Dossier Plante Connectées

Connected Flowers

Julien Khabbaz / Guillaume Faugeron / Quentin Fayet

Groupe F

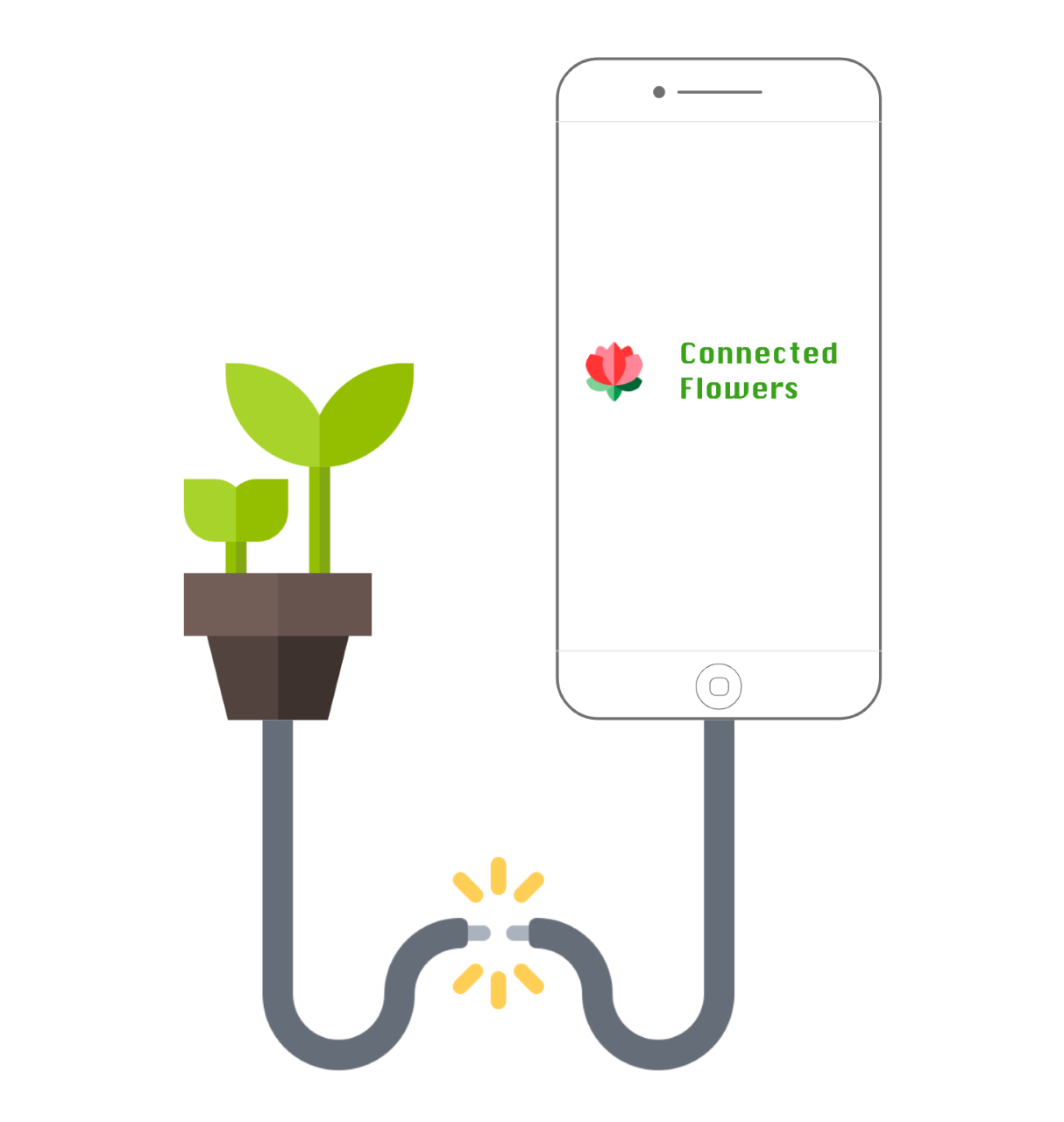


Table des matières

[Historique du document 2](#_Toc12391452)

[Introduction 3](#_Toc12391453)

[Objectif du document : 3](#_Toc12391454)

[Droit : 3](#_Toc12391455)

[Présentation de l'entreprise 4](#_Toc12391456)

[L’historique : 4](#_Toc12391457)

[L’équipe : 4](#_Toc12391458)

[Les objectif : 4](#_Toc12391459)

[Les outils : 4](#_Toc12391460)

[Présentation Projet 5](#_Toc12391461)

[Les besoins : 5](#_Toc12391462)

[Problématique : 5](#_Toc12391463)

[Axe de réflexion : 5](#_Toc12391464)

[Cahier des charges 6](#_Toc12391465)

[Description Fonctionnelle 7](#_Toc12391466)

[Architecture de la solution : 7](#_Toc12391467)

[Choix du matériel : 7](#_Toc12391468)

[Capteur d’humidité du sol 8](#_Toc12391469)

[Capteur de température 8](#_Toc12391470)

[Capteur de d’humidité 8](#_Toc12391471)

[Capteur de luminosité 8](#_Toc12391472)

[Méthode de fonctionnement : 8](#_Toc12391473)

[1. Collecte des données : 9](#_Toc12391474)

[2. Communication des données : 11](#_Toc12391475)

[3. Affichage des données : 11](#_Toc12391476)

[Bilan projet 13](#_Toc12391477)

[Problème rencontré : 13](#_Toc12391478)

[Ouverture : 13](#_Toc12391479)

[Conclusion : 13](#_Toc12391480)

# Historique du document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Auteur | Description |
| 1 | 26/11/2018 | Groupe F | Création du document |
| 1.1 | 13/01/2018 | Groupe F | Ajout cahier des charge |
| 1.2 | 16/04/2018 | Groupe F | Ajout description fonctionnel |
| 1.3 | 21/06/2018 | Groupe F | Mise à jour description fonctionnel |
| 1.4 | 25/06/2018 | Groupe F | Finition du document |
|  |  |  |  |
|  |  |  |  |

**Document Réalisé par le groupe F :**

Julien Khabbaz & Guillaume Faugeron & Quentin Fayet

**A l’attention de l’entreprise :**

Connected Flowers

# Introduction

## Objectif du document :

Ce document décrit la démarche de réalisation du projet plante connectées, ainsi que les objectifs de l’entreprise Connected Flowers.

## Droit :

Aucune partie de ce document ne peut être reproduite, archivée ou transmise sous aucune forme ou par quelconque moyen, électronique, mécanique, ou autre, sans accord préalable et écrit de Julien Khabbaz, Guillaume Faugeron ou Quentin Fayet.

# Présentation de l'entreprise

## L’historique :

Crée en 2018, Connected Flowers est une entreprise de solution logiciel pour entreprise et particulier. Constituer de 3 membres, notre établissement nouvellement crée a pour seul mots d’ordre « ambition, liberté, adaptabilité »

## L’équipe :

Un groupe de trois personne travaille en continu sur le projet plante connectée :

Guillaume Faugeron

* Guillaume Faugeron : chef de projet, expert phyton
* Quentin fayet : expert Arduino
* Julien Khabbaz : architecte application

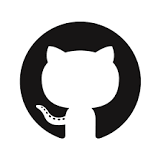
Quentin Fayet

## Résultat de recherche d'images pour "people icon" Les objectifs :

L’entreprise a pour ambition de révolutionner l’approche de la botanique, en simplifiant le suivi de croissance des plantes. Connected Flowers espère dans les cinq années à venir devenir un acteur incontournable du marché de la botanique-numérique.

Julien Khabbaz

## Les outils :

 La réalisation de ce projet avait pour exigence un travail d’équipe et une organisation sans faille pour des besoins de productivité et pour le bien des clients. A cette fin nous nous sommes servi d’outils de communication tel que discord pour un partage de fichiers et de la communication de travail dans l’équipe, ainsi que de Skype entreprise pour réaliser nos réunions de planification et préparation. De plus, nous nous servons de la suite office 365 et plus précisément de One drive, pour stocker et partager les données volumineuse et importante (son principal intérêt est une gestion des sauvegardes simplifié pour toute l’équipe). Enfin GitHub nous aura aidé à sauvegarder notre code pour le faire vérifier par la communauté ainsi que de se servir de librairie partagée sur GitHub.

Une librairie est un code à qui le programme fait appel pour faire fonctionner un capteur

# Présentation Projet

## Les besoins :

Au cours du dernier siècle, les technologies liées au numérique n’ont cessé de se s’implanté dans la conception de l’agriculture (surveillance GPS, analyse en temps réel des besoins des plante, distribution d’eau automatisé, etc.…), cependant ces technologies sont destinées à des professionnels et ne sont pas adapté à l’usage quotidien d’un particulier.

## Problématique :

Comment pourrons nous apporter une solution simple et efficace pour surveiller l'environnement d'une ou plusieurs plantes ?

## Axe de réflexion :

Pour répondre à cette question, nous avons travaillé sur trois solutions.

* Solution 1 : crée un module qui récupère les données de la plante et les affiche en temps réel
* Solution 2 : récupérer les valeurs de la plante et les stocker sur un périphérique de stockage
* Solution 3 : récupérer les valeurs de la plante, et les afficher sur un logiciel client.

Au vu du besoin énoncé dans la problématique, nous avons décider réaliser le projet numéro 3.

# Cahier des charges

* Posséder une base de données comportant les caractéristiques suivantes :
  + Le nom de la plante
  + Une Catégorisation de la plante
  + La Description
  + Une ou plusieurs Photos
  + L’humidité optimale du sol,
  + La température atmosphérique optimale
  + La luminosité optimale
  + La période de floraison
* Un module pouvant récupérer les informations suivantes :
  + La luminosité ambiante
  + L’humidité atmosphérique
  + La température atmosphérique
  + L’humidité du sol
* Un application cliente qui soit simple d’utilisation pouvant récupérer les informations du module, et les comparé à la base de données.

# Description Fonctionnelle

## Architecture de la solution :

Notre produit se présente de la manière ci-contre :

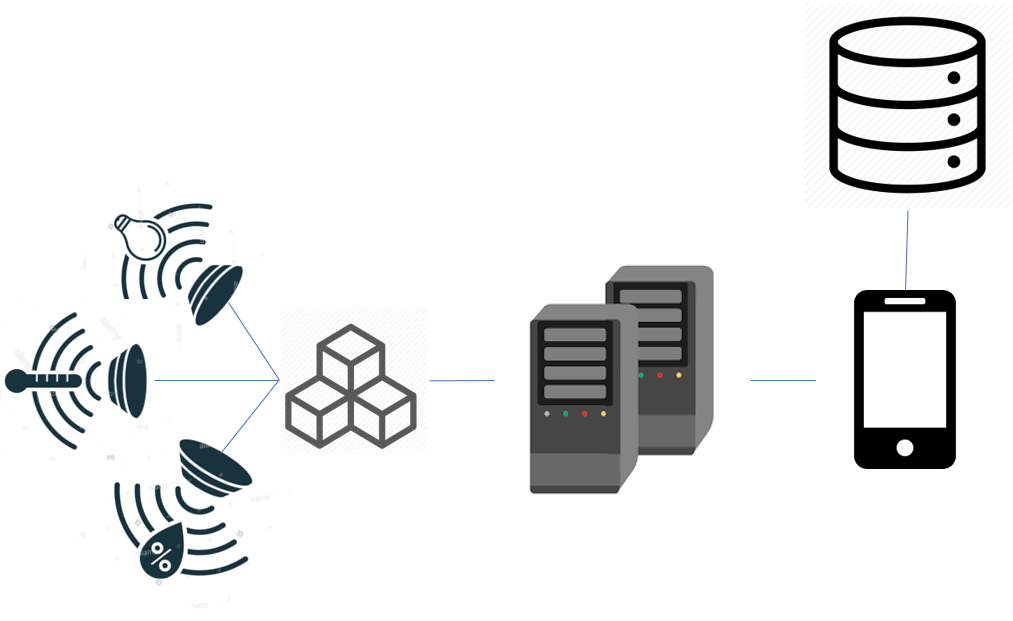


Fig 6

Fig 7

Fig 5

Fig 4

Fig 2

Fig 1

Fig 3

Notre module (*Fig. 4)* récupère les informations de trois capteur, capteur de température et d’humidité atmosphérique (*Fig. 1*), capteur de luminosité (*Fig. 2*) et capteur d’humidité du sol (*Fig. 3*), qu’il envoie alors au serveur (*Fig. 5*).

Le client (*Fig. 6*) récupère alors les données du serveur et les affiche. L’utilisateur peut alors choisir une plante de référence pour comparer les données. Une fois la plante de référence choisi, le client peut alors afficher les données de référence contenue dans la base de données en local (*Fig. 7)*. L’utilisateur pourra alors comparer les valeurs de référence aux valeurs renvoyer par le module.

## Choix du matériel :

Notre projet a besoin pour fonctionner des caractéristiques suivantes :

* Un capteur d’humidité du sol
* Un capteur de luminosité
* Un capteur de température atmosphérique
* Un capteur de d’humidité atmosphérique

Notre module devant être vendu a des particulier, nous devons être compétitif au niveau du prix, tout en ayant un des capteurs pouvant supporter un climat continental européen. Pour cela nous avons fait un tableau comparatif des différents capteurs dont nous avons besoin.

|  |  |  |  |
| --- | --- | --- | --- |
| Capteur d’humidité du sol | | | |
| Device | Consommation | Plage d’utilisation | prix |
| Grove 101020008 | 5V | 0 🡪 950 | 3.80€ |
| SEN0193 | 5V | 260 🡪 520 | 8.18€ |
| Octopus EF04027 | 5V | 0 🡪 255 | 2.95€ |

|  |  |  |  |
| --- | --- | --- | --- |
| Capteur de température | | | |
| Device | Consommation | Plage d’utilisation | prix |
| DHT11 | 5V | O 🡪 50°C | 4.95€ |
| DHT22 | 6V | -40 🡪 80°C | 10.20€ |
| Grove 101020015 | 5V | -40 🡪 125°C | 3.05€ |

|  |  |  |  |
| --- | --- | --- | --- |
| Capteur de d’humidité | | | |
| Device | Consommation | Plage d’utilisation | prix |
| DHT11 | 5V | 20 🡪 100% | 4.95€ |
| DHT22 | 6V | 0 🡪 100% | 10.20€ |
| EF10013 | 5V | 0 🡪 99% | 12.80€ |

|  |  |  |  |
| --- | --- | --- | --- |
| Capteur de luminosité | | | |
| Device | Consommation | Plage d’utilisation | prix |
| Spark fun APDS 9301 | 3V | 0 🡪 65535 | 5.95€ |
| VEML7700 | 5V | 0 🡪 120 lux | 5.94€ |
| TSL2561 | 3V | 0.1 🡪 40000 lux | 9.54€ |

Les capteurs choisis sont les capteurs les plus adapté à une utilisation occidentale. Leur utilisation simplifiée ainsi que leur nombre peu important permettront un suivit de l’application cliente plus accessible.

## Méthode de fonctionnement :

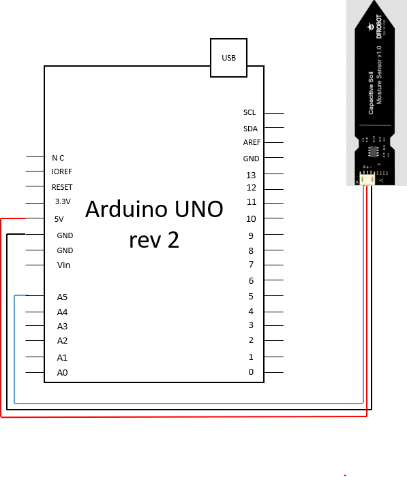
Le fonctionnement de notre module peut être réduis à trois étapes :

1. Collecte des données
2. Communication des données
3. Affichage des données

### Collecte des données :

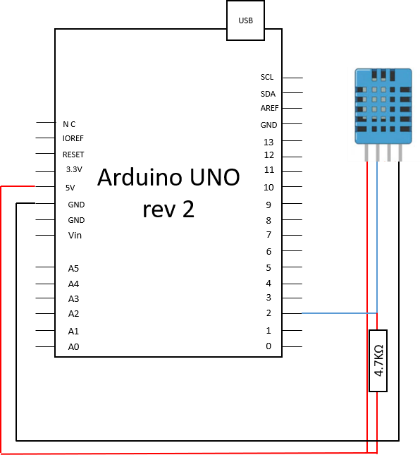
La collecte des données des capteur s’effectue grâce à une carte micro contrôlé « Arduino ». Sur cette carte son branché les capteurs choisis plus haut comme suit :

* Le capteur d’humidité du sol :



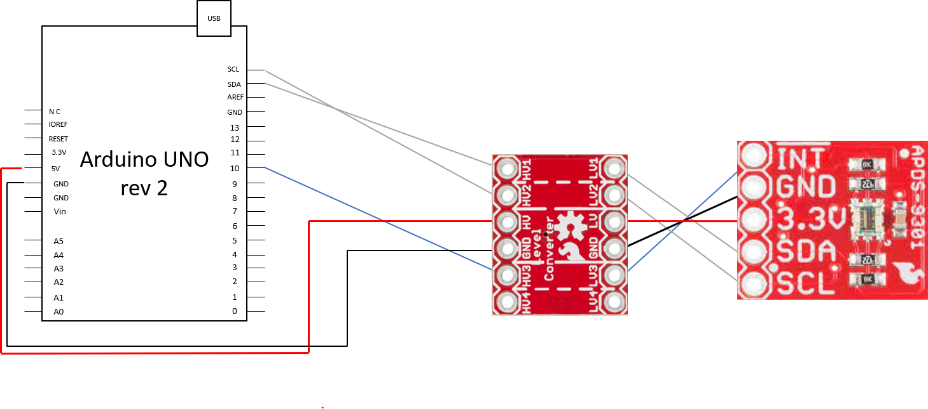
Ce capteur nécessite un calibrage avant utilisation pour récupérer un pourcentage précis, cependant il est commun de retrouver dans sa documentation technique qu’il a pour minimal 260 (un enivrement sec) et pour maximal 520 (un environnement liquide).

* Le capteur de température, humidité :



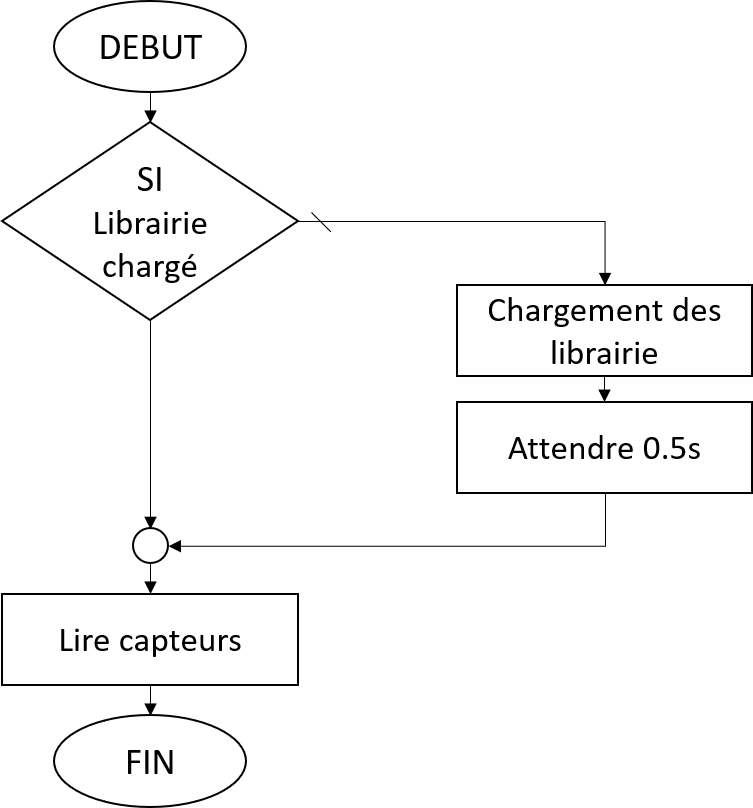
Le choix de ce capteur était judicieux, grâce à une librairie simple d’utilisation, nous avons pu maximiser l’efficacité du module avec la mesure de deux valeurs, la température et la luminosité. Ce capteur est un gain de place, de temps, d’argent, il est parfaitement adapté à ce module qui a pour but de satisfaire un public de particulier.

* Le capteur de luminosité :



Le capteur d’intensité lumineuse requière l’ajout d’un modulateur de tension afin de passer d’une tentions en entrée de 5V a une tension en sortie de 3.3V pour que le capteur puisse fonctionner correctement.

Principe de fonctionnement :



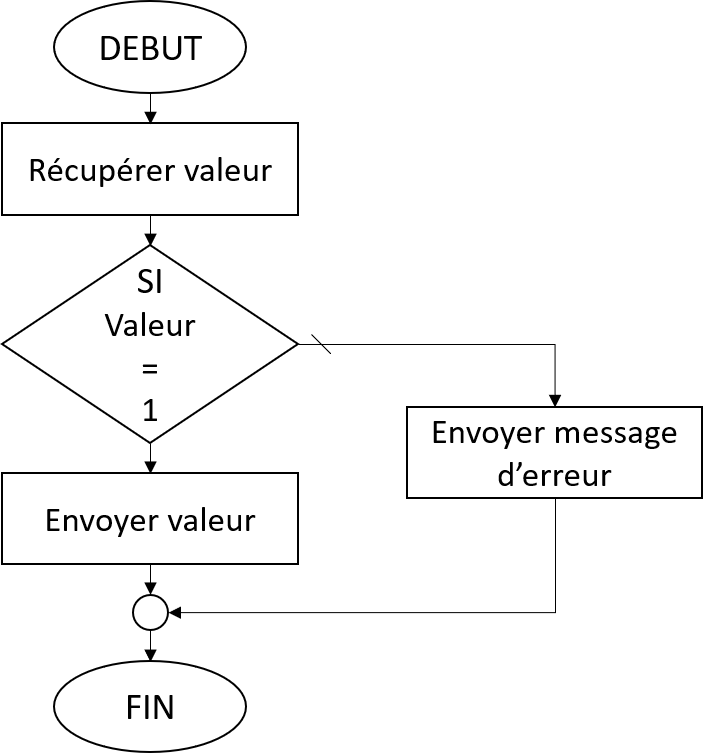
Synthèse :

Ces trois capteurs récupèrent les valeurs en continue d’es lors que le module sera connecté. Au vu du nombre de câble qui constitue le module, nous conseillons fortement la création et l’utilisation de circuit imprimer. Ils auront l’avantage de prendre moins de place et de générer moins d’erreur de faux contact.

### Communication des données :

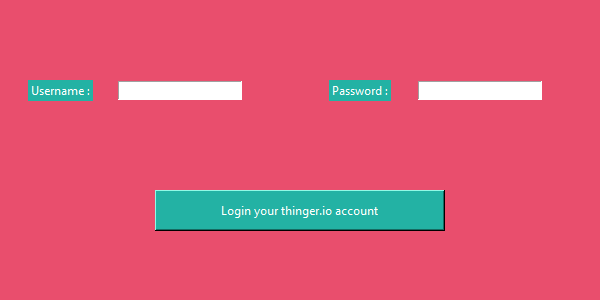
Pour l’échange de donné entre la carte nous avons fait le choix d’utiliser « thinger.io ». « Thinger.io » est une infrastructure cloud qui permet de connecter des périphériques à internet. Cependant si le produit se démocratise comme escompté nous conseillons l’acquisition d’un serveur dédié a ce module pour plus de stabilité et d’option d’évolution.

Principe de fonctionnement :

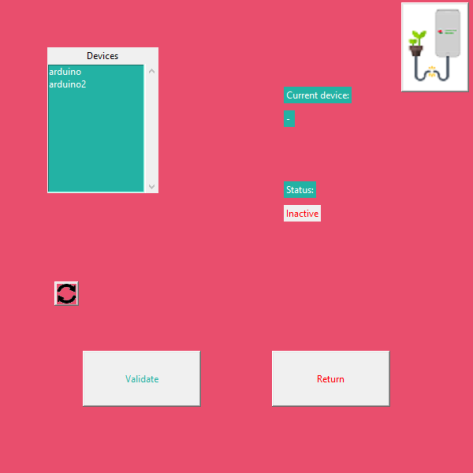


### Affichage des données :

Lorsque l’utilisateur lance le client, il doit tout d’abord s’authentifier afin d’évité la fuite d’information personnel.



Après s’être identifié celui-ci devra sélectionner le module qu’il souhaite consulter, chaque module aura sa propre ID (nommé « arduino1 » et « arduino2 » dans l’exemple ci-dessous).

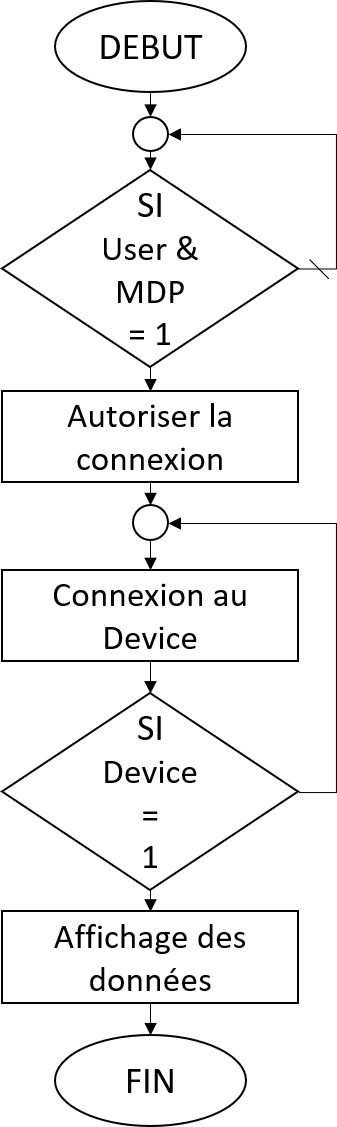


Enfin l’utilisateur pourra sélectionner une plante de référence pour comparer ses valeurs avec les valeurs optimal du plant.



Enfin a la fin de son utilisation, l’usager pourra se déconnecter du client.

Principe de fonctionnement



# Bilan projet

## Problème rencontré :

Lors de la réalisation de de ce projet, nous avons fait face à plusieurs difficultés :

* L’instabilité du matériel : la fait de ne pas posséder le matériel nous empêchais de le manipulé de manière précise. A savoir la création de carte électronique pour minimiser les risques de faux contact et de perte d’information. A cour terme cela nous aurais grandement simplifiée la tache en évitant réglé des problèmes d’électronique et de nous concentrer sur le développement. A long terme cela aurais permis une plus grande stabilité et donc une croissance plus rapide du projet (entrainant pas la même occasion l’ajout de nouvelle fonctionnalité)
* Un mauvais choix de composant : le fait d’avoir choisis la solution « thinger.io » nous a limitée dans notre démarche. Ce serveur étant utiliser pour un grand nombre de personne dans le monde, nous étions limités a un certain débit de requête ralentissant grandement notre vitesse de travail.

## Ouverture :

La présentation d’une maquette fonctionnelle n’est que l’effleurement du potentiel de ce projet. Voici quelque axe d’amélioration pour des mise à jour futur (cette liste est non exhaustive).

* Autonomie : le module dois aujourd’hui est relié à un secteur pour être alimenté en électricité. Avec l’ajout d’une batterie nous pourrions rendre le module plus accessible et plus adapté à son environnement.
* Gérer la base de données coté serveur. L’ajout d’une base de données coté serveur permettrais la libération de données sur l’application. Ainsi l’utilisateur aurait à sa disposition une base de données complète, détaillé et mise à jour sur son client.

## 