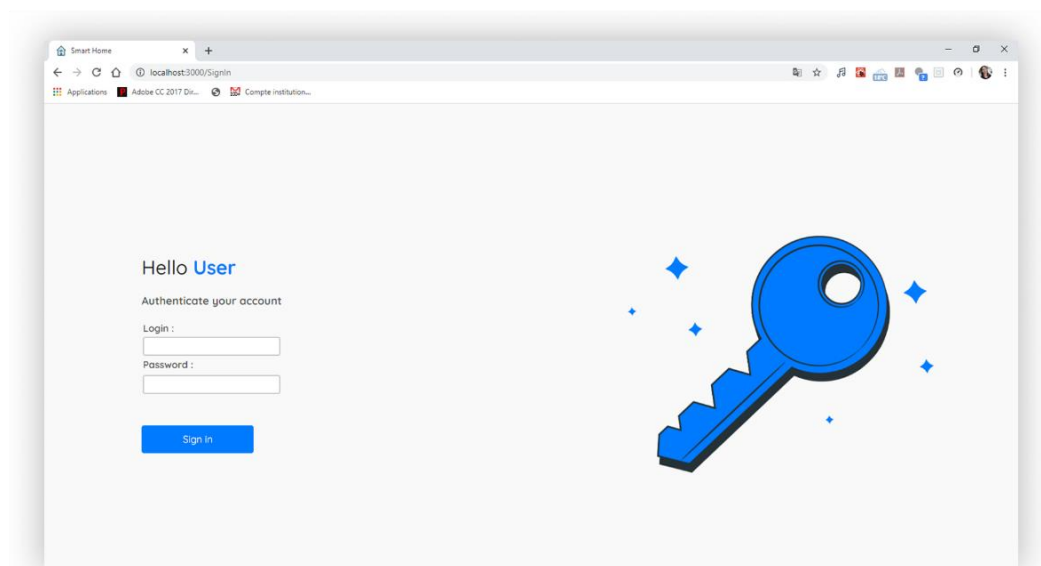


Figure 39. Interface d'authentification

Notre système d'authentification est connecté à notre base de données NoSQL de Firebase. Ceci permet de vérifier si les informations saisies correspondent à un compte que le SuperAdmin a déjà créé pour cet utilisateur, ou l'inexistence du compte. Une fois l'utilisateur réussit à s'authentifier, l'application renvoie l'interface d'accueil qui

correspond à son rôle, sinon un message d'erreur sera affiché, en spécifiant si l'erreur est dans le login ou bien dans le mot de passe.

2. Espace Admin :

Après l'authentification, l'utilisateur sera dirigé vers la page d'accueil présentée dans la figure suivante :

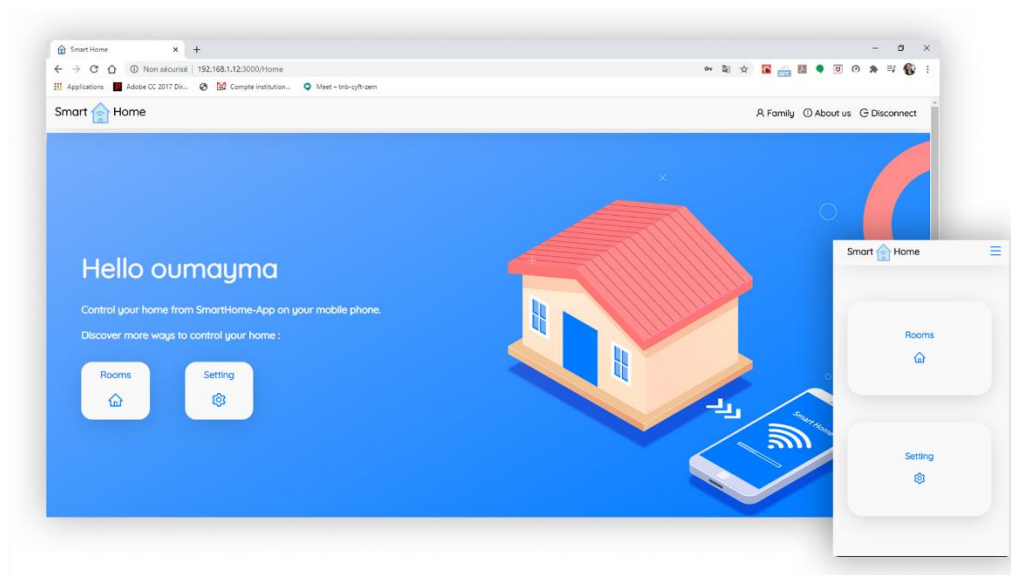


Figure 40. Interface d'accueil Responsive

Ensuite, l'utilisateur aura le choix entre deux actions :

- Se diriger vers les paramètres globaux des services de la maison en cliquant sur le bouton « Setting ». Ceci permet de se reporter sur un risque de Pluie, consulter ou changer l'état de l'alerte de feu, des lampes, de la porte principale ou du garage, etc.

La figure suivante présente l'interface des paramètres :

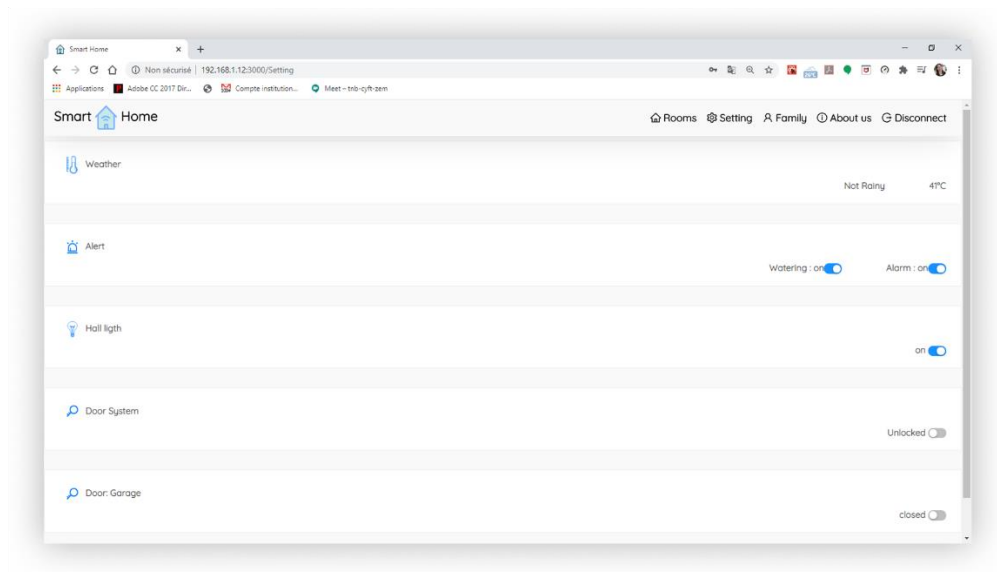


Figure 41. Interface de paramètres

- Se diriger vers les chambres de la maison en cliquant sur le bouton « Rooms ».

Les figures suivantes présentent l'interface des chambres personnalisées selon le plan de la maison de chaque utilisateur :

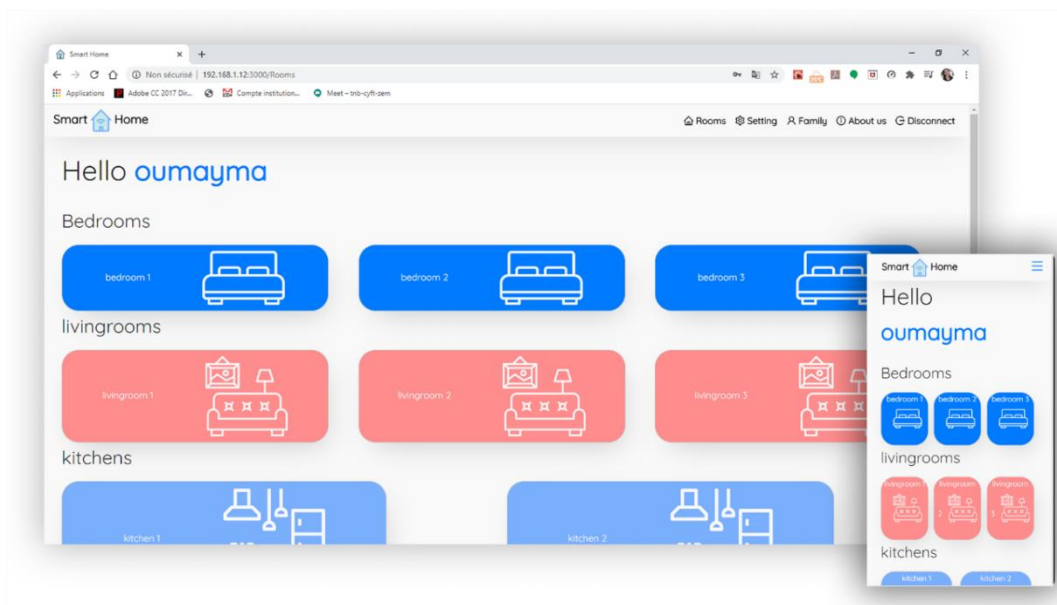


Figure 42. Interface responsive des chambres « User 1 »

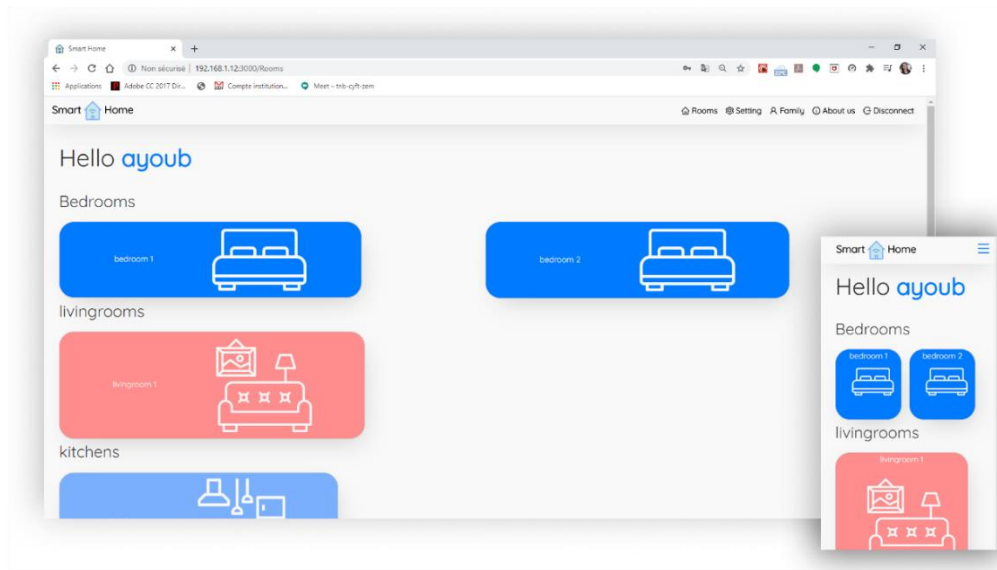


Figure 43. Interface responsive des chambres « User 2 »

L'utilisateur a donc la possibilité de contrôler des services implémentés dans chaque chambre individuellement.

Commençant d'abord par les deux interfaces; celle de la chambre à coucher et celle du salon, qui ont trois services semblables :

- Allumer / éteindre la lampe ;
- Allumer / éteindre le climatiseur et ajuster la température ;
- Ouvrir / fermer la fenêtre.

Après, viens l'interface de la cuisine qui comporte un seul service, celui d'ouvrir/fermer la fenêtre.

Les figures suivantes représentent quelques interfaces de ces derniers:

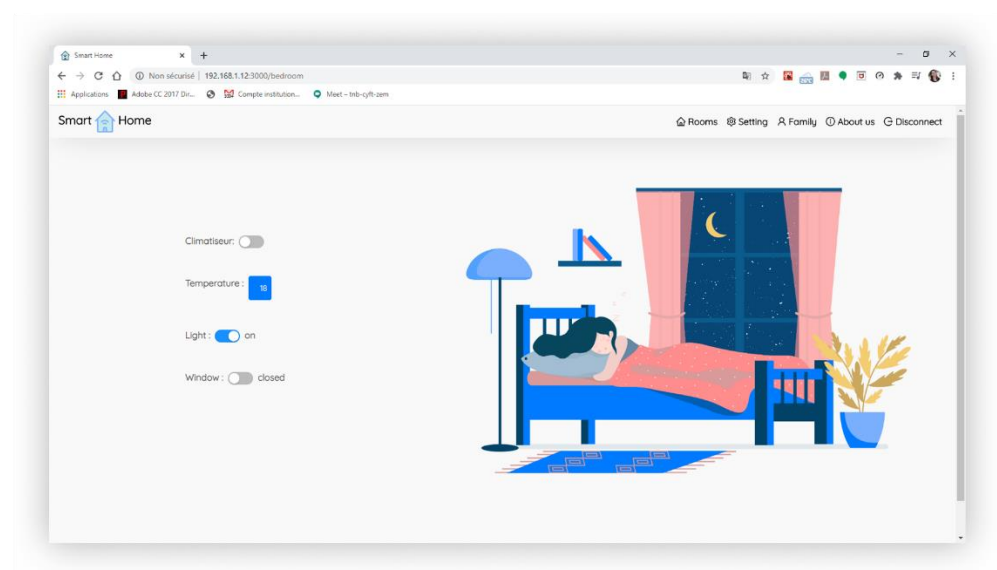


Figure 44. Interface Bedroom

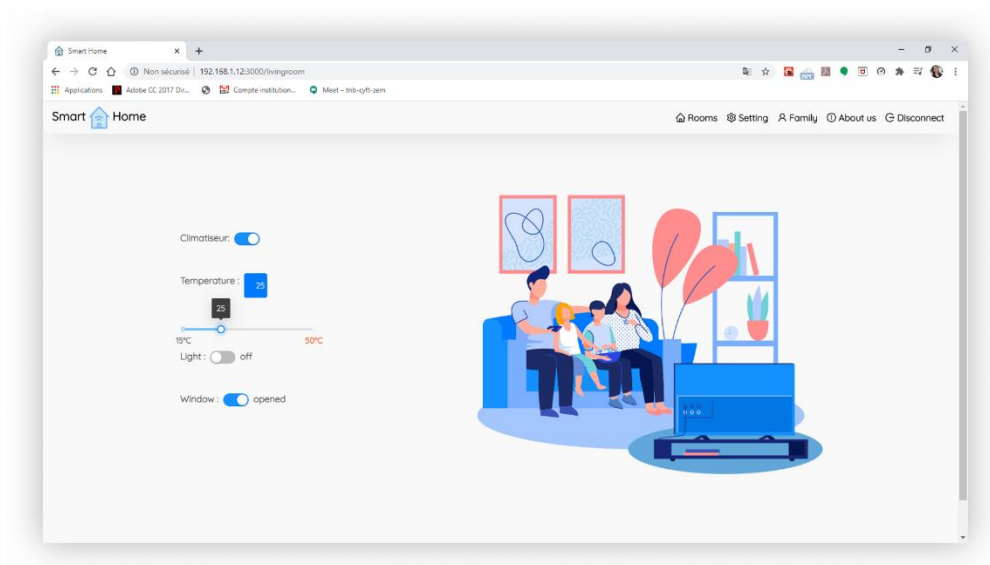


Figure 45. Interface Living room

L'utilisateur de type « Admin » a aussi le droit d'ajouter d'autres utilisateurs à sa maison, et ceci en leurs créant leurs propres comptes. Pour faire ceci, l'Admin clique sur le bouton « Family », et ajoute autant de membre qu'il souhaite.

Ces utilisateurs sont stockés dans la base de données avec leurs propres logins et mots de passe, plus la propriété « user » et en indiquant son admin.

La figure suivante présente l'interface de « Family » :

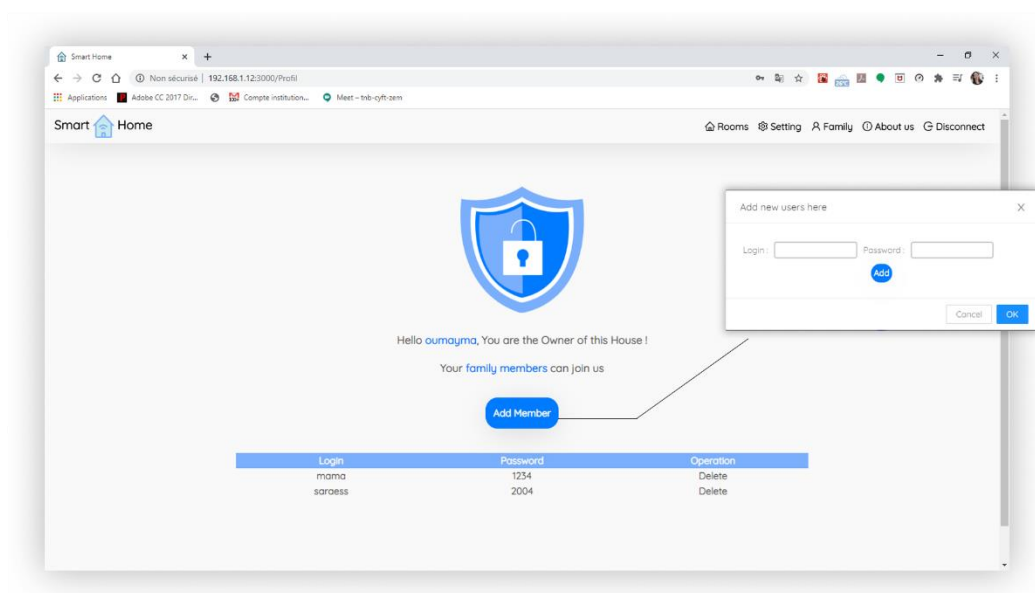


Figure 46. Interface Family

Nous n'avons pas oublié aussi les utilisateurs qui ne sont pas encore des clients en créant une page « About » dans l'application web, qui contient toutes les informations sur l'utilisation de cette application et ainsi les contacts important.

La figure suivante présente l'interface « About » :

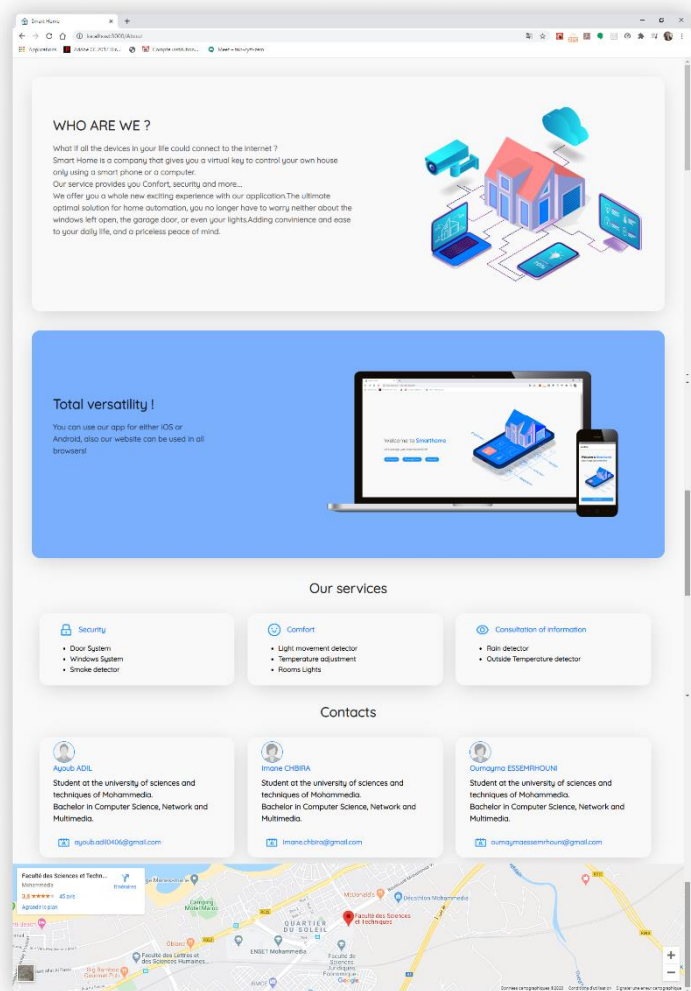


Figure 47. Interface About

IV. Application mobile : Android :

1. Introduction :

Comme pour l'application web, nous avons conçu une application mobile qui fonctionne sous Android et IOS, équivalente à un contrôleur interne et à distance des services de chaque maison. Ceci permettra à l'utilisateur de gérer plus facilement sa maison, simplement en installant l'application sur son téléphone mobile, sans avoir toujours besoin d'un navigateur.

2. Scénario de l'expérience utilisateur :

Une expérience utilisateur facile et claire dans une application est très importante.

La figure suivante représente la première partie du scénario de l'expérience d'un utilisateur avec l'application mobile :

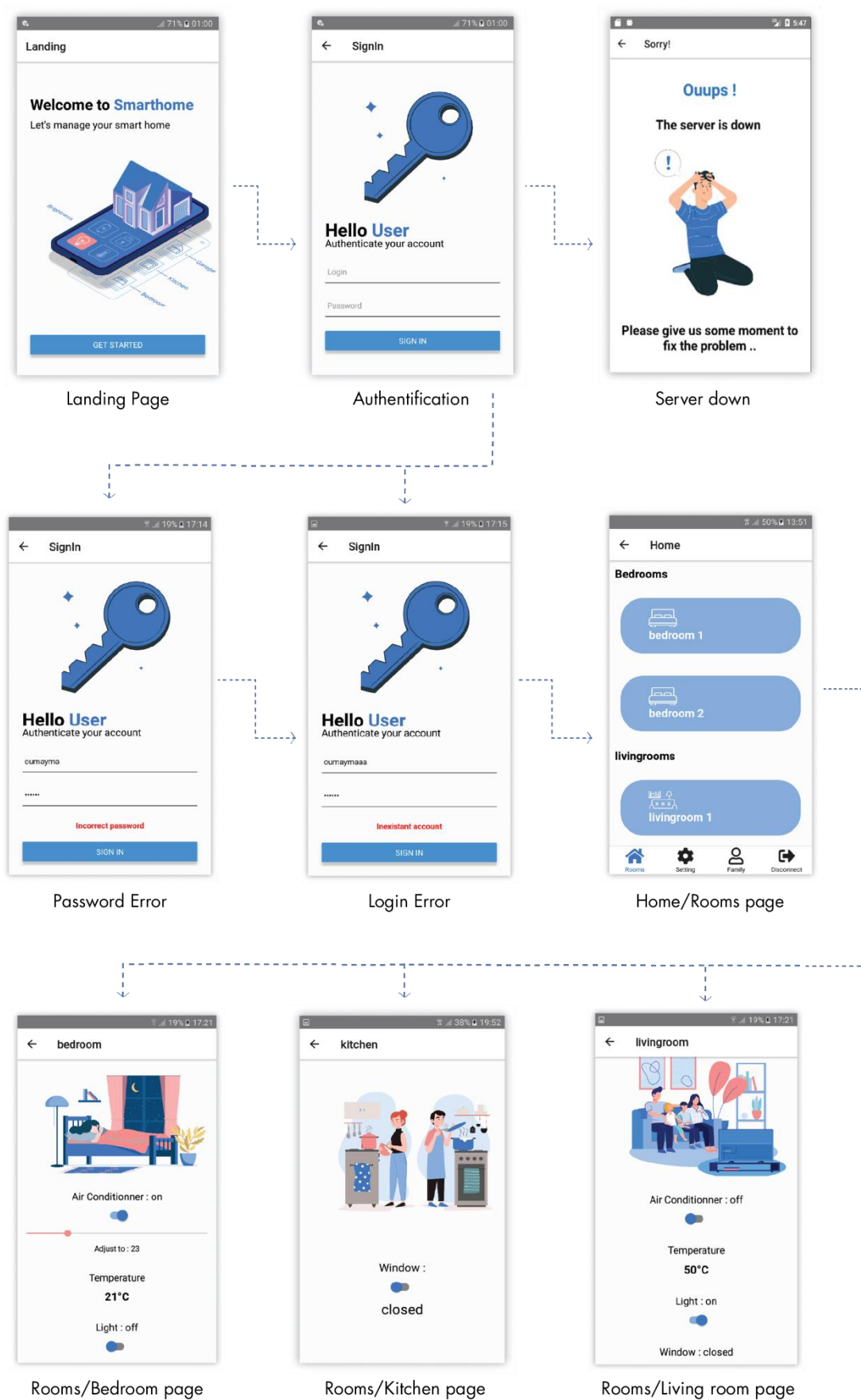


Figure 48. Scénario de l'application mobile (part 1)

La figure suivante représente la deuxième partie du scénario de l'expérience d'un utilisateur avec l'application mobile :

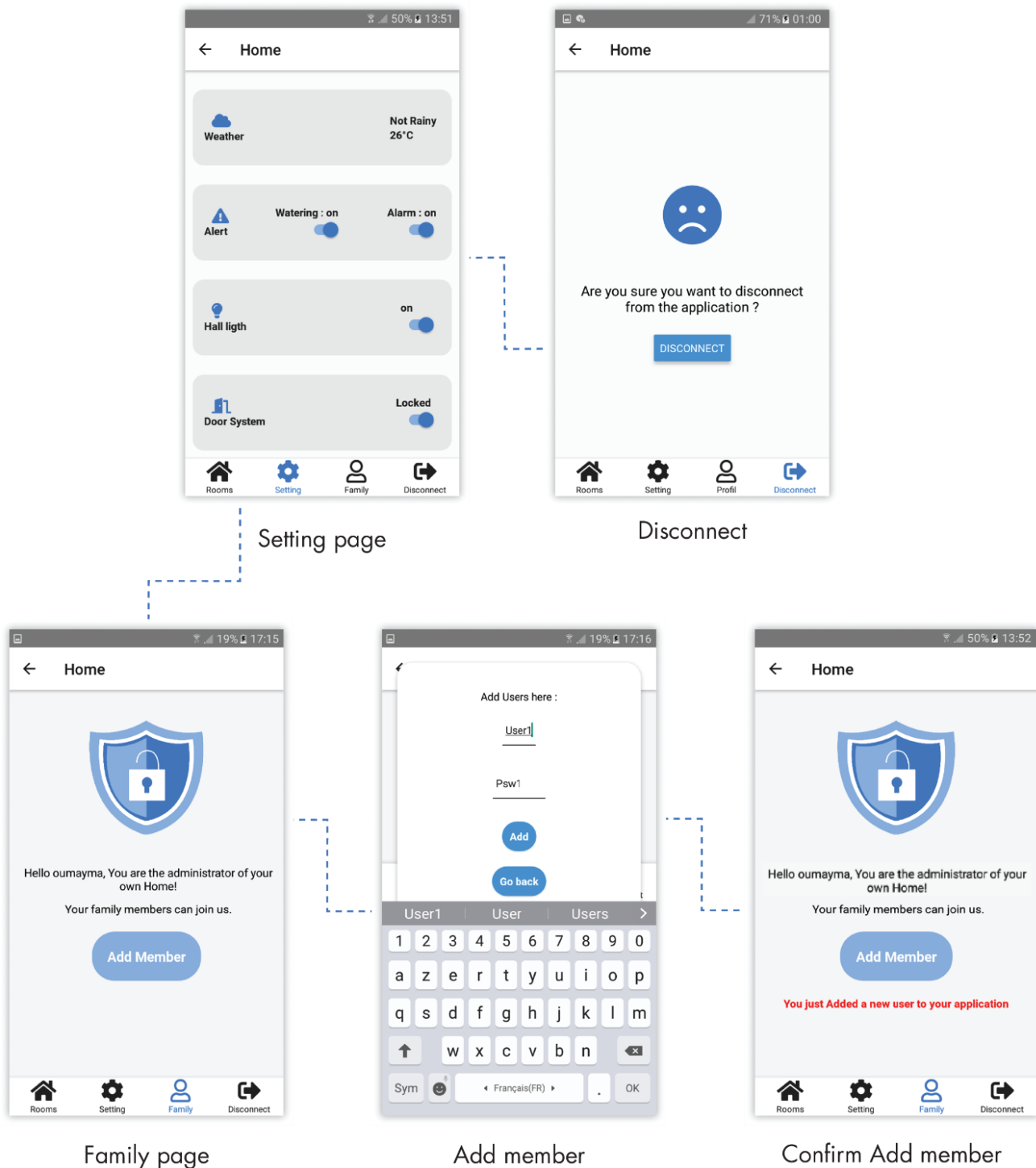


Figure 49. Scénario de l'application mobile (part 2)

Conclusion

Notre projet de fin d'étude est le fruit d'un travail de groupe incarné, plein de persévérance et d'innovation dans une période critique que notre pays a connu à cause du Covid-19. Ceci nous a pourtant appris plus à travailler en groupe et mener un projet collaboratif à distance en utilisant l'un des meilleurs outils utilisés par la plupart des développeurs dans le monde, en plus d'avoir l'opportunité de se former dans trois nouvelles technologies très importantes dans le marché de travail.

Tout ceci nous a permis d'acquérir plusieurs compétences en ce qui concerne le travail d'équipe et la gestion de projet en plus des compétences techniques en développement informatique, et surtout avoir une belle introduction au monde de l'internet des objets.

La solution que nous avons étudiée comporte le fonctionnement de plusieurs capteurs qui constituent pour nous des objets connectés, et nous avons développé une application web pour le SuperAdmin, représentant la société qui propose les services de la maison intelligente, et deux applications principales, web et mobile qui servent à contrôler à distance les services de la maison d'un habitant.

Cependant, ce projet est toujours en progression, puisqu'il reste quelques fonctionnalités que nous espérons ajouter par la suite, nous comptons aussi intégrer du matériel électronique à notre simulation pour une expérience en temps réel du fonctionnement de notre maison intelligente, mais malheureusement la période de stage a coïncidé avec le confinement.

Contre toute attente, la réalisation de ce projet dans cette période délicate n'était pas des moindres, mais les compétences uniques de chacun de nous, notre motivation et envie de donner le meilleur de nous-même nous a poussé à mener à bien cette épreuve.

Références

- <https://openclassrooms.com/fr/courses/5432586-explorez-le-panorama-des-objets-connectes>
- <https://openclassrooms.com/fr/courses/5432586-explorez-le-panorama-des-objets-connectes/5640571-familiarisez-vous-avec-le-vocabulaire-de-l-iot>
- <https://www.cisco.com/>
- <https://nodejs.org/en/>
- <https://create-react-app.dev/docs/getting-started/>
- <https://flask.palletsprojects.com/en/1.1.x/quickstart/>
- <https://ant.design/>
- <https://reactnative.dev/>
- <https://reactnavigation.org/docs/getting-started/>
- <https://firebase.google.com/>
- <https://www.adobe.com/products/xd.html>
- <https://trello.com/>