PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

Ministry of Higher Education and Scientific Research

ECHAHID HAMMA LAKHDAR UNIVERSITY - EL OUED

FACULTY OF EXACT SCIENCES

Computer Science department

End of Study Thesis

Presented for the Diploma of

ACADEMIC MASTER

Domain: **Mathematics and Computer Science**

Department: **Computer Science**

Specialty: **Artificial Intelligence and Distributed Systems**

Presented by:

- KAHLA MOHAMMED ELHABIB
- HARIZ BEKKAR MOHAMMED

Theme

SMART IRRIGATION

In front of jury: Supported in: 24-10- 2020

Dr.	Brahim Lejdel	MCA	President
M.	Chourouk Guettas	MAA	Reporter
M.	Othmani Samir	MAA	Supervisor

University year: 2019/2020

Acknowledgment

Every challenging job requires self-effort as well as the guidance of the elderly, especially those who have been very close to our hearts. I dedicate my humble effort to my father and mother, whose affection, love, encouragement and prays of day and night make me able to get such success and honor, day and night,

Along with all hardworking and respected Teachers

Abstract

In this dissertation, we suggest implementing the smart irrigation system. This system controls farm irrigation to ensure excellent performance, reduce farmers' manual efforts, and to provide the amount of water needed for plants, organize irrigation water use, and solve the problem of farms with large areas. For practical reasons, we suggest a mix between the automatic and manual systems based on fuzzy logic and equations for watering management. The automatic control of the system allows a smart way to manage the irrigation process scientifically, based on the sensors data collected from different parts of the farm. We use a simple algorithm to determine the region to be watered, and then we use fuzzy logic to get the right amount of water needed for plants (parcel). Then we use the irrigation management equations proposed by the World Food and Agriculture Organization to determine the length of time required for irrigation. Manual control is the alternative control mode where the human operator can learn about the climatic conditions of the farm parts through a website in the cloud and implement possible measures in the irrigation process. The operator or the human expert can obtain its climate data for the farm remotely stored in the cloud.

Résumé

Dans ce mémoire, nous proposons de mettre en œuvre un système d'irrigation intelligent. Ce système contrôle l'irrigation des fermes pour garantir d'excellentes performances, réduire les efforts manuels des agriculteurs et fournir la quantité d'eau nécessaire aux plantes, organiser l'utilisation de l'eau d'irrigation et résoudre le problème des fermes de grandes superficies. Pour des raisons pratiques, nous suggérons d'hybrider les systèmes automatiques et manuels basés sur la logique floue et les équations de gestion de l'irrigation. Le contrôle automatique du système permet d'une manière intelligente de gérer scientifiquement le processus d'irrigation, sur la base des données de capteurs collectées dans différentes parties de la ferme. Nous utilisons un algorithme simple pour déterminer la région à irriguer, puis nous utilisons la logique floue pour obtenir la bonne quantité d'eau nécessaire aux plantes (parcelle). Ensuite, nous utilisons les équations de gestion de l'irrigation proposée par l'Organisation mondiale pour l'alimentation et l'agriculture pour déterminer la durée nécessaire à l'irrigation. Le contrôle manuel est le mode de contrôle alternatif où l'opérateur humain peut se renseigner sur les conditions climatiques des parties de la ferme via un site Web dans le Cloud et mettre en œuvre des mesures possibles dans le processus d'irrigation. L'opérateur ou l'expert humain peut obtenir ses données climatiques pour la ferme stockées à distance dans le Cloud.

الملخص

نقترح في هذه المذكرة تنفيذ نظام المزرعة الذكية. يتحكم هذا النظام في ري المزرعة لضمان الأداء الجيد، وتقليل جهود المزارعين وتوفير كمية الماء اللازمة للمزروعات، وتقنين والتحكم في استخدام مياه الري وحل مشكلة مساحة المزارع الكبيرة. لأسباب عملية نقترح مزيجا بين النظام الالي واليدوي استنادا المنطق الضبابي ومعادلات خاصة بإدارة السقي، يتيح التحكم الالي للنظام الاستجابة بطريقة ذكية لإدارة عملية الري بطريقة علمية، استنادا لبيانات المستشعرات الموجودة في أجزاء المزرعة، نستخدم خوارزمية بسيطة في تحديد الجزء الواجب سقيه ثم نستخدم المنطق الضبابي للحصول على كمية الماء اللازمة للنبات ومن ثم نستخدم معادلات ادارة الري المقترحة من طرف منظمة التغذية والزراعة العالمية لتحديد المدة الزمنية اللازمة للسقي. التحكم اليدوي هو وضع التحكم البديل حيث يمكن للمشغل البشري التعرف على الظروف المناخية للأجزاء من خلال موقع الويب الموجود في السحابة وتنفيذ الإجراءات الممكنة في عملية الري، وأيضا يمكن للمشغل أو الخبير البشري الحصول بيانات مناخية للمزرعة عن بعد المخزنة في السحابة.

Table of Contents

List of Figures

List of Tables

General Introduction

General Introduction

Algeria is among the agricultural countries where agriculture is witnessing rapid growth, especially in desert areas that use groundwater as the only source of irrigation water, and this has led to an irrational consumption of groundwater, which may be a threat in the long term for us and for future generations.

Based on this problem, we had to automate the irrigation system in order to conserve water and make it used rationally and also increase agricultural productivity to improve the water conditions of plants with continuous monitoring. Here we propose a smart watering system that controls watering based on scientific methods and data such as climate data and soil moisture, and also supports farms with vast areas using the Internet of things, cloud service, and fuzzy logic algorithm.

This dissertation aims to create an intelligent, connected irrigation system that controls large parts of the farm. Because of COVID-19, we could not get some sensors, Arduino, and accessories from the Provider. Thus, we cannot construct a real, intelligent, and connected irrigation system in this project. For realizing our project, we use the simulation platform.

We divided our work into four chapters, where we talked in the first chapter about watering its types, the Internet and the most important characteristics and the cloud and its different kinds. In the second chapter, we talked about the fuzzy logic in detail, and in the third chapter we explained the proposed design for the general system. In the last chapter, we explained the tools and devices used in the implementation of the system with screenshots of the obtained model for the final system along with their descriptions.

Chapter 1: General information about irrigation and the Internet of things

1. Introduction

Farmers face a variety of challenges due to outdated agricultural techniques. Unplanned use of water inadvertently results in excess water usage, and This suggests that there is an urgent need to develop systems that prevent water overuse without imposing pressure on farmers.

The Internet of things (IoT) is a general term used to describe a network of devices capable of sending and receiving data in real time that streamlines the recording and processing of data and/or functioning of automated programs. This technology has entered the agriculture industry as a means of creating a non-manual, integrated system of sensors that can record quantitative measurements of soil condition, crop growth, and weather patterns.

2. Irrigation System

2.1. General definition of Irrigation

A technique that involves artificially providing crops with water to enable them to grow. This technique is used in farming to enable plants to grow when there is not enough rain, particularly in arid areas. It is also used in less arid regions to provide plants with the water they need when seed setting. (Water report 22: Deficit Irrigation Practices, FAO).

2.2. Smart Irrigation

Generally speaking, smart irrigation refers to the implementation of advanced technologies and connectivity equipment to monitor, analyze and control outdoor water usage. In most cases, this involves the installation of "smart controllers" which can automatically adjust irrigation applications to suit weather and soil conditions. These control systems are often supplemented with machine-to-machine communication, artificial intelligence and machine-learning applications that allow for more precise management of irrigation scheduling. According to the Alliance for Water Efficiency, most smart irrigation technologies fall under two classifications [15]:

- **Sensor-based control:** This method leverages real-time measurements from locally installed sensors to automatically adjust irrigation timing to the exact temperature, rainfall, humidity and soil moisture present in a given environment. This data is also supplemented with historic weather information to ensure farmers are able to anticipate unfavorable conditions.
- **Signal-based control:** Unlike sensor-based controls, these smart irrigation systems rely on weather updates transmitted by radio, telephone or web-based applications. These signals

are typically sent from local weather stations to update the "evapotranspiration rate" of the irrigation controller.

2.3. Types of Irrigation Systems

There are many different types of irrigation systems, depending on how the water is distributed throughout the field. Some common types of irrigation systems include [16]:

2.3.1. Surface irrigation

Water is distributed over and across land by gravity, no mechanical pump involved (Figure 1).

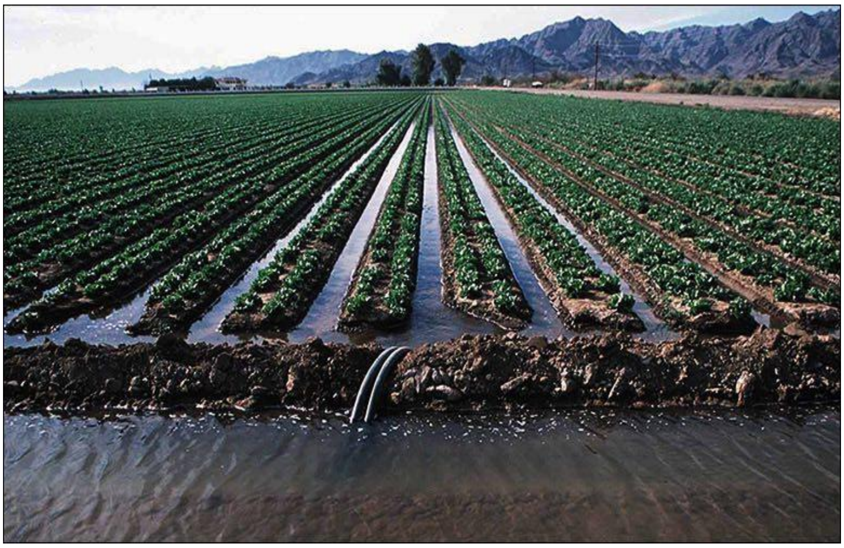

Figure 1: Surface irrigation

2.3.2. Drip irrigation

A type of localized irrigation in which drops of water are delivered at or near the root of plants. In this type of irrigation, evaporation and runoff are minimized (Figure 2).

Figure 2: Drip irrigation

2.3.3. Sprinkler irrigation

Water is distributed by overhead high-pressure sprinklers or guns from a central location in the field or from sprinklers on moving platforms (Figure 3).

Figure 3: Sprinkler irrigation

2.3.4. Center pivot irrigation

Water is distributed by a system of sprinklers that move on wheeled towers in a circular pattern. This system is common in flat areas of the United States (Figure 4).

Figure 4: Center pivot irrigation

3. Evapotranspiration

Evapotranspiration, often shortened to ET, is the process of transferring moisture from the earth into the atmosphere. Put simply, evaporation occurs when water vapor leaves the soil or a plant's surface. Transpiration involves the passage of water through a plant, from its roots through its vascular system. The sum of evaporation and transpiration is evapotranspiration (ET) [17].

3.1.1. Evaporation

Evaporation is the process whereby liquid water is converted to water vapour (vaporization) and removed from the evaporating surface (vapour removal). Water evaporates from a variety of surfaces, such as lakes, rivers, pavements, soils and wet vegetation [1].

3.1.2. Transpiration

Transpiration consists of the vaporization of liquid water contained in plant tissues and the vapour removal to the atmosphere. Crops predominately lose their water through stomata. These are small openings on the plant leaf through which gases and water vapour pass. The water, together with some nutrients, is taken up by the roots and transported through the plant. The vaporization occurs within the leaf, namely in the intercellular spaces, and the vapour exchange with the atmosphere is controlled by the stomatal aperture. Nearly all water taken up is lost by transpiration and only a tiny fraction is used within the plant [1].

3.2. Factors affecting evapotranspiration

Weather parameters, crop characteristics, management and environmental aspects are factors affecting evaporation and transpiration. The related ET concepts presented in (Figure 5) are discussed in the section on evapotranspiration concepts [1].

3.2.1. Weather parameters

The principal weather parameters affecting evapotranspiration are radiation, air temperature, humidity and wind speed. Several procedures have been developed to assess the evaporation rate from these parameters (Figure 5). The evaporation power of the atmosphere is expressed by the reference crop evapotranspiration (ETo) [1].

3.2.2. Crop factors (K_c)

The crop type, variety and development stage should be considered when assessing the evapotranspiration from crops grown in large, well-managed fields. Differences in resistance to transpiration, crop height, crop roughness, reflection, ground cover and crop rooting characteristics result in different ET levels in different types of crops under identical environmental conditions [1].

3.2.3. Management and environmental conditions

Factors such as soil salinity, poor land fertility, and limited application of fertilizers, the presence of hard or impenetrable soil horizons, the absence of control of diseases and pests and poor soil management may limit the crop development and reduce the evapotranspiration. Other factors to be considered when assessing ET are ground cover, plant density and the soil water content [1].

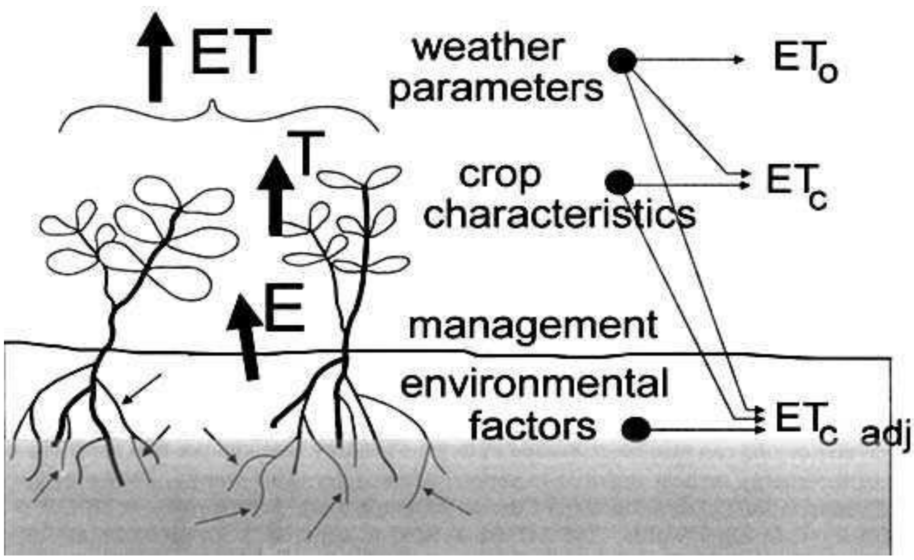

Figure 5: Factors affecting evapotranspiration with reference to related ET concepts

3.3. Evapotranspiration concepts

Distinctions are made (Figure 6) between reference crop evapotranspiration (ETo), crop evapotranspiration under standard conditions (ETc) and crop evapotranspiration under non-standard conditions (ETc adj) [1].

3.3.1. Reference crop evapotranspiration or reference evapotranspiration

Denoted as ETo or ETref, is the estimation of the evapotranspiration from the "reference surface".

The FAO Penman-Monteith method to estimate ET_o:

$$ET_o = \frac{0.408\Delta(R_n - G) + \gamma \frac{900}{T+273} u_2 (e_s - e_a)}{\Delta + \gamma(1 + 0.34u_2)} \qquad \text{...... Equation 1}$$

Where:

- ET_o reference evapotranspiration [mm day^{-1}]
- R_n net radiation at the crop surface [MJ m^{-2} day^{-1}],
- G soil heat flux density [MJ m^{-2} day^{-1}],
- T mean daily air temperature at 2 m height [°C],
- u_2 wind speed at 2 m height [m s^{-1}],
- e_s saturation vapour pressure [kPa],
- e_a actual vapour pressure [kPa],
- $e_s - e_a$ saturation vapour pressure deficit [kPa],
- Δ slope vapour pressure curve [kPa $°C^{-1}$],
- γ Psychrometric constant [kPa $°C^{-1}$].

3.3.2. The crop evapotranspiration under standard conditions (ETc)

The crop evapotranspiration under standard conditions, denoted as ETc, is the evapotranspiration from disease-free, well-fertilized crops, grown in large fields, under optimum soil water conditions, and achieving full production under the given climatic conditions.

The crop characteristics into the K_c coefficient:

$ET_c = K_c\ ET_o$ (58) ...... Equation 2

Where:

- ET_o reference evapotranspiration [mm day^{-1}]
- K_c Crop factors

3.3.3. Crop evapotranspiration under non-standard conditions (ETc adj)

The crop evapotranspiration under non-standard conditions (ETc adj) is the evapotranspiration from crops grown under management and environmental conditions that differ from the standard conditions. When cultivating crops in fields, the real crop evapotranspiration may deviate from ETc due to non-optimal conditions such as the presence of pests and diseases, soil salinity, low soil fertility, water shortage or waterlogging. This may result in scanty plant growth, low plant density and may reduce the evapotranspiration rate below ETc.

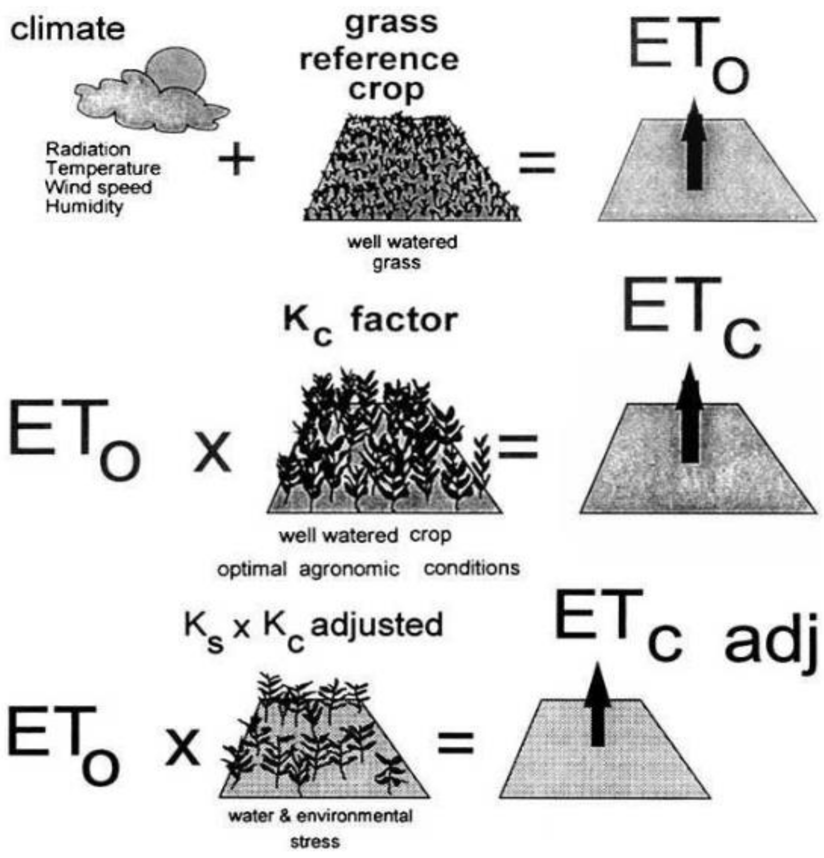

Figure 6: Reference (ETo), crop evapotranspiration under standard (ETc) and non-standard conditions (ETc adj)

3.4. Water amount equation

To calculate the amount of water needed for a given plot of land, we use the following equation [2]:

$WR = ET_c * A$ ...... Equation 3

Where:

- WR water requirement, liters per day for the area "A",
- ET_c crop evapotranspiration
- A = landscape area, m^2

4. Internet of Things (IoT)

4.1. Definition of IoT:

The Internet of Things (IoT) is a system of interrelated computing devices (Figure 7), mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction [3].

Figure 7: IoT

Internet of Things (IoT) can play a vital role in increasing productivity of farm. With the use of smart controllers, and sensors in farms and farming machineries, IoT-based smart irrigation offers real-time data and analytics, which enables farmers or growers to take corrective actions at right time.

4.2. Characteristics of IoT

The fundamental characteristics of the IoT are as follows [4]:

- **Interconnectivity:** With regard to the IoT, anything can be interconnected with the global information and communication infrastructure.

- **Things-related services:** The IoT is capable of providing thing-related services within the constraints of things, such as privacy protection and semantic consistency between physical things and their associated virtual things. In order to provide thing-related services within the constraints of things, both the technologies in physical world and information world will change.

- **Heterogeneity:** The devices in the IoT are heterogeneous as based on different hardware platforms and networks. They can interact with other devices or service platforms through different networks.

- **Dynamic changes:** The state of devices change dynamically, e.g., sleeping and waking up, connected and/or disconnected as well as the context of devices including location and speed. Moreover, the number of devices can change dynamically.

- **Enormous scale:** The number of devices that need to be managed and that communicate with each other will be at least an order of magnitude larger than the devices connected to the current Internet. Even more critical will be the management of the data generated and their interpretation for application purposes. This relates to semantics of data, as well as efficient data handling.

- **Safety:** As we gain benefits from the IoT, we must not forget about safety. As both the creators and recipients of the IoT, we must design for safety. This includes the safety of our personal data and the safety of our physical well-being. Securing the endpoints, the networks, and the data moving across all of it means creating a security paradigm that will scale.

- **Connectivity:** Connectivity enables network accessibility and compatibility. Accessibility is getting on a network while compatibility provides the common ability to consume and produce data.

4.3. Architecture of smart agriculture

While there are myriad bits that build a complete end-to-end IoT architecture, this architecture simplifies it down to three fundamental building blocks (Figure 8) [5]: Perception layer Sensors, actuators and edge devices that interact with the environment. Network Layer Discovers, connects and translates devices over a network and in coordination with the application layer. Application Layer, Data processing and storage with specialized services and functionality for users.

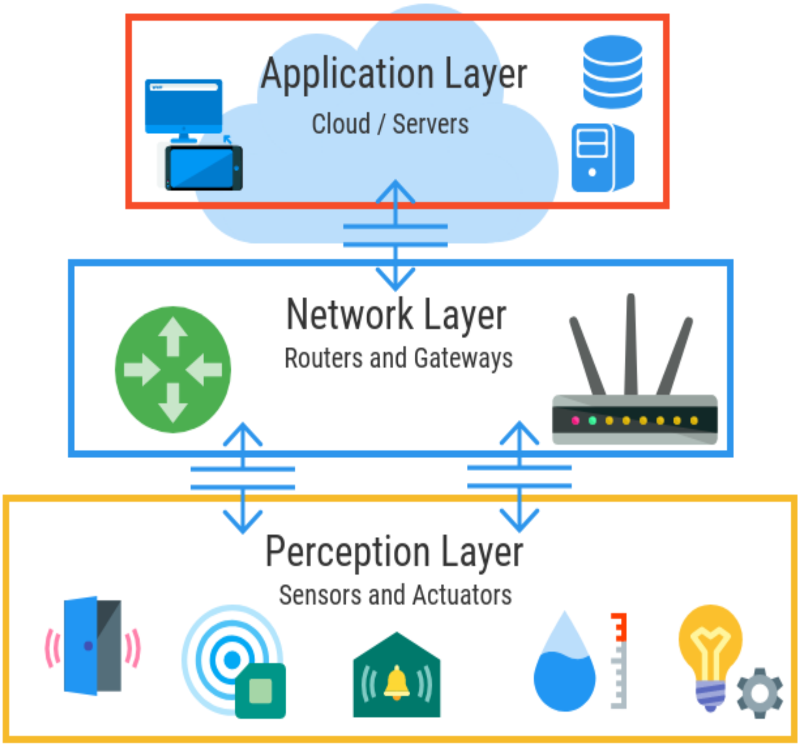

Figure 8: The fundamental Three Layer IoT Architecture

5. Cloud computing

5.1. What is cloud computing

Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining physical data centers and servers, you can access technology services, such as computing power, storage, and databases, on an as-needed basis from a cloud provider [19].

5.2. Benefits of cloud computing

- **Agility**

The cloud gives you easy access to a broad range of technologies so that you can innovate faster and build nearly anything that you can imagine. You can quickly spin up resources as you need them–from infrastructure services, such as compute, storage, and databases, to Internet of Things, machine learning, data lakes and analytics, and much more.

- **Elasticity**

With cloud computing, you don't have to over-provision resources up front to handle peak levels of business activity in the future. Instead, you provision the amount of resources that you actually need. You can scale these resources up or down to instantly to grow and shrink capacity as your business needs change.

Cost savings

The cloud allows you to trade capital expenses (such as data centers and physical servers) for variable expenses, and only pay for IT as you consume it. Plus, the variable expenses are much lower than what you would pay to do it yourself because of the economies of scale.

Deploy globally in minutes

With the cloud, you can expand to new geographic regions and deploy globally in minutes, so you can deploy your application in multiple physical locations with just a few clicks. Putting applications in closer proximity to end users reduces latency and improves their experience.

5.3. Types of cloud computing

The three main types of cloud computing include Infrastructure as a Service, Platform as a Service, and Software as a Service. Each type of cloud computing provides different levels of control, flexibility, and management so that you can select the right set of services for your needs [19].

Infrastructure as a Service (IaaS)

IaaS contains the basic building blocks for cloud IT. It typically provides access to networking features, computers (virtual or on dedicated hardware), and data storage space. IaaS gives you the highest level of flexibility and management control over your IT resources. It is most similar to the existing IT resources with which many IT departments and developers are familiar.

Platform as a Service (PaaS)

PaaS removes the need for you to manage underlying infrastructure (usually hardware and operating systems), and allows you to focus on the deployment and management of your applications. This helps you be more efficient as you don't need to worry about resource procurement, capacity planning, software maintenance, patching, or any of the other undifferentiated heavy lifting involved in running your application.

Software as a Service (SaaS)

SaaS provides you with a complete product that is run and managed by the service provider. In most cases, people referring to SaaS are referring to end-user applications (such as web-based email). With a SaaS offering, you don't have to think about how the service is maintained or how the underlying infrastructure is managed. You only need to think about how you will use that particular software.

6. Conclusion

In this chapter, we first talked about irrigation in general, then about different watering methods. The objective being to automate the control of a system comprising smart irrigation, Hence, we used IoT because it is the most reliable approach for monitoring and predicting critical variables such a-s water quality, ambient temperature and moisture, and soil condition, which impacts the crop quality, yield, and the returns. Thus, IoT-based smart irrigation practice has the potential to transform the traditional irrigation method by making it more efficient and productive.

Chapter 2: Fuzzy logic

1. Introduction

The classical methods of automation have been widely applied in many problems of industrial regulation. However, most physical systems have non-linearities and their parameters are often poorly understood and / or variable over time. For the control of such classes of systems, conventional methods of automation have shown their limits in terms of stabilization and performance.

With the development of digital computers, automation engineers are starting to take an interest in new control approaches such as adaptive control, predictive control, robust control, as well as techniques based on artificial intelligence, among these; fuzzy logic control.

An irrigation controller is the brain of an entire irrigation system, it directs water to plants, and allows the farmer or gardener to achieve optimal results, using the minimum amount of water.

In this chapter we will present a solution to the problem of managing irrigation systems by developing a controller based on the fuzzy approach.

2. Fuzzy logic

Fuzzy logic is an extension of Boolean logic by Lotfi Zadeh in 1965 based on the mathematical theory of fuzzy sets, which is a generalization of the classical set theory. By introducing the notion of degree in the verification of a condition, thus enabling a condition to be in a state other than true or false, fuzzy logic provides a very valuable flexibility for reasoning, which makes it possible to take into account inaccuracies and uncertainties [6].

3. Fuzzy Sets

Classical sets involve exactly defined values. This means that the Universe of Discourse is split into two groups—members and non-members.

Therefore, you cannot say that any member has a partial membership. For example, if you are pressing brakes or releasing them, these processes can be represented by 1 or 0. With Fuzzy Sets, on the other hand, you can have values in between as well. Therefore, you can say that the Fuzzy Sets have a degree of membership between 0 and 1. For example, you can have values like {0, 0.3, 0.5, 0.7, 1}. The 1 means a full brake, 0.7 means a little less brake, 0.5 means half the pressure, 0.3 means very little pressure, and 0 means no pressure. In the real world, you rarely see classical sets in action. You deal with the values represented by Fuzzy Sets.

There are several properties associated with Fuzzy Sets. The next sections explain them [7]:

- Law of Commutativity
- Law of Associativity
- Law of Distributivity
- Idempotent Law
- Involution Law
- Law of Transitivity
- …etc.

Law of Commutativity

The union or intersection of two sets produces the same the result, regardless of which set you list first. That means you can take A first or B first, but the results will always be the same.

$$(A \cup B) = (B \cup A)$$

$$(A \cap B) = (B \cap A)$$

Law of Associativity

Say you have three sets. The Law of Associativity says that a union or intersection of the first two sets with the third one is equal to a union or intersection of the last two sets with the first one.

$$(A \cup B) \cup C = A \cup (B \cup C)$$

$$(A \cap B) \cap C = A \cap (B \cap C)$$

Law of Distributivity

The Law of Distributivity says that a union of the first set with an intersection of the last two sets is equal to a union of the first set with the second, a union of the first set with the third, and then an intersection of both outputs. This same rule applies when you interchange the union and intersection operations.

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

Idempotent Law

The Idempotent Law states that an intersection or union of a set with the same set is the same set itself.

$$A \cup A = A$$

$$A \cap A = A$$

Identity Law

If Φ is an empty set and E is a universe of discourse, the Identity Law states that:

$$A \cup \Phi = A$$

$$A \cup E = E$$

$$A \cap \Phi = \Phi$$

$$A \cap E = A$$

In other words:

• A union of a set with an empty set leads to the same set

• An intersection of a set with an empty set leads to an empty set

• A union of a set with a Universe of Discourse leads to a Union of Discourse

• An intersection of a set with a Universe of Discourse leads to the same set

Law of Transitivity

The Law of Transitivity states that if A is a subset of B and B is a subset of C, A will be a subset of C.

If $A \subseteq B$, $B \subseteq C$, then $A \subseteq C$

Now, before you learn about the operations that can be applied to Fuzzy Sets, you need to understand the concept of membership functions.

4. Membership Functions

In the previous section, we showed that instead of having crisp values of 0 and 1, each element can be mapped to a value between 0 and 1. Each value is called the degree of membership and is represented by a curve, which depicts a function called a membership function. The value is called the membership value. In Figure 9, you can see the difference between crisp and Fuzzy Sets [7].

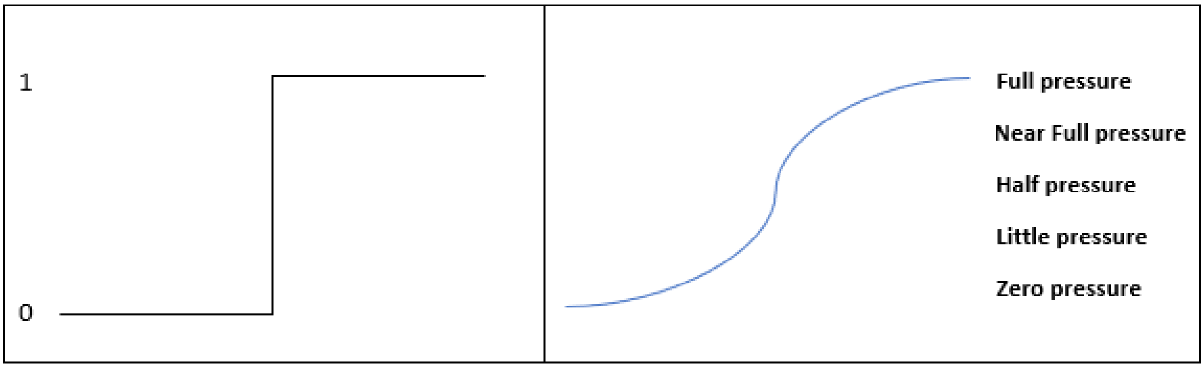

Figure 9: Difference between Fuzzy and crisp set representations

There are different types of membership functions, which are covered in detail in the next chapter. For now, let's list some of them and look at the curves that they represent (Figure 10,11,12,13).

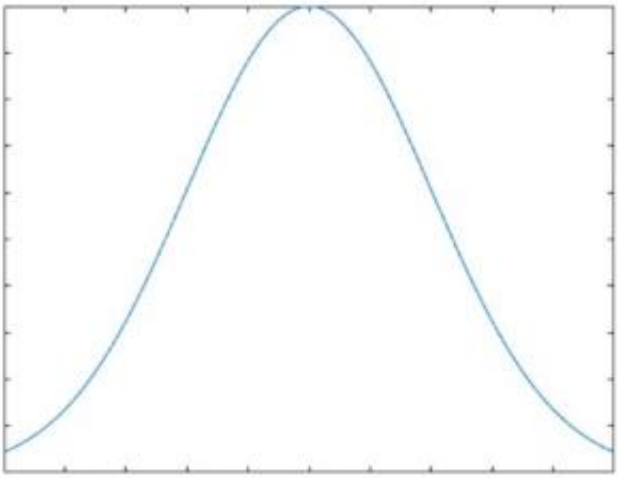

***Figure 10: Gaussian membership* function**

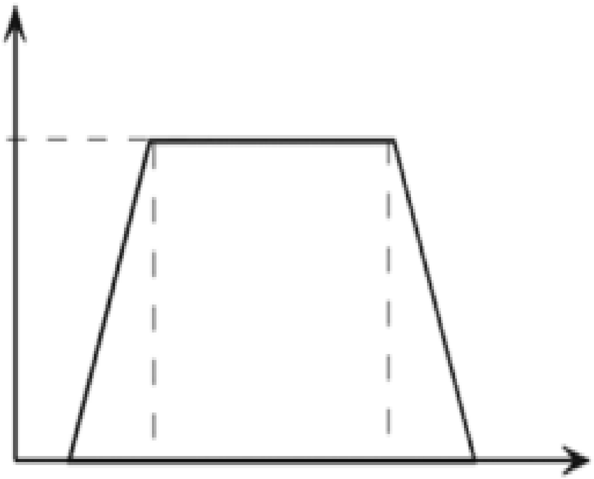

Figure 11: Trapezoidal membership function

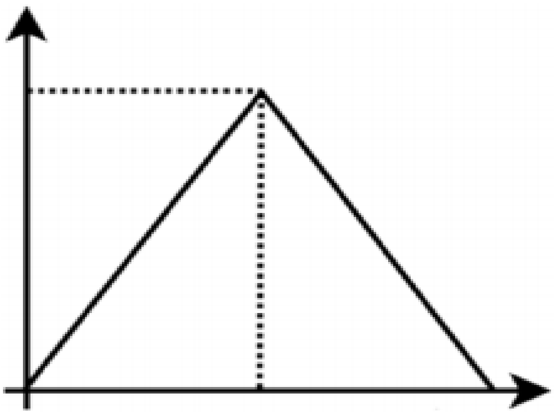

Figure 12: Triangular membership function

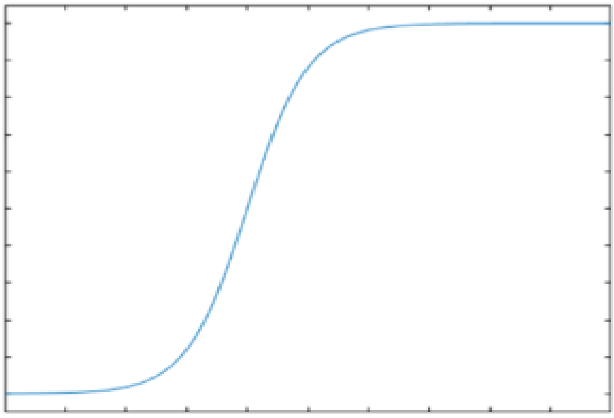

Figure 13:Sigmoidal membership function

5. Fuzzy Set Operations

Now that you have learned about the concept of membership functions, it's time to look at some of the operations that can be done with Fuzzy Sets. This section discusses the following operations [7]:

- Union
- Intersection
- Complement
- Product
- Equality
- Power
- Difference
- Disjunctive

Sum Suppose there are two Fuzzy Sets, A and B, having a membership value as μA(x) and μB(x), where X is the Universe of Discourse. Based on this information, the following sections consider each operation in detail.

Union

The union of two Fuzzy Sets A and B is a new Fuzzy Set, A ∪ B, also on X with a membership function defined as follows:

$$\mu_{(A \cup B)} = max\left(\mu_A(x), \mu_B(x)\right) = \mu_A(x) \wedge \mu_B(x)$$

Intersection

An intersection of Fuzzy Sets A and B is a new Fuzzy Set, A ∩ B, also on X whose membership function is defined by:

$$\mu_{(A \cap B)} = min\left(\mu_A(x), \mu_B(x)\right) = \mu_A(x) \vee \mu_B(x)$$

Complement

The complement of a Fuzzy Set A is A with this membership function:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

Product

The product of two Fuzzy Sets A and B is a new Fuzzy Set, A.B, with this membership function:

$$\mu_{A.B}(x) = \mu_A(x) \cdot \mu_B(x)$$

Difference

The difference of two Fuzzy Sets A and B is a new Fuzzy Set, A-B, which is defined as

$$A - B = (A \cap B)$$

Disjunctive Sum

The disjunctive sum is the new Fuzzy Set defined as follows:

$$A \oplus B = (\overline{A} \cap B) \cup (A \cap \overline{B})$$

Power

The alpha power of a Fuzzy Set A is a new Fuzzy Set Aα, whose membership function is as follows:

$$\mu_A^{\alpha}(x) = [\mu_A(x)]^{\alpha}$$

6. The steps of fuzzy logic

To implement fuzzy logic technique to a real application requires the following three steps (Figure 14) [8]:

Fuzzification

Convert classical data or crisp data into fuzzy data or Membership Functions (MFs)

Fuzzy Inference Process

Combine membership functions with the control rules to derive the fuzzy output

Defuzzification

Use different methods to calculate each associated output and put them into a table: the lookup table. Pick up the output from the lookup table based on the current input during an application

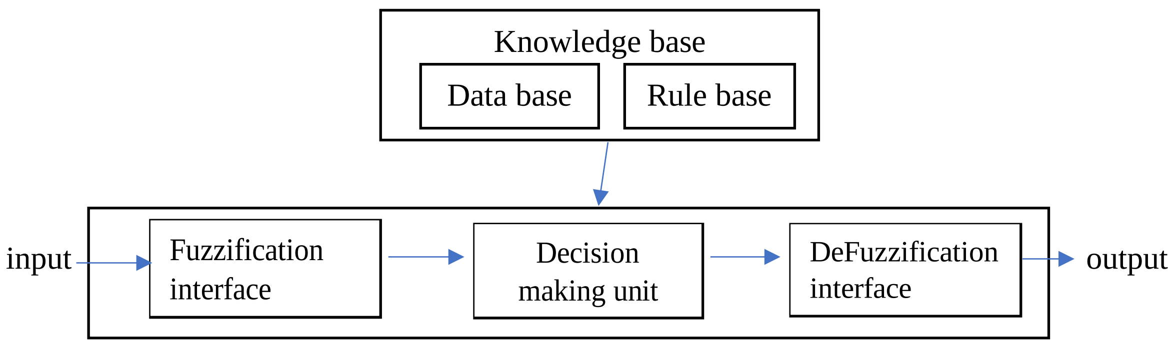

Figure 14: block diagram of fuzzy control

7. Conclusion

In this chapter, we introduced the basic theory of fuzzy logic. We have presented the structure of a fuzzy system which consists of three essential parts: fuzzification, inference and defuzzification. In the next chapter, we will present the principles of the proposed solution and the overall system architecture.

Chapter 3: Conceptual study

1. Introduction

The design phase is critical in determining the technical choices and preparing the implementation, it describes the solution (how the problem is solved), and it must serve as a support for the implementation and maintenance.

We used fuzzy logic to design a controller taking charge of the management of an irrigation system to track and regularize water levels.

In this chapter, we will design a connected, intelligent irrigation system. It controls large parts of the farm and how the irrigation process is divided into parts of the field, from which we will look at the overall structure of the system.

2. Related Work

Here we introduce some of the modern works and technologies that have been used for smart watering. So, we can take advantage of previous works. From a technical point of view, we will mention two works close to ours.

2.1. Design of Fuzzy Controller Based on Penman-Monteith Equation for Drip Irrigation

It is based on the equation Monteith-Penman equation to calculate the amount of water evaporated in the atmosphere and the amount of water lost from the soil [P. S. Vikhe…, 9/2019] [9].

Where the formula is based on the most important climatic factors, namely (Figure 15):

- Wind speed
- Temperature
- The humidity of the air
- Clear sunlight

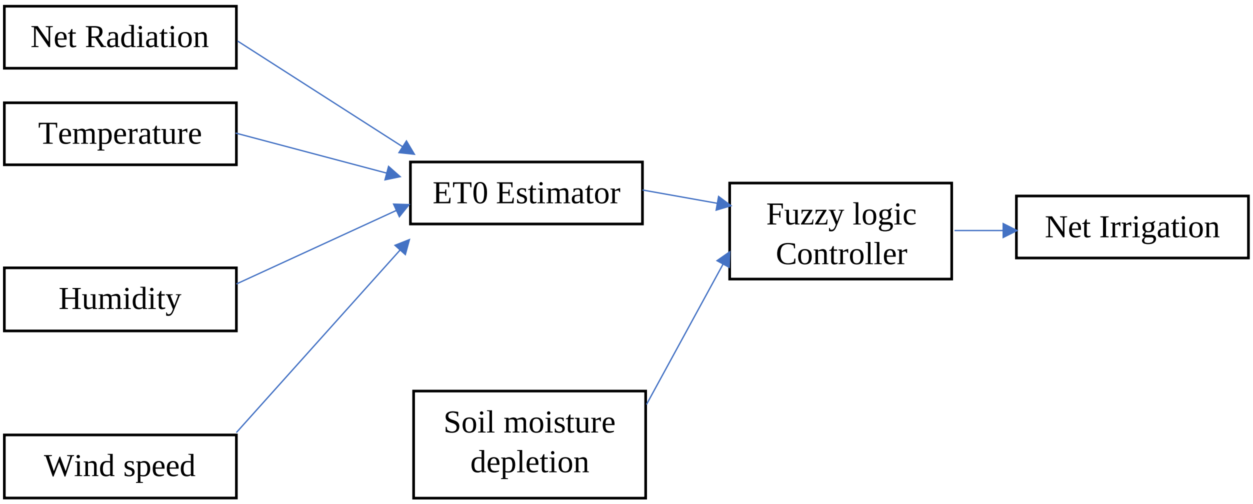

Figure 15: Schematic representation of proposed model for irrigation controller

When the soil water loss rate reaches 40%, the program starts with average humidity, air temperature, solar radiation, and wind speed to drain the soil moisture. Based on these intermediate data, the current yield of ETc is calculated to estimate the net irrigation (mm), to 60%, after which the system decides the amount of watering water and then counting the watering time.

2.2. A fuzzy logic-based irrigation management system in arid regions applied to the State of Qatar

This article also [Farid Touati…, 11/2012.] focuses on the equation Monteith-Penman equation and soil moisture. Still, there is an obstacle, and it must wait half an hour to extract the groundwater for irrigation. It is waiting for the humidity of up to 17% to enter the operation of the fuzzy system to obtain the amount of watering the water. Upon completion, from the process of watering, it sends data via ZigBee and to perform data analysis on a computer or other device (Figure 17) [10].

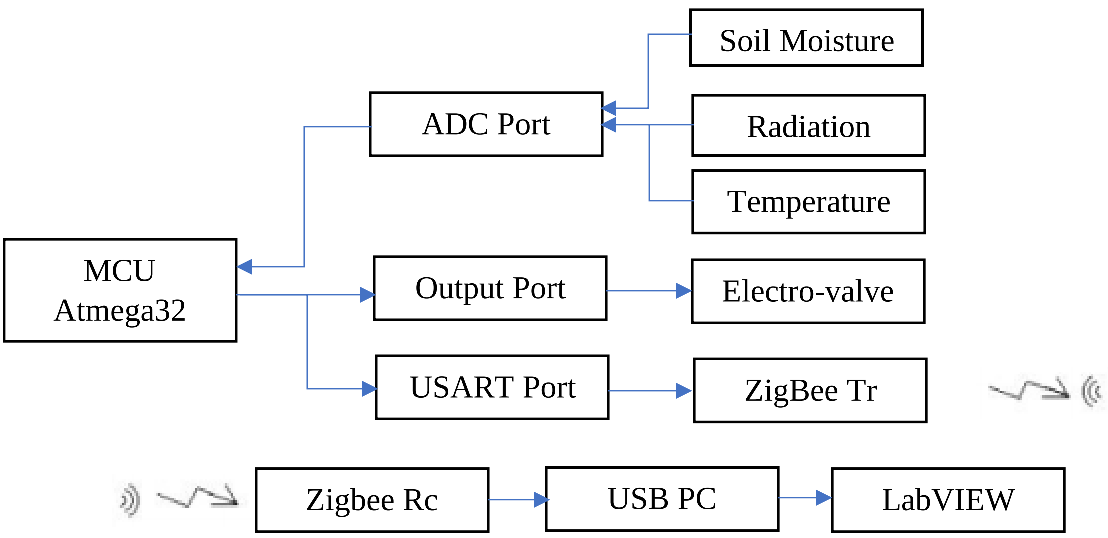

Figure 16: System structure (control and monitoring parts)

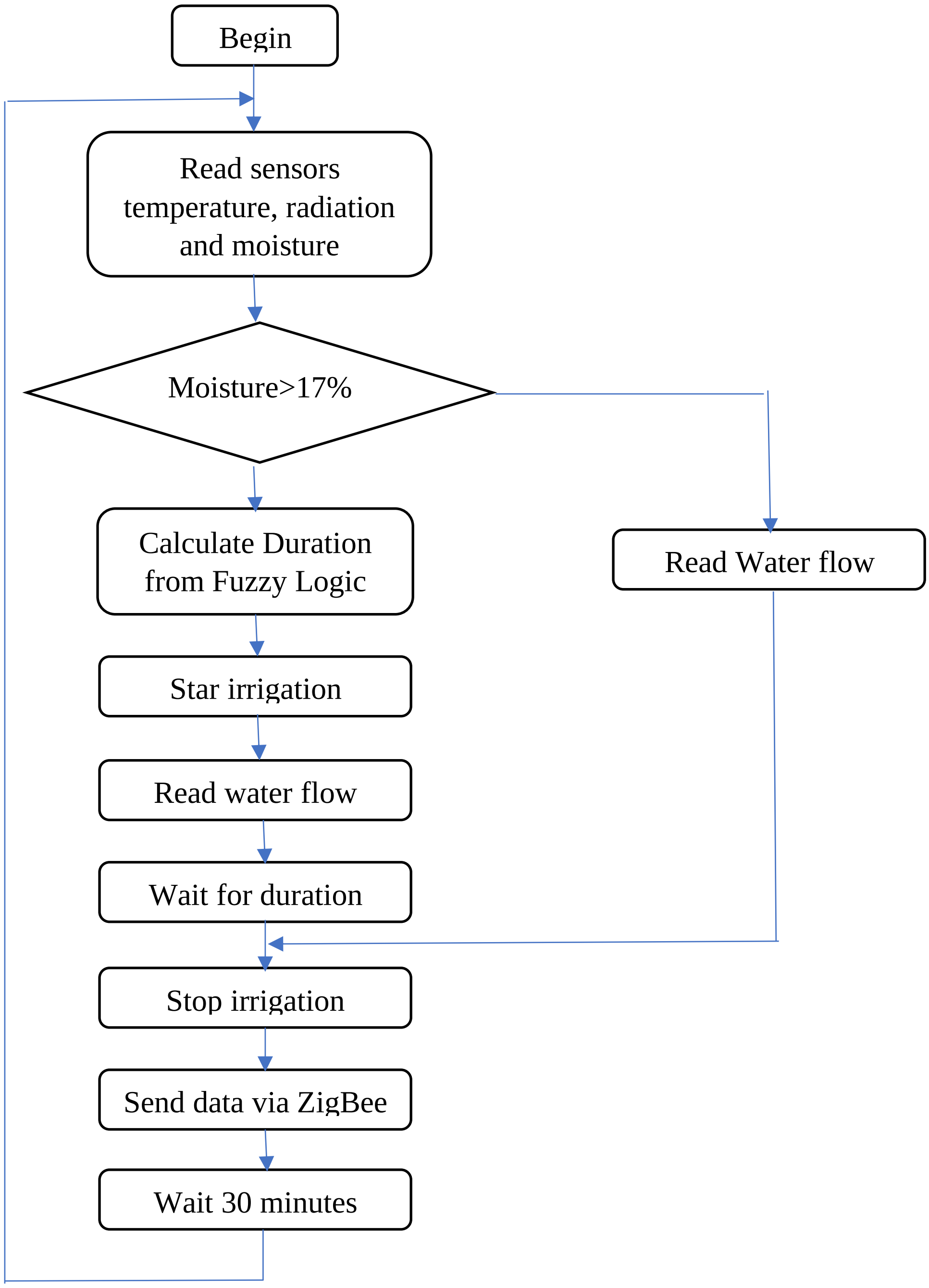

Figure 17: Flow chart software

After studying the literature, we found that the previous studies relied on several climatic and soil factors, with neglecting one of the most important factors, which is the plant itself, where the stages of plant development and the consumption of water were absent in each phase.

Also, the structure of the system is challenging to apply it to large areas, where it faces a problem in sending the information to the processing center. And starting from here, we begin to present the goal of our project.

3. Contribution

Based on previous works, we will design a system that must make use of climatic, soil and plant factors to evaluate and control the amount of water irrigation. This system supports irrigation in small segments to eliminate the problem of a large area of the farm. Also, it supports automatic and manual watering to give the administrator more control over the watering process when needed.

4. Objectives and principle of the proposed solution

Smart farm depends on money and technology in agriculture and crop production. The most important presented problems are:

The scheduling of irrigation, where we find there is a large waste in used water.
Also, the inefficiency of irrigation and the great efforts of its management, especially large farms.

4.1. Objective of the project

Our attempt in this project is to create an intelligent, connected irrigation system. It controls large parts of the farm, and data is collected from the sensors placed in different parts of the farm. The data is processed to obtain the information that help us make the appropriate decision, which give us insight into the situation:

- Control ensures automatic parcels irrigation where the human factor can interfere with the control process through an appropriate interface.
- Ensure complete control over the watering process of the parcels over large areas and ensure that the information reach the data processing center.
- Collect all data from parcels of the farm and its environment remotely via the cloud.
- Monitor and create statistics and strategic studies remotely and make forecasts based on these data.

Our system offers the user two control options: Automatic control using the intelligent system and another manual control of the farm via a computer.

4.2. Principle of the proposed solution

Our system is composed of three parts (Figure 18):

- The first part corresponds to the physical aspects and actuation subsystem; this subsystem works as a physical interface between the rest of the system and the farm. It recollects regularly data about temperature, humidity, wind speed, etc. from the environment and the parcels inside the farm. Also, it makes it possible to translate system commands into real actions on the farm.
- The second part is the central and essential part of the proposed system. It is designed to link the first and third parts to exchange data and commands between them as a Gateway. It also receives data from parts of the farm and sends it to the cloud (website). The data is saved for study, analysis and irrigation decision-making to improve the irrigation system.
- The third part is implemented as a dynamic website hosted on the cloud. It makes it possible to monitor the farm and generate several types of statistics for climatic factors and manage manual irrigation remotely. Thus, this site can be accessed from any device connected to the Internet such as smartphones, tablets, computers, etc.

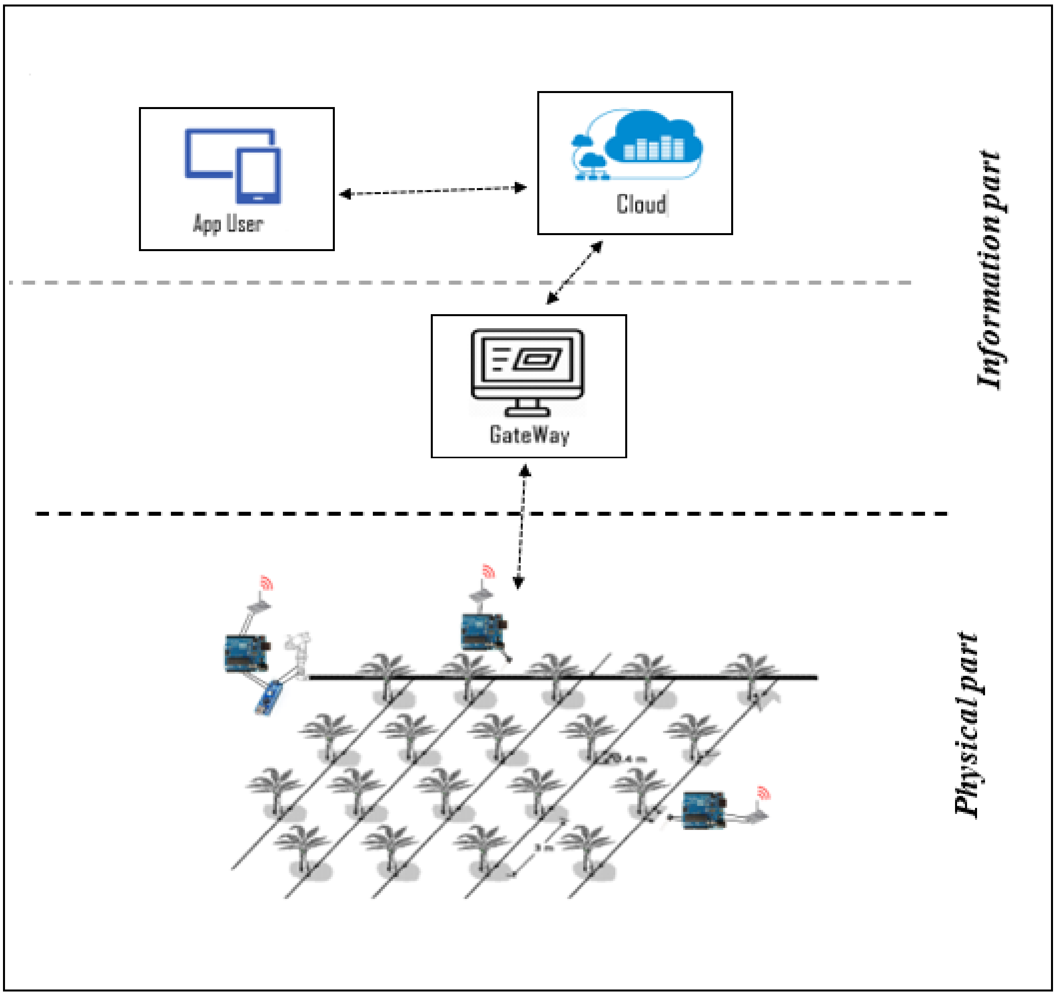

Figure 18: System element

5. General system structure

In this section, we detail the general structure of our system elements in Figure 18, Figure 19. This structure shows its large modules.

Below, we explain the role of each module and its relationship with others.

5.1. System modules

5.1.1. Real capture and command interface:

This interface consists of two units:

Information capture module:

Composed of:

- Microcontroller card
- Card ZigBee for send data
- A set of sensors.

This module makes it possible to collect data on the climate; it involves:

- Air temperature and humidity
- Soil moisture
- Wind speed

Actuator module:

Also composed of another microcontroller card controls the Solenoid valve to control the water flow for the regulation of irrigation after Received command by card ZigBee.

The actions to be under taken by this module proposing these components are:

- Turn on (component)
- Turn off (component).

It should be noted that these actions are elementary. The upper level manages complex actions.

5.1.2. Perception and Recognition module

The Perception and Recognition of Situations are the modules that recognize the different situations of parcels based on which decisions (of the system or the human operator) must be established. These situations are deduced from the raw data of soil moisture provided by the Capture module.

The various situations of parcels in the context of our project are:

- For the soil of parcel: dry or wet

- Determine any parcel that needs water most

The calculation process is triggered after the data required for a specific period has been collected.

After determining the parcel to be watered, the value of water required is calculated in the Decision module.

5.1.3. Decision module

As mentioned above, one of our goals is to create an intelligent system capable of making decisions independently. So, this module is designed to provide the two actions to be taken to perform the required service. Based on the weather information of the parcel, the required amount of water is determined using the crop evaporation equation (mentioned in the first chapter), and we use in its application the fuzzy logic system. This is due to the lack of some factors and the mathematical difficulty of calculation. Also, it computes the duration of this process.

5.1.4. Control module

The control module a website located in the cloud that has several functions. It controls the entire system by interacting with the communication interface. Also, this monitor climate change in the farm and allows manual remote system control for all parts of the farm. The collected data is then be presented in a statistical form and visualized in charts to make it more intuitive and easier for decision making. In addition, there is a local version for offline use.

5.1.5. Other modules

- **Decision Execution module:**

It is the module in which the system sends orders to start watering any part and stopping it to the Actuator module.

Also, on its level is to monitor the watering period and stop it at the end of that period.

- **Communication interface:**

This interface provides communication between the control module and the rest of the system via protocol HTTP using Rest API.

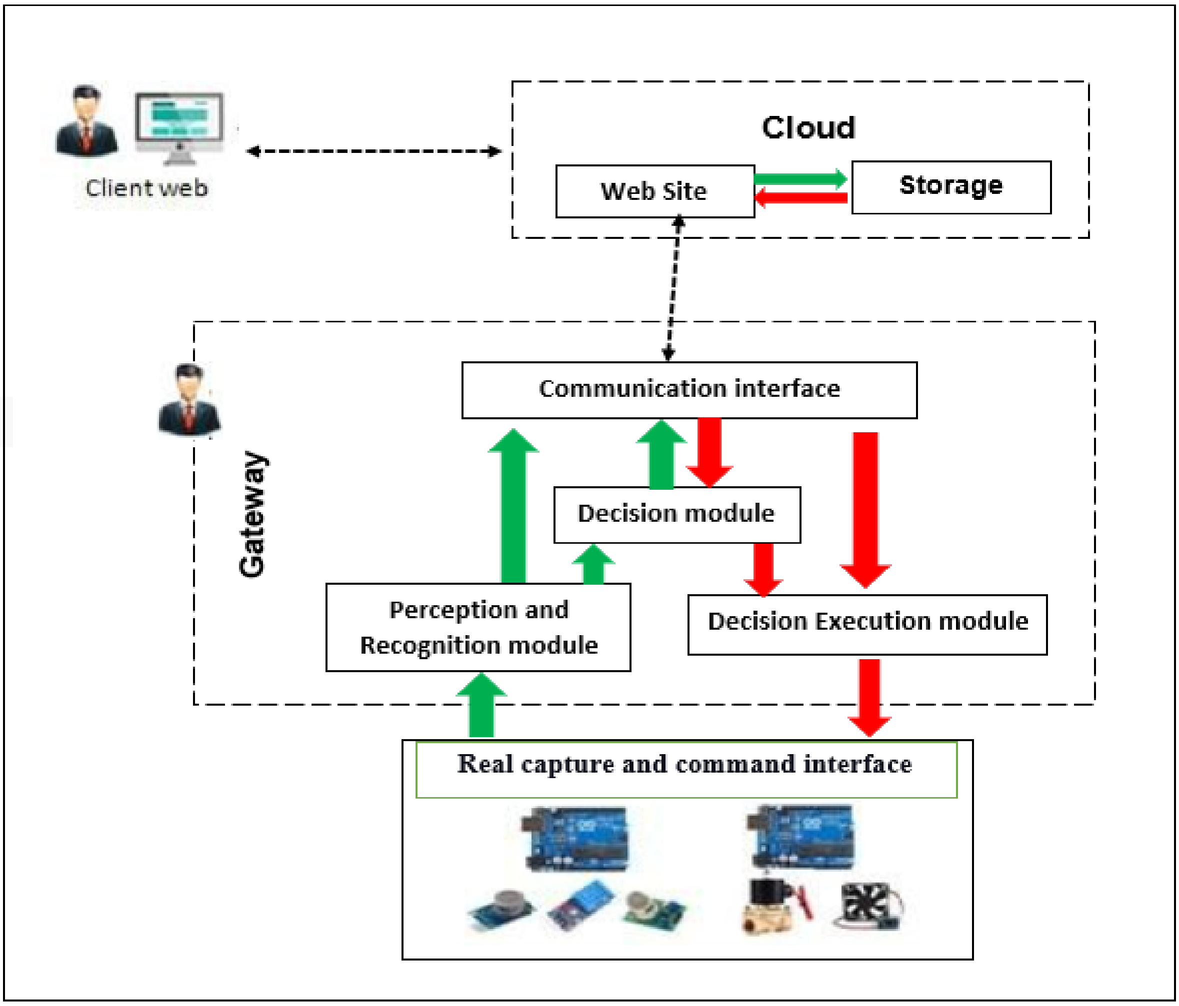

Figure 19: General system architecture

6. Principle of operation

- The system user is free to switch between automatic and manual control.
  In the case of automatic control, the operation of the system is governed by the data acquired by the sensors (via the *Capture* module).
- When there is a change in the situation that determines the parcel status, the decision module is triggered and determines the length of time required for watering to be taken each time.
- These actions concern the opening and closing of watering. The execution of operations must be carried out concurrently with the rest of the system.
- The site is designed to allow control of all the parcels. It also expands the local monitoring system with functions related to analysis and statistics.

7. Perception and Recognition module

After the system obtains soil moisture data for each part of the farm, in the case of the automatic system, it determines the part most in need of irrigation by comparing the soil moisture for each part with the minimum value needed by the plant (Figure 20).

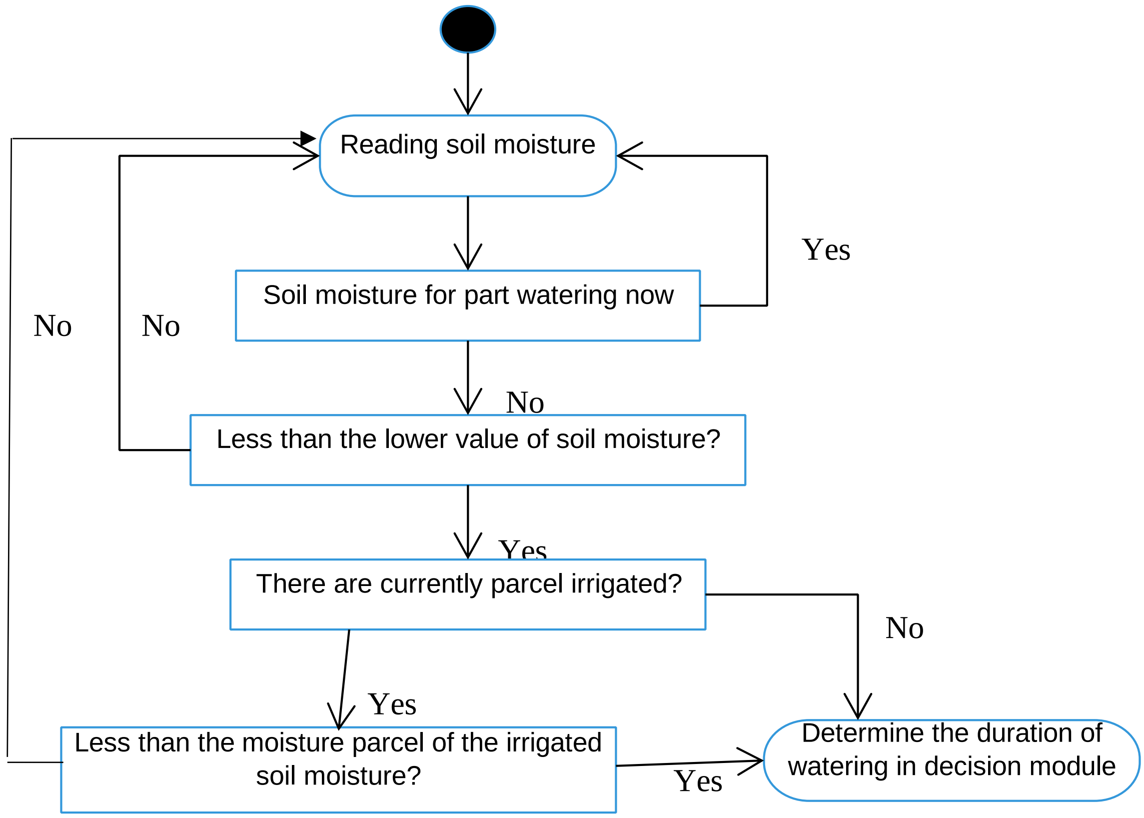

Figure 20: Activity diagram of Perception and Recognition module

When receiving soil moisture, if the system irrigates the parcel now, it re-read the information again and if it does not watering, then it makes sure that the moisture has reached the minimum allowed in the system or less if the system does not watering any other parcel now, goes to module of decision to determine the duration of irrigation

If the system is watering other parcels now, it comparing the parcel humidity with the obtained humidity, if you find it to be larger, it re-read the information again, and if it is less direct, go to the decision module.

8. Decision module

This is an essential part of the system.The amount of water required for irrigation is determined at its level using a fuzzy logic algorithm that begins from the climatic factors of the parcel to extract ET_o (chapter 1) and multiplied by kc Plant Factor to calculate the amount of water that requires ETc using equation 3(chapter 1).

To calculate the length of time required for irrigation, it is sufficient only to divide the volume of water by the flow capacity through the valves

8.1. Fuzzy logic algorithm

We use Fuzzy Logic to calculate ET_o due to the difficulty of calculating it, therefore due to a lack of climatic factors that are difficult to identify and some complex operations, so it is used to facilitate this process.

8.1.1. Fuzzification

The input variable, air humidity, wind speed, and air temperature, is converted to a language variable, The output of this controller is the reference evapotranspiration (ET0), the membership functions in the following triangular shapes (Figure 21).

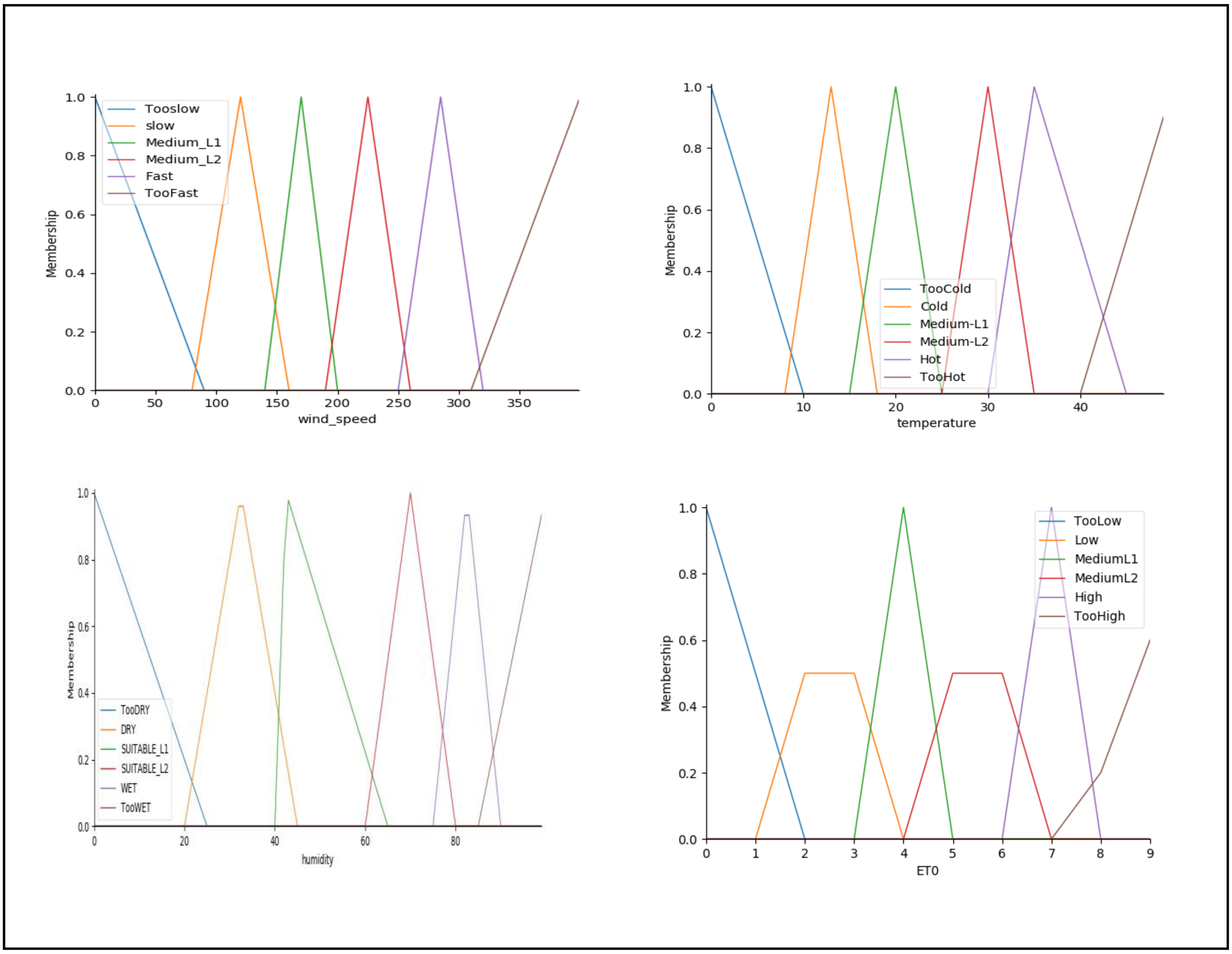

Figure 21: The membership functions

8.1.2. Inference mechanism

After studying data extracted manually from programs (CLIMWAT2, CRW8) that provide climate data for the past 30 years, available from the International Organization for Agricultural Nutrition, we obtained the fuzzy logic rules shown in the following tables:

humidity		Too Dry	Dry	Suitable-L1	Suitable-L2	WET	Too WET
Wind speed	Too Slow	Too Low	Low	Too Low	Too Low	Too Low	Too Low
	Slow	Low	Low	Low	Too Low	Too Low	Too Low
	Medium-L1	Low	Low	Low	Too Low	Too Low	Too Low
	Medium-L2	Low	Low	Low	Too Low	Too Low	Too Low
	Fast	Low	Low	Low	Too Low	Too Low	Too Low
	Too Fast	Low	Low	Low	Too Low	Too Low	Too Low

Table 1: Temperature Too Cold

humidity		Too Dry	Dry	Suitable-L1	Suitable-L2	WET	Too WET
Wind speed	Too Slow	Too Low	Low	Low	Low	Low	Low
	Slow	Low	Low	Low	Low	Low	Too Low
	Medium-L1	Medium-L1	Medium-L1	Low	Low	Too Low	Too Low
	Medium-L2	Medium-L2	Medium-L1	Low	Low	Too Low	Too Low
	Fast	Medium-L2	Medium-L1	Medium-L1	Low	Low	Too Low
	Too Fast	Medium-L2	Medium-L2	Medium-L1	Low	Low	Too Low

Table 2: Temperature Cold

humidity		Too Dry	Dry	Suitable-L1	Suitable-L2	WET	Too WET
Wind speed	Too Slow	Medium-L1	Medium-L1	Low	Low	Low	Low
	Slow	Medium-L1	Medium-L1	Medium-L1	Medium-L1	Low	Low
	Medium-L1	Medium-L2	Medium-L2	Medium-L1	Medium-L1	Low	Low
	Medium-L2	Medium-L2	Medium-L2	Medium-L1	Medium-L1	Low	Low
	Fast	High	Medium-L2	Medium-L2	Medium-L1	Low	Low
	Too Fast	High	High	Medium-L2	Medium-L1	Medium-L1	Low

Table 3: Temperature Medium-L1

humidity		Too Dry	Dry	Suitable-L1	Suitable-L2	WET	Too WET
Wind speed	Too Slow	Medium-L2	Medium-L1	Medium-L1	Medium-L1	Medium-L1	Medium-L1
	Slow	Medium-L2	Medium-L2	Medium-L2	Medium-L1	Medium-L1	Medium-L1
	Medium-L1	High	High	Medium-L2	Medium-L1	Medium-L1	Medium-L1
	Medium-L2	Too High	High	Medium-L2	Medium-L2	Medium-L1	Medium-L1
	Fast	Too High	Too High	High	Medium-L2	Medium-L1	Medium-L1
	Too Fast	Too High	Too High	High	Medium-L2	Medium-L1	Medium-L1

Table 4: Temperature Medium-L2

humidity		Too Dry	Dry	Suitable-L1	Suitable-L2	WET	Too WET
Wind speed	Too Slow	Medium-L2	Medium-L2	Medium-L2	Medium-L1	Medium-L1	Low
	Slow	High	High	Medium-L2	Medium-L2	Medium-L1	Low
	Medium-L1	High	High	Medium-L2	Medium-L2	Medium-L2	Medium-L1
	Medium-L2	Too High	High	High	Medium-L2	Medium-L2	Medium-L2
	Fast	Too High	Too High	Too High	High	Medium-L2	Medium-L2
	Too Fast	Too High	Too High	Too High	High	Medium-L2	Medium-L2

Table 5: Temperature Hot

humidity		Too Dry	Dry	Suitable-L1	Suitable-L2	WET	Too WET
Wind speed	Too Slow	Low	Low	Low	Low	Low	Low
	Slow	Medium-L1	Medium-L1	Medium-L1	Medium-L1	Low	Low
	Medium-L1	Medium-L2	Medium-L2	Medium-L1	Medium-L1	Medium-L1	Low
	Medium-L2	High	Medium-L2	Medium-L2	Medium-L1	Medium-L1	Low
	Fast	High	High	Medium-L2	Medium-L1	Medium-L1	Low
	Too Fast	Too High	Too High	High	Medium-L2	Medium-L1	Low

Table 6: Temperature Too Hot

8.1.3. Defuzzification

Because too much water for plants irrigation harms them, and too little water harms them and reduces their production efficiency. So, we were guided in the defuzzification step of using the "centroid" center of gravity. To have an average value that is neither high nor low.

9. Control module

The administrator can control the watering of all parts of the farm and can obtain Real-time climate and soil moisture data for each part. The design is as follows:

9.1. Architecture

- The communication module with HTTP protocol and using RestAPI technology to protect information and secure limited cloud access only when needed.
- Data is sent to the server, then stored.
- Displays statistics on the website so the user can view the results.
- Allows disabling and enabling automatic system operation.
- It allows controlling the parcel's watering system.

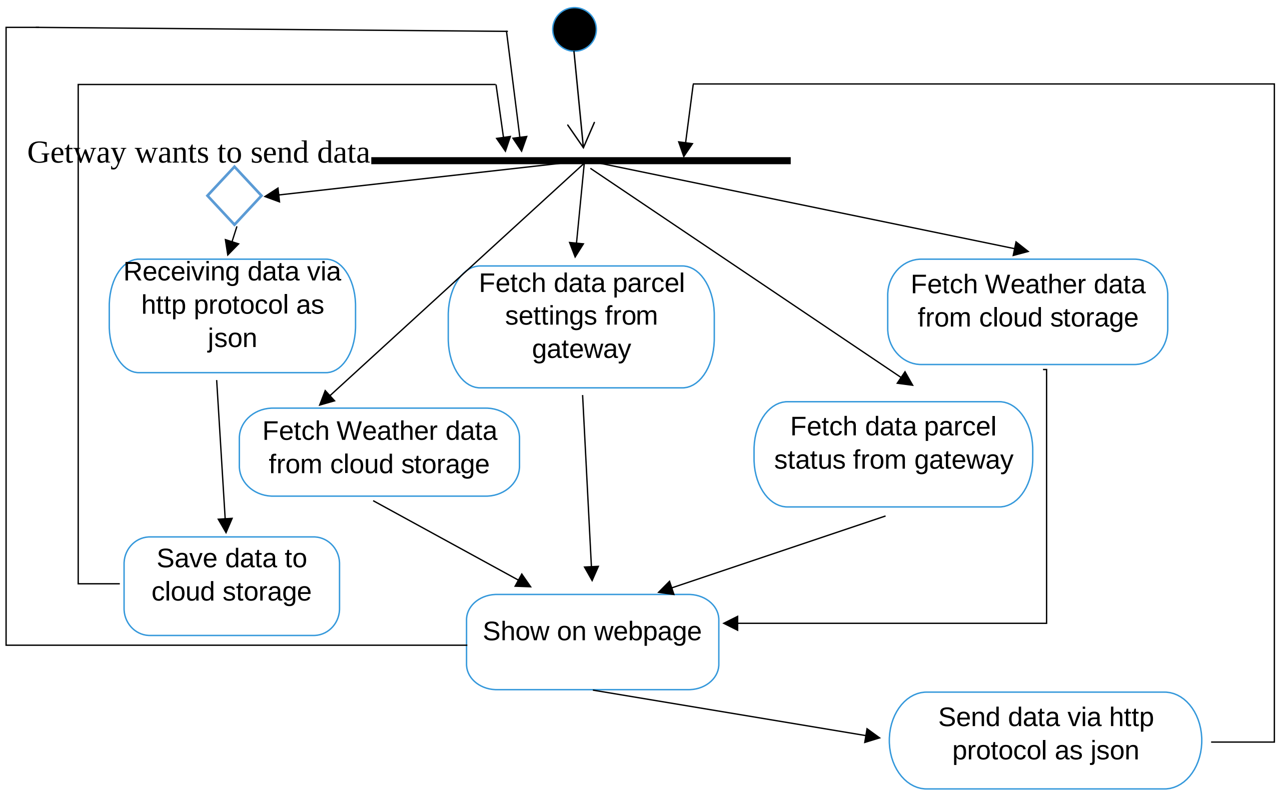

Figure 22: Activity diagram website management in Cloud

9.2. Component interaction:

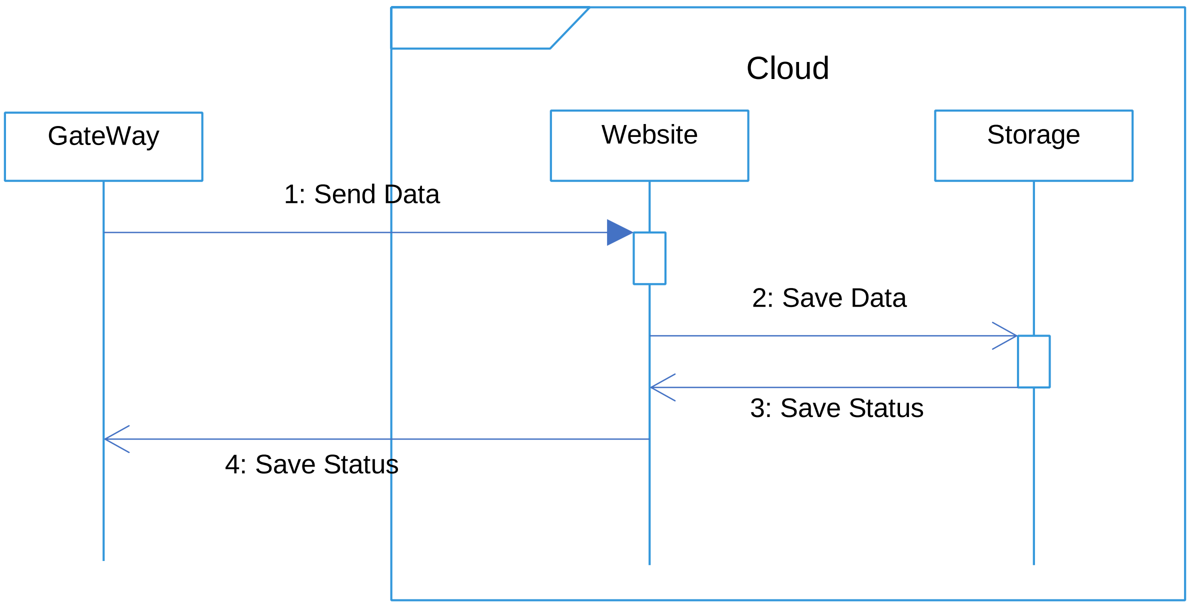

Figure 23: Sequence Diagram Storage data in Cloud

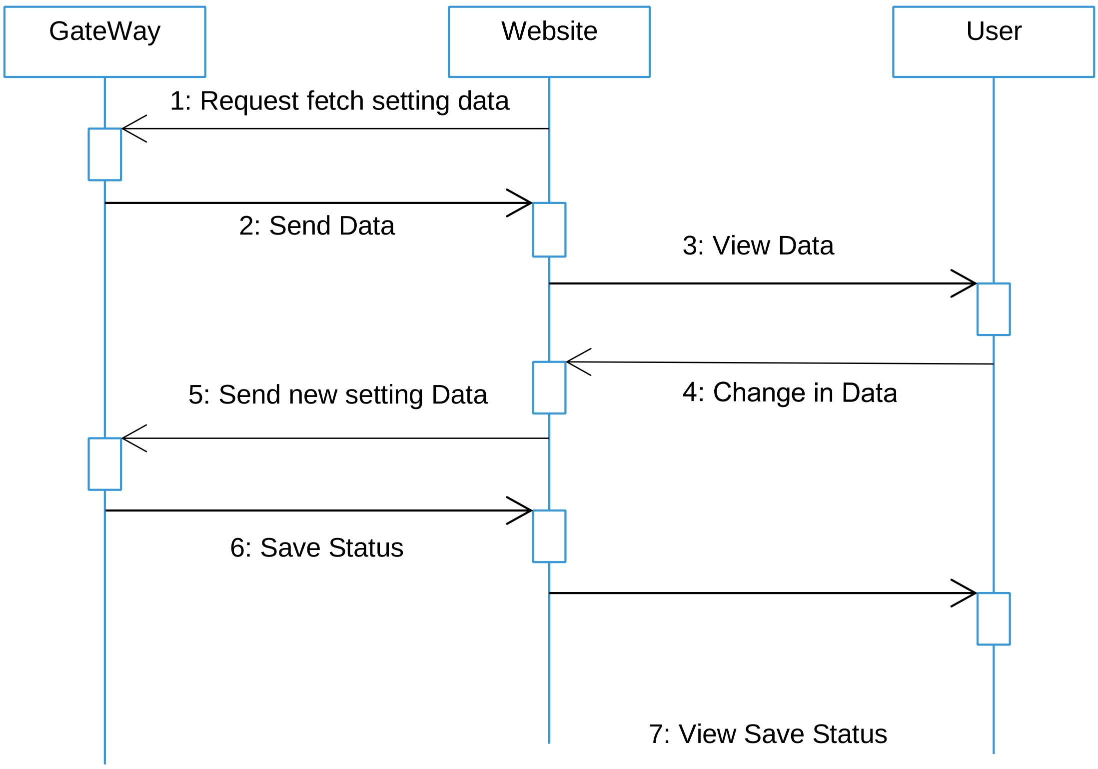

Figure 24: Sequence Diagram Change system configuration

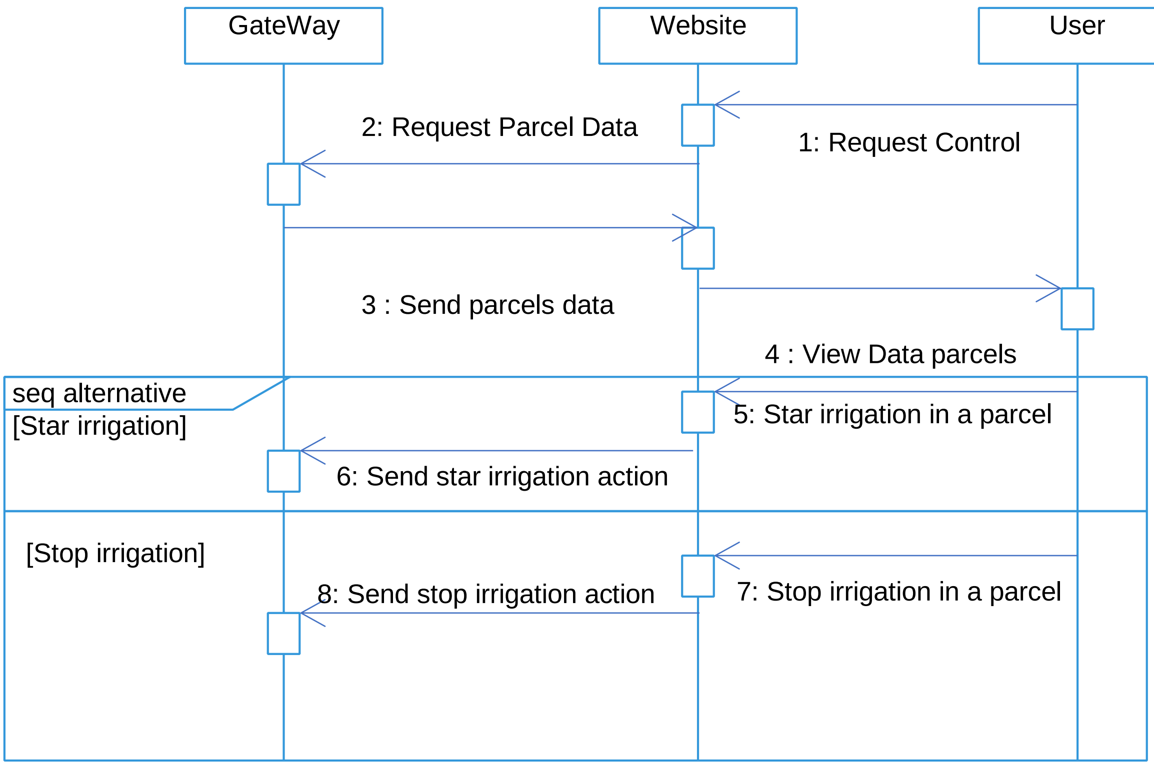

Figure 25: Sequence Diagram Control Irrigation

10. Irrigation management

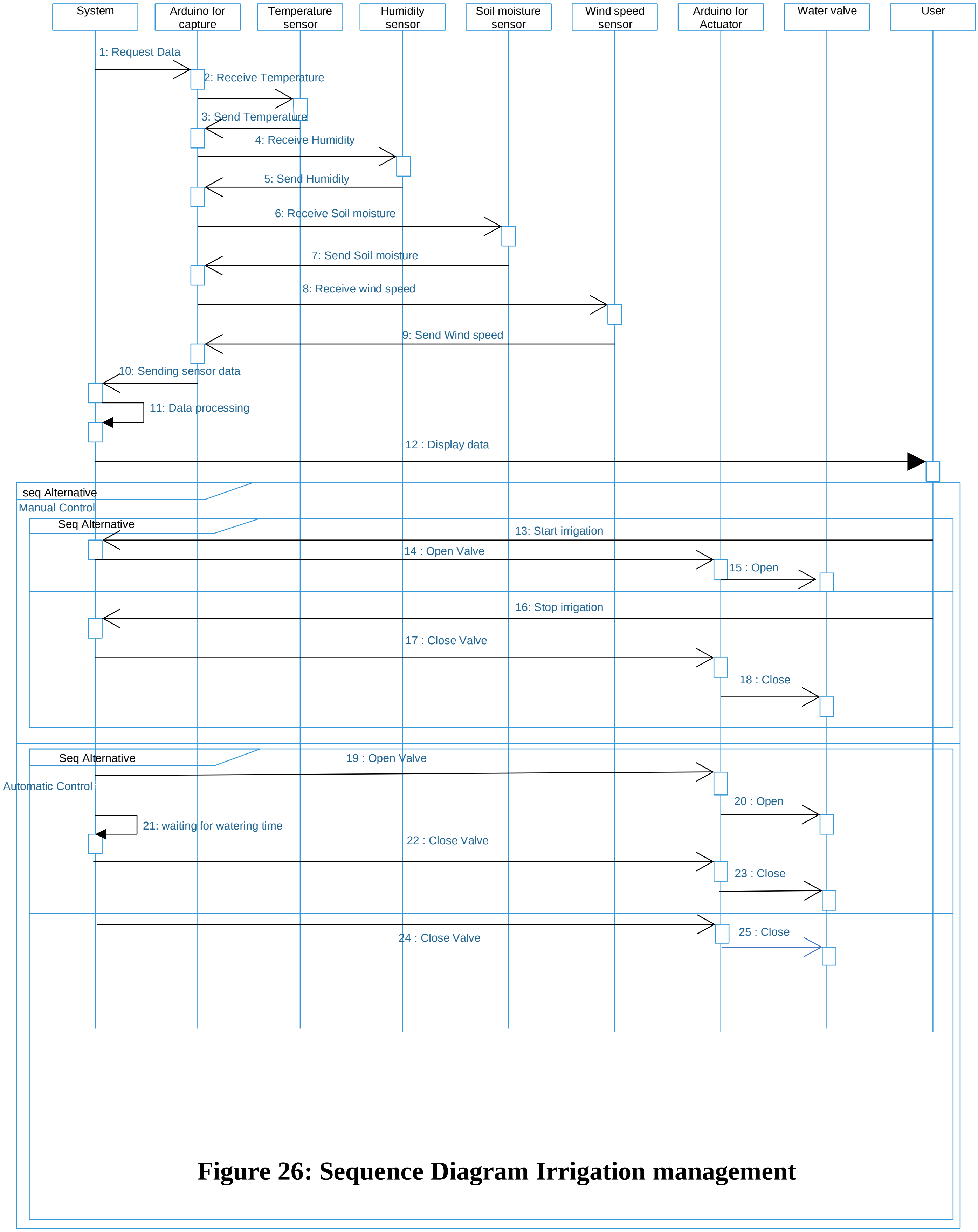

Figure 26: Sequence Diagram Irrigation management

- Sensors (temperature sensor - air humidity sensor - wind speed sensor - soil moisture sensor) record the values of temperature, wind speed, atmospheric humidity, and soil moisture, then they are sent to the Arduino board.

The data received from the sensors are first stored in the database and later when the system indicates that any segment of the farm needs water the relevant data are retrieved to calculate the mean temperature, mean air humidity and wind speed which are used as inputs in the fuzzy logic system to obtain the volume of water needed for irrigation. After determining the period of irrigation, in automatic mode, the decision unit sends requests to open or close the valve for irrigation (Figure 26).

11. Conclusion

In this chapter, we have presented all the details of the design of our system. First We introduced the principle of our solution and the overall architecture of the system. then We explained the details of each part, and We have provided everything related to fuzzy logic.

In the next chapter, we will present the details of the implementation.

Chapter 4 Realization and Implementation

1. Introduction

In this chapter, we present the hardware tools and software environments used to develop our system in addition to the different implementation platforms of its various parts and screenshots of its most essential elements and show the results obtained.

2. Physical interface for capture and actuation

It consists of an Arduino device connected to a group of sensors that receive data (Figure 27), Arduino processes the data and sends it via the ZigBee to the Gateway.

2.1 Physical diagram

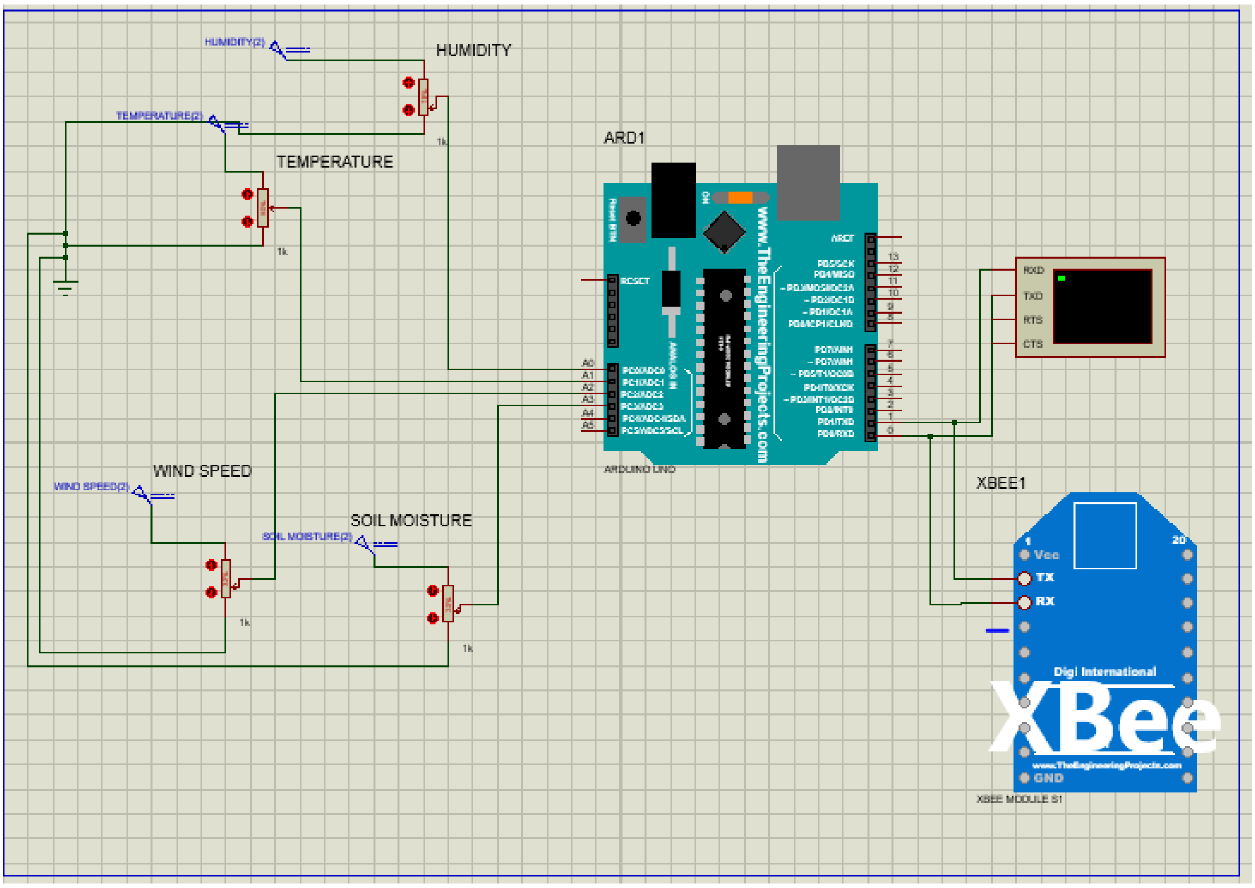

Figure 27: Physical diagram I

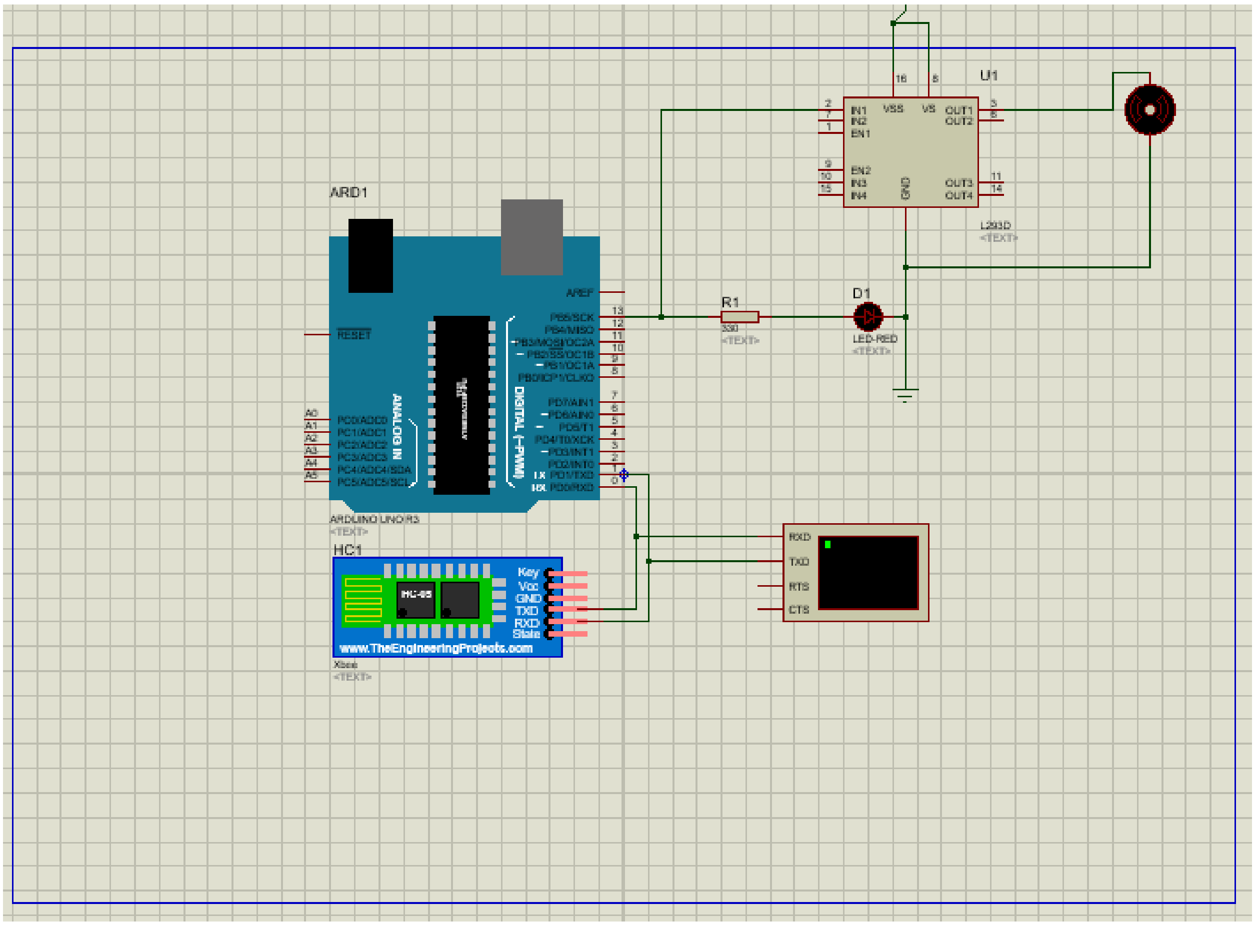

Figure 28: Physical diagram II

2.2 Components

2.2.1 Arduino UNO card

The Arduino card (Figure 29) constitutes the central part of our project. We used two cards:

- The first card is connected to the sensors, it has the function of recovering the data from the sensors then coding them, then sending them to gateway.
- The second card is connected to the actuators. Its function is to receive orders from gateway and then execute them.

Figure 29: Arduino UNO card.

2.2.2 Sensors

Sensors	Description
Temperature sensor	This module used as a temperature acquisition node for our project.
Humidity sensor	The humidity sensor, as its name suggests, allows measuring the ambient humidity.
Wind speed sensor	An anemometer is a tool that measures wind speed and pressure. Coupled with a weather vane, it can also give details of its direction. It is mainly used in meteorology, but the information it provides can also be used in smart farming.
Soil moisture sensor	It is a sensor to calculate the amount of water contained in the soil.

Table 7: Description of the sensors

2.2.3 Actuators

- **Valve**

It can be described as compact and practical, easily installed, and maintained—mechanical consumption without electric consumption, infinitely adjustable.

The pressure control valve (constant flow valve) functions as a water flow switch.

- **Pump**

The role of the water pump is to fill the tank, and we chose an electric pump for remote control.

3. Software environment and language

3.1 Proteus 8 Professional

The Proteus Design Suite is widely used across various industry sectors as a cost-effective solution for professional PCB design and as a rapid prototyping tool for R&D. Proteus Design Suite is found in High Schools, Colleges, and Universities across the world, teaching electronics, embedded design, and PCB layout to tens of thousands of students each year [11].

3.2 Language C for Arduino

The language used for Arduino programming is based on C / C ++ languages. An Arduino program consists mainly of a header calling on the program dependencies, a 'setup' function which initializes the variables involved, as well as a 'loop' loop which permanently executes the instructions of our program.

3.3 Arduino IDE

The creators of Arduino have developed software so that the programming of Arduino boards is visual, simple and complete at the same time.

This is what is called an IDE, which stands for Integrated Development Environment or "Integrated" Development Environment.

3.4 Virtual Serial Ports Emulator

Virtual Serial Port Emulator is a software application that replicates physical COM ports. The virtual serial ports that are created are fully compatible with operating systems and applications and are treated in the same way as a real port. [15]

4. Python application: Local control system

4.1 Material environment

To develop the python application, we used a laptop with the following characteristics:

- Intel (R) Core (TM) i5-2450M CPU @ 2.70GHz RAM 8GO.

- 64-bit operating system.

5. Languages and programming environment and libraries

5.1.1 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components. [16].

5.1.2 Pycharm

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains.

It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as Data Science with Anaconda. [17].

5.1.3 scikit-fuzzy

Fuzzy Logic Toolbox for Python. This package implements many useful tools and functions for computation and projects involving fuzzy logic, also known as grey logic.

5.1.4 NumPy

NumPy is a library for the Python programming language that allows for more data storage with less memory. With a multidimensional array and other resources, NumPy allows Python programmers to store numbers efficiently.

5.2 Graphic interface

In this section, we present a python Command Line Application (Gateway)

First and foremost, you must connect COM ports by Virtual Serial Ports Emulator between the Gateway app and the farm parts that are simulated with Proteus 8 Professional.

And immediately after the connection process, the reception of the captured data is controlled by the Sensors, and also the devices found in the farm are controlled.

Figure 30, presents the sequence of actions taken by the system as well as notes and tips.

```
C:\Users\ELHABIB\Anaconda2\python.exe E:/files/SystemIOT/Start_system.py
<= ..........2020-08-16 18:20:00 ..........  =>  welcome to GateWay Application
<= ..........2020-08-16 18:20:00 ..........  =>  Start Lead System settings
<= ..........2020-08-16 18:20:00 ..........  =>  IOT Communication settings
<= ..........2020-08-16 18:20:00 ..........  =>  parcel 1:  r : COM2 w : COM3
<= ..........2020-08-16 18:20:00 ..........  =>  End Lead System settings
<= ..........2020-08-16 18:20:00 ..........  =>  initialize IOT System start
<= ..........2020-08-16 18:20:00 ..........  =>  initialize Parcel System start
<= ..........2020-08-16 18:20:00 ..........  =>  initialize Fuzzy System start
<= ..........2020-08-16 18:20:00 ..........  =>  start add rules to system
<= ..........2020-08-16 18:20:48 ..........  =>  end add rules to system
<= ..........2020-08-16 18:20:48 ..........  =>  initialize Fuzzy System end
<= ..........2020-08-16 18:20:48 ..........  =>  initialize Time System start
<= ..........2020-08-16 18:20:48 ..........  =>  initialize Time System end
<= ..........2020-08-16 18:20:48 ..........  =>  initialize Parcel System End
<= ..........2020-08-16 18:20:48 ..........  =>  initialize IOT System end
<= ..........2020-08-16 18:20:48 ..........  =>  reading data from 1 ses........
serving at port 8040
```

Figure 30: staring gateway

Figure 31 displays the farm's meteorological data receiving via Gateway methods and sending it to the cloud.

```
<= ..........2020-08-16 18:52:38 ..........  =>  initialize Fuzzy System end
<= ..........2020-08-16 18:52:38 ..........  =>  initialize Time System start
<= ..........2020-08-16 18:52:38 ..........  =>  initialize Time System end
<= ..........2020-08-16 18:52:38 ..........  =>  initialize Parcel System End
<= ..........2020-08-16 18:52:38 ..........  =>  initialize IOT System end
<= ..........2020-08-16 18:52:38 ..........  =>  reading data from 1 ses........
serving at port 8040
INSERT INTO Parameter (Parcel,Temperature,Humidity,WindSpeed ,SoilMoisture,date)VALUES (1,51.2,18.4,32.8,33.8,date('now'))
conn
<sqlite3.Cursor object at 0x000000000838FF10>
<= ..........2020-08-16 18:52:49 ..........  =>  Reading data from parcel 1 :
<= ..........2020-08-16 18:52:49 ..........  =>  {1: 51.2, 2: 18.4, 3: 32.8, 4: 33.8}
<= ..........2020-08-16 18:52:49 ..........  =>  reading data from 1 ses........
INSERT INTO Parameter (Parcel,Temperature,Humidity,WindSpeed ,SoilMoisture,date)VALUES (1,51.2,18.4,32.8,33.8,date('now'))
conn
<sqlite3.Cursor object at 0x000000000838FF10>
<= ..........2020-08-16 18:52:59 ..........  =>  Reading data from parcel 1 :
<= ..........2020-08-16 18:52:59 ..........  =>  {1: 51.2, 2: 18.4, 3: 32.8, 4: 33.8}
<= ..........2020-08-16 18:52:59 ..........  =>  reading data from 1 ses........
```

Figure 31: Reading Data

Figure 32 shows sending orders to the farm in order to open and close the water valve for one of the parts.

```
<= .........2020-08-16 18:54:01 .........  =>  Reading data from parcel 1 :
<= .........2020-08-16 18:54:01 .........  =>  {1: 51.2, 2: 18.4, 3: 32.8, 4: 33.8}
<= .........2020-08-16 18:54:01 .........  =>  reading data from 1 ses........
INSERT INTO Parameter (Parcel,Temperature,Humidity,WindSpeed ,SoilMoisture,date)VALUES (1,51.2,18.4,32.8,0.0,date('now'))
conn
<sqlite3.Cursor object at 0x000000000838FF10>
<= .........2020-08-16 18:54:15 .........  =>  Reading data from parcel 1 :
<= .........2020-08-16 18:54:15 .........  =>  {1: 51.2, 2: 18.4, 3: 32.8, 4: 0.0}
<= .........2020-08-16 18:54:15 .........  =>  sql query :SELECT avg(Temperature),avg(Humidity),avg(WindSpeed) FROM Parameter
trm 51.2 hum 18.4 wind 32.8
<= .........2020-08-16 18:54:18 .........  =>  ET0 = 9.0
<= .........2020-08-16 18:54:18 .........  =>  Time Irrigation is   = 9.0 m
<= .........2020-08-16 18:54:18 .........  =>  start watering the parcel 1 (on)
COM3 on
<= .........2020-08-16 18:54:18 .........  =>  start parcel1
<= .........2020-08-16 18:54:18 .........  =>  reading data from 1 ses........
<= .........2020-08-16 18:54:19 .........  =>   def 0 s
 <= .........2020-08-16 18:54:19 .........  =>   def 0 s <= .........2020-08-16 18:54:19 .........  =>   def 0 s <= .......
```

Figure 32: Opening valve

6. Website and Cloud

In this section, we present the tools and software environments used to develop our website.

6.1 Programming languages and environments and libraries

6.1.1 Python

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

6.1.2 SQL

SQL (Structured Query Language) is a computer language used to exploit databases. It generally allows the definition, manipulation, and control of data security. In practice, the SQL language is used to create tables, add records in the form of rows, query a database, update it, or even manage the rights of users of this database. The vast majority of systems well supports it

6.1.3 Django

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.[18]

6.2 Cloud

6.2.1 PythonAnywhere

PythonAnywhere is an online integrated development environment (IDE), and web hosting service (Platform as a service) based on the Python programming language.]Founded by Giles Thomas and Robert Smithson in 2012, it provides in-browser access to server-based Python and Bash command-line interfaces, along with a code editor with syntax highlighting. Program files can be transferred to and from the service using the user's browser. Web applications hosted by the service can be written using any WSGI-based application framework.

6.3 Website

There are two copies of the site, a local copy and a copy in the cloud, but they only have the same work, one for local control and the other for remote control over the network, and the site has been uploaded to a cloud pythonAnyWhere

The login page (Figure 33) is used to protect the site from being accessed and only controlled by the administrator.

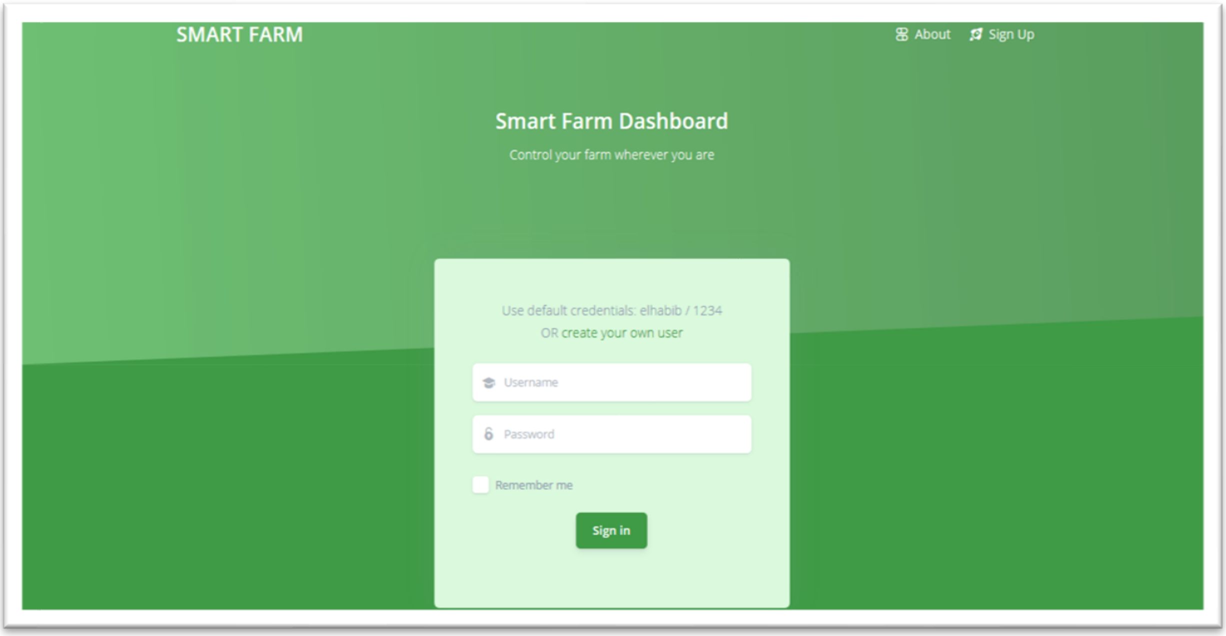

Figure 33: Login Page

Figure 34 presents a dashboard page in which the status of parts of the farm is reported in real-time and controlled.

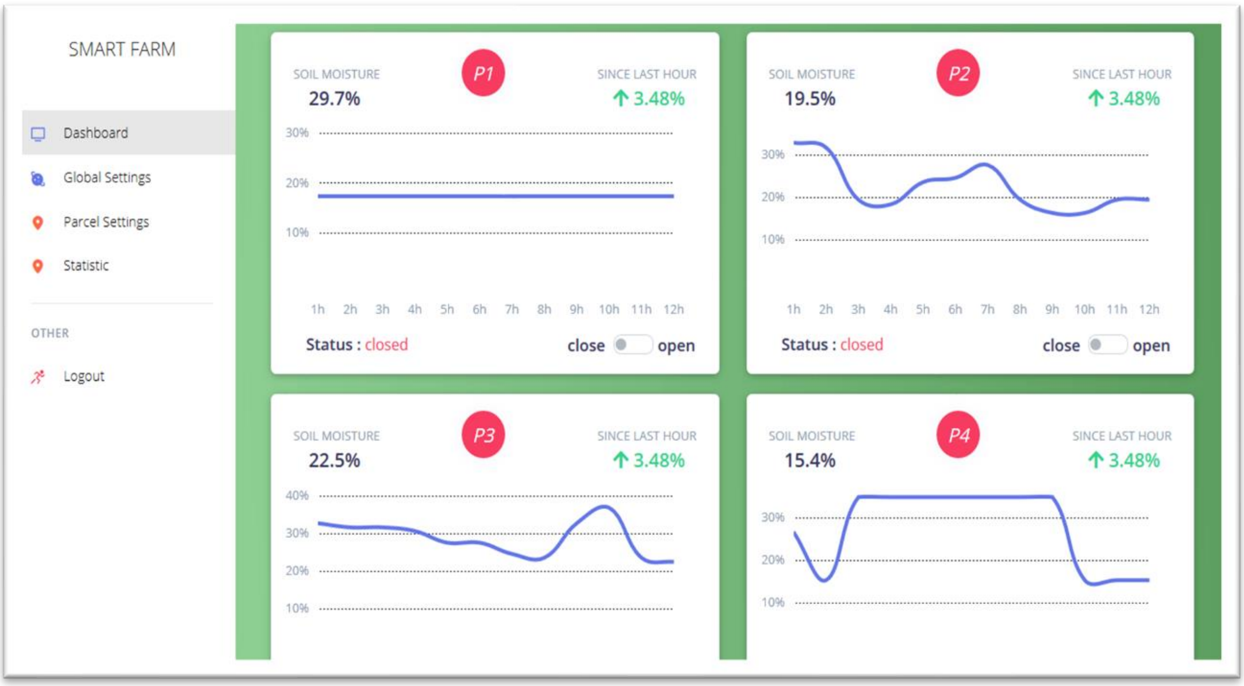

Figure 34: Dashboard Page

Figure 35 shows the manual control of the system, where the administrator can start or stop watering any part of the farm according to the data he has shown.

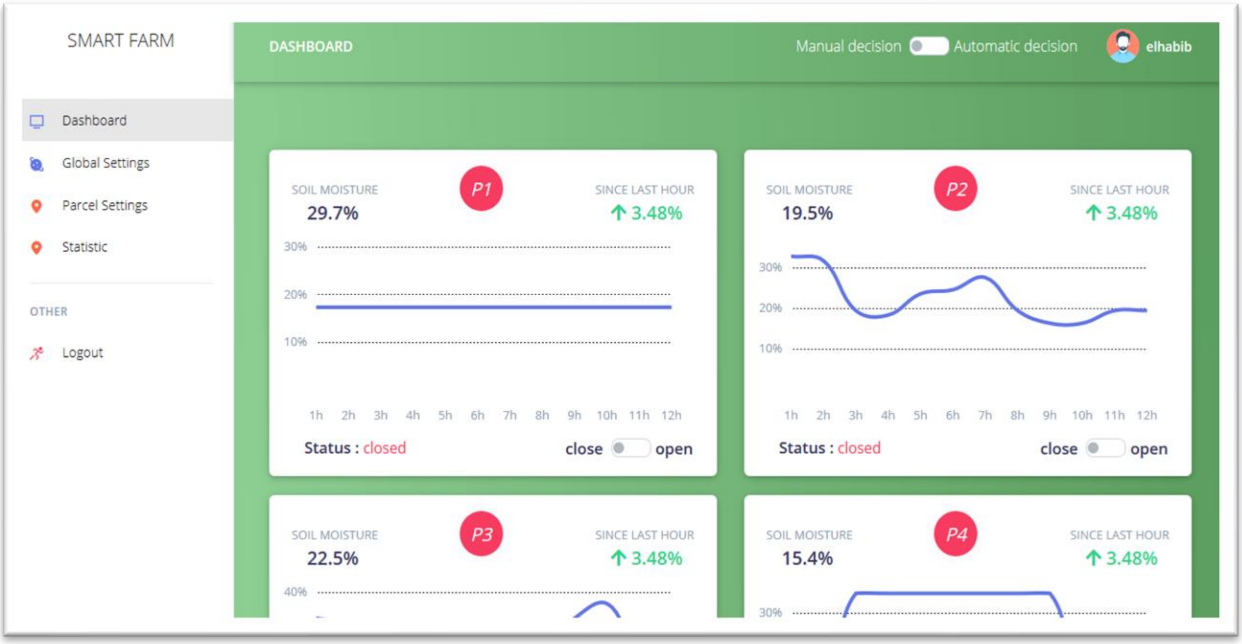

Figure 35: Manual Control

Figure 36 shows the automatic control of the system, where the system is responsible for starting or stopping irrigation for any part of the farm according to the weather data and according to the fuzzy logic algorithm.

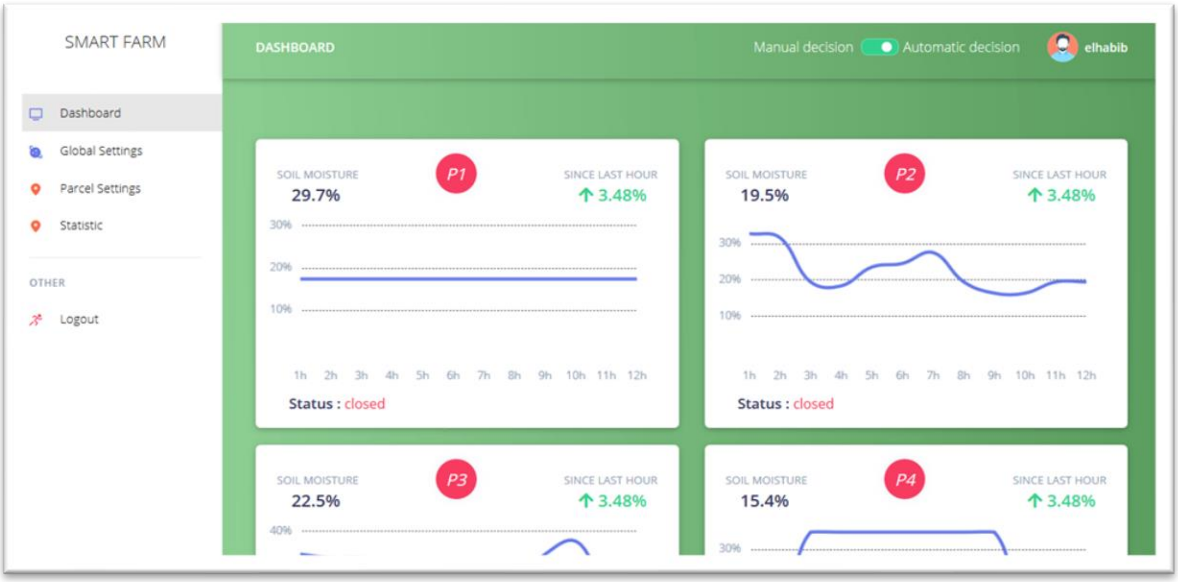

Figure 36: Automatic Control

Figure 37 presents the general system settings, including both the number of segments in the farm, the time to wait for automatic actions, the type of plant, and the minimum soil moisture for the plant.

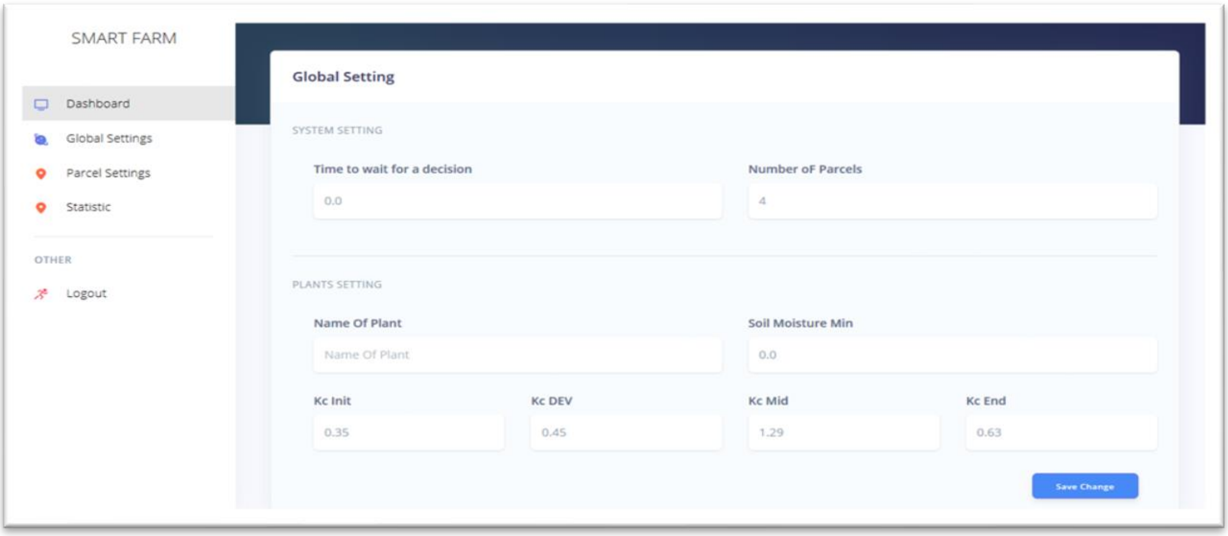

Figure 37: Global Setting

Figure 38 shows settings for each part of the farm (area of parts of the farm and water flow capacity for irrigation).

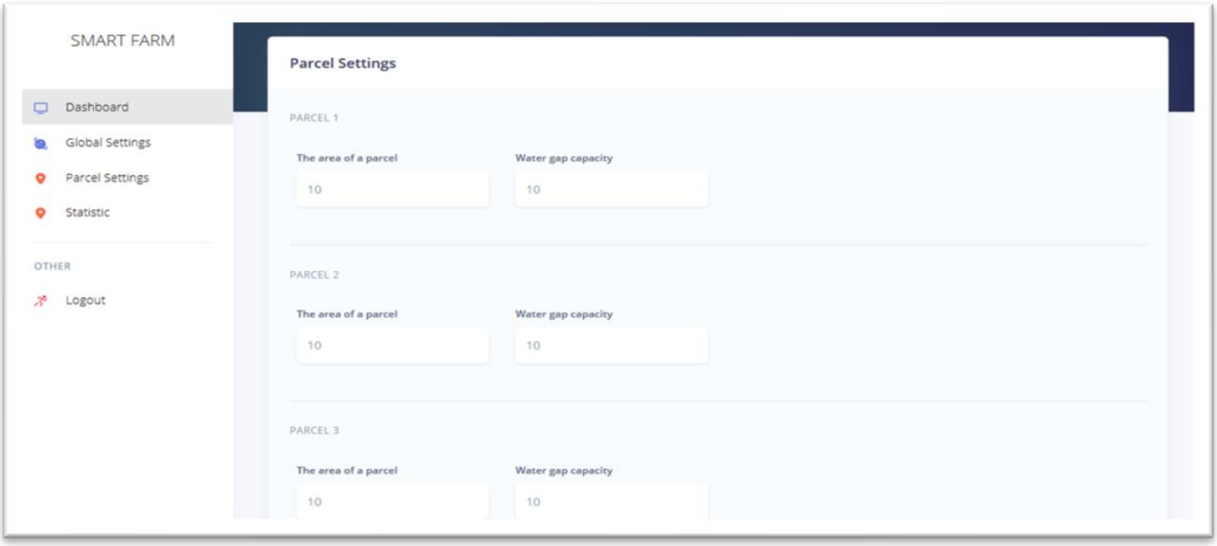

Figure 38: Parcels Setting

figure 39 presents demonstrate the general statistics of climatic factors for the last 12 hours, and the responsible person can decide to water the farm.

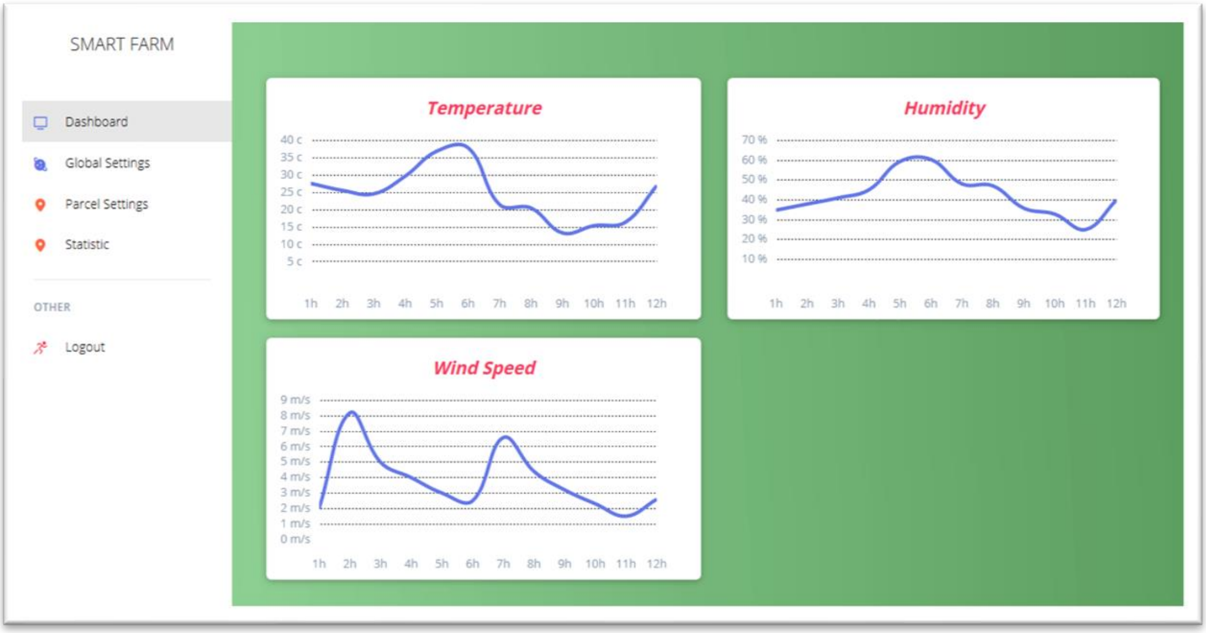

Figure 39: General statistics of climatic factors

7. Results and discussion

To test the effectiveness of the system and the accuracy of its results, we took the climate factors data for the farm and entered it into the system to obtain the following results:

Climate data for the farm is from a ***myirrigation.eu*** that provides any climate data for any specific area. We took climate data for seven consecutive days, and the data are shown in the following table.

Days	Temperature	Humidity	Wind Speed
Day 1	26.7	32	2.5
Day 2	27.05	27	3.1
Day 3	26.1	34	2.5
Day 4	25.4	40	6
Day 5	24.55	40	6.3
Day 6	24.9	40	6.3
Day 7	25.8	37	4.4

Table 8: climate data for seven consecutive days

The data came according to the following Line Charts (Figure 40,41,42):

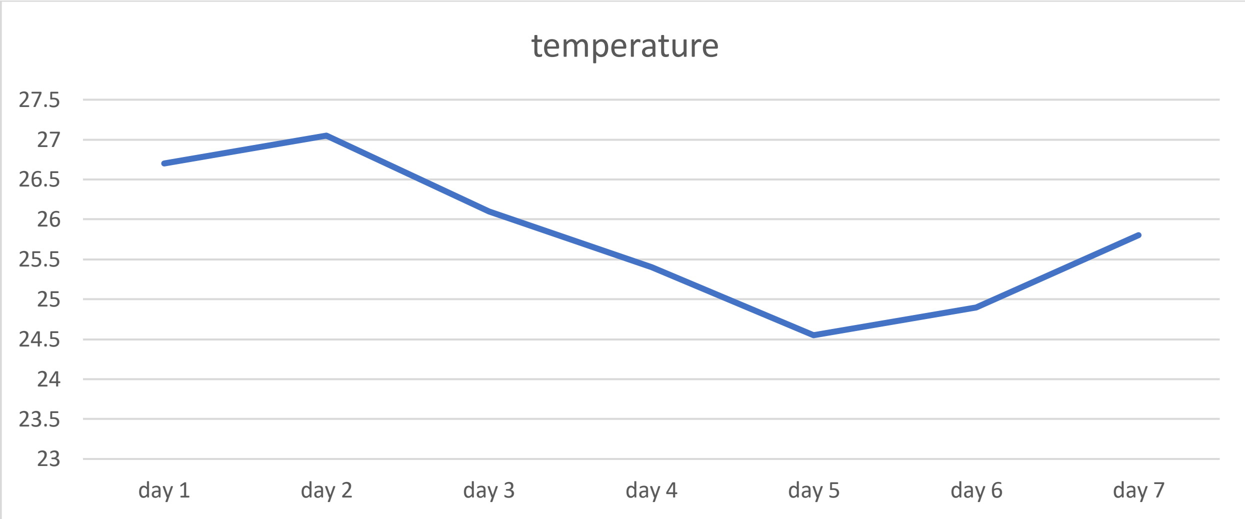

Figure 40: Temperature Line Chart

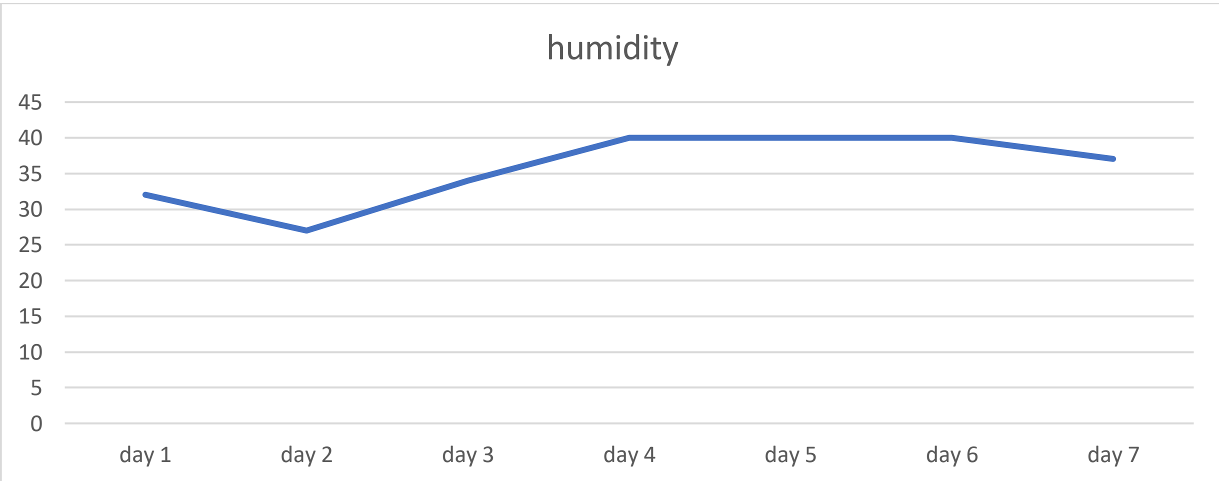

Figure 41: Humidity Line Chart

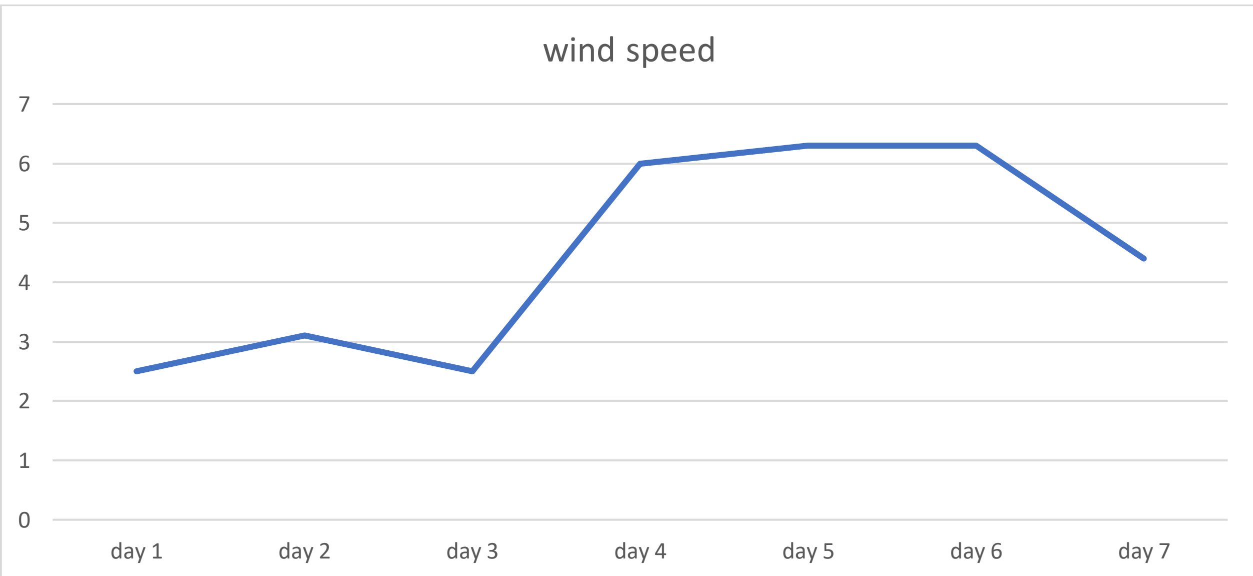

Figure 42: Wind Speed Line Chart

After entering the data into the fuzzy system algorithm, we get the following results:

Days	ET0
Day 1	7.659
Day 2	9
Day 3	7.7
Day 4	9
Day 5	7.91
Day 6	7.979
Day 7	9

Table 9: ET0 results

The data came according to the following Line Charts (Figure 43):

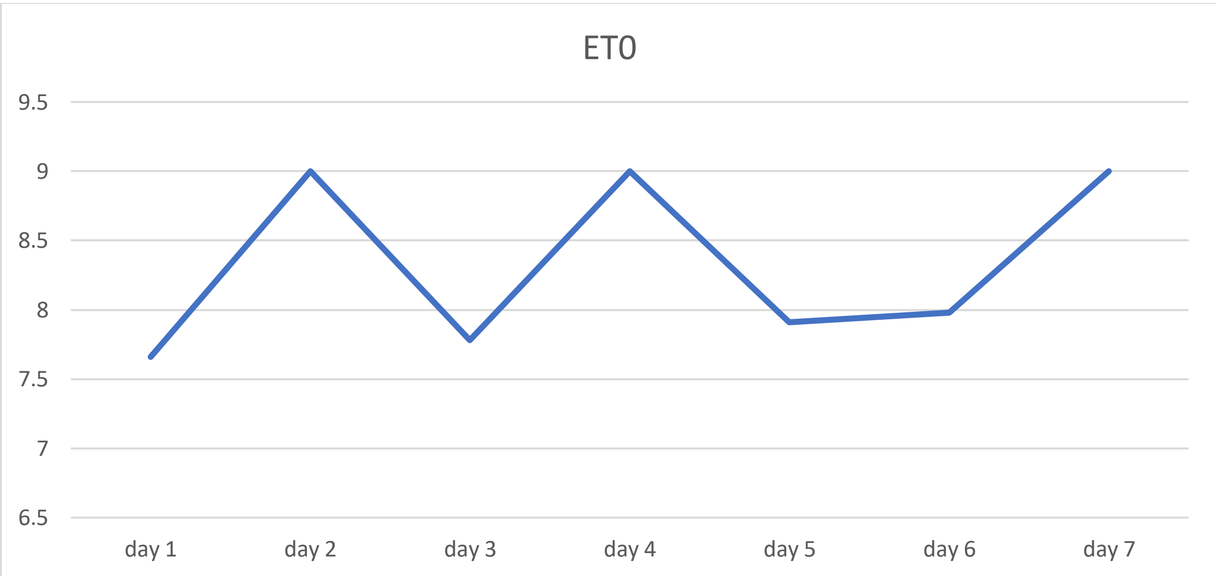

Figure 43: ET0 Line Chart

When using the data obtained in a field of 10,000 square meters and 10 liters of water flow rate, we get the following irrigation period in a minute:

Days	irrigation period in a minute
Day 1	164.69
Day 2	193.5
Day 3	167.26
Day 4	193.5
Day 5	170.06
Day 6	171.57
Day 7	193.5

Table 10: irrigation period in a minute

The data came according to the following Line Charts (Figure 44):

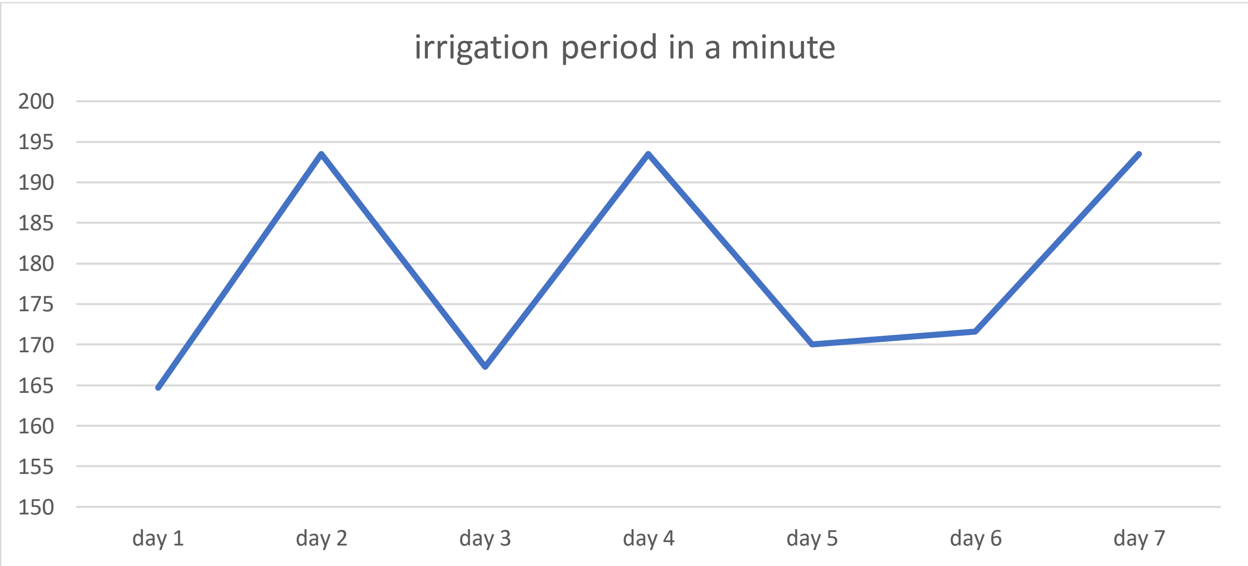

Figure 44: irrigation period in a minute

8. Conclusion

In this chapter, we provide details of implementing our system. We started by giving a comprehensive view of the system so that the reader can create the first picture of our system. Then we mentioned tools, environment, and programming languages. The physical part of the perception and operation of our system has also been presented in detail and with its connection diagram, including all components, as well as the obtained results.

General Conclusion

The project targets smart agriculture in general and watering systems in particular. Our overall goal was to develop a strategy for land controlling the irrigation with the ability to supervise and visualize the environmental situations of the and meteorological conditions in all parts of the farm. Thanks to its operability, this system can intelligently provide convenient services (automatic and semi-automatic) and control the watering of a large farm by dividing it into several segments to water each part individually. Another problem that we dealt with in the course of this work is that of automatic decision making and the use of data from sensors to make predictions.

To fully understand this problem, we suggested using a fuzzy logic system as a basis for merging and abstraction of sensor data. To give the system the ability to control the farm automatically, we also used equations to measure the irrigation time taken from the World Food and Agriculture Organization.

We enjoyed working on this topic because this project is a great idea that help us in our daily life. This system would constitute a step that could contribute to improving the country's economy, thanks to a series of agricultural compensation that avoids dependency, food security, legalization and reduction in water use, etc.

Outlook - Perspectives

Despite the efforts we made to accomplish this little work, and despite the concepts we gained, we believe that our contribution is only the beginning of a long road. The work we did could be improved, supplemented, and continued in several ways, including:

- Machine learning and history use.
- Using drones to take pictures of pants and analyze data to aid decision-making

References

[1] Allen, Richard G., et al. "Crop evapotranspiration-Guidelines for computing crop water requirements-FAO Irrigation and drainage paper 56." *Fao, Rome* 300.9 (1998): D05109.

[2] Radwan, A. A., et al. "Determining plant water use and landscape coefficients of selected nursery and landscape plants." *Misr Journal of Agricultural Engineering* 27.2 (2010): 521-529.

[3] Rouse, Margaret. "What Is IoT (Internet of Things) and How Does It Work?" IoT Agenda, TechTarget, 11 Feb 2020, internetofthingsagenda.techtarget.com/definition/Internet-of Things-IoT. Accessed 02/2020.

[4] Patel, Keyur K., and Sunil M. Patel. "Internet of things-IOT: definition, characteristics, architecture, enabling technologies, application & future challenges." *International journal of engineering science and computing* 6.5 (2016).

[5] Rahimi, Hamed, Ali Zibaeenejad, and Ali Akbar Safavi. "A novel IoT architecture based on 5G-IoT and next generation technologies." *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2018.

[6] Dernoncourt, Franck, and Emmanuel Sander. "Fuzzy logic: between human reasoning and artificial intelligence." *Master's thesis, ENS Ulm* (2011).

[7] Singh, Himanshu, and Yunis Ahmad Lone. "Fuzzy Neural Networks." *Deep Neuro-Fuzzy Systems with Python*. Apress, Berkeley, CA, 2020. 199-221.

[8] Bai, Ying, and Dali Wang. "Fundamentals of fuzzy logic control—fuzzy sets, fuzzy rules and defuzzifications." *Advanced fuzzy logic technologies in industrial applications*. Springer, London, 2006. 17-36.

[9] Vikhe, P. S., et al. "Design of Fuzzy Controller Based on Penman-Monteith Equation for Drip Irrigation.",2009.

[10] Touati, F., M. Al-Hitmi, and K. Benhmed. "A fuzzy logic based irrigation management system in arid regions applied to the State of Qatar." *WIT Trans Ecol Environ* 168 (2012): 189-199.

[11] Weis, Olga. "Top 7 Virtual Serial Port Emulator Software." *Eltima Software*, 29 Apr. 2020, www.eltima.com/best-virtual-com-port-emulators. Accessed 06/2020.

[12] Rossum, G. Van. "Python." *Executive Summary. URL: http://www. python. org/doc/essays/blurb. html* (2009).

[13] Haagsman, Ernst "Collaboration with Anaconda, Inc". PyCharm Blog. Retrieved 26 May 2019.

[14] Kuhlman, Dave. "A Python Book: Beginning Python." *Advanced Python, and Python Exercises* (2012).

[15] Max Burkhalter, "How IoT and smart irrigation are improving water conservation", 7/23/2019, https://www.perle.com/articles/how-iot-and-smart-irrigation-are-improving-water-conservation-40186095.shtml. Accessed 02/2020.

[16] "Types of Agricultural Water Use", 10/11/2016, https://www.cdc.gov/healthywater/other/agricultural/types.html, Accessed 02/2020.

[17] "Just What Is Evapotranspiration?" HydroPoint, 15 Jan. 2020, www.hydropoint.com/blog/just-what-is-evapotranspiration/, Accessed 03/2020.

[18] "Django." Django - PyCharm Guide, www.jetbrains.com/pycharm/guide/technologies/django, Accessed 06/2020.

[19] Cloud computing, https://aws.amazon.com/what-is-cloud-computing. Accessed 03/2020.