ΕΛΛΗΝΙΚΗ
ΔΗΜΟΚΡΑΤΙΑ

ΠΑΝΕΠΙΣΤΗΜΙΟ
ΜΑΚΕΔΟΝΙΑΣ

UNIVERSITY OF MACEDONIA

DEPARTMENT OF APPLIED INFORMATICS

MsC IN APPLIED INFORMATICS

INTERNET OF THINGS IN AGRICULTURE. AN ARCHITECTURRAL PROPOSAL FOR AN AUTOMATED WATERING SYSTEM BASED ON A WIRELESS SENSORS' NETWORK

ANTONIOS V. STOULOS

THESSALONIKI 2019

Acknowledgements

At the first place I would like to thank the Associate Professor Psannis Konstantinos for his guidance, knowledge sharing and assistance throughout the preparation and implementation of this project. I would also like to thank his scientific team for the valuable technical contribution that they offered.

Last but not least, I would like to thank my family, my friends and my colleagues for the inspiration and selfless support that they offered me during the implementation of this thesis.

Preface

In this thesis an automated watering system based on a wireless sensors’ network architectural design is presented. Initially, previous works and approaches are thoroughly described helping in a double sided way both by identifying the current state and by providing knowledge and technical background. This thesis makes a step forward compared to other approaches, as it will be analyzed, since it proposes a system where 3 entities will interact utilizing IoT, Cloud and mobile application approaches. Responsibilities and actions of each subsystem as well as the way of their interaction is explained providing a clear architectural design that could be implemented in the future.

Contents

List of Figures

List of Tables

1. Introduction

1.1 Subject in question

Internet of Things (IoT) is a rapidly evolving technology, having affected most of society's daily needs, which supplies automated solutions causing a significant improvement to the standards of living. The so called "Smart Cities", help solve environmental, traffic and public security issues, applications in pharmaceutical and medical industry, having sensors monitoring patient's condition and adjusting accordingly the required medication, "Smart Houses", where house devices (e.g. air conditions, heaters etc.) are controlled by sensors and are adjusted to user's needs are some of the examples where IoT technologies have contributed to society's well-being. Another sector where many IoT applications already exist, making a positive impact already but with a great margin for improvement, is precision agriculture. Currently, watering fields in traditional ways requires time, effort and significant funds from farmers while processes' optimization fails due to relying on empirical approaches. In this thesis, an automated watering system based on soil moisture sensors and the communication among them would be proposed. Specifically, a decision-making algorithm which will send directions according to the output to a number of actuators which will be responsible for the irrigation of the field, the architectural design for the sensors' communication and simulations regarding system's functionality will be presented. Finally, researches that have already been conducted on similar systems will be analyzed in order to extract valuable information regarding the general architecture of the proposed application.

1.2 Proposed solution

Proposed system's main aim is to assist in automating and optimizing processes that revolve around crop's watering. By combining a number of technologies, all of which belong to and conduct an IoT ecosystem, an application comes as a result which is taking advantage of sensors, IoT network communication, cloud technologies as well as smartphones in order to provide an automated solution in optimizing watering process while at the same time a user may have a clear overview and the ability to intervene any time he wants. The architecture of such a system is presented and thoroughly analyzed in this thesis, diving into details regarding communication protocols that will be used, software engineering design patterns that would ensure system's stability, extensibility and scalability.

1.3 Contribution

Through this thesis, a complete IoT application will be proposed from an architectural perspective. From analyzing the theoretical basis provided by previous researches and the design that will be submitted, a, theoretically, as flawless as possible model will be generated aiming to the optimization of the usage of water resources. On the other hand, proof of dominance of IoT applications over traditional methods will be cited, through sensible justification. Hence, introduction and acceptance of such solutions will become smoother to the average farmer/user who may not be familiar with technological approaches of that kind.

1.4 Thesis structure

This thesis is structured as following. In the second chapter a thorough presentation of previous literature is taking place. IoT is in the spotlight and its combination with the other technologies that will be used in the proposed system is examined. More specifically, various researches are presented where IoT is blended with Big Data technologies, cloud applications and smartphones. A brief presentation of Precision Agriculture is taking place by following its evolution from its creation till present. Finally, a detailed presentation of researches that deal with the integration of IoT technologies with agriculture is concluding the second chapter. This part is, in fact, providing all the theoretical basis in order to design the architecture of this thesis' proposed system. Each proposal is analyzed, highlighting advantages and disadvantages in these approaches, hence a clear path of how this application should be designed is offered through this investigation. In the third chapter the architecture of the proposed application is presented. Each component of it, specifically the deployed sensors, actuator and the base station which conduct the WSN, the cloud and the mobile application are presented. Their general functionality, their design from a software engineering aspect and the way their communication is implemented is demonstrate, justifying the choices that were made in order to prove the dominance and uniqueness of the proposed application. Finally, in the fourth chapter the conclusions that are extracted are presented while at the same time future approaches and extensions of this system are brought into conversation.

2. Literature review

2.1 Internet of Things introduction

Internet of Things is defined as a collection of connected devices where communication among them is possible through a variety of protocols. The selection of a communication protocol is based on the needs of each connected subsystem. The use of IoT devices has rapidly increased and by 2020 it is believed that there will be almost 30 billion connected devices [1]. Nowadays, this technology is being massively used in a wide variety of sectors, and its usage is expected to be dramatically increased. Healthcare, smart cities, unmanned vehicles and industrial production are only a few examples of sectors where IoT applications could automate processes supplying a great benefit to society and world economy. For example in healthcare, in Catarinucci's et al [5] research, an IoT based architecture for automatic monitoring and tracking of all critical entities in a hospital (patients, personnel, biometrical devices) is proposed. The designed system relies on the complementation of IoT from a variety of emerging technologies such as smartphones, Wireless Sensor Networks (WSN) and RFID via a multilevel network infrastructure. An algorithm for a media-based surveillance system for smart cities, is analyzed in [6] in order to cover challenges that have been raised on a security and privacy level while giving a better performance on each node's memory requirements. The previously mentioned challenges are concerns that affects every IoT sector and the solutions that are suggested from each research could be applied in every IoT project. Such a sector is the emerging technology of unmanned vehicles (e.g. drones) which could be enhanced with different kind of sensors, creating WSNs and increasing the capabilities of the whole IoT ecosystem. For example the following survey [7], delivers a clear perspective on how this integration could be done, with the analysis of the respective architecture. Finally, another sector where applications of IoT could be a great support in automating processes is industry. The energy that a plant consumes plays a vital role both on an environmental and economical perspective. Taking this as a fact, in Shrouf's et al research [8], a survey has been conducted in regards to, what is called, "Industry 4.0", where the architectural setup of flexible production volume for plants based on IoT factories is submitted. The expected benefits of such an approach are discussed and presented. Finally Plageras et al [65] taking into consideration the steps that technology have move forward in the fields of cloud computing, video encoding and mobile application propose an IoT system for ubiquitous health monitoring being composed by sensors, actuators and cameras having improved results in the required bandwidth for video and data transmission.

2.2 Thesis description

In this thesis' section, examples of IoT implementations in a wide variety of sectors will be presented. Its combination with other technologies such as Big Data, Cloud and smartphone applications will be highlighted by examining and analyzing other researchers' studies, in order to get a better understanding of IoT and its capabilities. Finally papers, regarding Precision Agriculture and its combination with WSNs and IoT, will be demonstrated. As a fundamental approach we can take into consideration Ray's study [28], which is a thorough presentation on Internet of Things general architecture patterns, fused with varying examples of usage, communication protocols that appropriately fit to the demands of this technology, while open issues are acknowledged. Epigraphically, this research mentions the structural components of IoT ecosystems, the “Things”, which may be sensors, house devices, cars, RFID transceivers, base stations for communication with the external Internet. All these intranets “Things” may produce, should be able to operate and communicate with the least possible power consumption, being uniquely identified, in an interoperable abstract, but still standardized, way. Standardization is great concern for the IoT ecosystem, raising barriers in its evolution as it will be revealed in the following chapters. The selected architecture for an IoT application is strongly depended on the context of it. For example, agricultural applications are, generally, decision making systems based on observations from WSNs in fields, thus a power efficient, low latency, stable architecture, supported by the respective communication protocol, should be arranged. On the other hand healthcare applications, since they will be probably deployed in a hospital or a clinic, they do have access to a stable, high speed protocol like Wi-Fi while power consumption and its source is not a first priority matter, so the architecture will be constructed accordingly. These are a couple of examples, among a lot, presented in the aforementioned study, which reveal the tremendous capabilities IoT applications have, in optimizing society’s well-being. Finally, security and performance issues are top-priority concerns that the IoT ecosystem will have to carry out.

2.3 Big Data and Internet of Things

Internet of Things concept can be deployed through in numerous and diverse applications, thus data could be collected practically from a countless number of sensors for literally every social or economic aspect. Apparently, obstacles such as privacy and security arise in such cases, but it is clear that overcoming these problems, would lead to a great amount of data that could strongly aid the technology of Big Data. Furthermore, it is undeniable that the generated profits could work the other way round, causing an optimization to Internet of Things applications, especially the ones that work as decision systems based on information and observations.

As reported by the IEE Spectrum [9] WSNs and Big Data will be two of the five technologies that will shape the world. In Zaslavsky’s et al [10] research, an overview on plenty of IoT applications that create an emerging amount of data which after its analysis, could provide valuable information for the evolution of each affected sector. Of course there are challenges that have to be faced, such as scalability, integration of an adaptive Machine Learning process due to the diversity of the produced information, performance optimization etc. Plageras et al [11] surveyed the combination among IoT, Big Data and cloud computing and monitoring focusing on the benefits that could occur from their integrational operation. The proposed solution gives an energy efficient “smart building”, based on observations that are supplied by sensors installed in it, which’s measurements are forwarded to a cloud application and are manipulated as Big Data. The whole system communication between the WSN and the external Internet is based on 6LowPAN protocol, which implements the IPv6 protocol but is ideal for low power devices, such as the sensors used in this research. In another study [12] evolution of sensors and their support from cloud and Big Data technologies is demonstrated in regards to healthcare applications. Specifically, data sent from sensors is being mined, analyzed and aggregated on cloud in the pursuit of any patterns and other valuable info that may occur from these computations. Extracted knowledge, except for patients’ condition, could apply to some environmental variables such as humidity, temperature etc., in order to have the safest possible results. In Manogaran’s et al study [13] an architecture is proposed in order to handle, store and analyze the enormous data produced by healthcare sensors in a scalable, secure and performant-wise way. The results prove that this architecture is far better compared to other models. The aforementioned studies, show the great connection that exists between Internet of Things, specifically the amount of data that could be generated, and Big Data applications. Valuable information could be extracted from patterns that are observed in the provided data, leading to great gains, especially to decision-making IoT applications.

Regarding precision agriculture, which is the main concept of this thesis, it is more than obvious that a blast of data could be produced from all the invoked "Things" such as WSNs, unmanned vehicles, smart mobile phones and cloud applications. The occurred observations would not only affect agriculture's optimization, but could also bring a perked status for environment and economy. For example, in Wolfert's et al study [14], a broad overview on Big Data applications in Precision Agriculture is delivered, while socio-economic issues that arise are mentioned. In order to identify and classify these controversies, a conceptual framework is proposed based on a hierarchical chain network management and data-driven strategies. In its conclusion, the study supports that Big Data is a game-changer in smart farming, but questions are addressed, such as privacy, security etc., that must be carried out, in order to maximize the adoption of Big Data in farming. Additionally, in Channe's et al research [15], a multidisciplinary model for agriculture, where IoT and Big Data analysis, among other technologies, are blended. Explicitly, the model suggests a periodical data collection from sensors, where Big Data analysis on it hands out requirements on fertilizers, best crop sequences etc., in order to increase the production with an improved cost control. In a more generic approach [16], Schönfeld et al, suggest that the incorporation of Big Data in farming, via a wide range of technologies like sensors, drones, location services etc. and the contributed data would significantly boost processes, while on the other hand practical concerns, especially from a legal aspect, may act as drawbacks to the evolution of such projects. Muangprathub et al [17] propose an IoT watering system for a smart farm based on Big Data analysis. Information is preprocessed, in order to distinguish noise from valuable observations, thus enhanced accuracy and efficiency will be accomplished. This method is followed by data reduction leading to a model's creation/expansion, which classifies, clusters and associates information. Algorithm is based on sequential if/then statements. A review on project Agrodat takes place [18]. Agrodat is a promising project, developed in Hungary, which aims to establish knowledge in agriculture, based on WSN and Big Data. As said in the research, through this framework, uncertainty is remarkably reduced. To be more precise, physical quantities such as wind speed and its direction, temperature and relative humidity are measured creating an immense amount of data, which is handled and aggregated on a cloud application. This process requires a high computational power due to diversity of semi/un-structured conceived data in order to extract patterns and help the optimization of cropping methods. Balasas et al [66] illustrate in their research the affect that has routing in package transmission for a Big Data project, taking measurements in real-time and observing any occurring errors that the most important routing algorithms have in such applications.

2.4 Cloud computing and Internet of Things

The aforementioned studies show a great correlation not only between IoT and Big Data, but also to their main connector which is Cloud computing. Cloud computing, as defined in Wikipedia, is shared pools of configurable computer system resources and higher-level services that can be rapidly provisioned with minimal management effort, often over the Internet. Cloud computing relies on sharing resources to achieve coherence and economies of scale, similar to a public utility. It is important to mention here that such services are available to anyone, at any place. Cloud is separated in three main service models, Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS), while the past years new trends and technologies have drew the attention such as Mobile "Backend" as a Service (MBaaS) and Serverless computing. Most of these service models can be used and integrated with IoT applications, and especially SaaS, PaaS and MBaaS may come handy in evolving applications on a computation and innovation level. Cloud services have come into existence since the beginning of this millennium and their usage exploded in the middle of the first decade of 2000s. Definitely, a milestone point was the creation of Amazon Web Services from Amazon at 2006, which is still a major player in Cloud computing, while other companies such as Google offer their respective packages. To writer's personal knowledge and experience, Amazon cloud services can stably serve a commercial application which can reach up to six hundred million requests per day.

Cloud applications can be found almost everywhere playing either a significant role or a more subsidiary one. In [22], the usage of Cloud in a variety of applications is presented, in order to show its importance. In healthcare, Cloud can support the improvement in efficiency, effectiveness and cost optimization. For example, health records for patients kept in cloud, could be used by applications responsible in backing the decision-making process, for a patient's treatment. Additionally, the created and stored data, could be available to anyone, so third world countries, which face significant hurdles in their medical system, could get advantage of this information. Of course, concerns such as privacy arise and should be dealt, but Cloud could greatly assist the improvement of this sector. These technologies could be applied to banking sector too. Web banking systems are huge and convoluting programs, where security, data integrity, and stability are more than essential. Using cloud applications, through micro-services, these issues could be handled by different service models, in a best-fit logic, thus better management of the whole banking system can be achieved, since it will not be fully depended to a monolithic server. Finally, the study mentions the relationship between Cloud and agricultural sector, where the enormous data, which is produced, can be collected by sensors and vehicles installed in rural areas, and serve the automation of crop processes, accomplish better decision-making systems for farmers, through the analysis and aggregation of this data, while at the same time remote monitoring could give a total overview of any field no matter the time and the place.

In Varghese's et al study [20], the trends in the modification of cloud infrastructure are analyzed and the architectures that should be introduced are presented. As in any technology, cloud moves from a monolithic logic, which only aimed to a load balancing among resources, to be broken down to many micro-clouds and cloudlets in order to improve latency, power consumption and frequency of communication between server and clients. Another emerging trend, is ad hoc clouds, which would work the way ad hoc computing would. This technique, could take advantage of clients, such as smart phones that are idle at the moment of request, by making usage of them as computing resource. This architecture is not new, since volunteering computing has been quite used the past ten years. On the other hand, micro-clouding is a better fit for fog computing architecture and, to writers' opinion, closer to the IoT world. All the above trends, if they resolve challenges as security, performance and stability, would benefit the technologies such as IoT, Big Data, the general enhancement of the already offered services of Cloud computing and of course self-learning systems. Energy consumption is a critical factor of IoT technologies. Taking as a fact that these devices can play a vital role in edge computing in the years to come, the integration of IoT and cloud computing, called Cloud of Things (CoT) is an interesting technology to be take a deep look at. In Ning's et al [21] research a green and sustainable energy framework is proposed, based on edge computing. The general concept sets the hypothesis that edge devices (ED), which belong to the world of IoT, would be deployed as backups and they would serve as computational resources in the networks they create. Cooperation among EDs is pivotal as it would highly improve scalability, stability and the mechanisms to eliminate latency and bottlenecks that might occur. The whole system is configurable, on an aspect of how many EDs would set up each required computational network, based on a heuristic algorithm. The results of the experiments which took place show that the proposed system is quite guaranteeing the sustainability of the created networks.

The aforementioned studies show the immense possibilities that could be brought into existence through the usage of Cloud technologies. Of course, as it is mentioned in [22], Internet of Things, and especially regarding agriculture, is no exception. Its diversity, the amount of information that could be extracted, its significance for the environmental and economic stability and evolution makes it a very interesting field not only to apply accepted Cloud solutions, but to expand the limits of both technologies. In Botta's et al study [23] a full overview is supplied on how these two technologies could be merged. As mentioned, one of the very first, yet very successful and commonly used even nowadays, WSN's that were created were networks of RFID tags. RFID is a device recognition system, where a transmitter sends a message with a unique tag that activates one unique receiver, which will finally read the sent message. Due to the exponential growth of interconnected things, such technologies face the critical issue of lack of "uniqueness". The obvious solution to this is the usage of IPv6 stack for IoT networks too. This approach overcomes the prior problem, while the IPv6 standard secures the required interoperability with already existing technologies and services, which are offered and deployed via Cloud. Cloud is divided, as said before, in three main service models, Software as a Service, Platform as a Service and Infrastructure as a Service, each of which could be potentially used by an IoT application, according to its needs and requirements. This is a win-win condition since IoT networks and applications can extend their computational capabilities when using cloud resources and services while on the other hand they can offer a huge amount of information to applications that need them, such as decision-making applications, machine learning systems and so on. Actually, Cloud also works as a middle layer between IoT and Big Data technologies by delivering the required data to the second stipulator, extracting patterns and valuable information, while it directs and manages the first stipulator's systems in order to work in a more optimized way. As Stergiou and Psannis illustrate in the overview they conducted in their study [67] or in the work that was proposed by Agrawal et al [68], there have been some significant advances over the past years regarding the cooperation of the aforementioned fields in delivering quality applications containing components of all 3 technological aspects.

As it is understood, Cloud can be pivotal in an easier acceptance and intrusion of IoT applications in more and more sectors of everyday life. It offers high computing power, endless storage and it simplifies communication of IoT with their "extranet" since it works, as said before, as a middle layer. IoT can be found developed in almost any kind of sector. Healthcare, agriculture, smart cities, smart houses, industrial applications are only a few examples. Cloud can support every kind of potential applications due to its abstract nature, thus a global evolution can be expected due to the combination of these two technologies. Of course there are challenges that should be faced. For instance, security and privacy, as it has been mentioned several times, is a serious concern for cloud applications. Stergiou et al [69] thoroughly analyze these issues and how it affects IoT and Big Data applications from a sustainability and efficiency aspect of the developed systems. On the other hand, performance is equally critical and improvements in "classic" software applications, on an architectural, coding and design aspect, should be applied in the prior mentioned technologies too. Also reliability and stability of such decoupled systems is something to be highlighted while at the same time legal and privacy issues might expose major dysfunctionalities. Finally, standardization in communication in WSNs, design patterns on a software development level, efficiency in power consumption and integration of data are some of the top priorities that should concern anyone willing to deal with such applications. This study [23] suggests that identification of each "Thing" is essential and IPv6 seems like the perfect fit, while Cloud services should be enhanced in order to optimize their usage by IoT applications. Scaling is also a matter of interest, with a rapidly growing number of "Things", and the proposed approach suggests the existence of "autonomous isolated WSNs in globally distributed interconnected networks" [23].

As said before, Cloud can be integrated in IoT agricultural applications providing support, especially to the ones that require a stable decision making system which will direct the actions of the "Things" in the network. Weather prediction is critical in such projects since it could potentially minimize power consumption and generally optimize the way an agricultural WSN would work. Such an application is proposed in Molthan's et al research [24]. This Cloud service predicts the weather in a scalable way, serving in the context of IaaS model. Its Virtual Machines (VMs) are divided in clusters and once a day a cluster is chosen to make a weather forecast, by creating a modeling structure for this process, based on data which are gathered on S3 Amazon Web Server.

2.5 Smartphones and Internet of Things

Another sector that can be strongly combined with Internet of Things are smartphones. The past years, applications have been developed where a user can monitor, for example, his house, getting information for temperature, humidity, movement inside the house etc. These measurements come from a wireless sensor network installed in his property and the user can take actions like turning on the heat, play music from a streaming application through a smart television, firing an alarm and in general control every "smart" object. A very well-known example of such an application is Google Home [19], which shows the efficient way IoT and smart phone applications can work together. Another application where smartphones and WSNs are blended in order to deliver, is described in Ghose's et al [25] study where users' smartphones work as "sensors" through their installed sensors, such as GPS, accelerometer, gyroscope, in order to capture bumps and potholes in a route. This information is uploaded onto a remote database in real-time. A backend service aggregates the supplied data and updates the maps that are offered through the application. In this study we observe a different kind of usage of smartphones regarding the IoT world. In Google Home application, a smartphone works as a "director" since it presents the monitored data and gives the option of actions. In the prior presented study, smartphones are, in fact, the ones that provide data and act as a fundamental part of an external WSN system. This diversity shows the extended capabilities that are supplied by combining these two technologies.

“Smart cities” is a rapidly growing concept where automations take place on an urban level making society’s life better. WSNs play a vital role in the development of such applications which will offer security, comfort and generally improve the well-being of citizens. Since the vast majority is an owner of a smartphone, it is obvious that the aforementioned WSNs could be created by them and their capabilities could enhance the functionalities of the developed applications. In Khatoun’s and Zeadally’s research [26] the big picture of the current status, challenges and possibilities that occur through the extended usage of IoT devices in a “smart city” context. For example applications for controlling and comparing the usage of natural resources in a household (electricity, water and gas usage etc.) can be developed, while on the other hand mobile phones can be engaged through the data that they can supply. More specifically, estimation of road traffic volume, transport demands, on-the-fly discovery of free parking lots are a few examples of applications that could be developed in the context of blending IoT and smartphone technologies. Google Maps has already integrated live traffic monitoring based on data sent from drives who are using it. Additionally, based on previous data and patterns it can estimate the duration of trip according to departure time. Google Maps is a great case of proving the quality level can be achieved with the usage of IoT and WSNs in smart cities. Of course, it is critical that the developed projects should be scalable, cost effective and power independed while respecting users’ privacy and security.

Agricultural field is another sector where blending IoT and smartphone could lead to solutions which will optimize the whole process, provide automations, ending up to a better, cost-effective, environmental friendly production. Until the end of the decade of 00s, agricultural processes were mostly based on human observations having no or little interference of smart devices aiming to their automation. Existing appliances, were either expensive or hard to reach by farmers [27]. Thus, the development of smartphone applications was a one-way road to follow, in order to improve the crop process. In [27], researchers present such projects, concentrating on the way smartphone sensors are used in agriculture, dividing them in four categories. Indicatively, under the farming category, researchers classified disease and diagnosis recognition approaches, where pictures of plant leaves, taken from smartphones, were analyzed and through pattern extraction algorithms plants’ health status could be identified. Similar developed solutions, are the ones that calculate needs of the plants regarding fertilizers, water and nutrition ingredients. Farm management applications is another category where farmers could organize human resources’ timetable, vehicle monitoring and natural resources management while on the other hand the third category is dealing with information systems, where location, disease and pest data are collected in order to extract patterns which will lead to solutions for potential problems. Finally, the last category is about extension services to farmers. For example, educative applications have been developed, teaching the best ways to deal with agricultural problems, designed by experts and delivered by state and private organizations. This research, is summing up with the conclusion that there is a lot of margin in developing agricultural applications. The evolution of smartphones and their sensors could lead to a wider range of extensions that is yet, not discovered.

2.6 Precision agriculture

Agriculture has been an essential sector of survival and development for thousands of years. It has been one of the most, if not the most, neuralgic source of victuals both for human and animals, while at the same time its production improvement is a major cause for economic development. For years, these processes were executed by farmers and agronomists based mostly on experience. Whenever actions were needed, human involvement was required, for example in order to take some measurements or water a field, meaning that this would take time and effort. The last years, as in every sector, agriculture is evolving through moving towards automation of its processes, engagement of decision-making systems and continuous monitoring of fields' status. This integration of technology in agriculture is called "Precision Agriculture" (PA). Quoting US's National Research Council definition given in 1997 "Precision Agriculture (PA) is a management strategy that uses information technologies to provide and process data with high data and spatial resolution for decision-making with respect to crop production" [2]. PA dates back to the middle of 1980s [3], starting with sensors to monitor humidity and environment temperature and reaching to a level where we have automated watering and fertilizing, unmanned vehicles to perform a variety of agricultural processes producing a high quality outcome. Though evolution is undeniable, there is a lot space for improvement and expansion of automation. It is obvious that Internet of Things can play a major role in the enhancement of PA via WSN's, unmanned vehicles, support from decision-making systems, integration of Cloud and Big Data technologies, while having smartphone applications playing a supporting role in connecting farmers to any potential or existing IoT application. Such an application will be proposed in this thesis. Its architectural design would include most of these technologies while the best possible communication protocol should be chosen in order to ensure its stability, power efficiency and scalability. To be more specific an automated watering system based on a wireless sensors' network, which will take soil moisture measurements, sent through a base station to a cloud application which will decide if watering process should be turned on or off based on soil moisture measurements. Farmers will have control and monitoring of the system through a smartphone application.

A major crop process which can be highly optimized through automation is watering. From the writer's personal experience, watering a field requires contacting other farmers, who all use the same water valve for their fields, in order to set a timetable where the watering hours (night hours are not excluded) and duration are based on experience or a specialist's opinion which might be completely outdated. This takes time and effort while cost is high since farmers must be at the field to switch on or off the water valve, so it's possible that they will have to drive for a long distance. Bad weather conditions can make this process even more difficult. The greatest deficit in this process is that the current field circumstances are not taken into consideration and so the outcome might not be the best one. IoT devices could provide solutions with less effort for farmers, since watering will start and stop automatically and profit of the production will be maximized since the underlying algorithm will calculate effectively if watering should start and the duration of it and as a result, water waste and electrical power consumption will be decreased. For example in Asmamaw's research for conservation tillage, 9 to 40% reduction of water usage was recorded for fields in Ethiopia [4].

2.7 Integration of Internet of Things in agriculture

Over the past few years there have been quite a lot of papers, researches and applications on how IoT can boost Precision Agriculture, whether this is achieved through field monitoring or automating processes like watering, fertilizing and harvesting. Results have shown, almost in every case, that cost and electric consumption are decreased while water usage is optimized leading to a high quality production. Having this as a fact we will examine, present and analyze researches in order to discern the current status of such systems and discover what the trends are, in order to support the proposal of this thesis.

In 2016, Carreras et al [29] proposed a testbed to evaluate an IoT platform in the context of PA, which was built on FIWARE framework. FIWARE is a framework of open source platform components which can be assembled together and accelerate development of Smart Solutions since it can interface a number of technologies, including Internet of Things [30]. Its source code can be found on GitHub and it has a powerful and growing community. In this research FIWARE was used in order to ensure and evaluate scalability, interoperability, communication and connectivity among developed agricultural applications. As it is mentioned previously, some of the main concerns of the IoT ecosystem is power consumption, deployment cost and communication among “Things”. Using a service oriented architecture based on the aforementioned framework, which can cover all application domains while it is designed to support a wide variety of technologies, interoperability and low cost deployment are ensured, eliminating the prior mentioned concerns.

Test scenario included distributed WSNs among different regions or countries, responsible to measure physical quantities related to environment and water valve actuators. This project aimed to a stable architectural design where computational resources was the only thing to be modified on the fly, according to requirements. The selected sensors’ communication protocol was ZigBee, chosen for its low power consumption, stability and required communication distance. Via data collection, which is context-aware since every sensor is registered with respect to its responsibility, system could extract knowledge in order to facilitate a decision-making functionality on a publish/subscribe communication model between base station and the directing cloud application. Results have shown a better performance when payload was low and as number of registries increased. As a conclusion, writers suggest that this paper is good starting point as a tutorial for anyone interested in developing a FIWARE application.

Referring to a previously mentioned argument, a great deficit in current agricultural processes is that weather circumstances are not taken into consideration or they can make manual labor far more difficult. For example, when rain falls, plants may get the required water for their needs but a satisfactory water quantity might be unexploited. If it could be harvested and reused, production cost would be decreased. Due to this situation, Sukhadeve and Roy developed and presented an advanced algorithm for water harvesting using IoT technologies [31]. A water tank will be used as a harvesting mechanism and a top to bottom farm setup is proposed, so water would travel from top level fields to the lower ones. An irrigation efficiency formula is developed in order to monitor the quality of water usage. Data for this formula will be granted by moisture sensors installed in each level of the area. At the same time, as an extension of the core functionality, sensors would act as pest repellers by transmitting low-high frequencies in an environmental friendly way. The selected communication protocol in order to support the operation of the system is Wi-Fi technology. To writer’s opinion, Wi-Fi is definitely a bad fit for this kind of applications. Its energy consumption is far higher than other communication protocols, the distance it covers might not be enough, even in a rural area, where no obstacles interject with the transmitted signal and finally its data rates are exceedingly high for the needs of an agricultural IoT application.

Throughout this overview of previous works in the field of integrating IoT with Precision Agriculture, it has been solidly indicated that energy consumption is major player in developing a successful and stable system. Since it will be deployed in a rural area, where an electrical outlet is not very easy to be found, the WSNs should rely on batteries, for example, and the chosen architecture should ensure their extended life duration on a power consumption aspect. Also, the algorithm used, should take into consideration this concern and be designed in such a way. Many researches have been done on developing energy efficient algorithms in this context.

One of them is Dhall's and Agrawal's study [32] where they propose a duty cycling algorithm for PA. The designed system is composed by a wireless sensor network, where each node would be responsible for collecting physical quantities such as temperature, humidity, air quality, soil moisture and images, where the execution will be done by sensors installed on drones. All this data would be aggregated at a base station and forwarded to a gateway. Inspired by Microsoft's recent works on PA, the chosen aggregation algorithm could potentially reduce energy consumption. The algorithm performs efficiently especially in bad weather conditions. Compared to other algorithms, results have shown that this proposal outperforms other approaches.

In detail, this approach, as said, was inspired by the Farmbeats project, run by Microsoft, which is a benchmark research work on PA. Flight-path planning of UAVs techniques are recommended while novel weather-aware IoT base station designs and inference techniques for compression of aerial images are included. Dhall and Agrawal introduce an Improved Duty Cycle (IDC) algorithm which is compared with two other similar appeals and outperforms them according to simulations in Network Simulator 2 in terms of energy efficiency and QoS. Epigraphically, duty cycling means that time is divided in periods and in each period a node is either awake or asleep. As the value of duty cycle falls, network's lifetime is extended. Additionally, the selection process of the path for the communication between a node and the base station, would take into consideration current energy status while the data transmission technique is affected by this variable too. Base of the proposed work, was Raghunathan's, Kansal's et al research at 2007 [33], in which a predictive algorithm for the data that would be sent by a node was developed. Results, regarding data rate of soil sensors, captured images and video simulations, in NS2 have shown significant improvement against a no duty cycle and previous duty cycle algorithms with varying weather conditions. It is important to mention, that in adverse weather conditions IDC was far better that the other two algorithms.

A more controlled environment, and a possible proving ground for the success of using IoT in agriculture and support its extended application, are greenhouses. Due to their nature, they are smaller areas than typical fields, environmental variables like temperature, air and humidity are much easier to control and monitor, bad weather conditions affect them less than fields and it is quite possible that they are close to a source of electric power, like an outlet, or in reach of Wi-Fi signal. Still automation processes can be applied to greenhouses too and they are quite similar to the ones applied to fields. Hong and Hsieh proposed a relevant application [34] where an integrated control strategy algorithm for irrigating romaine lettuce via Bluetooth was developed. Hsieh et al, in a research the conducted in 2010, reached to the result that drip irrigation is positively affecting a plant's growth in contrast to traditional techniques. Apparently, the most crucial factor in a crop’s growth, is the environmental conditions which are composed by temperature, humidity and air. Having an adaptive algorithm that could monitor a greenhouse in real-time and sense the needs of plants was one of the requirements of the study [34]. Thus, a WSN is deployed in it where temperature, humidity, soil moisture and air radiation are measured and according to the data they produce, a microcontroller sends them to a base station, which decides if watering should be irrigated. The selected communication protocol was Bluetooth, since it is stable and covers the requirements in distance, latency and data rates.

Outcome has shown that plant growing was almost equivalent in a comparison between this algorithm and classical approaches where irrigation is depended to a predefined timetable. However, electricity and water consumption was decreased from 10% to 60% in five different test cases, proving that IoT integration can massively help crop production. It is important to mention here the issues were raised because of using Bluetooth protocol for the communication among sensors, microcontrollers, activators and the base station. Even in a greenhouse, a small-scale area, with a limited number of obstacles, data could not be transmitted to the furthest sensing unit. As described in the research, the distance between base station and this location was 26 meters and 3 brick walls interfered the eye of sight. Romaine Lettuce is a short plant, so it does not affect the transmitted signal with its nature, unlike an olive tree for example. We can easily realize that in field conditions, Bluetooth protocol would be a bad fit, since the distances that would have to be covered are far greater, maybe more than 1 kilometer, and high trees would interfere the transmitted signal. For these reasons, another communication protocol could be more suitable for the needs of the application that will be designed in this thesis.

Internet of Things is a technology that its capabilities are highly extended when its usage is supported by smartphone applications. In that way, farmers can have an easy-to-use and familiar approach to monitor, control and take decisions for the IoT systems that are installed in their farms, a critical feature that can help having a successful human intervention, to whom automated processes will adapt. A study where the aforementioned approach is designed and implemented is Parameswaran’s et al work [35]. They deployed a WSN system where soil PH level, moisture and temperature are measured, values are collected and forwarded by an Arduino microcontroller [36] to a server. Microcontroller is powered by a 230 V battery, while it endorses the functionality of a WSN through contacting with a server, providing an LCD display where measurements are shown and finally serving as a switch for the watering process. It is clear that Arduino microcontroller is the most essential and critical part of this IoT system since it is a transceiver of data and actions in order to support the required automation.

This data is available for monitoring by an Android application which requests it from the server. Switching on and off the water valve are also offered as actions from the application, while at the same time the whole system can operate automatically and the function of the water valve is controlled by the microcontroller according to the available data. System is set in sleep mode if the received data do not exceed the set thresholds. As writers conclude, the whole process can work without the need of manual intervention, which is reducing cost and effort while results have shown that water consumption was indeed optimized according to plants’ needs.

Renewable energy sources are, apparently, an area that could solidly assist IoT applications for Precision Agriculture. Rural areas like fields, may not be close to an electric outlet or the batteries used, have a limited lifespan while they are not environmentally friendly. On the other hand, solar or wind power are typical examples of resources that can contribute to the operation of any system without worrying about their environmental cost nor about the scarcity of them. Deployment cost, might be higher but in the long term they are more beneficial on an economic aspect too, over classical approaches. These techniques could be applied not only in rural areas and fields but also in greenhouses, crofts and gardens in households. WSNs, actuators and microcontrollers for data gathering and aggregation, calculation, direction or forwarding to a server and receiving back a set of instructions are going to be operated by photovoltaic panels or wind harvesting energy adapters.

In the prior referred context, Archana and Priya developed an automated watering system for a household garden based on a microcontroller, powered by a photovoltaic panel [37]. System's main parts was a moisture sensor module which was attached to the microcontroller and taking scheduled measurements. These measurements are forwarded and presented by the microcontroller on an attached LCD screen. At the same time the logic of the system runs on the microcontroller, which also acts as an actuator and when values drop off the set threshold triggered a water pump. It is important here to mention that in this approach, the writers presented a couple of indirect ways to measure moisture. The first one is an acoustic technique where speed of sound in relation to the saturation of water in soils can give more accurate and calibrated results than simply taking the direct measurement of the sensor. The second one is based on measuring the electrical resistance of soil which is correlated with the existing moisture and gives more accurate results, too. Concluding, the whole process acts in a simple logic, since requests for watering are satisfied in a FIFO ordering.

Apparently the whole concept of deploying an IoT system in a rural area has a couple of significant drawbacks that must be resolved by researchers, compared to setting up one, in a controlled environment. Firstly, the communication protocol that will be selected must be cost-effective, consuming the least possible energy resources while fulfilling distance, data rate and latency requirements for such an application. Secondly, the selection of the energy resource that is going to provide power to the whole WSN system. Except for designing an energy-efficient algorithm, where nodes' operation must be scheduled in an optimized way while on the other hand, the application must be based on a sustainable resource that is going to last as much as possible. Batteries are an obvious choice, having cases where it has been observed that sensors' lifespan extended to 10 years [38] [39] [40]. Due to the rapid changes in the field of energy, it is very possible that critical improvements are not taken into consideration. In any case, a satisfactory option which is environmentally friendly and can serve even demanding systems are renewable energy sources. Especially for a country like Greece, where solar or wind power could be harvested almost the whole year round, such approaches must be encouraged. A study that covers the above observations is Gutiérrez's et al work [41] where an automated irrigation system based on WSN and GPRS module is presented.

Compared to the previously mentioned research [37], although both are in the same context, meaning the exploit of moisture sensors in order to actuate a watering system powered by solar power, this is a far more complex system since an enlarged wireless sensor network is deployed in a field, having to cover the need of communication between sensors and base station, and forwarding gathered data to a remote server while receiving instructions from it. As it will be presented, the whole system had to implement a communication protocol for each case respectively, where needs and requirements differ. Additionally, the developed algorithm is far more adaptive, combining manual and automatic irrigation. Finally a web app was developed not only to present data but to give the ability for human intervention on the whole process.

These are said to prove, that systems developed for fields and large areas are more possible to have greater requirements than the ones that are developed for household gardens, for example.

Getting into details regarding the aforementioned study [41], the proposed system was composed by distributed controllers which monitored temperature and soil moisture, transmitting data from a gateway to a web app, which was used as a presenter and actions could be dispatched, via a cellular Internet interface. The designed algorithm was based on a threshold logic, in order to trigger actions while power was supplied by photovoltaic panels. Writers highlight that agriculture sector uses 85% of available fresh water resources, a high percentage which indicates that usage must be optimized. In order to achieve this optimization, future used amounts must be determined. Because of that, on an empirical aspect, Crop Water Stress Index was defined over 35 years ago in order to verge required water amount. Over the years, this index was modified and calibrated, having reached to, what is called, evapotranspiration index which takes into consideration many factors such as weather conditions, solar radiation, humidity, temperature and more. Algorithm has a 4 level design approach. Manual irrigation for a predefined duration triggered via the web app, scheduled irrigation where timetable could be set by the user, automated irrigation for a predefined duration where trigger condition were moisture values collected by the sensors; if one's measurement was below a preset level, watering would start. Except for the main decision-making algorithm, power management algorithms were used in order to harvest energy in the most efficient way in order extend system's lifespan.

On a hardware aspect, writers decided to pick ZigBee protocol for the communication between the nodes and the base station. ZigBee is a protocol implementing the IEEE 802.15.4 which has become very popular for IoT applications. Its low energy consumption, satisfactory distance range, offering stability and a satisfactory data rate, have made it a go-to choice for WSNs design and implementation. An extension of ZigBee module was used for the implementation of the application, XBee-PRO S2. Keeping energy consumption on low levels while having a satisfactory covering range for a field, 1500 meters, is a top choice for similar applications. On the other hand base station would communicate with the web app via cellular technology. Impressively high speeds, stability, security, interoperability and, an always increasing, coverage have made it a safe choice especially if it is taken into consideration that 5G cellular communications are approaching, offering a great extension of systems' capabilities. These two modules would be applied to the respective microcontrollers that will be responsible for taking charge of directing and running the whole system. Though cellular protocol might have high energy consumption, the proposed energy resource is solar power which is renewable and, if we keep in mind that this study took place in Spain, a country where good weather conditions prevail, any needs that may occur could be covered.

Results were quite impressive. The system was tested for 136 days and water savings reached up to 90% compared to traditional agricultural practices, proving that it could be used in areas with water scarcity in order to improve sustainability. Algorithm's adaptiveness and the integration of a renewable energy source make the system a quite powerful decision-making program allowing control, monitor and intervention to system's functionality.

As seen in the previously analyzed researches, a wireless sensors' network may be placed in a household garden, in a controlled environment like a greenhouse or in a rural field where human intervention is quite more difficult than in the other 2 examples. Furthermore it is notable that the deployed WSN, in each previous case, may refer to a different type of agriculture, thus it is very possible that the developed algorithms will differ respectively. Additionally, each subsystem may communicate with another one and data must be aggregated. Any deployed IoT structure, may have more than one WSNs, each of which may be responsible to measure different kind of physical quantities, where some of them might be easier to measure if they belonged to the same cluster. So systems shall be designed in a scalable, abstract way aiming to be able to aggregate data from different clusters, identify transmitters, execute the correct calculations or correctly forward data and finally act according to the directions of the respective algorithm.

Ferrández-Pastor et al proposed such a system where a greenhouse is divided in homogenous clusters, in terms of measured physical quantities leading to a, finally, heterogeneous and scalable platform in order to collect, process, store and monitor data [42]. The system is affecting positively factors that ensure the optimization of crop's growth along with actuators and is divided in 6 subsystems, each of which is influencing a respective factor. Researchers mention the slow PA integration is due to suspicion against technology from agricultural community, cost deployment as well as initial expectations against to final results. Their research aims, through the proposed application, to defeat such claims.

Each cluster will have a group of sensors and actuators, where they must communicate. The selected communication model is MQTT (Message Que Telemetry Transport), where latency, stability and small packet size over weak networks is provided. Additionally, it handles a two way communication on a publish/subscribe logic. Regarding the communication among sensors, actuators and the gateway, Bluetooth protocol was selected due to its secure and stable nature. Although it does not cover a wide distance range, it can satisfy the needs of a greenhouse. As regards to the core functionality of the designed system, messages sent from sensors to the local gateway, will be forwarded to a cloud application, responsible for the storage, monitoring, analysis and direction of the subsystems. As a sequent, data serves as input to the running algorithms and then the outputs will be sent back to WSN's gateway which will notify the respective actuators. This is a multidimensional application, where different kind of communication models are chosen, like web sockets, RESTful APIs etc. The chosen protocol for the communication between the gateway and the rest architectural components is Wi-Fi, since the system is not in a suburban area, thus it can be autonomous, on an energy context, by being close to electrical outlet and, because of that, Wi-Fi is one of the best choice on a data rate, stability and security aspect.

It is important to mention here that, on a hardware aspect, the water will be granted by a tank to the greenhouse thus 3 WSNs are deployed, where the first one is used to monitor water tank having PH, EC and temperature sensors while there is an additional nutrient control. Photon IoT microcontroller is used for this WSN. The rest WSNs are based on a Raspberry Pi microcontroller, monitoring radiance and greenhouse climate respectively, having the relevant sensors and actuators. Researchers suggest the usage of a smart phone as a local gateway. Though it covers every need may occur to such a system on a computational capabilities aspect, writer does not consider it as a wise choice since it raises deployment cost rapidly while a microcontroller could cover all needs in a costless and more energy-efficient way.

Results have showed an improvement of 20% in water consumption using evapotranspiration (FAO Penman-Monteith [43] method) which is used to calculate the water needs of a crop. Researchers highlight the adaptiveness of the developed system both to different kind of agricultures and the current evolution state of a plant's growth, the need of human modification towards the optimization of the system and the extension of the capabilities of the application. This can be done by adding sensors to the system according to more complicated and evolved formulas in order to improve the whole process which can be, as said, a future work on their research.

Despite the rapidly increased usage of IoT applications, not only in agriculture, which is the main pillar of this thesis, but in a high number of sectors that affect daily life, economy etc., this technology lacks standardization [44]. It is out of question that the nature of the components that conduct an IoT system are diverse among them. Sensors, actuators, cameras, microcontrollers, communication modules, smartphones and cloud applications could form a complete functional application, where transmission of information should take place in a successful way. A major drawback though, is that in order for such an application to communicate with another one, probably a high number of modifications would be required in order to align these two systems. This problem has been identified and its solution lies on multiple levels. Protocols are a major concern that its resolution bears in proposals such as 6LowPAN and ROLL, an implementation of the IPv6 stack for low power devices and a routing protocol for heterogeneous networks respectively. These projects, though, are still evolving and are not stable yet [44]. On the other hand, on a hardware aspect, EPCglobal resolves the problem of uniquely identifying each "Thing" by introducing an Electronic Product Code for RFID tags [44]. Finally, on a more theoretical approach, "M2M standard introduces constants for Machine-to-Machine and sensor networks such as integration, addressing, QoS" [44].

Giri's et al research [44], taking into consideration the prior mentioned observations, comes up with an agricultural IoT framework for resources optimization (water, fertilizers, insecticides and manual labour) called AgriTech. This application is in the same context as the aforementioned studies and the ones that their analysis will follow. A number of sensors and actuators, uniquely identified by RFID tags, compose a wireless network being responsible for taking measurements and execute directions given by the interconnected cloud and mobile applications. An energy efficient communication protocol should be selected while at the same time its stability is a number one priority in order to avoid package collision. The collected data should be forwarded to a local gateway either in a multihop or single hop pattern according to current application needs. Microcontroller at gateway should be responsible to aggregate data before uploading it to a cloud application. An algorithm, at this point of the flow, will calculate if watering is required for the system getting extra input by weather APIs in order to reach to a safer and more optimized conclusion. Agritech's cloud ecosystem has elements from every service model of a cloud application (SaaS, PaaS and IaaS). Except for output which might generate instructions for the actuators, final result can be influenced by human intervention since the accompanying mobile application would allow an agronomist to rate the result and modify it according to its opinion. A farmer would have the ability to access the aforementioned data and decide which action to take regarding field's watering. Finally, local gateway will get the final instruction at each request and act accordingly.

Researchers end up trying to highlight the importance of such frameworks to the world economy. National economies that rely on agriculture, such as Indian, Chinese and Pakistani [44] would be benefited by applications like AgriTech due to the optimization of crop production while at the same time an eco-friendly process is introduced due to the fact, which has been noticed by a significant number of researches, that water and electricity consumption are reduced and human effort is less and less required. The diversity of AgriTech is something to highlight since the designed system integrates the usage of WSN, mobile and cloud applications. This fact gives the possibility for the expansion of the responsibilities such an application would have, according to the evolution of the conducting components.

It is plain to see that, though the seeming simplicity of such farming management systems' core logic, the amount of information that has to be handled correctly is quite large. Especially, when it comes to the end users, where it is very possible that their educational level is not high, although this is changing to the best over the years, the designed applications may be hard to use, thus disappointment against IoT systems will grow as reported by Sorensen et al. [45]. Additionally to that fact, it is clear that a main concern of managing such systems, is that utilization possibilities have not been fully exploited. So it is obvious that information handling and its presentation should be done in an optimized way. In order to achieve that, standardization has to be installed and defined [46]. Apart from that, it must be ensured that different systems can communicate and present information in a common way, thus move to a gradual optimization. Besides, the dissolving interoperability, that would be a result of standardization, would help enhance IoT systems' functionality and evolvement. At the moment, since standardization is on an ongoing status, autonomous communication among different IoT applications is almost impossible. Based on that assumption, a farmer is strongly connected to the first IoT system he will install in his field. Transitioning to or integrating with another one, would be an effortful process since all existing data should be transferred and be "transcripted" in a way that is compatible to the new system. Standardization would eliminate such problems giving freedom of movement both to end users, regarding the systems they will deploy in their fields, and to developers since knowledge sharing would be easier, thus exponential improvement in IoT applications would be almost certain.

Regarding each system's sustainability and viability, autonomy plays a vital role in this. A farming management system should, ideally, work in an adaptive way where, as time passes, optimization and improvement of the taken decisions would take place. Big data would help in reaching such a goal. Except for that, mechanisms for self-management should be introduced for capabilities extension. Self-healing, self-configuration, self-optimization and self-protection are some of the concepts would improve FMSs. For example, self-healing would be the isolation of a malfunctioning subsystem (e.g. a sensor) or troubleshoot connectivity issues. Regarding self-configuration, adaptiveness to the addition of new sensors, actuators, gateways and transformation of communication models (e.g. move from star to mesh topology, or change of communication protocol). As for the self-protection a system should be secure against malware attacks and exploit of information. Finally, self-optimization is the infinite process to taking better decisions, reducing latency in system's communication and increasing stability. This path is depended on numerous factors, thus the general evolution of IoT technology would help and be helped by systems' optimization [46].

Taking all of the above into consideration, Kaloxylos et al [46], presented the architecture of a farming management system providing an operational example. The main aim is to give farmers a tool where they could have a total control of their crops in terms of monitoring. According to the writers, a complete farming management system is composed by 6 crucial elements, a cloud application, data management services, a service delivery framework, the enablement of IoT services, an interface to network and devices and a security layer. Their harmonious coexistence would lead to a complete and functional solution. The proposal separates the application into 2 main parts, the cloud FMS where the application exploits data kept in continuously updated repository, APIs that would assist its decision capability, such as weather APIs and advisory services (e.g. e-vet, e-agriculturist [46]), Big Data integrations and more, would create a set of relative actions to the connected WSN system, consisted of sensors, actuators and a local gateway. This is the second part of application and, according to writers', is a miniature of the cloud application in terms of computational resources and capabilities. Thus, except for taking directions from the interconnected cloud application, the system can work as a standalone when Internet connection is not available. Apparently, in this case, decision-making will not be in its best state since a Cloud application has much more resources to take into consideration and extract a more calibrated result. Concluding, this study aimed to introduce a way where farmers become a crucial part in an interconnected automated agricultural world.

As it has been mentioned before, energy consumption is one of the main concerns of agricultural IoT applications. Due to the nature of this sector, autonomy of the systems is essential, since it is very possible that human intervention will be difficult in a rural area. Thus, the deployed wireless sensor network should be able to operate in the most optimized way insuring its satisfactory lifespan. Currently batteries are the most widely used solution due to the low deployment cost and ease of use. On the other hand, moving to more eco-friendly solutions is an imperative need because of climate change. As a result, renewable resources of energy should be used over traditional power methods. A project that adopts the aforementioned approach is Khand's et al research [47] where they propose a system for water distribution using alternate energy resources.

Having in mind the prior, this project capitalizes on solar power in order to operate the WSN, which is composed by water sensors, a microcontroller which will forward data to a web application and actuators directed by the microcontroller which will switch on or off a water pump. At this point, it is important to point out that the proposed system's sensors will monitor water values continuously, because of the operation based on solar power. In any case of a conventional form of power, like batteries, system's lifetime would be dramatically reduced or the logic should be modified to a more complex yet energy optimizing approach. Regarding the logic of the system, the continuous monitoring of the sensors will provide data to microcontroller which will be uploaded to server and will be used in order to make a decision according to preset lower and upper thresholds. Additionally, user will be able to keep historical track of system's measured values while at the same time, except for automated watering he can manually trigger the system. Nevertheless, despite the security that is provided by a renewable source of power, the way this system works is lacking algorithmic and practical efficiency since a significant amount of data is sent to a server (possible overflow) through a gateway of low computational power. It is important to keep in mind that microcontrollers deployed in rural areas are devices of limited capabilities. Systems should be designed in a smart and efficient way in order make the best out of the available hardware.

Regarding the results of this research, writers ran simulations for four different type of plants and the thresholds used in the algorithm, in each case, were set in line with the average irrigation requirement of a specific geographical region. As cited, web application keeps track of water and energy consumption, thus farmers can, in the long-term, set their budgets correspondingly to the results of the provided data in order to organize scheduled watering and have a better picture of the economic resources they are going to need for the stability and viability of their crops.

The importance of integrating IoT applications into agricultural process should be a top priority in the years to come. After all, as the world population is rapidly rising, inevitably feeding needs will rise too. Since agriculture is the main pillar in food production, its management has to be improved [48]. As seen in the results of the analyzed researches, such systems optimize crop production in a multidimensional aspect since it has been observed that water and energy consumption are improved while at the same time farmers and agronomists get an accurate overview of real-time status of their fields. Though logic may seem similar across the applications, implementation differs according to the needs and circumstances prevailing in each project. Shreekantha et al [49] conducted an extensive research on agricultural IoT applications giving an epigraphic overview on each of them, focusing mainly on hardware and communication aspects such as the selected microcontrollers and protocols.

As it has been observed so far, the designed algorithms for these IoT applications are depended on a number of variables. Undoubtedly, soil moisture is the most pivotal one, regarding a plant's growth since it gets water and nutrition mainly through soil. Apparently, factors such as temperature, humidity, PH and luminosity conditions play a major role in agricultures' growth rate. Having sensors to monitor all these physical quantities and general condition of the area gives a significant advantage in optimizing and organizing farmers' and agronomists' choices when addressing their crops' processes. Having all this data, the amount of required water by an agriculture can be accurately determined, thus optimization is achieved in water consumption. Additionally, if a system works in an automated way, manual labor is reduced, if not eliminated, thus less effort and funding is required. On the other hand, actions taken should comply with plants current growth state. There is a high correlation between crop's watering needs and growth phase, a fact that most researches ignore [50]. Setting thresholds on soil moisture, for example, regarding switching on or off a watering system, despite the growth state, would, apparently, have better results than an empirical approach but there is margin for more optimization. Because of that observation, Halim et al proposed a system that takes the growth state of the plant into consideration in order to automate watering process while monitoring physical quantities as the ones mentioned above [50].

The proposed system is composed by a nodes deployed in a greenhouse measuring physical quantities. Specifically, each node is a microcontroller having a temperature sensor, a humidity sensor, a light sensor, a soil moisture sensor and a carbon dioxide sensor attached to it. A base station, the system's coordinator, using beacon enabled mode, collects the measured values by the nodes. Beacon enabled mode is a technique to synchronize slave nodes to execute coordinator's request [51]. Tree topology is used for the communication between base station and nodes via ZigBee (having been mentioned before). Wi-Fi technology is used for the upload of the data to a web server, where it is kept in a MySQL server. Taking into consideration the current growth state of the plant, thresholds are set for the measured physical quantities. Since the proposed system will operate in a greenhouse, where everything can be controlled, the coordinating microcontroller will determine the operating period of the mechanisms that will ensure that the optimized process will be followed. The aforementioned data are visually available through a mobile and web application where users can monitor in real time the measured values and general operation of the system.

Results have shown that communication between nodes and base station, for the testing application that was deployed, was extremely stable. Credits should be given to ZigBee protocol. As it has been mentioned previously, it works great especially in low distance conditions, while at the same it has low power requirements making it a strong candidate when it comes to selecting a communication protocol for a WSN. The application works well regarding data integrity and visualization. As future works writers point out system’s energy independence, aiming to integrate solar power in order to supply energy to the WSN. Additionally the modification of topology is a mentioned action, moving from tree to a multi-hop network while at the same time extension of the capabilities is another goal by adding a couple of extra sensors, in order to calculate even more accurately the values that will make plants’ growth even more optimized. Finally user’s integration is another target for the future, by adding automated communication processes via SMS, for example, in order to alert the user whenever an abnormality takes place.

It is important to bring up that although this application, on an algorithmic aspect, is fulfilling all the requirements crop’s growth has, selecting Wi-Fi as the communication protocol between base station and the external web is not a great choice due to Wi-Fi’s high energy consumption and the requirement of a ground phone line which may be hard to install in a rural area. 6LowPAN is a communication protocol for low power devices implementing the IPv6 stack interface, thus a system implementing it, can adapt to modern protocol approaches. Finally, as the system’s responsibilities extend, it is an ambiguous choice to have a microcontroller running all the algorithms. These machines are of low computational power, thus a large amount of data may cause a crash of the application. It is wiser to delegate calculations to a cloud application, where computational power is incomparably higher and usage of external APIs and Big Data extensions is easier and more accurate, and wait for a response with the directions that should be executed.

As it has been discussed, standardization in WSN applications is a critical factor regarding their expansion. Currently, applications work mainly as standalones using a custom format for the messages that will be exchanged in the context of the different parts of a system [53]. Thus, communication with an external system, probably of the same scope, is almost impossible. Setting a standard format to send data across such applications (e.g. XML, JSON, and HTML) would allow the cooperation among them and the enhancement of their functionality by using data that are not internally available for a system. Apparently, standardization will eventually help researchers and developers overcome the barrier of designing a standalone IoT application, something that narrows down capabilities. Additionally, it is much easier to share knowledge in an entirely accepted format than adjusting it to required needs.

Standardization is major player in having interoperable WSN applications, an approach that is supported by Sawant’s et al paper [53], a case study for an interoperable agro-meteorological platform, responsible to observe and analyze regarding the water requirements of a citrus crop. Based on initiatives like FOODIE, GeoCENS, Sensor Asia and more [53] writers’ goals are the design of a sensor platform, in the context of web enablement for sensors (SWE), integration of interoperability and scalability among heterogeneous systems, based on SWE standards and demonstration of the results. The system, called SenseTube, is composed by a single board computer, Raspberry Pi [54], which collects and disseminates data from the sensor nodes (temperature and humidity) distributed across the field. Data collection operates on a configurable polling technique. Wi-Fi is the selected communication protocol for the system and the microcontroller is connected with the data processing layer of the application either via Virtual Private Network on a fixed IP address or via costless cloud services such as Dropbox. The selected encoding for the communication in the system is SensorML [55], an XML expansion designed specifically for sensors, including the measured values, the sent data to the cloud service and the actions need to be taken by the actuators after the calculations.

Regarding the middleware of the system, which is a module of SWE web services for data query and access, an “Alerts and Notifications” system is triggered by the results of the running algorithm based on predefined thresholds. This layer periodically requests the collected data from the deployed WSNs, maintaining an independed connection with each one of them, in order to ensure stability of communication. “Alerts and Notifications” subsystem helps in reporting the observed values to user, through a preselected way of communication (email, social media post, push notification etc.). Finally a visualization format of the prior middleware has been implemented in order to grant valuable information to the user, while M2M communication is supported too. This web based thin client, is designed in a way that ignores the complexities of the business logic of the whole system. As for the deployed case study, the WSN was installed in a citrus field measuring temperature and soil moisture, as it has been mentioned before. The results have shown accuracy of the sensed values regarding the prevailing weather conditions and irrigation events that took place. This system’s main advantage is that it has been implemented in a way so it could facilitate other systems. Its collected data are presented and sent in standardized format, giving valuable information which could be handled by decision-making and Big Data applications, while on the other hand its architecture could be a template for other observing and farming management projects.

Another decisive concept in IoT ecosystem is calibration in measurements, which is critical for the success and the sustainability of IoT applications in the context of Precision Agriculture. Ensuring that algorithm’s input data is in the most optimized format, meaning that noise in measurements is eliminated according to the physical quantity, will finally give the best possible results, thus system’s performance will be in its best shape. Soil moisture measured by a sensor is calibrated too in Mat’s et al proposal [56]. Epigraphically, the writers propose a greenhouse management system where quantities such as moisture and temperature are measured. This data act as input for an algorithm which decides if it will trigger or not different functions such as air circulation, watering etc. in order to improve the conditions that prevail inside a greenhouse. A test application was deployed and a comparison with a typical watering scheduled program is conducted in order to prove system’s success.

Application’s core logic is, as described in almost any previous case, based on predefined thresholds, by an agronomist [56], which trigger the respective system. Sensors communicate with a base station in order to send the collected data. In this system the base station is responsible to disseminate data to an external service where calculations for system’s irrigation will run, while at the same time data are available to users via a web application. Sensors and base station communication is implemented through XBee protocol, an extension of ZigBee which’s advantages has been analyzed previously [41], while on the other hand base station communicates with the Internet via Wi-Fi protocol using, specifically, the ESP8266 Wi-Fi module.

The most important part of this paper is the conducted analysis on how to calibrate measured values from a soil moisture sensor. A dedicated Volumetric Water Content (VWC) soil preparation method was used in order to calibrate moisture. Firstly, sensors read values from 3 different type of liquids in order to set 3 base levels. Namely, glycol, acetone and silicon oil were used, 3 liquids with different dielectric permittivity referred from the highest to the lowest one, respectively. This process is followed by correcting the value of the measured soil moisture. Specifically, a volume of soil is baked in order to eliminate any humidity, compressed in order to remove air and finally a predefined amount of water is injected in it. The ratio of water to soil is the VWC value. In this project VWC is calculated as the average ration of the aforementioned 6 different states.

Finally in order to calibrate and transform the raw measurements into VWC values, two more measurements of the soil moisture sensors take place. The first one reads the moisture level in plain air and the second one in plain water. This process gives a linear formula which is used in order to turn raw numbers into VWC.

The results of this study showed the comparison between the proposed system and a schedule based, on predefined intervals, watering approach. It was proved that research's outcome was spending less water than the traditional method since it saved 1.5 liter per tree per day, which is an outstanding result. Additionally, water was used in a far more optimized way, since the lowest threshold of 35 VWC, set by an agronomist for a chili crop, was never exceeded in contrast to the traditional method. In the second system, the one working on scheduled timetable, it was observed that watering would take place, regardless the fact that the crop had sufficient water funds at the moment of irrigation while in another case watering began after surpassing the lowest set threshold, proving the superiority of the proposed application.

In every presented research so far, soil moisture is measured via conductivity, having a sensor attached in the soil and taking periodical measurements. Especially in the latter one, a detailed description of how these values are calibrated in order to have more accurate results, thus an improved system, shows the importance of measurements in every IoT application. After all, this is the input data for every system, regardless its direction, and data's integrity and accuracy are pivotal. Additionally, the way these values are collected play a significant role in system's autonomy, scalability and stability. Sensors deployed in a field suffer from a lot of practical issues. Lifespan is strongly connected to the source of power and currently renewable sources are not the most popular choice. Weather conditions may critically affect their functionality too. For example, a storm or a hailstorm could significantly damage the hardware while at the same time extreme temperatures may cause a malfunction or a system's general failure. To writer's knowledge and personal experience electrical devices may not work properly in such cases. Still sensors are a go-to option in order to measure values but gradually these issues should be either resolved or other techniques should be introduced in order to measure the required physical quantities.

Such an approach is adopted in Mulla's research [3], where he reports epigraphically all the progress that has been achieved in sensors the past 25 years, mainly alluding to the advances and knowledge gaps that still exist. Main field of this research is monitoring, specifically using evolved techniques such as satellites, hyperspectral sensing etc. As said in other researches, and it must be mentioned that is the obvious path that must be followed, in the future IoT applications should run on renewable energy in order to be even more environmentally friendly. Additionally, manual labor should be reduced until elimination. Mulla suggests that unmanned vehicles or robots should take the responsibility of taking measurements in WSNs, where finally all this data should be dispatched to cloud services in order to take advantage of limitless computing power.

Remote sensing is the main pillar of this research. Mulla mentions 3 different techniques. Satellite monitoring has been initially used for remote sensing imagery in agriculture in the 70s having a significant evolution on the operations that were executed. Currently, its capabilities vary from identifying whether a crop is affected by a disease to developing growth patterns for a crop or finally measuring developed indices regarding soil conditions for crop's evolution. An obvious barrier for this technique, is the high cost that comes when using satellites while at the same time data may need a lot of calibration. The second technique is proximal remote sensing, meaning that sensors would be attached on other machines such as tractors, spreaders and sprayers. Through this way, for example, chlorophyll of the plants can be measured in an optimized way and its value can give useful conclusions regarding physical quantities that define crop's status, as in terms of health. Finally, hyperspectral sensing collects reflectance data over a wide spectral range, providing powerful insight on plants current condition. This technique goes back to the 80s and has evolved over the years. Apparently, the transmission of this sensed values requires an efficient way to be accomplished due to the overwhelming amount of data that will be sent. Stergiou et al [70] propose such algorithms which overcome this issue through the usage of IoT and cloud networking which can be achieved through the subnets that each IoT system creates.

Concluding, Mulla defined some improvements that could transpire for remote sensing in Precision Agriculture. Chemometric or spectral decomposition methods should be in the front line while on the other hand sensors should take responsibilities of direct estimation in a context of nutrient deficiencies, for example. Indices' development in the context of spectral and hyperspectral methods should continue since it gives multiple advantages such as assessing crop characteristics and stresses. Finally, real time satellite measurements should be enriched with historical data kept on them, thus systems could end up to the best possible decisions.

Water usage optimization is not the only factor which affects agriculture's output. As described in many aforementioned studies, an IoT application could take care of this concern in a monitoring and decision-making context. Development of IoT applications in the context of Precision Agriculture can improve processes in multiple ways, having a general positive impact. For example except for water losses, crops also suffer from insects and diseases. If these threats are not dealt immediately production may be in danger. A monitoring WSN application could alert its user directly after the identification of the trouble, thus taking precautions and resolving these issues could be done as quickly as possible. These are typical cases where IoT solutions could assist farmers and agronomists in way that their intervention and labor would be the least possible. Some other factors for crops' production decline are untimely harvesting, misuse of fertilizers, mishandling of ripened crops and lack of machineries [57]. All of these cases could possibly be dealt by WSN applications by means of remote sensing, automated methods and better analysis of the current state of the plants. Such a solution is proposed in the research of Shahzadi et al [57], where the designed system can be used in order to resolve the prior referred problems handing out a multiform approach in IoT applications.

An expert system, as the one that is proposed [57], is described as an artificial intelligence application, where it solves problems as a human would try to do. A lot of expert systems have been designed in the context of agriculture like CUPTEXT [58], LIMEX [59], CALEX [60] and many more. In this case, a set of sensors is deployed in fields in order to collect data such as humidity, soil moisture and leaf wetness. Each set is attached on a microcontroller which communicates with a base station. The selected communication protocol is XBee which has been described previously. Due to the fact this was an experimental approach in a controlled environment base station is emitting data to a PC by being attached on it via USB. Apparently, this solution would not work in real conditions. Researches aim, though, is to show the optimization such a system would offer. Obviously, although communication is major part in a WSN system, gateway's connection with the Internet is minor compared to the design of a full approach as the on presented here.

The algorithm of the system has 5 main pillars, a knowledge base, an inference engine agenda, a working memory, an explanation facility and a user interface. Knowledge base is predefined (e.g. set thresholds) and was shaped by experts' opinion, research of relative bibliography and input coming from the sensors' measurements. The output is the result of a simple if-else logic. Combining the above 5 pillars the system can extract a conclusion and send a recommendation to the farmer about his crops. Delegation of recommendation is sent to an accompanying Android application.

The results of the conducted experiments showed that the system could successfully identify cases where crop was being attacked by weeds and pests, disease diagnosis while it could predict if it was going to suffer from one according to sensors' measurements. Recommendation includes a plan for irrigations, in terms of quantity and start times, in agreement with the input data. Compared to expert systems of previous eras, this approach takes full advantage of the IoT concept, thus manual labor is significantly reduced. Despite the disadvantage of passing data from WSN's base station to a PC via USB, this is a diverse system supplying solutions in different problems while at the same time it makes use of XBee protocol which, as it has been mentioned, is one of the best choices for the communication between nodes in IoT applications. As future improvements writers suggest the completion of system by introducing actuators and cameras in order to automate processes and improve the quality and diversity of input data respectively.

All the presented decision-making systems had algorithms running on what is called Boolean logic. Specifically, whenever the input values were, for example, below or above a preset threshold algorithm should take actions according to this fact. It doesn't matter how close to that threshold this data may be. The fact that it is measured below or above that value is sufficient itself. Apparently, this is not a very delicate and adaptive logic. For example, in terms of common sense, a case where a value is below, but still very close to a threshold is different from a case where its measurement is two or three times less that threshold and it would be wiser if each circumstance was handled in a specialized way. Fuzzy logic algorithms are introducing this kind of approaches. Epigraphically, in Fuzzy logic variables' value may extend between 0 and 1 in contrast to Boolean logic variables, which's value can only be true or false (0 or 1). Such algorithms are highly applicable to any WSN application and especially the decision-making ones, since decided actions can be more relevant to the current state of the measured physical quantities.

Azaza et al adopted and implemented such a fuzzy logic algorithm in order to control key climate factors in a greenhouse aiming to a smart automation system where energy and water usage optimization are accomplished [61]. According to the writers, major factors for the prevailing climate in a greenhouse are temperature, humidity, CO2 and illuminance conditions. Sensing these values is resulting to extracting the climate condition, thus decide according to the benefit of the crop. In this research sensing functionality is based on fuzzy logic for all 4 factors. Specifically, a specialized formula is applied for temperature and humidity, where input data are sensed valued of the aforementioned physical quantities categorized in five different sets (Negative Big, Negative Medium, Zero, Positive Medium, Positive Big) and the output is categorized as very low, low, medium, high and very high for both of them. On the other hand for CO2 input data are categorized as sets of three (Negative, Zero, Positive) while the output is can belong to one of the four following categories (Zero, Low flow, Medium flow, High flow). Finally illuminance's logic is the most complex one since input data are 3 different categories, where each one is divided in subsets (Illumination Error: {Negative, Zero, Positive}; Solar radiation: {Cloudy, Middle, Clear}; Outside temperature: {Warm, Medium, Cold}) and output could be a member of one of two divided sets (screen: {Open, half open, closed}; lighting: {On, Off}). Additionally, measured values are calibrated in each sensor in order to have even more optimized results. Finally, as it is expected, according to the generated output, algorithm makes a general decision on the actions need to be taken.

Regarding the communication pillar of the system it follows a typical logic where sensing nodes communicate with a base station through ZigBee protocol. Base station is responsible for controlling these major climate factors and transmitting the sensed values to another station, which logs this data into a database in order to conduct an efficiency and performance analysis. Data is sent to this second station by implementing the GSM protocol. Thus, as in most cases, we have a system where multiple techniques are being used in order to satisfy the need of communication. Functionality of the system was compared to the greenhouse's crop performance when system was not deployed. As it has been observed, factors were stabilized when using the system to values which affected positively the growing crop. As a result, plants' growth was optimized compared to the previous period. Additionally, system's utilization helped in saving energy and water. Due to system's operation 22% less energy was used in order to cover the crop's needs while at the same time, 33% water usage reduction was achieved. This research proves that not only optimization in water and electricity consumption can be accomplished through IoT applications but crops can furtherly grow too.

Throughout this presentation of the relevant literature it is clear, that as years passed by and evolution of technology has a strong impact on the architecture and implementation of WSNs. For example, in a context of communication protocols Bluetooth was the top option in the previous decade. Despite the fact that it can cover a small distance and is an efficient protocol, in terms of energy consumption, its undoubted advantages make it a handy option. Such an approach is proposed in the research done by Kim et al [62], where a Boolean logic algorithm is implemented and communication is based on Bluetooth. This protocol is still a choice, but it can probably facilitate only household gardens or small greenhouses, due to its limitations. As we have seen so far [29] [30] [41] [51] [56] [61] [63], in most cases ZigBee protocol and its extensions (e.g. Xbee) are the current trend, since it is a protocol designed for low power devices. Although, other options may be able to cover greater distances or they have the advantage of implementing the IP stack, like 6LowPAN where communication with external resources like a Cloud service can be executed straight from a sensor node, they are still on an ongoing status regarding their stability and adoption from the technological society.

Another research where ZigBee protocol is utilized is the one implemented by Zulkifli et al [63]. They developed a Wireless Mesh Sensor Network (WMSN) accompanied by an active Radio Frequency Identification (RDIF) system which monitors soil moisture and temperature in a field. Additionally, an automated irrigation functionality is implemented. Analyzing this approach, RFID gives a main advantage to this system, since it eliminates human intervention and turns H2H (Human-to-Human) and H2M (Human-to-Machine) architectures to pure M2M (Machine-to-Machine) implementations. Through its practice, each sensor can be traced, in terms of position, since it is assigned with a tag with a unique identification number. Additionally, all measured data is saved on it, thus it takes the role of the microcontroller, which were applied in other researches. Tags work in 2 forms, active and passive and in this case active tags send an ID to the passive ones in order to identify transmitter and received data's sender. In more technical details, active RFIDs operate in higher frequencies thus they have a shorter lifespan.

ZigBee is the protocol used in this research for the communication among the nodes. As it has been mentioned previously, it implements IEEE 802.15.4 standard defining network and application layers while supporting profile and security mechanisms. Its main advantage is the low energy consumption, thus the system that implements it, has an expanded lifespan. As said in the prior paragraph, WSN's communication topology is defined as mesh. ZigBee also supports star and cluster-tree topologies, but in this case the mesh one was selected so data can be easily transmitted from node to node until reaching to the base station. A moisture sensor and a sprinkler will be attached on each node. According to the measured values of the moisture sensor the required amount of water will be sprinkled as result. All decisions taken are node based, which serves on system's autonomy although it limits its capabilities since an RFID is not able to make complicated calculations. What should be pointed out, is that sensors, stay in sleeep, for energy saving reasons, when in standby mode. When a significant change in moisture is sensed, node wakes up in order to send data to the base station.

Concluding system's architecture base station sends data to a web based application where it is analyzed and processed. This information can be accessed by users using a mobile phone or computer in order to be able to monitor in real time the current status of their crop. Additionally, users can see a comparing graph of the water consumption in their fields when using the proposed system against to a case where no automation is applied. The system was deployed to a 10 acres area, where 20 nodes where required in order to cover it, having the system running for 5 months and the results have shown almost 50% improved water usage, in terms of consumption, against an empirical approach.

2.8 Conclusions from Literature Review

What is common across all the aforementioned studies, no matter the logic of the selected algorithm, which platform their sensors' are based on, which communication protocol was selected or the extensibility of the designed system, is that they try to collect information. Every process, despite the sector it belongs to, is implemented as perfect as the amount and quality of information is given to it. Empirical approaches may produce results but undoubtedly they are not the best possible. Exploitation of information in order to extract patterns in crop's reaction to certain circumstances, analyzing growth conditions and the attempt to improve them, will derive improved production. This can be interpreted in many ways, since enhancing agriculture affects positively the quality of the produced nutriments, the water consumption, since it is optimized, which probably means that less water is going to be used in contrast to an empirical approach. Apart from that, on an economical aspect decreased water usage means more economical funds can be saved. Additionally, actuators will work for just the right amount of time thus, energy consumption will be decreased. Last but not least, the reduction of human intervention, in terms of manual labor, is a critical goal towards social improvement.

On a more technical aspect, we can sum up to the fact that most of the presented algorithms are under the same core logic. A WSN system senses some crucial factors about the conditions of the environment it is deployed. These values are handled in multiple ways according to the goal of the system. They are either used for presentation, or as input to a decision-making algorithm. Additionally, they could work for pattern extraction in the Big Data field. As we have seen, calibration according to application's requirements is a major factor in revising and correcting the expected results. On the other hand, energy consumption is a concern that has to be dealt. Some of the variables that affect it, are the selected communication protocol, the way a sensor node collects data and the amount of responsibilities each sensor has.

External parts of the presented systems such as web based applications, which either make calculations followed by the dissemination of instructions back to the WSN or they just present the sent data after some kind of processing, give an extension to the capabilities of an agricultural IoT project. Apparently, utilizing decision-making applications is pivotal, since they can easily exploit a number of sources of information, blend it with the sent data, resulting to really complex and smart automated systems. To writer's opinion, automation of the whole monitoring and irrigation system should be the main goal of the most agricultural IoT applications. Variously, when it comes to approaches, responsible to present data, demonstration should be as enriched as possible providing comparisons in order to reveal advantages of the deployment of WSNs in crops. Another path which could be investigated and expanded is the one where applications which collect such sensed data could work as input to other ones. In terms of software engineering, this information could be exploited through an API thus anyone interested in developing an agricultural application, whether it is about IoT or any other technological subfield, could have a source of available data. Finally, communication between WSNs and such applications is still a matter under investigation. Current solutions such as Wi-Fi or cellular approaches definitely cover applications' needs in terms of stability, security and latency. But other limitations, such as reachability, since WSNs will be probably deployed in a rural area, or energy consumption are significant drawbacks that have to be overcome so such solutions will be adopted. On the other hand, protocols that are designed for low power devices have barriers such as latency in data transfer, area of coverage is usually small and the most important thing is that there is not a standardized format for data, thus communication with external applications is difficult. Additionally, in cases where IPv6 protocol is implemented (e.g. 6LowPAN) projects' status are still under question and there is a lot of uncertainty in their practice.

In conclusion, after studying and presenting the main parts of the aforementioned researches, valuable information is gained in order to design an optimized automated irrigation system based on sensed values. Considering that such a system could be deployed in a greenhouse or a field we can recognize similar cases and choose the most suitable communication protocol, number of required nodes, sensors needed in order to track physical quantities which are going to work as input to the designed algorithm. Additionally, energy consumption matters that have been analyzed in IoT applications, regardless the sector they belong too, will be taken into consideration in order to design the architecture and functionality of the proposed system.

3. System Design and Architecture

3.1 Introduction of the proposed system

IoT applications provide a stable way towards improvement in every sector they are practiced on. They can offer real-time, rich information which can be exploited in a variety of ways regarding either presentation of it or take some actions according to it. Henceforth, an IoT system is able to be sufficient against the requirements an automated system would have. Such a system is the one that is proposed in this thesis and it regards automated irrigation application for watering in a field based on a wireless sensor network. Each network's node will be responsible for, regularly, measuring soil's moisture and forward these measurements to WSN's base station. All this data, will be forwarded to an interconnected cloud application and will be used as input, among others, for an algorithm which will calculate if the watering system should be triggered and the duration that will be switched on in order to satisfy crop's watering requirements. The output will be sent back to the base station which will be responsible to take the respective actions. The whole system will be complemented by an Android application, from which a user can set inputs for the aforementioned algorithm, review the collected data and the actions taken, study comparing graphs showing the efficiency of the system and finally take decisions putting aside the algorithm's output. In this chapter, each system's component will be analyzed on an architectural level, the communication among them will be inspected besides the energy requirements that this operation would have while simulations of it will be presented, thus a general overview of the system will be described throughout the presentation of them.

3.2 Flows

We can divide the proposed IoT application into 2 different major entities, whose flows' overlapping of them is not negated, the first one is, practically, the algorithm's operation, running on an automated mode, and the second one is the user, where intervention on the first case takes place. In this part, a detailed analysis of each possible scenario might occur from these 2 pillars will be presented in order to cover every use case of the system.

3.2.1 Automated mode

In Figure 3.1 the operational possibilities that can take place when the system runs on an automated mode are thoroughly described. Initially, and only one time in the lifespan of the system, user is called to insert some data regarding the type of field and the region it is located. This data are required inputs for the rest functionality of the algorithm, which will be extensively analyzed in the following parts. Thenceforth, measured data are taken into consideration along with input for the anticipated weather conditions in the region and the required watering duration comes as a result. If that value is greater than zero it is sent back to the WSN and the base station turns on the respective water pumps for the aforementioned duration. Otherwise, no order is sent and finally the system awaits for the next data collection. Apparently, system's functionality is pretty autonomous in terms of
human intervention.

Except for the initial input, which of course is pivotal considering that it affects 2 major functionalities, its operation is affected by variables that are strongly independed from any human intervention.
This functionality is designed to loop indefinitely for as long as energy is supplied to the WSN that will be deployed in the field and it will collect soil's moisture levels in regular intervals. These intervals can be, empirically, set according to region. For example, in a country like Greece where weather conditions are mostly stable an

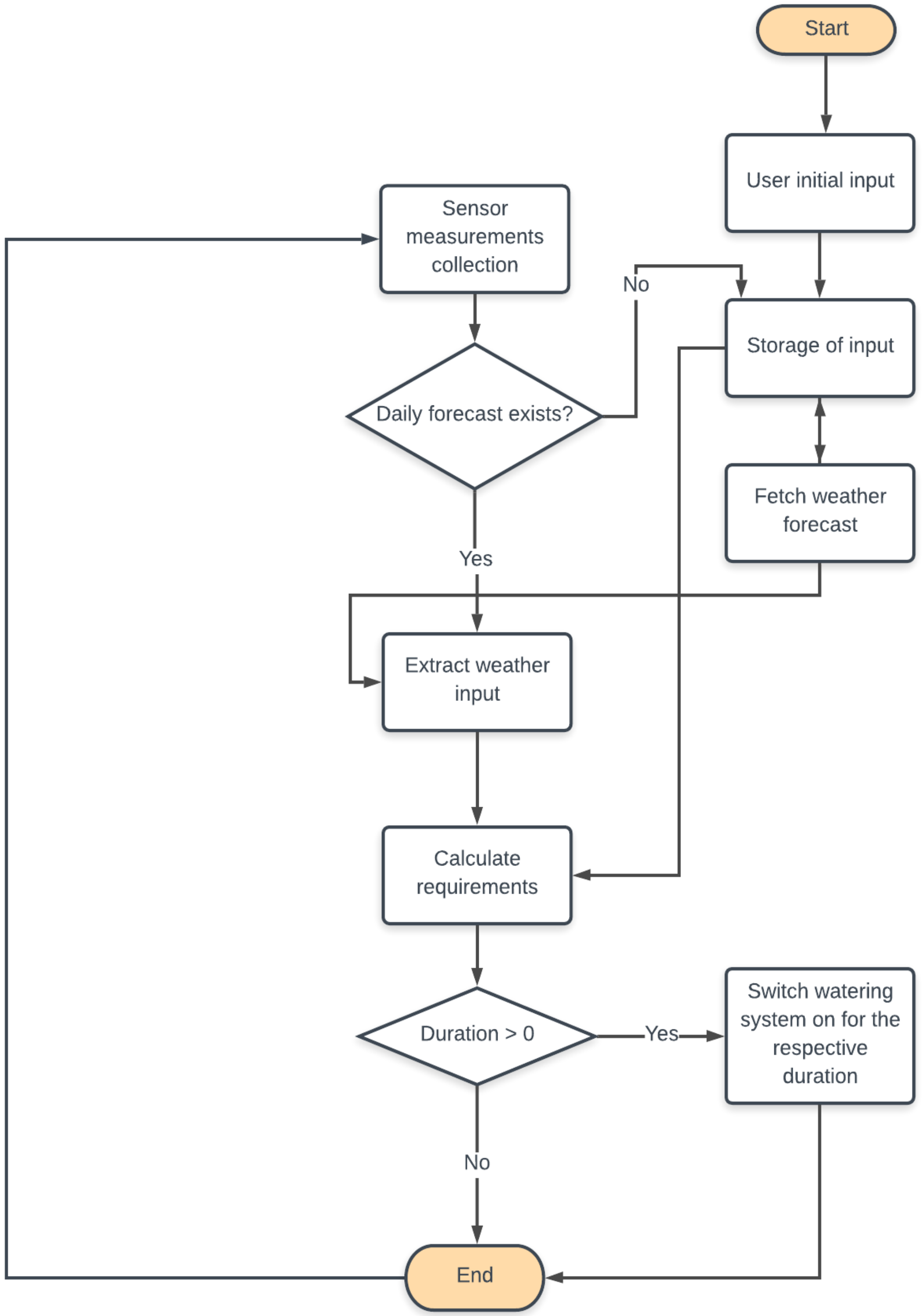


Figure 3.1: Algorithmic operation when system runs on automated mode

interval of two hours seems like a reasonable value. If this system was going to be deployed in a country, which's climate is tropical, this interval should modified accordingly. Likewise, moisture thresholds will be set according to the type of crop that is growing in the respective field. For instance, olive trees are tolerant to small amounts of water where crops like rice, cotton and wheat require great quantities for their growth. These two parameters are affected by user's input, so once they are set they will be used in every loop of the functionality. Subsequently, the relevant weather forecast is affected by the defined region and it will be fetched once a day from a selected API, therefore system's autonomy will be preserved. Finally all of that will be taken into consideration, put in the respective formula in order to investigate if the moisture level is below the set threshold. If this is true and no rain, or generally conditions that favor the creation of humid circumstances, is awaited, a value that will indicate the required watering time in order to cover crop's needs will be sent back to the base station.

Summarizing this first part, system's main algorithmic flow is a strongly independed entity which can bring automation to the watering process for a crop. Except for the

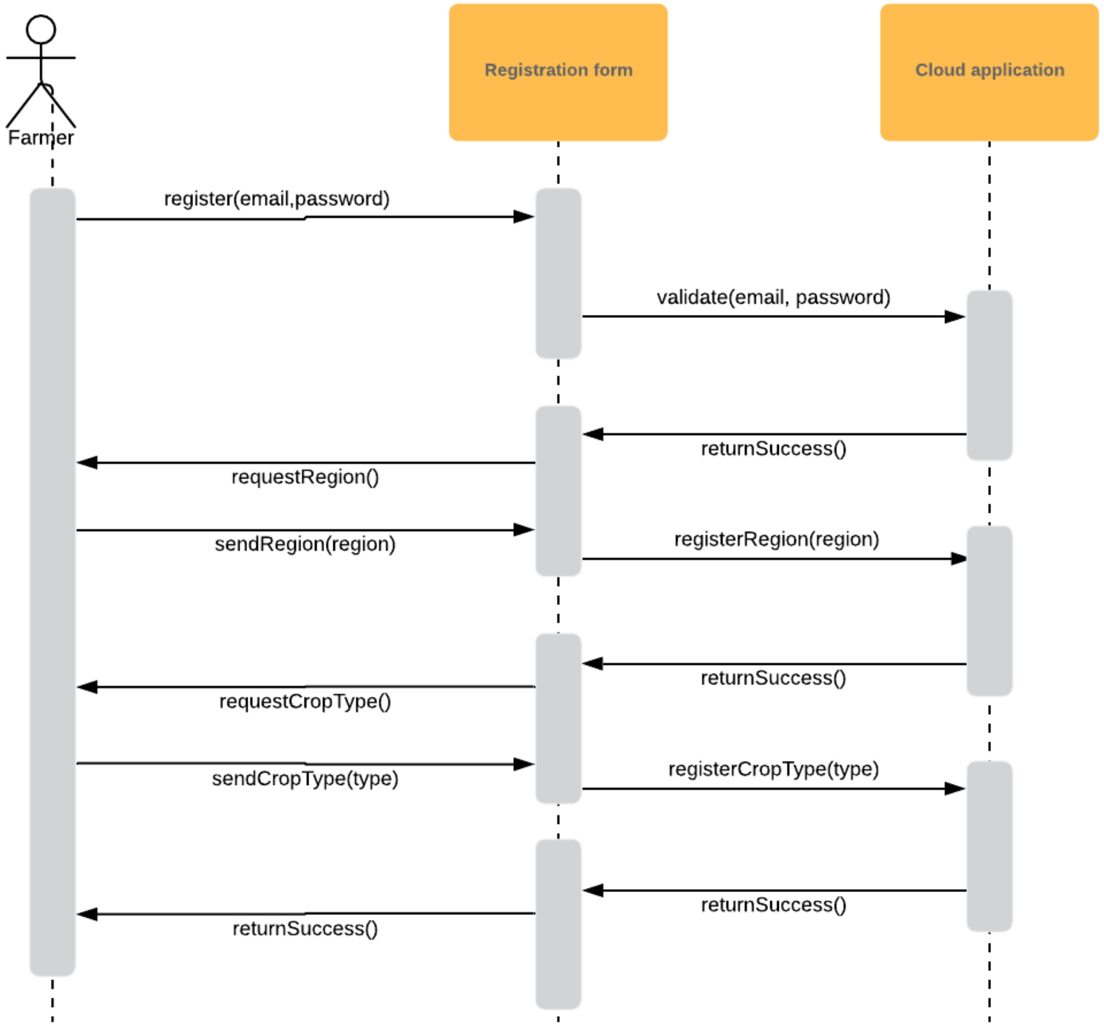


Figure 3.2: User registration flow

initial data, set by user, it is fully autonomous, taking into consideration a number of parameters in order to implement the designed logic. The whole process is designed to be repeated in a polling logic in order to collect WSN's measurements so as to produce an output. The possibilities of enchantment are numerous. Adding machine learning algorithms, extraction of knowledge via Big Data patterns, improvement of the efficiency of the current flow are only a few of the possible enhancements that could be applied to the automated entity of the system in order to enrich its capabilities.

3.2.2 User flows

Except for the automated mode, this system is designed to offer real-time information to the interested farmer, as well as the option to intervene in the automated functionality. Furthermore, user sets the initial data for the system which is pivotal in order to set value to vital variables of the algorithm, which decides if and what actions should be taken. Although, this thesis' goal is that the proposed system would reduce manual labor it is apparent that user's input as well as letting him monitor and control

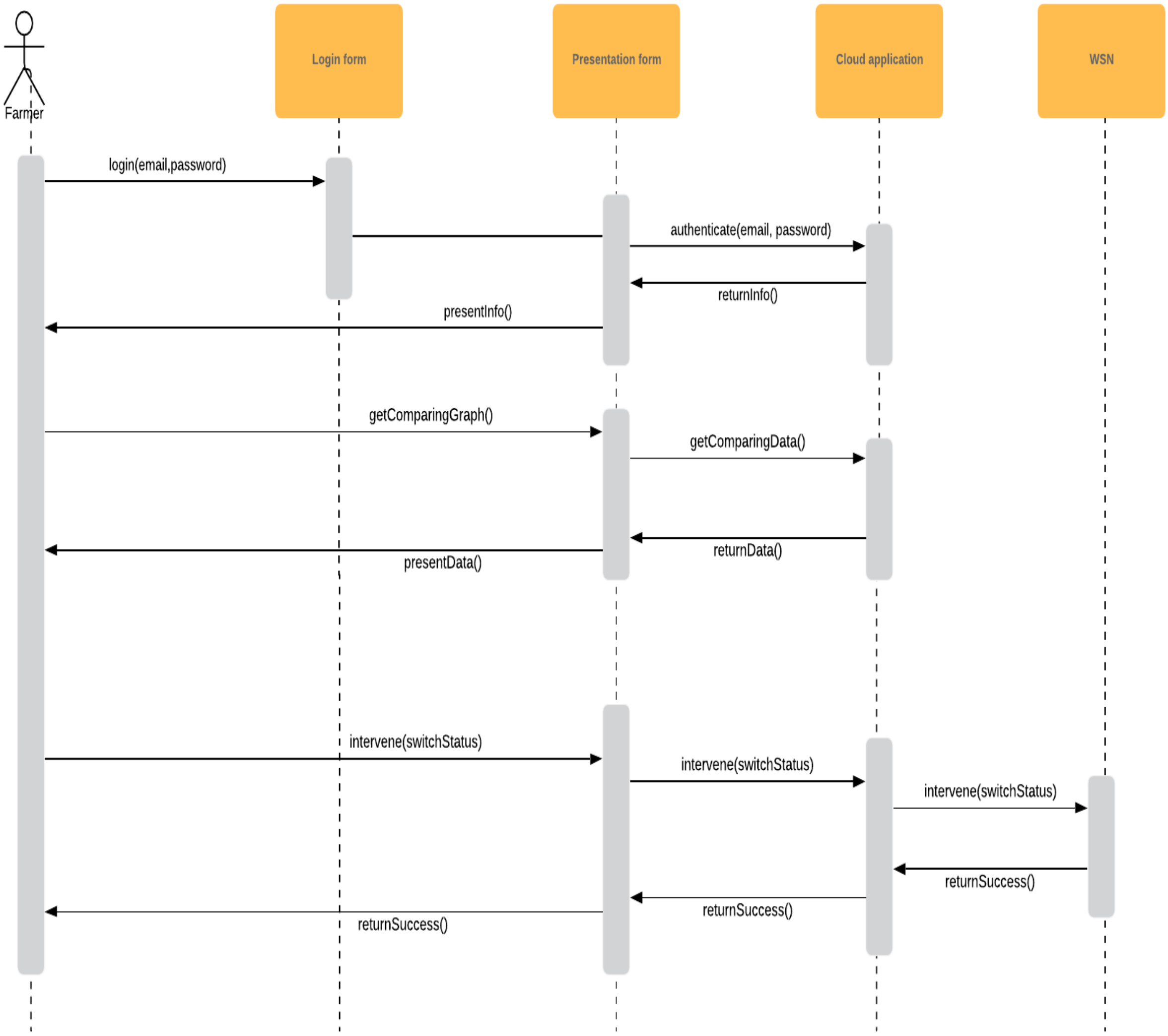


Figure 3.3: User main actions flow

the operation is quite fair. As said previously, user can interact with the system through an Android mobile application which will communicate with the cloud application through a RESTful API in order to satisfy his requests. In the following figures, the use cases of human interaction with the system will be presented and analyzed.

Initially, user is required to register in the through the mobile application. This happens in order to ensure data integrity, since a user would be able to monitor and control only one field. In this proposal, we assume that a user is connected with only one field in terms of entity relationships. Apparently, in real conditions a farmer can own more than one fields, which can be in different locations. But for the sake of ease, we make this assumption. From a software development aspect, system's scaling in order to support the connection of the farmer with more fields, is quite interesting. Some techniques that could resolve this challenge is the introduction of logging in through base station's microcontroller or taking advantage of the cellular network. Regarding the latter technique, if the base station communicates with the cloud application through mobile data a sim card would be required, which can, apparently, be uniquely identified through its mobile phone number. Since this number is known to the user, he can set it during his registration, so the connection between a farmer and a field will be easily achieved.

The registration process is described in Figure 3.2. User sets an email address, which will uniquely identify him for every flow that requires authentication, and a password. These fields are going through a validation (e.g. email uniqueness, strong password etc.). If this step is passed successfully, user is required to set the region his field belongs to. This could happen either automatically by taking advantage of the installed GPS sensor any Android smartphone disposes or user can manually insert the necessary information. Subsequently user has to set one last parameter which is the type of crop that is growing in this specific field, through a rich dropdown list which will be available to him. These two parameters are required for the flawless functionality of the algorithm. As it has been mentioned, the algorithm takes into consideration the weather forecast in order to extract a result. Apparently, only by correctly defining it the algorithm can have accurate results. Regarding the second one, which is the type of crop, each plant has different watering needs. Setting a moisture threshold for the algorithm is strongly connected to that fact. Finally, a success message is shown to the user in order to inform him that his registration was completed. To conclude, the process of registration is pivotal in order to set the necessary data which will ensure algorithm's correct functionality.

Obviously, there is a lot of margin for improvement and automation of the registration flow. It has been mentioned previously that user's field could be more than one in reality, so the whole process of supplying a region and a crop type could be repeated for as many times as the fields of the farmer are. Furthermore, each geographic area has certain conditions favor the growth of specific type of crops, which can be identified even empirically. So, in order to facilitate user's experience, by setting a region, the system could present only the aforementioned type of crops instead of showing a complete list of the available choices stored on an online database.

Logging in to the application, gives a full access to the capabilities of the system regarding the user. Monitor and control are the 2 generic functionalities that are supported in this case but both of them offer an enhanced experience on an informational and action-related aspect. Valuable data are supplied in order to give a detailed overview of field's status, either regarding its current condition, in regards to moisture levels, next scheduled watering and daily consumption, or a comparative approach, where graphs would show system's dominance against traditional techniques. At the same time, total control could be taken with reference to switching the system on and off, ignoring automated functionality and adjusting to farmer's personal will. In Figure 3.3, the aforementioned functionalities are presented in a sequential diagram in order to get a better understanding of the functionalities introduced in this paragraph.

In Figure 3.3, the main functionality, related to the farmer, of the application is described. In this diagram, we can see, what is called in software engineering, the happy path of the available options to the user. After successfully logging in, a back end service returns data relevant to the current condition of the field. As it has been mentioned, the WSN sensors measure the moisture levels and regularly forward them to the cloud application both as algorithm's input and as presentational data. In the latter case, this information is sent to the mobile application through a RESTful request, and the front-end is able to present the last moisture measurement, the estimated remaining time till the next watering and the moisture measurements that took place the day of request as well as the water that has been consumed until that moment.

Another functionality in this use case, is the presentation of comparing graphs between using this automated watering system against using traditional techniques. Cloud application has all the available data to calculate how much water has been consumed for the crop's needs and could even measure the respective consumed amount of energy in order to operate the WSN, regarding its functionality and the communication among the subsystems that conduct it. On the other hand traditional watering schedule, since it is based on empirical data, has a generally stable and predefined water and energy consumption, thus the can be easily calculated. Furthermore, the economic consequences of both statuses could be extracted and presented to the user. As a result, all this information would be presented to the user giving an overview on how the whole crop's growth process has been improved due to the usage of this application.

Finally, the last functionality that is offered to the user through the mobile application is actually the one that is the most interesting. In details, if the farmer, for any reason, wants to ignore the automated schedule that the cloud application implements, he can immediately change the operating status of the system. Apparently, this functionality is totally against the nature of the application which aims to offer optimization through optimization and reduction of human labor for the watering process. On the other hand though, firstly and most importantly a farmer may be able to have total control over his field. Additionally, even in this case he is one click away from switching system's status and he doesn't have to move to his field in order to do that. Thus, improvement to his experience can be achieved even under these circumstances.

In conclusion, all the possible functional scenarios have been presented in this part. We can see that the proposed system, despite the seeming simplicity of it, it offers a number of capabilities, split in automated ones and manual ones some of which are relative to the actions taken from the system or the user while others are useful for presenting the outcome of its usage. Of course, ideally the user would use his mobile application only to get the overview of his field condition, showing trust to the automated operation and its dominance against empirical approaches. Eventually, in either way, a farmer can have an optimized process to monitor and control his crop's growth by taking advantage of an IoT application. In the following chapters, all of the components, including the WSN, would be thoroughly analyzed, so the reader may get a clear and overall picture of system's architecture and functionality.

3.3 Analysis of Components

In this thesis an IoT application for automatic watering is proposed. It is a multi-level system, composed by a number of components. In details, a wireless sensor network is deployed in a field, where each node is will be responsible for regularly measuring soil's moisture levels and forwarding them to a base station. Subsequently, these measurements are uploaded to a cloud application, handled as input to an algorithm which takes into consideration a number of other parameters in order to extract a result whether the watering irrigation system should be switched on or off. This result is sent back to the base station which acts accordingly. A complementing mobile application assists in order to help the farmer monitor the current condition of his field while at the same time offers him total control over the watering system letting him switch it on or off at his will.

All the above parts interact with each other with flows that have been described previously. In this chapter we will take a closer look to each components, analyze its structure, either on hardware or software aspect regarding the case, explain its usage in the system and inspect its connection with the other components. Keeping in mind the valuable information and knowledge that came as an output from the presentation and analysis of the relevant literature, we will try to pick the best technological option for each component. In the end of this chapter a full description of every subsystem of the application will have been offered, thus a clear overview over the architecture and design of the system will be delivered while at the same time there will be an investigation on how the connection among them should be implemented.

3.3.1 Moisture sensors

The analysis will begin from the wireless sensor network that will be deployed to the field. Undeniably, moisture sensors provide the most important data for the successful

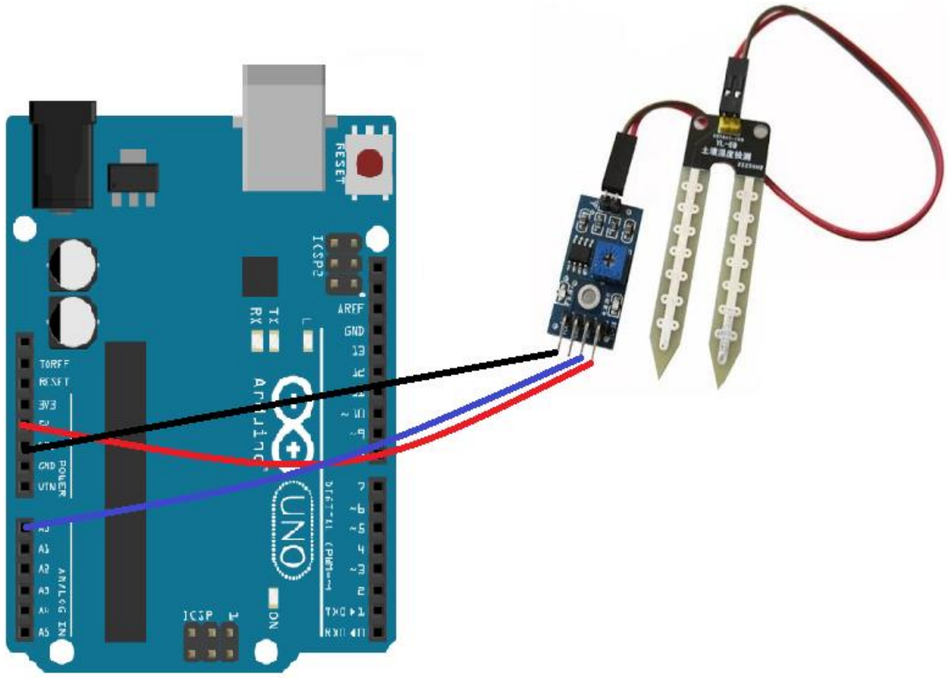


Figure 3.4: A moisture sensor connected on an Arduino board

operation of the algorithm. As it has been mentioned, sensors will regularly take measurements of soil's moisture in order to forward them to the cloud application to handle them accordingly. This is their only responsibility but its meaning is apparent and significant. These values not only act as input to system's main functionality but they are also used as infographic data so as to present current field's condition besides working as proof of superiority of this IoT application against to traditional watering techniques.

From a hardware aspect, in order to have an operational moisture sensor it should be connected onto a microcontroller board, like an Arduino or a Raspberry Pi. These boards have inputs to which the sensor is connected either directly or through a cable extension. Figure 3.4 explicitly describes how this connection looks like. Through interfaces, such a sensor could be connected with a wide variety of devices such as computers and smartphones. As it is obvious the sensor is powered by the board through their connection. The measurements is implemented with indirect ways, as it has been said previously [37] [56]. More specifically, due to the dependence on a number of variables, moisture can be indicated through the electrical resistance, dielectric constant

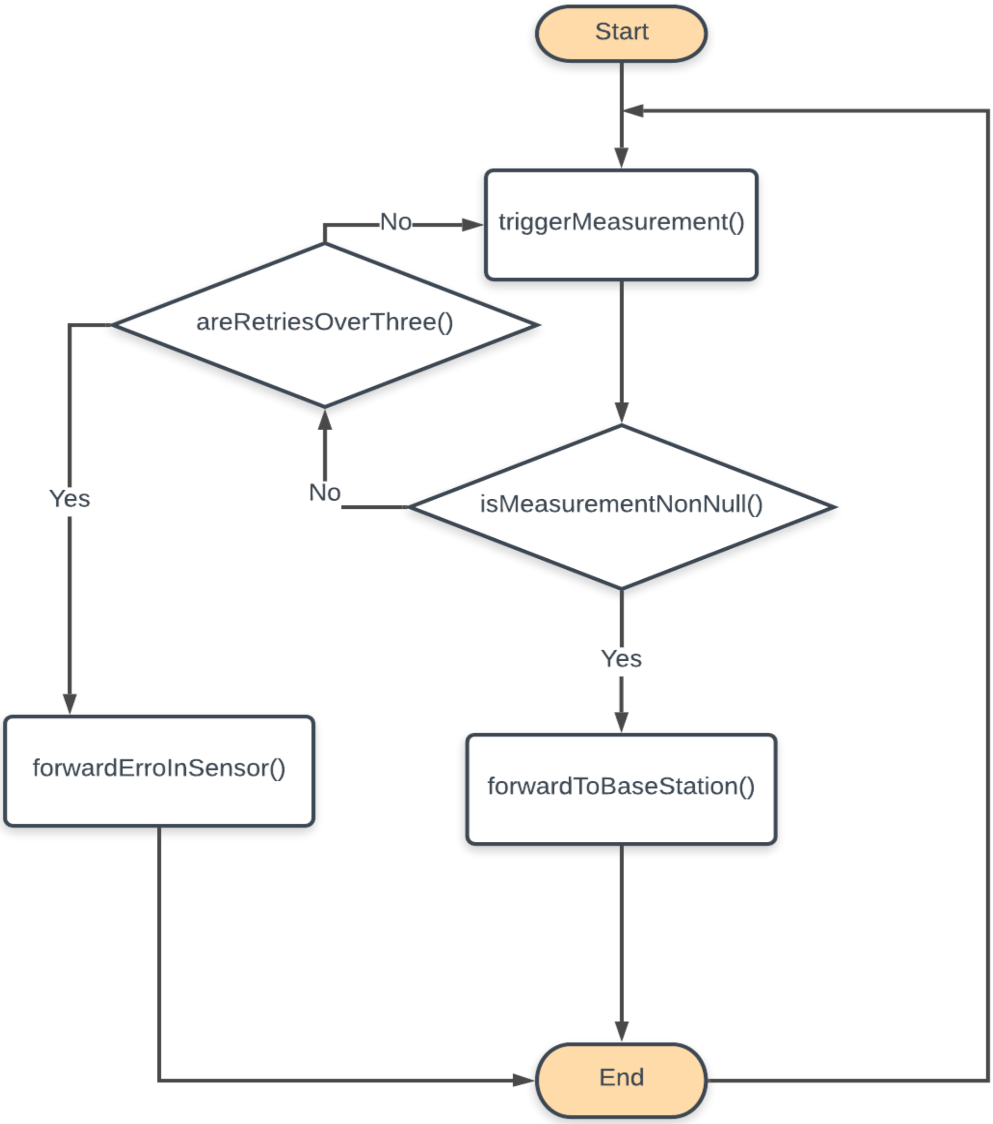


Figure 3.5: Algorithmic flows for moisture microcontroller

or interaction with neutrons. Its structural material may vary, having a very small error margin (up to 4%) in its measurements, taking as Vernier's product as an example [64]. Furthermore its power consumption is very low at 3 mA at 5 VDC, being able to operate in temperature range of -40°C to +60°C [64]. A sensor's price range varies according to its capabilities. For example, in Amazon's store one can find a sensor for 1 dollar while the aforementioned sensor costs 149 dollars.

Regarding the logic that will affect sensor's operation, a main goal, except for the accuracy in measurements, is the least possible energy consumption. Our WSN needs to be energy autonomous and make the best management of the supplied energy. Keeping in mind previous researches we can filter out the choices down to 2, battery and a renewable source of power. Since the latter one, has a high deployment cost while battery is a handier choice, it seems like a better fit for the needs of our system. Of course, in the future moving towards alternate energy sources is a go-to option but currently this implementation could be facilitated by a common battery. As for the moisture sensor's consumption, as said, its operation requires 3mA per measurement. If we make an assumption that in the worst case scenario (which would take place in summer where temperature is higher, thus watering should be more frequent), 7 measurements per day will take place consuming 21mA. Batteries' capacity varies, so if we assume that, in the worst case scenario, we have a case where capacity is equal to 50000mA (e.g. a typical power bank has such a specification), measurements could take place for almost 6.5 years. Apparently, this is not the only functionality this WSN would have, thus its lifespan, for the same basis, would be significantly lower. Nevertheless, it is obvious that regarding energy consumption, measurements will not significantly affect system's lifespan.

As said previously, accuracy in measurements is essential for the success of an IoT application. Incorrect values would be the main cause of mistaken results from the algorithm, thus water consumption would definitely be harmful for the crop while usage may be much more than it would be required, a fact that has economic and environmental consequences. Due to that fact, it is pivotal to ensure that the measured values are precise. According to the moisture sensor that is deployed in the field we can expect the respective quality of measurements. Allegedly, the higher the cost of the sensor is the higher its excellence is going to be. That means that it is going to supply accurate input to the algorithm, probably without needing to calibrate it either manually or programmatically. From what it has been described both in scientific researches and from companies which bestir in the field of agricultural IoT applications, moisture calibration is a time-consuming process which requires a solid equipment. On the other hand a machine learning application could simulate the aforementioned process in the long run, but it would require time both for the training of the system and for testing purposes. Thus, from an economic aspect it is much preferable to invest on a high quality sensor, ensuring the correctness and stability of the algorithm's input data.

Regarding the general algorithmic logic that will direct its functionality, a basic aim, as it has been mentioned, is to optimize its usage, on an energy consumption aspect and respect to the capabilities of how a microcontroller could handle a significant amount of data. Through this way, it can be ensured that our system's lifespan will be long and its accuracy would assist farmer's goals, both helping in the crop's growth and in improving the agricultural management with the data that are provided. In order to achieve that, sensors should work for the minimum possible time while providing correct data to the cloud application. As it will be analyzed furtherly in the following chapters, their operation should be modifiable according to the circumstances. For example, measurements' schedule should be defined according to the weather conditions. In winter, due to low temperature, water is not evaporated with the rate it would in summer. Thus, less measurements could be taken, while being confident that this would not affect the growth of the crop. To conclude, it is important to ensure that their operation, on any aspect, is optimized.

Moisture sensor with its attached microcontroller is a significant component of the WSN. Its ability to communicate with the base station in order to emit the measurements it has made its essential for the fine operation of the system. This connection should be unstoppable and uninterrupted due to the continuous operation of the system. Thus errors and their fallbacks should be designed correctly in order to ensure the successful operation of it according to the schedule that will be followed. This schedule should be kept onto the cloud application in order to assign as less as possible responsibilities to microcontrollers which are not rich regarding their computational resources. Furthermore, the cloud application is the one that will modify this schedule so instead of pushing it to its clients it will keep it on its side and act on a master-slave logic. So, the moisture sensor microcontroller, would work either on a publish-subscribe model, thus it will be able to listen to the requests of the base station or the measurement request will contain the interval until the next measurement. Keeping in mind that our system needs to be energy efficient and a socket approach would require this subsystem to remain in an active state, the latter approach seems more appropriate since it gives to the system the capability to be set in idle mode until it has to take a measurement.

In the best case scenario, measurements will successfully be taken and forwarded to the base station according to the schedule that has been designed. Both parts of the subsystem would work flawlessly regarding the measurement and the emission of the data. If sensor fails to read current moisture levels, a try ‘n error logic should be followed having an upper threshold to the number of retries. According to crop’s needs, may a skip of measurements is not so pivotal but an alert should be sent from the microcontroller to the base station in order to inform the cloud application to forward this error to the farmer to take further actions. On the other hand, if an instability is recognized on the microcontroller’s functionality, whether this is a bug to its routine or a network error, the base station should send a respective alert in order to be taken care accordingly. Through this way, system helps to identify correctly the problems and bugs that might occur thus assist the troubleshooting process which will be followed either by a programmer or an agronomist or a network technician. The first 2 aforementioned scenarios, which’s functionality is directed by the moisture’s sensor microcontroller, are collectively described in the Figure 3.5. The latter scenario, is more of base station’s responsibility to take actions, therefore it will be described in the respective analysis.

Moisture sensors are probably the most neuralgic component of the proposed IoT application. Their contribution is essential since they deliver information regarding the moisture in the field which is used in the algorithm that decides whether the water irrigation should be triggered or not. On the other hand, they provide the data that shows the current condition of the field to the farmer through the accompanying Android application. Their operation affects the lifespan of the WSN, so ensuring its optimization will consequently optimize the whole system’s functionality. To summarize, it should be highlighted that their interconnection with the rest system’s components, though the microcontroller they are attached on, they are the most essential “things” for this IoT application.

3.3.2 Actuators

Actuators is another component which contributes to the composure of this IoT application. Its responsibility, is simply to receive and execute directions, regarding the operational status of the watering system. Therefore, the logic behind its functionality or its actions might not be so complex but after all it is the part of this system that implements the final actions that should be taken, either coming as a result of the algorithm or as the will of the user, and in the aftermath are the ones that ensure the proper growth of the plants, the optimization of water usage and the dominance of such an IoT application against traditional techniques. Its purpose is to automatically switch on or off the watering system, according to directions it has received, without any need of human intervention to its service.

Taking its substance into analysis, an actuator is simply water pump which is attached, as everything, onto a microcontroller which takes directions regarding the operational status of it. Its functionality is quite simple since it is just a switch turning on or off according to the directions it has. Apparently its energy consumption is negligible, although its operation should be uninterrupted. It may receive an update of its status at any time of the day either due to algorithm’s calculation or farmer’s will. Due to the fact that the base station will be orchestrating the functionality and organizing the exchange of messages between the WSN and the cloud application, it could simply alert the actuator for any request that is practically about it. Thus, it will work in a client-server logic where actuator should expect packages from the base station.

Regarding the logic and the software that drives this part of this proposed IoT system, as it has been mentioned, the water valve will be attached onto a microcontroller which will control its operation. This subsystem will be one of the 3 main parts of the WSN which will be deployed to the field among with the soil moisture sensor and the base station. It will be able to communicate with the base station in order to get instructions regarding its operational status. A significant advantage of this subsystem is that it may be in idle state for a long period. This may even be days or months according to the weather conditions which may favor crop’s growth through natural ways (e.g. rain water). In any other case the duration that the valve has to be switched will be sent from the cloud application, either it is algorithm’s result or farmer’s decision, to the base station and sequentially forwarded to actuator’s microcontroller where it will act accordingly.

Happy path flow may fail either due to valve’s hardware failure, meaning that it may not switch on when it should or it did not stop its operation at the end of the sent duration. This could be determined in two ways. The direct one is that the attached microcontroller would get a failure response at each respective request while the indirect one will be identified by the following measurement executed by the moisture sensor. The measured value will be emitted to the cloud application and a comparison could take place. For example, if the direction was to switch on water irrigation and the latter measurement was lower than the previous one, apparently there was a failure in the system. Obviously, the second method would take a lot of time and may be harmful for the crop but it could be implemented as a backup security layer. In any case, fallbacks should be implemented and predicted in order to inform the user to take precautions and measures against such issues.

Water valve’s actuator is another crucial part in the network of “Things” that composes this application. It belongs to the WSN which will be deployed in the field and is responsible to automate every process that would traditionally be done by a farmer. Specifically, its assignment is to complete the task of switching the water valve’s operational status according to the duration it has received from the system in order to cover crop's needs. System's logic is a black-box to this component which only has to implement the aforementioned functionality. Despite that, it is an equally essential part of this set, since it is the one that supplies water to the crop and in conclusion carries out the core mission that this application has to accomplish. Finally, its specifications in terms of water consumption show the importance to optimize its functionality both for crop's growth and on a power consumption aspect.

3.3.3 Base station

Base station in terms of importance is the most pivotal component in our deployed WSN in this IoT application. It is nothing more than a microprocessor which acts as a transceiver of messages among the cloud application and the rest parts of the WSN. Specifically it is the middle man of this whole IoT system forwarding the moisture measurements according to the existing schedule while at the same time pulls off the delivery of the instructions sent by the cloud backend to the actuator regarding the operational status of the water pump. Communication between base station and each other component is implemented through a different protocol according to the requirements and standards that prevail in each case. Therefore, it should be programmed respectively to each scenario in order to carry out and ensure that there will not be a message collision while on the other hand error fallbacks would not lead the system to crash.

Microcontrollers is a rapidly evolving technological gadget with numerous usages. Practically it can pull off any functionality a microprocessor can do while at the same time their capabilities are continuously extending. Due to their nature are mostly used in automation and embedded applications unlike microprocessors, that can be found in PCs, and can be found in a wide variety of appliances like washing machines, smart TVs, smart air-conditions, alarm clock or finally in agricultural smart "things" which is what this thesis is dealing with. Except for implementing a number of functionalities they are capable of communicating, creating either internal networks or contacting with internet applications being able to implement a number of communication protocols which may be designed for low power devices of more traditional approaches, like Wi-Fi or cellular networks, by simply attaching the respective antenna module on them. This part gives us the opportunity to analyze the nature of a microcontroller because base station is simply a microcontroller. Apparently our observations also apply to the previous components since they are attached on microcontrollers which are responsible to implement the logic and the instructions they are supposed to.

A number of brands exist in the market each of which have their own advantages and disadvantages. The 2 most famous ones are probably Arduino and Raspberry Pi. As we will see later on, the choice depends on the requirements of the application that is being designed. It is very possible that both will be employed in an IoT system each of which will take care of different tasks in order to take full advantage of their capabilities respectively. More specifically, we will find out that the proposed application is such an example. Epigraphically we can say that Raspberry Pi has better technical specifications in contrast with the Arduino Uno board. Obviously, from the previous analyses moisture sensor and actuator do not have a lot of responsibilities thus Arduino Uno seems like a good fit for this case. On the contrary, base station which acts as the middle man in our system, manages and directs a quite big load of data and communicates through various protocols depending the case, should be a device that has the capabilities to stand up to the expectations and requirements. Therefore in this case Raspberry Pi would more ideally cover the needs of base station in terms of computational resources.

Arduino’s development started at 2003 as an alternative to BASIC stamp microcontroller which’s cost was around 100 dollars at that time, a significant expense for any student. Thus a much cheaper board was developed in order to cover student’s needs. Regarding Arduino UNO, its most popular edition, it has 14 digital and 6 analog pins to which external appliances can be connected, like a moisture sensor or a water valve actuator, or antenna modules in order to support different communication protocols. This modules and extensions are called *shields* and extend microcontroller’s functionality and capabilities to the desire of the developed. It is powered either by a USB connection or 9 Volt battery, although it can accept power between 7 and 20 Volts. Based on ATmega238P microchip clocked at 16MHz and with a flash memory of 32KB and a static RAM of 2KB it is apparent that on a resources aspect Arduino is not suitable for calculating and handling large amount of data. In Figure 3.6 we can see how an Arduino UNO board is. Regarding its cost, such a board would cost around 20 dollars which is quite an affordable price thinking of the potentials its usage would have for an application like the proposed one.

Regarding its programming language, C and C++ are the two options someone has, by default, in order to develop an Arduino application. Both are famous and well known languages among programmers thus the learning curve, either one is beginner or experienced, is very small. Furthermore, both are not, what someone would call, “heavy” programming languages since they require the minimum of resources unlike options like Java or C# which’s compilation and runtime consumes much more RAM. Apparently, someone can install any libraries’ package in order to work in another language, like Python, but Arduino’s IDE comes with C and C++ installed.

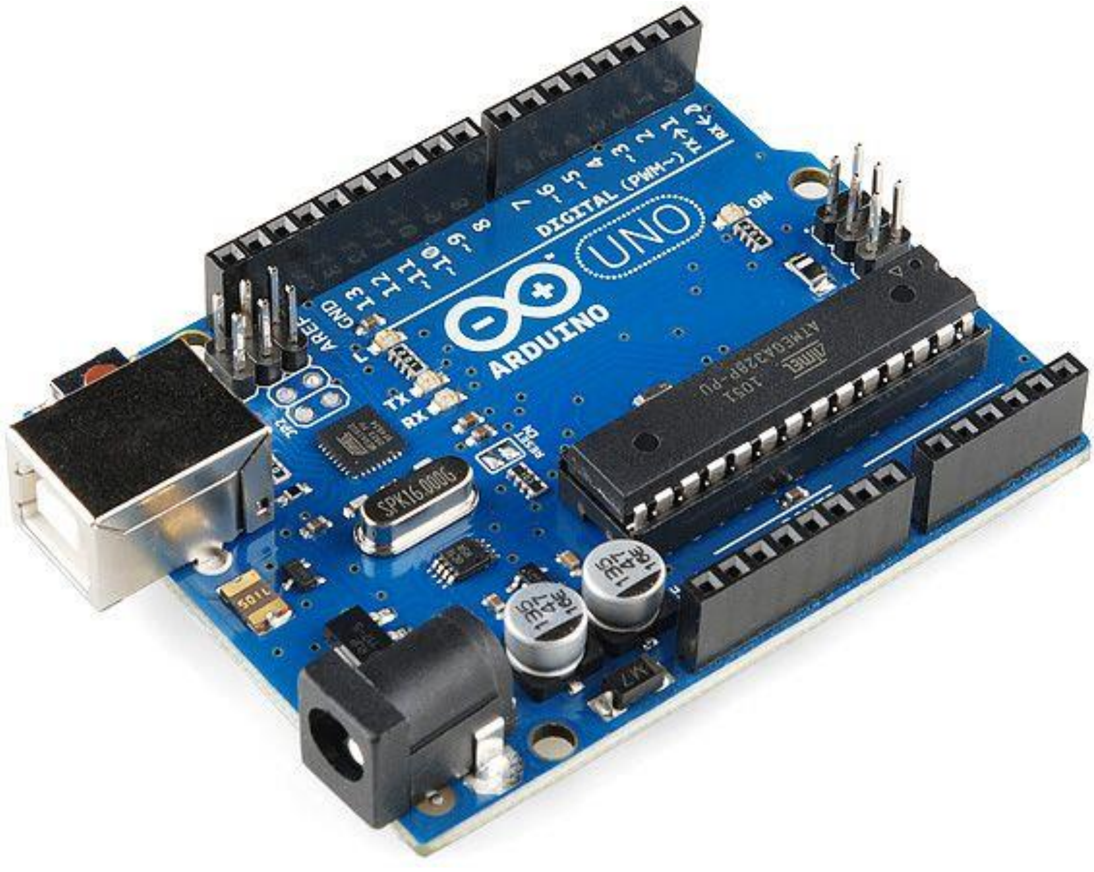


Figure 3.6: An Arduino UNO board

Additionally, this development environment can cover all the typical demands a developer would have like Debug mode, Log view etc.

As it has been mentioned, Arduino's capabilities are limited due to its low RAM memory so as a result the operations it can execute should be simple so it will stand up to the expectations. From a software engineering aspect, code should be well-written, avoiding redundant loops which increase the code's complexity. Data should not be kept for the long run on these microcontrollers for the same reason, since an overflow could be cause even in simple scenarios. On the contrary they should be immediately forwarded to the base station which will orchestrate WSN's operation thus its capabilities and responsibilities are greater.

Obviously, a board like it seems like the perfect choice for the moisture sensor and actuator component of the designed WSN since their flows are quite simple, could definitely be handled by Arduino's microchip in terms of resources and finally it is a costless option. Additionally, in terms of energy consumption each pin consumes 20mA per pin. Thus, for our application, since on each microcontroller will be 2 "Things" attached (moisture sensor and actuator respectively plus an antenna module) there will be a 40mA energy consumption per request for a fully functional case. According to the previously done calculations for the moisture sensor 301mA will be consumed daily in the worst case scenario. Lifespan falls dramatically to almost 166 days for a 50000mA battery, although we have to keep in mind that it is sparing option for such a system.

The alternative to Arduino is Raspberry Pi, and to be explicit and exact it is not an alternative but rather a far more capable and powerful board which can take on heavy calculations and tasks to which Arduino would probably fail to carry out. A significant difference between these two is that Arduino is a microcontroller while Raspberry Pi is a microprocessor, justifying its extended capabilities. The only reason they come into comparison is that they are both extensively used in IoT projects according to their requirements. Its release took place in 2012 in Britain and its initial purpose was to promote computer science in schools and developing countries. Its success was far

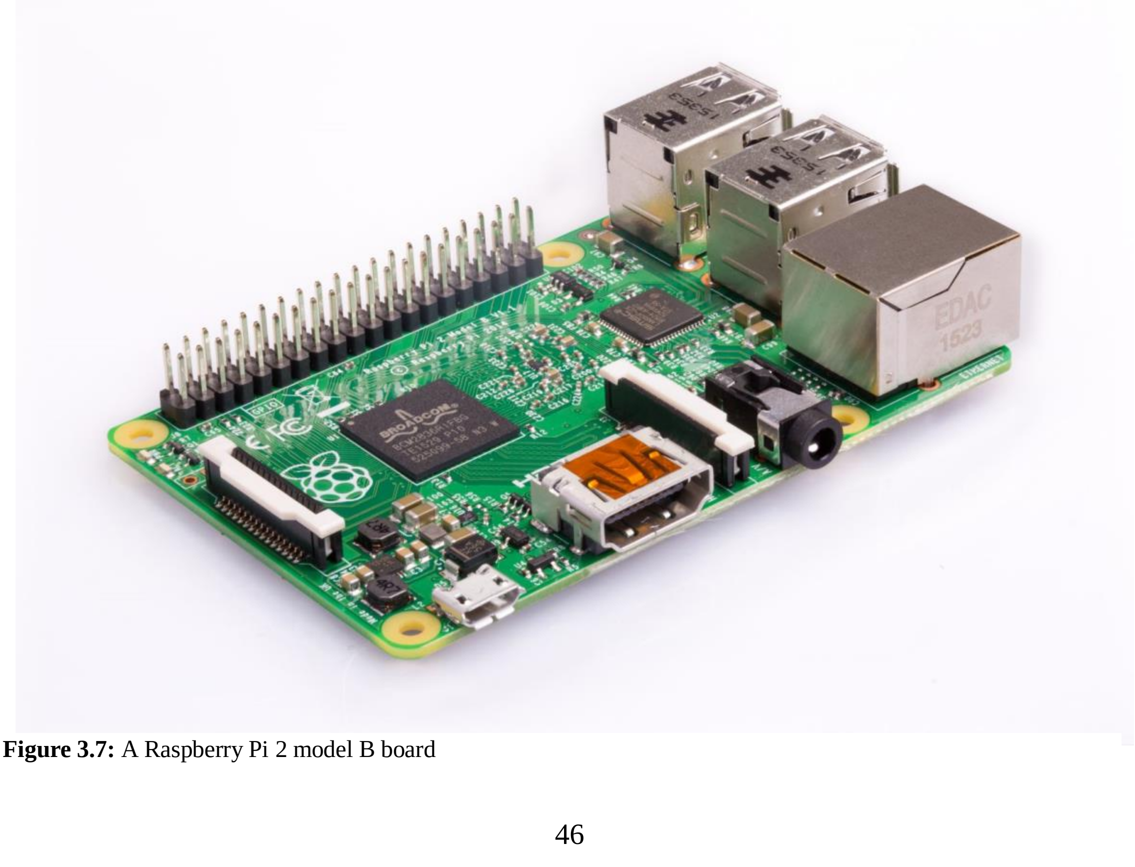


Figure 3.7: A Raspberry Pi 2 model B board

beyond its anticipation, where in 2015 was the best-selling British computer, having continuously increased sales.

From a hardware aspect Raspberry Pi's versions vary according to their capabilities having currently 8 supported models. Raspberry Pi 2 V1.1 would operate at 900 MHz and 1 GB RAM while the latter model is described to be 80% faster according to benchmarks in parallelized tasks. It has 17 general purpose I/O pins to which a number of extensions can be attached like a keyboard or a mouse while at the same time is capable of generating high resolution video of 1080p quality. Furthermore it comes with installed modules for Ethernet, Bluetooth and Wi-Fi where additionally, like in Arduino, with the respective module it can support any communication protocol. Finally, it supports a number of operating systems like Linux, Ubuntu, Windows IoT and Android Things. Its cost varies, with Raspberry Pi 2, for example, coming at almost 35 dollars. Having the aforementioned specifications we can easily understand that Raspberry Pi can overwhelmingly adapt to any challenge may occur in our system from a computational aspect.

The promoted programming languages are Python and Scratch while a number of other languages can be supported with the respective libraries. Its source code is closed although an unofficial open source version may be found on repositories like GitHub. Both aforementioned languages are considered between the easiest ones for someone to learn, regardless its programming experience. Especially, Python is rapidly gaining attention the last years being used in Big Data and machine learning projects due to its ease of use and fast execution. Such applications can be developed in any IDE supporting these languages.

Potentials in using Raspberry Pi are tremendous due to the fact of its continuously increasing capabilities. In the future such WSNs probably will not be in need of being supported by cloud applications. On the contrary, their autonomy will be based on microprocessors like it. Currently the orchestration of the WSN, the forward and delivery of respective messages and some calculations regarding error messages and fallbacks can be easily accomplished by such a powerful device. Connectivity with Internet is a barrier for that especially for projects that require this board to be placed at an urban site. Its resolution would practically cause and exponential growth in its usage.

The functionalities the base station will implement in this proposed system, will be analyzed latterly but it is apparent that Raspberry Pi will be the choice for the development of them. As it has been said, its cost is quite low for its capabilities but energy consumption might be quite high. Specifically its consumption ranges from 220 mA (1.1 W) average when idle to 820 mA (4.1 W) maximum under stress (monitor, keyboard and mouse connected). Apparently, such a consumption could not be facilitated by a battery. Due to the nature of the project an electrical extension could be created from the one that will supply power to the water valve, thus the base station would be based on that workaround. As a future approach integrating solar or wind power is a must in order to increase system's autonomy while optimizing its cost and environmental consequences.

As it has been underlined from the aforementioned observations, system's base station will be responsible for orchestrating the operation of the deployed WSN. It will work as a controller in the exchange of messages between the rest of the already analyzed components and the cloud application. Due to the nature of the application, where at any time user can trigger a switch of the current state of the watering system the base station should always be "listening" for such requests. If our system was based completely on the planned schedule, modified accordingly by the cloud application, it would change its state from idle to active and the other way round respectively. On the contrary, based on the proposed design it should always listen for any update coming from the cloud application. Thus, a publish-subscribe model would be again the go-to option for the communication model between the base station and the server. Regarding the communication model between base station and the moisture sensor, since the schedule is predefined and designed by the cloud application, as it has been said previously, moisture sensor can work in an idle-active model. On the other hand, communication with the actuator will have to follow a similar pattern to the one that was described previously.

Moisture sensor is one of the most important components since it regularly measurements soil's condition regarding moisture, emitting these values in order to be handled by the cloud application. Base station is the middle man for this communication, which's flow is pretty straightforward. The cloud application publishes a message to the base station requesting a value of moisture. Subsequently, base station would wake up the moisture sensor, so it will make a measurement. According to the result of this action it will send the respective message. If the measurement was taken and emitted successfully it will make a POST request to the application with the most latter value. In any case of error it will send either the error message it will receive from the moisture sensor. If communication between them fails, it will inform with the respective error code the cloud application. The distinction among each scenario is happening in order to troubleshoot more effectively the possible errors that may occur. On the other hand, if base station fails to forward the respective message due to connectivity issues or failure of the system the cloud application will be responsible for recovering from this case and alerting accordingly. This hierarchical setup in recognizing possible errors helps in decoupling our system's components in terms of failures and bugs. Architectonically, this means that each component could possibly be more easily reused and extended since it is depended as less as possible to its connected entities.

Actuators are the latter component of the deployed WSN which is taking directions from the base station. This direction simply orders for how long the watering system should be turned on in order to cover the watering needs of the crop. This instruction will come from the cloud application and it is either a result of the algorithm based on the current moisture conditions at the field or a command given by the farmer through the application. In any case, base station plays once again the role of the middle man in an exchange of messages. As it has been mentioned before, due to this functionality base station should always listen for messages from the application, therefore the publish-subscribe model will be the selected communication model. This is not obligatory for the communication model between the base station and the actuator since the desired functionality can be achieved through a simpler logic in order to minimize the duration that the system's microcontrollers will be active.

Regarding any failures and bugs that might occur the logic that will be followed is similar to the previous case. Nevertheless, in this scenario we have a logical advantage which will be described in this paragraph. In anything fails in the communication of these 2 components, the error could be recognized by the moisture measurement. Specifically, if, for example, an instruction for a one hour watering is sent from the cloud application it is expected that following moisture measurement will show a higher value compared to the previous one. If this fails, and no other error message has been sent we can understand that the actuator could not execute the task. Thus, troubleshooting this case can be done through the existing functionality. Apparently, there is a chance that a watering may be skipped but due to the nature of the crops this is quite safe regarding their long term growth and health. In any other case, where the base station would not able to deliver the message to the actuator due to a failure to their connection, it will send a respective error code.

For both of the previous cases we have not yet identified the communication protocol that should be implemented. Apparently, both components are low power devices, of low computational capabilities, depended on batteries in order to be powered and operate. Thus the communication protocol that should be selected should be stable and fast while on the other hand it should not exceed the requirements of the packages that will be sent and received (e.g. small size so there is no need for a great data rate) and the power consumption should be the least possible. As it is has been analyzed in the review of previous researches there have been designed relevant communication protocols which cover such needs but the selection among them is a decision which depends on a number of variables. So as result, protocols that are stable and cover communicational needs are great fits. For example, from the researches that have been analyzed previously, good examples of such protocols is ZigBee and Bluetooth for the internal communication that will take place in the network. Approaches like 6LowPAN, may be a better fit from an extensibility aspect, since it offers an IPv6 address to each node, thus it can automatically communicate with external nodes. But their ongoing status of development, comes with a high uncertainty thus it can't be considered as a perfect candidate. Other approaches like cellular networks or Wi-Fi are apparently selections that exceed the requirements of a WSN in terms of latency, data transfer rate besides the fact of the high energy consumption that such protocols have.

Finally we have to describe the functionality and the communication between the base station and the cloud application. In terms of significance, this is probably the most essential functionality of our IoT system. Without it, it would be nothing more than a couple of "Things" deployed in a field with no ability to communicate with the external Internet. In fact, it is the communication that gives so many potentials regarding the extensibility of functionalities and scalability of responsibilities and tasks such a system could accomplish. Therefore it is important to make the best possible selections regarding the communication model it will be followed between the base station and the cloud application. This communication should start with the deployment of the system. Base station would subscribe to a fixed web-socket address in order to listen to any message from the cloud app.

Its functionality practically has been described in the analysis of the rest components of the WSN. The orchestrator of this IoT system is responsible for successfully delivering the messages to the respective recipients. On the other hand it has to take care of recovering from any fallbacks that may occur and informing accordingly the cloud application. But we have not yet analyzed a critical step regarding the responsibilities of the base station. It has been mentioned that the way the cloud application will identify which WSN is connected to which farmer is quite interesting. It is an important part which has security, integrity and functional concerns that have to be resolved. Taking into consideration that a cellular module having a SIM card would uniquely identify each WSN besides the fact that it covers this application's requirements in terms of bandwidth and coverage it makes it a great candidate as the communication protocol which will support the connection between the base station and cloud system. In terms of security is one of the safest options and the only drawback is its energy consumption. But if we keep in mind that only a few bytes of data will be sent in each request, additionally to the fact that the base station could be powered by the source that will facilitate the water pump, maybe through an outlet, we can understand that the energy consumption will be minor and almost costless.

Since we have identified that a cellular communication protocol is a good fit for the needs of our application for contacting with the cloud system, we can complete this synopsis of base station's operation. After all this information, base station apparently acts as a controller of the directions that have to be implemented by the composing parts of the WSN. Whether it is collection of measurements or the execution of an instruction base station is responsible to correctly forward the message to the respective recipient and ensure its delivery implementing a different communication protocol according to the case. In any scenario where this fails it has to inform accordingly the cloud application so the respective measures are taken for the recovery of the system. Taking in mind that the responsibilities of it could be extended due to the capabilities that are offered by a microprocessor like Raspberry Pi the significance of this component of this IoT application will be automatically increased.

3.3.4 Cloud application

The cloud application is the most crucial component of this IoT application regarding its functionality. It is the part of the system that runs the main algorithm regarding the watering in the field, keeps and presents historical data proving the dominance of the system over traditional techniques, allows the farmer intervene on the algorithm's results and act according to his will and receives error messages that might occur and handles them accordingly. Apparently, most of the responsibilities are implemented by this component, thus its significance is easily understood. Due to this fact, it has to be perfectly designed since every other subsystem is connected to it and any failure of it may corrupt the whole functionality. Besides that, the success of the system is based on the successful run of all the algorithms that create the functionality of this application and will be described thoroughly in this chapter.

A cloud application is a software program where cloud-based and local components work harmonically together. Access to such systems is supplied through RESTful APIs or web-socket approaches in order to download or upload data. Logic components can be executed in an offline mode and their results can be handled online respectively. This is the core difference between a cloud application and a web-based application since the latter one requires a continuous internet connection in order to be functional. On the other hand, the operation of a cloud app is not depended on the connection between server and client but only the delivery or receival of data. The proposed application works on that concept since its core functionality could be executed offline. Calculation of the watering needs, assuming that the input data are available and extraction of measurement schedule are a couple of examples of operations that could be implemented regardless the internet connection.

The initial process that is executed by the cloud application is the registration of the user. This functionality can be offered through an API where the user would make a POST request in order to supply the required information for his registration. This flow has been described previously. The only part that was undefined was the implementation of the connection between the farmer and the field. From the analysis that took place in the previous chapter, the decision was to equip the base station with a cellular module which will have a doubled role. Firstly, it will serve in the communication between the cloud application and the base station and secondly through the SIM card that will be attached on it, each field could be uniquely identified through the phone number that will come with this card. So, the user should complete this field too, in order to connect atomically fields to farmers for integrity and security reasons. The received data shall be kept on a database, cryptographed, and used to validate user upon the login actions in order to present data and actions that refer to the correct field.

Regarding the usual operation of the cloud application it will have to exchange information with 2 entities, the deployed WSN and the Android application being run by the farmer, and perform incoming requests, scheduled algorithm executions and data handling operations. From an aspect of repetition, processes that are apply to the regular algorithm execution and request of moisture measurements are the ones that have to run continuously, thus their significance is pivotal for the smooth operation of the proposed IoT application. On the other hand, requests from the farmer, especially the ones regarding his intervention on the system, may be less, and probably none if the system successfully covers the watering needs of the crop, but are important on a user experience aspect. In this chapter we will thoroughly describe all these tasks that our cloud application will have to carry out on a regular on an on-demand basis from an architectonical aspect, analyze them in reference to their software engineering burden and try to give the best possible design taking variables like the previous 2 into consideration in order to extract a result which will stand out.

Analysis of the cloud application will initialized with algorithm which extracts the result if the crop requires irrigation of the watering system. Algorithm's execution can and will be done offline and that is one of the main reasons to back up the decision to deploy a cloud application instead of a web one. Its only task is to decide whether it should turn on watering system and the duration of this process. In order to achieve that it takes into consideration a number of parameters.

The first one, and probably most pivotal, is the current moisture level in the field. Through the web-socket that in communicates with the base station the cloud application sends the respective message in order to get a measurement. If for any reason this request fails due to connectivity issues, for example, after a predefined number of retries, one execution may be skipped, but it should broadcast a respective alert in order to take actions so troubles are resolved. Besides that, if any error message is received from the WSN it should act accordingly. On the happy path case, where a measurement is received it is used as input for the algorithm and for the presentational data that are shown in the Android application. As said in this study [61], fuzzy logic seems to overcome Boolean logic in some cases. The received value could be categorized as negative, neutral and positive. These three groups will have lower and upper values which will be defined through registration. User sets the crop type and which defines not only the thresholds for if watering system should be triggered but also these groups can be set through it. More specifically, each agriculture has different watering needs, therefore a value may "mean" something different from crop to crop. Thus, the aforementioned groups will be defined from the crop type and will be preset through the development of this application. If the task of our cloud application was to simply irrigate or not the watering system then this could be the selected approach on how to handle moisture levels. Besides that, algorithm's functionality could be simpler through this way. But the WSN is supposed to receive a duration of how long it will operate. Formulas exist on how to extract this duration, a logic which gives high accuracy to the results we will have. Thus, since this accuracy is required in our IoT application, no categorization will be applied to the measured moisture values and pure measurements will be used as input to the algorithm. Except for acting as an input to the algorithm, this parameter will be kept in a database every time is requested for historical and presentational purposes which will be analyzed later.

The significance of the current moisture level of the crop is undeniably apparent. It is the main input data for the main task that is implemented by the cloud application.

Lacking this information, the decision-making algorithm is completely agnostic of the current condition of the field. This is quite a similar scenario, as to a predefined watering schedule, like a traditional approach could be. Thus, the proposed algorithm would have no point of usage. On the contrary, this variable gives meaning to our whole system, as it has been pointed out in the analysis of the moisture sensor. Because of it, the proposed algorithm can extract accurate results and prove the dominance of such an IoT application against traditional techniques.

The second parameter that will be used as an input to the algorithm is the percentage of an upcoming rainfall in the field. This parameter is undoubtedly making the algorithm smart, since its blend with the rest of the variables plays a significant role in saving water for the crop. In order to get that percentage the cloud application can make a request to one of the numerous weather APIs and get value. Here, we can see the meaning of stating the region of the field. This is information will be sent in the request in order to get a weather forecast for the respective location. Subsequently, according to this percentage the algorithm will decide if the watering system should be triggered. Apparently if there is a high possibility of rainfall water could be saved by not switching on the system while the crop would get an amount of water which could cover partially or totally its needs. This variable is a perfect example of where fuzzy logic could be applied. Obviously, a rainfall percentage is not a yes or no result but an entity which value can be easily categorized. So, keeping in mind the previous parameter’s analysis this percentage can be categorized as impossible, possible, very possible and certain. This enumeration can help us simplify the algorithm, end up with a more clear method instead of a complex equation, even apply different formulas according to each case, thus creating a smart approach. Eventually, the percentage of an upcoming rainfall will get under an initial edit in order to be passed as a categorized value so it can be handled by the algorithm.

From the analysis and review of relative literature a very small number of researches took into consideration the weather forecast in order to use it as an asset to their proposed decision-making systems. This is quite surprising, due to the fact that weather conditions critically affect the condition of a field. Furthermore, a proactive exploitation of such the data that can be handed out from a weather API can fairly improve a decision-making system from a long-term accuracy aspect. On the contrary, if, for example, this proposed system ignored this kind of information in its decision-making process it would discover that a rainfall is taking place through the following moisture measurement, and as a result a fair amount of water may have been gone wasted. Thus the importance of this parameter is easily proved.

The final parameter that the proposed algorithm needs in order to extract a result are the lower and upper moisture thresholds. These numbers are the ones that set the limits which if are exceeded then the operational status of the watering system should be switched. In terms of significance they are no less important in a comparison to the moisture level in the field. In fact, they are the ones that define the logic of the algorithm and are compared against the current moisture levels in order to extract a result. Their definition comes through the crop type too, exactly the same way as the as a fuzzy logic approach would have been applied in order to set a categorization to moisture levels which would be different from the one crop type to another. Similarly, each type will have a different set of thresholds and they will be set during the development of the system. Upper threshold has a meaning of terminating the operation of the watering system, since excessive amount of water can be harmful for a crop. On the contrary, lower threshold indicates that is should be checked if the watering system should be triggered taking into consideration the weather forecast. This set of parameters will be pure numbers too, and will not be under any edit since it is required for a simple yet crucial number comparison.

Despite the fact that the value of these variables may not be gained through a “Thing” nor it adds any uniqueness to our system is at least equally important for our algorithm. The design that is proposed, which sets a different value according to the crop type gives the ability to the system to be extremely accurate on the results it will produce each time and these results are strongly depended to the values of these thresholds. If these values were not available, no algorithm could be designed and implemented, thus the proposed IoT application would be useless from a decision-making aspect and would have only informational responsibilities.

```
public class WateringAlgorithm{

     public static int main(String []args){
        float currentMoistureLevel = args[0];
        float rainfallPercentage = args[1];
        float[] thresholds = args[2];
        private static final int CROP_TYPE = 1;
        private static final int SOIL_TYPE = 2;

        if (currentMoistureLevel >= thresholds[1])
            return 0;

        int rainfallCase = getRainfallCase(rainfallPercentage);

        float currentDiff = currentMoistureLevel - thresholds[0];
        float generalDiff = thresholds[1] - thresholds[0];

        float waterCoveragePercentage = (currentDiff/generaDiff)*100;

        long duration = getWateringDuration(waterCoveragePercentage);

        switch(rainfallCase) {
            case 1:
                return calibrateDuration(duration);
            case 3:
                return calibrateDuration((long)(duration * 0.8));
            case 2:
                return calibrateDuration((long)(duration * 0.3));
            default:
                return 0;
        }
     }

     private int getRainfallCase(float percentage) {
         if(percentage <= 0.33)
            return 0;

         if(percentage <= 0.5)
            return 1;

        if(percentage <= 0.8)
            return 2;

        return 3;
     }

     private long getWateringDuration(int waterCoveragePercentage) {
         //TODO usage of the two constants. for sake of ease will return a default number
         return 7200;
     }

     private long calibrateDuration(long duration) {
         if(duration <= 1800)
            return 0;

        return duration;
     }
}
```

Figure 3.8: Source code for watering algorithm

Diving into the details the next thing that will be analyzed is the logic that governs this algorithm. Having the three aforementioned and analyzed variables, this proposed algorithm will try to extract the duration that is required to cover the watering needs of our crop. If the result is zero that means that the system will have to be turned off if it is not. This duration as we said will be sent to the WSN to be handled accordingly by the base station. Apparently this result can be calibrated in order to optimize functionality. For example, if the result is that the duration should be 1ms, obviously it would be a waste of resources from a communication and electrical consumption aspect to turn on a water valve for such a small amount of time. Hence, except for extracting a result our algorithm should modify this number accordingly, taking into consideration some constants.

From a logical aspect, we will thoroughly describe each possible scenario for our algorithm. Each of the thresholds sets a condition which leads to different execution paths. Initially, current moisture level is compared to crop's upper moisture level threshold. If current level is greater than this value then the algorithm should return a zero value in order to turn off. In any other case, a percentage of water need should be generated relative to the comparison between current moisture level and the lower threshold. If we subtract the latter one number from the first one and create an analogy between the result of the subtraction to the difference between the lower and the upper threshold we will have that percentage. The greater the positive difference of the first subtraction the lower will be the percentage. The decline of this difference will equally mean the increase of the watering needs. Taking into consideration approved literature [71] we could create an equation which taking into consideration the crop type, and thus the kind of soil that it is planted into, the water drip rate and apparently the current percentage of existing water in the filed would extract the required duration in order to cover crop's needs, which means that the moisture level after the watering will be almost equal to the upper threshold. The creation of this equation is out of the scope of this thesis since it requires an extensive and scientific knowledge and research on agriculture hydraulics. Due to the lack of this knowledge from the writer, any result would be questionable and doubtful. Regardless that fact, this percentage, from a logical aspect, shows how much water is currently at the field. The higher it is the less water is required from the water valve.

Continuing the analysis of the algorithm, having that duration the usage of the rainfall percentage is introduced. If it is categorized, as said, in four segments {impossible (0-33%), possible (34-50%), very possible (51-80%) and certain (81-100%)} each scenario will produce a different outcome despite the duration it has come as a result. For example, if the previous equation comes to the conclusion that the crop needs a watering for 2 hours but the rainfall percentage is 90%, the algorithm should send a zero value to the WSN. This is scalable, meaning that in the first category we totally ignore the rainfall percentage and as this value is higher the more critically it affects the final result. Consequently, each case will conversely treat the duration, meaning that the second category will slightly decrease the duration, the third will significantly reduce it and the fourth one will possibly ignore it. As we said this result has to be calibrated in order to be sent to the WSN and practically optimize the watering process. This calibration could be improved continuously assuming there was a machine learning algorithm behind it. In this thesis, an empirical and logical approach will be followed. Specifically, if the duration of the watering operation is under half an hour the sent result will be zero. The definition of this time is empirical based on personal experience of the writer in watering olive fields, since for that kind of tree this is a very small period of watering. The whole functionality will be presented in code too, specifically in Java, as it is shown in Figure 3.8.

The aforementioned analysis describes in detail the main task our cloud application is called to carry out. The successful execution of this algorithm is ensuring that our system will cover the watering needs of the field, optimize water usage proving the dominance of an IoT application and showing off the benefits a farmer can get. As it has been clear, although its design may seem and probably is quite simple, even say naive, its complexity is the lowest possible, thus from a software engineering aspect it is an enhanced method that puts the minimum burden on this cloud application. Taking into consideration that this task will be regularly repeated, and in any case where its complexity and its requirements in resources were above average, that could possibly cause a crash of the application. On the contrary, system is stably secured against such a threat through this design. The result of the algorithm, which is the watering duration in order to cover the needs of the crop, will be kept as the moisture level in a database for the same reasons as the ones described previously.

Apparently, optimizations and improvements can be easily applied for the sake of the accuracy. Introduction of other sensors in the deployed WSN like a temperature sensor, one that would measure soil's acidity or a sensor that would measure air's humidity or current wind conditions. Observing the values of these physical quantities would provide additional valuable information to our system about the current condition of the field, hence a more accurate and qualified equation in order to determine the watering duration could be created so as a result this IoT application would be much more improved in accomplishing its main task. Of course, in order to make that possible the intervention of an agronomist would be absolutely necessary, so as to provide a scientifically backed up proposal. It is undeniable that some these values could be acquired without deploying new sensors in the WSN. For example, any weather API

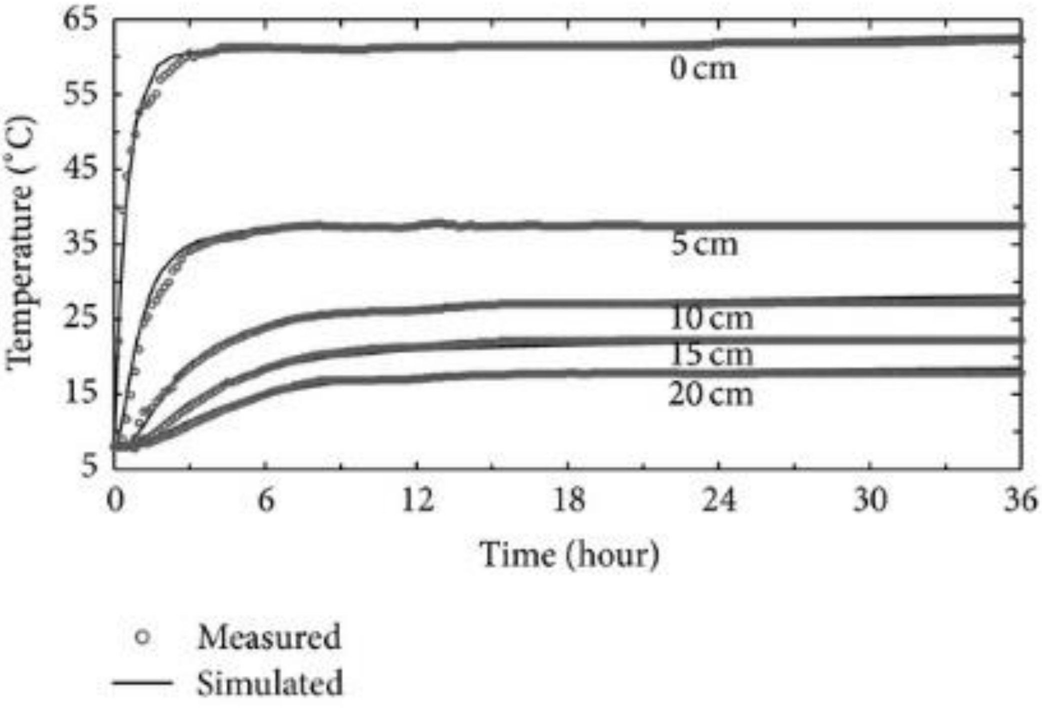


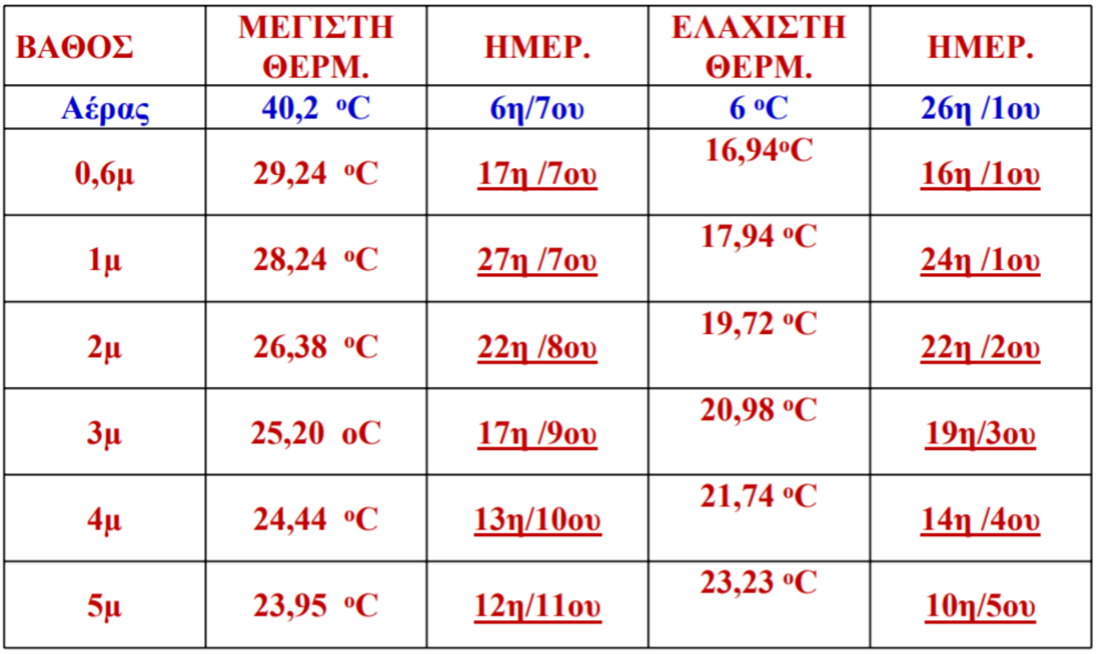
Figure 3.9: Soil temperature in different depths

could supply the temperature of a region. But since this is an IoT application and scalability and extensibility is a never-ending quest for any software, the introduction of this extra sensors would be a future task for this project.

Concluding the analysis and description of the watering algorithm, through a simple and optimized method this cloud application can extract results regarding the duration of watering that would cover a crop's needs while optimizing and reducing water waste and manual labor, engaging the agricultural world into IoT applications and proving the benefits of them. Furthermore, the execution of this algorithm on a cloud application gives the advantage of evolution, from a resource aspect since the growth of the responsibilities it will have, will not be stopped due to lack of computational power because a cloud application can be modified and deployed according to its needs almost immediately.

It is clear that a cloud application would not only be used for the regular execution of a simple algorithm. On the contrary, it has a lot of responsibilities one of which is to define this regularity. The logic behind this repetition is strongly connected to the weather conditions that prevail its time of the year. For example, in winter due to low temperatures water evaporation is generally lower than summer months where high temperatures cause a quicker recurrence of this phenomenon. Due to that axiom, measurements should be more frequent the months where there is high temperature and more rare for months where cold weather reigns. From that assumption, we could define this regularity once, during the deployment of the application. Although, taking into consideration that climate change has highly affected normality of the weather conditions this assumption may not provide the best possible results. For example, in Greece the past few year were recorded quite high temperatures. Hence, the proposed implementation should be equally adaptive to these conditions. Once again, the weather API that will be used in order to determine the rainfall percentage for the watering algorithm will now be taken advantage in order to define a weekly regularity according to general weather conditions.

An approach that could cover our needs in order to produce a schedule which will be designed according to weather conditions of the region in interest. It is quite safe to say



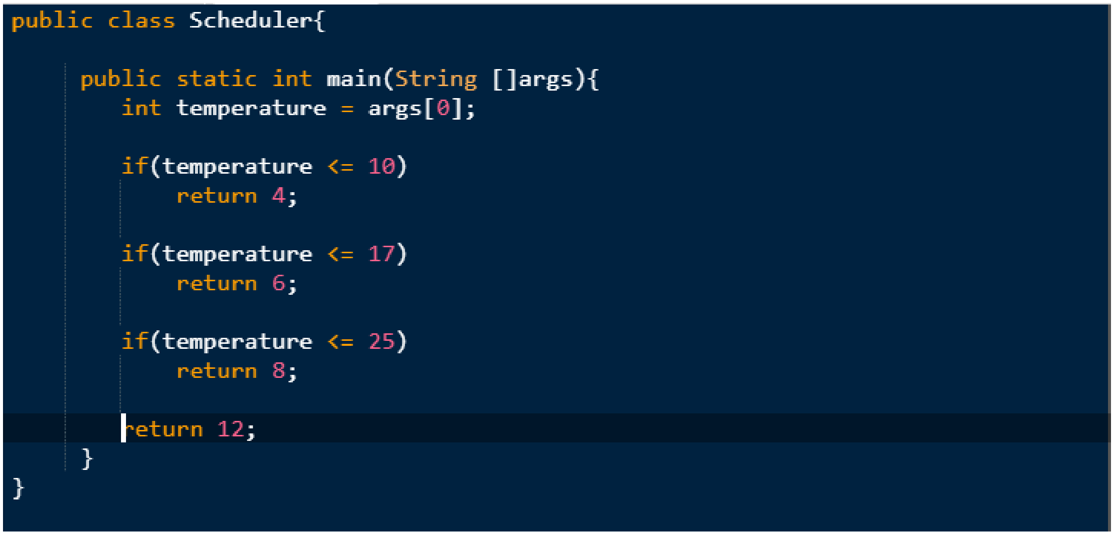
ΒΑΘΟΣ	ΜΕΓΙΣΤΗ ΘΕΡΜ.	ΗΜΕΡ.	ΕΛΑΧΙΣΤΗ ΘΕΡΜ.	ΗΜΕΡ.
Αέρας	40,2 ºC	6η/7ου	6 ºC	26η /1ου
0,6μ	29,24 ºC	17η /7ου	16,94ºC	16η /1ου
1μ	28,24 ºC	27η /7ου	17,94 ºC	24η /1ου
2μ	26,38 ºC	22η /8ου	19,72 ºC	22η /2ου
3μ	25,20 oC	17η /9ου	20,98 ºC	19η/3ου
4μ	24,44 ºC	13η/10ου	21,74 ºC	14η /4ου
5μ	23,95 ºC	12η/11ου	23,23 ºC	10η/5ου

Table 3.1: Soil temperature variance according to depth

that the execution of this task on a weekly basis would practically ensure system's synchronization to weather conditions in order to organize the best possible schedule of execution for the request of field's moisture. As it is described in the previous paragraph, weather conditions affect evaporation and consequently the watering needs of a crop, hence measurements should be more frequent as the evaporation rate increases. Another parameter, that could affect would be the energy consumption that this request would have. But as we have seen previously, this request has negligible demands in power, hence it will not be taken into consideration. The result of this method would be a table of UNIX timestamps which will be kept on cloud application's database and will be checked daily from an orchestrating continuously running offline method which will work on a countdown logic where each stoppage would mean the execution of the watering algorithm according to that schedule.

Regarding the execution of this method, it would be weekly in order to design next week's schedule. As for the logic it will be totally relevant to the prevailing temperatures each day. Additionally, evaporation is higher at soil's surface and analogically lower for greater depths. For instance, as we see in figure 3.9 temperature at surface may be 65 Celsius degrees while in a 20 points depth it reach at a maximum of 15. [72]

An interesting fact is that air temperature, which is the value that will be supplied by the weather API, is not linearly equal to soil's temperature. The following table is quite thorough in proving this assumption, having a constant air humidity of 25%. [73] As it becomes clear, variance of soil temperature is linearly depended to depth, so as depth increases the temperature variance becomes smaller. Table 3.1 is perfectly describing this statement.



```
public class Scheduler{

    public static int main(String []args){
        int temperature = args[0];

        if(temperature <= 10)
            return 4;

        if(temperature <= 17)
            return 6;

        if(temperature <= 25)
            return 8;

        return 12;
    }
}
```

Figure 3.10: Schedule algorithm in Java code

Ignoring variance, it is clear that soil temperature is analogous to air temperature. Except for that, we see that in hotter months soil temperature is higher compared to cold days. Through this table, we can safely move with an implementation that would be based on air temperature. Algorithmically, we will make 4 segments based on daily average air temperature and each group would be an escalation of the previous one. So as a result, in the first group will include temperatures below 10 Celsius degrees. The second one will include values between 11 and 17 degrees, the third one between 17 and 25 while the last one will include all the values above 26 degrees. Each group will be mapped to a number of measurements following the same logic. Hence in the first group, there will be 4 daily measurements, in the second 6, in the third 8 and the last one group will have 12 daily measurements. As it has been said, each produced value will be kept in a table of 7 positions on a database, which will be available to the watering algorithm. The aforementioned algorithm is presented in Java code in figure 3.10.

The combination of the two previously described methods will be done by an orchestrating offline application which will run 24/7. Its main task is to execute the scheduling application once a week in order to extract a schedule, which will direct the calls to the watering algorithm. In order to support the initial cronjob the only requirement is to communicate with a weather API in order to get the weekly weather forecast from which it will extract the average daily temperature, which will subsequently be passed as a parameter to the Scheduler's main method. This process will be repeated 7 times and the results will be kept in 7 position array which will direct the number of the daily required moisture measurements during each week, which in fact means the number of daily calls to the watering algorithm. The same call to the weather API could be used for the parameter that is required from the watering algorithm, only this time current temperature will be used instead of the average one. For the task could be used any scheduling library regardless the language this application is going to be programmed since its functionality is pretty straightforward, thus there is no need for an extravagant and complex approach.

These functionalities compose the, what could be described as, automated operation of the cloud application. The aim of our IoT project, which is the optimized automated watering of a crop, could be achieved from all the aforementioned components which are the WSN in the field, supplying information about the current condition of it regarding soil moisture and taking responsibility in executing taken from the cloud application regarding the operational status of the water valve, while on the other hand the cloud application requests this information and uses it in order to extract a result on how much time should the water valve be switched on. Except for the data about the moisture level in the field the cloud, application is assisted from information that were filed from the farmer during his registration and the current weather conditions, an information that can be taken from an external source like a third party weather API. All that knowledge taken from the WSN, though, could be exploited in so many ways, especially with ones that are useful to a farmer to have a better understanding of its crop condition. Additionally, all the actions taken from the automated processes have great impact on farmer's crop and they could be kept so they can be presented as statistical data which prove the improvement and optimization such an application causes to water usage. On the other hand, a farmer must always have the ability and the freedom to intervene on the automated functionality and act upon his will. Thus, the design of two more actions will be presented, which regard user's interaction with the whole system. Furthermore, an analysis of these two actions will be supplied in order to extensively describe their functionality. Alongside with the login and register actions which have been already described in this chapter, by the end of the presentation of these 2 final methods, a complete description of the cloud application's functionality will have been presented.

Register action as said is the one that supplies pivotal data to the cloud application so it will work according to the requirements. Setting the crop type and the region give the necessary data to processes which will carry out the automated watering process while supplying the telephone number that is registered in the cellular module of the WSN uniquely connects a WSN to farmer. The latter action is the one that gives the ability to the user to see the current status of its field, get access to historical data while at the same time intervene to the automated functionality triggering the watering system according to his will. In order to get access to the three aforementioned actions user has to successfully sign in to the system, which means that he must provide the correct combination of a username and a password. The description of these capabilities will follow in order to summarize the functionality of the cloud application.

After successfully logging in into the system, user gets access in the system and the capabilities that are offered through the mobile application. Initially, he can get a clear image of the current condition of its field through statistical data. More specifically, a daily water consumption until the time of the request can be offered, moisture levels of the crop according to measurements that have taken place until that time, the current operational status of the water valve, the remaining time until the watering process is switched off, assuming it is switched on, and finally the remaining time until the next measurement. Through all this information, a complete overview of the current condition of the field is offered through this service from the cloud application. This data can be fetched through a RESTful API in order to be presented through the mobile application. Taking into consideration that all this information will be kept in a database daily, user can get this data for every day since the system was deployed. Thus the only parameter this call would require is a date, so the service can determine and fetch the respective statistical data. Apparently, as we have already described, these values can be determined from the each time requested moisture level of the crop and τηε result of the watering algorithm. So it is quite clear that the WSN of this IoT application and the watering algorithm are, in fact, the 2 main pillars in almost every functionality the proposed system is offering to the user, either through automation or by presenting crucial information.

Current information regarding field’s condition are becoming in the long term historical data from which valuable inferences can be extracted. Explicitly, these daily values that were described in the previous paragraph could come into comparison with the respective values that would be extracted if traditional techniques were applied in crop’s watering process in order to understand deeper if such an IoT application overcomes these naïve methods. Keeping in mind that, through empirical knowledge, it is easy quite effortless to calculate these respective values for traditional methods. Since water drip rate is known a simple multiplication can provide the water that was used regardless the technique. Thus, total water usage can be determined in order to identify which technique would dominate on the other. This result is not only applied to water usage but also to electricity consumption, hence it affects environment and the economics of a farmer. Calibration is applied through categorization to the monthly sums of these values in order to achieve a more organized comparison between IoT application and traditional techniques from a performance perspective. This data will be available through a simple GET API call to the cloud application, so the user through his mobile application can get access to this extremely valuable information and as a result can manage and organize his actions more effectively.

The final method that will be described here will finalize the functionality of the proposed application and it is actually the one that it is more interesting regarding user’s experience. More specifically, it is where the user can intervene on system’s automated actions and act according to his will. To be honest, this goes against to the purposes of such a system. One of the operations of the proposed WSN is defied and cloud application’s main functionality is turned off for the sake of the user. Apparently, as we have highlighted previously, in an ideal scenario system’s automated functionality will cover any requirement user might have, but he must always have the ability to intervene according to this will. Besides the combination of the information that is provided through current and historical data blended with user’s knowledge and experience may produce results that could be achieved in the long term through a machine learning extension. So user can request at any time a change of the current operational status of the watering system. That means that he will send a custom watering duration which will be forwarded from the cloud application to the WSN and also be kept in the database. Handling to user’s request is no different to the results that are produced from the watering algorithm. Specifically, if user supplies a zero value system would turn off the system. In any other case, the calibration that is applied to watering algorithm’s results will be applied in order to avoid cases where the system is turned on for a very small amount of time.

The analysis and thorough description of these methods summarizes the presentation of the cloud application of our IoT system. It is apparent that this component is in fact the one that directs the functionality of every other entity of the system through either automated or upon-demand processes. A cloud application significantly enhances the capabilities of a WSN system and its incorporation in an IoT composition gives tremendous extensions and possibilities. Without it, information acquired from a network of sensors could not be exploited in an optimal way in order to either handle it for presentation or take some actions according to their values. Especially in a watering system where these values affect crop’s health and water consumption optimization is the only way through, a cloud application ensures the success in covering these requirements. Additionally, it assists in interconnecting a WSN with the external Internet in an advantageous way, since it acts as a gateway and a proxy to requests that may come in the future.

This cloud application integrates communication with a WSN and a mobile application while at the same time it makes regular calculations in order to cover the requirements the aforementioned requests from each client might have. Despite the fact that these methods are not complex nor have any concurrency and low latency issues since high speed are not top priorities in such a system, they are critical for the success of such a project since this affect the optimal growth of the crop while at the same time water usage optimization ensures that environmental and economic benefits. From a software engineering aspect, the proposed system has low coupling while cohesion is high, since it is a simple projects and apparently methods can easily be well called from a functional, sequential and procedural order. These methods that affect the automated and on-demand functionality of this cloud application and as a consequence this IoT system are designed also in a scalable and extensible way since the introduction, for example, of new sensors, hence new parameters to the watering algorithm, can be easily done without needing to rewrite the project from scratch nor refactor it in an extended scale.

3.3.5 Mobile application

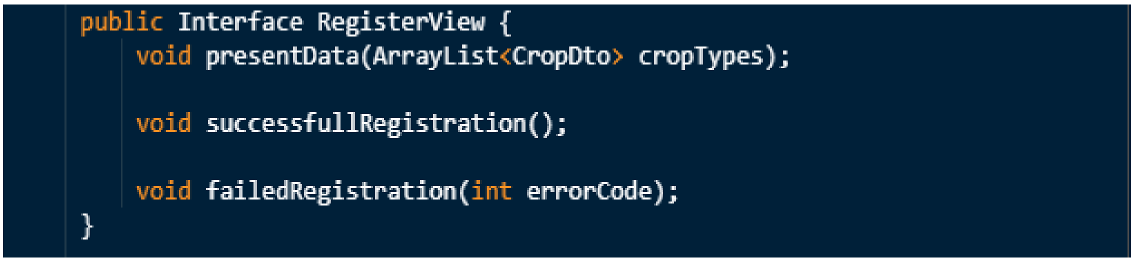
The final component that will be described in this proposed IoT system is the mobile application. It allows the farmer communicate with rest parts, playing a vital role in providing him a clear overview of the current condition of its crop while allowing his intervention on the system. At the same time, through the mobile application user can provide initial data during his registration, that ensure the correct and integral operation of the cloud application in the parameters that will be passed to the watering algorithm and the display of the respective data that will be requested. The aforementioned facts have been described already in the previous paragraphs, hence in this part while revisiting them an analysis of the mobile application functionality from an architectural and UX aspect will be conducted while the requirements for the communication with the cloud application will be analyzed.

Mobile applications is rapidly evolving technology and from the first ones that paved the way to introduction and integration of Internet of Things. In fact, they also "Things" belonging to this greater ecosystem where a device can communicate with other devices through Internet. In this IoT project, the role of this application is mostly presentational since it provides the user with data regarding current condition of the crop and comparison views against to traditional techniques. On the other, which it hands out an interventional functionality which helps the user act on the watering irrigation system according to his will. Apparently, this is not a convoluted system, as every component in this IoT application, but the possibilities of future enhancements in the functionalities of it, direct a clear and extensible architecture. Specifically, for this mobile application, which for design purposes, will address only Android devices the MVP design pattern will be applied.

MVP stands for Model-View-Presenter and is an extension of the MVC (Model-View-Controller) pattern found mostly in web applications. It focuses on mobile applications, helping to distinguish business logic from data presentation. This is a serious problem in front-end software engineering, since a bad practice is that one class takes responsibility of fetching, handling and presenting data ending up to god classes, great coupling and minimal cohesion which makes extension and enhancement of such a project a really difficult, and even impossible, process. Specifically for Android, classes that extend the Activity package are, epigraphically, classes which hold the views that present data. It is quite usual phenomenon, in smelly code projects, that these classes end up to fetch, handle and present data, keeping in all the objects in their entity and as a result it is impossible to extend and scale the functionalities of a mobile application. As we have already said, the possibilities of this project are quite surprising so the design of this mobile application should not be an exception from this philosophy. So our aim is to embrace these opportunities through the architectural proposal of it. Specifically, each activity, and generally every component that will be used to present data to the user, will implement an interface which will be used by a presenter which is responsible for communicating with the cloud application to fetch and send the respective information. Each model will be serialized and deserialized according to the receiver in order to facilitate the incoming and outgoing requests. Usually, view methods will take as parameter the data that are responsible to present and they are responsible to act according to whom value. For example if they expect a list and due to an error or due to lack of data, they should show the respective error view in order to inform the user.

This design pattern is strongly assisting the introduction of unit and functional testing in an application. More specifically, since the communication between the presenter and the view is being carried out through interfaces, hence it is based on pure Java classes and no Android-based code is introduced in it, unit testing can be mocked without needing to mock view based classes, which is considered a bad practice in mobile application development. Additionally, regarding functional testing, interface calls can be broken down as much as possible, so as a result every possible usage path ca be tested kin advance, knowing as result the ending up of every logical flow.

This mobile application is composed from 3 activities, Android's main class to hand out views which will present data, the RegisterActivity.class, the LoginActivity.class and the MainActivity.class. Each of this classes is responsible to implement an interface which is a communication contract between views and their respective presenters. Regarding registration form, in order to present the initial data it will have to make a



```
public Interface RegisterView {
    void presentData(ArrayList<CropDto> cropTypes);

    void successfullRegistration();

    void failedRegistration(int errorCode);
}
```

Figure 3.11: Register interface methods’ signatures

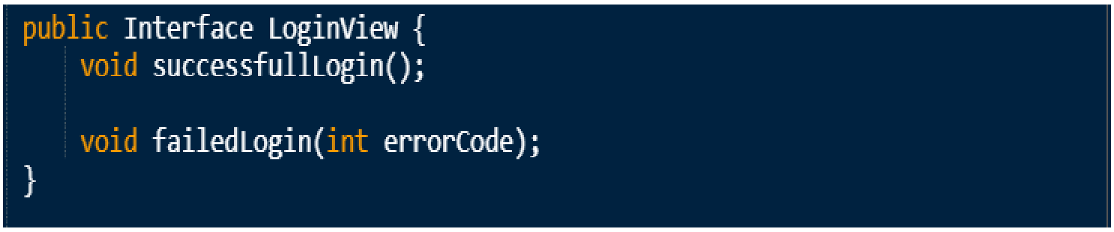
```
public class RegisterPresenter {

    private RegisterView view;

    public RegisterPresenter(RegisterView view) {
        this.view = view;
    }

    public void getData() {
        //make api call to cloud application to fill the arraylist
        ArrayList<CropDto> cropTypes = new ArrayList<CropDto>();
        view.presentData(cropTypes)
    }

    public void register(String region, int cropId, String email, String password, String fieldMobile) {
        //make api call with payload the seriealized parameters
        //if success
       view.successfullRegistration();
       ///if error
       view.failedRegistration;
    }
}
```

Figure 3.12: Register presenter methods and dummy implementation for presentational purposes

request to the cloud application in order to fetch the available crop types. Having this information the method that will present the information required will be the one with a signature taking as parameters an array of crop types. The rest data will be manually set from the user. After entering the required data the mobile application will post a payload with the inputs, which will have been validated on the client in order to ensure integrity and accuracy. Response will determine if a success of a failure, with the respective error code, will be presented. These methods will be defined on the interface and will be implemented through the activity. Presenter is responsible for the communication with the cloud application and takes the respective interface as a constructing parameter but is agnostic on how the data will be presented or which the caller of its functions is, hence ensuring extensibility, scalability and pure code. In this case data must not be persisting, thus it can be passed only as parameters in order to keep a lightweight object in memory. So in Java code we will have the classes that are shown in figure 3.11 and figure 3.12 respectively.

Regarding login it has only one action to implement from an aspect of communication with the cloud application and that one is to post authenticate user’s input in order to sign in him in the system. So from an architectural perspective, the respective view and presenter have not a lot of responsibilities. Specifically, presenter must make an API call to the cloud application with the suitable payload and has to inform the view of the result of this request. Despite that, LoginActivity is critical for the mobile application because through the authentication that is taking place in the context of it, user can get access to all the functionalities this system can provide to him, whether it is about presentational purposes or taking action regarding watering system’s operational status. Additionally, through this process it is ensured that user will be presented to data that refer to his uniquely connected crop, thus the integrity and trustworthiness of this IoT application is enhanced.

This is a very important step although it is considered deprecated in the mobile application development community. Specifically, it is a general guideline to provide as much content as possible without requiring a login. But in this case, since user can get access to information that are correlated with private economic data or take actions regarding water usage, it is obvious these actions must be protected on a verification level. In the future a possible extension of application's capabilities may offer information and knowledge that would not be related to private data, but the current proposal does not intend to do that, at least currently. In conclusion, despite the simplicity of this class it is the one that keeps the mobile application and consequently the whole IoT system safe from data manipulation, giving access to all the functionalities that can be provided from this design. The following images (Figure 3.13, Figure 3.14) present the responsibilities of the view and the presenter respective, once again in Java.



```
public Interface LoginView {
    void successfullLogin();

    void failedLogin(int errorCode);
}
```

Figure 3.13: Login interface method's signature

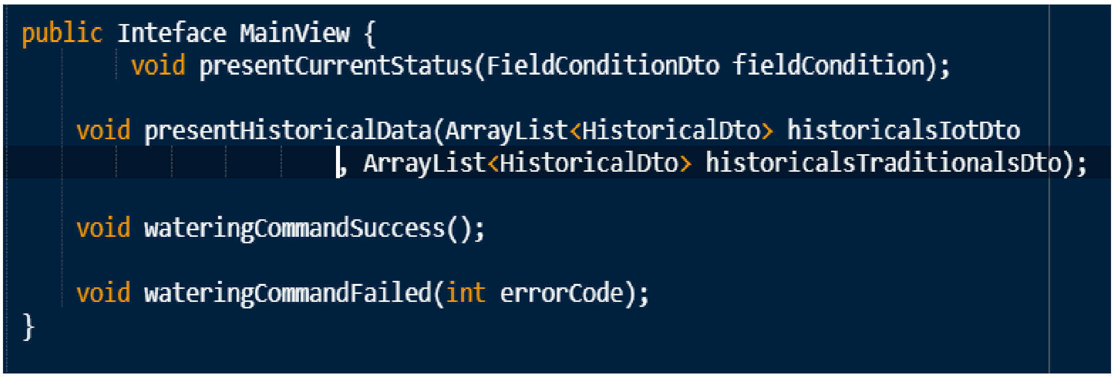
Finally, the last activity to analyze is the one that, from a UX aspect, is the most important to the user. MainActivity is the class that keeps views which are responsible for presenting information relative to crop's current status, historical data proving system's superiority over traditional techniques and the ability to intervene on automated operation. This class is no exception to the design pattern that is followed architecturally for the development of this application, hence a view-interface and the respective presenter will be responsible for carrying out the requirements that have been set. In that case, this is the most interesting component of the proposed mobile application. A class that presents a, possibly, large amount of information to the user has to be perfectly designed in order to reduce to the minimum network requests and

```
public class LoginPresenter {
    private LoginView view;

    public LoginPresenter(LoginView view) {
        this.view = view;
    }

    public void login(String username, String password) {
        //make api call with payload the serialized parameters
        //if success
        view.successfullLogin();
        //if error
        view.failedLogin(someResponseErrorCode);
    }
}
```

Figure 3.14: Login presenter methods and dummy implementation for presentational purposes

avoid keeping large objects on its memory. Thus, techniques like lazy loading, taking an offset and a page size as pivotal parameters in order to define current position, and data persistence is a couple of examples that could optimize the software design of this class.

Regarding the models are kept in the presenter and they are passed as parameters to the interface, implemented from the MainActivity, in order to present them accordingly.



```java
public Inteface MainView {
        void presentCurrentStatus(FieldConditionDto fieldCondition);

    void presentHistoricalData(ArrayList<HistoricalDto> historicalsIotDto
                       , ArrayList<HistoricalDto> historicalsTraditionalsDto);

    void wateringCommandSuccess();

    void wateringCommandFailed(int errorCode);
}
```

Figure 3.15: Main interface method’s signature

```java
public class MainPresenter {
    private MainView view;

    public MainPresenter(MainView view) {
        this.view = view;
    }

    public void getCurrentFieldCondition() {
        //make api call
        //if success
        view.presentCurrentStatus(someFieldCondition);
        //if error
        view.presentCurrentStatus(newFieldCondition);
    }

    public void getHistoricalData(int offset, int pageSize) {
        //make api call
        //if success
        view.presentHistoricalData(someHistoricalIotListDto,
                                   someHistoricalTraditionalListDto);
        //if error
        view.presentHistoricalData(newHistoricalIotListDto,
                                   newHistoricalTraditionalListDto);
    }

    public void waterTheField(long duration) {
        //make api call
        //if success
        view.wateringCommandSuccess();
        //if error
        view.wateringCommandFailed(someErrorCode);
    }
}
```

Figure 3.16: Main presenter methods and dummy implementation for presentational purposes

FieldConditionDto is a model that includes the moisture of the field according to the latter measurement, current operational status of the valve and the remaining time until the end of the watering assuming the valve is operating. As for the HistoricalDto, the model that is responsible for presenting the comparison between application's performance against the one of traditional techniques, hence passing two lists of these objects to the view referring to each performance respectively, it is an object where which keeps the monthly value of water and electricity consumption. Finally regarding user's intervention in this case the watering duration is the only parameter that is required for this call to the cloud application in order to execute user's demand.

Logic is quite simple this time too, so both the view and the presenter have only a small number of methods in their contract to implement in order to accomplish the requirements, despite the fact that their communication is significantly more frequent compared to what happens in the rest classes. More specifically, in the other scenarios communication between interface and the presenter takes place only when making a request to the cloud application (register and login actions respectively). On the contrary, in this case while initializing activity class there have to be done two requests in order to present current and historical data. Calls to load more historical data may take place throughout the usage of the application while user can choose to intervene to automated functionality at any time. Thus, it is apparent that this class is far more demanding in terms of resources and perfection of design.

Concluding regarding class' responsibilities, this is apparently the most interesting entity from a code development aspect. Business logic, despite its simplicity, has an interestingly demanding data handling due to the load of it that may be fetched from requests to historical data while the architectural design must always inform the user accordingly to the results of his actions. As in every previous case code examples of a high level design of the class are shown in order to get a better understanding of the blueprint of it (Figure 3.15 and Figure 3.16).

This class is not only interesting for its business logic but it is quite demanding on the UX design that has to be implemented in order to optimize user's experience. Generally, it is a standard that user can get access to all data with the least possible actions. Taking into consideration that most of the times he will probably want to know what the current condition of the field is, it is profound that this must be the first thing to present to him. Hence, after signing in to the system the first request must be the one that will fetch the current status of the field and with the successful completion of it, its display will take place. Historical data are of less importance from a logical aspect, so it may be a one-click-away action in order to execute the request of fetching it and presenting it in a graphical view that can be supported from several available libraries. From a high level aspect, a design with two tabs may facilitate the proposal that has been defined here. Regarding the action of the user with which he will be able to intervene on system's operational status, it should also be easily accessed, thus a sticky button at the bottom of the screen, independed from the current tab the user is on, or an action menu item on the header seems like a suitable choice. With this template, an easy access to all the critical data is provided, optimizing user's experience and engaging him in a more smooth way in accepting IoT projects through integrating mobile applications to them, with whom he may be already familiar, in order to explore the capabilities of IoT systems.

3.3.6 Conclusions of application's architectural design

The thorough description of the mobile application is the one that closes up the analysis of the proposed IoT application. This final component, which allows the user to interact with the system by getting information while having control over the main process that takes place in the automated operation of our system, complements the rest parts of the proposed architecture, serving as a tool for the user in order to execute a number of important functionalities. It is strongly connected to the cloud application, in order to get data and send instructions, but on the other hand through its registration form essential information are filled from the user, so as a result the algorithm can successfully run in order to extract results about the watering process. Mobile phones are a critical piece of the IoT ecosystem and their integration in this application enhances the extensibility and possible capabilities of the system.

This last component ends the presentation of this IoT system. Through the proposed design, a harmonic cooperation of all components is aimed, where entities are loosely coupled in terms of operational dependency but provide to each other all the required data to support their fine functionality. The versatility of this IoT application facilitates in the gradual extension of it by exploiting the potentials of each subsystem. Additionally the internal architecture of each component is outlined in a way to support the requirements with the minimal effort, whether that is computational resources, electrical consumption or optimization of the communication interface that is going to be used in each scenario. Furthermore, the designed flows, mostly the ones regarding user's experience and on a secondary level the ones that deal with the automated functionality of the cloud application and its cooperation with the WSN, aspire to provide an optimized and functional path in order to cover the needs that show up each time.

Regarding the communication among the subsystems, a part of which will be the main concern in the following and last chapter regarding the analysis of the architecture, selected protocols are tested solutions which ensure communications' stability. WSN, through its base station, will use cellular data in order to send and receive packages based on the websocket that will be established between base station and the cloud application. Despite the fact that this protocol is exceedingly overwhelming in terms of capabilities compared to the requirements this application has, but other protocols like 6LowPAN or LoRaWAN as it was derived from the literature review lack in stability since they are on an ongoing status regarding their development. On the other hand, approaches like ZigBee or Bluetooth, although they have minor energy consumption and their capabilities fit application's requirements, are more suitable for small area networks rather than communicating with a cloud application. As for the mobile application, the installed solutions like Wi-Fi and cellular data cover the needs of our application. Finally, cloud application has to implement mostly offline processes, with calls to external weather APIs, in order to fetch the required data to execute, for example, the watering algorithm.

The aforementioned findings and assumptions for this IoT application complete the analysis of it as a software system. Such a system is no exception from the general guidelines in software development and architecture, thus all the approaches that were proposed. Some parts may are still unclear, like the completed methods that are going to be used in the watering algorithm or specific libraries that will be used in graphical components used in the mobile application. This happens because either the definition of such methods is out of the scope of this thesis, since the assistance of an agronomist is required in order to ensure the stability of the proposed approach, or the selection of a library it is not a matter of concern or worth of external analysis. Closing up, this architectural approach tries to cover every requirement this path has, taking care of every possible usage scenario while aims to system’s stability, extensibility and scalability.

4. Conclusions

4.1 Comparison between the proposed system and similar approaches

In the previous chapter the proposed system has been thoroughly described by analyzing each component of it and the way they interact with each other. Communicational and functional architecture have been brought into table, which's design aims to create a stable yet extensible and easily enhanced system. Responsibilities have been divided in a way that dependencies between components is the least possible while at the same time each subsystem's entities are strongly connected. That means that possible failure of a component would not affect functionality of another one. Apparently, side effects would interfere but in any case operation would not be stopped. Additionally, this lack of dependency assists the extensibility of the application since the modification and addition of features of an entity would not directly affect the rest ones, thus minor changes would enhance system's general improvement.

In terms of functionality, this IoT application utilizes the capabilities of a wireless network of a soil moisture sensor and a base station equipped with a Raspberry Pi, which collects measurements from this sensor at regular intervals in order to forward them to a cloud application which will calculate whether the crop is in need of watering. This information will be transmitted back to the aforementioned network, specifically base station will handle it, in order to inform accordingly an actuator to change the operational status of a water valve for a specified duration. Cloud application extracts this result through a decision-making algorithm which is based on the aforementioned measurements, current weather conditions and data defined from the farmer during his registration. It also extracts the regularity of measurements by adapting to current weather conditions which are strongly connected to the moisture levels that will prevail in a field. Finally, it is responsible for serving an API which will facilitate requests from the mobile application. This application is the last component and the portal for a user/farmer to interact with the whole system. Through his registration he supplies crucial data which help the algorithms' functionality while at the same time he can get an overall overview of the current status of his crop in terms of current moisture levels while at the same time he can requests data that will bring in comparison results in water usage when using such a system against applying traditional techniques. Last but not least, he may interfere to the automated functionality and change on-demand the current operational status of the watering system.

The first impression may be a little misleading. More specifically, this application indeed looks almost completely similar to researches that have been presented in the second chapter. In fact, this is not far from truth, since in the context of a big picture this application gets information from a WSN which are handled accordingly from a cloud application in order to decide about the status of the watering system. Most approaches that adopt these functionalities seem to follow this main idea. However, despite this undeniable fact, the overall setup of this approach, the architectural design, the technologies used and the design patterns that are applied on an architectural level make this system stand out against approaches that have been analyzed previously, thus adds value to the current standards of agricultural IoT applications especially if we take into consideration other accepted architectural approaches [75]. In this chapter these differences will be analyzed in way that will help any reader to understand the significance of this project.

From a logic aspect, as it has been already pointed out, this application has minor differences compared to similar IoT projects. But its setup is really a standout comparing to others. Specifically almost every research ([29], [31], [32], [34], [35], [37], [46], [49], [56]) proposing a similar approach, present a communication where decisions are taken either internally in the WSN or after communicating with a web/cloud service. On the contrary, this thesis proposes a system where 3 entities coexist harmonically supporting the optimization of water usage while informing the user about the current status. Only two researches integrate a similar approach, but either the information is presented only through a website [35], hence there is no utilization of the capabilities a smartphone can provide, or there is no automation and communication with a cloud application [57], thus lack of resources may be faced in a future expansion of this specific system.

The importance of the aforementioned observation is literally the vision that is encapsulated in the extension of usage of IoT applications in society’s everyday life. Specifically, enhancing an IoT system with the capabilities that are offered from smartphones which are irreplaceable to current lifestyle can lead to the creation of a continuously interconnected huge ecosystem of multiple functionalities. Through them, without ignoring the assistance offered from the respective cloud applications, various IoT systems could setup a communication among them, thus exploiting functionalities of each other which can lead to tremendous possibilities. This thesis, proves that the integration of a mobile application to a seemingly simple and trite approach can lead to a great user experience, by engaging the user to the core functionality of the automated mode of an IoT system, while at the same time it broadens the horizon of extending functionalities of such a tool. Improving such an important agricultural process like watering can have positive impact on multiple sectors like economy, environment and technology. Apparently, security plays a vital role in the stability that is offered by such a system. In this thesis, this part is not analyzed as other components but in future works getting support by other researches [76] it could be improved significantly.

Architectonically, each entity can affect any other but this can happen only through a middle man, in this case the cloud application, which orchestrates the functionality and the accomplishment of each request done. The fact that the whole orchestration and core logic is being executed in a cloud application ensures that in any case of extension, there will not be any concern regarding computational resources. On the other hand, the existence of a smartphone application in this system can give tremendous opportunities due to the rapid growth of the capabilities of smartphones. Last but not least, the design of the WSN makes it isolated in terms of functionality but strongly connected to the cloud application when it comes to decision making. That means, that if, for example, a new sensor node was added in order to measure another physical quantity in order to improve accuracy no other node would be affected and the same implementation would be applied for the communication of the new sensor with the base station. Most, if not all, researches did not emphasize on the software engineering aspect in terms of architecture. Taking into consideration the expected radical growth of IoT application architectural design is a pivotal concern that should be executed as perfectly as possible in order to take advantage of this growth in the smoothest way. Bad architecture could lead to rewriting or even abandonment of a project. This thesis offers a pattern to be followed by future researches on how to combine functionalities from different entities, in order to conduct a diverse IoT ecosystem.

Regarding each component specifically, the proposed WSN is far different from most approaches. Specifically, on a high level it has minor responsibilities since it simply gathers data and executes instructions received from the cloud application. That decision facilitates the whole concept of extensibility. Assigning responsibilities to a component of low computational resources is an uncertain decision since this may backfire if massive extension of capabilities would be required. On the contrary, its functionality can be easily extended by adding sensors but modification of existing codebase would be insignificant, thus stability would be offered. As for the communication inside the subsystem, the proposal follows the standards that have been set by previous works since it doesn't aim to provide a new way/protocol of communication in the IoT ecosystem. So, after taking into consideration all previous similar works ([29], [30], [41], [51], [56], [61], [63]) that have been reviewed and analyzed, it is apparent that ZigBee protocol is a number one choice for a WSN like the one proposed here. In respect to the communication with the cloud application, the selected cellular network protocol ensures stability, high speeds and outrageous capabilities taking into consideration that 5G networks are coming in the game in next few years. Additionally, the selection of cellular network gives an easy workaround regarding login process that is supported from the mobile application of the system, relative to the correct connection between data of a WSN and the user that requests them.

The decision to assign the core responsibilities to the cloud application is another reason which makes this approach a stand out against previous works. A number of researches assigned data aggregation and complex calculations to microcontrollers/processors, which as a result may cause a system crash due to overflow. On the contrary, in this case WSN's responsibility is to simply collect and forward data and execute instructions. A cloud application, can increase its capacity on demand thus be modified according to the needs of the system. Additionally, integrating a cloud system to a machine-learning application in order to enhance the automated functionality of an IoT application, is far easier compared to implementing such an algorithm to a WSN, which would mean the redeployment of the whole system. Apparently, big data comes into play having all this information available, meaning that patterns could be extracted, hence optimization could be achieved [74]. That would not be possible if any other entity was responsible of any core functionality.

Another differentiating factor between this application and the rest ones that have been investigated is the quality of the parameters to the watering algorithm. Most approaches would take as many physical quantities as possible into consideration in order to extract a result. Apparently, this is not wrong at all, but the most important variable to identify whether a crop is in need of watering or not, according to formulas found in scientific books [71], is soil moisture. Additionally, soil moisture is highly depended to weather conditions, hence this approach gives significant weight into considering them in order to extract a result. Other approaches would clearly rely, on the current circumstances, practically taking near future out of the equation. Adding another source of knowledge to this system ensures its diversity and accuracy of results. Obviously, the rest of the physical quantities that other researches mention could be taken into consideration in future extensions of the system but they wouldn’t have the impact weather forecast has to such an application.

4.2 Future work

Undeniably, IoT is an emerging technology which’s evolution will affect everyday life and world economy in a diverse way. Agriculture is a critical sector which has economic, environmental and dietary extensions. Automating its standard procedures will produce a totally positive outcome for the aforementioned extensions. More specifically, optimizing watering process is probably one of the most important and critical operations that can be improved. The approach that has been described throughout this thesis can easily support this statement and the architectural setup that has been proposed makes this system a standout.

Regarding future works that could be applied on this thesis the implementation of the aforementioned system is the one of the most profound approaches. Apparently, coding for such a complex and multilevel system would not be something easy while on the other hand testing its functionality would take a lot of time since a number of edge cases should be tested while a system that is operating 24/7 can produce a number of unexpected bugs until it is stabilized and its errors are fixed. Nevertheless, the implementation would extract all the potential value this project has offering a system that is, at least in Greece to writer’s knowledge, one of its kind. Additionally, the improvement of the algorithm by introducing more factors in order to enhance accuracy and optimization that is brought by the project is definitely a top choice for future approaches. Apparently, collaboration with agronomists is obligatory in order to get the best possible algorithm that would have equivalent results. Finally, introducing Big Data algorithms in the project would assist the self-improvement of the system while it would provide knowledge to other possible interconnected systems.

Bibliography

[1] A. Nordrum, Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated. IEEE Spectrum, 2016

[2] G.H. Peters, P. Pingali, Tomorrow's Agriculture: Incentives, Institutions, Infrastructure and Innovation, 2002

[3] Mulla, D. J. (2013), Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps, Biosystems Engineering, 114(4), 358–371. doi:10.1016/j.biosystemseng.2012.08.009

[4] D. K. Asmamaw, A Critical Review of the Water Balance and Agronomic Effects of Conservation Tillage under Rain-fed Agriculture in Ethiopia, 2016

[5] Luca Catarinucci, Danilo De Donno, Luca Mainetti, Luca Palano, Luigi Patrono, Maria Laura Stefanizzi, and Luciano Tarricone. An IoT-Aware Architecture for Smart Healthcare Systems. IEEE Internet of Things Journal 2015

[6] V. A. Memos, K. E. Psannis, Y. Ishibashi, Byung-Gyu Kim, B.B. Gupta. An Efficient Algorithm for Media-based Surveillance System (EAMSuS) in IoT Smart City Framework. 2017

[7] Naser Hossein Motlagh, Tarik Taleb Osama Arouk. Low-Altitude Unmanned Aerial Vehicles-Based Internet of Things Services: Comprehensive Survey and Future Perspectives. IEEE Internet of Things Journal, Vol. 3, No. 6, December 2016 899 - 922

[8] F. Shrouf, J. Ordieres, G. Miragliotta. Smart Factories in Industry 4.0: A Review of the Concept and of Energy Management Approached in Production Based on the Internet of Things Paradigm. 2014

[9] E. Ackerman and E. Guizzo, "5 technologies that will shape the web," Spectrum, IEEE, vol. 48, no. 6, pp. 40-45, June 2011.

[10] A. Zaslavsky, C. Perera, D. Georgakopoulos, Sensing as a Service and Big Data, 2013

[11] A.P. Plageras, K.E. Psannis, C. Stergiou, H. Wang, B.B. Gupta, Efficient IoT-based sensor BIG Data collection-processing and analysis in Smart Buildings, Future Generation Computer Systems (2017), https://doi.org/10.1016/j.future.2017.09.082

[12] F. Firouzi, A. M. Rahmani, K. Mankodiya, M. Badaroglu, G.V. Merrett, P. Wong, B. Farahani, Internet-of-Things and big data for smarter healthcare: From device to architecture, applications and analytics, Future Generation Computer Systems (2018)

[13] G. Manogaran, R. Varatharajan, D. Lopez, P.M. Kumar, R. Sundarasekar, C. Thota, A new architecture of Internet of Things and big data ecosystem for secured smart healthcare monitoring and alerting, Future Generation Computer Systems (2017), https://doi.org/10.1016/j.future.2017.10.045

[14] S. Wolfert, L. Ge, C. Verdouw, M. Bogaardt, Big Data in Smart Farming – A review, Agricultural Systems 153 (2017) 69–80

[15] H. Channe, S. Kothari, D. Kadam, Multidisciplinary Model for Smart Agriculture using Internet-of-Things (IoT), Sensors, Cloud-Computing, Mobile-Computing & Big-Data Analysis, Int.J.Computer Technology & Applications,Vol 6 (3),374-382 (2013)

[16] Schönfeld M.., Heil R., Bittner L. (2018) Big Data on a Farm—Smart Farming. In: Hoeren T., Kolany-Raiser B. (eds) Big Data in Context. SpringerBriefs in Law. Springer, Cham

[17] J. Muangprathub, N. Boonnam, S. Kajornkasirat, N. Lekbangpong, A. Wanichsombat, P. Nillaor, IoT and agriculture data analysis for smart farm, Computers and Electronics in Agriculture Volume 156, January 2019, Pages 467-474,

[18] Lovas, Róbert and Koplányi, K and Élő, G (2018) Agrodat: A Knowledge Centre and Decision Support System for Precision Farming Based on IoT and Big Data Technologies. ERCIM NEWS (113). pp. 22-23. ISSN 0926-4981

[19] https://en.wikipedia.org

[20] B. Varghese, R. Buyya, Next generation cloud computing: New trends and research directions, Future Generation Computer Systems 79 (2018) 849–861

[21] Z. Ning, X. Kong, F. Xia, W. Hou, X. Wang, Green and Sustainable Cloud of Things: Enabling Collaborative Edge Computing, IEEE Communications Magazine (Volume: 57 , Issue: 1 , January 2019, pp: 72-78)

[22] O. K. J. Mohammad, Recent Trends of Cloud Computing Applications and Services in Medical, Educational, Financial, Library and Agricultural Disciplines, ICFET '18 Proceedings of the 4th International Conference on Frontiers of Educational Technologies, Pages 132-141

[23] A. Botta, W. de Donato, V. Persico, A. Pescap`e, Integration of cloud computing and Internet Of Things: A survey, *Future Generation Computer Systems* (2015), http://dx.doi.org/10.1016/j.future.2015.09.021

[24] A. L. Molthan, J. L. Case, J. Venner, R. Schroeder, M. R. Checchi, B. T. Zavodsky, A. Limaye, R. G. O'Brien, Clouds In The Cloud, Weather Forecasts and Applications within Cloud Computing Environments, 2015 American Meteorological Society DOI:10.1175/BAMS-D-14-00013.1

[25] Ghose A., Biswas P., Bhaumik C., Sharma M., Pal A., & Jha, A. (2012). Road condition monitoring and alert application: Using in-vehicle Smartphone as Internet-connected sensor. 2012 IEEE International Conference on Pervasive Computing and Communications Workshops.doi:10.1109/percomw.2012.6197543

[26] Khatoun R., Zeadally S. (2016). Smart cities. Communications of the ACM, 59(8), 46–57. doi:10.1145/2858789

[27] S. Pongnumkul, P. Chaovalit, N. Surasvadi, Applications of Smartphone-Based Sensors in Agriculture: A Systematic Review of Research, Hindawi Publishing Corporation, Journal of Sensors, Volume 2015, Article ID 195308, 18 pages, http://dx.doi.org/10.1155/2015/195308

[28] P. P. Ray, A survey on Internet of Things architectures, Journal of King Saud University - Computer and Information Sciences, Volume 30, Issue 3, July 2018, Pages 291-319

[29] R. M. Carreras, J. A. P. Franco, A. Iborra, A Testbed to Evaluate the FIWARE-Based IoT Platform in the Domain of Precision Agriculture, Sensors 2016

[30] https://www.fiware.org

[31] V. Sukhadeve, S. Roy, Advance Agro Farm Design with Smart Farming, Irrigation and Rain Water Harvesting Using Internet of Things Advance Agro…, December 2016

[32] R. Dhalla, H. Agrawal, An Improved Energy Efficient Duty Cycling Algorithm for IoT based Precision Agriculture, Procedia Computer Science 141 (2018) 135–142

[33] J. Hsu, A.Kansal, S. Zahedi, M. B. Srivastava, V. Raghunathan (2006), "Adaptive Duty Cycling for Energy Harvesting," *International Symposium on Low Power Electronics and Design (ISLPED'06)*:180-185.

[34] G. Hong, C. Hsieh, Application of Integrated Control Strategy and Bluetooth for Irrigating Romaine Lettuce in Greenhouse, IFAC-PapersOnLine 49-16 (2016) 381–386

[35] G. Parameswaran, K. Sivaprasath, Arduino Based Smart Drip Irrigation System Using Internet Of Things, International Journal of Engineering Science and Computing, May 2016, 5518 – 5521
[36] https://www.arduino.cc/
[37] P. Archana, R. Priya, Design and Implementation of Automatic Plant Watering System, International Journal of Advanced Engineering and Global Technology, Vol 4, Issue 1, January 2016
[38] https://firefly-iot.com/product/firefly-arduino-shield-2-4ghz/
[39] https://www.cooking-hacks.com/documentation/tutorials/lorawan-for-arduino-raspberry-pi-waspmote-868-900-915-433-mhz/
[40] https://www.cooking-hacks.com/sigfox-module-for-arduino-waspmote-raspberry-pi-intel-galileo-868-mhz-7184
[41] J. Gutiérrez, J. F. Villa-Medina, A. Nieto-Garibay, M. Á. Porta-Gándara, Automated Irrigation System Using a Wireless Sensor Network and GPRS Module, IEEE Transactions On Instrumentation And Measurement, Vol. 63, No. 1, January 2014
[42] F. J. Ferrández-Pastor, J. M. García-Chamizo, M. Nieto-Hidalgo, J. Mora-Pascual and J. Mora-Martínez, Developing Ubiquitous Sensor Network Platform Using Internet of Things: Application in Precision Agriculture, Sensors 2016, 16, 1141; doi:10.3390/s16071141
[43] Cai, J.; Liu, Y.; Lei, T.; Pereira, L.S. Estimating reference evapotranspiration with the FAO Penman–Monteith equation using daily weather forecast messages. Agric. For. Meteorol. 2007, 145, 22–35.
[44] A. Giri, S. Dutta, S. Neogy, Enabling Agricultural Automation to Optimize Utilization of Water, Fertilizer and Insecticides by implementing Internet of Things (IoT), 2016 International Conference on Information Technology (InCITE) – The Next Generation IT Summit
[45] Sorensen, C.G., Fountas, S., Nash, E., Pesonen, L., Bochtis, D., Pedersen, S.M., Basso, B., Blackmore, S.B., 2010. Conceptual model of a future farm management information system. Comput. Electron. Agric. 72 (1), 37–47.
[46] A. Kaloxylos, R. Eigenmann, F. Teye, Z. Politopoulou, S. Wolfert, C. Shrank, M. Dillinger, I. Lampropoulou, E. Antoniou, L. Pesonen, H. Nicole, F. Thomas, N. Alonistioti, G. Kormentzas, Farm management systems and the Future Internet era, Computers and Electronics in Agriculture 89 (2012) 130–144
[47] Q. U. Khand, A. R. Barket, Smart Water Distribution for Irrigation System (SWDIS), International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-5, Issue-4
[48] F. O. Bamigboye, E. O. Ademola, Internet of Things (Iot): It’s Application for Sustainable Agricultural Productivity in Nigeria, Proceedings of the iSTEAMS Multidisciplinary Cross-Border Conference University of Professional Studies, Accra Ghana 2016
[49] D.K. Sreekantha, A. M. Kavya, Agricultural Crop Monitoring using IOT- A Study, 2017 11th International Conference on Intelligent Systems and Control (ISCO)
[50] A. Halim, M. N. Hassan, A. Zakaria, L. M. Kamarudin, Internet of Things Technology for Greenhouse Monitoring and Management System Based on Wireless

Sensor Network, Malaysian Technical Universities Conference on Engineering and Technology, At Johor Bahru, Malaysia, October 2015
[51] https://www.igi-global.com/dictionary/the-beacon-enabled-mode/55743
[52] http://www.memsic.com/memsic/
[53] S. Sawant, S. S. Durbha, J. Adinarayana, Interoperable agro-meteorological observation and analysis platform for precision agriculture: A case study in citrus crop water requirement estimation, Computers and Electronics in Agriculture 138 (2017) 175–187
[54] https://www.raspberrypi.org/
[55] http://www.sensorml.com
[56] I. Mat, M. R. M. Kassim, A. N. Harun, I. M. Yusoff, IoT in Precision Agriculture Applications Using Wireless Moisture Sensor Network, 2016 IEEE Conference on Open Systems (ICOS), October 10-12, 2016, Langkawi, Malaysia
[57] R. Shahzadi, M. Tausif, J. Ferzund, M. A. Suryani, Internet of Things based Expert System for Smart Agriculture, *(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 9, 2016*
[58] El-Dessouki, A., S. Edrees, and S. El-Azhari. "CUPTEX: An integrated expert system for crop management of cucumber." ESADW-93, May, 1993.
[59] Mohammed, M., K. EI-Arby, and A. Rafea. "LIMEX: an integrated expert system for lime crop management." In Second IFAC/IFIP Enr AGENG Workshop on AI in Agriculture, Netherlands. 1995.
[60] Plant, R. E., F. G. Zalom, J. A. Young, and R. E. Rice. "CALEX/Peaches, an expert system for the diagnosis of peach and nectarine disorders." HortScience, Vol. 24, no. 4 ,1989.
[61] Azaza M, et al. Smart greenhouse fuzzy logic based control system enhanced with wireless data monitoring. ISA Transactions (2015), http://dx.doi.org/10.1016/j.isatra.2015.12.006
[62] Y. Kim, R.G. Evans, Software design for wireless sensor-based site-specific irrigation, Computers and Electronics in Agriculture 66 (2009) 159–165
[63] C. Z. Zulkifli and N. N. Noor. Wireless Sensor Network and Internet of Things (IoT) Solution in Agriculture. Pertanika J. Sci. & Technol. 25 (1): 91 - 100 (2017)
[64] https://www.vernier.com/products/sensors/sms-bta/
[65] Andreas Plageras, Kostas Psannis, Yutaka Ishibashi and Byung-gyu Kim, IoT-based surveillance system for ubiquitous healthcare, 42nd Annual Conference of IEEE Industrial Electronics Society, Piazza Adua, 1 - Firenze (Florence), Italy October 24-27, 2016. [DOI: 10.1109/IECON.2016.7793281]
[66] Evangelos Balasas, Kostas Psannis and Manos Roumeliotis, Performance Evaluation of Routing Protocols for BIG Data application, 6th International Symposium & 28th National Conference on Operational Research, “OR in the digital era - ICT challenges”, Thessaloniki, University of Macedonia, Geece, June 8-10, 2017
[67] Christos Stergiou and Kostas E. Psannis, Recent advances delivered by mobile cloud computing and Internet of Things for Big data applications: A Survey, International Journal of Network Management, doi: 10.1002/nem.1930 May 2016
[68] Dharma Agrawal, B. B. Gupta, Shingo Yamaguchi, and Kostas E. Psannis, Recent Advances in Mobile Cloud Computing, Wireless Communications and Mobile Computing, https://doi.org/10.1155/2018/5895817, Volume 2018, Article ID 5895817

[69] Christos Stergiou, Kostas E. Psannis, B.B. Gupta, and Yutaka Ishibashi, Security, Privacy & Efficiency of Sustainable Cloud Computing for Big Data & IoT, Sustainable Computing, Informatics and Systems, Elsevier, June 2018, https://doi.org/10.1016/j.suscom.2018.06.003

[70] C. Stergiou, Kostas. E. Psannis, A. P. Plageras, Y. Ishibashi, B.-G. Kim, Algorithms for efficient digital media transmission over IoT and cloud networking, Journal of Multimedia Information System, vol. 5, no. 1, pp. 27-34, March 2018

[71] Γ. Α. Τερζίδης και Ζ. Γ. Παπαζαφειρίου. *Γεωργική Υδραυλική*. Εκδόσεις Ζήτη 1997

[72] Jiangbo Han and Zhifang Zhou, Dynamics of Soil Water Evaporation during Soil Drying: Laboratory Experiment and Numerical Analysis, The Scientific World Journal Volume 2013, Article ID 240280, 10 pages http://dx.doi.org/10.1155/2013/240280

[73] Vrachopoulos M. Gr., Geothermy - Natural Step (Solar-ground energy), GTC – Rhodes Technologies & Applications RER in islands 8-9 May 2009

[74] C. Stergiou, K. E. Psannis, "Recent advances delivered by Mobile Cloud Computing and Internet of Things for Big Data applications: a survey", Wiley, International Journal of Network Management, pp. 1-12, May 2016. Doi:10.1002/nem.1930

[75] C. Stergiou, K. E. Psannis, A. P. Plageras, G. Kokkonis, Y. Ishibashi, “Architecture for Security in IoT Environments”, in Proceedings of 26th IEEE International Symposium on Industrial Electronics, 19-21 June 2017, Edinburgh, Scotland, UK.

[76] C. Stergiou, A. P. Plageras, K. E. Psannis, T. Xifilidis, G. Kokkonis, S. Kontogiannis, K. Tsarava, A. Sapountzi, “Proposed High Level Architecture of a Smart Interconnected Interactive Classroom”, in Proceedings of IEEE conference SEEDA-CECNSM 2018, 22-24 September 2018, Kastoria, Greece.