

Ministère de l'Enseignement Supérieur de la Recherche et de l'Innovation

**Université Assane SECK de Ziguinchor**  
**UFR Sciences et Technologies**  
**Département Informatique**



**Mémoire de fin d'études**

Pour l'obtention du diplôme de Master

Mention : informatique

Spécialité : Génie Logiciel

**Sujet :**

**Conception d'un système d'irrigation automatique**

Présenté par : Cheikh Talla KOUNDOUL

Soutenance le 07/07/2020

Sous la direction de Dr Youssou FAYE

Sous la supervision du Pr. Salomon SAMBOU

**Membres du jury**

Salomon SAMBOU	Professeur	Président de jury	UASZ
Youssou FAYE	Maître de Conférence Titulaire	Encadreur	UASZ
Ousmane DIALLO	Maître de Conférence Titulaire	Rapporteur	UASZ
Mouhamadou GAYE	Maître de Conférence	Rapporteur	UASZ
Ismaila DIALLO	Ingénieur	Maître de stage	ZIP

2019 - 2020

## RÉSUMÉ

L'irrigation est une technique de plus en plus utilisée à travers le monde dans le but d'accroître la quantité et la qualité des récoltes. Le Sénégal ne fait pas exception à cette règle d'autant plus qu'il est soumis pendant sept ou huit mois de l'année à des conditions d'aridité particulièrement sévère. Nous trouvons alors plusieurs techniques d'irrigation utilisées dans le pays dépendant principalement de la disponibilité de l'eau et des moyens. Les plus utilisées sont cependant traditionnelles comme l'irrigation par arrosoir, par submersion ou encore l'irrigation à la raie. En plus de nécessiter un travail physique pénible, ces dernières gaspillent de l'eau et dégradent le sol entraînant ainsi une baisse de la productivité agricole.

L'objectif de ce projet est de proposer un système d'irrigation automatique afin de pallier ces problèmes et renforcer durablement la production agricole. La problématique est donc la suivante : comment faire en sorte d'utiliser au mieux l'eau, en même temps que les terres et les ressources humaines de façon à augmenter la productivité des cultures irriguées et ainsi participer à la sécurité alimentaire du pays ?

Afin de répondre à la problématique, nous avons effectué une étude sur les différents problèmes rencontrés avec les techniques d'irrigation traditionnelles avant de proposer une solution. Cette solution a été concrétisée à travers la conception d'un système appelé **AgroDigital** composé :

- ◆ d'un dispositif pour arroser automatiquement les champs en fonction de leurs besoins en eau mesurée par des capteurs. En cas de problème technique, ce dispositif envoie un SMS (Short Message Service) d'alerte,
- ◆ d'un tableau de bord disponible en ligne pour interagir et suivre le dispositif à temps réel. À travers ce tableau de bord, on peut déclencher ou arrêter manuellement l'arrosage en envoyant des commandes au dispositif.

Le dispositif est basé sur la carte électronique Arduino Mega programmée en une variante du langage C++. Quant à la création du site web (le tableau de bord), nous avons utilisé la technologie Firebase de Google pour la partie **Back end** et le framework Angular2 pour la partie **Front end**. Nous nous sommes aidés des modèles de cycle de vie d'un logiciel et du langage de modélisation **UML** pour définir les différentes phases et modèles de conception du système après une définition claire des besoins.

## ABSTRACT

Irrigation is a technique increasingly used around the world to increase the quantity and quality of crops. Senegal is no exception to this rule, especially since it is subject to particularly severe arid conditions for seven or eight months of the year. We then find several irrigation techniques used in the country depending mainly on the availability of water and means. However, the most commonly used are traditional, such as sprinkler irrigation, submersion, or skate irrigation. In addition to requiring heavy physical labor, these wastewaters and degrade the soil, thus reducing agricultural productivity.

The objective of this project is to propose an automatic irrigation system to alleviate these problems and sustainably increase agricultural production. The problem is therefore the following: how to make the best use of water, along with land and human resources to increase the productivity of irrigated crops and thus contribute to the country's food security?

To respond to the problem, we have carried out a study on the various problems encountered with traditional irrigation techniques before proposing a solution. This solution was concretized through the design of a system called **AgroDigital** composed:

- ◆ a device to automatically water the fields according to their water needs measured by sensors. In the event of a technical problem, this device sends an SMS (Short Message Service) alert,
- ◆ a dashboard available online to interact and track the device in real-time. Through this dashboard, you can manually start or stop watering by sending commands to the device.

The device is based on the Arduino Mega electronic board programmed in a variant of the C++ language. As for the creation of the website (the dashboard), we used Google's Firebase technology for the back end part and the Angular2 framework for the front end part. We used software lifecycle models and UML modeling language to define the different phases and design models of the system after a clear definition of the needs.

## **DÉDICACES**

*Je dédie ce modeste mémoire de fin d'étude à mes chers parents qui m'ont tout donné, m'ont toujours soutenu et n'ont jamais ménagés aucun effort pour ma réussite.*

*Un merci particulier à ma mère **Soda DIACK** qui ne cesse de se sacrifier pour moi, qu'Allah lui octroie une longue vie remplie de bonheur et de santé.*

*J'en profite aussi pour rendre hommage à mon père **Cheikh Demba KOUNDOUL**, que la terre lui soit légère, qu'Allah l'accueille au paradis **Firdaws**.*

*Je dédie aussi ce travail à toute ma famille, à mon oncle, à mes frères et sœurs, à mes amis et aussi à mes camarades.*

## **REMERCIEMENTS**

### **A Allah le Miséricordieux le Tout Miséricordieux, qui par Sa Grâce et Sa Miséricorde nous a donné la santé de mener à bien ce travail.**

Avant d'aller plus loin, je tiens à remercier toutes les personnes qui ont participé de près ou de loin à la réussite de ce mémoire.

Je commence par mon encadrant Mr Youssou FAYE pour son implication et sa disponibilité.

Je remercie mon maître de stage Mr Ismaïla DIALLO ainsi que Mr Baba Issa CAMARA, initiateur du projet, pour le temps qu'ils m'ont accordé.

Je remercie aussi le président du jury d'avoir bien voulu superviser et présider cette soutenance de mémoire ainsi que les rapporteurs d'avoir accepté de parcourir ce document.

Un grand merci également à tout le personnel de Ziguinchor Institut Polytechnique pour l'accès sans limites ni conditions au laboratoire informatique qu'ils m'ont accordé, pour leur aide et encouragement.

À tous les enseignants du département Informatique, je dis merci pour toutes les connaissances théoriques et pratiques et l'accompagnement qu'ils m'ont apportés tout au long de mon cursus universitaire.

Je n'oublie pas mes camarades de promotion pour ces années passées ensemble dans la joie et la bonne humeur.

## Table des matières

LISTE DES FIGURES.....	X
LISTE DES TABLEAUX.....	XII
LISTE DES ABRÉVIATIONS.....	XIII
INTRODUCTION GÉNÉRALE.....	1
PREMIÈRE PARTIE : CONTEXTE DU SUJET ET ÉTUDE DE L'EXISTANT.....	3
CHAPITRE I : DESCRIPTION ET CONTEXTE DU SUJET.....	3
I - Présentation de Ziguinchor Institut Polytechnique.....	3
II - Le contexte justificatif du projet.....	3
II.1 - L'origine du projet.....	3
II.2 - Techniques d'irrigation utilisées au Sénégal.....	3
II.2.a - Les techniques d'irrigation traditionnelles.....	4
II.2.b - Les techniques d'irrigation modernes.....	5
II.3 - Problèmes liés aux techniques d'irrigation traditionnelles.....	5
II.3.a - L'utilisation de l'eau mal maîtrisée.....	5
II.3.b - La salinisation des sols.....	5
II.3.c - Efforts physiques considérables.....	6
II.3.d - Baisse de la productivité.....	6
III - Objectif du projet.....	6
III.1 - Objectif général du projet.....	6
III.2 - Objectifs spécifiques du projet.....	6
IV - Travail demandé.....	6
V - Solution proposée.....	7
CHAPITRE II : PRÉSENTATION DES DIFFÉRENTS SYSTÈMES D'IRRIGATION.....	8
I - Qu'est-ce que l'irrigation ?.....	8
II - Qu'est-ce qu'un système d'irrigation automatique ?.....	8
III - Les différents systèmes d'irrigation.....	8
III.1 - L'irrigation gravitaire.....	9
III.1.a - Arrosage par ruissellement (par planches ou par calant).....	9
III.1.b - Arrosage par submersion (ou inondation).....	9
III.1.c - Arrosage par infiltration (à la raie).....	10
III.1.d - Arrosage par cuvette (ou bassin).....	10
III.2 - L'irrigation par goutte à goutte.....	11
III.2.a - L'unité de tête (contrôle/commande) :.....	11
III.2.b - Les distributeurs.....	11
III.3 - L'irrigation par aspersion.....	12
IV - Comparaison entre les différents systèmes d'irrigation.....	13

DEUXIÈME PARTIE : CYCLE DE DÉVELOPPEMENT D'AGRODIGITAL ET ANALYSE DES BESOINS FONCTIONNELS.....	15
CHAPITRE III : CYCLE DE DÉVELOPPEMENT D'AGRODIGITAL.....	15
I - Pourquoi un modèle de cycle de vie d'un logiciel ?.....	15
II - Qu'est-ce que le cycle de vie d'un logiciel ?.....	15
III - Présentation de quelques modèles de cycle de vie traditionnels.....	16
III.1 - Modèle en cascade.....	16
III.2 - Modèle en V.....	17
III.3 - Modèle en spirale.....	17
IV - Présentation de quelques modèles agiles.....	18
IV.1 - ASD (Adaptative Software Development).....	18
IV.2 - RAD (Rapid Application Development).....	19
IV.3 - UP (Unified Process).....	19
IV.4 - Scrum.....	20
IV.5 - Synthèse des différences fondamentales entre approche traditionnelle et approche agile.....	22
IV.6 - Notre équipe projet.....	23
CHAPITRE IV : SPÉCIFICATION ET ANALYSE DES BESOINS FONCTIONNELS.....	24
I - Qu'est-ce que l'UML ?.....	24
II - Spécification et analyse des besoins de l'application avec Arduino.....	25
II.1 - Spécification des besoins.....	25
II.1.a - Identification des acteurs.....	25
II.1.b - Identification des fonctionnalités.....	25
II.1.c - Description de quelques cas d'utilisation.....	26
II.1.c.1 - Description du cas « Envoyer les données sur internet ».....	26
II.1.c.2 - Description du cas « Envoyer le nouvel état du moteur ».....	26
II.1.c.3 - Description du cas « Exécuter la nouvelle commande ».....	27
II.2 - Analyse des besoins fonctionnels.....	28
II.2.a - Analyse du cas « Envoyer les données sur internet ».....	28
II.2.a.1 - Diagramme de séquence.....	28
II.2.a.2 - Diagramme d'activité.....	29
II.2.b - Analyse du cas « Envoyer le nouvel état du moteur ».....	29
II.2.b.1 - Diagramme de séquence.....	29
II.2.b.2 - Diagramme d'activité.....	30
III - Spécification et analyse des besoins du site web.....	31
III.1 - Spécification des besoins.....	31
III.1.a - Identification des acteurs.....	31
III.1.b - Identification des fonctionnalités.....	32
III.1.c - Le Backlog du produit.....	32

III.1.d - Planification des sprints.....	34
III.1.e - Description textuelle de quelques cas d'utilisation.....	34
III.1.e.1 - Description textuelle du cas d'utilisation « S'inscrire ».....	34
III.1.e.2 - Description textuelle du cas d'utilisation «Se connecter ».....	35
III.1.e.3 - Description textuelle du cas d'utilisation « Confirmation d'inscription ».....	36
III.1.e.4 - Description textuelle du cas d'utilisation « exporter les données ».....	36
III.1.e.5 - Description textuelle du cas d'utilisation « Modifier son mot de passe ».....	37
III.1.f - Le diagramme de cas d'utilisation.....	38
III.2 - Analyse des besoins fonctionnels.....	38
III.2.a - Analyse du cas « S'inscrire ».....	39
III.2.a.1 - Diagramme de séquence.....	39
III.2.a.2 - Diagramme d'activité.....	39
III.2.b - Analyse du cas « Se connecter ».....	40
III.2.b.1 - Diagramme de séquence.....	40
III.2.b.2 - Diagramme d'activité.....	41
III.2.c - Analyse du cas « Exporter ses données ».....	41
III.2.c.1 - Diagramme de séquence.....	41
III.2.c.2 - Diagramme d'activité.....	42
III.2.d - Analyse du cas « Modifier son mot de passe ».....	43
III.2.d.1 - Diagramme de séquence.....	43
III.2.d.2 - Diagramme d'activité.....	44
TROISIÈME PARTIE : CONCEPTION, IMPLÉMENTATION ET PRÉSENTATION D'AGRODIGITAL.....	45
CHAPITRE V : ARCHITECTURE ET CONCEPTION D'AGRODIGITAL.....	45
I - La conception générale.....	45
I.1 - Architecture générale du système de déploiement.....	45
I.2 - Schéma récapitulatif du système.....	46
I.3 - Architecture d'AgroDigital.....	47
I.3.a - Architecture de l'application avec Arduino.....	47
I.3.b - Architecture du site web.....	48
I.3.b.1 - Qu'est-ce que le modèle d'architecture MVC ?.....	48
I.3.b.2 - Architecture du système du site web.....	50
I.4 - Diagramme de paquetage du site web.....	50
I.5 - Diagramme de déploiement du système.....	51
II - La conception détaillée du site web.....	52
II.1 - Présentation des classes.....	52
II.2 - Le diagramme de classe.....	53
CHAPITRE VI : IMPLÉMENTATION ET PRÉSENTATION D'AGRODIGITAL.....	54
I - Les outils mis à ma disposition.....	54



II - Les tâches périphériques.....	54
II.1 - Concernant la carte Arduino.....	54
II.2 - Concernant le site Web.....	54
III - Difficultés rencontrées et solutions apportées.....	55
IV - Compétences acquises.....	55
V - Présentation de l'API Firebase.....	55
V.1 - Tout d'abord pourquoi Firebase ?.....	55
V.2 - Présentation de Firebase.....	56
V.3 - Cloud Firestore.....	56
V.4 - Cloud Functions.....	57
V.5 - Authentification (Firebase Auth).....	57
V.6 - Hosting.....	57
VI - Implémentation et présentation de l'application avec Arduino.....	57
VI.1 - Matériels utilisés.....	57
VI.1.a - La carte électronique utilisée.....	57
VI.1.b - La carte Arduino Méga.....	58
VI.1.c - Le module SIM800L.....	58
VI.1.d - Les différents capteurs utilisés.....	61
VI.1.d.1 - Le capteur de température et d'humidité de l'air.....	61
VI.1.d.2 - Le capteur d'humidité du sol.....	61
VI.1.e - Composants utilisés pour tester le système.....	62
VI.1.e.1 - La mini pompe à eau.....	62
VI.1.e.2 - Le tuyau d'arrosage.....	63
VI.1.e.3 - Le relais.....	63
VI.2 - Le montage de l'application.....	64
VI.3 - Librairies utilisées.....	64
VI.3.a.1 - Arduino SIM800L library.....	64
VI.3.a.2 - ArduinoJSON.....	64
VI.3.a.3 - Adafruit DHT Humidity & Temperature Sensor Library.....	64
VI.4 - Outils utilisés.....	65
VI.4.a.1 - Arduino IDE.....	65
VI.4.a.2 - Fritzing.....	65
VI.5 - Présentation de l'application Arduino.....	65
VII - Implémentation et présentation de la plate-forme web.....	66
VII.1 - Technologies utilisées.....	66
VII.1.a - Le framework Angular.....	66
VII.1.b - TypeScript.....	67
VII.1.c - JQuery.....	67

VII.1.d - Bootstrap.....	67
VII.1.e - Font Awesome.....	67
VII.1.f - ChartJS.....	67
VII.1.g - Ng Date Time Picker.....	67
VII.2 - Outils utilisés.....	68
VII.2.a - Visual Paradigm Community Edition.....	68
VII.2.b - Visual Studio Code.....	68
VII.3 - Arborescence des paquetages et fichiers de l'application web.....	68
VII.4 - Présentation de l'application web.....	71
VII.4.a - La page d'inscription.....	72
VII.4.b - La page d'authentification.....	73
VII.4.c - La page d'accueil.....	73
VII.4.d - La page de la température ambiante.....	74
VII.4.e - La page d'historique des arrosages.....	75
VII.4.f - La page d'exportation des données.....	75
VII.4.g - La page d'administration.....	76
VII.4.h - Activer/Désactiver un compte utilisateur.....	76
CONCLUSION GÉNÉRALE.....	77
ANNEXE.....	78
I - Le code côté serveur pour Arduino.....	78
II - Exemple de deux méthodes du paquetage service pour obtenir les données mesurées par les capteurs .....	78
III - Extrait du contenu de quelques fichiers de configuration et de dépendance.....	79
III.1 - Contenu du fichier de dépendance package.json.....	79
III.2 - Contenu du fichier de configuration app-routing.module.ts.....	80
III.3 - Contenu du fichier de configuration app.module.ts.....	81
BIBLIOGRAPHIE ET WEBOGRAPHIE.....	82

## LISTE DES FIGURES

Figure 1: Arrosage par ruissellement[31].....	9
Figure 2: Arrosage par submersion[32].....	9
Figure 3: Arrosage par infiltration[31].....	10
Figure 4: Arrosage par cuvette[31].....	10
Figure 5: Exemple de goutteur[33].....	11
Figure 6: Exemple de tuyau à parois poreuses[34].....	12
Figure 7: Exemple d'asperseur avec canon et tourniquet[15].....	13
Figure 8: Exemple d'asperseur avec Une rampe oscillante[15].....	13
Figure 9: Modèle en cascade[18].....	16
Figure 10: Modèle en V[18].....	17
Figure 11: Modèle en spirale[19].....	17
Figure 12: Modèle Adaptative Software Development[20].....	18
Figure 13: Les quatre phase du modèle Unified Process[20].....	20
Figure 14: Illustration du processus de Scrum[20].....	21
Figure 15: Diagramme de séquence du cas « Envoyer les données sur internet ».....	28
Figure 16: Diagramme d'activité du cas « Envoyer les données sur internet ».....	29
Figure 17: Diagramme de séquence du cas « Envoyer le nouvel état du moteur ».....	30
Figure 18: Diagramme d'activité du cas « Envoyer le nouvel état du moteur ».....	31
Figure 19: Diagramme de cas d'utilisation du site web.....	38
Figure 20: Diagramme de séquence du cas « S'inscrire ».....	39
Figure 21: Diagramme d'activité du cas « S'inscrire ».....	40
Figure 22: Diagramme de séquence du cas «Se connecter».....	40
Figure 23: Diagramme d'activité du cas «Se connecter».....	41
Figure 24: Diagramme de séquence du cas «Se connecter».....	42
Figure 25: Diagramme d'activité du cas «Se connecter».....	43
Figure 26: Diagramme de séquence du cas « Modifier son mot de passe ».....	43
Figure 27: Diagramme d'activité du cas « Modifier son mot de passe ».....	44
Figure 28: Architecture générale de déploiement d'AgroDigital.....	46
Figure 29: Schéma récapitulatif du système.....	47
Figure 30: Architecture de l'application avec Arduino.....	48
Figure 31: Exemple du modèle d'architecture MVC[23].....	49
Figure 32: Architecture du système du site web.....	50
Figure 33: Diagramme de paquetage du site web.....	51
Figure 34: Diagramme de déploiement du site web.....	52
Figure 35: Présentation des interfaces.....	52
Figure 36: Diagramme de classe du site web.....	53

Figure 37: Exemple de collection de Firestore.....	56
Figure 38: La carte Arduino Mega[26].....	58
Figure 39: Kit complet du module SIM800L.....	59
Figure 40: Face interne du module SIM800L.....	59
Figure 41: Antenne interne.....	60
Figure 42: Antenne externe.....	60
Figure 43: Branchement de l'antenne externe.....	60
Figure 44: Le capteur DHT11.....	61
Figure 45: Capteur d'humidité.....	61
Figure 46: La pompe à eau.....	62
Figure 47: Tuyau d'arrosage.....	63
Figure 48: Module relais 5V.....	63
Figure 49: Montage du dispositif.....	64
Figure 50: Le dispositif.....	65
Figure 51: Angular framework[29].....	66
Figure 52: Arborescence des paquetages et fichiers de l'application.....	69
Figure 53: Détails d'un composant.....	70
Figure 54: Présentation de paquetages.....	71
Figure 55: Arborescence du site.....	72
Figure 56: La page d'inscription.....	72
Figure 57: La page de connexion.....	73
Figure 58: La page d'accueil.....	74
Figure 59: La page de la température ambiante.....	74
Figure 60: La page d'historique des arrosages.....	75
Figure 61: L'exportation des données.....	75
Figure 62: La page d'administration.....	76
Figure 63: Activer ou Désactiver un compte utilisateur.....	76
Figure 64: Fonction qui enregistre les données mesurées par les capteurs.....	78
Figure 65: Exemple de méthodes d'accès aux données.....	79
Figure 66: L'interface Mesure.....	79
Figure 67: Le fichier package.json.....	80
Figure 68: Le fichier app-routing.module.ts.....	80
Figure 69: Le fichier app.module.ts.....	81

## LISTE DES TABLEAUX

Tableau 1: Tableau de comparaison des différents systèmes d'irrigation[16].....	14
Tableau 2: Synthèse des différences fondamentales entre approche traditionnelle et approche agile[20].....	23
Tableau 3: Notre équipe projet.....	23
Tableau 4: Identification des acteurs pour Arduino.....	25
Tableau 5: Identification des fonctionnalités.....	25
Tableau 6: Description du cas « Envoyer les données sur internet ».....	26
Tableau 7: Description du cas « Envoyer le nouvel état du moteur ».....	27
Tableau 8: Description du cas « Exécuter la nouvelle commande ».....	27
Tableau 9: Identification des acteurs du site web.....	32
Tableau 10: Identification des fonctionnalités du site web.....	32
Tableau 11: Le Backlog du produit.....	33
Tableau 12: Les sprints du produit.....	34
Tableau 13: Description textuelle du cas d'utilisation « S'inscrire ».....	35
Tableau 14: Description textuelle du cas d'utilisation « Se connecter ».....	36
Tableau 15: Description textuelle du cas d'utilisation « Confirmation d'inscription ».....	36
Tableau 16: Description textuelle du cas d'utilisation « exporter les données ».....	37
Tableau 17: Description textuelle du cas d'utilisation « Modifier son mot de passe ».....	37

## LISTE DES ABRÉVIATIONS

API :	Application Programming Interface
CDN :	Content Delivery Network
CNTAGR :	Centre National de formation des Techniciens en Agriculture et en Génie Rural de Ziguinchor
CSS:	Cascading Style Sheets
CSV :	Comma-Separated Values
C++ :	langage de programmation compilé permettant la programmation sous de multiples paradigmes
EXCEL :	logiciel de la suite bureautique Office de Microsoft et permet la création de tableaux, de calculs automatisés, de plannings, de graphiques et de bases de données.
FM :	une bande d'émissions qui utilisent la modulation de fréquence pour la voie monophonique et une sous-porteuse pour la transmission du son...
Framework :	ensemble d'outils disponibles pour développer mieux et plus vite. .
GPRS :	Global Packet Radio Service
GSM :	Global System Mobiles
HTML:	Extensible Hypertext Markup Language
HTTP:	HyperText Transfer Protocol
IDE:	Environnement de Développement Intégré
JavaScript:	langage de script orienté objet principalement utilisé dans les pages HTML exécutés du côté client.
JSON :	JavaScript Object Notation
LESS :	langage dynamique de génération de CSS
MVC:	Model View Controller
pH :	Potentiel Hydrogène
SASS :	Syntactically Awesome Stylesheets
SDE :	Sénégalaise Des Eaux
SDK :	Software Development Kit
SIM :	Subscriber Identity Module
SMS:	Short Message Service
SSL:	Secure Socket Layer
TP :	travaux pratiques
UASZ :	Université Assane SECK de Ziguinchor
UML :	Langage de Modélisation Unifié
ZIP :	Ziguinchor Institut Polytechnique

# INTRODUCTION GÉNÉRALE

Face à l'accroissement démographique et aux contraintes climatiques, l'irrigation est apparue comme une clef du développement au Sénégal où la principale activité économique est l'agriculture. Nombreux sont ceux qui pratiquent l'agriculture irriguée aujourd'hui que ce soit au niveau de leur champ, de la cour de leur maison ou de leur terrasse, en milieu rural comme urbain. Cependant, les techniques d'irrigation utilisées restent pour la plupart traditionnelles dont l'efficacité laisse à désirer. Ces dernières sont le plus souvent onéreuses en eau et en efforts physiques en plus de précipiter la dégradation du sol avec la remontée de la langue salée.

Or l'eau est une ressource naturelle précieuse dont les besoins en approvisionnement domestique, en industrie et surtout en agriculture ne cessent de croître. L'agriculture irriguée reste en effet le secteur qui consomme le plus d'eau au Sénégal. Les prélèvements sur les ressources en eau ont été estimés en 2013 à 92,98 % pour l'agriculture, 4,41 % pour l'usage domestique et 2,61 % pour l'industrie[1]. A cela s'ajoute les précipitations, en forte baisse depuis plusieurs années (200 mm en 30 ans) qui décroissent de plus de 1 100 mm/an dans le sud à 300 mm/an dans le Nord du pays[2].

Trouver une solution à ces problèmes semble urgent et passe forcément par une maîtrise du choix des systèmes d'irrigations utilisés. Le sujet de mon stage s'oriente justement dans ce sens. Ce projet dénommé **AgroDigital** qui signifie agriculture digital ou encore l'agriculture intelligente consiste à mettre en place un système informatique embarqué permettant de gérer de manière plus intelligente l'agriculture moderne.

Ce système se chargera d'arroser de manière automatique les végétaux en fonction de leurs besoins en eau mesurée par des capteurs. Il sera aussi possible de le déclencher ou de l'arrêter à distance. Il enverra des SMS d'alerte dès que le pH ou le niveau de l'eau d'arrosage atteint un seuil critique ou en cas de problème technique. Les données mesurées sont ensuite envoyées vers un serveur distant pour les afficher à temps réel à l'aide d'un tableau de bord.

Ces données collectées pourront être analysées pour aider à adapter notre agriculture aux changements climatiques.

C'est ainsi que nous avons pu réaliser un déploiement de notre dispositif d'irrigation composé de capteurs installés sur un jardin et d'une station de contrôle permettant de mener les actions suivantes :

- ◆ arroser seulement si le sol est sec,
- ◆ s'arrêter automatiquement si le sol devient assez humide,
- ◆ suivre et interagir avec le système à distance à l'aide d'une plate-forme web,
- ◆ notifier par SMS en cas de panne technique ou si le pH ou le niveau de l'eau d'arrosage atteint un niveau critique,
- ◆ collecter des données qui pourront être analysées afin d'adapter notre agriculture aux changements climatiques.

Ce stage a été fait à l'école d'enseignement supérieur polytechnique Ziguinchor Institut Polytechnique (ZIP) pour une durée de quatre mois.

Pour réaliser ce projet, notre travail sera structuré en trois grandes parties comportant chacune deux chapitres.

- ◆ La première partie, « Contexte du sujet et étude de l'existant », contient :
  - ✓ chapitre I : description et contexte du sujet : dans ce chapitre nous traitons le sujet dans les détails, nous identifions les problèmes et proposons une solution,
  - ✓ chapitre II : Présentation des différents systèmes d'irrigation : ce chapitre nous permet de nous familiariser avec les différents systèmes d'irrigation qui existent dans le milieu de l'agriculture.
- ◆ La deuxième partie, « Cycle de développement d'AgroDigital et analyse des besoins fonctionnels », est composée de :
  - ✓ chapitre III : Cycle de développement d'AgroDigital : dans ce chapitre nous présentons certains modèles de cycle de vie d'un logiciel puis, après les avoir comparés nous en choisissons un,
  - ✓ chapitre IV : Spécification et analyse des besoins fonctionnels : c'est dans ce chapitre que l'étude des besoins exprimés par les utilisateurs du système est faite dans les détails.
- ◆ La troisième partie, « Conception, implémentation et présentation d'AgroDigital », comporte :
  - ✓ chapitre V : Architecture et conception d'AgroDigital : dans ce chapitre nous définissons l'architecture des différentes parties qui composent notre de système avant de passer à la conception,
  - ✓ chapitre VI : Implémentation et présentation d'AgroDigital : dans ce dernier chapitre, nous implémentons notre système et nous expliquons les grands points de sa réalisation.



# **PREMIÈRE PARTIE : CONTEXTE DU SUJET ET ÉTUDE DE L'EXISTANT**

## **Sommaire**

CHAPITRE I : DESCRIPTION ET CONTEXTE DU SUJET.....	3
I - Présentation de Ziguinchor Institut Polytechnique.....	3
II - Le contexte justificatif du projet.....	3
III - Objectif du projet.....	6
IV - Travail demandé.....	6
V - Solution proposée.....	7
CHAPITRE II : PRÉSENTATION DES DIFFÉRENTS SYSTÈMES D'IRRIGATION.....	8
I - Qu'est-ce que l'irrigation ?.....	8
II - Qu'est-ce qu'un système d'irrigation automatique ?.....	8
III - Les différents systèmes d'irrigation.....	8
IV - Comparaison entre les différents systèmes d'irrigation.....	13

# CHAPITRE I : DESCRIPTION ET CONTEXTE DU SUJET

Du **03 janvier** au **03 mai 2019** j'ai fait un stage d'une durée de quatre mois au département informatique de l'école d'enseignement supérieur polytechnique Ziguinchor Institut Polytechnique (ZIP), portant sur l'implémentation d'un système d'irrigation automatique.

Dans la suite de ce chapitre, après une brève présentation de ZIP, nous allons délimiter le sujet, le contextualiser, identifier les problèmes, dégager nos objectifs et proposer une solution.

## I - Présentation de Ziguinchor Institut Polytechnique

ZIP est un établissement d'enseignement supérieur technologique et professionnel privé, reconnu par l'État et agréé par le ministère de la Formation Professionnelle, de l'Apprentissage et de l'Artisanat.

Fondée en 2015, ZIP est située au quartier de Diabir près de l'université Assane SECK de Ziguinchor. Il s'est fixé pour mission la formation des cadres techniciens, techniciens supérieurs et ingénieurs en mesure d'assumer un haut niveau d'expertise au sein des entreprises modernes.

Les principaux domaines de compétence de l'établissement sont :

- ◆ l'enseignement pluridisciplinaire au plus haut niveau dispensé à des élèves et étudiants rigoureusement sélectionnés;
- ◆ la recherche scientifique et le développement technologique en collaboration avec des organismes nationaux et internationaux, et la valorisation des travaux qui en résultent ;
- ◆ l'entrepreneuriat et la valorisation de projets innovants.

## II - Le contexte justificatif du projet

Pour contextualiser le sujet, faisons le tour des principales techniques d'irrigation pratiquées au Sénégal et de leurs limites avant de proposer une solution. Commençons pour cela par l'idée même du projet.

### II.1 - L'origine du projet

Lors d'un voyage à Bignona, département de la région de Ziguinchor, le promoteur de ZIP, Mr Baba Issa CAMARA, natif de la région de Ziguinchor et Maître de conférence à l'Université de Lorraine région de Metz en France dans le domaine des mathématiques, a visité un village où les femmes pratiquaient le maraîchage. Elles puisaient de l'eau de puits avec des seaux pour arroser leurs plantations. L'idée de ce projet, selon mon maître de stage M. Ismaïla DIALLO ingénieur informaticien et enseignant à ZIP, est née de ce voyage.

### II.2 - Techniques d'irrigation utilisées au Sénégal

Diverses techniques d'irrigation sont pratiquées au Sénégal selon la disponibilité de l'eau et des moyens utilisés. Elles sont essentiellement utilisées en horticulture. Nous avons alors deux types d'exploitants. D'une part, nous distinguons un grand nombre de petits producteurs individuels (80 %) avec des parcelles

morcelées (entre moins de 0,1 et 1 ha), avec des moyens rudimentaires et une irrigation totalement manuelle. D'autre part, nous trouvons des systèmes de production en régie directe (grandes exploitations modernes) ou indirecte (production contractualisée par un exportateur qui collecte et commercialise la production de petits exploitants regroupés ou non en association)[3].

## **II.2.a - Les techniques d'irrigation traditionnelles**

L'irrigation est pratiquée au Sénégal dans un contexte socio-économique difficile, marqué par la faiblesse des ressources financières, le coût élevé des aménagements et le manque de professionnalisme. L'irrigation par submersion (riz) et à la raie sont les techniques les plus pratiquées dans les régions de St-Louis, Matam, Ziguinchor, Kolda et Tambacounda où sont localisés les plus grandes étendues de superficies irriguées au Sénégal[4].

Au niveau des Niayes, nous pouvons noter deux méthodes d'irrigation traditionnelles qui exploitent l'eau manuellement :

- ◆ l'exploitation par les « céanes » largement dominant surtout dans la frange proche littorale où la nappe phréatique est affleurante ou à une profondeur dépassant rarement trois mètres. Ce type d'exploitation de l'eau entraîne une irrigation manuelle à l'aide d'arrosoirs et de seaux. Cependant, dans les Niayes de Mboro, contrairement aux autres régions, une amélioration qui tend vers une mécanisation de ce système est perceptible. En effet, les « céanes » ont des envergures plus larges et les profondeurs sont augmentées. Ceci a permis d'avoir suffisamment d'eau et consécutivement une utilisation de la motopompe, une augmentation des surfaces cultivées et une utilisation de lance pour l'arrosage. Compte tenu de son originalité, nous avons désigné ce système sous le terme de « céanes améliorés »[5],
- ◆ l'exploitation par les puits traditionnels est fréquente également dans les zones où la nappe est assez profonde. Elle nécessite également d'une manière générale une irrigation par arrosoir ou par seau. Ces deux types d'exploitation de l'eau, excepté le système des « céanes améliorés », ne favorisent pas l'exploitation de grandes surfaces ; et le temps consacré à l'arrosage et au stockage de l'eau dans les bassins est long[5].

Lors d'une étude sur le périmètre maraîcher de Keur Saïb Ndoye l'auteur stipule que : « Le plus répandu est le mode d'irrigation manuelle par arrosoir qui est utilisé à 100 % dans la zone. En effet seuls deux maraîchers sur les 65 de l'échantillon (soit 3 %) ont une motopompe. Mis à part ces deux producteurs tous les autres qui utilisent l'eau des puits pratiquent une exhaure manuelle à l'aide d'une poulie. Mais le constat est que la motopompe est destinée à faire sortir l'eau du puits, de la mettre dans des bassins et non pour arroser les cultures »[6].

Ceci confirme les propos du technicien agricole Monsieur Aniset Bassaine recueillis lors de notre entretien. Formateur au Centre National de formation des Techniciens en Agriculture et en Génie Rural de Ziguinchor (CNTAGR), il a eu à participer à de nombreux projets de maraîchage dans la région de Tambacounda. Il affirme que la technique d'irrigation manuelle par arrosoir était la seule utilisée dans ces projets. Il ajoute que c'est la technique la plus utilisée dans l'irrigation au Nord comme au Sud du Sénégal.

### **II.2.b - Les techniques d'irrigation modernes**

Les techniques d'irrigation modernes concernent essentiellement l'irrigation par goutte-à-goutte et l'irrigation par aspersion, qui exploitent l'eau d'une manière plus efficace. Bien que peu nombreux, ces techniques sont utilisées par certains exploitants agricoles au Sénégal. En effet, dans les Niayes, nous trouvons deux modes d'exploitation de l'eau modernisés, l'un à l'aide de forage personnel et l'autre à partir du réseau de la Sénégalaise Des Eaux (SDE). Ces deux modes d'exploitation sont plus présentes dans la région de Dakar et Thiès notamment sur les axes Sangalcam, Mboro, Sébikhotane et Pout. La taille des exploitations est supérieure à 80 hectares dans les exploitations qui disposent de forages personnels. Toutefois, ces deux systèmes d'exploitation ont entraîné de nouvelles techniques d'irrigation notamment l'irrigation par aspersion et par goutte à goutte[5].

Notons aussi que l'irrigation par goutte-à-goutte est utilisée dans certains centres de formation agricole comme CNTAGR ainsi que dans les fermes Natangué installées dans certaines régions du pays. Financées par l'Agence espagnole de Coopération, chaque ferme Natangué, d'une superficie de 1 à 2 hectares, dispose des infrastructures et équipements suivants : un puits équipé d'une pompe à énergie solaire et d'un réservoir d'eau, une aire réservée à la production horticole avec un système de goutte-à-goutte, un poulailler et, pour quelques-unes d'un étang piscicole[7].

L'arrosage par aspersion est aussi utilisé surtout par les cultivateurs de banane et d'ananas selon Monsieur Aniset Bassaine.

## **II.3 - Problèmes liés aux techniques d'irrigation traditionnelles**

### **II.3.a - L'utilisation de l'eau mal maîtrisée**

Le principal problème rencontré avec les techniques d'irrigation traditionnelles est l'utilisation non maîtrisée de l'eau.

L'efficacité des systèmes d'irrigations pratiqués est très faible, parfois même inférieure à 50 %. Ce qui signifie qu'un faible volume d'eau est effectivement utilisé pour l'alimentation des plantes. Sur 100 litres près de 50 litres sont perdus. Ces volumes sont rejetés dans le système de drainage et finissent par s'évaporer ou sont perdus par infiltration. Avec la proximité des nappes phréatiques, l'excès infiltration entraîne la remontée du niveau de la nappe, ce qui dans bien des cas, s'accompagne de remontées capillaires et de la salinisation des sols[4]. Ce qui nous amène à notre deuxième point, la salinisation.

### **II.3.b - La salinisation des sols**

Le surplus d'eau utilisé n'est pas sans conséquence, il provoque la destruction des sols cultivables.

En effet, 20 % des terres irriguées ont des problèmes de salinités. Lorsque l'irrigation est trop abondante pour être absorbée par les racines des plantes (absence d'un système efficace de drainage), le sol est humidifié en profondeur, permettant au sel de remonter à la surface[8].

La salinisation des terres demeure une réelle menace pour la sécurité alimentaire au Sénégal. Le phénomène, selon Mame Yacine Badiane Ndour, chercheuse à l'Institut sénégalais de recherche agricole, touche plus d'un million d'hectares de surfaces cultivables dans les régions côtières. Ce qui équivaut à 10 000 km<sup>2</sup>, soit, plus de deux fois la superficie d'une région comme Diourbel (centre du Sénégal)[9].

### ***II.3.c - Efforts physiques considérables***

Il y a aussi le problème lié au travail physique. En effet, selon Monsieur Aniset Bassaine, 6 arrosoirs de 10 à 11 Litres sont nécessaires pour une planche de 10 m<sup>2</sup>. S'ajoute à cela le temps de remplissage des bassins avec de l'eau puisée ou d'un forage selon les moyens. Si nous prenons l'exemple de l'irrigation par ruissellement, en plus de devoir l'organiser en planches, le terrain doit être incliné de sorte qu'il y est une pente de 0,2 à 3 %.

### ***II.3.d - Baisse de la productivité***

Tous les problèmes cités plus haut influent fortement sur la productivité. Le sel qui remonte à la surface détruit le sol et les cultures. Si nous prenons l'exemple de l'irrigation par arrosoir, l'effort physique nécessaire est si important que la surface cultivée s'en retrouve limitée, ce qui par la même occasion, limite la productivité.

## **III - Objectif du projet**

### **III.1 - Objectif général du projet**

L'objectif général du projet est de participer à la sécurité alimentaire du Sénégal par la proposition d'un système d'irrigation efficace qui permet de maîtriser l'utilisation de l'eau et de protéger les terres cultivables tout en réduisant l'effort physique fourni.

### **III.2 - Objectifs spécifiques du projet**

Les objectifs spécifiques du projet sont les suivants :

- ◆ proposer un système d'irrigation automatique qui n'arrose le sol qu'en cas de nécessité pour permettre de maîtriser totalement l'utilisation de l'eau,
- ◆ protéger le sol de la salinisation,
- ◆ réduire l'effort physique et augmenter les surfaces cultivées et donc la productivité,
- ◆ permettre à l'agriculteur de s'absenter tout en gardant un œil sur son champ ou jardin à l'aide d'un tableau de bord, et des messages émis par le système en cas de problème,
- ◆ donner à l'agriculteur la possibilité d'interagir avec le système en déclenchant ou en arrêtant l'arrosage à distance. Cela permet de ne pas se limiter à l'arrosage automatique qui se base sur un certain taux d'humidité du sol car selon les plantes, les besoins en eau peuvent différer,
- ◆ enfin, construire une base de données contenant des données sur le climat pouvant servir, après analyses, à améliorer l'agriculture sénégalaise menacée par les changements climatiques.

Maintenant que nous avons identifié les problèmes rencontrés avec les techniques d'irrigation traditionnelles et avons dégagé nos objectifs, parlons des exigences exprimées par ZIP avant de proposer notre solution qui doit les prendre en compte.

## **IV - Travail demandé**

L'objectif du stage est, comme explicité plus haut, d'implémenter un système d'irrigation automatique.

Ce système devra satisfaire certaines exigences exprimées par ZIP à savoir :

- ◆ le nom du système est **AgroDigital** ;
- ◆ la carte électronique programmable utilisée est **Arduino Mega** ;
- ◆ les capteurs utilisés : capteur d'humidité du sol et de l'air, capteur de température de l'air, capteur de pH et capteur de niveau d'eau ;
- ◆ un signal (SMS) doit être envoyé si le pH du sol ou le niveau de l'eau du réservoir atteint un seuil critique ou encore s'il y a un problème technique ;
- ◆ la plate-forme Web doit être conçue avec le framework **Angular** ;
- ◆ enfin la conception doit coûter la moins chère possible pour une durée de quatre mois.

## V - Solution proposée

Notre solution est la suivante : concevoir un système informatique embarqué qui permet d'arroser seulement si le sol est sec puis s'arrête automatiquement si ce dernier devient assez humide à l'aide de capteurs d'humidité disposés à même le sol. Cela permettra :

- ◆ d'éviter d'utiliser plus d'eau qu'il n'en faut et donc la salinisation du sol,
- ◆ de réduire considérablement l'effort physique, car il n'y aura rien à manœuvrer ce qui augmentera la surface cultivable.

Outre l'humidité du sol, la température et l'humidité ambiantes seront aussi mesurées par des capteurs, le tout envoyé dans une base de données. Cela permettra de suivre et d'interagir avec le système à distance à l'aide d'une plate-forme web. À travers ce site web, il doit être possible de déclencher ou d'arrêter l'arrosage. Le système devra aussi nous notifier par SMS en cas de panne technique ou si le pH ou le niveau de l'eau d'arrosage atteint un niveau critique.

Enfin, former une base de données avec les données collectées qui pourront être analysées pour adapter notre agriculture aux changements climatiques.

Il est important de noter que notre solution ne peut être utilisée que par un système d'irrigation disposant d'un système de pompage ou d'électrovanne pouvant être automatisé, en l'occurrence un système par goutte-à-goutte ou par aspersion.

L'électrovanne est un accessoire qui permet d'ouvrir ou de fermer l'arrivée d'eau de façon automatique, il est donc contrôlable par Arduino.

## Conclusion

Dans ce chapitre, nous avons fait le tour du sujet en l'étudiant dans les détails avants de nous fixer des objectifs pour résoudre les problèmes rencontrés avec les techniques d'irrigation traditionnelles. Afin d'atteindre ces objectifs, nous avons proposé notre solution. Mais avant d'aller plus loin, familiarisons-nous avec les systèmes d'irrigation existant dans le chapitre suivant.

## **CHAPITRE II : PRÉSENTATION DES DIFFÉRENTS SYSTÈMES D'IRRIGATION**

Dans ce chapitre, nous allons présenter les principaux systèmes d'irrigation existants dans le milieu de l'agriculture après avoir défini ce que sont l'irrigation et le système d'irrigation automatique. Un tableau comparant l'efficacité de ces systèmes d'irrigation sera aussi proposé avant de clore ce chapitre.

### **I - Qu'est-ce que l'irrigation ?**

« Fourniture artificielle et répartition systématique de l'eau pour l'agriculture et l'horticulture afin d'augmenter la production et améliorer sa qualité[10]. »

Il existe différents types de techniques d'irrigation qui diffèrent en fonction de la distribution sur le terrain de l'eau obtenue à partir de la source. En général, l'objectif est de fournir de l'eau à l'ensemble du terrain de manière uniforme, de sorte que chaque plante ait le volume d'eau dont elle a besoin, ni trop, ni trop peu.

### **II - Qu'est-ce qu'un système d'irrigation automatique ?**

La notion d'arrosage automatique désigne les systèmes permettant de fournir de l'eau à des plantes en routine durant un certain temps sans intervention humaine, plutôt par aspersion, mais il peut s'agir de système de type « goutte-à-goutte ». Ce sont la temporisation et le mouvement des jets qui sont automatisés, et parfois la pression. Ces systèmes d'arrosage peuvent être pilotés localement ou à distance et de manière plus ou moins automatisée ; une énergie hydraulique et/ou électrique est souvent nécessaire pour cette automatisation[11].

Pour faire simple, l'arrosage automatique désigne un ensemble de composants formant un système dont le but est d'approvisionner en eau un espace vert selon certaines conditions sans une quelconque intervention de l'homme.

Les premiers systèmes d'arrosage automatiques ont été mis au point en 1960. Ils étaient peu efficaces et très coûteux à l'époque, mais ont connu des améliorations progressives au fil des développements technologiques.

Il existe différents systèmes d'irrigation automatiques vendus sur le marché avec des fonctionnalités et qualités différentes. Ces derniers sont tous basés sur l'un des systèmes d'irrigation que nous allons citer ci-dessous.

### **III - Les différents systèmes d'irrigation**

Les systèmes d'irrigation peuvent être classés en trois grandes catégories à savoir l'irrigation gravitaire, l'irrigation goutte à goutte et l'irrigation par aspersion.

L'irrigation goutte à goutte et l'irrigation par aspersion sont aussi appelées irrigation sous pression.

### **III.1 - L'irrigation gravitaire**

L'irrigation gravitaire regroupe l'ensemble des techniques d'arrosage dans lesquelles la distribution de l'eau à la parcelle se fait entièrement à l'air libre par simple écoulement à la surface du sol. La répartition de l'eau est assurée grâce à la topographie du terrain, et aux propriétés hydriques du sol (ruissellement, infiltration, et capillarité)[12].

Il existe différents modes d'irrigation gravitaire, nous allons en citer certains :

#### ***III.1.a - Arrosage par ruissellement (par planches ou par calant)***

Consiste à faire couler une mince couche d'eau sur un sol incliné de 0,2 à 3 %, qui s'infiltrera verticalement jusqu'à l'humidification de cette tranche de sol, le débit à déverser est en fonction de la pente, de la largeur et de la longueur de la planche[12].



*Figure 1: Arrosage par ruissellement[31]*

#### ***III.1.b - Arrosage par submersion (ou inondation)***

Le principe consiste à donner au sol une couche d'eau plus au moins épaisse, qu'on laisse séjourner le temps nécessaire pour qu'elle pénètre par infiltration à la profondeur utile permettant ainsi au sol de mettre en réserve l'eau indispensable au développement des cultures [12].



*Figure 2: Arrosage par submersion[32]*



### **III.1.c - Arrosage par infiltration (à la raie)**

C'est une méthode qui consiste à faire distribuer l'eau par des rigoles ou raies avec un débit relativement grand (5 à 10 l/s) comparativement aux autres procédés. Dans cette méthode, une partie seulement du sol reçoit directement l'eau, le reste est humecté par infiltration latérale [12].



Figure 3: Arrosage par infiltration[31]

### **III.1.d - Arrosage par cuvette (ou bassin)**

Il est le plus connu des modes d'irrigation gravitaire, l'eau est apportée sous forme d'une nappe dans un bassin (qui peut être cloisonné) aménagé sur un sol nivelé (pente de 0,1 à 1 %). C'est une technique traditionnelle de montagne qui a été adaptée aux zones irriguées.

L'irrigation est faite par le découpage de la sole en plusieurs bassins (ou média) élémentaires dont les dimensions moyennes sont 40 m<sup>2</sup>.

Ces bassins sont irrigués par des canaux (*seguias*) de distribution qui à leur tour sont alimentés par une *segua* mère. L'eau est dérivée vers celle-ci en opérant une seule brèche sur l'arroseur.



Figure 4: Arrosage par cuvette[31]

Au niveau de la parcelle, l'eau suit l'itinéraire suivant : Arroseur-Segua de distribution pour enfin arriver aux bassins ou *media* irrigués à tour de rôle[12].

## III.2 - L'irrigation par goutte à goutte

Dans l'irrigation goutte à goutte, l'eau est livrée à la plante à faible dose entraînant ainsi l'humidification d'une fraction du sol. Ceci permet de limiter les pertes par évaporation et percolation. Elle permet aussi de réduire le développement des mauvaises herbes. Elle met également en œuvre des équipements fixes et légers, et permet la fertirrigation. Dans la plupart des cas, elle exige une automatisation à travers des contrôleurs associés à des vannes volumétriques et/ou hydrauliques et des électrovannes[13].

L'installation est composée d'une source d'eau, d'une station de pompage, d'une unité de tête, des canalisations principales et secondaires, de porte rampes et rampes, et enfin de distributeurs.

Décrivons certains de ces composants :

### III.2.a - L'unité de tête (contrôle/commande) :

Ses caractéristiques et équipements dépendent des besoins du système. Habituellement il comprend une vanne de sectionnement, des vannes de contrôle, une unité de filtrage, un injecteur d'engrais et d'autres petits accessoires[14].

### III.2.b - Les distributeurs

Pour les distributeurs on distingue les goutteurs, les gaines de micro-irrigation et les tuyaux à parois poreuses.

#### ◆ Les goutteurs

Les goutteurs sont de petits distributeurs en plastique de haute qualité. Ils sont montés à intervalle régulier sur de petits tuyaux flexibles en Polyéthylène. L'eau pénètre dans les goutteurs sous une pression d'environ 1 bar et ressort sans pression sous forme de gouttelettes continues avec un faible débit de 1 à 24 litres/heure[14].

Cependant tous les goutteurs ne sont pas montés sur le tuyau en usine, d'autres peuvent être achetés séparément et montés sur le tuyau d'arrosage en fonction des besoins.



Figure 5: Exemple de goutteur[33]

#### ◆ Les gaines de micro-irrigation

Il s'agit de lignes de tuyaux à paroi mince avec des points d'émission espacés de 10, 20, 30 ou 45 cm ou tout autre espacement, fournissant de plus faibles débits que les goutteurs habituels à de très faibles pressions,

c'est-à-dire 0,4 à 1 litre/heure sous 0,6 à 1 bar. Ce sont des tuyaux à goutteurs intégrés, car les goutteurs sont posés dans les parois du tuyau à des intervalles fixés à la fabrication. Ces gaines constituent des conduites latérales avec goutteurs prêtes à l'emploi qui offrent une uniformité d'application très élevée[14].

#### ◆ Les tuyaux à parois poreuses

Ces tuyaux de petit diamètre (environ 16 mm) sont des tubes flexibles à parois minces poreuses en fibres de PE, en Chlorure de polyvinyle, en caoutchouc... Ils permettent le passage à basse pression de l'eau et des éléments nutritifs par transpiration au travers du tube, qui vont irriguer les plantes. Le débit dans les tuyaux poreux n'est pas précis, car la dimension des pores est variable et instable. Ils sont utilisés comme tuyaux latéraux goutteurs en dessous de la surface. L'application de cette technique est limitée, bien qu'elle offre quelques avantages[14].



Figure 6: Exemple de tuyau à parois poreuses[34]

### III.3 - L'irrigation par aspersion

L'arrosage par aspersion est une manière d'irriguer qui consiste à asperger d'eau la surface plantée sous forme de pluie artificielle. L'arrosage se fait par surfaces rectangulaires ou par cercles, selon le type d'appareils utilisés par l'arrosant.

Une installation d'irrigation sous pression est généralement composée d'un équipement fournissant la pression nécessaire à son fonctionnement, d'appareils de mesure et de contrôle de débit, et d'une conduite principale amenant l'eau jusqu'aux conduites secondaires et tertiaires. D'autres éléments peuvent être utilisés, notamment un filtre et un dispositif d'adjonction d'éléments fertilisants[13].

Avec ce système d'arrosage, la distribution de l'eau se fait par des asperseurs. Il existe plusieurs types d'asperseurs de débit et de portée différentes selon les besoins.

#### ◆ Asperseurs avec Canon et tourniquet

Canons et tourniquets sont des systèmes rotatifs de projection de l'eau depuis un point fixe. Utilisés pour un arrosage circulaire[15].



Figure 7: Exemple d'asperseur avec canon et tourniquet[15]

◆ **Asperseurs avec Une rampe oscillante**

Sur cet arroseur mobile, les jets sont disposés en ligne sur une rampe qui oscille de droite à gauche à 180°[15].



Figure 8: Exemple d'asperseur avec Une rampe oscillante[15]

## IV - Comparaison entre les différents systèmes d'irrigation

Le tableau ci-dessous montre les différents rendements (rapport entre la quantité d'eau nécessaire à la culture et la quantité d'eau à fournir) en fonction du mode d'irrigation utilisé. Il permet de conclure que la micro-irrigation permet de réduire la consommation d'eau entre 20 et 40 % par rapport aux méthodes traditionnelles[16].

Technique d'Irrigation	Efficacité potentielle (%)
Inondation	40-50
Ruissellement	55-70
Arroseur rotatif	65-80
Canon d'arrosage	60-65
Goutteurs	80-95
Micro-aspersion	80-90

Tableau 1: Tableau de comparaison des différents systèmes d'irrigation[16]

Notons que l'inondation et le ruissellement sont des techniques d'irrigation gravitaire, l'arroseur rotatif et le canon d'arrosage des techniques d'irrigation par aspersion et les deux autres des techniques d'irrigation par goutte-à-goutte.

Le mode d'irrigation enterrée n'est pas listé ici, car il s'agit tout simplement de faire du goutte-à-goutte ou de l'aspersion non pas à la surface mais sous-terre. On peut utiliser pour cela une gaine de micro-irrigation en ce qui concerne le goutte-à-goutte et des turbines escamotables avec des tuyères pour l'aspersion.

AgroDigital vas permettre d'automatiser ces systèmes d'irrigation, en l'occurrence le goutte-à-goutte et l'aspersion qui peuvent être dotés d'un système de pompage ou d'électrovanne. Ce sont donc les seuls systèmes éligibles pour utiliser AgroDigital parmi ceux cités dans ce tableau.

## Conclusion

Dans ce chapitre, nous avons défini ce que sont l'irrigation et un système d'irrigation automatique avant de présenter les différents systèmes d'irrigation qui existent. Un tableau comparatif nous a aidé à voir nettement la différence d'efficacité qui existe entre les techniques traditionnelles et les techniques modernes.

# **DEUXIÈME PARTIE : CYCLE DE DÉVELOPPEMENT D'AGRODIGITAL ET ANALYSE DES BESOINS FONCTIONNELS**

## **Sommaire**

CHAPITRE III : CYCLE DE DÉVELOPPEMENT D'AGRODIGITAL.....	14
I - Pourquoi un modèle de cycle de vie d'un logiciel ?.....	14
II - Qu'est-ce que le cycle de vie d'un logiciel ?.....	14
III - Présentation de quelques modèles de cycle de vie traditionnels.....	16
IV - Présentation de quelques modèles agiles.....	18
CHAPITRE IV : SPÉCIFICATION ET ANALYSE DES BESOINS FONCTIONNELS.....	24
I - Qu'est-ce que l'UML?.....	24
II - Spécification et analyse des besoins de l'application avec Arduino.....	25
III - Spécification et analyse des besoins du site web.....	31

## CHAPITRE III : CYCLE DE DÉVELOPPEMENT D'AGRODIGITAL

Dans ce chapitre, après avoir présenté certains modèles de cycle de vies d'un logiciel, nous allons en choisir un pour nous aider à définir les différentes étapes de conception de notre système.

### I - Pourquoi un modèle de cycle de vie d'un logiciel ?

Comme souligné plus haut, nous devons réaliser ce projet dans le délai imparti de quatre mois. Afin d'assurer la conformité aux exigences, la qualité et la modération du coût, nous devons maîtriser toutes les étapes de la définition des besoins au déploiement en passant par l'analyse et la conception. Il est ainsi important d'organiser et d'optimiser le découpage en étapes de la réalisation du projet. C'est là qu'entrent en jeu les modèles de cycle de vie d'un logiciel. Ces derniers permettent d'intégrer, de coordonner et de gérer les différentes phases du cycle de vie d'un logiciel.

### II - Qu'est-ce que le cycle de vie d'un logiciel ?

Le « cycle de vie d'un logiciel » (en anglais software lifecycle), désigne toutes les étapes du développement d'un logiciel, de sa conception à sa disparition. L'objectif d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la validation du développement logiciel, c'est-à-dire la conformité du logiciel avec les besoins exprimés, et la vérification du processus de développement, c'est-à-dire l'adéquation des méthodes mises en œuvre.

L'origine de ce découpage provient du constat que les erreurs ont un coût d'autant plus élevé qu'elles sont détectées tardivement dans le processus de réalisation. Le cycle de vie permet de détecter les erreurs au plus tôt et ainsi de maîtriser la qualité du logiciel, les délais de sa réalisation et les coûts associés.

Le cycle de vie du logiciel comprend généralement à minima les activités suivantes :

- ◆ **définition des objectifs**, consistant à définir la finalité du projet et son inscription dans une stratégie globale,
- ◆ **analyse des besoins et faisabilité**, c'est-à-dire l'expression, le recueil et la formalisation des besoins du demandeur (le client) et de l'ensemble des contraintes,
- ◆ **conception générale**, il s'agit de l'élaboration des spécifications de l'architecture générale du logiciel,
- ◆ **conception détaillée**, consistant à définir précisément chaque sous-ensemble du logiciel,
- ◆ **codage (Implémentation ou programmation)**, soit la traduction dans un langage de programmation des fonctionnalités définies lors de phases de conception,
- ◆ **tests unitaires**, permettant de vérifier individuellement que chaque sous-ensemble du logiciel est implémentée conformément aux spécifications,
- ◆ **intégration**, dont l'objectif est de s'assurer de l'interfaçage des différents éléments (modules) du logiciel. Elle fait l'objet de tests d'intégration consignés dans un document,

- ◆ **qualification (ou recette)**, c'est-à-dire la vérification de la conformité du logiciel aux spécifications initiales,
- ◆ **documentation**, visant à produire les informations nécessaires pour l'utilisation du logiciel et pour des développements ultérieurs,
- ◆ **mise en production**,
- ◆ **maintenance**, comprenant toutes les actions correctives (maintenance corrective) et évolutives (maintenance évolutive) sur le logiciel.

La séquence et la présence de chacune de ces activités dans le cycle de vie dépend du choix d'un modèle de cycle de vie entre le client et l'équipe de développement[17].

### III - Présentation de quelques modèles de cycle de vie traditionnels

#### III.1 - Modèle en cascade

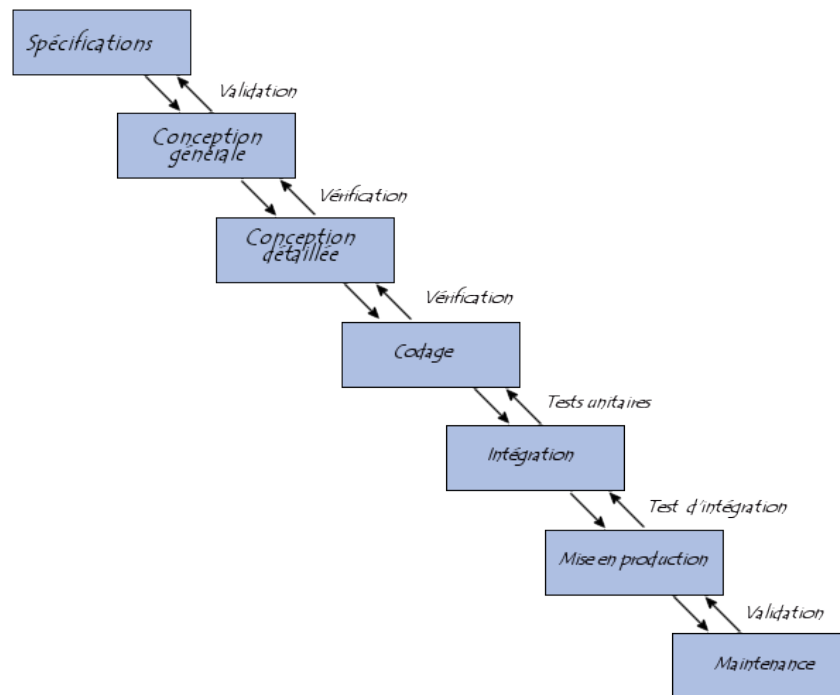


Figure 9: Modèle en cascade[18]

Le modèle de cycle de vie en cascade a été mis au point dès 1966, puis formalisé aux alentours de 1970. Il définit des phases séquentielles à l'issue de chacune desquelles des documents sont produits pour en vérifier la conformité avant de passer à la suivante[18].



### III.2 - Modèle en V

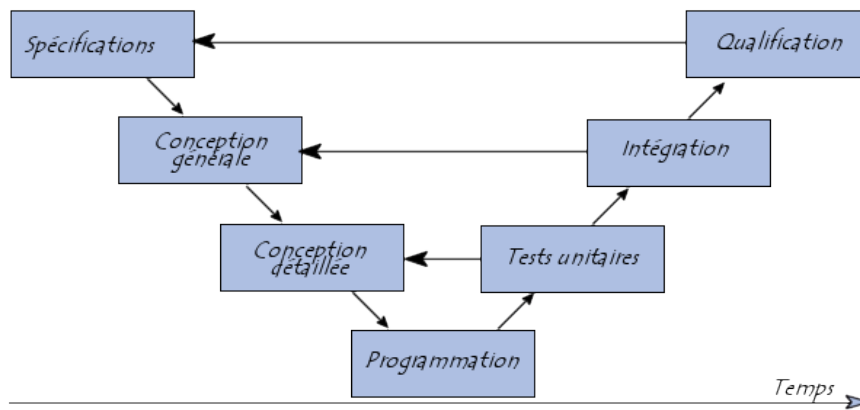


Figure 10: Modèle en V[18]

Le modèle de cycle de vie en V part du principe que les procédures de vérification de la conformité du logiciel aux spécifications doivent être élaborées dès les phases de conception [18].

### III.3 - Modèle en spirale

Le modèle en spirale a été défini par Barry Boehm en 1988 dans son article « A Spiral Model of Software Development and Enhancement ». Ce cycle ressemble beaucoup au cycle en V, mais ce cycle permet mieux de gérer les risques à chaque itération. On n'a pas la possibilité de retourner en arrière.

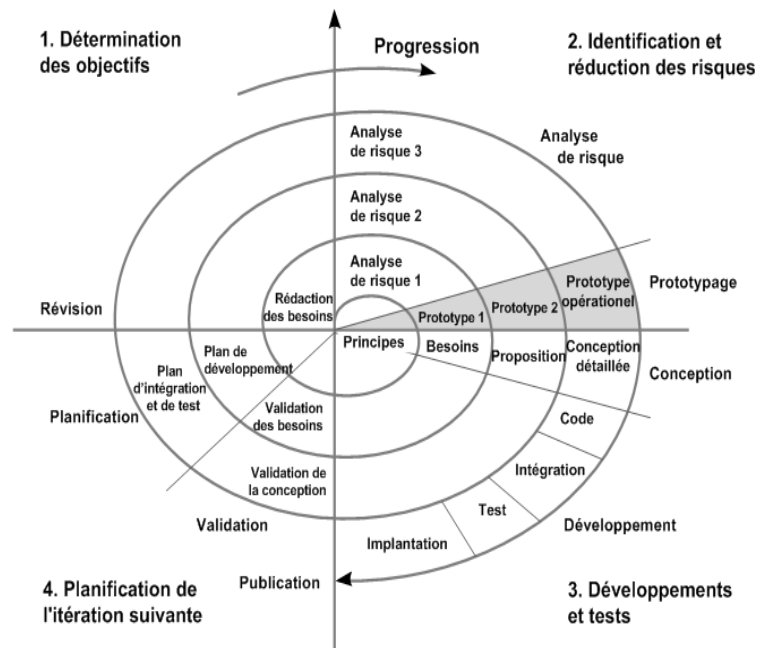


Figure 11: Modèle en spirale[19]

Cette méthode constitue quatre phases.

1. Détermination des objectifs ;  
Ici on détermine les différents objectifs à attendre dans l'itération courante.
2. Analyse et résoudre les risques ;

On analyse les différents risques causés par cette itération.

3. Développement et vérification de la solution pour planifier la phase suivante ;

Commencer les développements de la solution en question.

4. Revue des résultats et vérification du cycle pour valider le développement ;

La recette permet de s'assurer formellement que le produit est conforme à l'objectif [19].

## IV - Présentation de quelques modèles agiles

### IV.1 - ASD (Adaptative Software Development)

En 2000, Jim Highsmith (signataire du Manifeste) publie un ouvrage sur la méthode ASD, Adaptative Software Development, a collaborative approach to managing complex systems.

Le cycle de vie d'un projet ASD se déroule autour d'une série de cycles en trois volets :

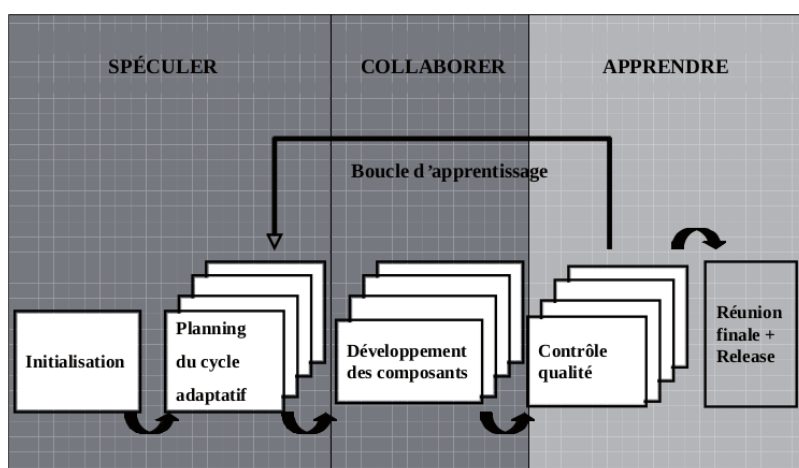


Figure 12: Modèle Adaptative Software Development[20]

- ◆ Spéculation
  - ✓ initier le projet (mission, contraintes, collaborateurs, expression des exigences, identification des risques...),
  - ✓ déterminer la durée du projet, le nombre d'itérations et les dates associées (4 à 8 semaines par itération),
  - ✓ affecter un objectif (mission) à chaque itération,
  - ✓ dresser une liste de tâches à réaliser.
- ◆ Collaboration
  - ✓ livraison des composants,
  - ✓ communication forte et assez informelle.
- ◆ Apprentissage

- ✓ contrôle qualité,
- ✓ suivi et bilan d'avancement,
- ✓ communication forte et assez informelle.

Ses caractéristiques principales sont :

- ◆ focaliser sur l'objectif (mission focused),
- ◆ se baser sur des composants (component-based),
- ◆ itérer,
- ◆ découper le temps et fixer des deadlines (timeboxing),
- ◆ piloter le projet par les risques (risk-driven development),
- ◆ accepter le changement.

Ce cycle de vie permet un apprentissage et une adaptation permanents aux évolutions du projet[20].

## IV.2 - RAD (Rapid Application Development)

Si RAD n'est pas à proprement parler une méthode agile, il n'en demeure pas moins qu'elle en est à l'origine, puisque dès la fin des années 1980, James Martin présentait la première approche (semi)itérative incrémentale préconisant un usage intensif des techniques de communication facilitée. Jean-Pierre Vickoff a adapté le RAD au contexte français (voir <http://www.rad.fr>).

Cinq phases structurent le développement rapide d'applications :

- ◆ **L'initialisation** : cette phase définit le périmètre du projet, organise le travail par thème et détermine les ressources nécessaires.
- ◆ **Le cadrage** : il s'agit de la phase d'expression des besoins, au cours de sessions, réunions qui favorisent la productivité des groupes de travail grâce à des techniques d'animation spécifiques.
- ◆ **Le design** : c'est la phase de conception au cours de laquelle le système est modélisé. Les utilisateurs y sont associés pour valider les modèles, l'ergonomie et la cinématique générale de l'application.
- ◆ **La construction** : l'équipe ou SWAT (unité d'élite) construit l'application de façon itérative, par module ou par thème, que valident les utilisateurs toujours impliqués dans le projet.
- ◆ **La finalisation** : cette dernière phase officialise la livraison globale de l'application, déjà partiellement validée par les utilisateurs dans les phases précédentes[20].

## IV.3 - UP (Unified Process)

Tout comme le RAD, le Processus unifié (UP) est sujet à controverse lorsqu'on l'assimile aux méthodes agiles ; certains déplorent sa lourdeur et sa grande similarité avec un cycle en cascade ; d'autres, au contraire, le considèrent comme une méthode agile très documentée et qui, par là même, permet une grande adaptation au contexte du projet. Si l'on veut être impartial, on pourrait dire qu'il se situe à l'intermédiaire des deux approches et qu'il est, grâce à sa flexibilité, agile selon qu'on le rend agile ou pas.

UP est le fruit d'une collaboration entre Ivar Jacobson, Grady Booch et James Rumbaugh, tous trois à l'origine de travaux sur le développement orienté objet et les techniques de modélisation.

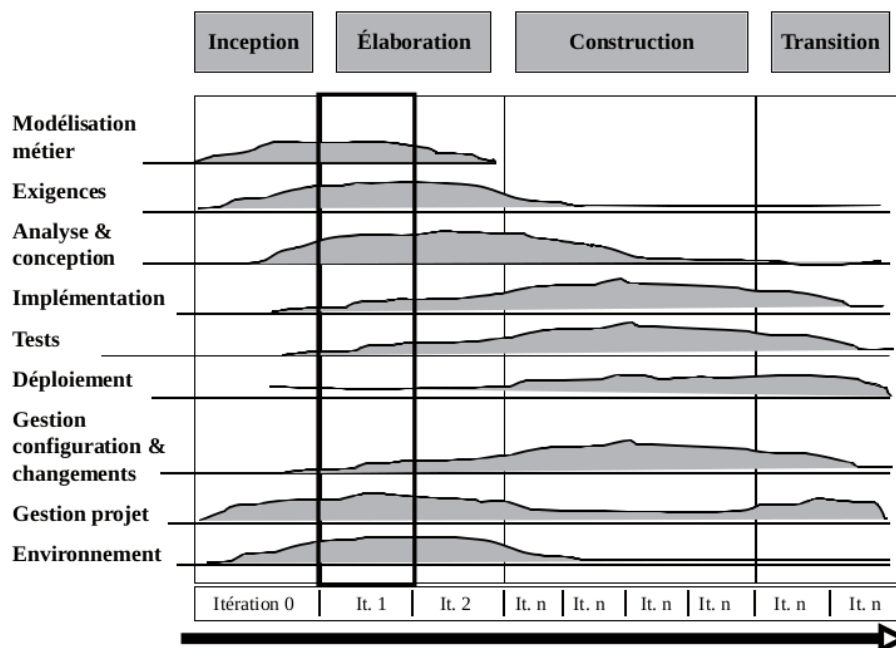


Figure 13: Les quatre phase du modèle Unified Process[20]

Le Processus unifié se décompose en quatre phases et neuf disciplines décrites par des activités, des rôles et des artefacts :

- ◆ **Inception** : la phase d'inception a pour objectif d'obtenir un consensus de l'ensemble des parties prenantes sur la vision ou la portée du projet, le périmètre et l'estimation globale.
- ◆ **Élaboration** : la phase d'élaboration a pour objectifs d'établir et de valider l'architecture de référence, de lever les principaux risques ainsi que de spécifier en détail les exigences ; c'est une phase exploratoire.
- ◆ **Construction** : la phase de construction a pour objectif de livrer une première version opérationnelle du système, accompagnée de sa documentation.
- ◆ **Transition** : la phase de transition a pour objectif de déployer et mettre en production la version finale de l'application, testée par les utilisateurs.

Toutes les disciplines se déroulent durant une itération[20].

#### IV.4 - Scrum

Les racines de Scrum se trouvent dans la publication de « The New Product Development Game » de Takeuchi et Nonaka ; cet article, paru en 1986 dans Harvard Business Review, démontre la fin des approches classiques dans le développement de nouveaux produits et met en avant les vertus des petites équipes pluridisciplinaires intégrées et soudées dans une stratégie plus flexible.

Inspirés par cette nouvelle approche et par le lean management, Ken Schwaber et Jeff Sutherland (signataires du Manifeste) ont développé Scrum, en 1993.

## Qu'est-ce que le lean management ?

Le lean management est un mouvement de pensée né au Japon, en particulier dans l'industrie automobile chez Toyota. Celui-ci veille à la réduction des pertes générées à l'intérieur d'une organisation (industrie, services...), pour une production dite « au plus juste » (just in time).

Les objectifs sont la réduction des cycles de production, la réduction des stocks, l'amélioration de la productivité et l'optimisation de la qualité.

Le cycle de vie Scrum est rythmé par des itérations de quatre semaines, les sprints ; la liste des exigences initiales, dressée et hiérarchisée avec le client constitue le référentiel, le **product backlog**.

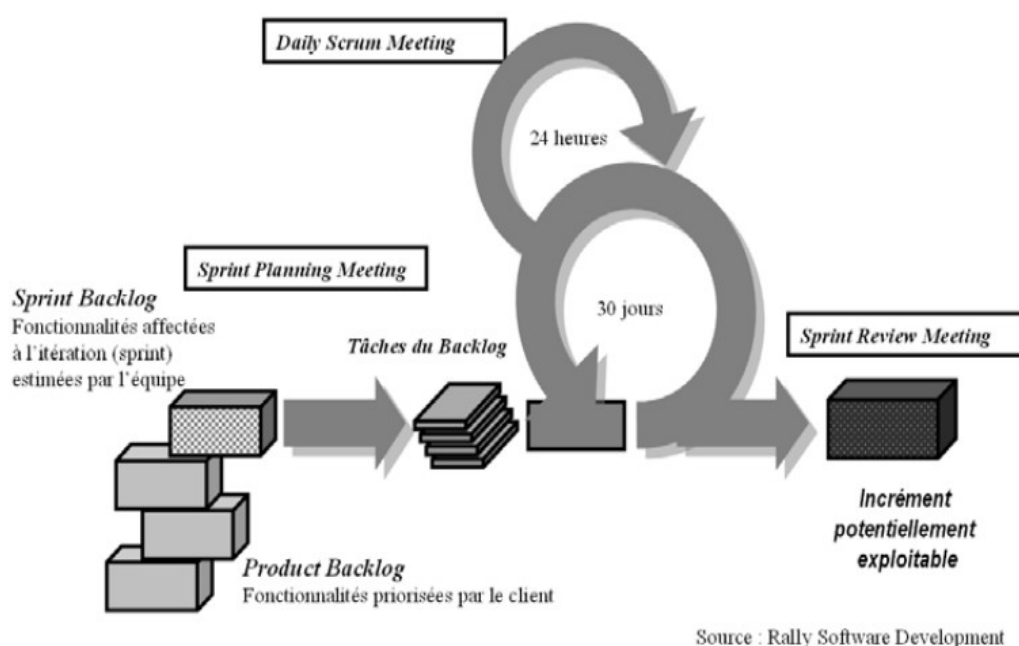


Figure 14: Illustration du processus de Scrum[20]

Avant chaque sprint, au cours d'une réunion de planification, le sprint planning meeting, on sélectionne, dans le product backlog, les exigences les plus prioritaires pour le client qui seront développées, testées et livrées au client ; elles constituent le sprint backlog, un sous-ensemble du product backlog.

Au cours du sprint, chaque jour, une réunion d'avancement ou mêlée est organisée avec tous les membres de l'équipe pour s'assurer que les objectifs du sprint seront tenus. À la fin du sprint, une démonstration des derniers développements est faite au client (Sprint Review Meeting), puis une revue de fin d'itération, une rétrospective, donne lieu à un bilan qualitatif sur le fonctionnement de l'équipe.

Scrum définit également un certain nombre de rôles au sein de l'équipe, étendue au client.

- ◆ **Le ScrumMaster** : c'est le manager du projet, chargé d'assister l'équipe pour appliquer la philosophie et les pratiques de Scrum ; il est celui qui facilite la résolution des problèmes.
- ◆ **Le product owner** : il est le « propriétaire » du product backlog, en tant que représentant des utilisateurs et, par conséquent, du retrait ou de l'ajout des exigences, de leur « priorisation » en

fonction de la valeur ajoutée qu'elles apportent aux utilisateurs et à l'organisation. Il est fortement impliqué dans les réunions de planification et d'avancement.

- ◆ **L'équipe** : elle est composée de tous les corps de métier nécessaires à l'implémentation d'une fonctionnalité. Elle est responsable de ses décisions et s'autogère.

Les valeurs mises en avant par Scrum sont les suivantes : visibilité, inspection et adaptation.

- ◆ **Visibilité** : pour évaluer les résultats du processus, ceux-ci doivent être réels et non soumis à interprétation. Une fonctionnalité est réellement implémentée (done) lorsque tous les acteurs s'entendent sur les modalités d'évaluation du résultat et sur le résultat lui-même. Le résultat est tangible, visible et non hypothétique.
- ◆ **Inspection** : l'inspection consiste à vérifier les écarts par rapport à l'objectif initial. De nombreux et fréquents points de contrôle et d'inspection sont prévus par Scrum.
- ◆ **Adaptation** : l'adaptation est nécessaire lorsque, précisément, des écarts non acceptables sont constatés ; des ajustements seront décidés immédiatement afin d'éviter des écarts plus importants encore. Scrum est favorable à de petits ajustements fréquents[20].

#### IV.5 - Synthèse des différences fondamentales entre approche traditionnelle et approche agile

Thème	Approche traditionnelle	Approche agile
Cycle de vie	En cascade ou en V, sans rétroaction possible, phases séquentielles.	Itératif et incrémental.
Planification	Prédictive, caractérisée par des plans plus ou moins détaillés sur la base d'un périmètre et d'exigences définies et stables au début du projet.	Adaptative avec plusieurs niveaux de planification (macro – et micro-planification) avec ajustements si nécessaires au fil de l'eau en fonction des changements survenus.
Documentation	Produite en quantité importante comme support de communication, de validation et de contractualisation.	Réduite au strict nécessaire au profit d'incrément fonctionnels opérationnels pour obtenir le feedback du client.
Équipe	Une équipe avec des ressources spécialisées, dirigées par un chef de projet.	Une équipe responsabilisée où l'initiative et la communication sont privilégiées, soutenue par le chef de projet.
Qualité	Contrôle qualité à la fin du cycle de développement. Le client découvre le produit fini.	Un contrôle qualité précoce et permanent, au niveau du produit et du processus. Le client visualise les résultats tôt et fréquemment.
Changement	Résistance voire opposition au changement. Processus lourds de gestion des changements acceptés.	Accueil favorable au changement inéluctable, intégré dans le processus.

Thème	Approche traditionnelle	Approche agile
Suivit de l'avancement	Mesure de la conformité aux plans initiaux. Analyse des écarts.	Un seul indicateur d'avancement : le nombre de fonctionnalités implémentées et le travail restant à faire.
Gestion des risques	Processus distinct, rigoureux, de gestion des risques.	Gestion des risques intégrée dans le processus global, avec responsabilisation de chacun dans l'identification et la résolution des risques. Pilotage par les risques.
Mesure du succès	Respect des engagements initiaux en termes de coûts, de budget et de niveau de qualité.	Satisfaction client par la livraison de valeur ajoutée.

Tableau 2: Synthèse des différences fondamentales entre approche traditionnelle et approche agile[20]

Après cette étude comparative de ces modèles de cycle de vie d'un logiciel, nous avons jeté notre dévolu sur Scrum qui est d'ailleurs l'un des modèles agiles les plus utilisés.

Ce choix se justifie en partie par le fait que Scrum, comme toutes les méthodes agiles, permet d'éviter les inconvénients comme l'effet tunnel et la rigidité de l'approche présents dans la quasi-totalité des méthodes traditionnelles.

D'autre part, manquant moi-même d'expérience dans le domaine, il me semble plus raisonnable d'adopter une approche qui permet facilement de revenir sur certaines fonctionnalités à des fins d'améliorations et d'ajustements. En effet, avec les modèles classiques, une fois le plan de management du projet validé, il constitue la référence de base et donc les changements ne sont pas faciles à prendre en compte.

## IV.6 - Notre équipe projet

Acteur	Rôle	Grade
Ismaila DIALLO	Scrum master	Ingénieur en génie logiciel et maître du stage
Baba Isaa CAMARA	Product owner	Maître de conférence à l'université de Lorraine région de Metz en France dans le domaine des mathématiques
Cheikh Talla KOUNDOUL	Scrum team (équipe)	Étudiant en Master 2 génie logiciel

Tableau 3: Notre équipe projet

## Conclusion

Nous avons présenté certains des modèles de cycle de vie d'un logiciel les plus utilisés. Nous nous sommes rendu compte qu'ils pouvaient être divisés en deux groupes à savoir les approches traditionnelles et les approches agiles. Après un tableau comparatif de ces deux groupes de modèle de cycle de vie, nous avons jeté notre dévolu sur le modèle agile Scrum. Scrum va nous aider à définir les différentes phases de la conception de notre système.

## CHAPITRE IV : SPÉCIFICATION ET ANALYSE DES BESOINS FONCTIONNELS

Dans cette partie, il sera question d'étudier en détail l'ensemble des fonctionnalités offertes par notre système. Pour cela, nous allons nous baser sur les besoins exprimés par ZIP à travers le sujet de stage. À l'issue de cette étude, nous allons produire des représentations graphiques à l'aide du langage de modélisation UML (Langage de Modélisation Unifié). Pour mieux organiser notre travail, nous allons utiliser le modèle agile Scrum.

Comme notre système est composé de deux applications différentes, nous allons structurer cette partie comme suit : d'abord nous définissons ce qu'est UML, ensuite nous passons à la spécification et à l'analyse des besoins de l'application Arduino pour finir avec le site web.

### I - Qu'est-ce que l'UML ?

La notation UML est un **langage visuel** constitué d'un ensemble de schémas, appelés des **diagrammes**, qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour **représenter** le logiciel à développer : son fonctionnement, sa mise en route, les actions susceptibles d'être effectuées par le logiciel, etc.

Réaliser ces diagrammes revient donc à **modéliser les besoins** du logiciel à développer[21].

UML définit neuf types de diagrammes divisés en deux catégories :

- ◆ Diagrammes statiques (structurels) :
  - ✓ diagramme de classe
  - ✓ diagramme d'objet
  - ✓ diagramme de composant
  - ✓ diagramme de déploiement
  - ✓ de diagramme de cas d'utilisation.
- ◆ Diagrammes dynamiques (comportementaux) :
  - ✓ diagramme d'activité
  - ✓ diagramme de séquence
  - ✓ diagramme d'état-transition
  - ✓ de diagramme de collaboration.



## II - Spécification et analyse des besoins de l'application avec Arduino

### II.1 - Spécification des besoins

La définition des besoins consiste à décrire les différents services rendus par notre application et surtout à qui devra-t-elle les rendre. Cela passe donc forcément par l'étude des futurs utilisateurs de l'application et comment ces derniers vont l'utiliser.

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié. Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données[22].

Quant aux fonctionnalités, elles représentent les services rendus par l'application à un acteur. Elles peuvent être externes (nécessite l'intervention d'un acteur) ou internes au système.

#### II.1.a - Identification des acteurs

Dans le tableau suivant, nous listons les différentes entités qui interagissent avec l'application.

Acteur secondaire	Rôle (s)
Téléphone de l'utilisateur	✗ Recevoir des messages (SMS) de la part d'Arduino.
Base de données	✗ Recevoir les données envoyées par Arduino (données mesurées par les capteurs). ✗ Recevoir le nouvel état du moteur (la pompe à eau) de la part d'Arduino. ✗ Stocker la nouvelle commande à exécuter.

Tableau 4: Identification des acteurs pour Arduino

Notons que la base de données est un acteur secondaire car ne fait que recevoir les données envoyées par Arduino de même que le téléphone de l'utilisateur qui ne fait que recevoir les SMS envoyés par Arduino.

#### II.1.b - Identification des fonctionnalités

Le tableau suivant représente les fonctionnalités.

Fonctionnalités	Acteur secondaire
Envoyer un signal d'alerte si le pH ou le niveau de l'eau atteint un seuil critique ou s'il y a des problèmes de connexion internet (problème de réseau).	Téléphone de l'utilisateur
Envoyer les données mesurées sur internet.	Base de données
Envoyer le nouvel état du moteur lorsque celui-ci vient d'être allumé ou éteint.	Base de données
Lire et exécuter la nouvelle commande envoyée par l'utilisateur à partir de son tableau de bord.	Base de données

Tableau 5: Identification des fonctionnalités

### **II.1.c - Description de quelques cas d'utilisation**

#### **II.1.c.1 - Description du cas « Envoyer les données sur internet »**

Identification	
numéro	1
nom	Envoyer les données sur internet
Acteur secondaire	Base de données
Description	Les données mesurées par les capteurs sont envoyées à temps réel sur un serveur de base de données distant.
Auteur	ZIP
Dates	03/03/2019
Préconditions	
Démarrage	Le système collecte les données pour les poster sur le serveur
Scénario nominal	
1	Le système se connecte à internet
2	Le système formate en JSON les données et les envoie
3	Le système se déconnecte
Scénarios d'exception	
1	La connexion à internet échoue. Le système envoie à l'utilisateur un SMS pour l'alerter de cet échec.

Tableau 6: Description du cas « Envoyer les données sur internet »

#### **II.1.c.2 - Description du cas « Envoyer le nouvel état du moteur »**

Identification	
numéro	2
nom	Envoyer le nouvel état du moteur
Acteur secondaire	Base de données
Description	L'humidité du sol a atteint un seuil fixé (soit le sol est trop sec, soit il est assez humide)
Auteur	ZIP
Dates	03/03/2019
Préconditions	
Démarrage	Le système collecte les données via les capteurs puis les vérifie.
Scénario nominal	
1	Le système se rend compte que le sol est trop sec ou assez humide.
2	En mode automatique, le système allume la pompe d'eau si le sol est trop sec ou l'éteint s'il est assez humide. En mode manuel, le système informe l'utilisateur par SMS pour que ce dernier déclenche ou arrête manuellement l'arrosage selon l'état du sol.

3	Le système se connecte à internet.
4	Le système formate en JSON le nouvel état et met à jour la base de données.
5	Le système se déconnecte.
Scénarios d'exception	
2	La connexion à internet échoue. Le système envoie à l'utilisateur un SMS pour l'alerter de cet échec.

Tableau 7: Description du cas « Envoyer le nouvel état du moteur »

### II.1.c.3 - Description du cas « Exécuter la nouvelle commande »

Identification	
numéro	3
nom	Exécuter la nouvelle commande
Acteur	Base de données
Description	Le moteur est manuellement allumé ou éteint avec des commandes (déclencher ou arrêter l'arrosage) envoyées par l'utilisateur via son tableau de bord. Dans ce cas, le système passe en mode manuel. Il est possible de repasser en mode automatique en envoyant la commande « <i>passer en mode automatique</i> ».
Auteur	ZIP
Dates	03/03/2019
Préconditions	
Démarrage	L'utilisateur veut déclencher ou arrêter l'arrosage manuellement ou passer en mode automatique.
Scénario nominal	
1	L'utilisateur clique sur le bouton : <ul style="list-style-type: none"> <li>✗ <b>Déclencher l'arrosage (ON)</b> pour allumer le moteur,</li> <li>✗ <b>Arrêter l'arrosage (OFF)</b> pour éteindre le moteur,</li> <li>✗ <b>Passer en mode automatique (AUTO)</b> pour repasser en mode automatique.</li> </ul>
2	Le système lie la table des commandes sur la base de données et exécute la nouvelle commande.
3	Si la commande est <b>ON</b> ou <b>OFF</b> : <ul style="list-style-type: none"> <li>✗ le système allume ou éteint le moteur,</li> <li>✗ se connecte à internet,</li> <li>✗ formate en JSON le nouvel état du moteur et met à jour la base de données,</li> <li>✗ passe en mode manuel.</li> </ul> Si la commande est <b>AUTO</b> , le système repasse en mode automatique .
4	Le système se déconnecte si la commande fut <b>ON</b> ou <b>OFF</b> .
Scénarios d'exception	
1	La connexion à internet échoue. Le système envoie à l'utilisateur un SMS pour l'alerter de cet échec.

Tableau 8: Description du cas « Exécuter la nouvelle commande »

## II.2 - Analyse des besoins fonctionnels

À ce stade, il est question d'analyser les fonctionnalités de l'application en l'occurrence les différents cas d'utilisations identifiés précédemment. Pour cela, nous allons utiliser les diagrammes de séquences et d'activité d'UML.

### II.2.a - Analyse du cas « Envoyer les données sur internet »

#### II.2.a.1 - Diagramme de séquence

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique donné. Ils permettent de montrer les interactions d'objets dans le cadre d'un scénario d'un diagramme des cas d'utilisation.

Tous les diagrammes de séquence réalisés dans ce document représentent chacun le scénario nominal ainsi que les scénarios alternatifs du cas d'utilisation étudié.

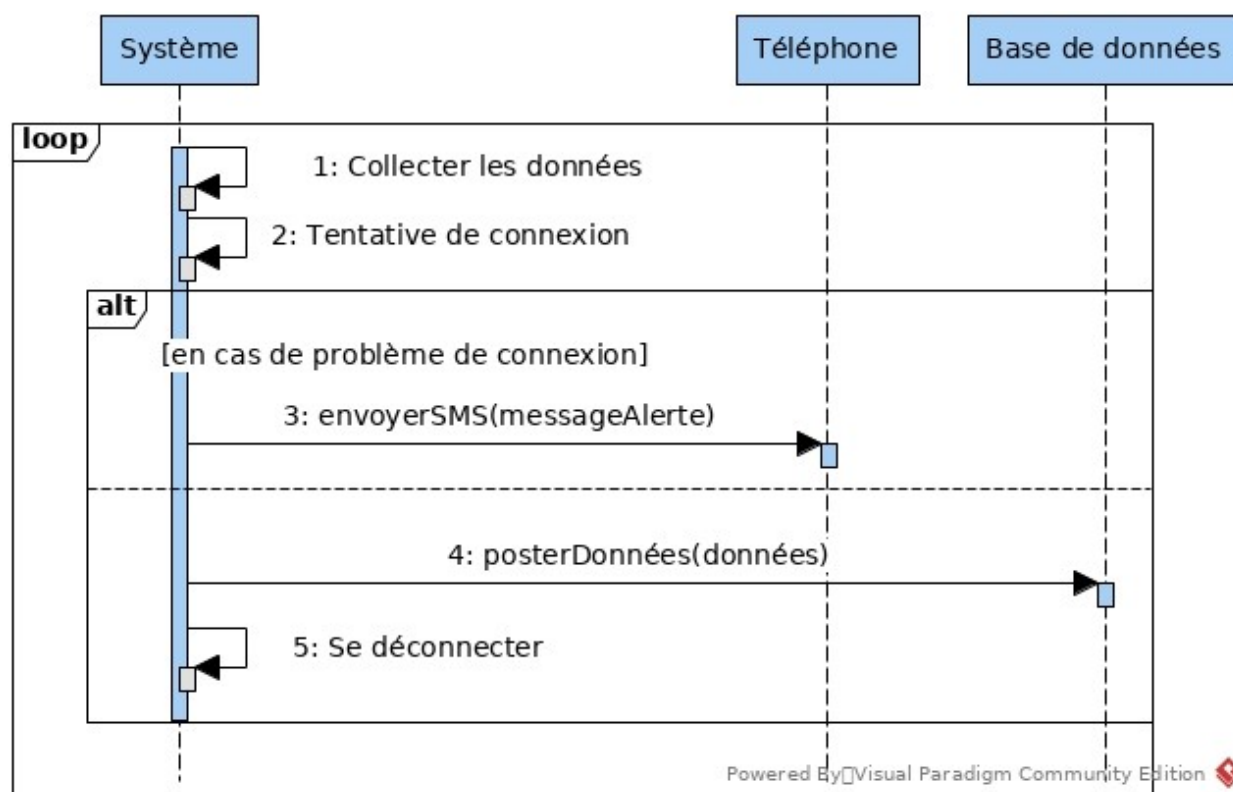


Figure 15: Diagramme de séquence du cas « Envoyer les données sur internet »

Les données ne sont envoyées sur internet (l'opération **posterDonnées(données)**) que s'il n'y a pas de problème de connexion internet.

**NB** : la collecte des données ne fait clairement pas partie du cas étudié ici. Elle est là pour faciliter la lecture et la compréhension du diagramme. Elle est présente au niveau de tous les diagrammes de cette partie d'analyse pour les mêmes raisons.

### II.2.a.2 - Diagramme d'activité

Le diagramme d'activité est un diagramme de comportement qui montre l'enchaînement des actions et des décisions au sein d'une activité du système.

Il permet de décrire les grandes lignes et orientation des actions du fonctionnement interne, en montrant les alternatives et les différentes relations possibles avec des sous-systèmes. On peut par exemple l'utiliser pour préciser le comportement d'un cas d'utilisation avec tous ses scénarios alternatifs.

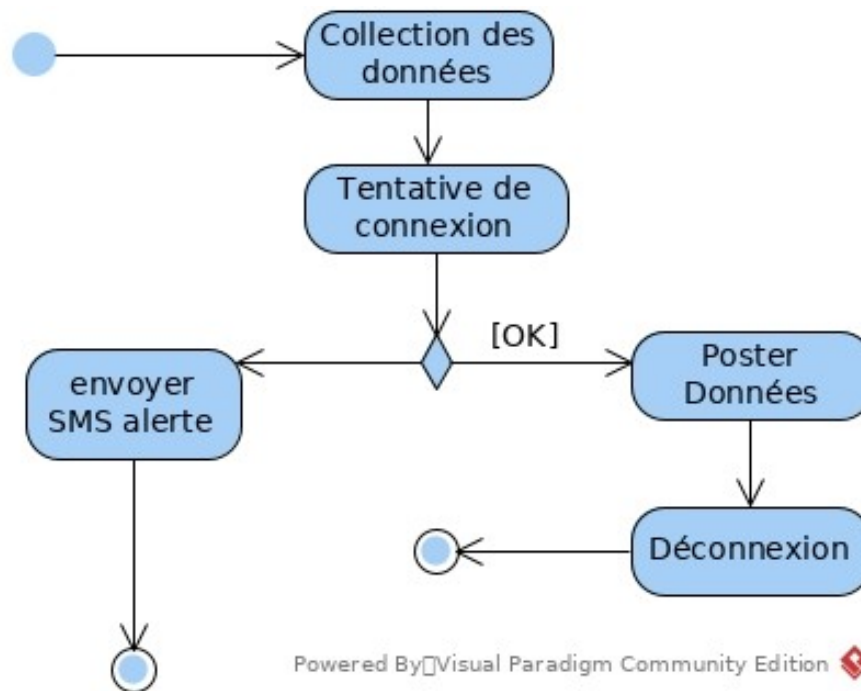


Figure 16: Diagramme d'activité du cas « Envoyer les données sur internet »

### II.2.b - Analyse du cas « Envoyer le nouvel état du moteur »

#### II.2.b.1 - Diagramme de séquence

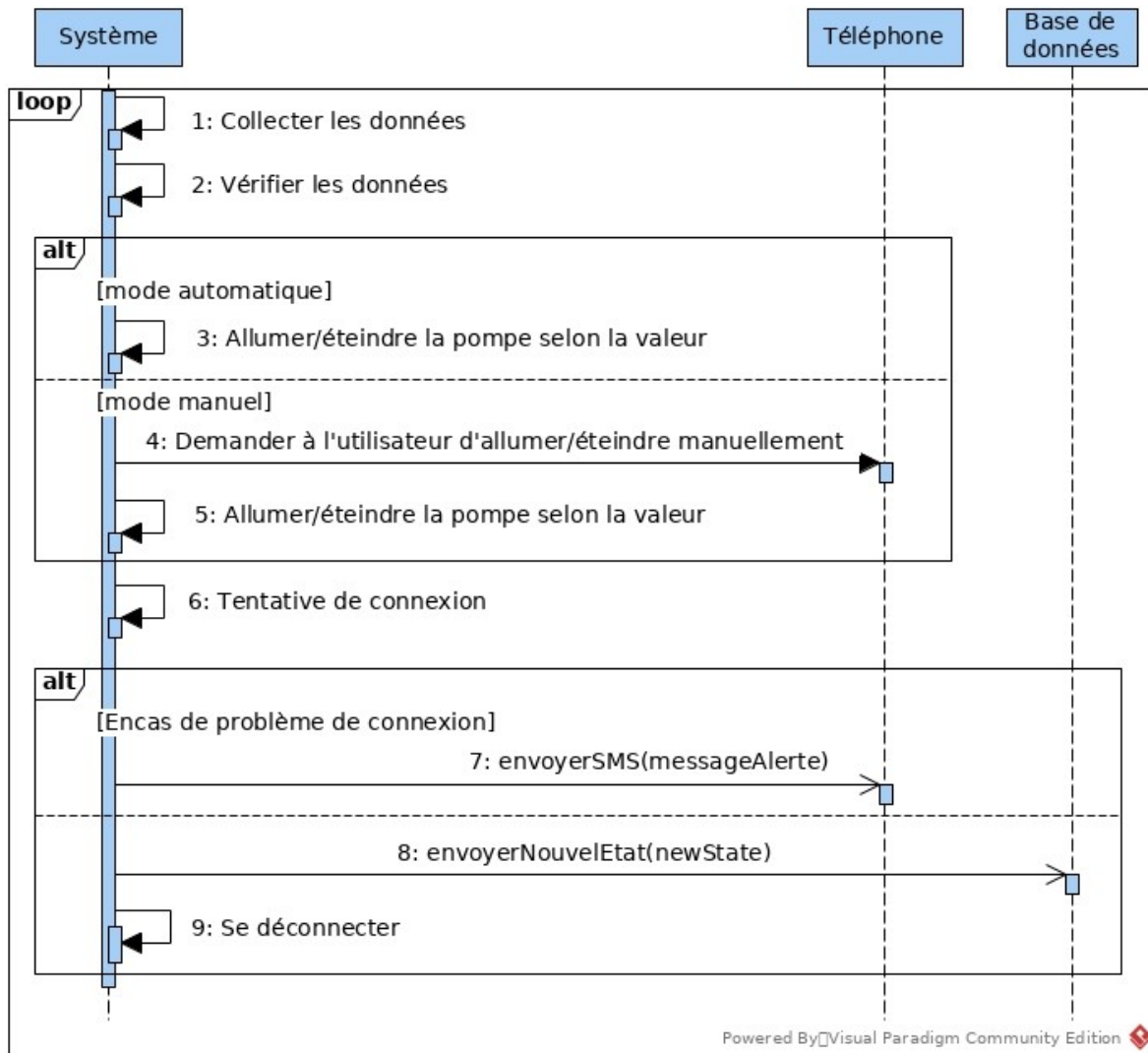


Figure 17: Diagramme de séquence du cas « Envoyer le nouvel état du moteur »

Pour allumer ou éteindre la pompe, nous nous basons sur l'humidité du sol. À chaque fois que cette dernière atteint une valeur seuil, la pompe à eau est allumée ou éteinte selon cette valeur et ce nouvel état est enregistré dans la base de données.

### II.2.b.2 - Diagramme d'activité

Le diagramme d'activité nous montre clairement que le cas peut se solder par un échec ou par un succès selon que le système parvienne ou pas à établir la connexion.

L'échec se termine par l'envoi d'un signal d'alerte et le succès par une déconnexion.

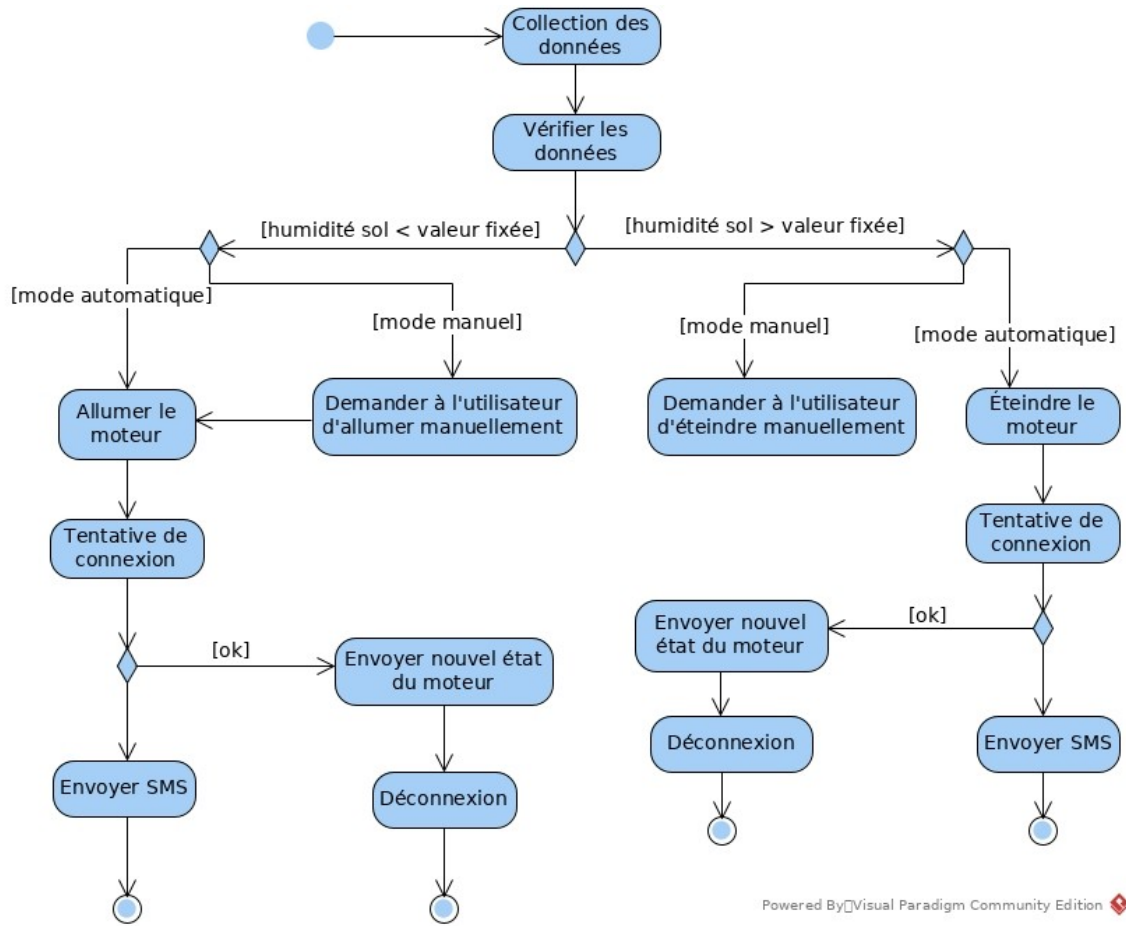


Figure 18: Diagramme d'activité du cas « Envoyer le nouvel état du moteur »

### III - Spécification et analyse des besoins du site web

#### III.1 - Spécification des besoins

##### III.1.a - Identification des acteurs

Le tableau suivant représente l'ensemble des acteurs qui interagissent avec l'application ainsi que leurs rôles.

Acteur	Description	Rôles
Administrateur	Un membre qualifié du personnel de ZIP	<ul style="list-style-type: none"> <li>x Se connecter sur la plate-forme</li> <li>x Ajouter un compte administrateur</li> <li>x Gérer les comptes utilisateurs</li> <li>x Modifier son mot de passe</li> <li>x Consulter et Modifier ses informations personnelles</li> <li>x Visualiser les données de sa plantation</li> </ul>

Acteur	Description	Rôles
Utilisateur	Une personne souhaitant bénéficier du système d'irrigation automatique	<ul style="list-style-type: none"> <li>✗ S'inscrire sur la plate-forme</li> <li>✗ Confirmer son inscription</li> <li>✗ Se connecter</li> <li>✗ Modifier son mot de passe</li> <li>✗ Consulter et Modifier ses informations personnelles</li> <li>✗ Visualiser les données de sa plantation en ligne via la plate-forme</li> <li>✗ Déclencher ou arrêter l'arrosage via la plate-forme</li> </ul>

Tableau 9: Identification des acteurs du site web

### III.1.b - Identification des fonctionnalités

Identifions maintenant les fonctionnalités de l'application.

Fonctionnalité (s)	Acteur (s)
Gérer les utilisateurs (Ajouter, Modifier, Supprimer, Activer, Désactiver)	Administrateur
Ajouter un administrateur	Administrateur
Se connecter	Administrateur, Utilisateur
S'inscrire	Utilisateur
Confirmer son inscription en cliquant sur le lien de vérification envoyé par mail	Utilisateur
Consulter ses informations personnelles	Administrateur, Utilisateur
Modifier ses informations personnelles	Administrateur, Utilisateur
Modifier son mot de passe	Administrateur, Utilisateur
Visualiser les données de sa plantation	Utilisateur
Exporter les données de sa plantation	Utilisateur
Envoyer des commandes à Arduino	Utilisateur

Tableau 10: Identification des fonctionnalités du site web

### III.1.c - Le Backlog du produit

Le backlog du produit correspond à une liste priorisée des besoins et des exigences du client. Les éléments du Backlog produit, appelés aussi des histoires utilisateurs, sont formulés en une ou deux phrases décrivant de manière claire et précise la fonctionnalité désirée par le client.

- ◆ **ID** : un nombre unique pour identifier une histoire.
- ◆ **Titre** : le résumé des **User Story**.
- ◆ **User Story** : une phrase décrivant la fonctionnalité désirée.
- ◆ **Business value** : valeur métier dirigeant la priorisation du développement des histoires utilisateurs suivant les attentes et les besoins du client, allant de 0 à 100.



- ◆ **Story point** : effort nécessaire à la réalisation d'une **User Story**. Il est estimé par l'équipe et fixe dans le temps. Un point correspond à quatre heures de charge de travail pour chaque personne.

Le tableau suivant représente les **User Stories** et leur estimation.

ID	Titre	En tant que	Je souhaite	Afin de	Business value	Story point
1	S'authentifier	administrateur	Entrer mon login et mon mot de passe	accéder à l'espace administrateur de la plate-forme	90	5
2	Gérer les utilisateurs	administrateur	Créer, Modifier, Supprimer des comptes utilisateurs et être averti par mail à chaque nouvelle inscription d'un utilisateur	Gérer les utilisateurs	90	5
3	Gérer son profil	administrateur	Modifier mon mot de passe, Consulter et Modifier mon profil	Gérer mon compte	30	2
4	Créer un compte	Utilisateur	Entrer mon nom, mon prénom, mon adresse, mon numéro de téléphone, mon mail, mon login et mon mot de passe puis recevoir un mail de confirmation.	m'inscrire à la plate-forme	90	6
5	S'authentifier	Utilisateur	Entrer mon login et mon mot de passe	accéder à l'espace utilisateur de la plate-forme	90	5
6	Gérer son profil	Utilisateur	Modifier mon mot de passe, Consulter et Modifier mon profil	Gérer mon compte	50	2
7	Consulter le tableau de bord	Utilisateur	Voir mon tableau de bord, voir l'historique des arrosages, exporter les données	Garder un œil sur mon espace vert	80	12
8	Envoyer des commandes à son système d'irrigation	Utilisateur	Cliquer sur un bouton pour déclencher ou arrêter l'arrosage ou passer en mode automatique	Contrôler à distance mon système d'irrigation	70	8

Tableau 11: Le Backlog du produit

### III.1.d - Planification des sprints

À présent que nous avons un Backlog produit complet, définissons nos sprints dans le tableau suivant.

ID	Nom du sprint	Date de début	Date de fin
Sprint 1	Réalisation des modules d'authentification et de création, de modification de compte et de gestion de profil	03/03/2019	03/04/2019
Sprint 2	Réalisation du module de gestion des comptes utilisateurs	04/04/2019	25/04/2019
Sprint 3	Réalisation du module consultation du tableau de bord	26/04/2019	30/05/2019
Sprint 4	Réalisation du module envoyer des commandes à son système d'irrigation automatique	2/06/2019	23/06/2019

Tableau 12: Les sprints du produit

### III.1.e - Description textuelle de quelques cas d'utilisation

#### III.1.e.1 - Description textuelle du cas d'utilisation « S'inscrire »

Identification	
numéro	1
nom	Inscription
Acteur	Utilisateur
Description	L'utilisateur doit créer un compte pour pouvoir accéder à l'application
Auteur	ZIP
Dates	03/03/2019
Préconditions	Être connecté à internet
Démarrage	L'utilisateur demande la page d'inscription
Scénario nominal	
1	Le système affiche le formulaire d'inscription à l'utilisateur
2	L'utilisateur remplit le formulaire et l'envoie
3	Le système envoie un mail de vérification à l'utilisateur sous forme de lien cliquable
4	L'utilisateur ouvre sa messagerie et clique sur le lien
5	Le système averti par mail les administrateurs de cette nouvelle inscription
6	Un administrateur configure et active le compte de l'utilisateur
Scénarios alternatifs	
2	L'utilisateur fait une erreur en remplissant le formulaire, le système lui envoie un message d'erreur. Le scénario nominal reprend au <b>point 1</b> .

3	L'utilisateur a fourni une adresse mail inexistante le lien de vérification n'est pas envoyé. L'utilisateur doit exécuter à nouveau le cas « <b>S'inscrire</b> ».
4	Le lien de vérification expire avant que l'utilisateur ne clique dessus, l'utilisateur doit réaliser le cas « <b>se connecter</b> ».
Scénarios d'exception	
1,2	L'utilisateur décide de quitter la page d'inscription. Le cas se termine par un échec.

Tableau 13: Description textuelle du cas d'utilisation « S'inscrire »

### III.1.e.2 - Description textuelle du cas d'utilisation «Se connecter »

Identification	
numéro	2
nom	Se connecter
Acteur	Utilisateur, Administrateur
Description	Permet de s'authentifier avant d'accéder à la plateforme.
Auteur	ZIP
Dates	03/03/2019
Préconditions	Avoir au préalable un compte.
Démarrage	L'utilisateur demande la page d'authentification
Scénario nominal	
1	Le système affiche le formulaire d'authentification.
2	L'utilisateur remplit le formulaire et l'envoie.
3	Le système vérifie que le compte est activé et que le propriétaire l'a confirmé.
4	Le système le redirige vers la page d'administration s'il s'agit d'un administrateur et vers la page d'accueil sinon.
Scénarios alternatifs	
2	L'utilisateur fait une erreur en remplissant le formulaire, le système lui envoie un message d'erreur. Le scénario nominal reprend au <b>point 1</b> .
3a	Le compte n'est pas confirmé. Le système lui renvoie un lien de vérification par mail. Si l'utilisateur clique sur <b>Oui</b> et confirme son inscription, le scénario nominal reprend au <b>point 1</b> .
3b	Le compte n'est pas activé. Le système notifie l'utilisateur que le compte n'a pas encore été activé. Il doit patienter ou contacter l'administrateur.
Scénarios d'exception	
1, 2	L'utilisateur décide de quitter la page d'inscription. Le cas se termine par un échec.

3a	Si l'utilisateur clique sur <b>Non</b> . Le cas se termine par un échec.
----	--

Tableau 14: Description textuelle du cas d'utilisation « Se connecter »

### III.1.e.3 - Description textuelle du cas d'utilisation « Confirmation d'inscription »

Identification	
numéro	3
nom	Confirmation d'inscription
Acteur	Utilisateur,
Description	Permet à l'utilisateur de confirmer son inscription.
Auteur	ZIP
Dates	03/03/2019
Préconditions	Renseigner lors de l'inscription, une adresse mail valable.
Démarrage	L'utilisateur ouvre sa messagerie et clique sur le lien de confirmation.
Scénario nominal	
1	L'utilisateur clique sur le lien de confirmation.
2	Le système vérifie bien que le lien n'a pas expiré.
3	Le système notifie l'utilisateur que son inscription a été confirmée.
Scénarios alternatifs	
2	Le lien de confirmation a expiré. Le système en informe l'utilisateur. Ce dernier pourra à nouveau recevoir un lien de vérification s'il réalise le cas « <b>Se connecter</b> ».

Tableau 15: Description textuelle du cas d'utilisation « Confirmation d'inscription »

### III.1.e.4 - Description textuelle du cas d'utilisation « exporter les données »

Identification	
numéro	4
nom	Exporter ses données
Acteur	Utilisateur
Description	Exporter en format Excel ou CSV les données mesurées par les capteurs de sa plantation
Auteur	ZIP
Dates	03/03/2019
Préconditions	Être authentifié au préalable
Démarrage	L'utilisateur clique sur le bouton exporter.
Scénario nominal	
1	L'utilisateur choisit un format (Excel ou CSV) et un intervalle de dates puis clique sur exporter.

2	Le système cherche les données comprises dans cet intervalle de dates dans la base de données.
3	Le système formate par la suite les données reçus selon le choix de l'utilisateur et les mets dans un fichier.
4	L'utilisateur enregistre le fichier.
<b>Scénarios d'exception</b>	
2	La base de données ne trouve pas de données comprises dans l'intervalle de dates choisi, le cas se termine par un échec.

Tableau 16: Description textuelle du cas d'utilisation « exporter les données »

### III.1.e.5 - Description textuelle du cas d'utilisation « Modifier son mot de passe »

Identification	
numéro	5
nom	Modifier son mot de passe
Acteur	Utilisateur, Administrateur
Description	Permet de changer le mot de passe du compte courant.
Auteur	ZIP
Dates	03/03/2019
Préconditions	Être d'abord authentifié.
Démarrage	L'utilisateur ou l'administrateur demande la page de changement de mot de passe.
Scénario nominal	
1	L'utilisateur ou l'administrateur clique sur le bouton de modification de mot de passe.
2	Le système lui demande de renseigner son adresse mail via un formulaire.
3	Le système vérifie bien que l'adresse mail est identique à celle du compte courant avant de lui afficher le formulaire de modification de mot de passe.
4	L'utilisateur renseigne le nouveau mot de passe et envoie.
5	Le système l'informe que le mot de passe a été modifié.
Scénarios alternatifs	
3	L'adresse mail renseignée par l'utilisateur est invalide ou n'est pas celle attendue. Le système lui envoie le message d'erreur correspondant. Le scénario nominal reprend au <b>point 2</b> .

Tableau 17: Description textuelle du cas d'utilisation « Modifier son mot de passe »

Clarifions tout cela avec des diagrammes de cas d'utilisation.

### III.1.f- Le diagramme de cas d'utilisation



Figure 19: Diagramme de cas d'utilisation du site web

Pour la visualisation du tableau de bord, des diagrammes représentant l'évolution des données en fonction du temps sont proposés. L'utilisateur peut choisir un intervalle de deux dates données pour visualiser les données mesurées dans cette période, c'est la réalisation du cas « Choisir intervalle de dates ».

Cependant, pour exporter les données, un intervalle de dates doit obligatoirement être spécifié.

Mises à part l'inscription et la confirmation de compte, l'utilisateur doit obligatoirement s'authentifier avant d'accéder au système.

Un administrateur ajouté doit obligatoirement confirmer son compte puis s'authentifier avant d'accéder à la plate-forme.

### III.2 - Analyse des besoins fonctionnels

Utilisons pour cela les diagrammes de séquence et d'activité des différents cas d'utilisation décrits précédemment.

### III.2.a - Analyse du cas « S'inscrire »

#### III.2.a.1 - Diagramme de séquence

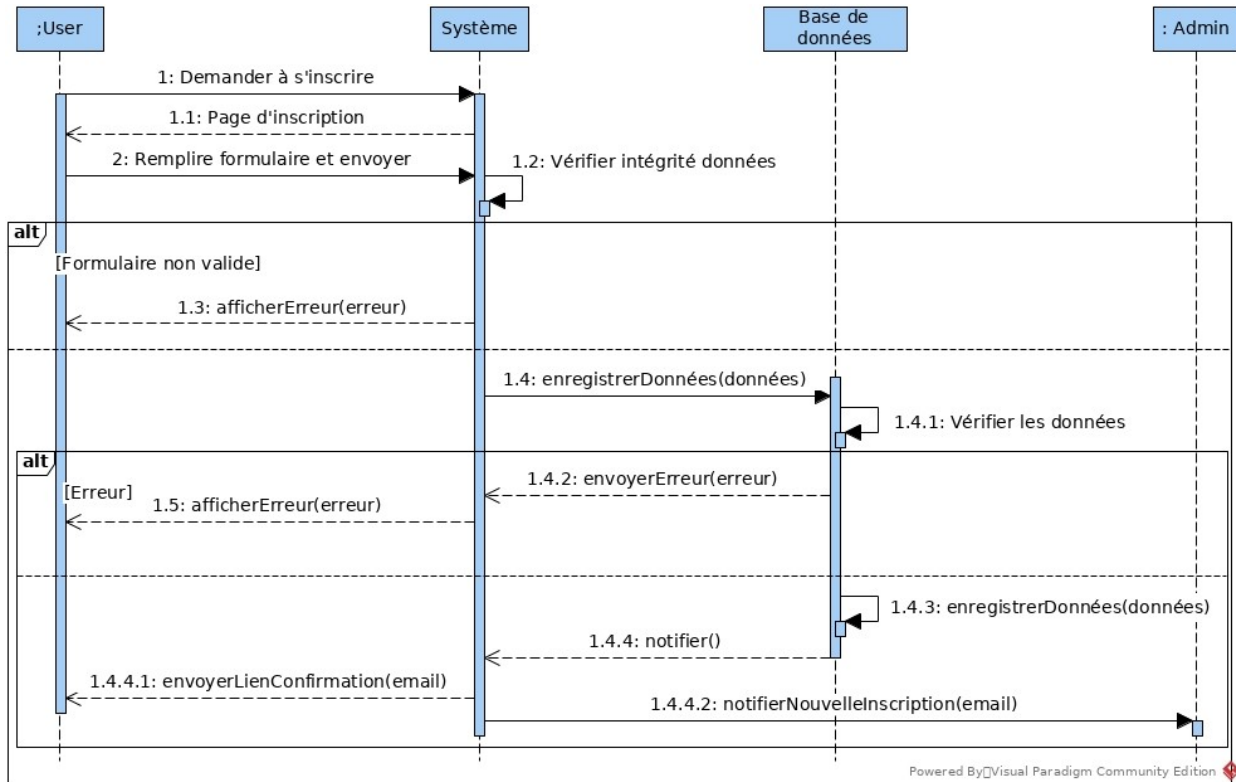


Figure 20: Diagramme de séquence du cas « S'inscrire »

Après le remplissage, le formulaire est validé (format correct de l'adresse mail, nombre de caractères du mot de passe...). Cette validation est faite coté client (Angular2) et un message d'erreur est affiché au visiteur le cas échéant.

Avant d'enregistrer le nouveau compte dans la base de données, une nouvelle vérification est effectuée, par exemple est ce que l'adresse mail donnée n'est pas déjà utilisée... Cette vérification est faite par le serveur de base de données. En cas d'erreur, un message d'erreur est envoyé au serveur d'application qui se charge de l'afficher au visiteur.

À la fin de l'inscription, un mail est envoyé aux administrateurs du système pour la finalisation.

#### III.2.a.2 - Diagramme d'activité

À travers le diagramme d'activité ci-dessous, nous pouvons voir que les actions « Envoyer lien de confirmation au visiteur » et « Informer les admins de la nouvelle inscription » sont exécutées indépendamment mais pour terminer le cas étudié, ces dernières doivent d'abord terminer leur exécution.

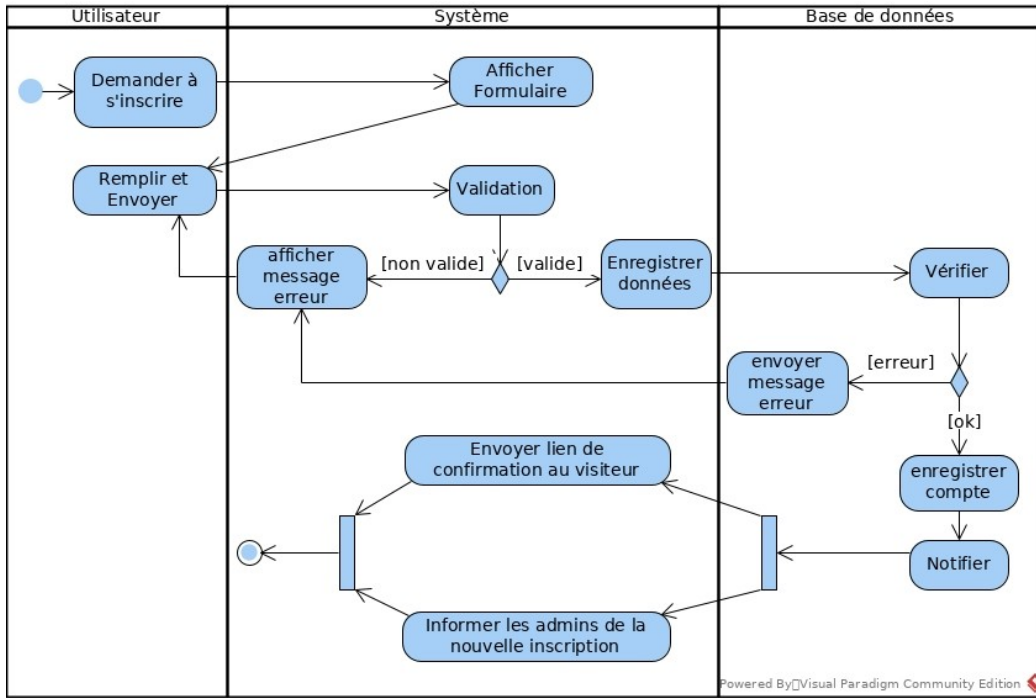


Figure 21: Diagramme d'activité du cas « S'inscrire »

### III.2.b - Analyse du cas « Se connecter »

#### III.2.b.1 - Diagramme de séquence

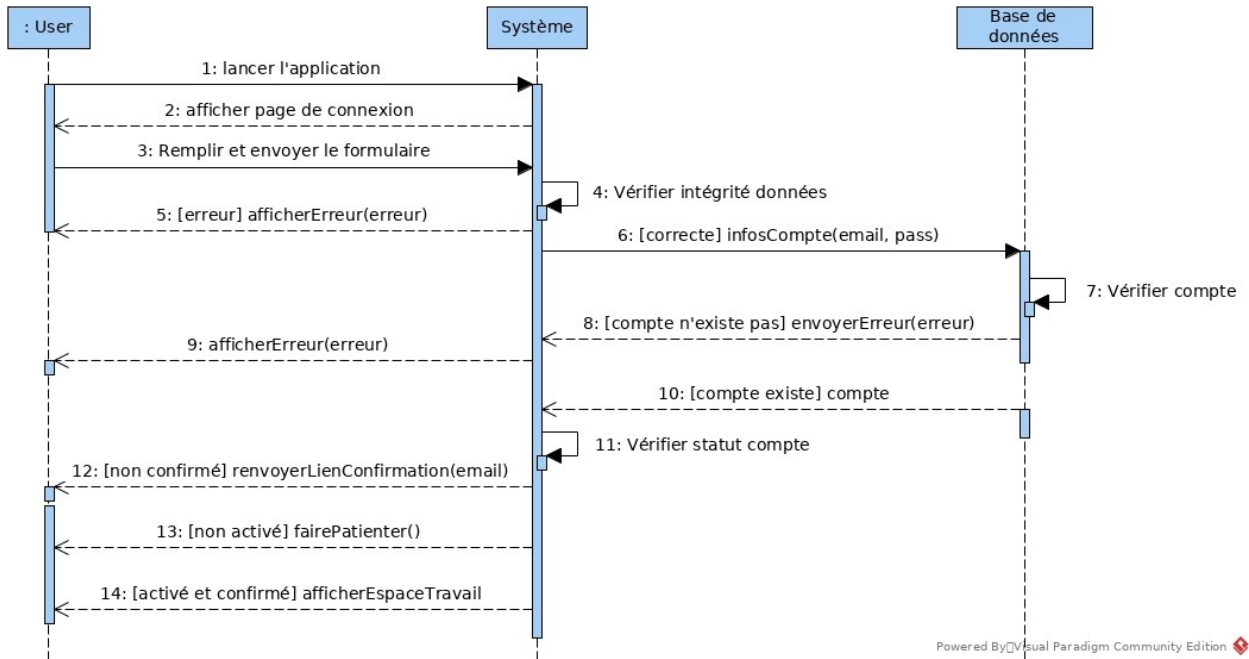


Figure 22: Diagramme de séquence du cas «Se connecter»



Le diagramme montre trois niveaux de vérification. La première, faite par Angular2 vérifie l'intégrité des données (nombre de caractères du mot de passe, format du mail), la deuxième vérifie l'existence du compte au niveau de la base de données alors que la troisième vérifie le statut du compte avant de décider de la suite.

### III.2.b.2 - Diagramme d'activité

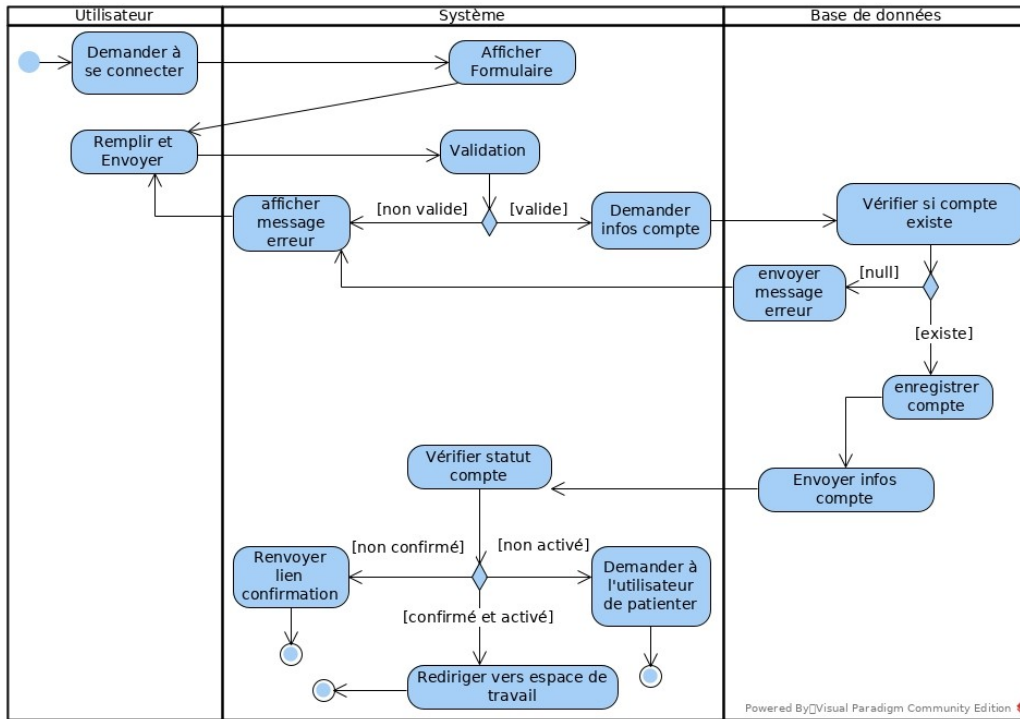


Figure 23: Diagramme d'activité du cas «Se connecter»

Le diagramme d'activité montre trois possibilités de terminer le cas à l'issue de la vérification du statut du compte.

Si le compte n'est pas activé, on demande au visiteur de patienter le temps qu'un administrateur finalise l'inscription (préparer et configurer le matériel d'arrosage).

### III.2.c - Analyse du cas « Exporter ses données »

#### III.2.c.1 - Diagramme de séquence

Avant d'exporter les données, l'utilisateur doit au préalable choisir un intervalle de dates et un format de fichier.

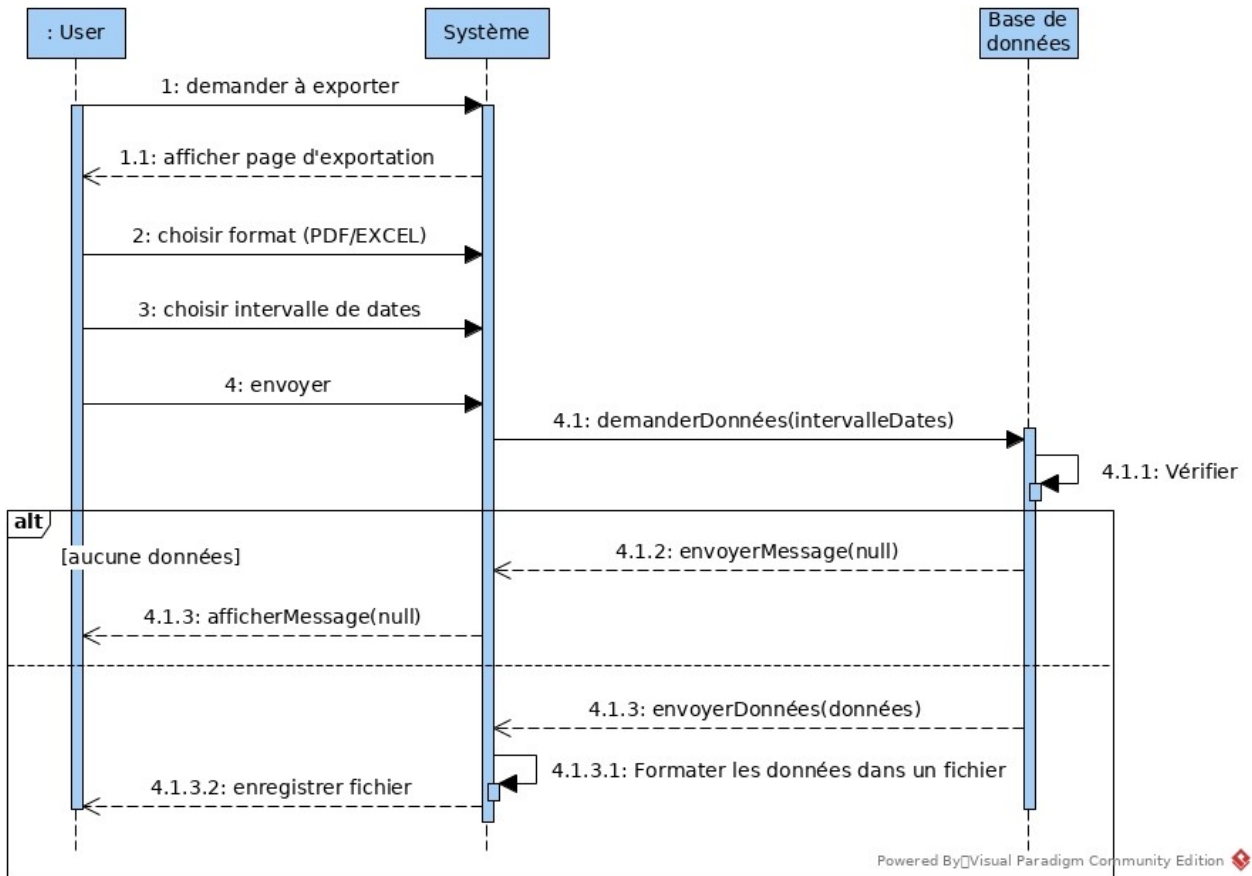


Figure 24: Diagramme de séquence du cas «Se connecter»

### III.2.c.2 - Diagramme d'activité

Au niveau du diagramme d'activité nous voyons bien que les actions « choisir format » et « choisir intervalle de dates » sont indépendantes. Cependant, avant d'exporter les données, c'est deux étapes sont indispensables.

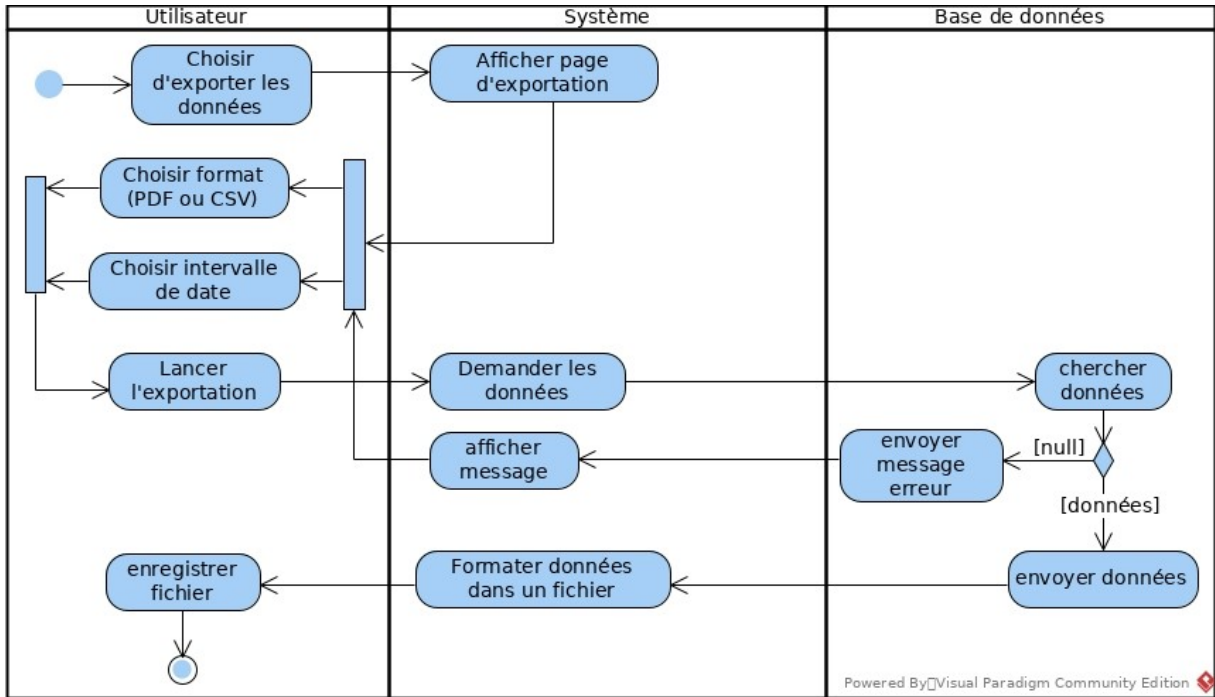


Figure 25: Diagramme d'activité du cas «Se connecter»

### III.2.d - Analyse du cas « Modifier son mot de passe »

#### III.2.d.1 - Diagramme de séquence

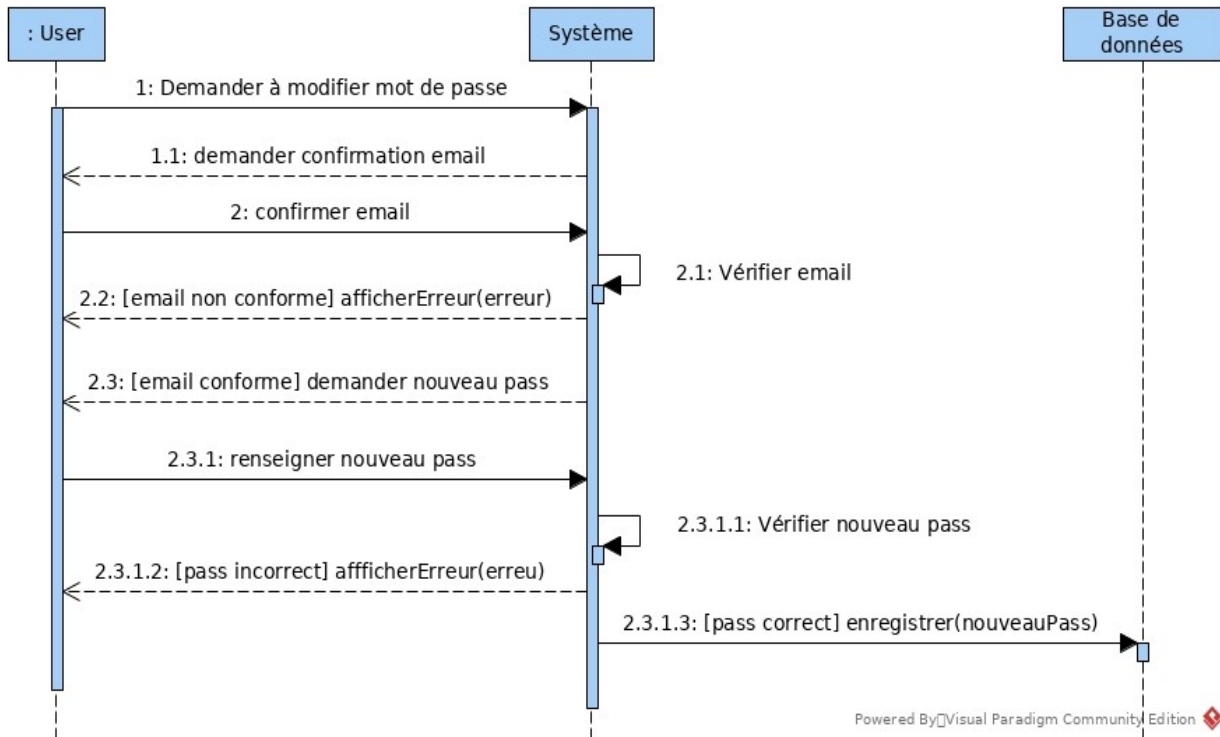


Figure 26: Diagramme de séquence du cas « Modifier son mot de passe »

Pour des raisons de sécurité, Firebase gère la modification du mot de passe indépendamment des autres informations du compte.

### III.2.d.2 - Diagramme d'activité

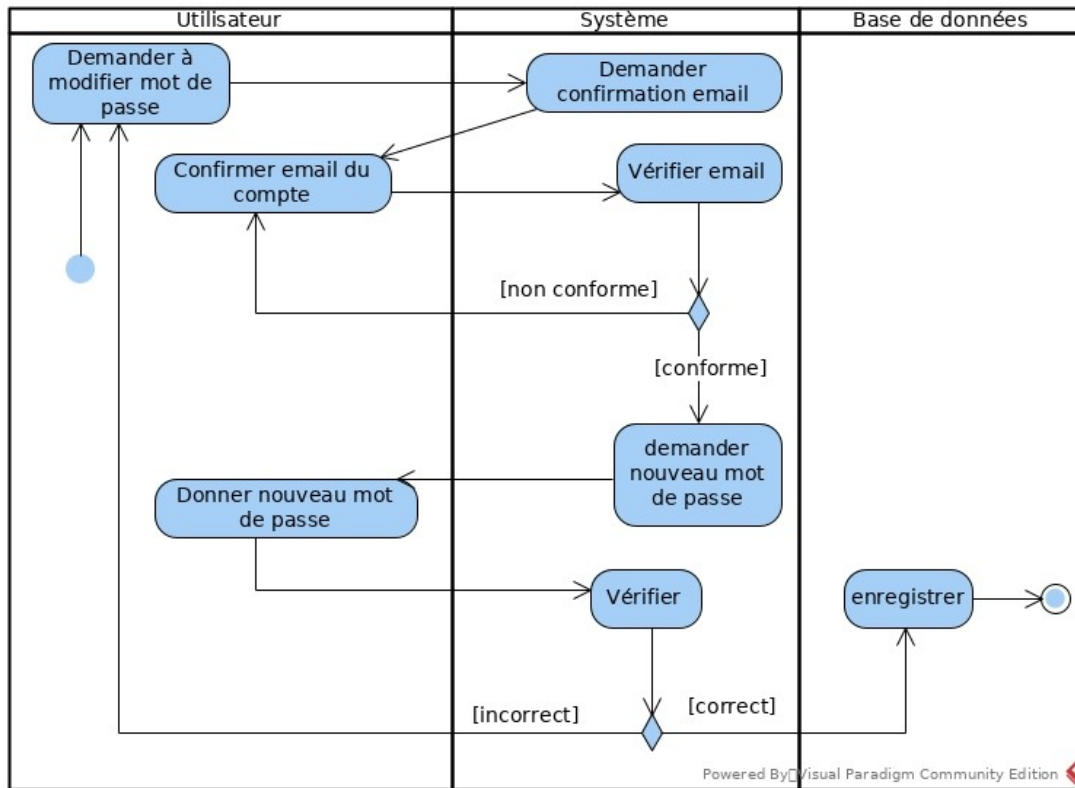


Figure 27: Diagramme d'activité du cas « Modifier son mot de passe »

## Conclusion

Ce chapitre est d'une importance capitale avant la phase de conception de tout système informatique. Il nous a aidé à éclaircir les besoins exprimés par le client et à définir les différentes étapes à suivre pour les réaliser. À présent, nous pouvons passer à la conception du système dans le chapitre suivant.

# TROISIÈME PARTIE : CONCEPTION, IMPLÉMENTATION ET PRÉSENTATION D'AGRODIGITAL

## Sommaire

CHAPITRE V : ARCHITECTURE ET CONCEPTION D'AGRODIGITAL.....	45
I - La conception générale.....	45
II - La conception détaillée du site web.....	52
CHAPITRE VI : IMPLÉMENTATION ET PRÉSENTATION D'AGRODIGITAL.....	54
I - Les outils mis à ma disposition.....	54
II - Les tâches périphériques.....	54
III - Difficultés rencontrées et solutions apportées.....	55
IV - Compétences acquises.....	55
V - Présentation de l'API Firebase.....	55
VI - Implémentation et présentation de l'application avec Arduino.....	57
VII - Implémentation et présentation de la plate-forme web.....	66

# CHAPITRE V : ARCHITECTURE ET CONCEPTION D'AGRODIGITAL

Cette phase est incontournable lors de la réalisation d'un projet informatique. Elle décrit les différents éléments qui composent le système, leurs relations et interactions ainsi que la manière dont les données vont être organisées et traitées.

Elle peut être scindée en deux parties à savoir la conception générale et la conception détaillée.

Ce chapitre sera donc structuré comme suit : d'abord nous présentons la conception générale puis la conception détaillée.

## I - La conception générale

Dans cette partie, nous allons présenter l'architecture, le diagramme de paquetage et de déploiement de notre système. Cela nous permettra d'avoir une vue d'ensemble du système et de son fonctionnement.

### I.1 - Architecture générale du système de déploiement

L'image suivante, figure 28, représente l'architecture du système d'irrigation une fois déployé. Nous pouvons distinguer ses différents composants annotés en couleur bleue.

- ◆ Nous avons d'abord le **dispositif** contenu dans une boîte blanche, nous y trouvons la carte Arduino Mega, le module SIM800L qui permet de se connecter à internet et d'envoyer des SMS. C'est dans cette boîte que se trouve aussi le branchement des capteurs à la carte Arduino. Pour faire sortir les fils des différents capteurs, nous avons percé la boîte de petits trous.
- ◆ Nous avons ensuite le **panneau solaire** qui permet d'alimenter le dispositif en électricité ainsi que la **pompe** qui doit être immergée dans le **réservoir d'eau**. Nous utilisons un simple seau avec un couvercle pour contenir l'eau d'arrosage.
- ◆ Les **capteurs** sont encadrés en rouge :
  - ✓ **DHT11** est le capteur d'humidité et de température de l'air. Il est pendu à son fil de connexion à la carte Arduino et exposé à l'air libre,
  - ✓ **CH** est le capteur de l'humidité du sol qui est directement enfoncé dans le sol.
- ◆ Nous voyons aussi que le dispositif d'arrosage est suivi et contrôlé à distance via **internet** à l'aide du **tableau de bord** qui peut déclencher ou arrêter l'arrosage en envoyant des commandes au dispositif.
- ◆ Si besoin est (problème technique, niveau de l'eau critique...), le dispositif envoie un **SMS** à l'utilisateur sur son numéro de **téléphone** pour l'informer.
- ◆ Nous pouvons voir aussi le **tuyau d'arrosage** que nous avons percé de petits trous pour tester et simuler un système d'irrigation au goutte-à-goutte.

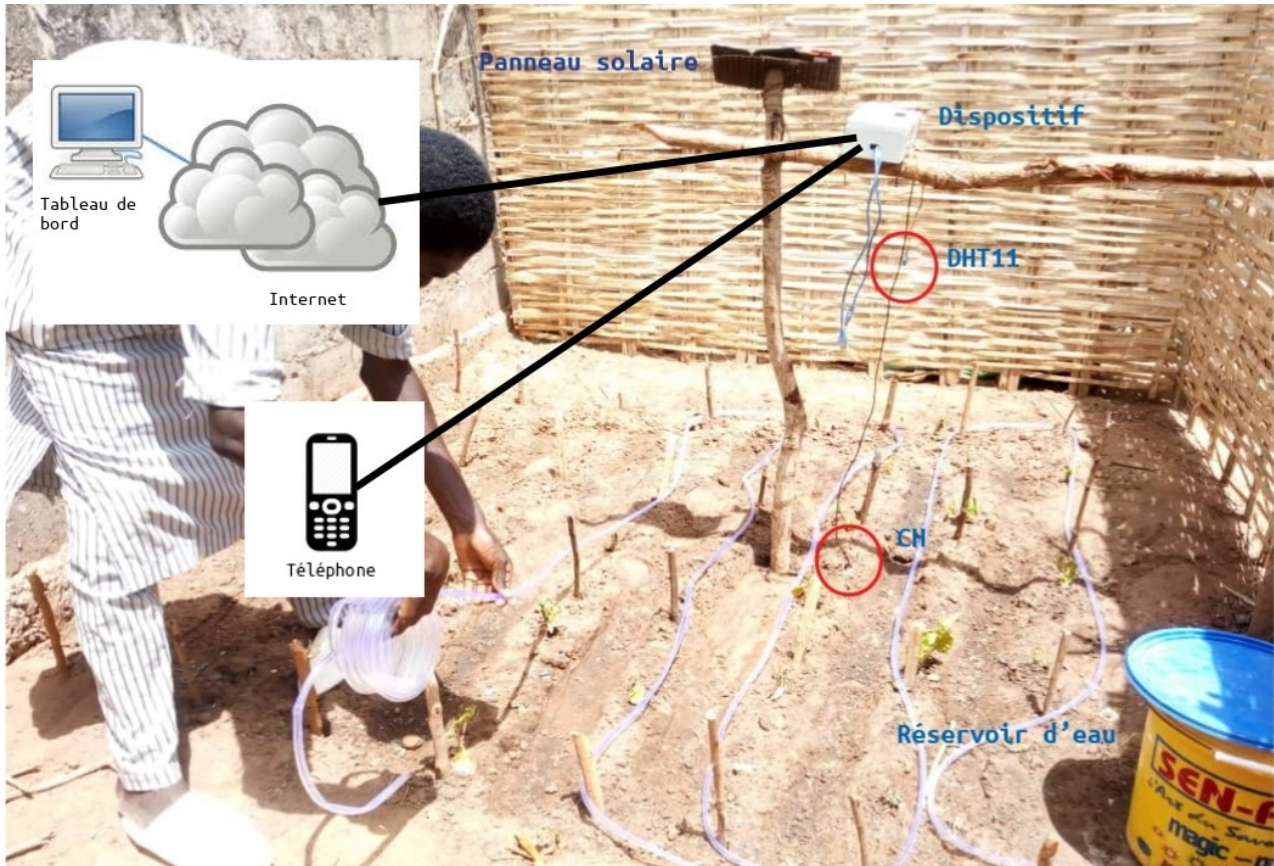


Figure 28: Architecture générale de déploiement d'AgroDigital

Proposons maintenant un schéma récapitulatif le système pour avoir une vue d'ensemble de ses différents composants et de leurs interconnexions.

## I.2 - Schéma récapitulatif du système

À travers ce schéma, figure 29 suivante, nous voyons clairement que le dispositif d'arrosage et la plate-forme web ont en commun la base de données Cloud Firestore et les fonctions cloud de Firebase. Les composants Cloud Functions et Cloud Firestore de l'API Firebase seront expliqués en détails dans le chapitre suivant. Pour le moment retenons juste que Cloud Functions représente des fonctions JavaScript exécutées côté serveur (environnement Node.js) permettant à Arduino d'accéder à la base de données Cloud Firestore. Ces fonctions sont aussi utilisées par le site web pour la gestion des comptes utilisateur.

La **pompe** immergée dans le **réservoir** d'eau est connectée à Arduino par l'intermédiaire d'un module **relais**.

Le module **SIM800L** est directement connecté à Arduino. Ce dernier le contrôle à l'aide de commandes AT qui permettent le contrôle total du module. SIM800L est utilisé pour envoyer les SMS à l'utilisateur mais aussi pour envoyer les données et recevoir les commandes envoyées par l'utilisateur sur internet.

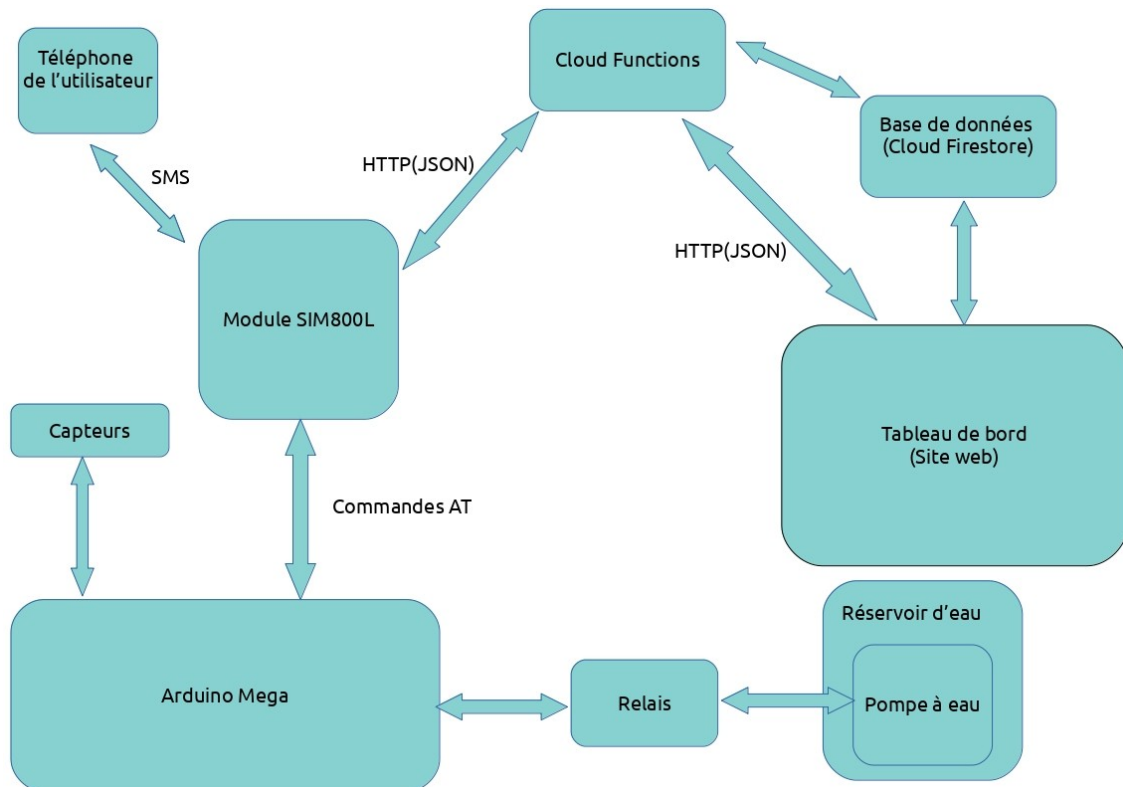


Figure 29: Schéma récapitulatif du système

### I.3 - Architecture d'AgroDigital

L'architecture d'un système informatique décrit comment ce lui-ci doit être conçu de manière à répondre aux spécifications fonctionnelles du client.

Il existe plusieurs types d'architecture, cependant les plus connues sont celles basées sur le principe de client-serveur en l'occurrence : l'architecture 2-tiers, l'architecture 3-tiers et l'architecture N-tiers.

En ce qui nous concerne, nous allons utiliser l'architecture 3-tiers pour la réalisation de notre projet.

#### I.3.a - Architecture de l'application avec Arduino

Du côté d'Arduino, nous utilisons bien une architecture trois tiers bien que la couche cliente ne se préoccupe pas de la présentation. Nous avons donc trois couches :

- ◆ **la couche cliente (présentation)**: après avoir collecté les données, **Arduino** utilise le module **SIM800L** pour envoyer ces données à la couche de traitement,
- ◆ **la couche de traitement** : constituée par les fonctions de cloud (**Firestore Cloud Functions**), cette couche est chargée de formater en JSON les données reçues d'y ajouter le champ de la date et de les enregistrer au niveau de la base de données,
- ◆ **la couche de stockage** : il s'agit de **Cloud Firestore** qui se charge de stocker les données.
- ◆ **Commande AT** : ce sont des commandes qui permettent la gestion complète du module SIM800L par Arduino.



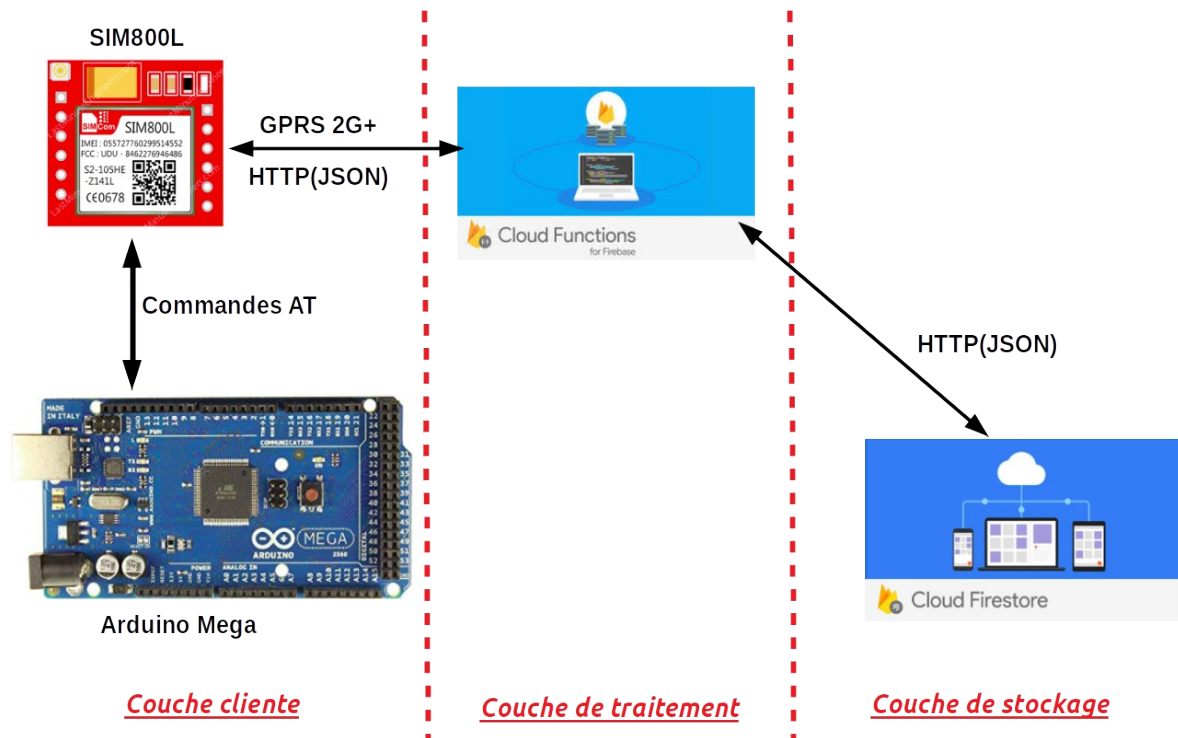


Figure 30: Architecture de l'application avec Arduino

### ***1.3.b - Architecture du site web***

Afin d'appliquer l'architecture trois tiers à notre application web, nous utiliserons le modèle d'architecture MVC.

#### ***1.3.b.1 - Qu'est-ce que le modèle d'architecture MVC ?***

Le MVC « modèle vue contrôleur » est une architecture de développement visant à séparer le code source en modules.

En effet, ce modèle très répandu, consiste à séparer distinctement l'accès aux données (bases de données), la vue affichée à l'utilisateur et la logique métier.

Cette architecture est le plus communément retrouvée au sein d'applications web mais existe également au niveau des applications lourdes.

Ainsi, comme nous pouvons le voir, il y a trois couches distinctes :

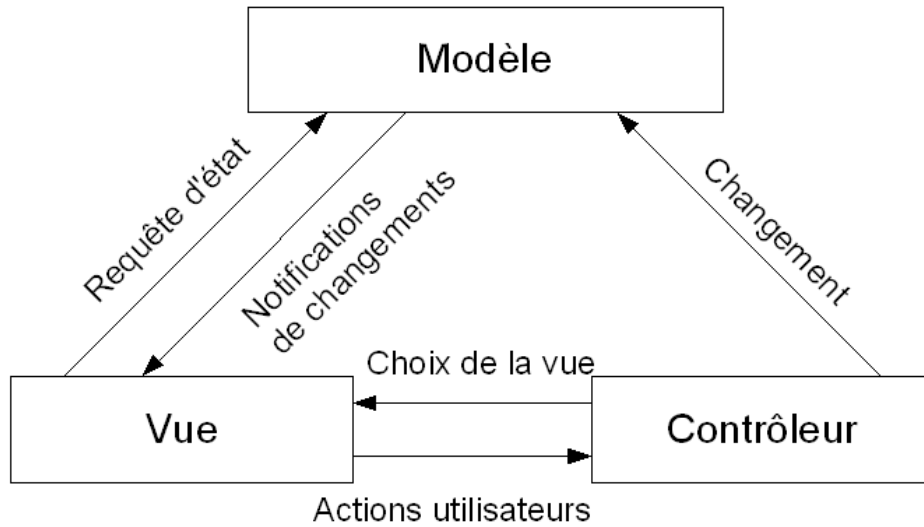


Figure 31: Exemple du modèle d'architecture MVC[23]

◆ Le modèle

Le modèle définit les données utilisées par l'application. En effet, c'est ici que le lien se fera entre notre application et la base de données.

Par exemple, on pourrait trouver les utilisateurs ou encore les différents articles pour un site de ventes en ligne.

Ces données pourront être mises à jour dans le contrôleur et affichées au niveau de la vue.

◆ La vue

La vue définit la façon dont les informations seront affichées à l'écran (via des composants par exemple). Il s'agit de l'interface utilisateur.

C'est ici qu'on utilisera les données récupérées par le modèle afin de les présenter à l'utilisateur.

Par exemple, pour un site de ventes en ligne, ce serait la page du produit qui s'affiche à l'écran.

◆ Le contrôleur

Dans le contrôleur, nous retrouvons toute la logique métier. En effet, lorsque l'utilisateur interagit avec la vue, la requête est traitée par le contrôleur. Il fonctionne comme un « listener », c'est-à-dire qu'il attend que l'utilisateur interagisse avec la vue pour en récupérer la requête.

Ainsi, c'est le contrôleur qui définira la logique d'affichage, et affichera la vue suivante à l'écran[23].

### I.3.b.2 - Architecture du système du site web

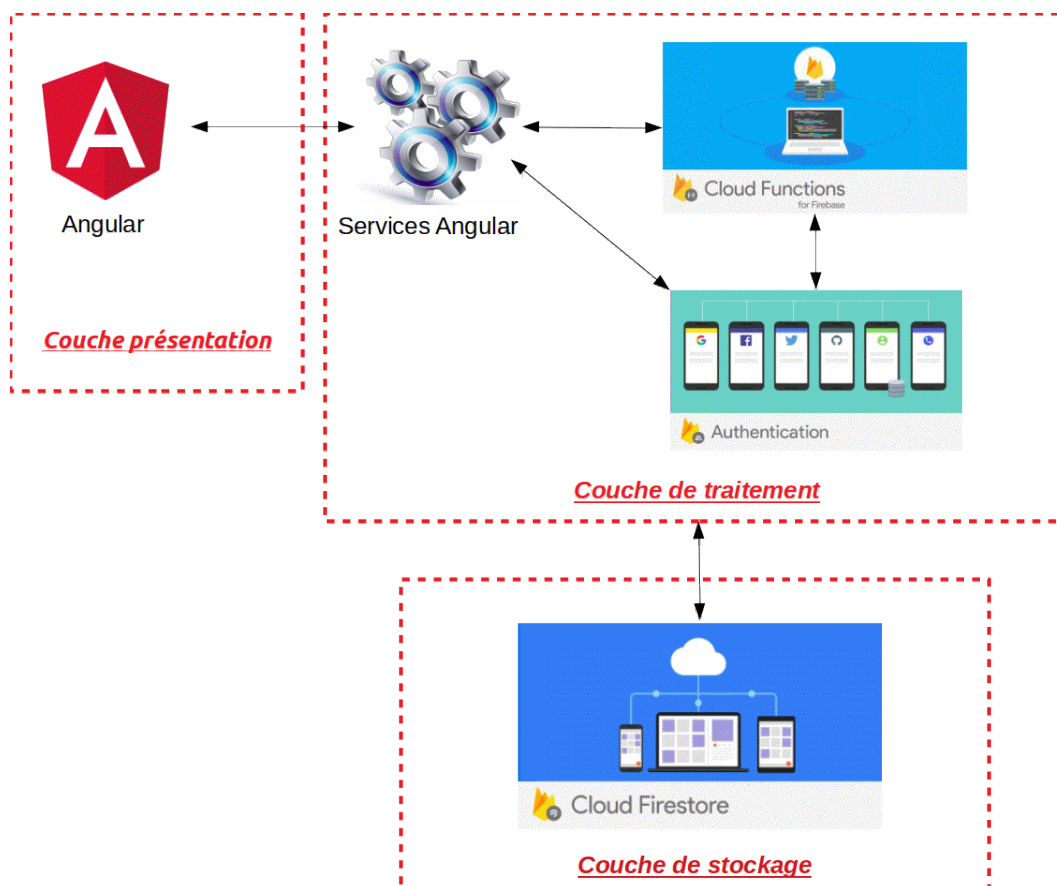


Figure 32: Architecture du système du site web

### I.4 - Diagramme de paquetage du site web

Un diagramme de paquetage est un diagramme UML qui fournit une représentation graphique de haut niveau de l'organisation de l'application, et aide à identifier les liens de généralisation et de dépendance entre les paquetages.

Notre système compte huit paquetages dont les principaux sont :

- ◆ **admin** : il gère les comptes utilisateurs du système,
- ◆ **account**: il gère l'inscription, l'authentification, le profil des utilisateurs et d'autres fonctionnalités relatives au compte (vérification de mail, changement de mot de passe...),
- ◆ **dashboard** : il contient tout ce qui compose le tableau de bord (les graphes et l'exportation de données),
- ◆ **guards**: permet de sécuriser l'accès aux différents URLs du site selon les droits accordés au compte courant,
- ◆ **services** : il contient les services utilisés par les autres paquetages comme l'accès à l'API Firebase (Base de données, Authentification...),
- ◆ **models** : il contient les classes de l'application utilisées surtout par le paquetage **services**,

- ◆ **config** : contient les fichiers de configuration du site,
- ◆ **outils** : contient un ensemble d'outils utilisés par les autres paquetages.

Le diagramme suivant montre les relations qui existent entre ces paquetages.

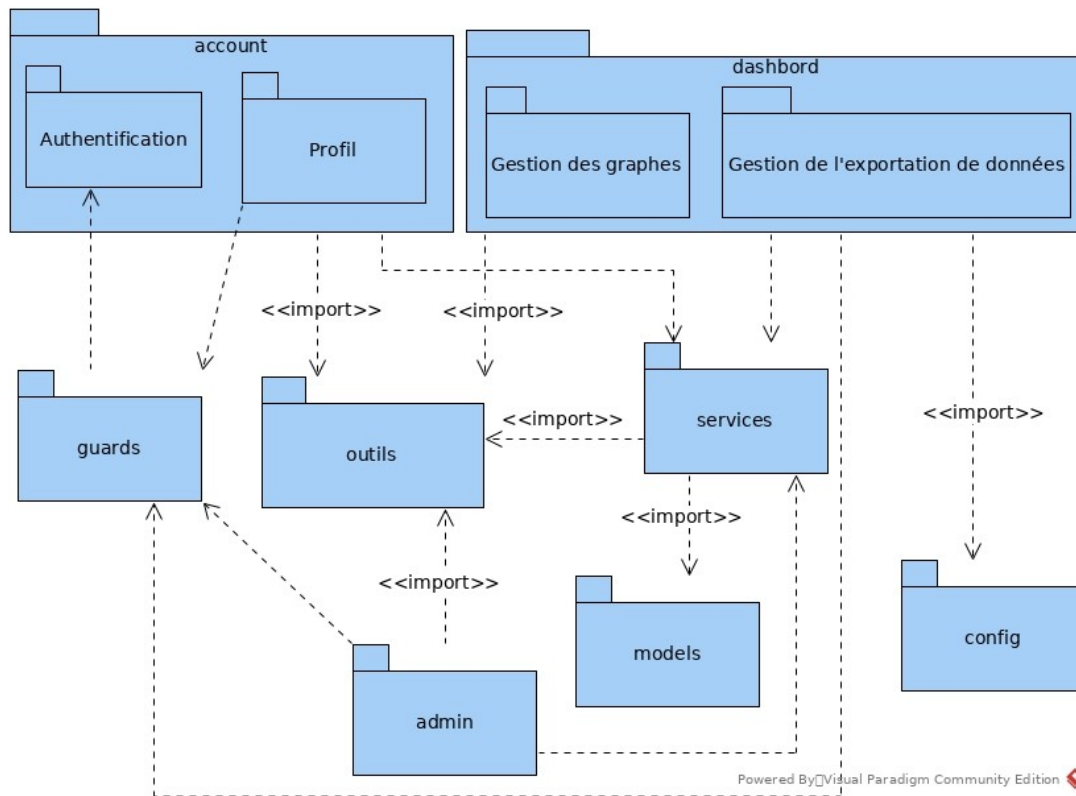


Figure 33: Diagramme de paquetage du site web

## I.5 - Diagramme de déploiement du système

Le diagramme de déploiement fait partie des diagrammes structuraux (statique) d'UML, il représente la disposition physique des ressources matérielles qui constituent le système. Il montre ainsi la répartition des composants (élément logiciels) sur ces matériels et la nature des connexions de communication entre ces différentes ressources matérielles.

- ✓ Le serveur **Node.js** se charge d'afficher les pages d'**Angular** au visiteur sur un navigateur web.
- ✓ L'**application web** envoie les requêtes de l'utilisateur et lui retourne le résultat.
- ✓ Le **serveur de base de données** stocke les données et gère les droits d'accès à ces dernières.

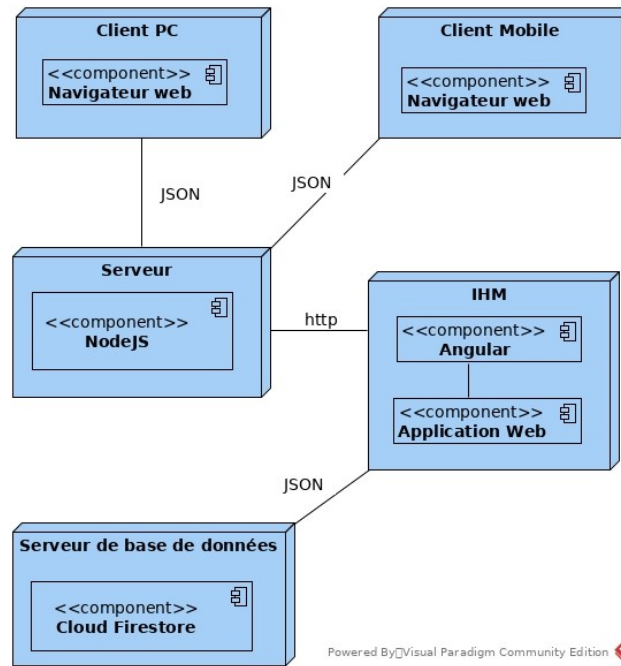


Figure 34: Diagramme de déploiement du site web

## II - La conception détaillée du site web

La phase de conception détaillée est la dernière phase nous permettant de voir une représentation statique du système à mettre en place.

Ici, il sera question d'entrer plus en détail dans la conception du système. Explorons pour cela, avec un diagramme de classe, le contenu du paquetage **Models** présenté précédemment. Ce paquetage contient les données utilisées par notre application ainsi que leur organisation.

Cependant avant de passer au diagramme de classe, faisons une petite présentation de ces dites classes.

### II.1 - Présentation des classes

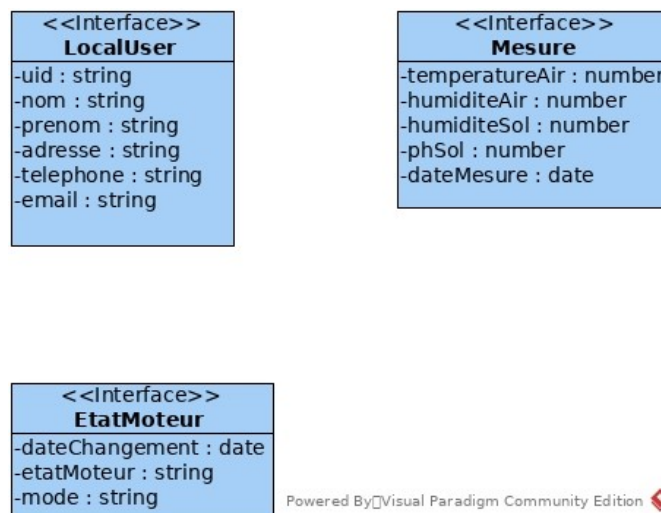


Figure 35: Présentation des interfaces

Comme nous n'avons pas besoins d'opérations au sein de nos classes, nous utilisons juste des interfaces pour créer des types des données.

- ◆ l'interface **LocalUser** : regroupe toutes les informations qui identifient et caractérisent un compte utilisateur du système,
- ◆ l'interface **Mesure** : elle regroupe toutes les données mesurées par les capteurs,
- ◆ **EtatMoteur** : qui permet de savoir à quelle date la pompe d'eau a été allumée ou éteinte et comment (manuellement ou automatiquement) grâce à l'attribut mode.

## II.2 - Le diagramme de classe

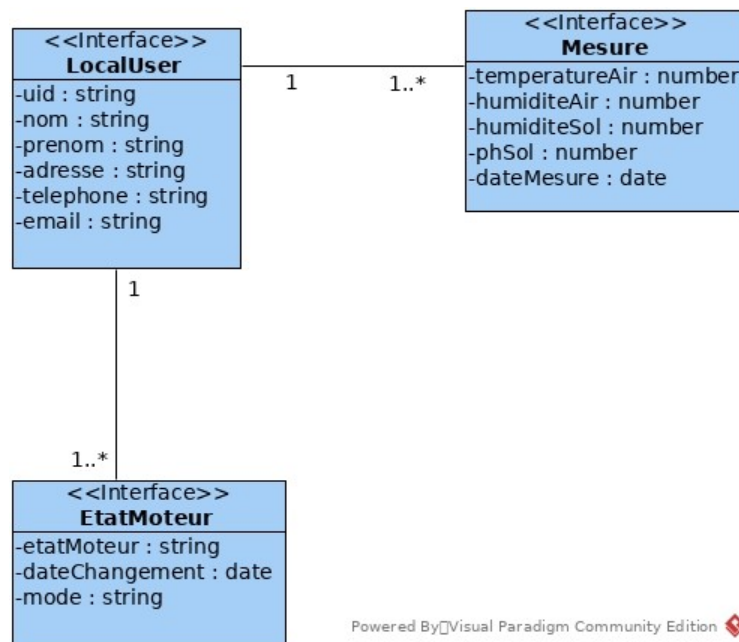


Figure 36: Diagramme de classe du site web

## Conclusion

Dans ce chapitre, nous avons choisi un modèle d'architecture de notre système pour une meilleure organisation de notre code source. Nous avons ensuite défini l'architecture physique ainsi que la manière dont les données seront représentées et traitées à l'aide d'un diagramme de classe. Il ne nous reste plus qu'à l'implémenter avant de le présenter, ce que nous allons voir dans le chapitre suivant.

# CHAPITRE VI : IMPLÉMENTATION ET PRÉSENTATION D'AGRODIGITAL

Dans ce chapitre, il sera question d'expliquer les grands points de la réalisation de notre système. Cette phase est la concrétisation des phases précédentes par le billet de l'outil informatique.

La réalisation d'un système informatique exige cependant un choix minutieux des technologies à utiliser, des langages de programmation, des types de serveurs... Ainsi, pour détailler notre implémentation, nous essayerons de faire le tour des outils, matériels et technologies utilisés avant de présenter les principales fonctionnalités de notre système en commençant par l'application avec Arduino suivi de la plateforme web. Cependant, avant tout cela, parlons un peu du stage.

## I - Les outils mis à ma disposition

J'ai effectué le stage au département informatique de ZIP où je disposais :

- ◆ d'une salle contenant les composants électroniques nécessaires au projet (Carte Arduino, module SIM800L, pompe à eau, fils de connexion, fer à souder...),
- ◆ d'un panneau solaire de 6volts pour alimenter le système,
- ◆ d'un tuyau d'arrosage pour les tests,
- ◆ de connexion internet,
- ◆ d'un bureau.

## II - Les tâches périphériques

### II.1 - Concernant la carte Arduino

- ◆ J'ai commencé par une documentation et des TP simples sur Arduino et certains de ses composants pour me familiariser avec cette technologie,
- ◆ puis j'ai testé l'envoi et la réception de message SMS,
- ◆ enfin, j'ai réalisé un TP sur l'envoi des données sur internet en utilisant l'API Firebase proposée par Google et qui sera expliquée un peu plus tard dans ce document.

### II.2 - Concernant le site Web

- ◆ Tout d'abord je me suis auto-formé en Angular et j'ai fait pas mal de TP sur le sujet,
- ◆ en suite vient la documentation sur comment intégrer Firestore et Angular dans un projet Web,
- ◆ enfin, puisque le site doit afficher des graphes, je me suis documenté et réalisé des TP sur la bibliothèque JavaScript ChartsJS qui dessine des graphes.

### **III - Difficultés rencontrées et solutions apportées**

Au tout début j'avais choisi de travailler avec le Framework Spring Boot pour le Back End car pour le côté Front End ZIP m'a imposé Angular. Après quelques semaines, je devais chercher un hébergement adéquat pour la plate-forme web. Après quelques jours de recherches je suis tombé sur l'API Firebase qui utilise une base de données NoSQL qu'on peut synchroniser à temps réel. De plus Firebase propose plusieurs SDK dont JavaScript avec NodeJs, ce qui me permettait d'utiliser le même langage qu'Angular qui est conçu en JavaScript. J'ai donc jeté mon dévolu sur Firebase au prix d'un retard considérable sur le temps imparti.

Je peux dire que toutes les technologies que j'ai utilisées au cours de ce stage sont nouvelles pour moi. Je me suis donc auto-formé en chacune d'elles et ce n'était pas facile, car la plupart des documentations officielles sont en anglais. Mais j'ai essayé de traduire avec Google Traduction et de regarder des vidéos sur le sujet pour comprendre. La documentation m'a pris énormément de temps.

La dernière difficulté concerne l'envoi des données sur internet. En effet, quand j'ai testé l'envoi des données avec seulement une carte Arduino et le module SiM800L, tout marchait à la perfection. Mais quand j'ai intégré les autres composants comme les capteurs, la pompe à eau et les composants de connexion, le module SIM800L n'avait plus assez de tension pour acheminer les données vers internet. La solution a été de brancher directement la borne positive du module à la broche 5volts de la carte Arduino au lieu de passer par la diode que j'avais installée au début pour éviter que le module ne soit endommagé par une surtension.

### **IV - Compétences acquises**

Ce stage m'a été très bénéfique, car il m'a offert l'occasion d'apprendre des technologies nouvelles qui sont très utilisées actuellement en milieu informatique.

Pour être plus précis, j'ai maintenant des connaissances solides en Angular ainsi qu'en Firebase sans oublier Arduino.

C'était aussi l'occasion d'approfondir mes connaissances en JavaScript, HTML et CSS et d'apprendre à utiliser un ensemble de bibliothèques basées sur JavaScript.

J'ai même appris à souder et à concevoir des circuits électriques fonctionnels.

Cependant notre système dépend beaucoup de l'API Firebase de Google tant du côté d'Arduino que du côté du site web, aussi nous allons faire une petite présentation de cette API avant d'entrer en profondeur dans l'implémentation.

### **V - Présentation de l'API Firebase**

#### **V.1 - Tout d'abord pourquoi Firebase ?**

Nous avons quatre bonnes raisons d'utiliser Firebase :

- ◆ la première est que notre plate-forme est conçue avec le Framework Angular conformément aux exigences. Ce dernier étant utilisé coté client, nous avons besoins d'accéder à une API. La conception d'une API n'est cependant pas une tâche aisée, nous avons donc opté pour celle proposée par Firebase,



- ◆ la deuxième raison est liée à la base de données NoSQL de Fireabse. En effet, les capteurs envoient en permanence des données. La base de données sera donc très probablement partitionnée dans un futur proche à cause de cette quantité importante de données. Contrairement aux bases de données relationnelles, les bases de données NoSQL sont faciles à partitionner et à distribuer sur différents serveurs,
- ◆ quant à la troisième raison, elle concerne l'accès aux données. Firebase propose une base de données à temps réel, ce qui est très approprié pour la conception d'un tableau de bord qui affiche les données à temps réel,
- ◆ enfin, la quatrième raison est que cette API est sécurisée, facile à utiliser, très répandue et bien documentée.

## V.2 - Présentation de Firebase

Firebase est un ensemble de services d'hébergement pour n'importe quel type d'application (Android, iOS, Javascript, Node.js, Java, Unity, PHP, C++...). Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale (Google, Facebook, Twitter et Github), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel. Lancé en 2011 sous le nom d'Envolv, par Andrew Lee et par James Templin, le service est racheté par Google en 2014. Il appartient aujourd'hui à la maison mère de Google: Alphabet[24].

L'objectif premier de Firebase est de nous libérer **de la complexité de création et de la maintenance d'**une architecture serveur, tout en nous garantissant une scalabilité à toute épreuve (plusieurs milliards d'utilisateurs) et une simplicité dans l'utilisation.

Pour cela, Firebase à été décomposé en plusieurs produits très riches parmi lesquels nous allons juste présenter ici ceux utilisés dans notre projet :

## V.3 - Cloud Firestore

Avec Cloud Firestore, nous pouvons stocker et synchroniser les données entre les utilisateurs et les appareils à l'échelle mondiale grâce à une base de données NoSQL hébergée dans le cloud. Cloud Firestore nous offre une synchronisation en direct et une assistance hors ligne ainsi que des requêtes de données efficaces[25].

La base de données est structurée sous forme d'une ou de plusieurs collections contenant chacune au moins un document dans lequel les données sont stockées.

L'image suivante montre cette structure :

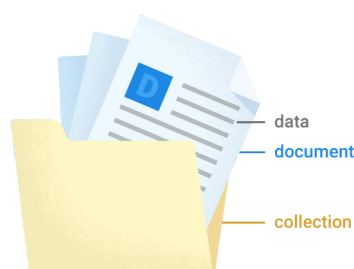


Figure 37: Exemple de collection de Firestore

Cependant, une collection peut contenir plusieurs documents mais ne peut pas contenir d'autres collections. Un document peut contenir une ou plusieurs collections mais ne peut pas contenir d'autres documents.

## V.4 - Cloud Functions

Les fonctions cloud sont des fonctions JavaScript à usage unique qui sont exécutées dans un environnement Node.js sécurisé et géré. Elles ne sont exécutées que lorsqu'un événement spécifique surveillé est émis.

Nous pouvons donc créer des fonctions déclenchées par les produits Firebase, telles que les modifications des données dans la base de données en temps réel, les nouvelles inscriptions d'utilisateurs...

Les fonctions cloud nous permettent d'étendre notre application avec du code backend personnalisé sans avoir à gérer et faire évoluer nos propres serveurs[25].

## V.5 - Authentification (Firebase Auth)

Ce produit nous aide à gérer nos utilisateurs de manière simple et sécurisée. Il propose plusieurs méthodes d'authentification, notamment le courrier électronique et le mot de passe, des fournisseurs tiers tels que Google ou Facebook, et l'utilisation directe de votre système de compte existant. Nous pouvons créer notre propre interface ou profiter de l'interface utilisateur open source entièrement personnalisable proposée par Firebase[25].

## V.6 - Hosting

Hosting permet de simplifier l'hébergement Web avec des outils spécialement conçus pour les applications Web modernes. Lorsque nos ressources Web sont téléchargées, Firebase les transfère automatiquement vers son CDN mondial et lui donne un certificat SSL gratuit afin que nos utilisateurs bénéficient d'une expérience sécurisée, fiable et à faible latence, peu importe où ils se trouvent[25].

# VI - Implémentation et présentation de l'application avec Arduino

## VI.1 - Matériels utilisés

### VI.1.a - La carte électronique utilisée

Pour réaliser ce projet, nous avons besoin d'une carte électronique programmable, il en existe plusieurs (Arduino, Raspberry Pi, BeagleBone, STM nucleo...). Ces cartes sont très utilisées pour développer des projets **DIY (Do It Yourself)** qui signifie (faites le vous-même).

Nous avons choisi la carte Arduino parce qu'elle est économique, conviviale, très populaire et est focalisée sur la simplicité d'utilisation et de programmation.

La carte Arduino est programmable en une variante du langage de programmation C++ qui nous est familier avec un IDE gratuit appelé Arduino IDE.

Elle possède une communauté très active, une très bonne documentation et un grand nombre d'exemples de projets réalisés téléchargeables sur internet.

Cependant, il existe plusieurs cartes Arduino avec différentes configurations et caractéristiques de puissance. Nous avons choisi d'utiliser le modèle **Mega** parce qu'elle offre plus de mémoire, de puissance de calcul et d'entrées sorties.

L'**Arduino Mega** est le modèle le plus perfectionné et puissant de la célèbre carte électronique. Elle permet d'effectuer un maximum d'actions et délivre un potentiel tel, qu'il est possible de se pencher sur les montages les plus lourds et gourmands en code. Cette carte peut être utilisée par les amateurs confirmés, mais est principalement destinée aux experts qui pourront en faire un usage plus professionnel[26].

### **VI.1.b - La carte Arduino Méga**

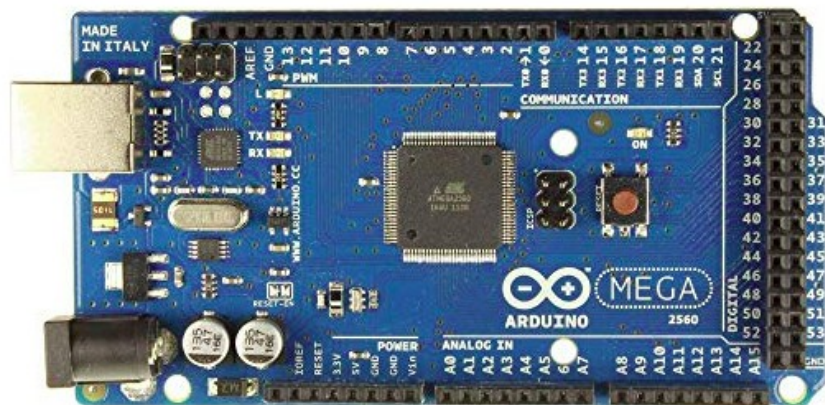


Figure 38: La carte Arduino Mega[26]

Pour commencer, il faut revenir à l'origine de l'Arduino. Cette carte électronique programmable sert de microcontrôleur pour la conception et le pilotage de montages électroniques, mécaniques, domotiques ou robotiques. En effet, grâce à l'Arduino, il est possible de créer des systèmes ingénieux amateurs ou professionnels, comme des systèmes de contrôle pour diverses composantes de notre maison. Grâce à Arduino, il est possible de se servir de son smartphone comme d'une télécommande pour par exemple, allumer ou éteindre les lumières de la maison, ou encore baisser ou lever les volets électriques, réguler la température de la maison en pilotant le chauffage à distance, etc[26]...

### **VI.1.c - Le module SIM800L**

Arduino ne peut pas à elle seule se connecter sur internet ou envoyer des SMS, c'est pourquoi il nous faut un module supplémentaire pour jouer ce rôle. Nous avons jeté notre dévolu sur le module GSM SIM800L dont nous avons pris une capture du kit complet comme le montre la figure 39.

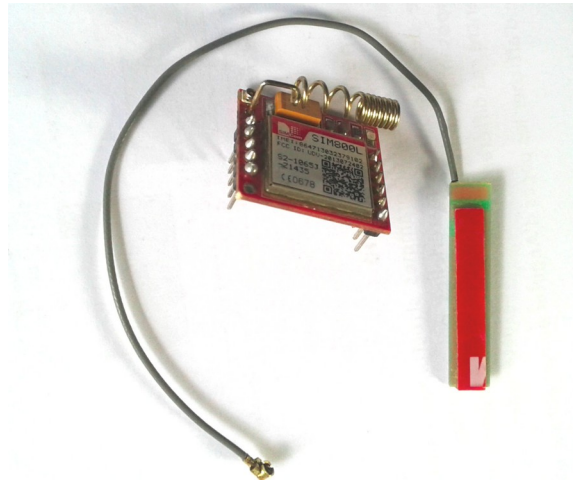


Figure 39: Kit complet du module SIM800L

Le module GSM SIM800L est l'un des plus petits modules GSM du monde avec une taille de 2.2cmx1.8cm. C'est un module puissant qui démarre automatiquement et recherche automatiquement le réseau. Il inclut notamment le Bluetooth 3.0+EDR et la radio FM (récepteur uniquement). Il permet d'échanger des SMS, de passer des appels mais aussi, et c'est nouveau, de récupérer de la data en GPRS 2G+. Ainsi il est possible de faire transiter des données sur une très longue distance, si par exemple la radio FM ou le Bluetooth ne suffit plus.

Ce module nécessite une alimentation entre 3,4 V et 4,4 V. L'alimentation 5 V de l'Arduino ne lui convient donc pas. Pour contrer ce problème d'alimentation, on ajoute une diode 1N4007 entre le 5 V de l'Arduino et le pin VCC du SIM800L. Le SIM800L nécessite un pic de courant d'environ 2A[27].

En retournant le module, figure 40, nous voyons sur l'autre face un compartiment réservé à l'insertion d'une carte micro SIM par exemple (Orange, Tigo, Espresso...).

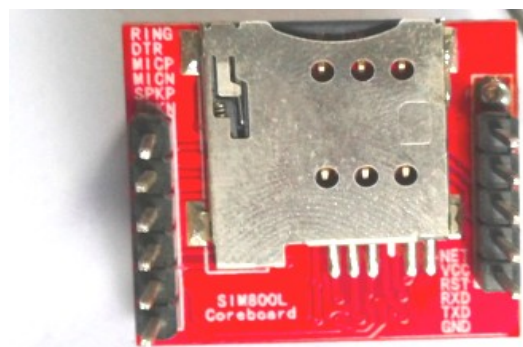


Figure 40: Face interne du module SIM800L

Comme le montre la figure 39 précédente, le module comporte deux antennes.

**La première**, figure 41, est faite d'un fil ayant la forme d'un ressort, elle doit être soudée à la broche **NET** du module. Elle est très utile dans les endroits étroits .



Figure 41: Antenne interne

La deuxième, figure 42, est très utile lorsque le module est par exemple à l'intérieur d'un boîtier métallique. Elle comporte un fil avec un connecteur IPX et un petit circuit imprimé.

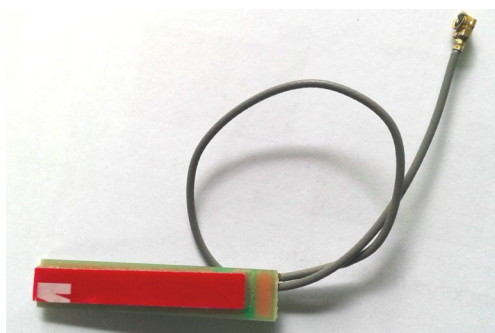


Figure 42: Antenne externe

Cette antenne est connectée au module à l'endroit encadré dans la figure 43.

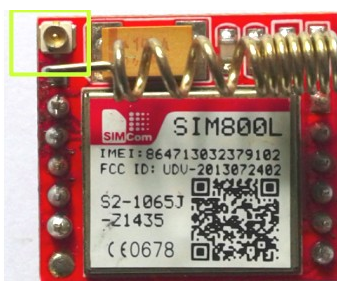


Figure 43: Branchement de l'antenne externe

Ce module dispose de six pins de part et d'autre soit douze au total. Cependant, nous n'en utiliserons que cinq :

- ◆ les pins **VCC** et **GND** : qui servent à alimenter le module en électricité et sont branchés respectivement sur le **5 V** et la masse de la carte Arduino ;
- ◆ le pin **RX** (réception) : il est branché à la broche 11 de notre carte Arduino et s'occupe de la réception des données (par exemple commandes envoyées par Arduino) ;
- ◆ le pin **TX** (transmission) : ce pin transmet les informations que le module veut communiquer et est branché à la broche 10 de la nôtre carte Arduino ;

- ◆ le pin **RST** (RESET=réinitialisation) : il est branché sur le pin 12 de notre carte et sert à initialiser le module au besoin.

### *VI.1.d - Les différents capteurs utilisés*

#### *VI.1.d.1 - Le capteur de température et d'humidité de l'air*

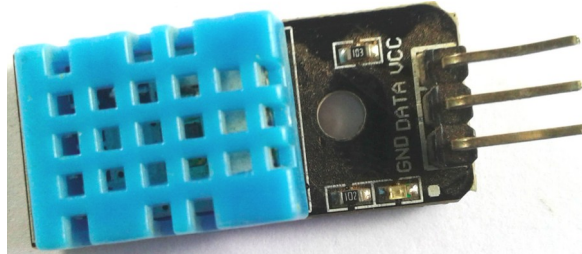


Figure 44: Le capteur DHT11

La température et l'humidité de l'air sont mesurées par un seul capteur dénommé le DHT11, figure 44. Le capteur DHT11 est capable de mesurer des températures de 0 à +50 °C avec une précision de +/- 2 °C et des taux d'humidité relative de 20 à 80 % avec une précision de +/- 5 %. Comme nous le voyons sur la figure 44, ce capteur comporte trois pins dont les deux **VCC** et **GND** qui servent à alimenter le capteur en électricité et l'autre **DATA** branché sur une broche analogique de la carte Arduino qui permet de recueillir les données mesurées.

#### *VI.1.d.2 - Le capteur d'humidité du sol*



Figure 45: Capteur d'humidité

Le capteur d'humidité du sol utilisé est, comme montré par la figure 45, composé de deux éléments.

Le premier élément est composé d'un **fil** assez long et d'une tête ayant deux sortes de barres qui sont en fait des **sondes**. Ce sont ces deux sondes qui seront enfoncées dans le sol pour prendre des mesures. Ces mesures sont ensuite transformées en signal analogique et numérique grâce au deuxième composant, **la platine**, qui est un circuit imprimé et est directement relié à la carte Arduino.

Ce capteur envoie des données comprises entre 0 et 1023. Plus la valeur envoyée est grande et plus le sol est sec.

La platine possède quatre bornes :

- ◆ les bornes **VCC** et **GND** pour l'alimentation du capteur branchées respectivement au 5 V et à la masse de la carte Arduino,
- ◆ les bornes **A0** et **D0** représentant respectivement la sortie analogique et numérique. En ce qui nous concerne, nous utilisons la sortie analogique uniquement.

Nous avons branché deux de ces capteurs à la carte Arduino, l'un pour mesurer l'humidité du sol et l'autre pour surveiller la quantité de l'eau d'arrosage.

Ce capteur est donc aussi utilisé pour mesurer le niveau de l'eau afin d'envoyer un signal (SMS) d'alerte si ce niveau atteint un seuil critique.

### ***VI.1.e - Composants utilisés pour tester le système***

Afin de tester le bon fonctionnement du système, nous avons utilisé une mini pompe à eau avec un relais et un tuyau.

#### ***VI.1.e.1 - La mini pompe à eau***

Pour la distribution de l'eau avec une certaine pression, nous utilisons la pompe à eau **Comet Pompe Submersible Elegant** d'une tension de 12 V avec un débit maximal de dix litres par minutes (10 l/mn). Nous avons pris une capture de la nôtre présentée à la figure 46.

Elle ne possède que deux bornes **VCC** et **GND** qui servent à son alimentation en électricité.



Figure 46: La pompe à eau

### VI.1.e.2 - Le tuyau d'arrosage

Le tuyau d'arrosage utilisé pour les tests est le **Tuyau Cristal Diam 10 × 14**, figure 47, d'un diamètre intérieur de 10 mm et d'un diamètre extérieur de 14 mm.



Figure 47: Tuyau d'arrosage

Nous l'avons percé de petits trous pour en faire un tuyau micro-poreux (ou tuyau à parois poreuses) afin de simuler une irrigation par goutte-à-goutte.

### VI.1.e.3 - Le relais

La carte Arduino est alimentée avec une tension de 5 V alors que la pompe nécessite 12 V pour fonctionner. Elle ne peut donc pas être reliée directement à Arduino d'où l'utilisation d'un relais.

Arduino va envoyer ses commandes au relais qui se chargera de couper ou de raccorder la pompe à sa source d'alimentation. Le relais est juste un interrupteur contrôlable par une carte Arduino.

Il nous faudra donc trouver une source d'alimentation pour la pompe autre que la carte Arduino.

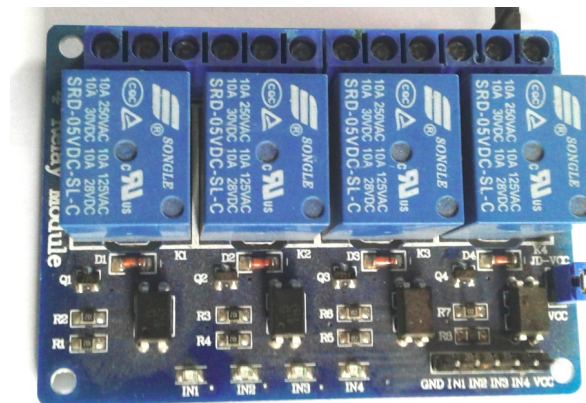


Figure 48: Module relais 5V

Cinq bornes nous intéressent au niveau du relais :

- ◆ les bornes VCC et GND : branchés respectivement sur les bornes 5 V et GND de la carte Arduino ;
- ◆ la borne INX : branché sur un pin de notre carte Arduino, ce pin permet le contrôle du relais ;
- ◆ les bornes COMX et NCX : ceux-ci permettent de fermer et d'ouvrir l'interrupteur du relais.



La lettre X représente un nombre entier, parce qu'en général, comme le montre la figure 48, plusieurs relais sont construits sur un même circuit imprimé pour contrôler par exemple plusieurs pompes à eaux. Chaque relais est dans ce cas numéroté par un entier différent en commençant par 1.

## VI.2 - Le montage de l'application

Pour effectuer ce montage, nous avons utilisé le logiciel **Fritzing** présenté un peu plus en bas dans ce document.

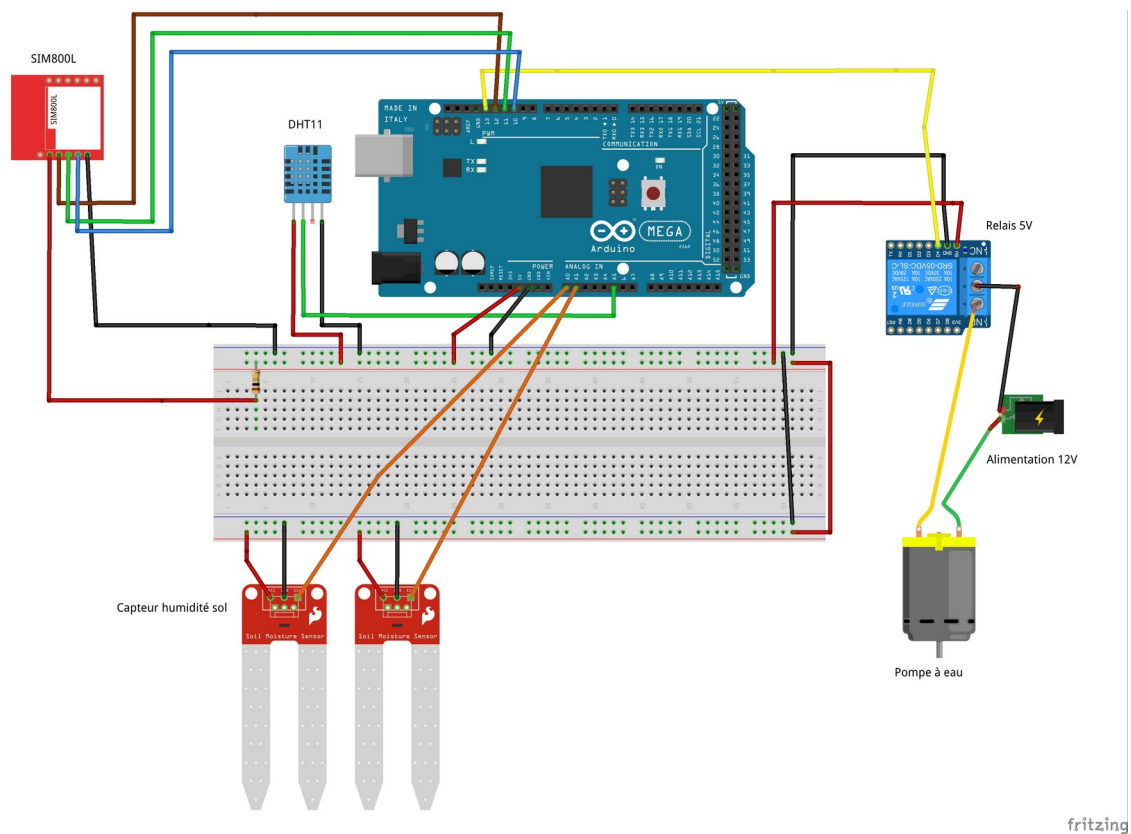


Figure 49: Montage du dispositif

## VI.3 - Librairies utilisées

Afin de manipuler simplement certains composants de l'application, nous avons eu recours à des librairies.

### VI.3.a.1 - *Arduino SIM800L library*

Elle utilise les commandes AT pour effectuer des requêtes http vers une API JSON en l'occurrence vers notre API Firebase Cloud Functions.

### VI.3.a.2 - *ArduinoJSON*

Celle-ci est utilisée par la librairie précédente pour formater les données en JSON.

### VI.3.a.3 - *Adafruit DHT Humidity & Temperature Sensor Library*

Cette librairie permet d'obtenir l'humidité et la température de l'air mesurées par le capteur **DHT11** présenté précédemment.

## VI.4 - Outils utilisés

### VI.4.a.1 - *Arduino IDE*

Pour la programmation de l'application, nous avons utilisé **Arduino IDE** qui est un environnement de développement gratuit spécialement conçu pour les projets basés sur la carte Arduino. Il est disponible en téléchargement pour Windows, Mac et Linux.

Cet éditeur nous permet d'écrire notre code, de le compiler et de le téléverser au niveau de la carte Arduino. Il intègre aussi des fonctionnalités de colorations syntaxique et nous permet de repérer les erreurs.

### VI.4.a.2 - *Fritzing*

Fritzing est un logiciel libre de conception de circuit imprimé qui permet de concevoir de façon entièrement graphique le circuit et d'en imprimer le typon.

Se voulant dans la ligne d'Arduino et de processing, Fritzing est un projet de logiciel libre, destiné aux non-professionnels de l'électronique. Il a notamment pour vocation de favoriser l'échange de circuits électroniques libres et d'accompagner l'apprentissage de la conception de circuits.

Le logiciel conçu par la faculté de sciences appliquée de l'Université de Potsdam et dont le développement est assuré par la fondation, également nommée Fritzing, est un logiciel d'édition de circuit imprimé. Il est disponible dans seize langues dont le français. Il est adapté aux débutants ou confirmés en électronique pour faire rapidement des circuits simples, et est également un bon outil didactique pour apprendre à bidouiller en électronique par la pratique : il est utilisable en classe à partir de 12 ans et en université[28].

## VI.5 - Présentation de l'application Arduino

Pour présenter l'application Arduino, nous avons pris une capture, figure 50, du montage réalisé.

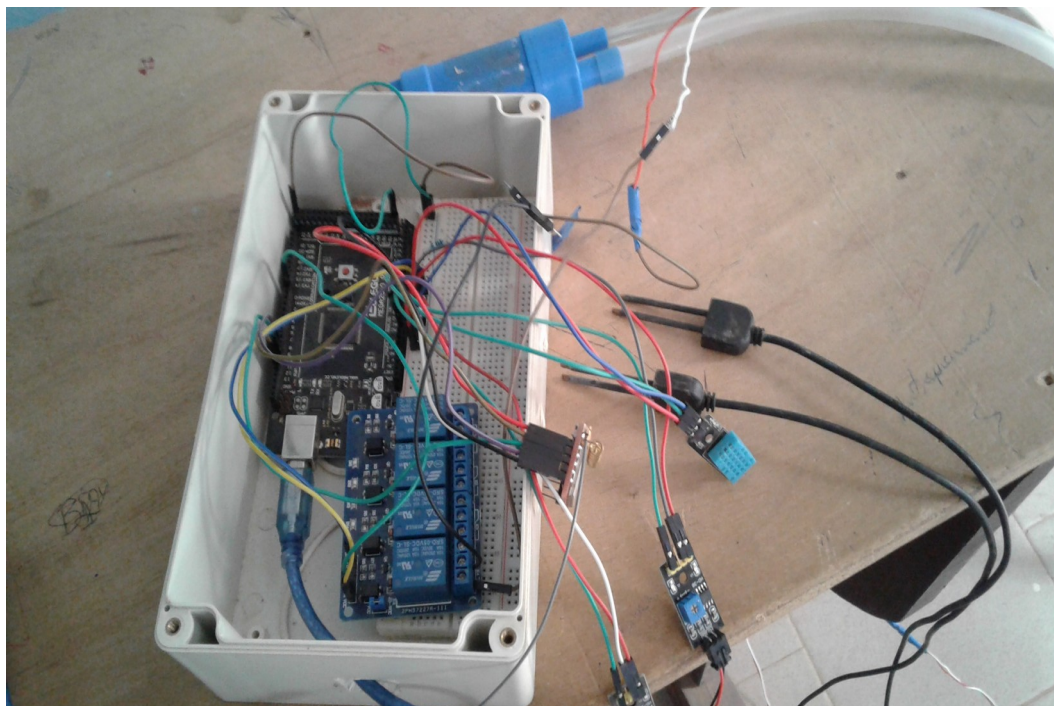


Figure 50: Le dispositif

## VII - Implémentation et présentation de la plate-forme web

### VII.1 - Technologies utilisées

#### VII.1.a - *Le framework Angular*



Figure 51: Angular framework[29]

Angular, figure 51, est un framework Javascript côté client qui permet de réaliser des applications de type « Single Page Application ». Il est basé sur le concept de l'architecture MVC (Model View Controller) qui permet de séparer les données, les vues et les différentes actions que l'on peut effectuer.

Depuis 2009 jusqu'à aujourd'hui, Google a sorti trois différentes versions d'Angular: AngularJS (2009), Angular 2 (2016) et Angular 4 (2017). À partir de la version 2, le framework a été complètement réécrit. AngularJS n'est donc pas compatible avec les versions ultérieures. En revanche, il est tout à fait possible de migrer entre la version 2 et la version 4[29].

Angular fournit nativement tout le nécessaire pour produire une application entière avec une configuration standard :

- ◆ Configuration de build et d'optimisation complète,
- ◆ Module d'animations,
- ◆ Module de routages,
- ◆ Module de formulaires,
- ◆ Debuggage,
- ◆ Tests unitaires,
- ◆ ...

Les applications Angular sont en grande majorité codées en TypeScript, même si d'autres langages sont supportés, comme Dart. Ces langages sont convertis en JavaScript avant l'exécution (le mot officiel est « transpilé ») pour permettre aux applications Angular de s'exécuter dans tous les navigateurs Internet.

En ce qui nous concerne, nous allons utiliser le langage TypeScript.

### ***VII.1.b - TypeScript***

TypeScript est un langage open source développé par Microsoft, le code écrit en TypeScript est trans-compilé en JavaScript. TypeScript apporte un typage et une phase de compilation au développement d'application frontend JavaScript. Il permet ainsi de créer des classes, des interfaces, de l'héritage, des variables, des énumérations...

### ***VII.1.c - JQuery***

jQuery est une bibliothèque JavaScript rapide, petite et riche en fonctionnalités. Il rend les choses comme la traversée et la manipulation de documents HTML, la gestion des événements, l'animation et Ajax beaucoup plus simples avec une API facile à utiliser qui fonctionne sur une multitude de navigateurs. Avec une combinaison de polyvalence et d'extensibilité, jQuery a changé la façon dont des millions de personnes écrivent JavaScript[29].

### ***VII.1.d - Bootstrap***

Bootstrap est une boîte à outils open source pour le développement avec HTML, CSS et JavaScript. Il est la bibliothèque de composants frontaux la plus populaire au monde et permet de créer des projets réactifs et axés sur les mobiles sur le Web.

Pour nous faciliter la tâche, Bootstrap propose des variables et mixins Sass, un système de grille réactif, de nombreux composants pré-construits et des plugins puissants construits sur jQuery.

### ***VII.1.e - Font Awesome***

Font Awesome est l'une des boîtes à outils les plus populaires du Web contenant des icônes vectorielles et des logos sociaux facilement utilisables dans nos projets web.

Les icônes vectorielles proposées sont évolutives et peuvent être facilement personnalisées en taille, couleur, ombre portée et tout ce qui peut être fait en utilisant CSS.

Font Awesome se base sur les langages CSS, LESS et SASS.

### ***VII.1.f - ChartJS***

ChartJS est une librairie JavaScript permettant de dessiner des graphiques simples et flexibles. Cette librairie propose huit différents graphiques dont les graphiques en bâton, des graphiques circulaires, en points de nuages, en courbe...

### ***VII.1.g - Ng Date Time Picker***

Cette librairie permet d'entrer une date et/ou une heure soit par saisie de texte, soit en choisissant une date dans le calendrier et/ou une heure dans la minuterie. Il est composé de plusieurs composants et directives qui fonctionnent ensemble.

## VII.2 - Outils utilisés

### VII.2.a - *Visual Paradigm Community Edition*

Logiciel de modélisation intuitif qui prend en charge plus de 50 diagrammes comme UML, Diagramme d'exigences, ERD, Org. Graphique, organigramme, etc. Gratuit pour des utilisations non commerciales, y compris des projets éducatifs, à but non lucratif et personnels. Utilisation et mises à niveau gratuites pour toujours.

### VII.2.b - *Visual Studio Code*

Visual Studio Code est un éditeur de code open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il supporte l'auto-complétion, la coloration syntaxique, le débogage, et les commandes Git.

L'histoire remonte à novembre 2015. Lors d'un événement à New York, Microsoft lançait en grande pompe un tout nouvel outil à destination des développeurs : Visual Studio Code (ou VS Code). Proposée en open source sous licence MIT, l'application, disponible à la fois pour Windows, Linux et Mac, se présente sous la forme d'un environnement multi langages léger. À l'époque, les observateurs sont peu nombreux à croire en la réussite de l'initiative. Jusqu'alors, Microsoft n'était jamais parvenu à faire émerger un logiciel open source sur la scène informatique mondiale.

Deux ans après, le résultat est pourtant là : VS Code s'est imposé parmi les briques open source de référence. L'IDE se classe même en tête des projets avec le plus grand nombre de contributeurs sur la plateforme GitHub, en comprenant les plugins associés[30].

## VII.3 - Arborescence des paquetages et fichiers de l'application web

L'application web offre principalement trois grandes fonctionnalités à savoir la gestion des comptes utilisateur dans le paquetage **account**, l'administrateur dans le paquetage **admin** et enfin la gestion du tableau de bord dans le paquetage **dashboard** qui sont représentées dans la figure 52 suivante.

Comme le montre la figure suivante, le paquetage **account** compte plusieurs fonctionnalités qui sont regroupées dans des composants d'Angular :

- ✓ **registration**: dans ce composant nous gérons la création de compte utilisateur,
- ✓ **sign-in** : dans ce composant nous gérons l'authentification,
- ✓ **change-password** : permet à un utilisateur connecté de changer son mot de passe,
- ✓ **forgot-password** : permet de réinitialiser son mot de passe en cas d'oublie,
- ✓ **profil** : gère la modification de profil,
- ✓ **verify-email** : gère l'envoi des mails de confirmation de compte.

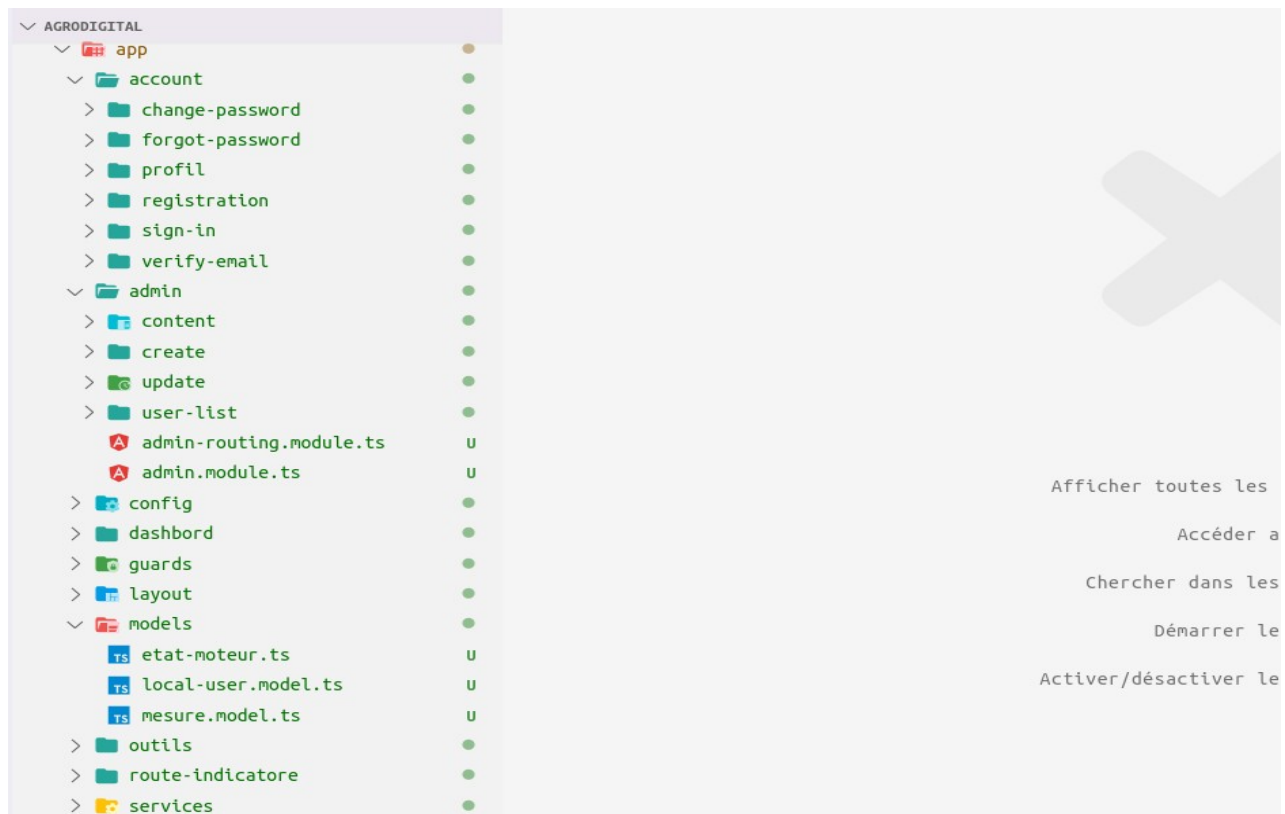


Figure 52: Arborescence des paquetages et fichiers de l'application

Le paquetage **admin** quant à lui regroupe tout ce qui concerne l'administration des comptes utilisateur :

- ✓ **create** : est le composant qui permet à un administrateur de créer un compte utilisateur,
- ✓ **update** : permet de modifier un compte utilisateur sauf son mot de passe,
- ✓ **user-list** : se charge de lister tous les comptes utilisateurs afin d'accéder aux deux fonctionnalités citées plus haut. Il donne aussi accès à la suppression, à l'activation et à la désactivation de compte utilisateur.

C'est au niveau du paquetage **dashbord** qu'est géré tout ce qui concerne le tableau de bord :

- ✓ **charts** : dans ce composant, on retrouve les différents diagrammes qui servent à afficher les données ainsi que leur manipulation,
- ✓ **controlbar** : permet de voir au premier coup d'œil les informations pertinentes comme la température et l'humidité ambiante ainsi que le niveau d'humidité du sol avec des jauges et des codes de couleur. Il permet aussi de savoir si l'arrosage est déclenché ou arrêté, à quelle heure et surtout si c'est arrivé automatiquement ou manuellement (envoi d'une commande),
- ✓ **history** : permet de consulter l'historique des arrosages avec tous les détails,
- ✓ **export-data** : donne la possibilité d'exporter les données en EXCEL ou en CSV pour un intervalle de temps choisi,

- ✓ les composants **humidite-air**, **temperature-air** et **humidite-sol** permettent d'afficher un diagramme représentant respectivement l'évolution de l'humidité de l'air, de la température de l'air et de l'humidité du sol pour un intervalle de temps choisi avec la possibilité d'utiliser un diagramme linéaire ou en barre.

Un composant angulaire, figure 53, est un dossier constitué de quatre fichiers d'extension différentes:

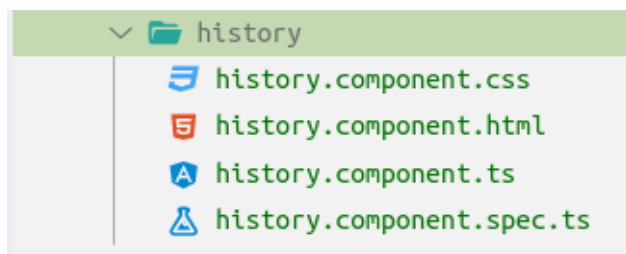


Figure 53: Détails d'un composant

- ✓ un fichier **HTML** qui va contenir le code HTML du composant,
- ✓ un fichier **CSS** qui va contenir la définition des styles,
- ✓ un fichier **TypeScript** d'extension « **.ts** » qui va contenir le code JavaScript du composant. Ce fichier peut accéder aux différents services du paquetage **services** afin de passer au fichier HTML les données à afficher. C'est aussi dans ce fichier qu'est définie le nom du composant utilisé pour l'inclure à l'endroit désiré du site,
- ✓ enfin un fichier d'extension « **.spec.ts** » qui va permettre de tester le bon fonctionnement du composant.

Nous avons aussi les paquetages **layout**, **models** et **services**, figure 52, qui sont tout aussi importants :

- ✓ **layout** contient les différentes parties de notre site web à savoir la barre de navigation, la barre latérale gauche et le pied de page,
- ✓ dans **models**, nous retrouvons les interfaces qui représentent les données manipulées et gardées en base de données ; nous pouvons remarquer sur la figure 54, la définition de l'interface qui représente l'état de la pompe à eau appelée ici moteur,
- ✓ **services** quant à lui regroupe les fonctionnalités qui nous permettent d'interagir avec la base de données en utilisant les interfaces définies dans le paquetage **models** .

Voir l'annexe pour des exemples de code du paquetage **service**.

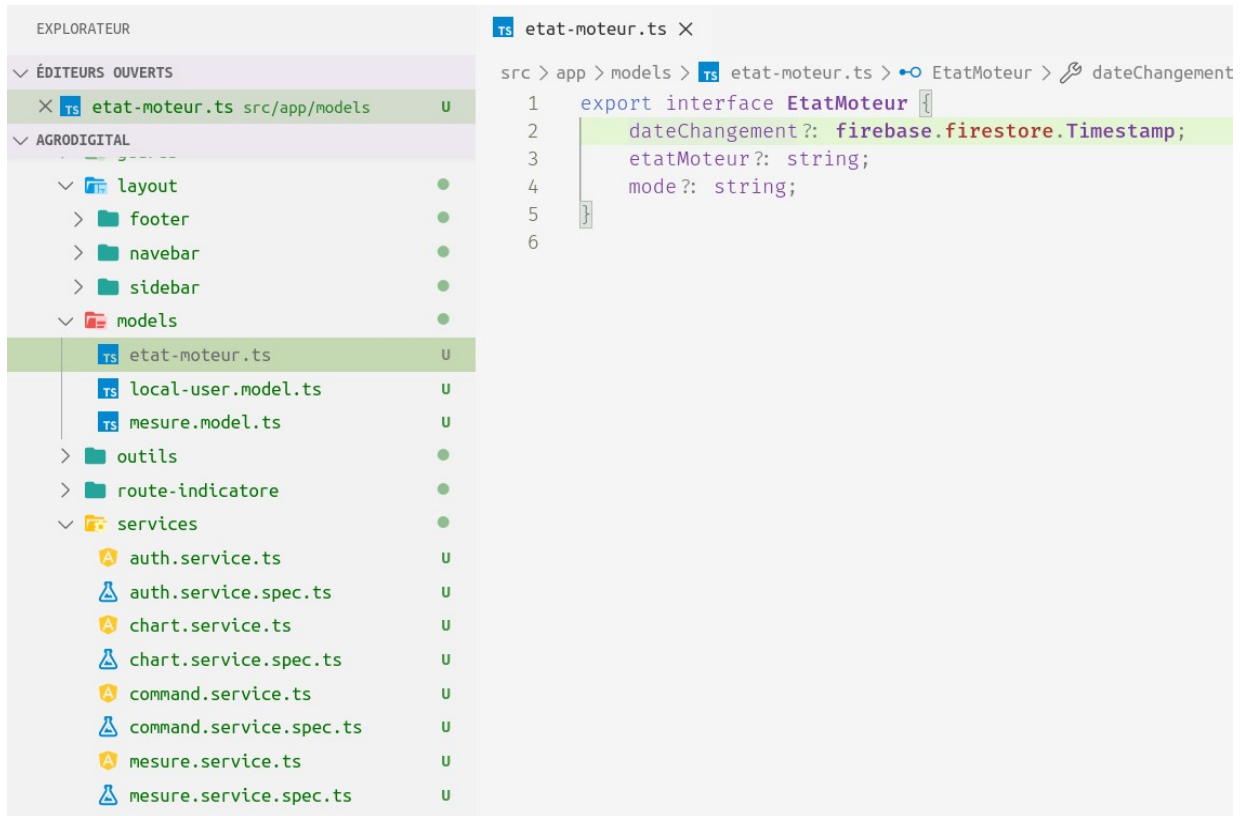


Figure 54: Présentation de paquetages

À la racine de notre projet, figure 54 toujours, se trouvent trois fichiers particulièrement importants à savoir **app-routing.module.ts**, **app.module.ts** et **package.json**. Un extrait du contenu de chacun de ces fichiers se trouve dans la partie annexe.

- ✓ **package.json** contient l'ensemble des dépendances de notre site ainsi que leur version. Ces dépendances sont téléchargées et gardées dans le dossier **node\_modules** lui aussi à la racine du projet,
- ✓ c'est dans le fichier **app-routing.module.ts** que sont définies les routes ou URL permettant d'accéder à un composant donné. Il fait donc la correspondance entre un composant de notre site et l'URL à appeler.
- ✓ le fichier **app.module.ts** est le fichier de configuration principale du site. Elle initialise les dépendances installées, les routes définies dans le fichier **app-routing.module.ts**, l'API Firebase...

## VII.4 - Présentation de l'application web

Pour la présentation du site web, commençons par le diagramme de navigation, figure 55, pour avoir une vue d'ensemble de notre plate-forme.

La plate-forme accepte deux types d'utilisateurs à savoir un administrateur et un utilisateur simple. L'administrateur peut en plus de la gestion des comptes utilisateur, faire tout ce que fait un utilisateur simple.



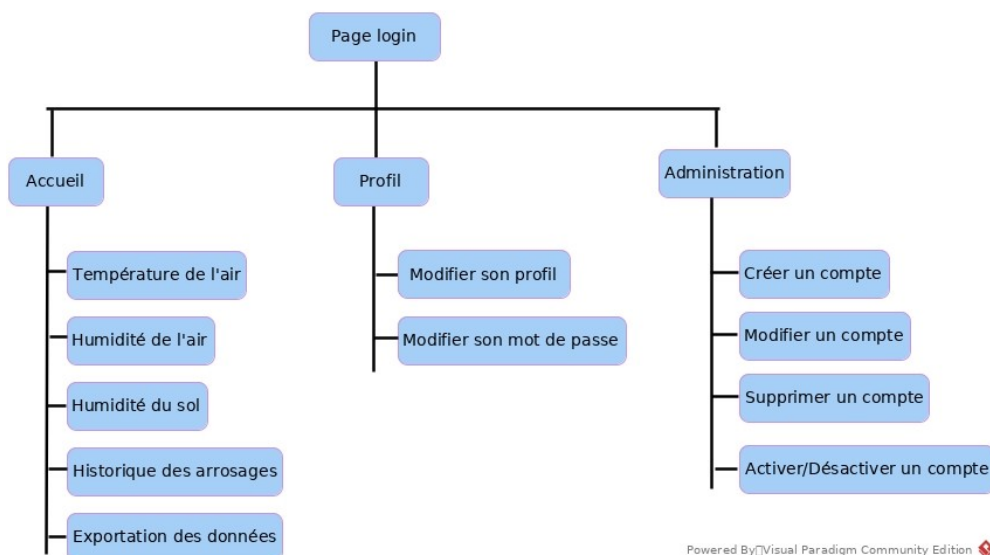


Figure 55: Arborescence du site

Pour présenter certaines des fonctionnalités offertes par notre site web, nous avons effectué les captures d'écran ci-dessous.

#### VII.4.a - La page d'inscription

Avant d'être considéré comme un utilisateur du système, il faut d'abord créer un compte. La figure 56 suivante est une capture de la page d'inscription.

**Agro-Digital**  
Ziguinchor institut polytechnique (ZIP)

### Inscription

**NB:** Assurez-vous de renseigner une adresse email valide ainsi qu'un numéro de téléphone. Nous devons pouvoir vous joindre afin de finaliser votre inscription.

Prénom  Nom

Adresse

Téléphone  Nom d'utilisateur

Email

Mot de passe  Confirmation

Figure 56: La page d'inscription

### **VII.4.b - La page d'authentification**

Après s'être inscrit, nous devons nous connecter en fournissant notre adresse mail et notre mot de passe comme présenté par la figure 57. Si le mot de passe est oublié, il est possible de le récupérer en cliquant sur le lien « *Mot de passe oublié ?* ».

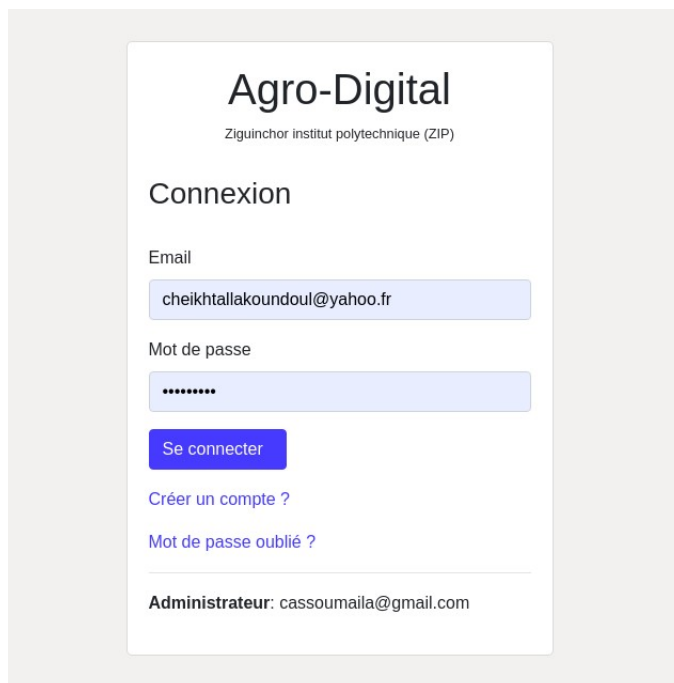


Figure 57: La page de connexion

### **VII.4.c - La page d'accueil**

Après l'authentification, nous accédons à la page d'accueil, figure 58, si nous sommes de simples utilisateurs tandis qu'un administrateur va directement accéder à la page d'administration. Nous voyons le lien « *Administration* » parce que nous utilisons ici un compte administrateur.

À partir de l'accueil, plusieurs fonctionnalités sont accessibles notamment toutes celles listées au niveau de la barre latérale gauche.

Pour accéder au profil et le modifier, il faut cliquer sur le bouton à l'extrême droite de la barre de navigation supérieure et choisir sur la liste déroulante le lien « *Profil* ».

Nous voyons ici quatre carrées qui permettent de voir en un coup d'œil, les informations pertinentes avec des codes de couleur et des appréciations.

Nous voyons aussi deux graphes dont le premier en partant de la gauche, représente l'évolution de l'humidité du sol en fonction du temps tandis que le deuxième représente les variations de l'humidité et de la température ambiante en fonction du temps. Ces deux données sont représentées sur le même graphe pour deux raisons. La première est qu'elles sont mesurées par un seul et même capteur, et la deuxième est de gagner de l'espace afin que toute la page rentre sur l'écran de l'utilisateur.

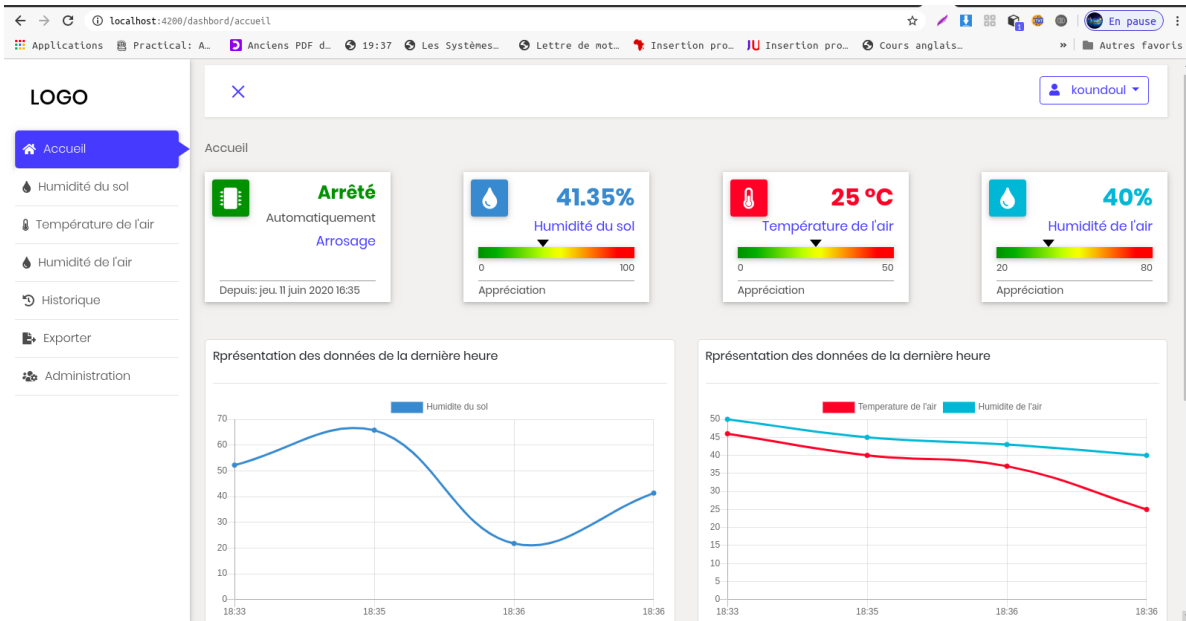


Figure 58: La page d'accueil

#### VII.4.d - La page de la température ambiante

Ici, figure 59, nous avons choisi d'afficher le graphe en barre. En bas du graphe, nous avons un champ qui nous permet de sélectionner un intervalle de dates pour visualiser les données enregistrées dans cet intervalle.

Nous pouvons aussi laisser l'affichage par défaut qui représente les données envoyées d'il y a une heure jusqu'à présent.

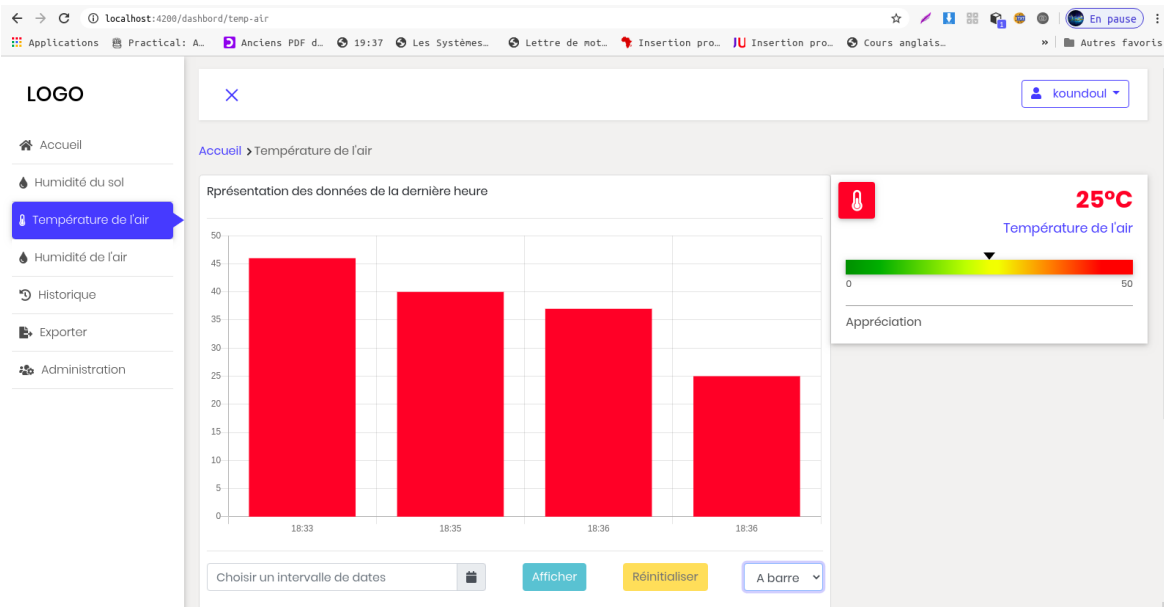


Figure 59: La page de la température ambiante

### VII.4.e - La page d'historique des arrosages

Cette page, figure 60, nous donne accès à l'historique des arrosages avec tous les détails nécessaires.

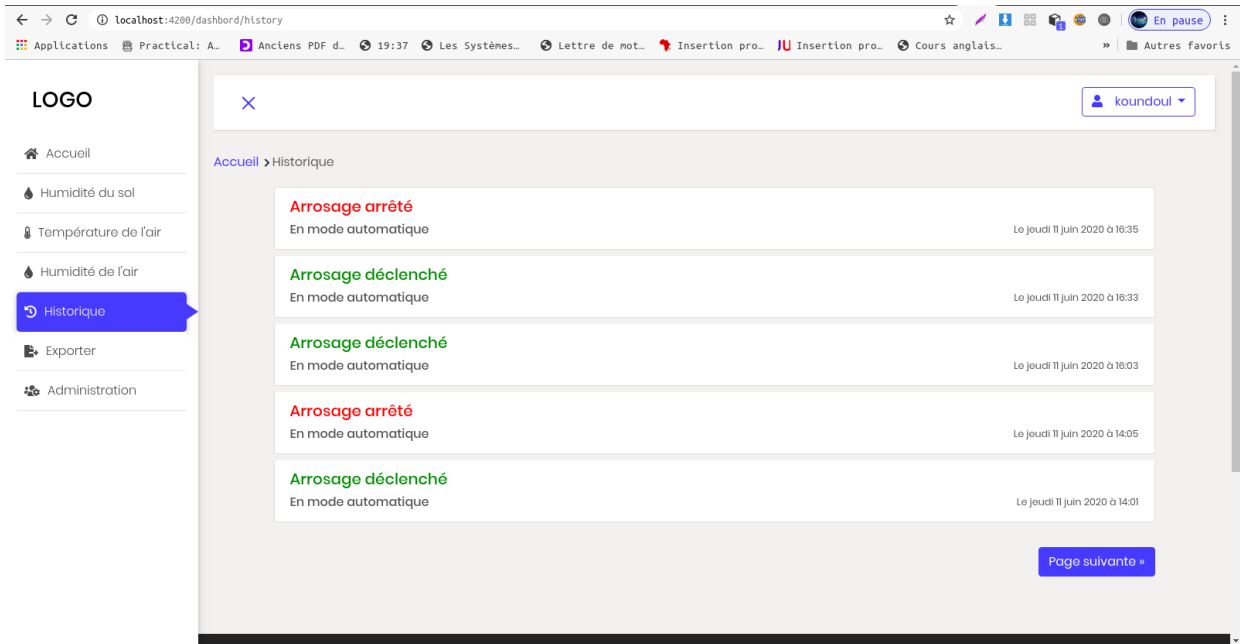


Figure 60: La page d'historique des arrosages

### VII.4.f - La page d'exportation des données

Pour exporter des données en CSV ou en EXCEL, figure 61, nous devons choisir un intervalle de dates et cliquer sur l'un des deux boutons d'exportation.

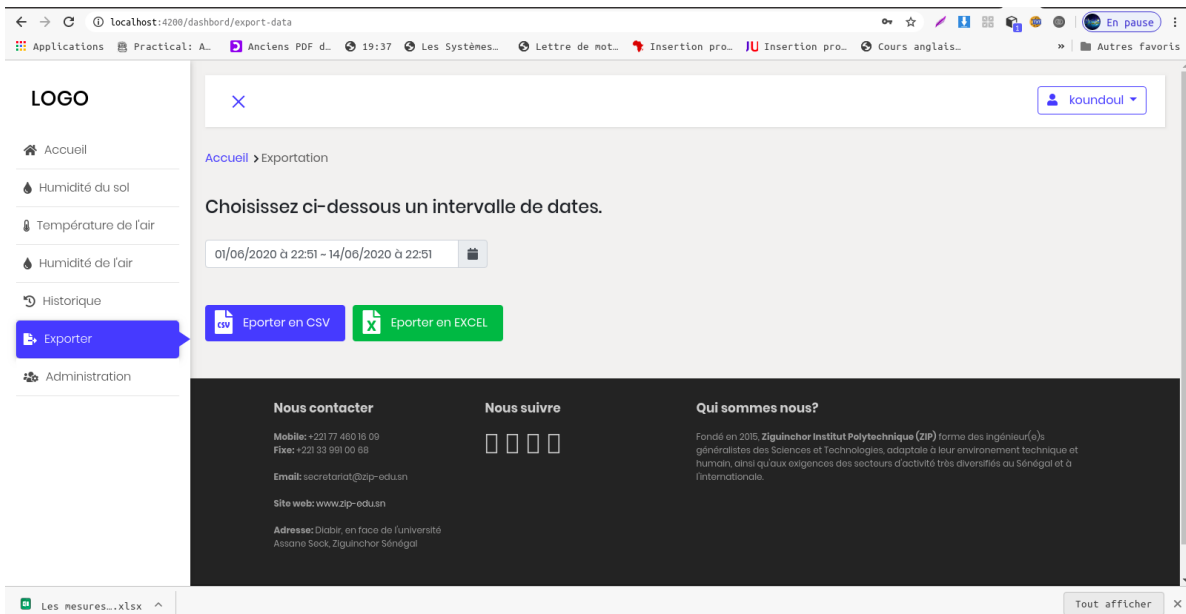


Figure 61: L'exportation des données

### VII.4.g - La page d'administration

Cette page, figure 62, donne accès à la gestion des comptes utilisateur.

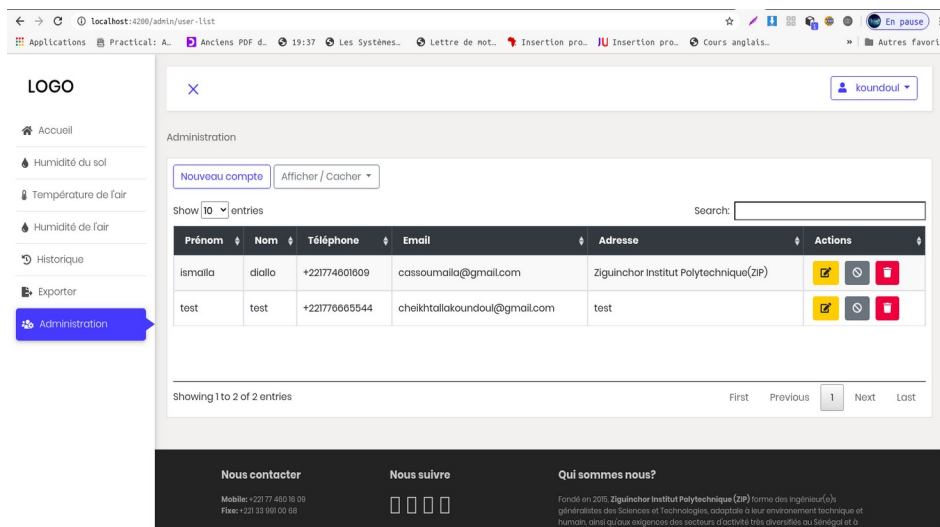


Figure 62: La page d'administration

### VII.4.h - Activer/Désactiver un compte utilisateur

La figure 63 représente l'activation ou la désactivation d'un compte à partir le page d'administration.

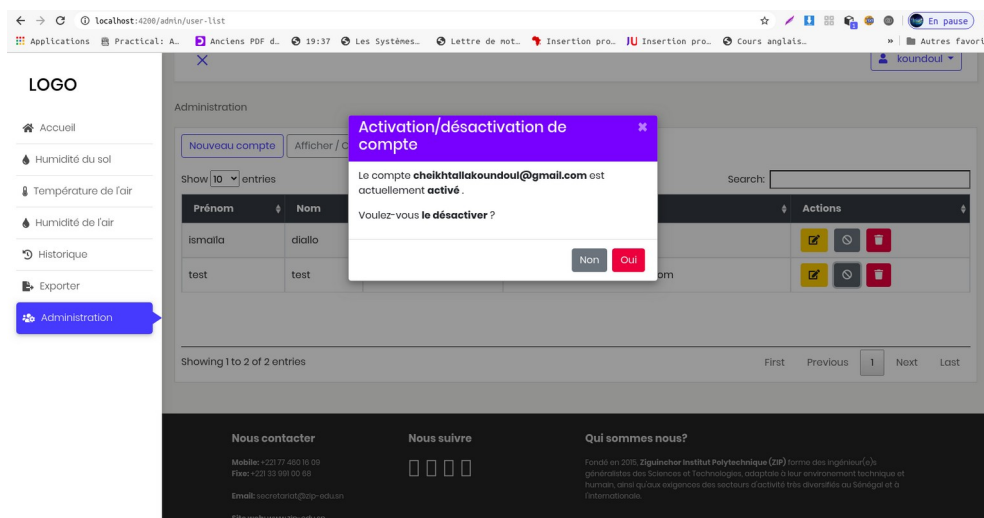


Figure 63: Activer ou Désactiver un compte utilisateur

## Conclusion

Dans ce chapitre, nous avons fait le tour des grandes lignes de l'implémentation d'AgroDigital. Nous avons présenté les technologies et outils utilisés pour y parvenir avant de présenter le résultat obtenu à travers des captures d'écran.

## CONCLUSION GÉNÉRALE

Ce mémoire avait pour objectif l'implémentation d'un système informatique embarqué permettant de gérer de manière plus intelligente l'agriculture moderne pour pallier les limites des techniques d'irrigation traditionnelles. Plus précisément, il consistait à faire en sorte d'utiliser au mieux l'eau, en même temps que les terres et les ressources humaines de façon à augmenter la productivité des cultures irriguées dans le but de participer à la sécurité alimentaire du pays.

Il a fallu dans un premier temps identifier les problèmes rencontrés avec les méthodes d'irrigation traditionnelles et proposer une solution avant de nous familiariser avec les différents systèmes d'irrigation qui existent. Nous avons ensuite étudié et analysé cette solution en détail à l'aide de Scrum et des fonctionnalités attendues ce qui nous a permis de faire la conception en utilisant UML. Après avoir défini l'architecture de notre système, il ne nous restait alors plus qu'à l'implémenter avec les outils adéquats avant de le présenter.

En plus d'apporter une solution aux problèmes identifiés, AgroDigital comporte d'autres avantages : il est interactif et contrôlable à distance, nous pouvons donc partir en vacances ou au travail sans soucis. Il permet de collecter des données climatiques pour concevoir une base de données.

Une amélioration possible est de développer une fonctionnalité permettant de tirer des informations à partir des données collectées. Cela permettra peut-être d'optimiser et d'adapter notre agriculture aux changements climatiques.

Il serait aussi mieux d'envoyer les commandes par SMS plutôt que de le faire à travers le tableau de bord pour déclencher ou arrêter l'arrosage. En effet, nous ne disposons pas forcément d'une connexion internet en permanence. Cette fonctionnalité a été testée mais le dispositif finit toujours par se bloquer et il faut redémarrer. C'est la raison pour laquelle nous avons utilisé la plate-forme web pour envoyer les commandes.

## ANNEXE

### I - Le code côté serveur pour Arduino

Pour envoyer les données sur internet, voici, sur la figure 64 suivante, la fonction appelée par Arduino.

Il s'agit d'une fonction du cloud écrite en JavaScript. Nous pouvons constater que la date est ajoutée avant de formater le tout en JSON et de l'enregistrer dans la base de données.

L'identifiant utilisé ici correspond à celui attribué à l'utilisateur lors de son inscription sur la plate-forme web.

Le niveau de l'eau est aussi envoyé pour prévoir le jour où l'on voudra l'affiché sur le tableau de bord. Mais pour le moment cette valeur n'est pas utilisée.

```

109
110 exports.insertMesure = fonctions.https.onRequest((req, res) => {
111     const identifiant = req.body.identifiant;
112     const temperatureAir = req.body.temperatureAir;
113     // elle varie de 0 à 1023, on garde seulement le pourcentage en base de données
114     const humiditeSol = Number((1023 - Number(req.body.humiditeSol)) * 100 / 1023).toFixed(2);
115     const humiditeAir = req.body.humiditeAir;
116     const niveauEau = req.body.niveauEau;
117     const phSol = req.body.phSol;
118     const dateMesure = admin.firestore.FieldValue.serverTimestamp();
119
120     const mesure = {
121         temperatureAir: temperatureAir,
122         humiditeSol: humiditeSol,
123         humiditeAir: humiditeAir,
124         niveauEau: niveauEau,
125         phSol: phSol,
126         dateMesure: dateMesure
127     };
128
129     admin.firestore().collection('users/' + identifiant + '/mesures')
130         .add(mesure)
131         .then(() => {
132             return res.status(200).send('Ok');
133         })
134         .catch(err => {
135             return res.status(500).send('Error: ' + err);
136         });
137
138 });

```

Figure 64: Fonction qui enregistre les données mesurées par les capteurs

### II - Exemple de deux méthodes du paquetage service pour obtenir les données mesurées par les capteurs

Voici deux exemples de méthode de la classe **MesureService** du paquetage **service**, figure 65 et figure 66. La première permet de récupérer au niveau de la base de données les mesures dont la date d'enregistrement est supérieur ou égale à une date donnée, tandis que la deuxième récupère les mesures dont la date d'enregistrement est comprise entre deux dates données.

Cette classe, **MesureService**, utilise pour cela l'interface **Mesure** définie dans le paquetage **models** comme représentés par les figures suivantes.

```
getMesures(startDate: Date): Observable<Mesure[]> {
  return this.afs.collection(this.users_coll).doc(this.user_id)
    .collection(this.mesures_coll, ref =>
      ref.where('dateMesure', '≥', startDate)
        .orderBy('dateMesure', 'desc'))
    .valueChanges();
}

getMesuresBetwenToDates(startDate: Date, endDate: Date): Observable<Mesure[]> {
  return this.afs.collection(this.users_coll).doc(this.user_id)
    .collection(this.mesures_coll, ref =>
      ref
        .where('dateMesure', '≤', endDate)
        .where('dateMesure', '≥', startDate)
        .orderBy('dateMesure', 'desc'))
    .valueChanges();
}
```

Figure 65: Exemple de méthodes d'accès aux données



```
src > app > models > mesure.model.ts > ...
1  export interface Mesure {
2    temperatureAir?: number;
3    humiditeSol?: number;
4    humiditeAir?: number;
5    phSol?: number;
6    niveauEau?: number;
7    dateMesure?: firebase.firestore.Timestamp;
8  }
9
```

Figure 66: L'interface Mesure

### III - Extrait du contenu de quelques fichiers de configuration et de dépendance

#### III.1 - Contenu du fichier de dépendance package.json



```

package.json > {} dependencies
16  "@angular/common": "~8.2.14",
17  "@angular/compiler": "~8.2.14",
18  "@angular/core": "~8.2.14",
19  "@angular/fire": "^5.4.2",
20  "@angular/forms": "~8.2.14",
21  "@angular/platform-browser": "~8.2.14",
22  "@angular/platform-browser-dynamic": "~8.2.14",
23  "@angular/router": "~8.2.14",
24  "@fortawesome/fontawesome-free": "^5.13.0",
25  "angular-fusioncharts": "^3.0.4",
26  "angular2-chartjs": "^0.5.1",
27  "angular7-csv": "^0.2.12",
28  "bootstrap": "^4.5.0",
29  "core-js": "^3.6.5",
30  "datatables.net": "^1.10.21",
31  "datatables.net-dt": "^1.10.21",
32  "file-saver": "^2.0.2",
33  "firebase": "^5.8.6",
34  "firebase-admin": "^8.12.1",
35  "fusioncharts": "^3.15.1",
36  "jquery": "^3.5.1",
37  "moment": "^2.26.0",
38  "ng-pick-datetime": "^7.0.0",
39  "ngx-toastr": "^10.0.4",
40  "popper.js": "^1.16.1",
41  "rxjs": "~6.4.0",
42  "tslib": "^1.13.0",
43  "xlsx": "^0.14.3",
44  "zone.js": "~0.9.1"
45  },
46  "devDependencies": {
47    "@angular-devkit/build-angular": "^0.803.27",
48    "@angular/cli": "^8.3.27",

```

Figure 67: Le fichier package.json

### III.2 - Contenu du fichier de configuration app-routing.module.ts

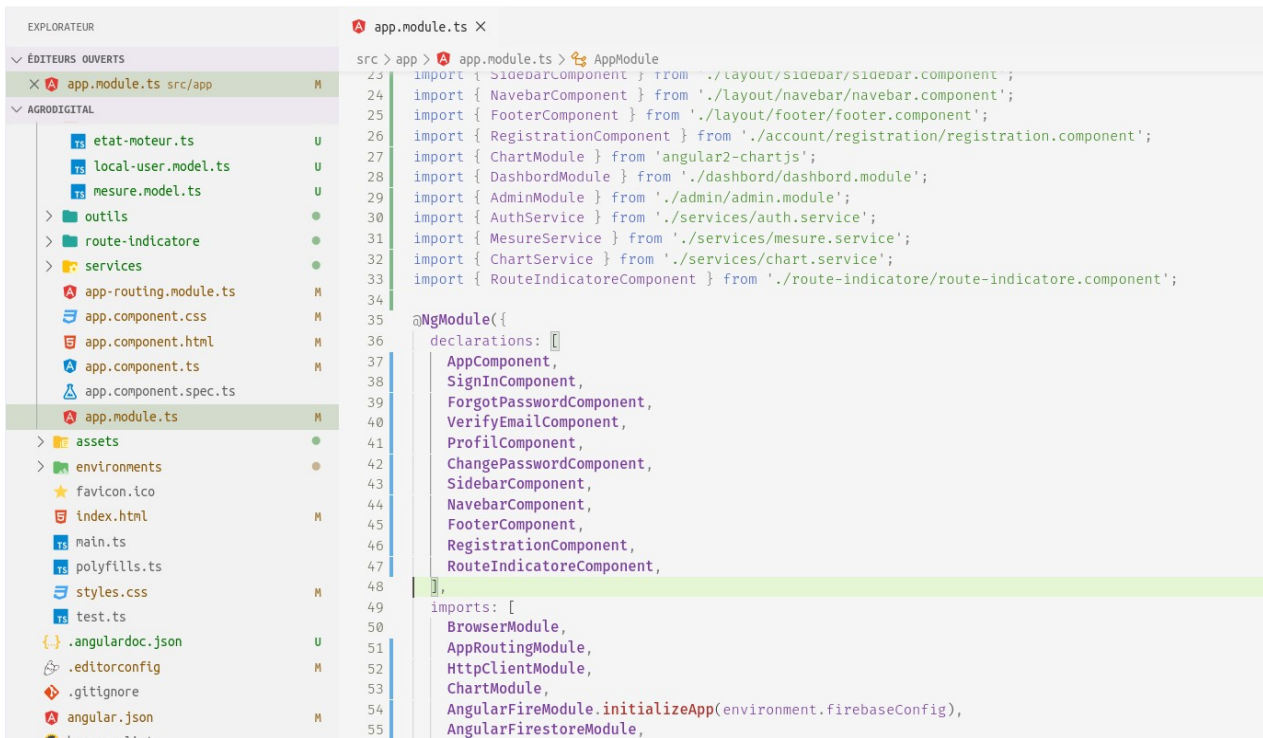
```

src > app > app-routing.module.ts > routes > path
1  import { ProfilComponent } from './account/profil/profil.component';
2  import { SecureInnerPagesGuard } from './guards/secure-inner-pages.guard';
3  import { ForgotPasswordComponent } from './account/forgot-password/forgot-password.component';
4  import { SignInComponent } from './account/sign-in/sign-in.component';
5  import { NgModule } from '@angular/core';
6  import { Routes, RouterModule } from '@angular/router';
7  import { VerifyEmailComponent } from './account/verify-email/verify-email.component';
8  import { AuthGuard } from './guards/auth.guard';
9  import { ChangePasswordComponent } from './account/change-password/change-password.component';
10 import { RegistrationComponent } from './account/registration/registration.component';
11
12 const routes: Routes = [
13   {path: '', redirectTo: '/sign-in', pathMatch: 'full'},
14   {path: 'sign-in', component: SignInComponent, canActivate: [SecureInnerPagesGuard]},
15   {path: 'verify-email', component: VerifyEmailComponent, canActivate: [SecureInnerPagesGuard]},
16   {path: 'forgot-password', component: ForgotPasswordComponent, canActivate: [SecureInnerPagesGuard]},
17   {path: 'profil', component: ProfilComponent, canActivate: [AuthGuard]},
18   {path: 'change-password', component: ChangePasswordComponent, canActivate: [AuthGuard]},
19   {path: 'registration', component: RegistrationComponent, canActivate: [SecureInnerPagesGuard]}
20 ];
21
22 @NgModule({
23   imports: [RouterModule.forRoot(routes)],
24   exports: [RouterModule]
25 })
26 export class AppRoutingModule { }
27

```

Figure 68: Le fichier app-routing.module.ts

### III.3 - Contenu du fichier de configuration app.module.ts



```
src > app > app.module.ts > AppModule
23 import { SidebarComponent } from './layout/sidebar/sidebar.component';
24 import { NavebarComponent } from './layout/navebar/navebar.component';
25 import { FooterComponent } from './layout/footer/footer.component';
26 import { RegistrationComponent } from './account/registration/registration.component';
27 import { ChartModule } from 'angular2-chartjs';
28 import { DashbordModule } from './dashbord/dashbord.module';
29 import { AdminModule } from './admin/admin.module';
30 import { AuthService } from './services/auth.service';
31 import { MesureService } from './services/mesure.service';
32 import { ChartService } from './services/chart.service';
33 import { RouteIndicatoreComponent } from './route-indicatore/route-indicatore.component';
34
35 @NgModule({
36   declarations: [
37     AppComponent,
38     SignInComponent,
39     ForgotPasswordComponent,
40     VerifyEmailComponent,
41     ProfilComponent,
42     ChangePasswordComponent,
43     SidebarComponent,
44     NavebarComponent,
45     FooterComponent,
46     RegistrationComponent,
47     RouteIndicatoreComponent,
48   ],
49   imports: [
50     BrowserModule,
51     AppRoutingModule,
52     HttpClientModule,
53     ChartModule,
54     AngularFireModule.initializeApp(environment.firebaseConfig),
55     AngularFirestoreModule,
```

Figure 69: Le fichier app.module.ts

# BIBLIOGRAPHIE ET WEBOGRAPHIE

- [1] M. Diaw, S. Faye, W. Stichler, et P. Maloszewski, « Isotopic and geochemical characteristics of groundwater in the Senegal River delta aquifer: implication of recharge and flow regime », *Environ. Earth Sci.*, vol. 66, n° 4, p. 1011-1020, juin 2012, doi: 10.1007/s12665-010-0710-4.
- [2] MM. G. TIBALDESCHI, X. BOULENGER, et M. R. N. BA, « Sénégal - \_Projet\_d\_appui\_a\_la\_petite\_irrigation\_locale\_PAPIL\_- Rapport\_d'évaluation.pdf », juill. 2003. [https://www.afdb.org/fileadmin/uploads/afdb/Documents/Project-and-Operations/S%C3%A9negal\\_-\\_Projet\\_d\\_appui\\_a\\_la\\_petite\\_irrigation\\_locale\\_PAPIL\\_- Rapport\\_d\\_%E2%80%99%C3%A9valuation.pdf](https://www.afdb.org/fileadmin/uploads/afdb/Documents/Project-and-Operations/S%C3%A9negal_-_Projet_d_appui_a_la_petite_irrigation_locale_PAPIL_- Rapport_d_%E2%80%99%C3%A9valuation.pdf) (consulté le juill. 10, 2020).
- [3] C. DIOP, A. NIANG, A. SECK, P. SMITS, et S. TATO, « Renforcement des capacités de micro-irrigation pour l'intensification de l'horticulture - Zone des Niayes », juill. 2004. <http://www.fao.org/tempref/agl/IPTRID/hortica.pdf> (consulté le juill. 10, 2020).
- [4] « ref11x06\_3.pdf ». [https://www.ctc-n.org/sites/www.ctc-n.org/files/UNFCCC\\_docs/ref11x06\\_3.pdf](https://www.ctc-n.org/sites/www.ctc-n.org/files/UNFCCC_docs/ref11x06_3.pdf) (consulté le juill. 10, 2020).
- [5] S. T. Fall, A. S. Fall, I. Cissé, A. Badiane, C. A. Fall, et M. B. Diao, « Intégration horticulture - élevage dans les systèmes agricoles urbains de la zone des Niayes (Sénégal) », *Bull. APAD*, n° 19, Art. n° 19, juin 2000, Consulté le: juill. 10, 2020. [En ligne]. Disponible sur: <http://journals.openedition.org/apad/444>.
- [6] « Memoire Online - Le rôle du périmètre maraàcher de Keur Saàb Ndoye dans l'approvisionnement du marché central de Thiès en produits maraàchers (légumes) - ThÃ©ophile Marc NDIONE », *Memoire Online*. [https://www.memoireonline.com/01/12/5190/m\\_Le-rle-du-perimetre-maracher-de-Keur-Sab-Ndoye-dans-l-approvisionnement-du-marche-central5.html#toc8](https://www.memoireonline.com/01/12/5190/m_Le-rle-du-perimetre-maracher-de-Keur-Sab-Ndoye-dans-l-approvisionnement-du-marche-central5.html#toc8) (consulté le juill. 10, 2020).
- [7] « Le Ministre de l'Agriculture et l'Ambassadeur d'Espagne au Sénégal visitent Casamance ». [http://www.exteriores.gob.es/Embajadas/DAKAR/fr/Noticias/Pages/Articulos/20190109\\_NOT2.aspx](http://www.exteriores.gob.es/Embajadas/DAKAR/fr/Noticias/Pages/Articulos/20190109_NOT2.aspx) (consulté le juill. 10, 2020).
- [8] « Salinisation », *Wikipédia*. mai 14, 2020, Consulté le: juill. 10, 2020. [En ligne]. Disponible sur: <https://fr.wikipedia.org/w/index.php?title=Salinisation&oldid=170852407>.
- [9] O. news, « Sénégal : la salinisation des sols attriste les agriculteurs – Ouestaf, une information fiable et indépendante sur les questions qui traversent l'Afrique. » <https://ouestaf.com/?p=20239307> (consulté le juill. 10, 2020).
- [10] « p050fgestiondelirrigation.pdf ». <https://www.eijkelkamp.com/files/media/Brochures/p050fgestiondelirrigation.pdf> (consulté le juill. 10, 2020).
- [11] « Arrosage automatique », *Wikipédia*. mai 12, 2020, Consulté le: juill. 10, 2020. [En ligne]. Disponible sur: [https://fr.wikipedia.org/w/index.php?title=Arrosage\\_automatique&oldid=170785465](https://fr.wikipedia.org/w/index.php?title=Arrosage_automatique&oldid=170785465).
- [12] « Irrigation gravitaire traditionnelle », *agronomie*, févr. 27, 2016. <https://agronomie.info/fr/irrigation-gravitaire-traditionnelle/> (consulté le juill. 10, 2020).
- [13] « Les différents systèmes d'irrigation », *AQUA6*. [https://www.aqua6.info/blog/25\\_les-differents-systemes-d-irrigation.html](https://www.aqua6.info/blog/25_les-differents-systemes-d-irrigation.html) (consulté le juill. 10, 2020).
- [14] « a1336f06.pdf ». Consulté le: juill. 10, 2020. [En ligne]. Disponible sur: <http://www.fao.org/3/a1336f/a1336f06.pdf>.
- [15] A. LAGUEYRIE, « Arroser par aspersion », *Rustica.fr*. <https://www.rustica.fr/l-eau-jardin/arrosage-par-aspersion,4246.html> (consulté le juill. 10, 2020).
- [16] A. CISSE, « E53 - La micro-irrigation. Le procédé « goutte à goutte » », *Wikiwater*. <https://wikiwater.fr/E53-La-micro-irrigation-Le-procedede-goutte-a-goutte> (consulté le juill. 10, 2020).
- [17] « Cycle de vie d'un logiciel », *CommentCaMarche*. <https://www.commentcamarche.net/contents/473-cycle-de-vie-d-un-logiciel> (consulté le juill. 10, 2020).
- [18] A. VARATHARAJAH, « Cycle de vie d'un logiciel ». <https://web.maths.unsw.edu.au/~lafaye/CCM/genie-logiciel/cycle-de-vie.htm> (consulté le juill. 10, 2020).
- [19] « Cycle en spirale | SUPINFO, École Supérieure d'Informatique ». <https://www.supinfo.com/articles/single/2188-cycle-spirale> (consulté le juill. 10, 2020).
- [20] V. Messenger Rota, *Gestion de projet: vers les méthodes agiles*. Paris: Eyrolles, 2008.
- [21] C. ROELS, « UML, c'est quoi ? », *OpenClassrooms*. <https://openclassrooms.com/fr/courses/2035826-debutez-lanalyse-logicielle-avec-uml/2035851-uml-c-est-quoi> (consulté le juill. 10, 2020).

- [22] P. Roques, *UML 2 par la pratique: études de cas et exercices corrigés*. Paris: Eyrolles, 2006.
- [23] N. JUNG, « L'architecture MVC, qu'est ce que c'est ? | SUPINFO, École Supérieure d'Informatique ». <https://www.supinfo.com/articles/single/8729-architecture-mvc-qu-est-ce-que-c-est> (consulté le juill. 10, 2020).
- [24] « Firebase », *Wikipédia*. déc. 24, 2019, Consulté le: juill. 10, 2020. [En ligne]. Disponible sur: <https://fr.wikipedia.org/w/index.php?title=Firebase&oldid=165675352>.
- [25] « Cloud Firestore », *Firestore*. <https://firebase.google.com/docs/firestore?hl=fr> (consulté le juill. 10, 2020).
- [26] « Arduino Mega : Utilisation et fonctionnement », *Arduino France*, févr. 18, 2019. <https://www.arduino-france.com/review/arduino-mega-utilisation-et-fonctionnement/> (consulté le juill. 10, 2020).
- [27] « Module GSM SIM800L - Prise en main ». <https://letmeknow.fr/shop/fr/blog/94-tuto-module-gsm-sim800l-prise-en-main> (consulté le juill. 10, 2020).
- [28] « Fritzing », *Wikipédia*. juin 08, 2020, Consulté le: juill. 10, 2020. [En ligne]. Disponible sur: <https://fr.wikipedia.org/w/index.php?title=Fritzing&oldid=171811465>.
- [29] G. RARCHAERT, « Introduction à Angular | SUPINFO, École Supérieure d'Informatique ». <https://www.supinfo.com/articles/single/6124-introduction-angular> (consulté le juill. 10, 2020).
- [30] « Visual Studio Code, l'IDE open source que les devs s'arrachent ». <https://www.journaldunet.com/web-tech/developpeur/1204957-visual-studio-code-l-ide-open-source-que-les-devs-s-arrachent/> (consulté le juill. 10, 2020).
- [31] L. M. RIDHA et A. ABDERRAHMANE, « Etude et réalisation d'un Système d'irrigation automatique », 2018. <http://193.194.80.11/xmlui/bitstream/handle/123456789/2669/tesstet.pdf?sequence=1&isAllowed=y> (consulté le juill. 11, 2020).
- [32] B. Lebioda, « Les techniques d'irrigation (1): irrigation gravitaire », déc. 24, 2009. <http://www.canaux-et-territoire.info/d2/index.php?post/2009/12/24/38-les-techniques-d-irrigation-i-irrigation-gravitaire> (consulté le juill. 11, 2020).
- [33] « Goutteur - tous les goutteurs pour votre arrosage goutte à goutte ! ». <https://www.pompes-direct.com/arrosage/goutte-a-goutte/goutteur/index.html> (consulté le juill. 11, 2020).
- [34] « Tuyau micro poreux 25 M goutte à goutte spécial irrigation - Achat / Vente tuyau - buse - tête Tuyau micro poreux 25 M - Cdiscount ». <https://www.cdiscount.com/jardin/arrosage/tuyau-micro-poreux-25-m-goutte-a-goutte-special-ir/f-1632707-idm3662897011162.html> (consulté le juill. 11, 2020).