



UNIVERSITE SIDI MOHAMED BEN ABDELLAH
FACULTE DES SCIENCES ET TECHNIQUES DE FES

Projet de Fin d'Etudes
Licence Sciences et Techniques Génie Informatique

**Conception et réalisation d'une application Mobile
de supervision domotique**

Lieu de stage : Laboratoire LERSI

Réalisé par :

BELKARCHI Badr Eddine

Encadré par :

Pr Mohamed OUZARF

Pr Anas MANSOURI

Soutenu le 05/06/2018 devant le jury composé de :

Pr M.OUZARF

Pr I.CHAKER

Pr M.ZAHI

Année Universitaire 2017-2018

Remerciement

Tout d'abord, je tiens à remercier tout particulièrement et à témoigner toute ma reconnaissance aux personnes suivantes, pour leur dévouement et leur soutien dans la concrétisation de ce projet :

Mon encadrant du stage **Pr.Anas MANSOURI**, pour ses conseils éclairés, sa patience, sa disponibilité et avoir mis à ma disposition tous les moyens et matériels disponibles.

Mon grand respect et mes sincères remerciements vont à mon encadrant **Pr.Mohamed OUZARF**, pour toute l'aide et les conseils apportés durant ce projet.

Je remercie également les membres du jury qui ont accepté d'évaluer mon travail.

J'exprime enfin ma profonde gratitude à tous ceux qui ont participé de près ou de loin au bon déroulement de ce travail.

Merci à tous.

Table de matière

I.INTRODUCTION GENERALE	7
II. Chapitre 1 : Contexte Général du projet	9
1. Contexte	9
2- Problématique & objectif	11
3-Présentation du l’organisme d’accueil	12
4-Cahier de charge	13
5-Conclusion	15
III. Chapitre 2 : Analyse et conception du projet	16
1-Diagrammes des cas d’utilisation	16
2-Diagramme de séquences	18
3-Diagramme de classe	20
4-Environnement matériel du projet	21
a) Plate-forme Raspberry Pi 2	22
b) Capteur du température DS18B20 & Led.....	25
5- Environnement logiciel du projet	26
a) Entreprise Architect & Java	26
b) Python & SQLite	27
c) Android & Android Studio	28
d) XML	31
e) Protocole SSH	31
IV. Chapitre 3 : Réalisation du projet	32

1-Connexion client-serveur	32
2-Communication entre Raspberry et Java application par SSH.....	34
3- Présentation des différentes interfaces de l'application	36
V. Conclusion et perspectives	41
Bibliographie	42

Liste des figures

Figure 1 : système domotique

Figure 2 : Architecture du système

Figure 3 : Diagramme du Gantt

Figure 4 : diagramme de cas d'utilisation

Figure 5 : diagramme de séquences authentification

Figure 6 : diagramme de séquences ajout serveur

Figure 7 : Diagramme de classe

Figure 8 : Répartition des technologies

Figure 9 : Raspberry Pi 2 model B

Figure 10 : OS. Raspbian

Figure 11 : GPIO du raspberry pi

Figure 12 : Capteur DS18B20

Figure 13: LED

Figure 14 : Logo d'Enterprise Architect

Figure 15 : Logo de JAVA

Figure 16 : Logo de Python

Figure 17 : Logo de Sqlite

Figure 18 : Logo de S.E. android

Figure 19 : Statistiques d'utilisation du S.E. android

Figure 20 : Logo d'Android studio

Figure 21 : Code du saisie au terminal

Figure 22 : Configuration du SSH 1

Figure 23 : Configuration du SSH 2

Figure 24 : Librairie JSch

Figure 25 : Utilisation le protocole SSH en code Java

Figure 26 : Script Python lié au capteur du température

Figure 27 : Espace authentification de l'application

Figure 28 : Espace d'ajout serveur

Figure 29 : Consultation de l'état des différents capteurs et lampes

Figure 30 : Graph de la température durant les derniers 24 Heures

Figure 31 : données détaillés d'un capteur

Introduction Générale

Tout le monde rêve de pouvoir automatiser plusieurs fonctions de sa maison (allumer la lumière, ouvrir la porte de garage...). Il s'est ainsi développé un domaine particulier de l'automatique dédié à l'usage de la maison appelé domotique.[1]

La domotique vise à apporter des solutions simples et intuitives aux problématiques d'économie d'énergie, de sécurité et de confort en centralisant le contrôle des différents équipements d'une maison ou d'un bâtiment.[1]

La domotique exploite l'évolution des systèmes embarqués, des objets connectés, et les technologies de développement des applications mobiles.

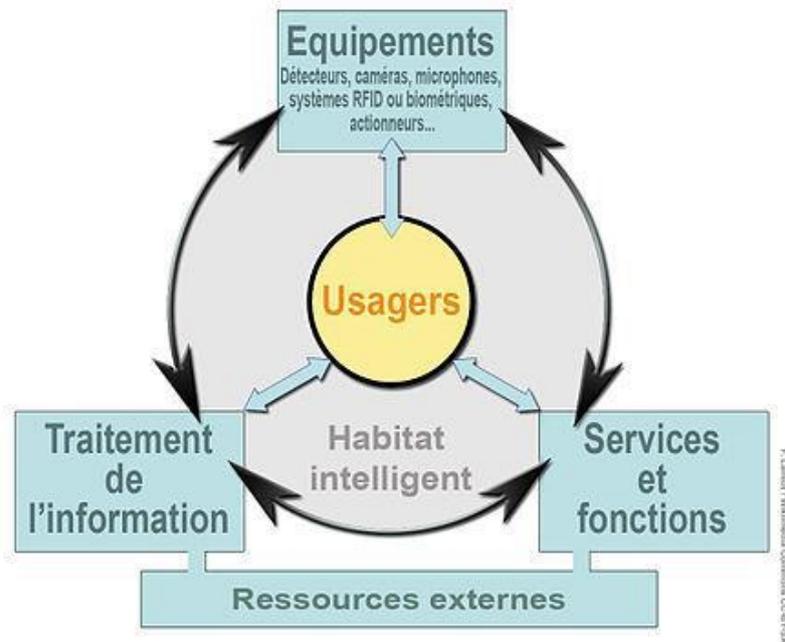


Figure 1 : système domotique

Dans ce contexte que s'inscrit ce projet de PFE, qui consiste à concevoir et développer une solution mobile connectée à une plate-forme embarquée, afin d'automatiser et de commander à distance l'environnement domotique de façon autonome.

Ce manuscrit est réparti en trois chapitres, Le premier chapitre présente le contexte général du projet, le deuxième chapitre illustre l'analyse et la conception du projet et le dernier chapitre contient la réalisation et le test de la solution proposée.

Chapitre 1 : Contexte Général du projet

Ce chapitre présente le contexte du projet, suivi par une présentation de l'organisme d'accueil, ensuite à savoir les problèmes et le cahier des charges de l'application réalisée durant cette période.

1-Contexte

Les premiers développements de la domotique sont apparus au milieu des années 1980 grâce à la miniaturisation des systèmes électroniques et informatiques. Dès lors, l'industrie a concentré ses expérimentations sur le développement d'automates, d'interfaces et d'outils apportant confort, sécurité et assistance au sein d'un édifice. [2]

Aujourd'hui, la domotique se définit comme étant l'ensemble des techniques et technologies permet tant de superviser, d'automatiser, de programmer et de coordonner les tâches de confort, de sécurité, de maintenance et plus généralement de services dans l'habitat individuel ou collectif. [2]

La domotique inclut également les notions de communications inter-équipements, d'environnement autonome d'interactions multimodales.

La domotique s'est imposée dans différents domaines, autant dans l'industrie que chez le particulier. Cependant, les entreprises peinent à diversifier leurs offres et la plupart des investissements industriels réalisés à ce jour dans ce domaine se sont soldés par des échecs.

Pour réaliser une installation domotique, les fabricants se concentrent sur les trois technologies suivantes:

- Les ondes radio (essentiellement le wifi ou la radio fréquence)
- Les courants porteurs en ligne (CPL)
- Les technologies filaires (bus)

Ce projet vise à exploiter les technologies des ondes radio, particulièrement les réseaux wifi afin de contrôler des différents objets domotiques connectés.

2- Problématique & objectif

Le projet a pour but de répondre à un besoin domotique, puisque l'utilisateur ou bien le client cherche à se connecter avec sa maison et savoir plusieurs facteurs, comme (température, humidité, l'état de la lumière..).

Même s'il existe bien sûr déjà des solutions commerciales, l'objectif de ce projet est de créer un système embarqué à faible cout, générique et modifiable facilement.

Dans ce projet qui suit l'architecture client-serveur ou le client est l'application mobile et le serveur est une plate-forme embarqué de type Raspberry Pi. L'utilisation de cette plate-forme rend le système extrêmement modulable, et ineffaçable avec d'autres équipements domotique (autres capteurs, des caméras, des moteurs...)

3-Présentation du l'organisme d'accueil

Laboratoire LERSI : Laboratoire Energies Renouvelables et Systèmes Intelligents

Sous la tutelle de l'université Sidi Mohamed Ben Abdellah, le laboratoire LERSI se compose de trois équipes :

- **Microélectronique et Systèmes Embarqués (MSE)**
- Système à Energies Renouvelables : Intégration Gestion Intelligence (SERIGI)
- Réseaux, Télécommunications et Ingénierie Logicielle (RTL)

Notre projet s'inscrit au sein de l'équipe **MSE**, Microélectronique et Systèmes Embarqués sous l'encadrement du Professeur Anas MANSOURI, Pr à l'ENSA de Fès.

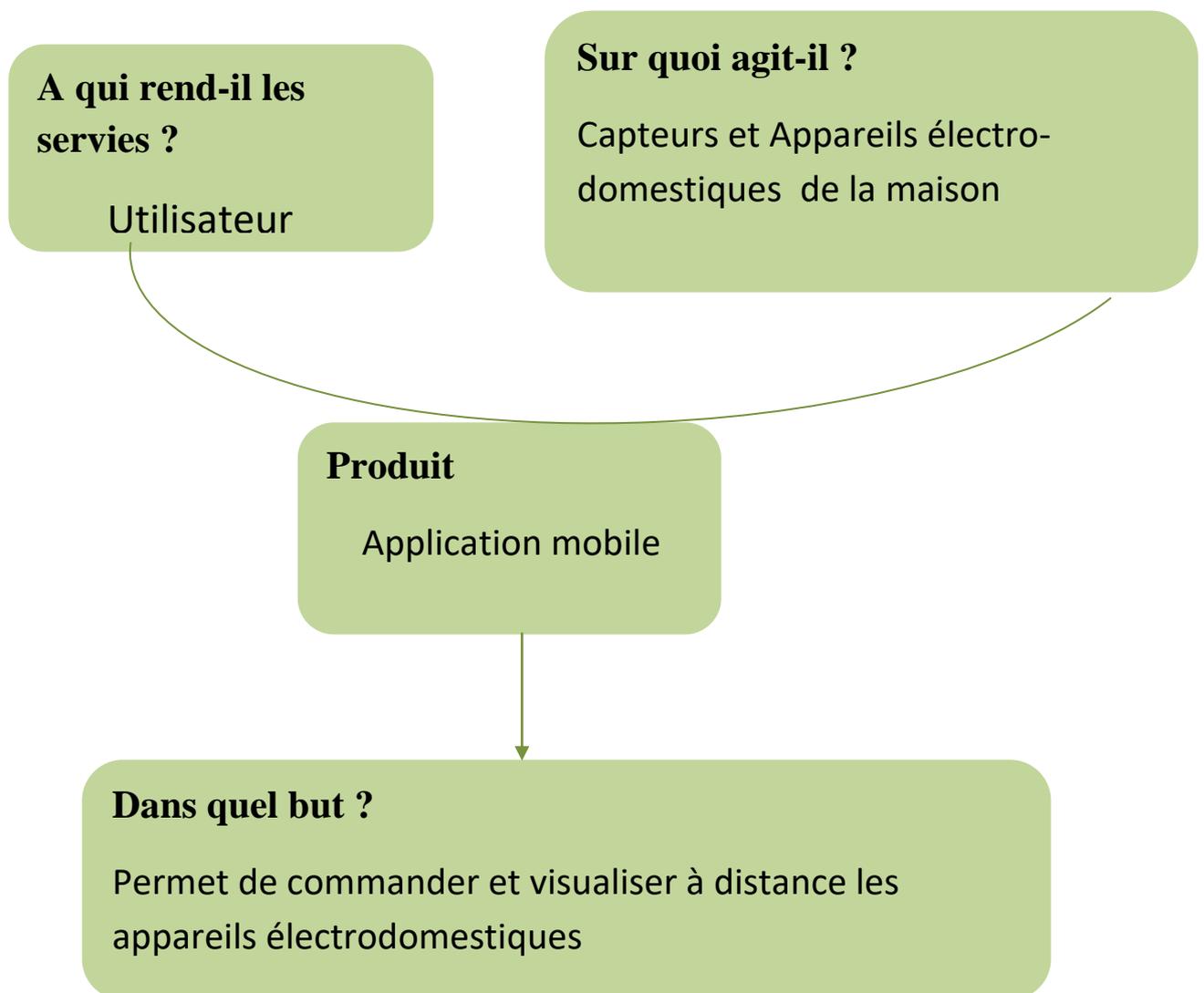
Les thèmes de recherche de l'équipe MSE sont :

- Microélectronique, Electronique des composants
- Systèmes embarqués temps réel, Co-design Hard/Soft
- Le système de transport Intelligent
- Traitements et compression d'images et de vidéo en temps réel
- CEM des circuits intégrés

4-Cahier de charge

L'objectif de ce projet est de mettre en place une application mobile pour le contrôle domotique à partir d'un Smartphone et en utilisant un serveur embarqué à base de plate-forme Raspberry Pi 2.

La figure illustre l'architecture globale du système



L'application doit communiquer avec le système à base réseau WIFI, ainsi avoir une authentification personnelle pour accéder aux différentes fonctions domotiques.

L'application doit permettre de commander et visualiser l'état d'appareils électrodomestiques et d'autres informations à distance. Plus précisément, il s'agirait de pouvoir commander à partir d'un Smartphone des appareils situés dans une habitation éloignée tels que des lampes, le chauffage, et visualiser des informations telles que la température de la maison.

L'application doit enregistrer les différentes valeurs de température durant 24h et affiche un courbe qui analyse la température, le gaz et l'humidité de la maison durant la journée.

Planification du projet :

L'organisation des taches effectuées tout au long de la période du stage se représentent dans le diagramme de Gant Suivant :

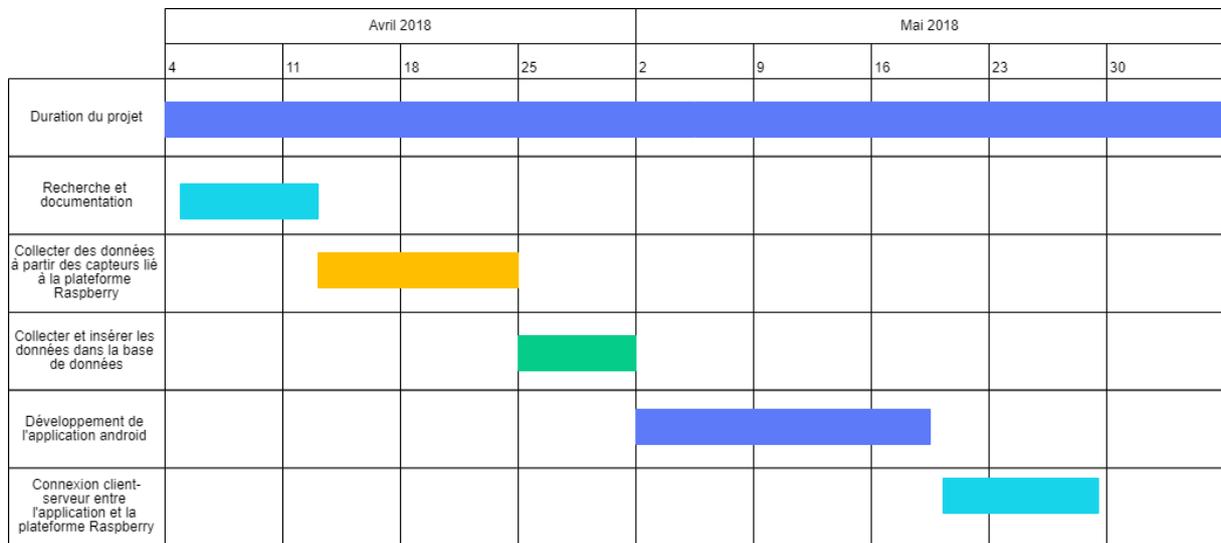


Figure 2 : Diagramme du Gantt du projet

Ce chapitre présente une description globale du projet, qui nécessite de fonctionner sur deux niveaux :

→ Niveau serveur : la plate-forme embarquée joue le rôle du serveur qui répond aux requêtes de l'application et la collection des données.

→ Niveau client : L'application mobile joue le rôle du client qui lance des requêtes à la plate-forme embarquée et d'acquisition des données.

Chapitre 2 : Analyse et conception du projet

Ce chapitre présente les différents modèles proposés pour la conception de l'application mobile en utilisant langage UML.

1-Diagramme de cas d'utilisation :

En analysant le cahier des charges et en fonction des contraintes imposé par le système, nous avons proposé le diagramme des cas d'utilisation suivant

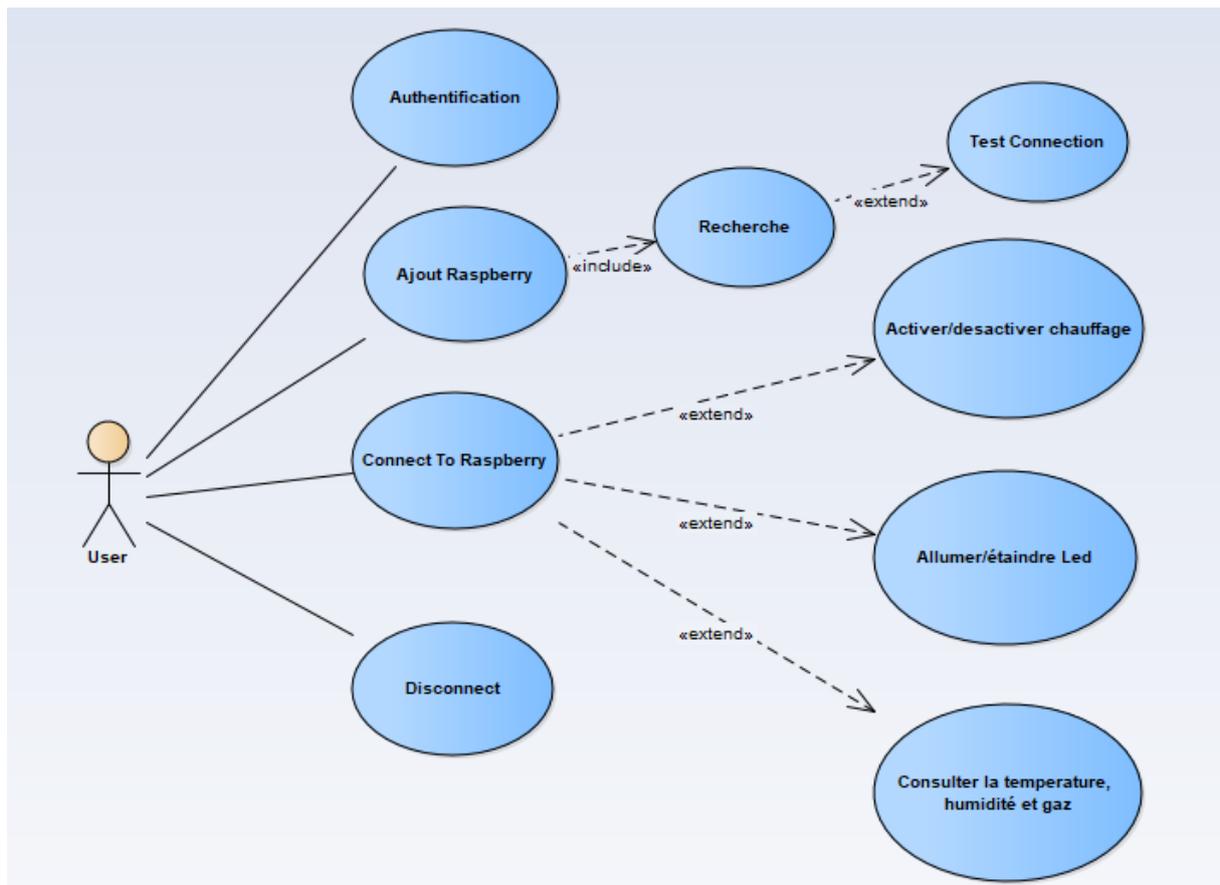


Figure 3 : diagramme de cas d'utilisation globale du système

2-Diagramme de séquences

Description des scénarios du cas d'utilisation

L'utilisateur saisit l'adresse email et un mot de passe pour qu'il puisse de se connecter, le système teste l'existence de cette identité dans la base de données et afficher soit :

- ✓ -les informations que l'utilisateur a introduit sont justes donc le système va afficher l'interface principale.
- ✓ -les informations sont fausses donc le système va afficher un message d'erreur.

La figure illustre les différentes séquences d'authentification

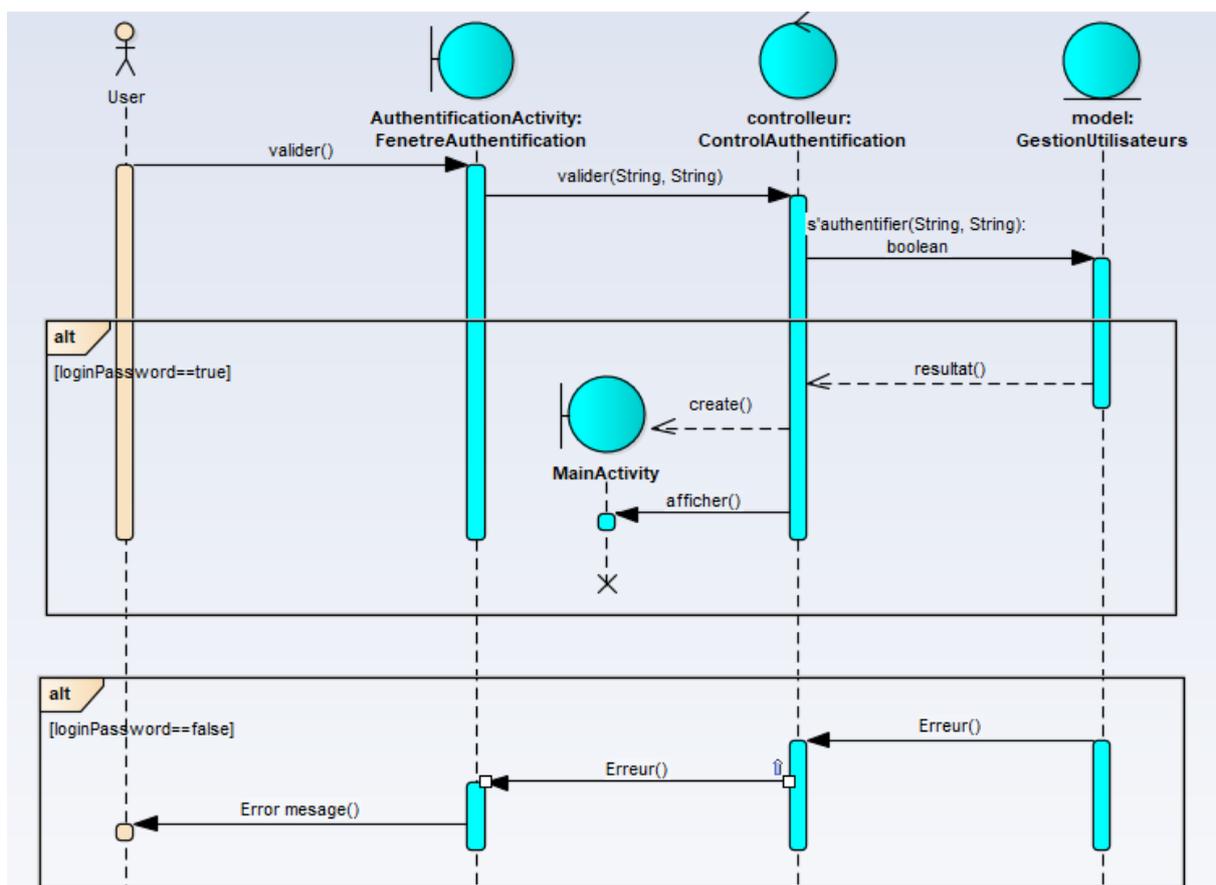


Figure 4 : diagramme de séquences authentification

Description des scénarios du cas d'utilisation :

Après l'authentification l'utilisateur doit ajouter les informations du serveur embarqué Raspberry pi.

Après la saisie de l'adresse IP, le nom, le mot de passe et port, Le système vérifie est-ce que ces informations sont-ils disponibles dans la base de données et on a 2 cas possibles :

- ✓ -Si les informations n'existent pas, il les ajout dans la base de données.
- ✓ -Si les informations sont déjà enregistrées dans la base de données il déclenche un message d'erreur.

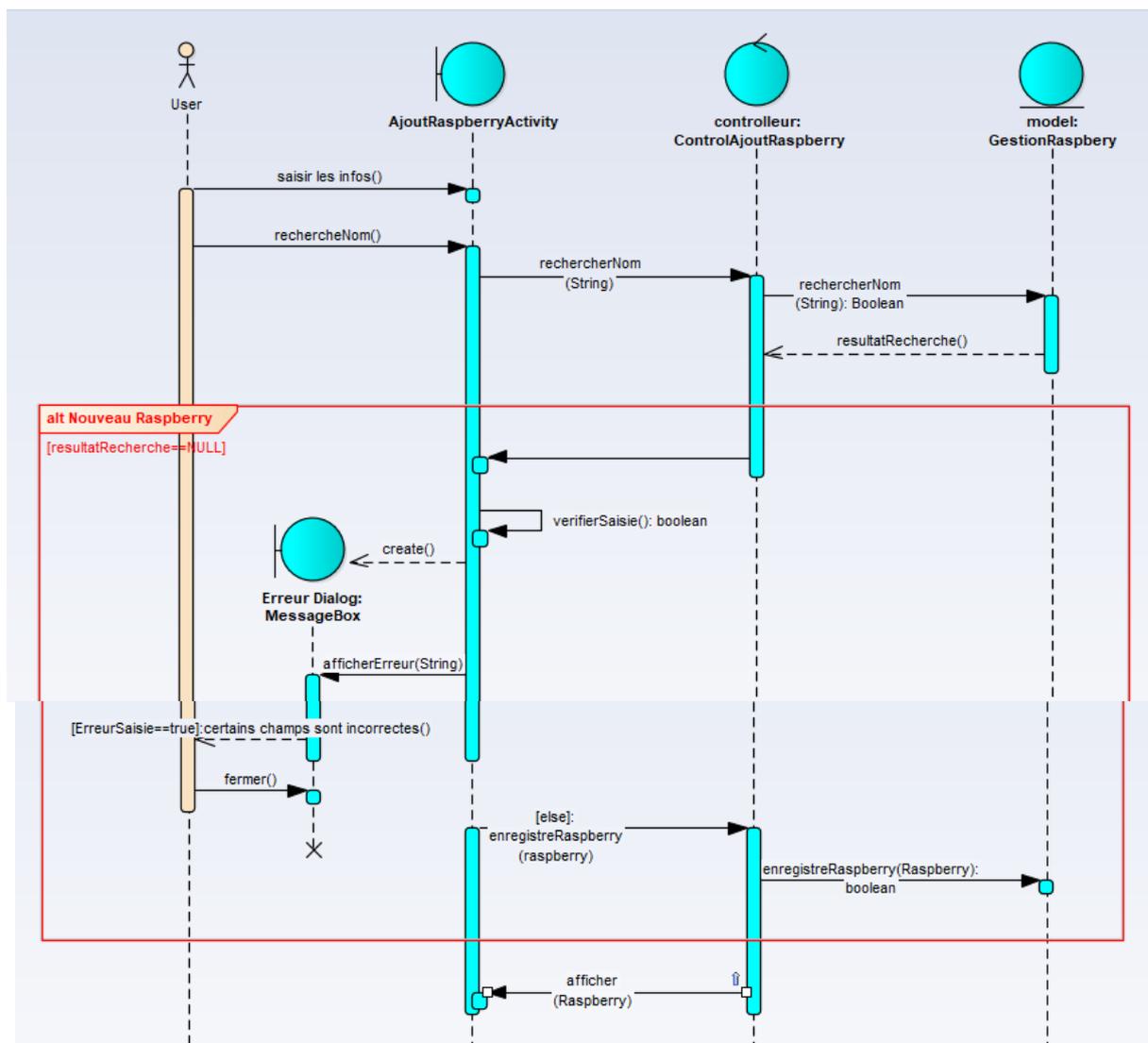


Figure 5 : diagramme de séquences ajout serveur

3-Diagramme de classe

Pour la réalisation des différentes fonctionnalités, nous proposons le diagramme de classe suivant

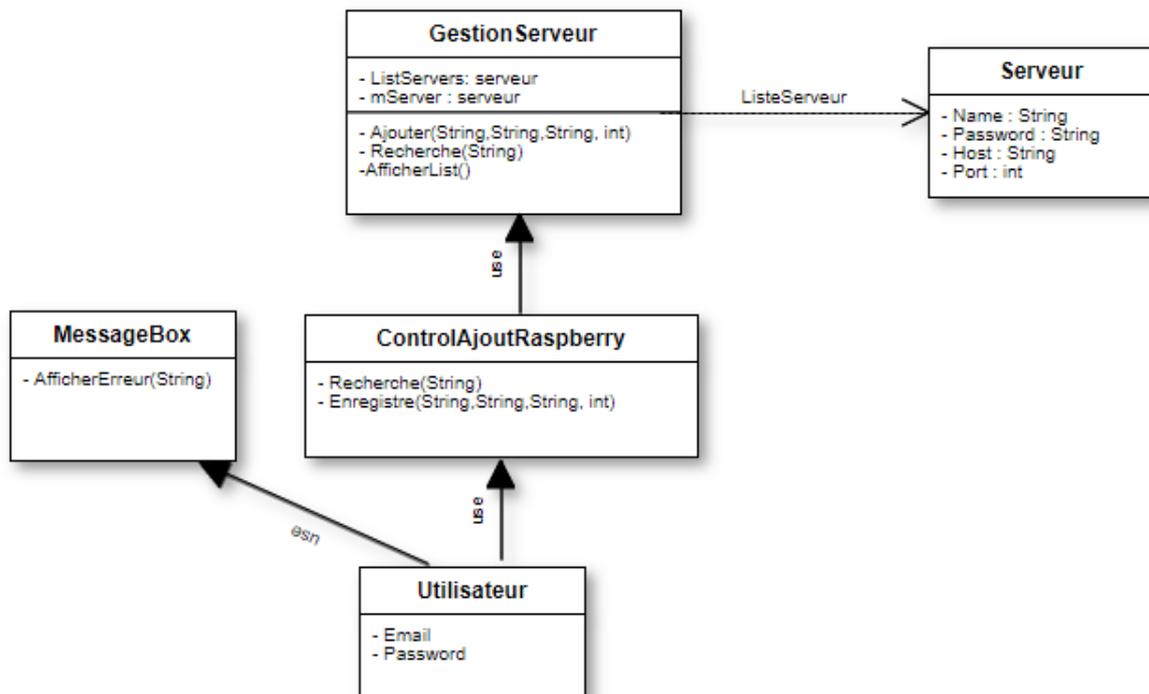


Figure 6 : Diagramme de classe

4-Environnement matériel du projet

La figure montre les différents outils utilisés dans ce projet

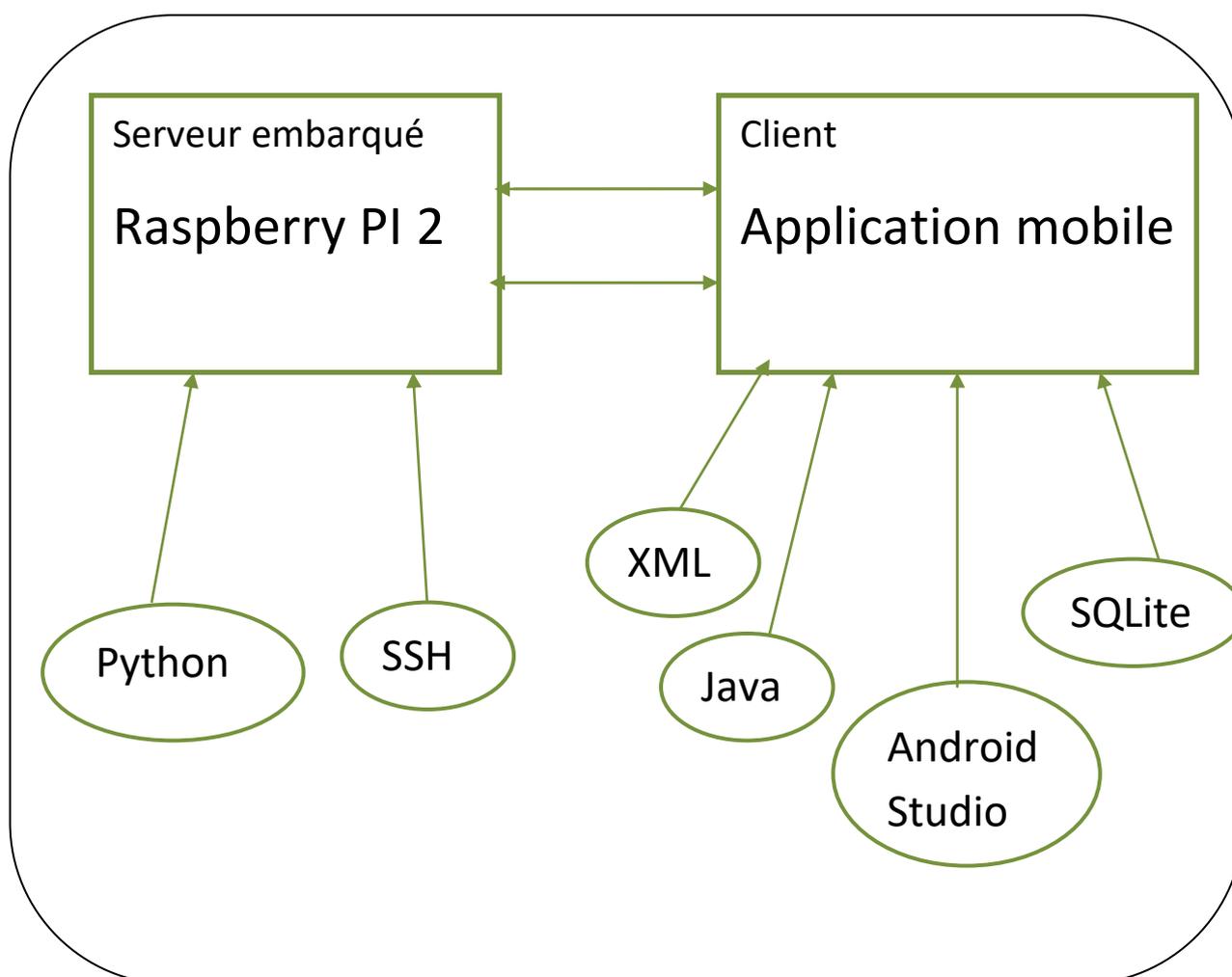


Figure 7 : Répartition des technologies

Plate-forme Raspberry Pi 2

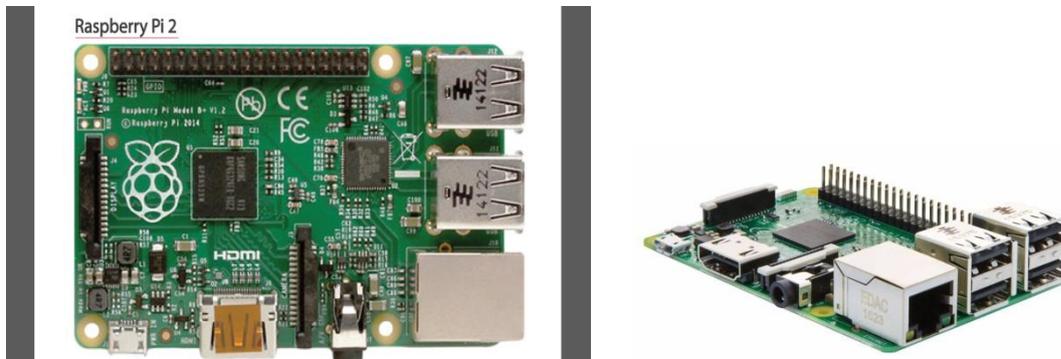


Figure 8 : Raspberry Pi 2 model B

Le Raspberry Pi est une plateforme embarquée très populaire dans les solutions domotiques. Il est actuellement décliné en trois modèles (**modèle A**, **modèle B**, **modèle B+**) se différenciant par leurs composants et ainsi leur prix. [3]

Cette carte présente les caractéristiques suivantes :

- Processeur : BroadCom quad-core cadencé à 900 MHz.
- Mémoire : 1 Go de RAM.
- Ports USB : 4 ports USB 2.0.
- Sorties vidéo : HDMI et Jack.
- Unité de lecture/écriture : MicroSD.
- Réseau : Ethernet.
- Dimensions : 85,60 mm × 53,98 mm × 17 mm.

Système d'exploitation

“Raspbian” s'agit d'un système d'exploitation GNU/Linux, spécialement conçu et optimisé pour les Raspberry Pi. Sa version actuelle est basée sur la dernière version de la distribution Debian. [4]

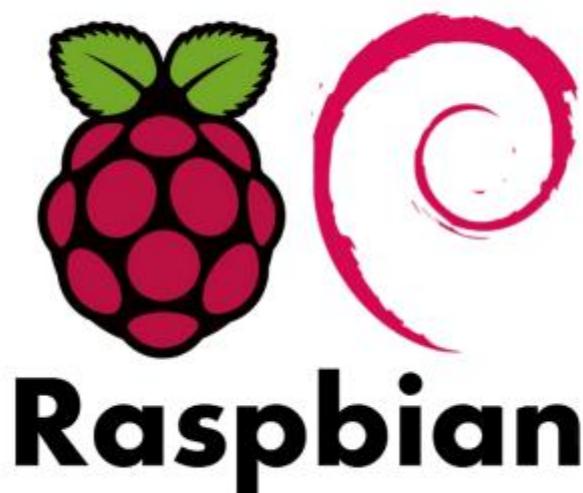


Figure 9 : OS. Raspbian

Entrées/sorties GPIO

La plate-forme Raspberry Pi donne accès à des entrées et sorties numériques appelées GPIO (General Purpose Input & Output) contrôlées par le processeur ARM.

Elles sont à usage multiple :

- En entrée numérique tout ou rien, pour détecter un interrupteur par exemple.
- En sortie numérique tout ou rien, pour activer un relais par exemple.

- En sortie numérique PWM, pour contrôler la puissance moyenne d'un led par exemple.
- En protocole I2C, d'échanger avec une ou plusieurs puces.
- En protocole SPI, idem.
- En protocole UART, d'échanger avec une seule puce (ou un pc).

D'autres usages sont possibles (audio PCM, vidéo sur les connecteurs DSI et CSI).

Plusieurs connecteurs donnent accès aux GPIO, mais le principal est un connecteur comportant 2 rangées de 13 picots mâles distants du pas standard de 2,54 mm. [5]

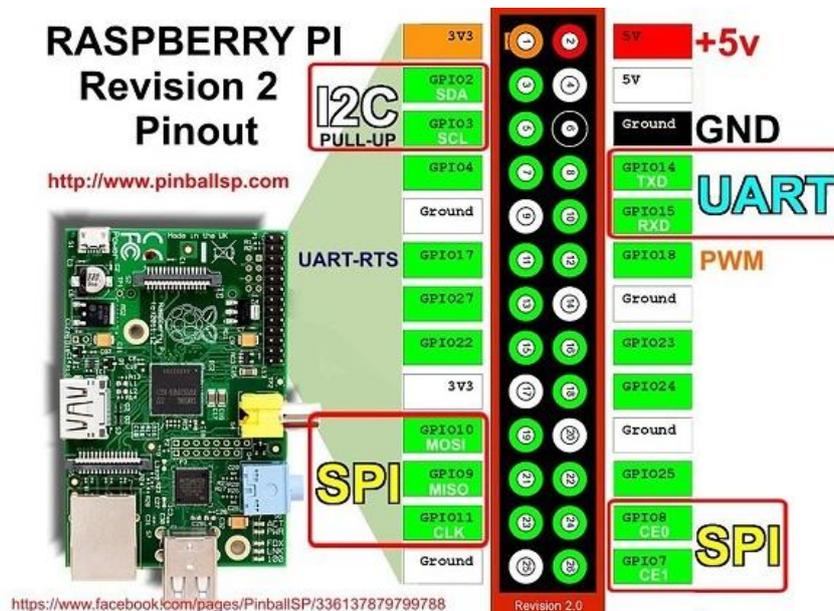


Figure 10 : GPIO du raspberry pi

Capteur de température DS18B20 & LED

Le capteur DS18B20 est un capteur de température numérique.



Figure 11 : Capteur DS18B20

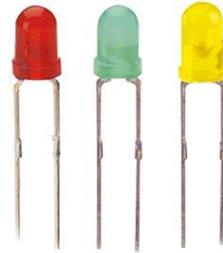


Figure 12: LED

Dimensions : 28mm x 12 mm X 10 mm

Tension de fonctionnement : 5V

Résolution : 9-12octets

Plage de mesure : -55C à 125C

Précision : 0.5C

5-Environnement logiciel du projet

Entreprise Architect



Figure 13 : Logo d'Enterprise Architect

Enterprise Architect un outil d'analyse et de création UML, couvrant le développement de logiciels du rassemblement d'exigences, en passant par les étapes d'analyse, les modèles de conception et les étapes de test et d'entretien. Cet outil graphique basé sur Windows, peut être utilisé par plusieurs personnes et conçu pour vous aider à construire des logiciels faciles à mettre à jour. Il comprend un outil de production de documentation souple et de haute qualité.[6]

Java

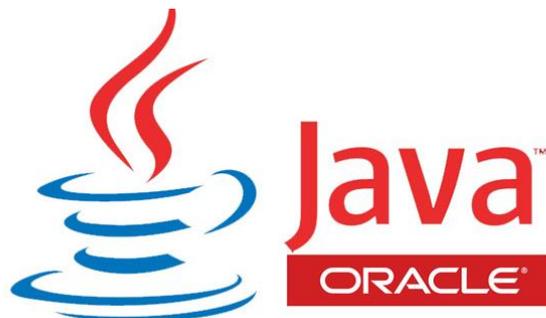


Figure 14 : Logo de JAVA

C'est un langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Java donne aussi la possibilité de développer des programmes pour téléphones portables et assistants personnels. Enfin, ce langage peut-être utilisé

sur internet pour des petites applications intégrées à la page web (applet) ou encore comme langage serveur (jsp). [7]

Python



Figure 15 : Logo de Python

Python est un langage de script de haut niveau, structuré et open source. Il est multi-paradigme et multi-usage. Développé à l'origine par Guido van Rossum en 1989, il est comme la plupart des applications et outils open source, maintenu par une équipe de développeurs un peu partout dans le monde. Conçu pour être orienté objet, il n'en dispose pas moins d'outils permettant de se livrer à la programmation fonctionnelle ou impérative. [8]

SQLite

SQLite est une base de données open source, qui supporte les fonctionnalités standards des bases de données relationnelles comme la syntaxe SQL. La base de données nécessite peu de mémoire lors de l'exécution (env. 250 ko), ce qui en fait un bon candidat pour être intégré dans d'autres environnements d'exécution. [8]



Figure 16 : Logo de Sqlite

SQLite sous Android :

SQLite est intégrée dans chaque appareil Android. L'utilisation d'une base de données SQLite sous Android ne nécessite pas de configuration ou d'administration de la base de données.

Vous devez uniquement définir les instructions SQL pour créer et mettre à jour la base de données. Ensuite, celle-ci est gérée automatiquement pour vous, par la plate-forme Android.

L'accès à une base de données SQLite implique l'accès au système de fichiers. Cela peut être lent. Par conséquent, il est recommandé d'effectuer les opérations de base de données de manière asynchrone. [8]

Android :



Figure 17 : Logo de S.E. android

Android est un système d'exploitation mobile basé sur le noyau Linux et développé actuellement par Google.

Lancé en juin 2007 à la suite du rachat par Google en 2005 de la *startup* du même nom, le système avait d'abord été conçu pour les Smartphones et tablettes tactiles, puis s'est diversifié dans les objets connectés et ordinateurs comme les télévisions (Android TV), les voitures (Android Auto), les ordinateurs (Android-x86) et les *smartwatch* (Android Wear).

En 2015, Android est le système d'exploitation mobile le plus utilisé dans le monde avec plus de 73 % de parts de marché dans les smartphones. [9]



Figure 18 : Statistiques d'utilisation du S.E. android

Le Java développement Kit (JDK)

C'est un ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être compilé, transformé en bytecode destiné à la machine virtuelle Java.

Android Studio



Figure 19 : Logo d'Android studio

Android studio est un nouvel environnement pour développement et programmation entièrement intégré qui été récemment lancé par Google pour les systèmes Android. Il a été conçu pour fournir un environnement de développement et une alternative à Eclipse qui est l'IDE le plus utilisé.

Android Studio offre :

- Un environnement de développement robuste.
- Une manière simple pour tester les performances sur d'autres types d'appareils.
- Des assistants et des modèles pour les éléments communs trouvés sur tous les programmeurs Android. Un éditeur complet avec une panoplie d'outils pour accélérer le développement de votre application.[\[10\]](#)

XML

'L'Extensible Markup Language', abrégé XML, est un langage de balisage permettant de définir différents espaces de noms, via l'usage de chevrons encadrant les balises

Protocole SSH

Le protocole **SSH** (*Secure Shell*) a été mis au point en 1995 par le Finlandais Tatu Ylönen.

Il s'agit d'un protocole permettant à un client (un utilisateur ou bien même une machine) d'ouvrir une session interactive sur une machine distante (serveur) afin d'envoyer des commandes ou des fichiers de manière sécurisée :

Les données circulant entre le client et le serveur sont chiffrées, ce qui garantit leur confidentialité (personne d'autre que le serveur ou le client ne peut lire les informations transitant sur le réseau). Il n'est donc pas possible d'écouter le réseau à l'aide d'un analyseur de trames.

Le client et le serveur s'authentifient mutuellement afin d'assurer que les deux machines qui communiquent sont bien celles que chacune des parties croit être. Il n'est donc plus possible pour un pirate d'usurper l'identité du client ou du serveur[11]

Chapitre 3 :

Après avoir exprimé les différentes fonctionnalités envisagées par l'application, ainsi que sa conception, ce chapitre présente la mise en œuvre des principales fonctions proposées pour tester le fonctionnement de l'application.

1-Connexion client-serveur

Premièrement, on configure une adresse IP statique au Raspberry Pi. Le réseau reconnaît le Raspberry pi à chaque fois qu'il se connecte par son adresse MAC, et il l'associe l'adresse IP choisi.

Ensuite, dans le terminal on saisit la commande suivante pour aller à l'interface de configuration

```
sudo raspi-config
```

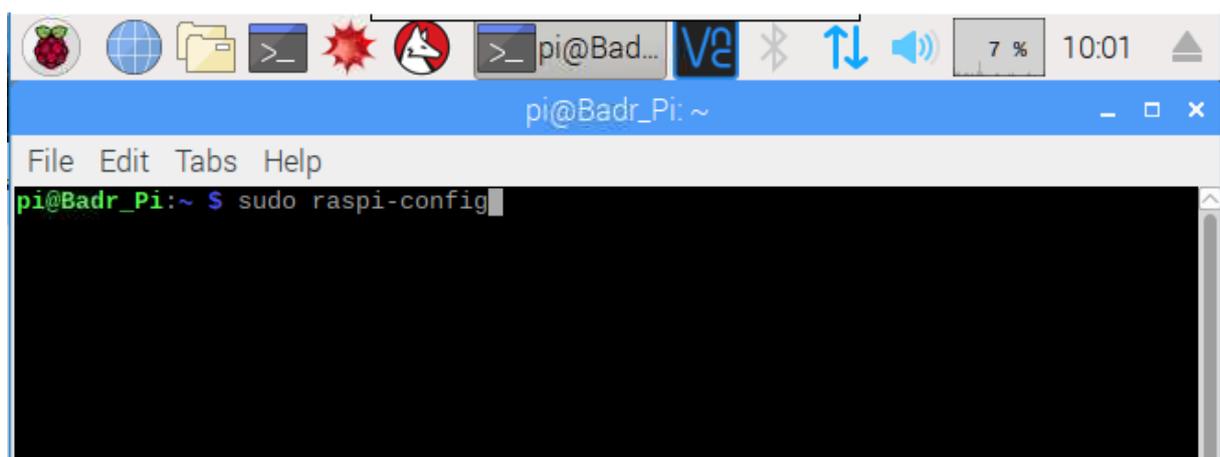


Figure 20 : Code du saisie au terminal

On active l'objet 'Interfacing Options', ensuite 'SSH'.

Comme le montre les figures suivantes

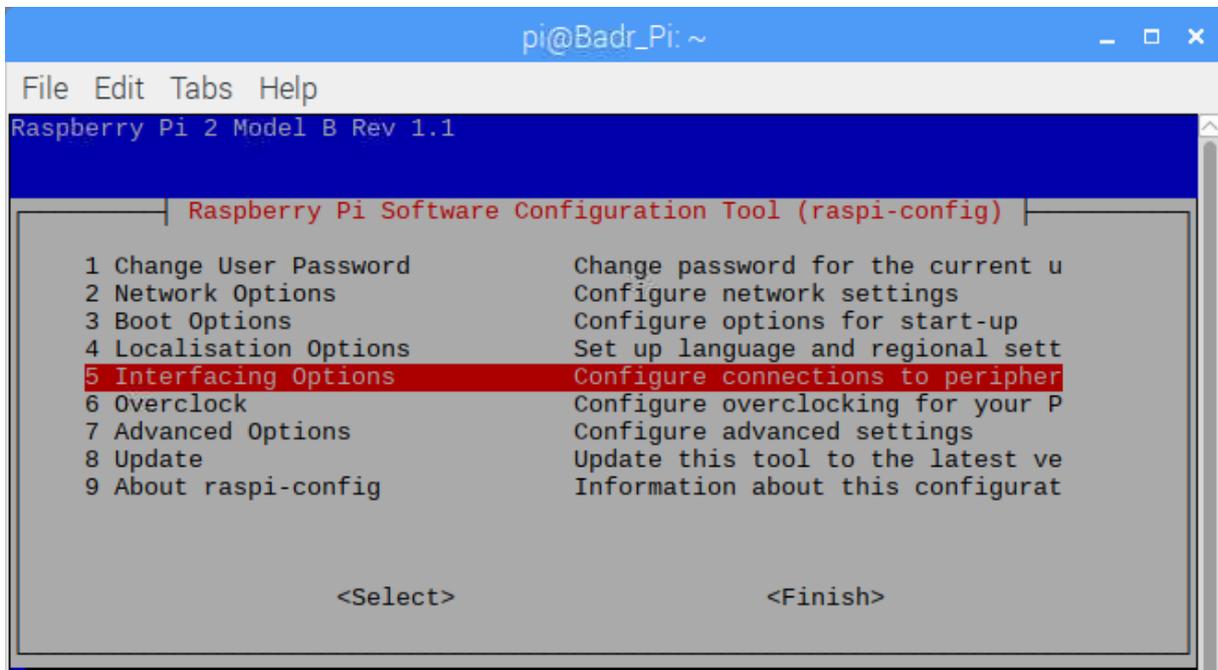


Figure 21 : Configuration du SSH 1

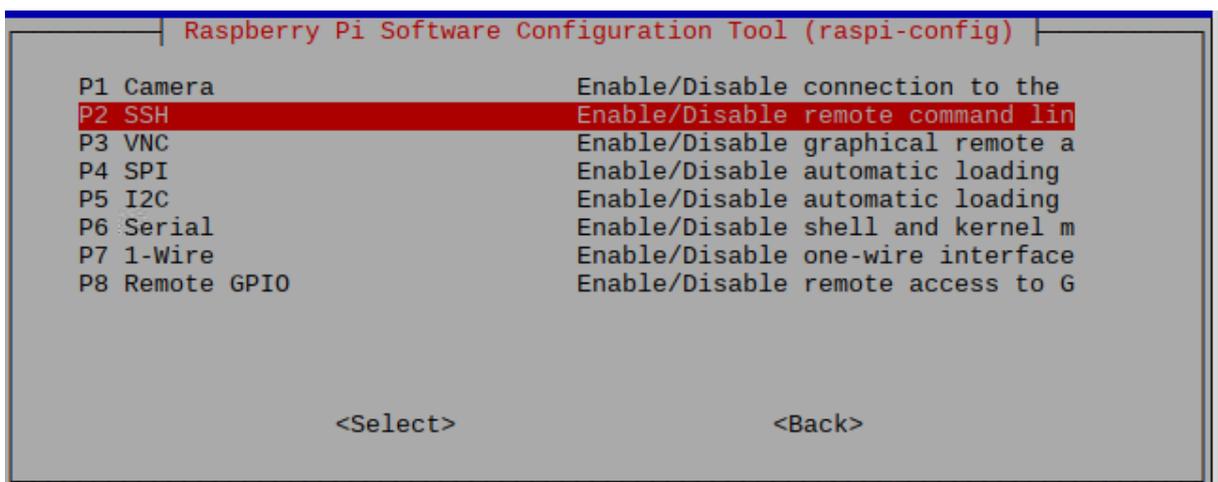


Figure 22 : Configuration du SSH 2

Puis Activer le protocole SSH.

Utilisation de la Librairie JSCH

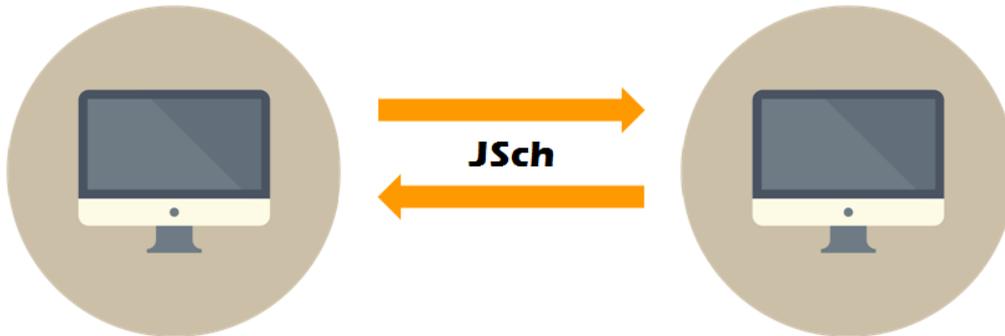


Figure 23 : Librairie JSch

JSch est une implémentation Java pure de SSH2. JSch vous permet de vous connecter à un serveur sshd et d'utiliser le transfert de port, le transfert X11, le transfert de fichiers, etc..., et vous pouvez intégrer ses fonctionnalités dans vos propres programmes Java. [12]

2-Communication entre Raspberry et Java application par SSH

```
try {
    JSch jsch = new JSch();
    Session session = jsch.getSession(currentUser /* pi */, currentHost /*192.168.1.10 */, currentPort /* 22 */);
    session.setPassword(currentPassword /* raspberry */);
    session.setConfig("StrictHostKeyChecking", "no");
    session.setTimeout(10000);
    session.connect();

    ChannelExec channel = (ChannelExec) session.openChannel( type: "exec");
    if (x == true) {
        channel.setCommand("./test 1");
        channel.connect();
    } else if (x == false) {
        channel.setCommand("./test 0");
        channel.connect();
    }
    channel.disconnect();
} catch (JSchException e) {
    e.printStackTrace();
}
```

Figure 24 : Utilisation le protocole SSH en code Java

Côté serveur, nous avons développé un script python qui prend en charge et l'acquisition des données du capteur de température, comme le montre la figure suivante

```
from w1thermsensor import W1ThermSensor
from time import sleep, strftime, time

sensor = W1ThermSensor()
while True:
    with open("temp.txt","a") as log:
        log.write("{0}?\{1}\n".format(strftime("%Y-%m-%d
%H:%M:%S"), str(sensor.get_temperature()))))
        log.close()
        sleep(30)
```

Figure 25 : Script Python lié au capteur de température

3-Présentation des différentes interfaces de l'application

Interface d'authentification

L'utilisateur s'identifie par une adresse email et un mot de passe, un message d'erreur s'affiche dans une boîte de dialogue si les entrées saisies sont erronés.(Figure 26)

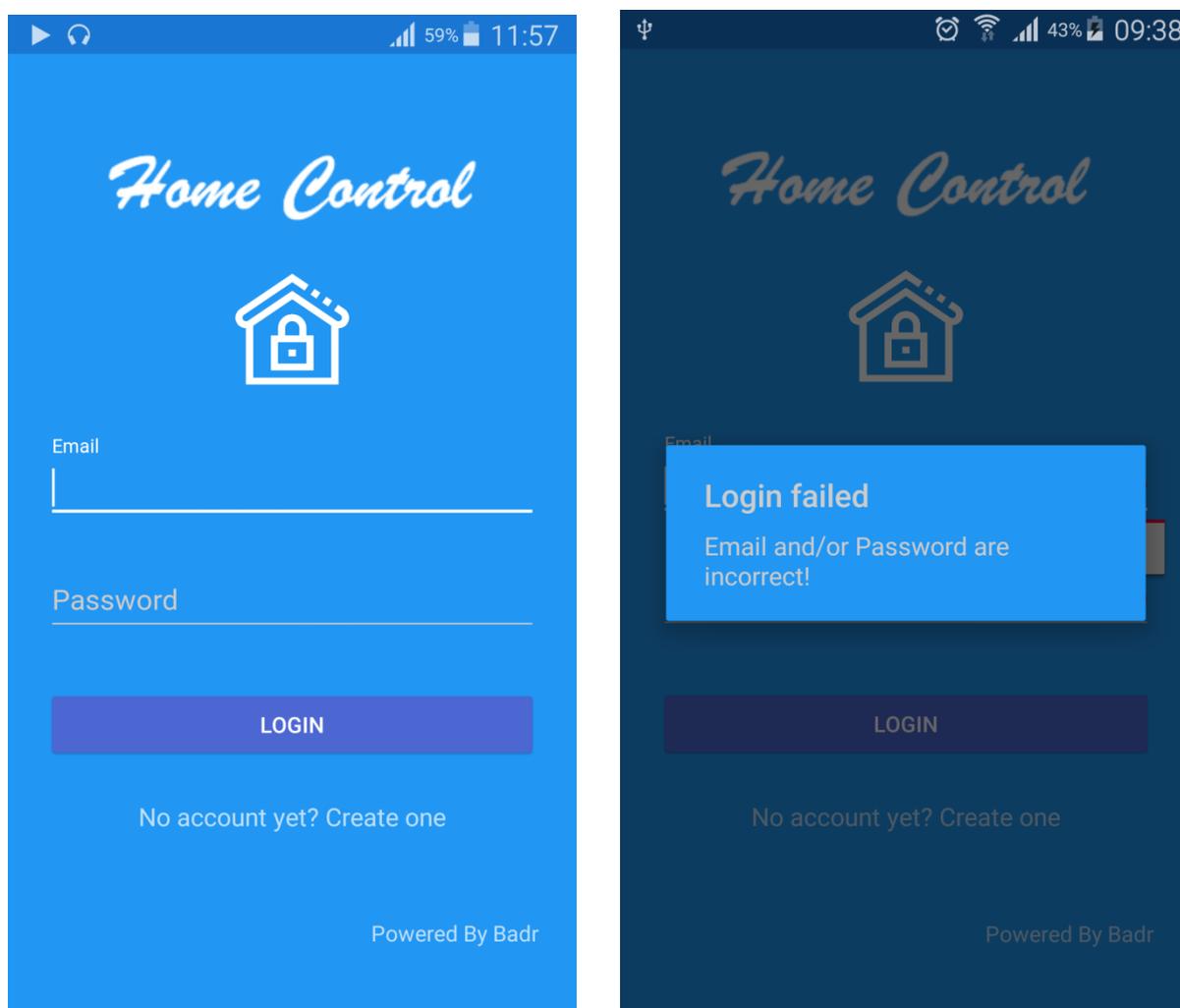
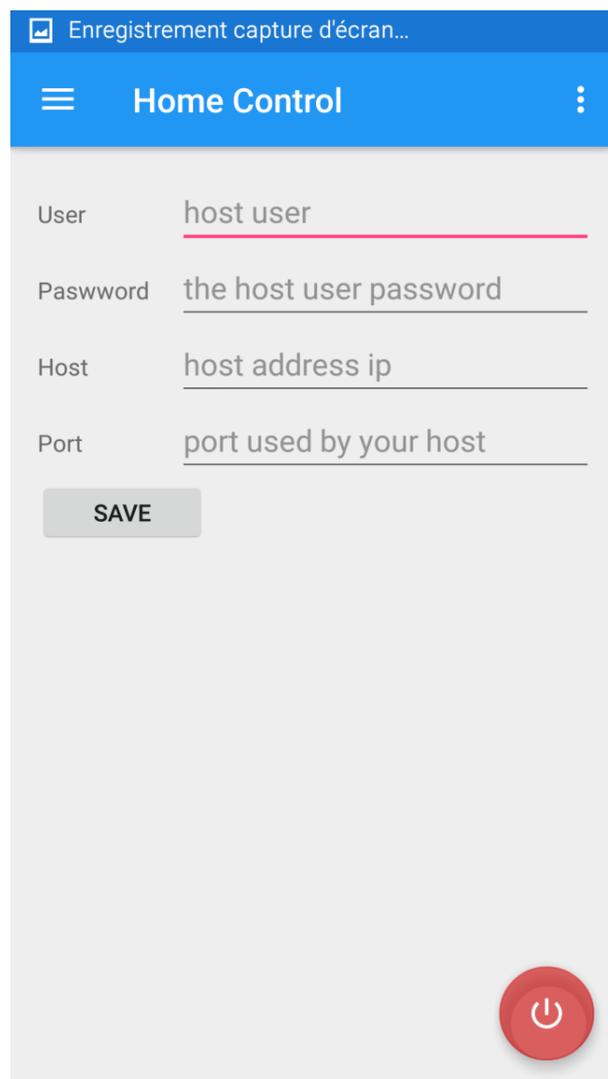


Figure 26 : Espace authentification de l'application

Interface d'ajout d'un serveur raspberry pi

L'utilisateur ajout les informations « Le nom du serveur, le mot de passe, l'adresse IP du serveur, et le port » .(Figure 27)

Pour que l'application mobile puisse se connecter au serveur Rapsberry Pi, ces données ajoutées seront enregistrés dans la base de données.



The screenshot shows a mobile application interface titled "Home Control". At the top, there is a blue header with a menu icon on the left and a three-dot menu icon on the right. Below the header, there is a form with four input fields: "User" with the value "host user", "Paswword" with the value "the host user password", "Host" with the value "host address ip", and "Port" with the value "port used by your host". Below the form is a grey "SAVE" button. In the bottom right corner, there is a red circular button with a white power icon.

Figure 27 : Espace d'ajout serveur

Interfaces principales

L'interface principale se compose d'un TabLayout de trois activités : (Figure 28)

→LIGHTS pour consulter l'état des Lampes et les contrôler aussi.

→SENSORS pour consulter Les informations globales des capteurs installés.

→GRAPH pour tracer une courbe d'évolution de la température durant les dernières 24 heures. (Figure 29)

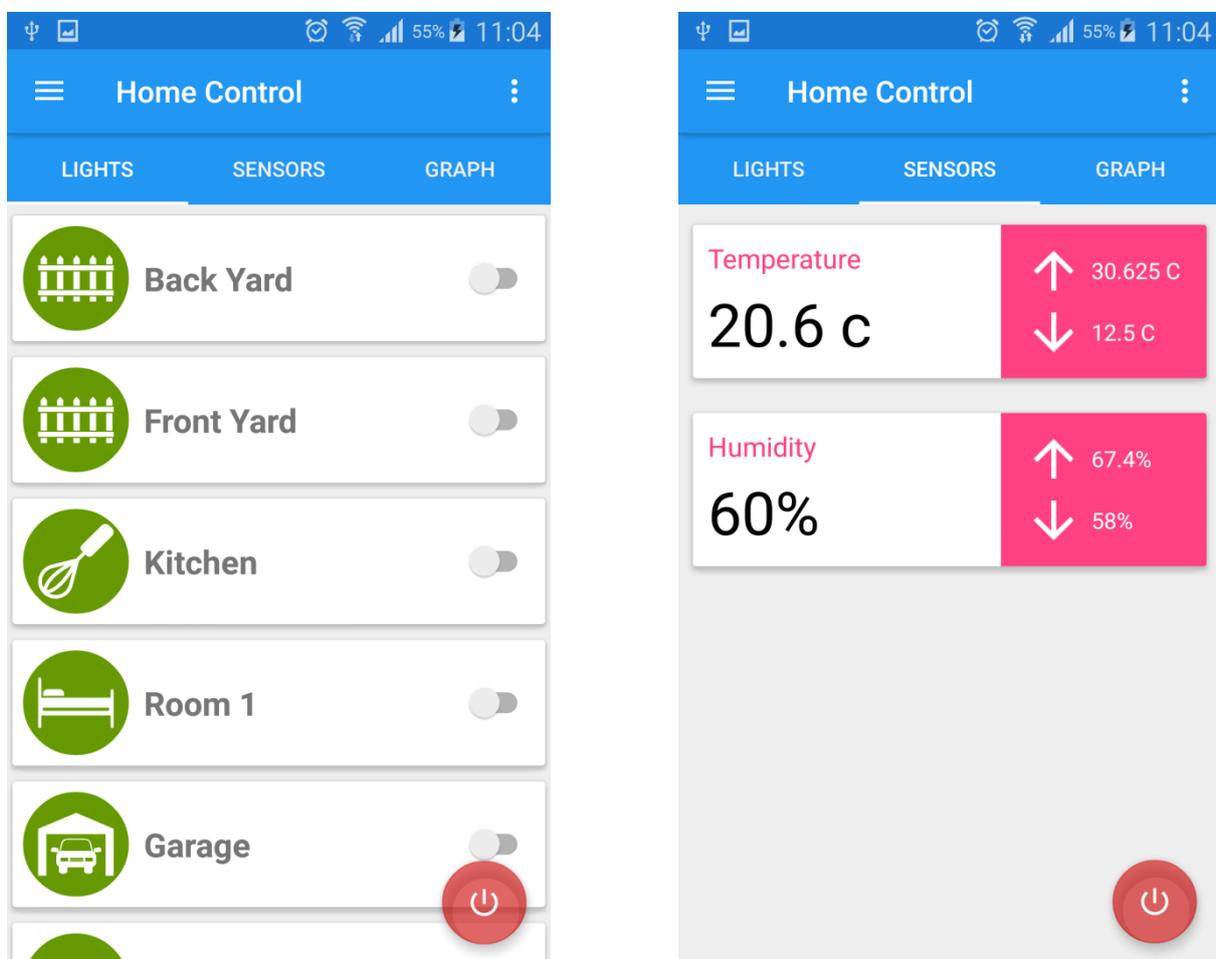


Figure 28 : Consultation de l'état des différents capteurs et lampes

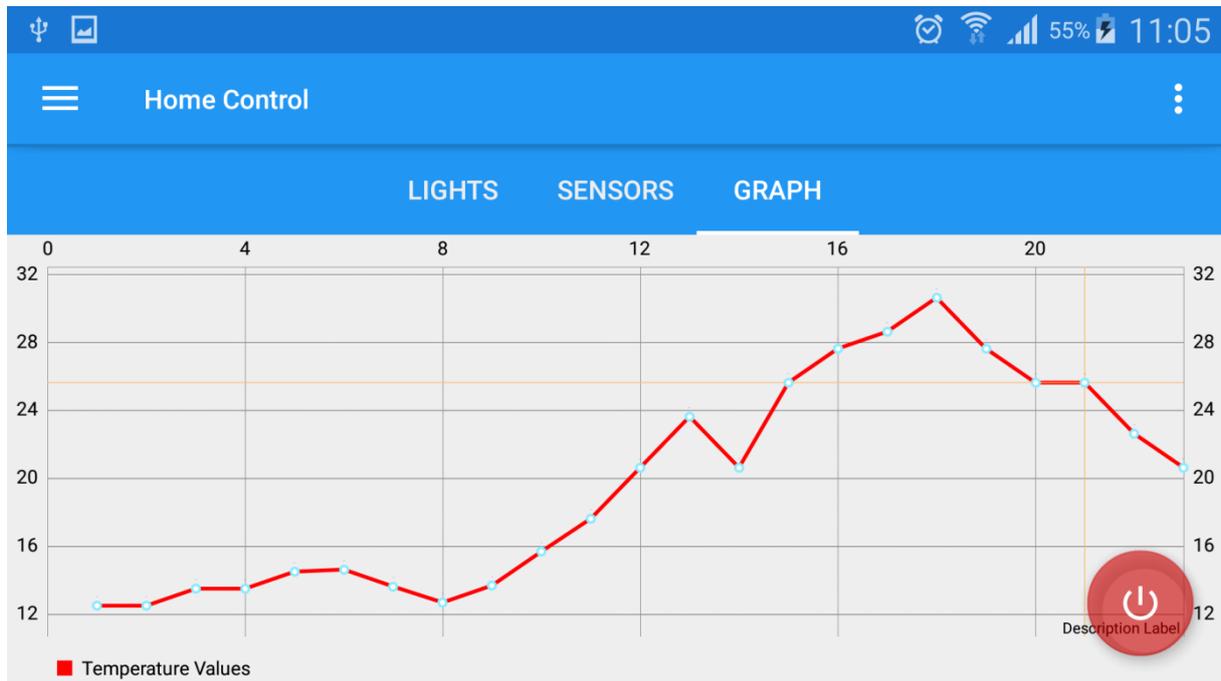


Figure 29 : Graph de la température durant les derniers 24 Heures

Les éléments de l'activité SENSORS sont cliquable, et amènent à une interface où l'utilisateur peut consulter en détail les données de la courbe.



Figure 30 : données détaillées d'un capteur

Conclusion et perspectives :

Ce projet vise à mettre en place une solution mobile à faible coût de contrôle domotique et exploite les dernières technologies dans le domaine de la connectivité.

Nous avons pu réaliser un système composé d'un serveur embarqué et un ensemble d'outil logiciel et matériel de haut niveau.

Notre système est générique et modifiable facilement et peut intégrer d'autres fonction comme l'ajout des caméras de surveillances et le contrôle de la consommation énergétiques de la maison .

Webographie

[1] <http://www.pobot.org/Entrees-sorties-GPIO-sur-Raspberry.html>

[2] <https://fr.wikipedia.org/wiki/Domotique>

[3] <http://www.jcraft.com/jsch/Readme.html>

[4] <http://www.domolink.ma/X10theorie.html>