CYPRUS UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF COMMUNICATION & INTERNET STUDIES

[1]

BACHELOR'S THESIS:

"E-AGRICULTURE: DEVELOPING A DECISION SUPPORT SYSTEM FOR PRECISION FARMING"

Loukas Konstantinou

Limassol 2018

[1] Cyprus University of Technology: http://green.cut.ac.cy/wp-content/uploads/sites/42/2015/09/cut-gr.png

CYPRUS UNIVERSITY OF TECHNOLOGY

FACULTY OF COMMUNICATION AND MEDIA STUDIES

DEPARTMENT OF COMMUNICATION AND INTERNET STUDIES

Bachelor’s Thesis:

“E-AGRICULTURE: DEVELOPING A DECISION SUPPORT SYSTEM FOR PRECISION FARMING”

Loukas Konstantinou

Supervisor

Dr. Labros Labrinos

Limassol 2018

Copyrights

ACKNOWLEDGEMENTS

Finally, I am sitting here trying to summarize my time as student these past four years. It is a genuine pleasure to express my deepest sense of gratitude and appreciation to my mentor Dr. Labros Labrinos. His keen interest, meaningful assistance and tireless guidance and patience have helped me to accomplish this task. I also owe a deep sense of gratitude to the thesis committee: Dr. Nikolas Tsapatsoulis and Dr. Constantinos Djouvas for their insightful observations, their engaging questions and stimulating conversations. Last but not least, I would like to thank Lucy who has taught and put up with me in the past four years and also my joy knows no bounds in expressing my cordial gratitude to my best friend and thesis buddy Helen. I humbly extend my appreciation to all the people, colleagues and classmates who have willingly helped me out with their knowledge and intelligence. Thank you.

ABSTRACT

The present thesis, entitled "e-Agriculture: Developing a Decision Support System for Precision Farming", was pursued by Loukas Konstantinou, an 8th semester student of the Department of Communication and Internet Studies at the Cyprus University of Technology, under the supervision of Dr. Labros Labrinos and was completed in May, 2018.

This research belongs to the field of Information and Communication Technologies but takes on a cross-disciplinary approach, since it engages the field of agriculture and specifically the farming sector. The purpose of this study is to develop an effective Decision Support System that gathers weather based and agricultural data, formulates them and displays the most prominent results, prompting the user to take the appropriate action. By achieving this goal, this research also contributes to Precision Agriculture and sustainable food production. The overall Decision Support System comprises hardware and software elements and it is broken down into a pair of activities. Each activity consists of two smaller parts for better management. The methodology that was adopted for the purposes of the current study is quantitative, while the method is the experiment. For accomplishing this experiment, particular equipment was employed for gathering the data, a platform was utilized for storing the information and an Android application was developed for formulating and presenting the most crucial results back to the user.

Additionally, as far as the outcomes are concerned, the Decision Support System has been fully developed and formed, according to the comprehensive and detailed system design, composition and arrangement of the various components. A complete testing of the system in actual farmlands and with farmers or farming consultants was not possible due to time constraints. Nevertheless, the proposed Decision Support System is available for deployment and usage.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

API:	Application Programming Interface
DSS:	Decision Support System/s
HTTP:	Hypertext Transfer Protocol
Hz:	Hertz
ICT:	Information and Communications Technologies
IDE:	Integrated Development Environment
IoT:	Internet of Things
IP:	Internet Protocol
km/h	Kilometer/s per Hour
LoRaWAN:	Long-Range Wide-Area Network
mm:	Millimeters
MQTT:	Message Queuing Telemetry Transport
NLP:	Natural Language Processing
PA:	Precision Agriculture
PF:	Precision Farming
SWOT:	Strengths, Weaknesses, Opportunities, Threats
USB:	Universal Serial Bus

CHAPTER 1: INTRODUCTION

"We are stuck with technology when what we really want is just stuff that works", according to Adams (2002). We often hear about the potential of innovative technology, which is going to transform our lives and, if leveraged on a larger scale, it will thoroughly innovate every sector as we know it. However, this does not apply in the case of agriculture.

A research, conducted by the McKinsey Global Institute (2015), showed that the agricultural sector had the lowest digital penetration rate, compared to any other industry. In detail, the productivity percentage dropped by one percent, between 2005 and 2014, whereas other industries have witnessed massive productivity improvements. Although Cisco predicts the connection of fifty billion devices by 2020 (Evans, 2011), agriculture fails to keep up to the expectations.

For the above-mentioned reasons, the goal of this research becomes quite apparent. The purpose of the current experimental study is to develop an effective Decision Support System (DSS) that satisfies the need for digital agricultural transformation. Nevertheless, it is necessary to state that effectiveness derives from multiple studies that have been reviewed (and will be demonstrated in the next chapters).

To meet the objectives, the research focuses on the design and the creation of a DSS. The development involves both hardware and software. As far as the hardware is concerned, sensors that take into consideration different sources of information, are used to acquire data. At the same time, a formulated system is responsible for providing and motivating the farmer with the best possible decision.

To summarize, the following chapters will deal with specific topics and affairs. The chapter that comes right after the introduction, defines the problem and explains the necessity for conducting this research, thus, specifying the research question. Next, a literature review will be presented, focusing on similar studies that are related to the research field. By the time the revision is finished, relevant terms will be defined and explained in the chapter of theoretical background. Moreover, the chapter of methodology will describe the tools for accomplishing the purpose of this study, as well as the overall research process, including the sampling and the analysis. The next three chapters will demonstrate the requirement analysis, the overall design (focusing on the theoretical formation) and the implementation of the suggested DSS (aiming the overall construction of the system) respectively. Finally, the last chapter will present the final conclusions, concentrating on the suggestion of improvements and highlighting limitations that the study faced.

CHAPTER 2: PROBLEM DESCRIPTION – STUDY NECESSITY

In the next years to come, the field of agriculture and especially the farming industry will play a key role in shaping everyone's lives. According to Meola (2016), the United Nations Food and Agriculture Organization estimates that the world will have to produce 70% more food in 2050 than it did in 2006, to cover the increasing food demands of the growing population. In line with what was stated, the Joint Research Centre of the European Commission (2014) claims that an "agri-tech revolution" is essential and the only way to push for this kind of modernization is through innovation-driven solutions. As a result, agriculture needs to enter a new phase, called "e-Agriculture".

Chauhan (2015), defines e-Agriculture as an emerging field that utilizes modern information and communication technologies to enhance agricultural and rural development. For that purpose, a modern farming management concept has been developed, known as Precision Farming (PF), that uses Internet of Things (IoT) technologies to increase agricultural output, reduce wastage of resources and cut down production costs (Balafoutis et al., 2017). PF, as the researchers explain, is often based on the usage of DSS. Hayman (2011), describes these kind of systems as computer-based, interactive entities that offer decision-making information, based on various sources of information. As a result, anyone would assume that developing these kind of systems, equals with boosting Precision Agriculture (PA), therefore e-Agriculture expansion. However, various studies present a contradicting conclusion.

According to Nguyen, Wegener and Russell (2005), the uptake and adoption of such systems is slow and gradual. Besides that, the prospects of DSS acceptance and approval are poor. The findings of their research summarize that modern DSS need to be effective. To achieve this, these systems need to be inexpensive, low cost and be able to adapt in every possible situation. If these criteria are met (according to the scientists), then these systems will be widely adopted. Lindblom, Lundström and Ljung (2014), conducted a similar research and their discoveries were no different (even after nine years). Even though they highlight the fact that DSS are the core of PA, they are still not widely accepted and adopted. The reason for that, is the lack of effectiveness. Ineffectiveness derives from the lack of user-centered design, high cost of such systems and inadequate sources of information. Finally, an investigation that took place in Hungary by Lencsés, Takács and Takács-György (2014), presented the following. Farmers, especially Europeans, hesitate to adopt PF, because of DSS. The extracted opinion is that these systems increase working time and have high operational costs. Consequently, it is understood that agriculture cannot be transitioned into e-Agriculture if PF is not implemented with effective DSS.

The conclusion that derives from the aforementioned studies, is that current agricultural DSS lack effectiveness. However, it is highly crucial to state that effectiveness is not faced as unidimensional in each study. The results showed that user-centered design, adaptability, low cost and combination of different sources of information, are all parameters of effectiveness. That said, these elements are crucial for the present study, since the goal is to develop an effective DSS that is going to upgrade PF and be widely accepted by farmers.

Having outlined the opposing perspectives, the following research question has been formed: "How can an effective Decision Support System be developed, so it can advance Precision Farming?". This is the main objective that the research is called to achieve. To create a DSS, that covers all the essential and necessary aspects of effectiveness (as decided from already stated studies).

CHAPTER 3: LITERATURE REVIEW

The purpose of this chapter is to present previous studies, related to the field of Smart Agriculture. As a matter of fact, the common ground between the following academic works and the current research, is the development of a DSS for agricultural support. Nonetheless, a presentation and review of former studies, will offer a better understanding on the different methodological approaches, scientific tools and concepts of the agricultural DSS. Before presenting the studies, it is important to point out that the literature review is classified in two categories. These two categories emerged from patterns that have been identified, while reviewing pre-existing knowledge. The first category introduces studies that attempted to create effective DSS, but their focus of attention was exclusively on the tools that were used. The second category includes studies that generated effective DSS, but targeted specific agricultural goals.

First study to be presented, is the one by Li Tan (2016) and named "Cloud-based Decision Support and Automation for PA in Orchards". According to Tan (2016), even though PA is showing its potentials through better data-driven decision systems, it cannot reach any further development without better technology and improved tools. In this case, the researcher developed a framework for a Cloud-based DSS, that collects data from various sources, arranges specific decisions and controls various devices (deployed in the fields) from Cloud. The main purpose was to build a Cloud-based Decision Support and Automation System from ground, to boost and improve PA for specialty crops. As stated by Mell and Grance (2011),

> "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

Even though the purpose of Tan's paper is to form an efficient DSS like the present research, the tools that are used, are quite distinctive. In our case, Cloud computing is not the main priority and cannot serve as the sole mechanism of the DSS. The tools that are going to be used are not predetermined. If the tools are decided in advance, the effectiveness of such systems becomes narrowed.

Another similar study aimed to develop an effective DSS by fixating again on a specific tool. This time on Natural Language Processing (NLP). Prasad, Prasad and Kulkarni (2008), suggested the development of a DSS based exclusively on NLP. Before proceeding into further details, it is an obligation to define what NLP is. According to Goldberg (2017),

> "Natural language processing (NLP) is a collective term referring to automatic computational processing of human languages. This includes both algorithms that take human-produced text as input, and algorithms that produce natural looking text as outputs."

The scientists highlighted the role of NLP in implementing a DSS. As stated in their paper, DSS that incorporate limited vocabulary, fail to work smoothly, when they are exposed to realistic, complex situations with real-world ambiguity. Having that in mind, they proposed a DSS that incorporates NLP. They also listed a series of problems that NLP will solve (if incorporated in a DSS). These issues include segmentation, word sense disambiguation, speech acts and plans. As has been noted, there is a special attention to NLP and how can it be used as a tool. Nevertheless, this research provided some useful insights on how the DSS could be better developed in the future.

Moreover, some of the studies that deal with agriculture and DSS, shift their attention from the tools that are used, to the purpose that the DSS is built for. For example, Venkatalakshmi and Devi (2014), directed the development of a proposed DSS for seasonal crops. In other words, the central idea was to build a system that provides information about the expected yield in each season, with better accuracy. Seasonal crops, according to the scientists, are the grown crops after the harvest of the regular crops. The DSS that they came up with, backs PA, but settles only for seasonal farming. In their analysis, the DSS does take a variety of inputs (such as season, type of soil etc.) and through predictive analytics, the system decides which crop is better for planting and what are the needed fertilizers. Additional information that is given to the farmers, is a report about the maximum yield and revenue, based on the cost of the crops. In contrast to the present research, Venkatalakshmi and Devi (2014)'s study applies only on seasonal crops. However, one of the features that the present research considers as a cornerstone, is adaptability. Not just for seasonal crops but for all kinds of crops.

Finally, the scientific paper of Ascough II et al. (2001) deals with the creation of an integrated DSS for Sustainable Agriculture in the Great Plains. The Great Plains, as explained by the researchers, face limited agricultural production due to soil water and nutrient unavailability. The purpose of the study was to design a DSS that handles and supplements these resources, without causing any further damage to the surrounding environment. Furthermore, the plan provided not just crop but livestock management guidance as well. Therefore, the most obvious fact in this case, is the particularization of the DSS. The system is directed at the agricultural issues of the Great Plains, a geographical area with unique attributes and idiosyncrasy. On the contrary, the current research does not focus on a specific geomorphological area and considers flexibility as a main attribute of DSS effectiveness.

By reviewing previous research, we can identify the gaps in the respective literature. The purpose of this study is to address some of these gaps and weaknesses. Although this study will not attempt to develop an effective DSS in line with previous studies, which either focus on creating DDS by using specific tools or build DSS for precise reasons, yet it will deal with effectiveness as multidimensional. Effectiveness, in this study, is not prearranged by the tools or the agricultural purposes.

CHAPTER 4: THEORETICAL BACKGROUND

The current chapter is dedicated to the definition and description of the theoretical framework, on which the basic concept of the research is explored and understood. The following terms, theories and models not only give direction to the research, but also provide scientific justification for the current development of the DSS.

Smart Agriculture/e-Agriculture:

e-Agriculture or Smart Agriculture, as defined by the Food and Agriculture Organization of the United Nations (2005),

> "is an emerging field in the intersection of agricultural informatics, agricultural development and entrepreneurship, referring to agricultural services, technology dissemination, and information delivered or enhanced through the Internet and related technologies. More specifically, it involves the conceptualization, design, development, evaluation and application of new (innovative) ways to use existing or emerging information and communication technologies."

e-Agriculture goes beyond the consolidation of the fields of agriculture and Information and Communications Technologies (ICT). Based on Malavade and Akulwar (2016), the modern agricultural industry is being transformed by the usage of ICT, by helping farmers to compete with rising challenges. Different parameters, such as water management, crop monitoring, soil observation and livestock productivity are easily managed with the least possible efforts.

Resulting from all the previous information, it is understood that this research appertains to this field. The proposed system is promptly related to e-Agriculture, as it is a recommendation for creating better and more effective agricultural systems. Consequently, this work aims to improve agricultural DSS and revitalize PF.

Precision Farming/Agriculture:

PF or better known as PA, is a modernized farming management concept that uses technological and digital techniques to enhance agricultural production processes (Schrijver, 2016). In other words, PA intents to generate more agricultural output while using less input and taking into consideration the exact needs of crops and livestock. As Schrijver further clarifies, input can be water, energy, fertilizers, pesticides and more. PF is based upon a combination of technologies, all "borrowed" and supported by IoT. All these technologies are organized in a certain manner, to create an information management system (known as a DSS). Under all these circumstances, it is quite clear that PF is a way of approaching Smart Agriculture, that helps farmers and soils to work better. There is no way of achieving PF, if an effective DSS is not developed. In the current research, the proposed DSS is planned to maximize PF, by getting all the appropriate data and suggest the best solution for each case.

Decision Support System:

Power (2002), explains that a DSS is an interactive, computer-based application that analyzes raw data from different sources of information, to support decision-making activities. Moreover, Alter (1980), identifies three major characteristics of these systems. Firstly, the way that DSS are designed, is to assist decision processes. Also, a DSS's mission is to support rather than automate decision making. Finally, DSS should respond in a fast manner, to the needs of the users. As a conclusion, these systems help humans take a decision, relevant to a situation. The development of an effective DSS (as defined previously) is considered the cornerstone of this study. The DSS is the "engine" that will expand and boost PF.

IoT:

According to Rose, Eldridge and Lyman (2015), the term IoT describes a system in which physical objects could be connected to the Internet by sensors. This technology is often incorporated in sensors, systems and networked devices. All these products take advantage of this computing power and interconnection to offer new possibilities. Similarly, the GSMA Association (2014), defines IoT as the use of connected devices to leverage data, collected by installed sensors in physical objects. The term "thing" in the world of IoT can be anything. From embedded microchips in cattle, Internet-connected vehicles or biometric clothes, anything that can get an Internet Protocol (IP) address and be equipped with the ability to transfer data all over the Internet, belongs to the sphere of IoT. Part of the suggested DSS includes technological equipment that derives from IoT. For example, the sensors (which gather data from the environment), are a vital piece of the proposed DSS puzzle. They have their own unique IP address and they are part of a greater system that attempts to boost PF. In this case, IoT becomes part of the solution.

Long-Range Wide-Area Network (LoRaWAN):

One of the protocols that will be brought into service, is LoRaWAN. In accordance with LoRa Alliance (2015), LoRaWAN is a media access control protocol that is most suitable for wide area networks. It is intended for allowing intercommunication between low-powered devices and Internet-connected applications over wide range wireless connections. Furthermore, all boards and devices that implement the LoRaWAN technology (the Waspmote Agriculture Sensor Board is also included), operate in a low radio frequency spectrum and support bi-directional communication between them and a gateway. As LoRa Alliance furtherly specified, a common LoRaWAN network is consisted of gateways, end-devices and a central network server. Gateways form the bridge between the end-devices and the server. As far as the connection is concerned, gateways are connected to the network server with the help of standard IP connections. These connections can be achieved with high bandwidth networks like Wi-Fi, Cellular or Ethernet. At the same time, low power, single-hop, wireless networks, like LoRaWAN, are used by devices to connect and forward messages to any gateway.

Message Queuing Telemetry Transport (MQTT):

According to Durkop, Czybik & Jasperneite, (2015), MQTT is a lightweight, messaging protocol, designed to minimize network bandwidth, device resource demands but also establish and maximize accuracy, reliability and successful transmission of the data. Moreover, MQTT, as the researchers furtherly explain, uses a publish/subscribe model which requires the usage of a broker. All clients involved in this model can publish messages on a specific topic and subscribe to topics, to receive the broadcasts. As seen in the figure below, the MQTT broker is responsible for receiving the messages, filtering them based on topic and distributing them among the subscribers (if they subscribed to the specific topic, on the same broker). In addition, they highlight the fact that in this kind of model, the publishers and the subscribers connect indirectly through the broker. Lastly, as stated by the scientists, due to its characteristics, MQTT is increasingly becoming one of the primary protocols for IoT deployments. For the purposes of this study, MQTT will be mainly used for transmitting and receiving data from the ThingSpeak platform.

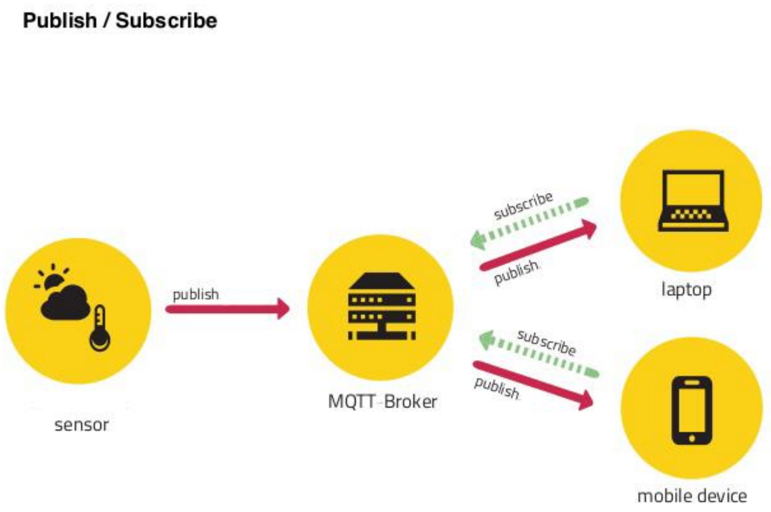

Figure 1: MQTT Architecture Diagram

Android:

In terms of software, there are two components envisaged: the first involves device programming which is described separately and the second focuses on end-user device programming which will be based on the Android operating system. Lee (2012), explains that Android is a mobile operating system and not a device or a product (like some people tend to confuse it). It is open source and Linux-based, that offers a unified approach to application development (thus its main advantage). Moreover, Android applications are coded in the Java programming language, using the Software Development Kit provided by Android Studio. The apps, after their creation, can be easily packed and downloaded from an online store (usually Google Play). For the needs of this study, an Android application will be created, to display the most prominent decision to the user.

ThingSpeak:

One of the most important units in the present study, which will assist to the storage of the data, is the ThingSpeak platform. According to Gomez Maureira, Oldenhof & Teernstra (2014), ThingSpeak is an IoT platform that grants the collection and storage of real-time sensor data to its cloud. The data can be stored or retrieved from the platform by using the Hypertext Transfer Protocol (HTTP) or MQTT protocol. Simultaneously, the authors mention that the ThingSpeak platform has integrated supportive tools from the Matrix Laboratory environment, that allows the monitoring, analyzing and visualization of the uploaded data. Finally, as the researchers conclude, this platform permits and authorizes the creation of sensor logging, location tracking apps and supports the formation of a social network of IoT channels with status updates.

Waspmote Agriculture Sensor Board:

A hardware component that is essential to the development of this project, is the Waspmote Agriculture Sensor Board. According to the manufacturer Libelium (2017), this board allows the monitoring of multiple environmental parameters. This board (as seen below) includes all the necessary sockets, so it can establish a connection with various agricultural sensors. The most typical sensors include devices that measure air and soil temperature, humidity, solar radiation, rainfall and leaf wetness.

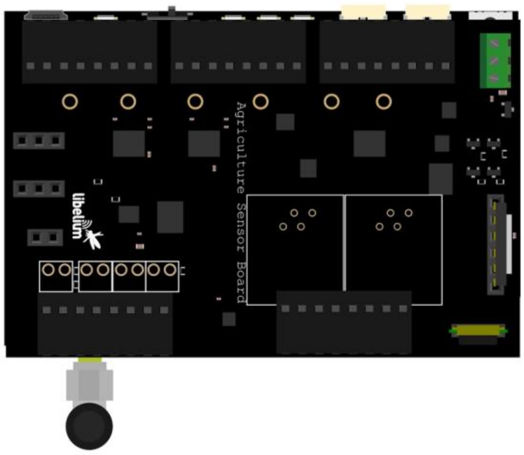

Figure 2: Waspmote Agriculture Sensor Board

CHAPTER 5: METHODOLOGY

For answering the previously stated research question, a specific methodology is adopted, which is quantitative and is primarily based on an experiment. The experiment is conducted to identify the best possible solution for the issue that has been raised. In this case, the buildout of an effective DSS. The following paragraphs give a rundown of the three main compartments that comprise the methodology. The research tools that are used for resolving the research question, the sample that this study addresses and the research process.

Research Tools:

A variety of tools is required to cover the hardware and software aspects of the DSS. The required hardware equipment includes the Agriculture Sensor Board, the Waspmote Board, sensors that take as an input different sources of information and the essential cables to connect the system. In terms of software, the compulsory programs are the following. The Waspmote Integrated Development Environment (IDE), the Android Studio, the NetBeans IDE and the ThingSpeak platform. The usage and role of each tool will be explained in the following chapters when the research finally enters its construction phase.

Sample:

Farmers and agricultural advisors consist the sample, that the final product is going to address. According to a study that was conducted by Nguyen, Wegener and Russell (2005) in Australia, regarding the usage of DSS in agriculture, the most prominent conclusions revealed a specific pattern. DSS are poorly adapted by farmers themselves. On the contrary, most farmers have an indirect contact with these systems, through agricultural advisors and farm consultants. As stated, farmers usually hire agricultural consultants to assist them with farm strategies, plans and the software that comes with it. The research concluded that farmers simply do not have that much time to operate and master DSS.

As a result, both farmers and farm consultants compose the sample of this research. The only limitation that must be taken into consideration is the following: only farmers (and farm consultants as well) who are willing to implement the PF approach, comprise the sample. There is no use in answering the research question, if the sample includes farmers who dismiss and reject the PA method.

Procedure:

For developing a DSS, a strategic method must be chosen from the beginning. For the current research, the iterative waterfall model is adopted. As described by Royce (1970), this model has six sequential phases and works perfectly with small and medium scale projects. However, for the objectives of the present study, the last two stages that deal with product launch and maintenance will be omitted. As aforesaid, Royce describes the four stages as following.

Firstly, the requirement analysis. In this stage, all possible requirements and criteria are captured with the intention of making a consistent system. Furthermore, all possible dangers and risks need to be taken into consideration. In the same fashion, all requirements must be examined for the DSS and consider all probabilities and threats.

The second stage refers to the system design. The design of the system is developed, after all specifications, from the previous step, are covered. Under the same circumstances, software and hardware requirements will be specified, thus arranging the overall structure of the DSS.

Thirdly, the system enters its developmental phase, by assembling the hardware and writing the code. As the guidelines dictate, the DSS must be formed in this stage, combining and incorporating the hardware and the software.

Lastly, after all the previous steps are completed, it is time to test the system and confirm that it operates as expected. It is an obligation to record all results and address anomalies or malfunctions.

Although this approach offers a quite stable and firm product, the whole process is fixed. All phases are cascaded to each other, creating a linear procedure where overlaps are not permitted. Nevertheless, as Royce points out, if an iterative interaction is added in each phase, the project will be more likely to succeed. Additionally, by allowing revisions and reflections during the process, risk and uncertainty drop dramatically and changes can be accommodated. That's why (as shown in the figure below), iterations are allowed in each step, so flaws can be forestalled without violating time constraints.

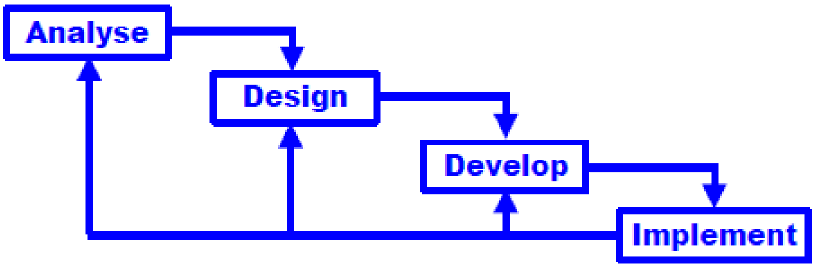

Figure 3: Iterative Waterfall Model

CHAPTER 6: TIMELINE

Undoubtedly, a specific timeline must be followed to accomplish the development of an effective DSS. The following diagram depicts the necessary goals that need to be achieved in each month. Particularly, in the first month of 2018, the goal is to write down the specific requirements for the DSS and analyze the specifications of the system. Note that this procedure is quite iterative and circular, until all the criteria are met. In February, the design of the DSS is going to be the focal point of the procedure. In this stage, the DSS must be formed theoretically, including all the input sensors, the processing algorithm and the output device. As a result, in March, the procedure will focus on the development and the construction of the proposed DSS. The proper operation of the system is vital and pivotal. In the next month, the DSS must be put in the testing phase, to verify its proper function and correspondence. Finally, the succeeding month will be dedicated to the research results. Objectives that were accomplished or missed are going to be addressed, the function of the system compared to the initial goal must be recorded and findings need to be summarized. Admittedly, this part is pressing, because these conclusions must be incorporated into the final report. The oral defense of the thesis and its submission will follow later that month.

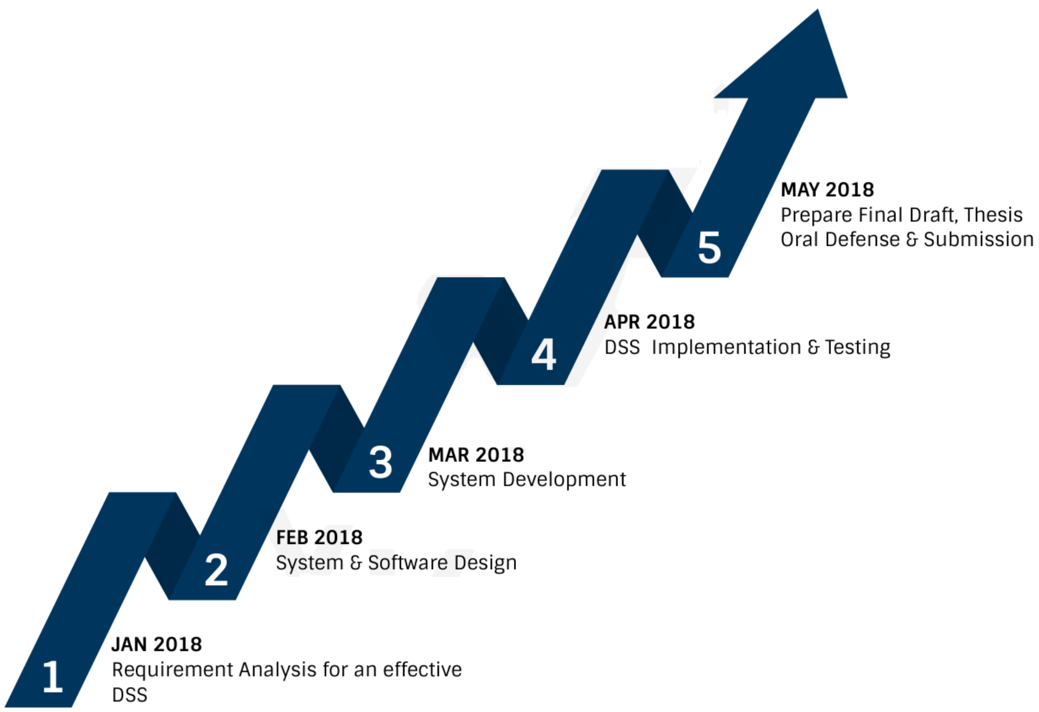

Figure 4: Research Timeline

CHAPTER 7: REQUIREMENT ANALYSIS

As previously indicated, the first stage that needs to be covered is the requirement analysis. In this phase, it is highly important to figure out and recognize the strengths and the weaknesses of the proposed DSS. Additionally, it is equally imperative to accentuate the opportunities that may arise during the development and the implementation of the system. Finally, the detection of possible menaces and risks that may occur and pose a threat to the project must be addressed and analyzed.

Therefore, in order to develop an adequate, satisfactory and sufficient awareness of the situation which is going to contribute to the strategic planning and decision-making of the proposed DSS, a technique must be followed. In this case, the Strengths, Weaknesses, Opportunities, Threats (SWOT) structured planning method allows us to identify, organize and evaluate these four aspects and elements, regarding the planned system. As reported by Humphrey (1960), SWOT analysis is an analytical framework that is used to regulate, determine and define several, decisive components of a company, organization, department or even (in our case) an individual project.

However, before breaking down the SWOT analysis process into the four main elements, it is of paramount importance to point out the following remarks, as stated by Humphrey himself. A SWOT analysis does not rely on objective assessments and scientific evaluations. On the contrary, a SWOT analysis process is strongly based on two abilities. On the skills of detecting and understanding internal and external causes that may have a strong affect and impact on a project. In other words, a search involving the SWOT framework is a detailed examination of various factors, that leads to a proper preparation and precaution, rather than trying to make meticulous and conscientious predictions.

Initially, the first element of the SWOT analysis that needs to be examined is Strengths. Strengths, as defined by Ommani (2010), are the required attributes, that are decisive for the desired success of the project. In this present proposal, strengths derive from the actual purpose per se. The first characteristic that can be considered as an advantageous resource is the user-centered design. Particularly, the application that is going to provide the farmer with the best possible resolution, will be developed based on key principles of user-centered design. Specifically, Virkus et al. (2016), issued a mobile developer’s guide which covers all the necessary aspects, attributes and characteristics about application development (especially on portable devices). By taking into consideration the guidelines, the final application will offer the best overall user experience, thus, providing robustness to the current study.

Secondly, adaptability and flexibility are regarded as dynamic attributes of the suggested system. The recommended DSS is not limited nor fixed by certain particularities. The DSS that is brought forward, considers adaptability as a cornerstone, not a restriction. Attributes, such as the weather, the soil and the agricultural output, are not treated as peculiarities in any way. The propounded DSS is not only pliable and malleable to the needs of the different users, but it also mandates consistent commands to the various, internal components of the system (always in respect to the context and environment changes). Moreover, the procurement of data by different sources of information, is an additional advantage. One of the prime objectives of the current research is to acquire data by a combination of resources, such as soil, temperature, atmospheric pressure, humidity, etc. Consequently, the farmer will be provided with a wealth of information and will have access to explicit knowledge and decision-making capabilities, that were not achievable before.

Lastly, as far as the financial aspect of the system is concerned, the development of the DSS, especially the hardware component, is based on affordable, low-priced equipment. Instead of investing on costly and overpriced apparatus, the present proposal's mission is to create an inexpensive, accessible PA system. For example, the software that is going to be created, is predominantly available on the World Wide Web. Furthermore, the hardware is provided and offered by the Department of Communication and Internet Studies at the Cyprus University of Technology. Nevertheless, if the suggested DSS is produced on a larger-scale, then the total cost needs to be taken into consideration.

Subsequently, another point that must be examined is Weaknesses. Osita, Onyebuchi & Nzekwe (2014), describe Weaknesses as factors that could prevent successful, prosperous results and outcomes or derail negatively a project, long before it is even initiated. The first possible weakness that may cause harm, is the essential necessity of owning a portable device. One of the most fundamental compartments that constitute and comprise the total system, is the application. The application is needed for presenting the captured, formulated information and prompting the user to take an appropriate decision. If the user lacks the possession of such device, then it is not possible to grant access to the collected data. Additionally, a second vulnerability is the concept of PF itself. If farmers are not interested in PA and they are not willing to embrace the emerging PA technologies that are being developed, then this aspect poses a threat to the whole project. The sample of the current study, as it was specified in the previous chapter, needs to invest in PF.

When it comes to opportunities, Ommani (2010), characterizes them as advantageous, helpful, external elements that function in a favorable way to achieve the purposes of a project. These kinds of constituents can positively position a product or a business on a higher level. For instance, an opportunity that seems to arise is the support and aid that comes from the European Union. Specifically, as declared by the European Commission (2017), a Common Agricultural Policy has been initiated and activated until the year of 2020. One of the objectives that this policy has listed, is to support and promote farmers' ability to invest in PF technologies. The suggested DSS could have the opportunity of getting funded, since the European Commission has shown keen interest in fostering and promoting innovative PA technologies.

Finally, Hovardas (2014), names all the external components that could cause gravely negative consequences to the overall success of the project, as threats. Likewise, the most prominent threat in the development of the current DSS is the emergence of new technologies. Specifically, the evolution of technologies, such as the "Cloud", threatens to undermine programs like this one. As seen in the chapter of "Literature Review", the study that was conducted by Li Tan in 2016, concluded that PA can reach its full potential by using improved and enhanced technologies. In his case, the framework which he developed was based on Cloud. Resulting from the previously expressed assertions, it is clearly understood that a watchful eye must be kept on emerging technologies, to keep the system as competitive as possible. The proposed DSS must integrate and consolidate any breakthrough technological innovations that can help it achieve maximum potential and competence.

In summation, the requirement analysis, which was carried out with the help of the SWOT matrix, aided in capturing all the significant principles of the proposed system. The internal strengths and weaknesses of the DSS have been perceived. Meanwhile, the external opportunities and threats have been identified, hence, the recognition of points that may benefit or harm the project. The already stated four elements encapsulate all the necessary information, enabling us to move on to the next phase called "System Design".

CHAPTER 8: SYSTEM DESIGN

Detailed Description:

As previously mentioned, the expected outcome of this paper is the development of a DSS that applies to the field of agriculture and particularly the farming sector. In this chapter, the overall development of the system is examined, while going through the different key elements, including the hardware and the software. The proper functioning of the system depends on two central activities, namely the acquisition and transmission of the obtained data, but also the storage and the presentation of the formulated data. Moreover, this subchapter will address all four dimensions of effectiveness, as specified and recognized in prior chapters and illustrate how the DSS is going to cover them through the detailed design of the general system.

Initially, the first conception that needs to be discussed is the powering of the DSS. Without electrical power, the system is incapable of operating or functioning properly. Consequently, two pieces of hardware are mandatory for powering up the system. These components include a solar panel and a set of rechargeable batteries. Since the DSS (the hardware aspect of it to be precise) will be deployed in the farming fields, it needs to reach a full electrical autonomy, without requiring power plugs and sockets for connecting to a primary alternating current power supply or a constant change of batteries over time. Libelium (2017), declares that a Waspmote Agriculture Sensor Board can use rechargeable batteries as a power source if a solar panel is connected to the panel port of the board. In this manner, the case of supplying the board with power will be resolved.

Additionally, the development of a substructure must be achieved, to accomplish the procurement and transferal of the data. However, it is highly important to specify the frequency of capturing the data, before even defining which data must be collected. According to a survey on agricultural production methods, conducted by the European Union in 2010, the frequency of data collection must be based on a regular pace. Moreover, the survey underlined the significance of eliminating and diminishing irregular intervals between data measurements. As a result, for the purposes of the present study, the recording and filtering of data every 15 minutes was determined, to investigate agricultural weather patterns.

Subsequently, to enact the purpose of the current dissertation, the sources of information ought to be determined. However, it is crucial to keep in mind that one of the aspects of DSS effectiveness is the fusion and consolidation of multiple sources of information. In this case, as stated by Ganesan (2005), the agricultural production has evolved into a convoluted field that obligates the acquisition of knowledge and information from various diverse resources. Therefore, by ensuring a variety of sources of information, this facet of effectiveness will be covered. Resulting from all the above, eight types of data will act as input information for the DSS. The eight sources of information are the following: temperature, humidity, atmospheric pressure, soil humidity, leaf wetness, wind speed, wind direction and precipitation. In this manner, the current study ensures that the collected knowledge is not only agricultural, but weather based as well.

The next step, regarding the second part of the first activity (meaning the transmission of the data), a connection must be established between the concerned parts. For this reason, LoRaWAN will be brought into service. Instead of choosing to implement a direct connection from the Waspmote Agriculture Sensor Board to a database or a server with the help of a high bandwidth network, LoRaWAN was chosen. The reason why LoRaWAN was preferred, was due to the fact that the Waspmote Agriculture Sensor Board already integrates this technology. If any other technology was chosen (e.g. Wi-Fi, 3G etc.) for establishing a connection, then the purchase of additional modules and microcontrollers would be inevitable, thus, increasing significantly the total cost of the project.

Furthermore, by using LoRaWAN, a connection between Waspmote Boards can be established and entrenched. As stated by Libelium (2017), a direct communication can be authorized between Waspmote Boards by enabling the peer to peer mode. This mode allows the exchange of data between multiple boards by using the radio transceiver, hence, the installation, creation and set up of a network made up of Waspmote Boards that transmit the information that was collected by the sensors. In such manner, not only the data broadcasting action is resolved, but also the adaptability of the system is settled. Through the instrumentality of LoRaWAN, the DSS can adjust to different types of conditions, without having the need to utilize supplementary technologies nor being dependent on alternative communication protocols.

Moreover, a gateway is imperative for concluding the transmission process. Zachariah et al. (2015), define gateways as hardware devices and network nodes that are used for joining two dissimilar networks, so the devices of one network can communicate with the devices of another network. For the thesis, a gateway is significant for receiving the sensor readings and redirecting them to a server for storage. As a result, the inauguration of a connection between the Waspmote Board and a gateway is established through LoRaWAN. Any gateway within reach of the Waspmote device will receive the sensor readings and redirect them to a database.

At this point, the focus of attention is being shifted from the transmission of the data to their storage, thus, introducing the first part of the second activity, billed as data repository. For the present study, the ThingSpeak platform was chosen for saving and retaining the information. As reported by Acharya (2016), the ThingSpeak platform is an open source, decentralized, crowdsourced IoT application and Application Programming Interface (API) that allows various devices (including gateways) to establish a connection and grants the persistent storage and retrieval of data by using the HTTP or MQTT protocol. Moreover, ThingSpeak supports Representational State Transfer calls over HTTP and MQTT methods over MQTT, to update or retrieve data (Nakhuva & Champaneria, 2015). In this way, all captured sensor data will be collected and saved for the objectives of the project. As claimed by Fysarakis et al. (2016), the MQTT protocol is ideally suited for IoT projects, since the bandwidth usage, response time, throughput rate and intermittent connectivity are highly considered for accomplishing such projects. For the above-mentioned reasons, MQTT was chosen for transmitting the data from a gateway to the ThingSpeak platform.

Moving on to the latter part of the second activity, it is compelling to point out the process of the data procurement, before even mentioning the presentation of the information to the user.

The information that is stored in the ThingSpeak database must be conveyed to an application, where all the necessary process will occur. Being consistent with what was stated, ThingSpeak supports both the HTTP and MQTT protocol. Nakhuva and Champaneria, (2015), suggest that both protocols are beneficial for retrieving historical data within a strictly defined and determined time range from a ThingSpeak channel. Moreover, one of the services that this project is going to rely on for formulating the data, is the OpenWeatherMap. Specifically, the purpose of employing the OpenWeatherMap online service is to aid in the provision of weather forecast, accurate meteorological data and assistance in the decision-making process.

Following the retrieval of the data from the database, the formulation and information presentation comes into play as part of the second activity. As previously said, the application will be developed for the Android Operating System. According to Bala, Sharma & Kaur (2015), the Android Operating System is the most popular and widely used smartphone platform and software of choice. For this prime reason this platform was chosen. The final application will retrieve data from the ThingSpeak service and the OpenWeatherMap, process them and suggest the appropriate decision to the user. The app will be named and called "AgriWeather", since it is addressing this topic. For the needs of the current research, it was previously mentioned that the application needs to be as user friendly as possible. In other words, a simple design and a lightweight structure must be used. Similarly, the navigation through the app needs to be straightforward, uncomplicated and smooth, without excessive and unreasonable information that may trigger defects or cause malfunctions.

As Virkus et al. (2016) suggest, the data presented in an application should have a continuous flow, without being scattered and distributed all over the screen. A top-down approach should be adopted in demonstrating the information. The simple information should be presented at the top of the page, while the complicated material should be positioned and located right after. Moreover, as the designers claim, the buttons should be consistently positioned at the bottom of the screen, after all the data presentation is completed. In this manner, the user can assume control of the perceived information conveniently.

Correspondingly, the initial application screen will feature the logo and two buttons. The first button will provoke the call of a page where all the knowledge about the weather will be allocated, while the second button will lead to a page where all the necessary information needs to be registered. However, the following point must not be overlooked. The first screen containing the weather and agricultural data cannot show anything until all the required information is filled. It must be reminded that adaptability is one the central attributes that this study deals with. Under those circumstances, the versatility of the DSS cannot be fulfilled, if the system cannot get weather forecast information about the exact place that it has been employed.

With this intention in mind, the user must fill in four different types of information to initiate the process. The first piece of information that the user must give is the type of crops that are planted to the farmland. The second vital piece of information is the location. To get the appropriate forecast from the OpenWeatherMap service, the user must specify the exact location that the DSS is settled. The altitude of the cultivated land is the third information that the user must register. Along with the location of the field, the elevation of the land is an important parameter for making the appropriate predictions. Similarly, the fourth and final data that must be filled in, is the number of days that the crop or plant can endure and survive without water. The following paragraphs justify and defend the selection and choice of the above-mentioned parameters. Before moving on, it is pressing to point out that this action takes place only once and not each time the application is launched. At the same time, the user may change the information of the system, according to the place that the DSS has been deployed.

Thereafter, as soon as the user completes the filing of the appropriate data and clicks the confirmation button, the application will automatically redirect the user back to the initial screen. Provided that all the required information is registered successfully, the first button that leads to the formulated data will be available for clicking. By choosing this button, the computed info will be demonstrated to the user. Yet a precise description should be given, not just for the presentation of the data but also about their calculation. To begin with, acting in accordance with the proposed design guidance, this screen should present all the information in a continuous, constant manner and the plain information should be presented first. As a result, simple information regarding the last recorded temperature, wind direction, wind speed and amount of precipitation needs to be introduced at the top of the screen.

However, as far as the wind speed is concerned, a point must be addressed. Instead of merely demonstrating the wind velocity, this info could be broken down into two sections. The first section introducing the speed of the wind and the second part offering a further description of the wind status. According to Nakhuva and Champaneria (2015), the wind speed can be classified into the following 13 categories. Initially, if the speed is lower than one kilometer per hour (km/h), the wind can be classified as calm. Secondly, if the wind speed is in the range of one and less than six, it can be categorized as light air. Thirdly, if the speed is more than six and less than twelve, it can be sorted as light breeze. The wind is classified as gentle breeze if the speed exceeds 12 km/h and less than 20. Moreover, if the wind surpasses the 20 km/h mark, is classified as moderate breeze and if the 29 km/h is passed, then the wind is recognized as a fresh breeze. The wind can be characterized as a strong breeze, if the speed is located between the range of 39 and 49 km/h. Next, winds are classified as near gales (49 to 60 km/h), gales (61 to 73 km/h) and strong gales (74 up to 87 km/h). If the wind gets caught speeding in excess of 88 km/h, it can be officially categorized as a storm. However, if the speed is more than 102 km/h, then the wind is classified as a strong storm. Finally, the wind is considered a hurricane, if it exceeds the speed of 117 km/h.

After completing the wind speed sector, it is high time the atmospheric pressure was addressed. As stated by Clitheroe (2010), plants and crops transpire three times faster and lose water at a higher rate, if the temperature exceeds 30 degrees Celsius and most importantly, the atmospheric pressure is lower than normal. Meanwhile, as the author furtherly suggests, the atmospheric pressure is inversely proportional to the altitude. The value of normal atmospheric pressure at sea level differs from the value of normal atmospheric pressure at a higher level. As a result, the user is asked to give the altitude of the farmland (as seen previously), to calculate if the barometric pressure is higher or lower than expected. Resulting from all the above, the system will take into consideration not only the barometric pressure but the temperature as well, to perform this equation. First, if the current atmospheric pressure is higher than the expected, the system will notify the user that this is happening. Second, if the value of the atmospheric pressure is the expected one, the system will print out that the pressure in the atmosphere is normal. However, if the pressure is lower than expected and the temperature is 30 degrees Celsius and higher, the system will immediately acknowledge that the plants are going to experience an increased water loss. Finally, if the pressure is lower than expected but the temperature is lower than 30 degrees Celsius, the system will declare that the atmospheric pressure is just lower than normal.

The soil status is also vital and critical when it comes to agricultural decision making. As claimed by Libelium (2017), a watermark sensor provides farmers with better understanding of how rapid the soil water is being depleted and drained in different parts of their field. Furthermore, Libelium issued a chart depicting the output frequency of the adaptation circuit in function of the soil water tension. This chart, in conjunction with a table that was released by WatchDog in 2015 regarding the interpretation of the Watermark sensor readings, assisted in creating the following outputs.

Specifically, there are six possible outcomes of this reading. If the Watermark sensor shows a value of zero (or less than zero), a message will appear, noting the user that the soil is dangerously dry for production. However, if the Watermark value is less than 500 Hertz (Hz), the soil (especially if it is heavy clay) needs irrigation. Moreover, if the value is more than 500 and less than 1000 Hz, the soil still needs irrigation but this time except the heavy clay type. In case the value is more than 1000 and less than 5000 Hz, the soil is adequately wet, except for coarse sands which start losing water. If the sensor value depicts more than 5000 Hz, the soil is saturated and if the value surpasses the mark of 7000 Hz, it means that the soil is oversaturated.

The crops' health is one of the parameters that the DSS could address as well. Rowlandson et al. (2015), define leaf wetness as the presence of liquid (primarily water) on the surface of a crop canopy. As the researchers furtherly claim, leaf wetness can be associated with the infection of plants and crops since the constant occurrence of free water on the plant canopy along with high temperatures trigger the development of bacterial, fungal and oomycete diseases. Specifically, a guide was publicized by the scientists, depicting this relationship. If leaf wetness is recorded for more than ten hours while the average temperature is 30 degrees Celsius and more, then the development of crops and plants diseases is certain. So, for the purposes of the research the following calculation was formulated. If leaf wetness is recorded for more than ten hours, the system will immediately check if the average temperature during this period of time exceeds the mark of 30 degrees Celsius. If it does, then the system will alert the user that the crops need health assessment. If it doesn't, the user will be notified that the crops are not facing any health issues yet.

Transpiration is an additional issue that should be also mentioned. The United States Department of Agricultural Forest Service (1992), mentions in one of its published handbooks that humidity and temperature play an important role when it comes to transpiration. One of the points that the department emphasized is that humidity and temperature are independent measures but when it comes to crop growth, they correlate to one another. The plants need to transpire at an optimum rate in order to provide the best harvest results possible. If the values of temperature and humidity do not surpass a certain threshold, the transpiration circumstances are ideal (meaning that the crop is not losing water, nor it is drowning). The chart depicted below, as conceived by TAPROOT (2017), displays this correlation. For the objectives of the project, the system will read both the humidity and temperature values. If the temperature and humidity values are between the proposed and suggested range, the system will print out that the transpiration circumstances are ideal, else the DSS will notify that the transpiration circumstances are not ideal, prompting the user to act.

		Relative Humidity													
°C	°F	100%	95%	90%	85%	80%	75%	70%	65%	60%	55%	50%	45%	40%	35%
15	59	0.0	0.8	1.7	2.5	3.4	4.2	5.1	5.9	6.8	7.6	8.5	9.4	10.2	11.1
16	60.8	0.0	0.9	1.8	2.8	3.7	4.6	5.5	6.4	7.3	8.2	9.1	10.0	10.9	11.8
17	62.6	0.0	1.0	2.0	2.9	3.9	4.9	5.8	6.8	7.8	8.8	9.7	10.6	11.6	12.6
18	64.4	0.0	1.0	2.0	3.1	4.1	5.1	6.2	7.2	8.2	9.3	10.3	11.3	12.4	13.4
19	66.2	0.0	1.1	2.2	3.3	4.4	5.5	6.6	7.7	8.8	9.9	11.0	12.1	13.2	14.3
20	68	0.0	1.2	2.4	3.5	4.7	5.9	7.0	8.2	9.4	10.6	11.7	12.8	14.0	15.2
21	69.8	0.0	1.2	2.4	3.7	4.9	6.2	7.4	8.6	9.9	11.1	12.4	13.7	14.9	16.1
22	71.6	0.0	1.3	2.6	3.9	5.3	6.6	7.9	9.2	10.5	11.9	13.2	14.5	15.8	17.2
23	73.4	0.0	1.4	2.8	4.2	5.6	7.0	8.5	9.9	11.3	12.7	14.1	15.4	16.8	18.2
24	75.2	0.0	1.5	3.0	4.5	5.9	7.4	8.9	10.4	11.9	13.4	14.9	16.4	17.9	19.4
25	77	0.0	1.6	3.2	4.8	6.4	8.0	9.5	11.1	12.7	14.3	15.9	17.4	19.0	20.5
26	78.8	0.0	1.7	3.4	5.1	6.7	8.4	10.1	11.8	13.4	15.1	16.8	18.4	20.1	21.8
27	80.6	0.0	1.8	3.5	5.3	7.1	8.9	10.7	12.4	14.2	16.0	17.8	19.6	21.3	23.1
28	82.4	0.0	1.9	3.8	5.7	7.6	9.5	11.4	13.3	15.1	17.0	18.9	20.7	22.6	24.5
29	84.2	0.0	2.0	4.0	6.0	8.0	10.0	12.0	14.0	16.0	18.0	20.0	22.1	24.1	26.1
30	86	0.0	2.1	4.2	6.4	8.5	10.6	12.7	14.8	17.0	19.1	21.2	23.3	25.4	27.5
31	87.8	0.0	2.2	4.5	6.7	9.0	11.2	13.4	15.7	17.9	20.2	22.4	24.6	26.9	29.1
32	89.6	0.0	2.4	4.7	7.1	9.5	11.9	14.2	16.6	19.0	21.3	23.7	26.1	28.4	30.8
33	91.4	0.0	2.5	5.0	7.5	10.0	12.5	15.0	17.6	20.1	22.6	25.1	27.6	30.1	32.6
34	93.2	0.0	2.7	5.3	8.0	10.6	13.3	15.9	18.6	21.2	23.9	26.5	29.2	31.8	34.5
35	95	0.0	2.8	5.6	8.4	11.2	14.0	16.8	19.6	22.4	25.2	28.0	30.8	33.61	36.4

Figure 5: Transpiration Chart

The last parameter that will be calculated has to do with precipitation. In order to notify the user if rainfall is expected, two elements must be put into service. The first factor is how many days the crop can get through without water (as previously examined). The number of days will be saved in a variable and the system will immediately check if rainfall is expected in the next days. The weather forecast will be provided by the OpenWeatherMap service and the system will respond accordingly back to the user.

Finally, the bottom screen could make room for the introduction of two buttons that lead to different screens. The first button leading to a screen that depicts a line graph which represents the amount of precipitation over time. Similarly, the second button, if chosen, leading to a secondary screen that likewise illustrates a line graph. In this case, this graph depicts the recorded temperature over the course of time. Both screens will represent the data in such a way, that users can gain insights and attain clearer and more distinct perceptions about the changing temperature variations, the weather conditions that affect their farmlands and give them the opportunity to adapt to the new climate changes.

The user-centered design, which was declared as the third element of effectiveness, is covered with design principles and outlines, as specified above. The design of the mobile application, from its conception through its completion, is based on precise instructions, rules and directions, all based on user needs and requirements.

Overall, the general structure of the DSS will follow and reflect the guidelines, as specified above. Nevertheless, through the particularized and extensive description and explanation of the system, all elements are rigorously determined, without allowing nor leaving any room for doubts or questioning. Moreover, by concluding the precise demands of the system and the needed pieces, the overall cost becomes reducible in addition to being manageable and controllable. Furthermore, most of the software programs are openly and freely available on the World Wide Web, thus, saving the research from unnecessary costs. Consequently, the final aspect of effectiveness regarding the financial cost is dealt with the preceding arguments.

Structure and Architecture:

The overall DSS, as aforementioned, is comprised of two main activities. The first activity includes the collection of the data and their delivery to the ThingSpeak platform. The second activity covers all the necessary actions for depositing the data and presenting them to the user as soon as the formulating process is fulfilled. The following figure depicts the general structure of the system, along with the sequence of the activities.

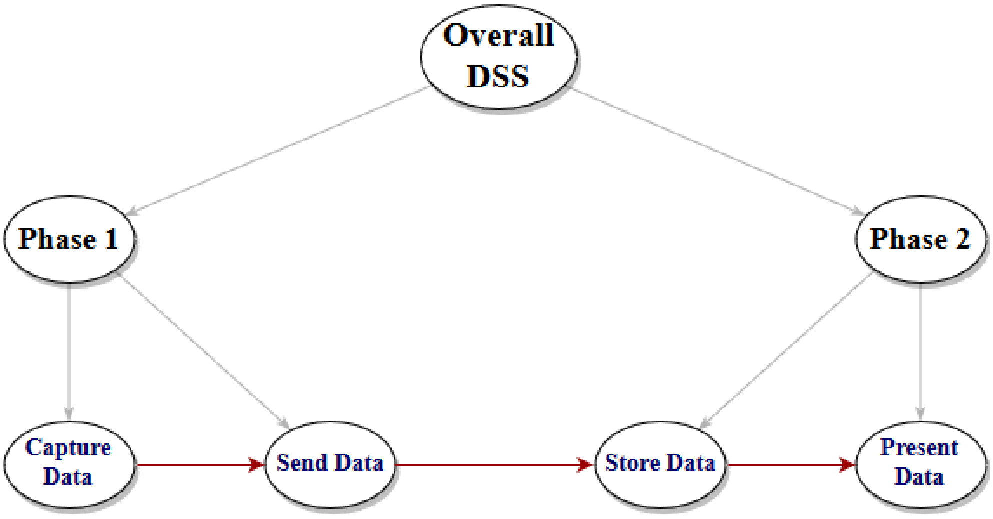

Figure 6: General structure of the DSS

It is worth noting that both activities are intertwined. As soon as the first activity is completed, the second activity is initiated, with the final purpose of informing and inciting the user to take the appropriate action for the growing crops. Without breaking down the activities to their fundamental phases, it would not only be intractable but also impractical to manage and complete the research. As depicted in the figure underneath, the architecture of the DSS describes in detail the technologies, protocols and devices that were deployed for the prerequisites of the present study.

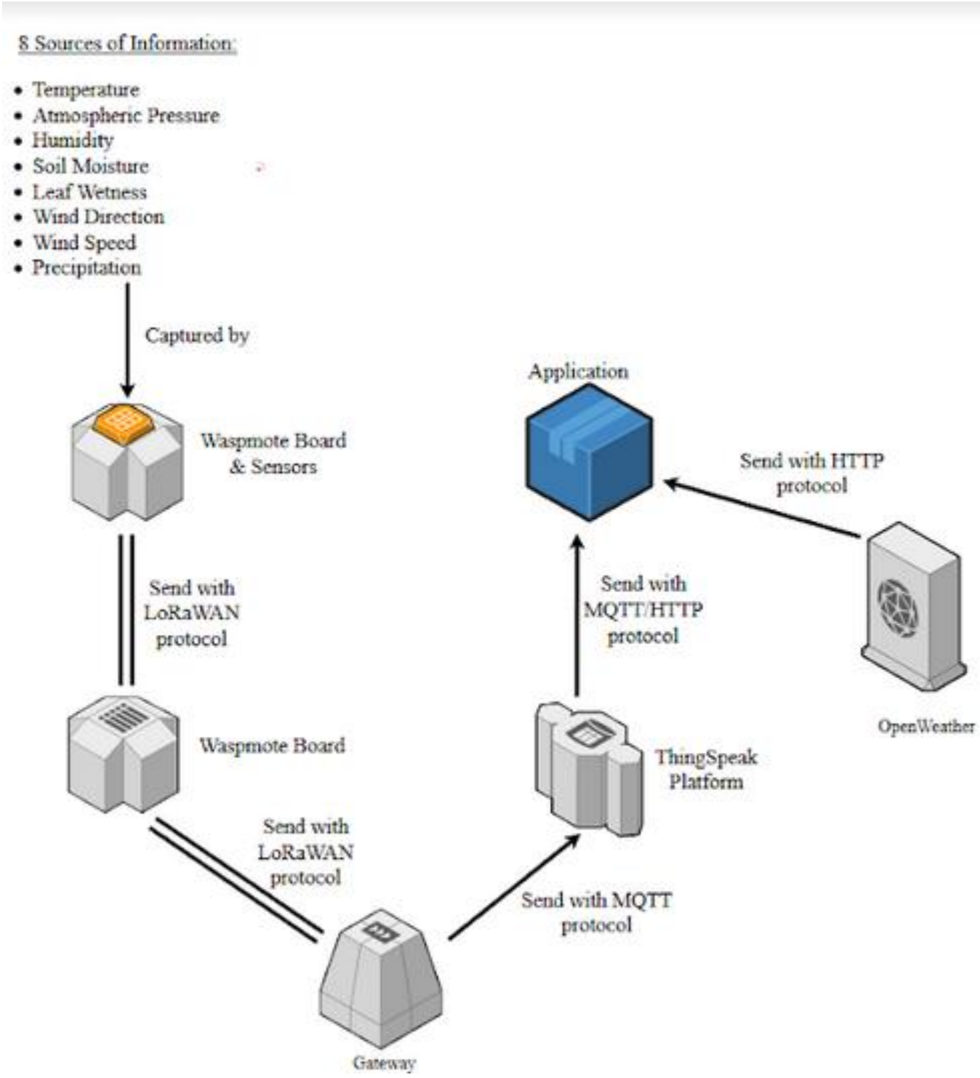

Figure 7: Architecture of the DSS

<u>Hardware:</u>

For the acquisition of the necessary data, a single-board microcontroller along with specific sensors will be utilized. Specifically, the Waspmote Agriculture Sensor Board has all the appropriate sockets, so all the relevant sensors can be connected. As far as the sensors are concerned, **four** agriculture-oriented and weather devices will be used. All the sensors are designed and manufactured by Libelium, a company that produces hardware and software developmental kit for wireless sensor networks (Asin, 2017). Libelium (2017), gave an explicit description on how the sensors perform and operate. One of the sensors that this study will employ, is the Watermark sensor. As the company reports, this sensor measures the moisture of the soil and returns the frequency output in Hz. The second sensor is the Leaf Wetness Sensor, which reads and returns the percentage of condensation on leaf surfaces, as furtherly explained by the company. The third sensor that will be utilized is the Temperature, Humidity and Atmospheric Pressure sensor (also known as BME280). On the authority of Libelium, the BME280 sensor is responsible for measuring the temperature in Celsius, the barometric pressure in Pascal unit and the percentage of humidity in the atmosphere, thus, covering all the necessary atmospheric conditions.

The fourth sensor is the Weather Station (WS-3000). As furtherly indicated by Libelium, this station is consisted of three different sensors, namely, a wind vane, an anemometer and a pluviometer. The wind vane turns freely on a platform and can distinguish up to 16 different wind directions. Moreover, as the company designates, the anemometer returns the value of the wind speed in km/h. Specifically, its output is a digital signal whose frequency is commensurate and proportional to the wind speed. Lastly, the pluviometer is necessary for calculating the precipitation in millimeters (mm). The sensor functions in a particular manner, as pointed out by Libelium. The pluviometer sensor comprises a small bucket and an internal switch. Once the bucket is filled (an approximate value of 0.28 mm of water), the switch closes, thus, causing an interruption, emptying the bucket and sending the value to the board.

Besides the sensors, there are four more hardware components of equal importance. The first element is the solar panel. The photovoltaic panel is responsible for collecting solar energy, to power up and recharge the batteries. The four rechargeable batteries play the role of an external power source, supplying all the required power to the Waspmote Board. According to Libelium (2017), the board requires at least 5 volts to function properly. The third components are the Universal Serial Bus (USB) cables. The USB cables are crucial for transferring and uploading the code sketches to the board, to acquire the appropriate data from the sensors. The final hardware elements, for completing this research, are the LoRaWAN modules and their antennas. Particularly, the inauguration of a connection between the Waspmote Boards but also between the boards and a gateway is established through the LoRaWAN module and its antenna. As the company also states, these two pieces of hardware operate in 868 Hz for the regions of Europe and enable seamless interconnection to any LoRaWAN products or network infrastructure. Furthermore, the range may vary from 2 up to 22 kilometers and any gateway within reach of the Waspmote device will receive the sensor readings and redirect them to a server for storage.

Software:

Apart from the hardware, it is equally important to give emphasis on the software that will be utilized for achieving the purposes of the current study. Overall, there are four parts related to the software aspect of the project, the first two related to the capture and transmission of the data and the latter to the storage and presentation of the information. At first, the Waspmote IDE is crucial for developing the code, to accomplish the procurement of the desired data. This IDE is freely provided by Libelium itself. The code sketches (created with this IDE and written in C++), will be uploaded to the Waspmote Board. Additionally, the NetBeans IDE will provide the environment for developing a program written in Java. The developed program will be accountable for capturing the data (sent from the Waspmote Board) and uploading them for storage to the database.

As far as the second activity is concerned, the first unit that will be brought into service, is the ThingSpeak platform. As previously described, this platform is necessary for storing real-time data, acquired from the sensors. However, the process of formulating and presenting the data is as important as storing them. For this final step, the Android Studio comes into play. With the aid of this IDE, an app will be developed (written in Java), to fetch the data from the ThingSpeak platform, do all the imperative calculations and display the results.

User Scenario:

For the present research, it is dire and essential to generate a design scenario that reflects the actions taken by a user. As stated by Nardi (1992), the purpose of a user scenario is to provide a solid, concrete vision on how the human activity can be supported with the help of technology. Moreover, the author asserts that a scenario will ensure that the system design is firmly rooted on the needs of a user and will describe in detail the interaction that needs to take place, so a task can be carried out.

Supposedly, a farmer has deployed and installed the system at a crops field, hence, achieving the integration of the system with the growing plants. The next step that must be completed, to make the DSS functional, is to download and install the app to a portable device. The necessity of Internet connection for downloading the app is obvious. As soon as the app download and installation are completed, the user needs to fill all the required information, by clicking the button with the description "SETTINGS". This button will lead to a screen, where the farmer will be asked to fill in four different types of information. This information is vital, in order to initiate the process and make the DSS serviceable. The first piece of information that the user must give is the type of crops that are planted and harvested. The second vital piece of information is the location of the field. Furthermore, the user needs to choose the altitude of the farming terrace and the water needs of the crops (how many days these crops can survive without water). As soon as the user finishes with the completion of all the necessary data, the "CONFIRM" button needs to be clicked and all the answers will be submitted and used as an input for configuring the system. Consequently, the app will lead the user back to the initial screen, where the "GET DATA" button will be available for clicking. Once this button is clicked, all the formulated data will be presented to the user, prompting him to take the appropriate action.

CHAPTER 9: IMPLEMENTATION

This chapter will analyze the overall development and construction process of the suggested DSS, thus, presenting the results and findings of the current research. Before moving on to the explanation of the execution and implementation, it is highly important to acknowledge that the system has been entirely completed and developed according to the system design (as analyzed in the previous chapter). Due to the fact that the structure of the system had been broken down into a pair of activities, made the project manageable, achievable and led to its full completion. The following subchapters are grouped by activities, to give special focus and consideration to each action.

Phase 1

Data Acquisition

To begin with, all four sensors were connected to their appropriate sockets to obtain the eight different types of data. Meanwhile, both the solar panel and the set of four rechargeable batteries were connected to the proper pockets, so as to continuously provide the Waspmote Board with electrical power. With the help of the Waspmote IDE, a sketch was developed and uploaded to the board by using a USB cable. The code written in this sketch activates the board every 15 minutes and collects all the data that were obtained from the sensors. However, one of sensor readings that requires additional consideration is the pluviometer. As previously mentioned, once the pluviometer bucket is filled, an interruption is caused that empties the bucket and sends the value to the board. However, if rainfall occurred while the Waspmote Board was switched off, the sensor reading would be lost if it was not retained in a variable. As a result, the Waspmote Board is activated as soon as an interruption is caused, and the amount of precipitation is being continuously stored in a variable. If precipitation is not detected, the value of the variable will be zero. As soon as the Waspmote Board is activated (after 15 minutes), the value of the variable is transmitted, thus, starting the recording of the precipitation from the beginning.

Figure 8: Installed System

Data Transmission

For the purposes of this research, a network of Waspmote Boards was created. Particularly, the acquired sensor data were transmitted from one Waspmote Board to another, to certify that a network of these boards can be established. For the transmission of the data (from the first Waspmote Board to the second), both antennas were used and placed on each board. Although, the sensor readings were eight in general, they had to be merged into one variable to ensure their proper transferal. Instead of obligating the board to set up eight different transmissions for each one of the data readings, the data merged into one variable (each reading separated with a line space), thus, setting up just one transmission. Nevertheless, it is crucial to specify that the data readings had to be converted into hexadecimal format, to be transmitted through LoRaWAN radio frequencies. As a result, the first board collected the data and converted them into hexadecimal format. As soon as the conversion was completed, the information was transmitted to the second board, where a reverse conversion was occurred, so the data could be changed from hexadecimal to string format.

However, it would be an oversight if the following point was not addressed. Due to equipment limitations, a gateway was not available for usage. Nonetheless, instead of using a gateway to receive the data from the second Waspmote Board and redirect them to ThingSpeak, a personal computer played that role. The personal computer was connected to the second Board with a USB cable, to receive the data. For successfully setting up a java – Waspmote communication and properly acquiring the sensors readings from the second Waspmote Device, a java program was developed with NetBeans IDE. With the help of the RXTX library, the java program communicated to the serial port (that the second Waspmote device was connected) and retrieved the sensor readings that were stored in a string variable. Next, the program separated each of the values (with the help of the existing line spaces that already isolated the values from one another) and prepared them for the storage process.

Phase 2

Data Storage

As aforementioned, the ThingSpeak platform would be used for the storage of the data and the MQTT protocol would be utilized for transmitting the information. Therefore, the java program had to implement the Paho Client MQTT library, to achieve the successful broadcast of the data with the MQTT protocol. As far as the ThingSpeak platform is concerned, a channel was created with the name "Waspmote Agricultural Weather Data" and eight fields were created for the storage of the information (each field corresponding to each data type). Furthermore, each channel in ThingSpeak is given a unique identification code and a write and read API key for submitting and retrieving data respectively. The implemented code for the java program included the channel identification code and the write API, so as to submit the data to the created channel. As seen in the figure below, each table is unique to the data it represents. As a further matter, each table depicts all the collected values of the data it represents and visualizes them as a line graph. The only exception is the table that represents the wind direction, where the data are saved as strings and could not be visualized.

Figure 9: Four of the Waspmote Agricultural Weather tables

As soon as the data preparation was completed, the java program published the information to the channel, submitting each data to the appropriate field. It is important to highlight the fact that each time a submission is published to the channel, the timestamp of the submission is also recorded and stored to the channel. As a conclusion, the user can also be notified when was the last time the channel was updated.

Data Presentation

For the formulation and presentation of the data, the ensuing process was followed. An application was developed with Android Studio, to make all the appropriate calculations and display the most prominent results, thus, prompting the user to take the appropriate decision. In the following figure, the initial screen of the app is shown, which contains two buttons. The first button, entitled "Get Data", leads to a screen where all the data (which were prepared methodically), are presented in a systematic and concise manner. This button can be activated if only the user fills in the appropriate information, by clicking the second button called "Settings". This procedure happens only once, when the application is initially launched. The application will save the configurations determined by the user, without requiring the re-registration of data every time the user starts the application.

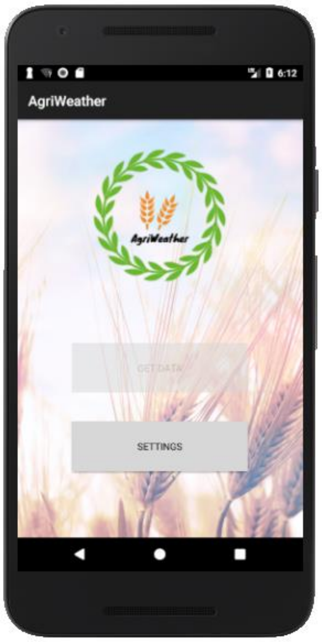

Figure 10: Initial screen of the app

As seen in the figure below, as soon as the button "Settings" is clicked, the following screen will appear, urging the user to fill in the blank fields with the appropriate data. The first information that the user needs to fill in is the name of the planted crops. Location of the farmland is the second type of information that the user also needs to write down. The third info that must be determined is the altitude of the field in feet. The user is given the option to pick one of the seven choices, so the approximate elevation of the field can be defined. The options are -1000, -500, zero, 500, 1000, 1500 and 2000. The last piece of information that must be submitted is the number of days the crop can survive without water.

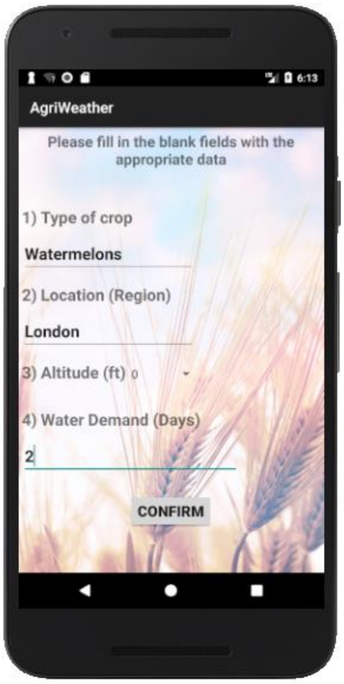

Figure 11: Settings screen

As soon as the user hits the “Confirm” button, the application will check if all the fields are completed. If they are not, the system will ask the user to insert the data to the appropriate field, else the application will redirect the user back to the initial screen. When the user successfully enters and registers all the necessary information, the application notifies the user that the entries have been inserted successfully and the “Get Data” button becomes available for clicking.

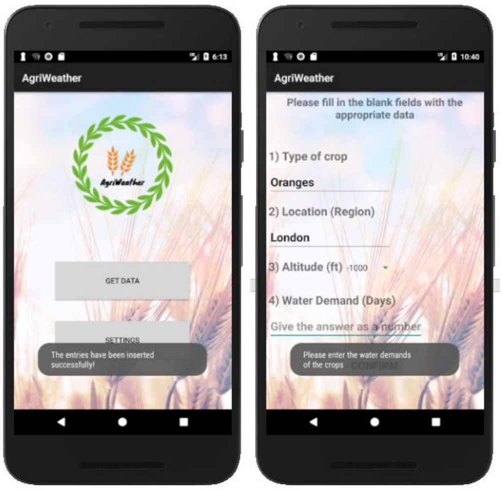

Figure 12: Successful data registration (left) - Unsuccessful data registration (right)

Back to the initial screen, the “Get Data” button leads to the configuration of the data. All outcomes, as discussed in the “System Design” chapter, are presented in this screen. All the information is retrieved from the ThingSpeak platform, with the help of the HTTP and the MQTT protocols. In this manner, not only the appropriate calculations can be shown at the configuration screen but also line graphs can be generated for the temperature and precipitation sensor values (as seen in the figure below).

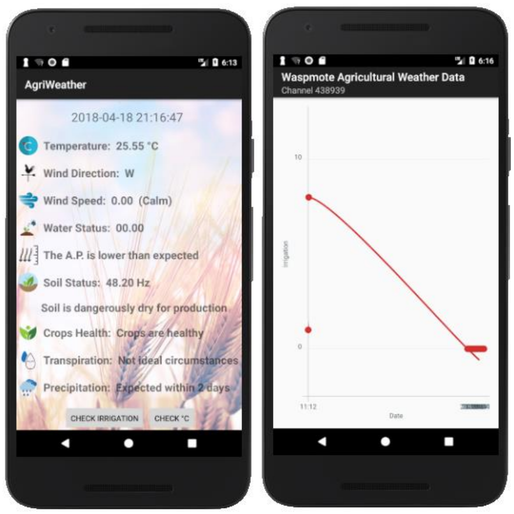

Figure 13: Get Data screen (left) - Line Chart screen (right)

CHAPTER 10: CONCLUSIONS

In summation, the previously stated research question has been answered through the detailed, specific and meticulous requirement analysis, system design and implementation of the DSS. The current research contemplated and examined previous studies, to narrow down and define effectiveness and determine ways to achieve it. The suggested DSS takes into consideration all four aspects of effectiveness and investigates them thoroughly and extensively. First and foremost, user-centered design principles were followed for the development of the application, as specified by Virkus et al. (2016). Furthermore, the aspect of adaptability was attained by adopting specific communication technologies that are resilient, buoyant and not intransigent. In addition, the input requirements that the system expects from the user (meaning the type of crop, location, altitude and days the crop can survive without water), upgrade the proposed DSS, by designing it to be flexible for all crops or plants and adaptable to any terrace. As far as the cost of the DSS is concerned, low-priced and inexpensive hardware supplies have been chosen, making the PA system affordable. Lastly, the present study proposed a considerable consolidation of different sources of information by combining eight different sources, thus, covering the final aspect of effectiveness at the same time.

Nonetheless, it is also essential to underline the two segments of the system. By breaking down the general structure into a pair of activities, the goal became clear and uncomplicated to achieve. As far as the first activity is concerned, the process of gathering data and transmitting them with the assistance of communication protocols, was successful. Additionally, the two components of the second activity also displayed and demonstrated prosperous results and outcomes. The storage of the data was accomplished without any malfunctions and the application displays the most prominent conclusions to the user, hence, reacting to the input data and formulating them accordingly.

However, a research would be inadequate, partial and deficient if further suggestions were not addressed and limitations were not highlighted. Both recommendations and restraints were identified while the research was being developed and unfolded.

Improvements:

The first matter that needs to be addressed, is associated with the input data that the user needs to provide to the system, specifically to the application. To be precise, instead of obliging users to add the necessary data about the crops (namely the name of the crop and how many days it can endure without water), an alternative method could be utilized. Particularly, a database that stores information about the crops and their water day demands could be created and be affiliated with the DSS. Rather than requesting the data, the user could only select the type of crops that are planted, and the application would automatically generate a request to the database, retrieving information regarding the crop or the plant that the user chose. As a result, the application would retrieve the data from the database and would incorporate it to the decision-making process.

Limitations:

One of the restrictions that the study faced, was the lack of feedback. Specifically, even though the DSS was designed and assembled by studying and researching agricultural, weather and farming related material, there was no feedback from actual users themselves. Due to time constraints, it was not feasible and achievable to invite users to assess and evaluate the system. Any omissions, oversights and exclusions should be reported and documented by a group of people who are associated with the agricultural sector. The sampling could include farmers, growers, farming consultants or even agronomists and scientists who are related to the crop production, soil conservation and preservation, agricultural education or the field of farming in general. Consequently, a pragmatic acumen and useful insights will be gained, by conducting a testing on the performance of the DSS.

Another limitation ought to be acknowledged, is the absence of a gateway during the system implementation. Since this piece of hardware was not available for usage, an alternative mechanism had to be chosen under these circumstances. On these terms, a personal laptop was appointed and designated to fill the position. Surely, even though the DSS is satisfactorily functional, the gateway will aid in updating the system and making it completely equipped.

CHAPTER 11: EPILOGUE

Through this research, the goal to develop an effective DSS and advance PA was greatly achieved. All aspects of effectiveness were consciously covered in great depth and the proposed DSS was a product of merging existing knowledge and understandable long-term promises for the future. These promises are linked to sustainable agricultural practices, resource- protection policies and dire need of boosting farming technologies. Moreover, the completed research contributes to the field of agriculture and strengthens the vision for a technological agricultural revolution.

CHAPTER 12: BIBLIOGRAPHY

Acharya, C. (2016). IOT Platforms: A Brief Study. *International Journal of Advanced Research in Computer Science and Technology*, *4*(2), 2–5.

Adams, D. (2002). *The Salmon of Doubt: Hitchhiking the Galaxy One Last Time*. London: Pan Books.

Alter, S. (1980). *Decision support systems: current practice and continuing challenges (Addison-Wesley series on decision support).* Boston: Addison-Wesley

Asin, A. (2017). The Future of Farming through the Internet of Things perspective. *Ciheam*, (February). Retrieved from http://www.ciheam.org/uploads/attachments/403/21_LIBELIUM_Edited.pdf

Bala, K., Sharma, S., & Kaur, G. (2015). A Study on Smartphone based Operating System. *International Journal of Computer Applications*, *121*(1), 17–22.

Balafoutis, A., Beck, B., Fountas, S., Vangeyte, J., Van Der Wal, T., Soto, I., … Eory, V. (2017). Precision agriculture technologies positively contributing to ghg emissions mitigation, farm productivity and economics. *Sustainability (Switzerland)*. https://doi.org/10.3390/su9081339

Chauhan, R. M. (2015). ORIENTAL JOURNAL OF Advantages And Challeging in E Agriculture. *Oriental Journal of Computer Science and Technology*, *8*(3).

Clitheroe, F. (2010). *X-kit Fet G11 Life Sciences*. South Africa: Pearson.

Dürkop, L., Czybik, B., & Jasperneite, J. (2015). Performance evaluation of M2M protocols over cellular networks in a lab environment. *2015 18th International Conference on Intelligence in Next Generation Networks, ICIN 2015*, 70–75. https://doi.org/10.1109/ICIN.2015.7073809

European Commission. (2017). Direct Payments for Farmers 2015-2020, (May). Retrieved from file:///C:/Users/Bridget/Desktop/new/CAP EXPLAINED DIRECT PAYMENTS FOR FARMERS 2015-2020.pdf

Eurostat. (2010). Survey on agricultural production methods. Retrieved January 8, 2018, from http://ec.europa.eu/eurostat/statistics-explained/index.php/Survey_on_agricultural_production_methods

Evans, D. (2011). The Internet of Things - How the Next Evolution of the Internet is Changing Everything. *CISCO White Paper*, (April), 1–11. https://doi.org/10.1109/IEEESTD.2007.373646

Food and Agriculture Organization of the United Nations (FAO). (2005). “e-Agriculture”: A Definition and Profile of its Application. *Bridging the Rural Digital Divide*, (November), 1–6.

Fysarakis, K., Askoxylakis, I., Soultatos, O., Papaefstathiou, I., Manifavas, C., & Katos, V. (2016). Which IoT Protocol? *2016 IEEE Global Communications Conference (GLOBECOM)*. https://doi.org/10.1109/GLOCOM.2016.7842383

Ganesan, V. (2005). Decision Support System “Crop-9-DSS” for Identified Crops. *Agriculture*, *1*(12), 473–475.

Goldberg, Y. (2017). *Neural Network Methods in Natural Language Processing (Synthesis Lectures on Human Language Technologies).* San Rafael: Morgan & Claypool Publishers

Gómez Maureira, M. A., Oldenhof, D., & Teernstra, L. (2014). ThingSpeak – an API and Web Service for the Internet of Things. *World Wide Web*. Retrieved from http://mediatechnology.leiden.edu/images/uploads/docs/wt2014_thingspeak.pdf%0Ahttps://staas.home.xs4all.nl/t/swtr/documents/wt2014_thingspeak.pdf

GSMA Association. (2014). Understanding the Internet of Things (IoT). *Gsma Connected Living*, (July), 15. https://doi.org/10.5120/19787-1571

Hayman, P. (2011). Decision support systems in Australian dryland farming: A promising past, a disappointing present and uncertain future. *New Directions for a Diverse Planet. Proceedings of the 4th International Crop Science Congress’*, (2003), 1–9.

Hovardas, T., & Ph, D. (2014). Strengths, Weaknesses, Opportunities and Threats (SWOT) Analysis: A template for addressing the social dimension in the study of socio- scientific issues, 1–12.

Humphrey, A. (1960). How to Use SWOT Analysis. *Journal of Strategic Management*, *12*(5), 78–124. Retrieved from http://applications.creativeengland.co.uk/assets/public/resource/85.pdf

Ii, J. C. A., Shaffer, M. J., Hoag, D. L., Mcmaster, G. S., Dunn, G. H., Ahuja, L. R., & Mark, a. (2001). GPFARM: An Integrated Decision Support System for Sustainable Great Plains Agriculture Display Output Indices. *Organization*, 951–960.

Joint Research Centre (JRC) of the European Commission. (2014). Precision Agriculture: an Opportunity for Eu Farmers- Potential Support With the Cap 2014 - 2020. *European Union*, 56. https://doi.org/10.2861/58758

Lee, W.-M. (2012). *Beginning Android Application Development*. Hoboken: John Wiley & Sons.

Lencsés, E., Takács, I., & Takács-György, K. (2014). Farmers’ perception of precision farming technology among Hungarian farmers. *Sustainability (Switzerland)*, *6*(12), 8452–8465. https://doi.org/10.3390/su6128452

Libelium. (2017). *Smart Agriculture 3.0 Technical Guide*.

Libelium. (2017). *Watermark Sensors Interpretation Reference.*

Lindblom, J., & Ljung, M. (2014). Agriculture through Sustainable IT Failures and success factors in AgriDSS. *The 11th European IFSA Symposium*, *2*, 49–57.

LoRa Alliance. (2015). LoRaWAN, What is it ? A technical overview of LoRa and LoRaWAN.

Malavade, V. N., & Akulwar, P. K. (2016). Role of IoT in Agriculture. *IOSR Journal of Computer Engineering*, 56–57.

Manyika, J., Ramaswamy, S., Khanna, S., Pinkus, G., Sethupathy, G., & Yaffe, A. (2015). Digital America: A tale of the haves and have-mores. *McKinsey Global Institute*, (December), 107. Retrieved November 10, 2017 from https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/the-internet-of-things-the-value-of-digitizing-the-physical-world

Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology. *National Institute of Standards and Technology, Information Technology Laboratory*, *145*, 7. https://doi.org/10.1136/emj.2010.096966

Meola, A., (2016). *Why IoT, Big Data & Smart Farming Are the Future of Agriculture.* Retrieved November 10, 2017 from http://www.businessinsider.com/internet-of-things-smart-agriculture-2016-10

Nakhuva, B., & Champaneria, T. (2015). Study of Various Internet of Things Platforms. *International Journal of Computer Science & Engineering Survey*, *6*(6), 61–74. https://doi.org/10.5121/ijcses.2015.6605

Nardi, B. A. (1992). The use of scenarios in design. *ACM SIGCHI Bulletin*, *24*(4), 13–14. https://doi.org/10.1145/142167.142171

Nguyen, N. C., Wegener, M., & Russell, I. (2005). Decision support systems in Australian agriculture: state of the art and future development. *AFBM Journal*, *4*(1990), 15–21.

Ommani, A. R. (2010). Strengths, weaknesses, opportunities and threats (SWOT) analysis for farming system businesses management: Case of wheat farmers of Shadervan District , Shoushtar Township, Iran. *African Journal of Business Management*, *5*(22), 9448–9454. Retrieved from http://www.academicjournals.org/AJBM%5Cnhttp://www.academicjournals.org/journal/AJBM/article-full-text-pdf/21F95E420498

Osita, I., Onyebuchi, I., & Nzekwe, J. (2014). Organization's stability and productivity: the role of SWOT analysis an acronym for strength, weakness, opportunities and threat. *International Journal of Innovative and Applied Research*, *2*((9): 23-32), 1–12. Retrieved from http://www.journalijiar.com

Power, D.J., (2002). *Decision support systems: Concepts and resources for managers*. Westport: Greenwood Publishing Group

Prasad, M., Prasad, R., & Kulkarni, U. (2008). A Decision Support System for Agriculture Using Natural Language Processing (ADSS). *Proceedings of the International MultiConference of Engineers and Computer Scientists*, *I*, 19–21.

Rowlandson, T., Gleason, M., Sentelhas, P., Gillespie, T., Thomas, C., & Hornbuckle, B. (2015). Reconsidering Leaf Wetness Duration Determination for Plant Disease Management. *Plant Disease*, *99*(3), 310–319. https://doi.org/10.1094/PDIS-05-14-0529-FE

Rose, K., Eldridge, S., & Lyman, C. (2015). The internet of things: an overview. *Internet Society*, (October), 53. https://doi.org/10.1017/CBO9781107415324.004

Royce, D. W. W. (1970). Managing the Development of large Software Systems. *Ieee Wescon*, (August), 1–9.

Schrijver, R. (2016). *Precision agriculture and the future of farming in Europe Scientific Foresight Study*. https://doi.org/10.2861/020809

Tan, L. (2016). Cloud-based Decision Support and Automation for Precision Agriculture in Orchards. *IFAC-PapersOnLine*, *49*(16), 330–335. https://doi.org/10.1016/j.ifacol.2016.10.061

TAPROOT. (2017, March 29). What is Vapor Pressure Deficit (VPD) and Why it's Important to Your Indoor Garden. Retrieved March 20, 2018, from https://taphydro.com/blogs/news/what-is-vpd-or-vapor-pressure-deficit

U.S. Department of Agriculture Forest Service. (1992). The Container Tree Nursery manual. *USDA Agricultural Handbook 674 -Atmospheric Environment*, 47–71.

Venkatalakshmi, B., & Devi, P. (2014). Decision support system for precision agriculture. *International Journal of Research in Engineering and Technology*, 3(7), 849–852. http://www.academia.edu/8160091/DECISION_SUPPORT_SYSTEM_FOR_PRECISION_AGRICULTURE

Virkus Robert, Akbari Mostafa, Alfut Anna, Thain Ian, Bradley Davoc, Bloor Richard, Churchill Dean, Harty Julian, Heubel Bob, T. M. (2016). *Don’t Panic: Mobile Developer’s Guide to The Galaxy, 17th Edition*.

WatchDog. (2015). WATERMARK Soil Moisture Sensor - MODEL 200SS. *2015*. Retrieved from http://www.irrometer.com/pdf/sensors/403 SensorWeb5.pdf

Zachariah, T., Klugman, N., Campbell, B., Adkins, J., Jackson, N., & Dutta, P. (2015). The Internet of Things Has a Gateway Problem. *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications - HotMobile ’15*, 27–32. https://doi.org/10.1145/2699343.2699344

CHAPTER 13: APPENDICES

The appendices that follow, demonstrate code snippets which cover the basic programming commands that were used for accomplishing the DSS.

Appendix A: Code for acquiring sensor data

```
temp = Agriculture.getTemperature();
  humd  = Agriculture.getHumidity();
  pres = Agriculture.getPressure();
  watermark1 = wmSensor1.readWatermark();
  leafvalue = lwSensor.getLeafWetness();
  anemometer = weather.readAnemometer();
  vaneSensor.getVaneFiltered();
  switch (vaneSensor.vaneDirection)
  {
    case  SENS_AGR_VANE_N   :
      vane = "N";
      break;
    case  SENS_AGR_VANE_NNE :
      vane = "NNE";
      break;
    case  SENS_AGR_VANE_NE  :
      vane = "NE";
      break;
    case  SENS_AGR_VANE_ENE :
      vane = "ENE";
      break;
    case  SENS_AGR_VANE_E   :
      vane = "E";
      break;
  case  SENS_AGR_VANE_ESE :
      vane = "ESE";
      break;
```

```
  case  SENS_AGR_VANE_SE  :
    vane = "SE";
    break;
  case  SENS_AGR_VANE_SSE :
    vane = "SSE";
    break;
  case  SENS_AGR_VANE_S   :
    vane = "S";
    break;
  case  SENS_AGR_VANE_SSW :
    vane = "SSW";
    break;
  case  SENS_AGR_VANE_SW  :
    vane = "SW";
    break;
  case  SENS_AGR_VANE_WSW :
    vane = "WSW";
    break;
  case  SENS_AGR_VANE_W   :
    vane = "W";
    break;
  case  SENS_AGR_VANE_WNW :
    vane = "WNW";
    break;
  case  SENS_AGR_VANE_NW  :
    vane = "WN";
    break;
  case  SENS_AGR_VANE_NNW :
    vane = "NNW";
    break;
}
```

```
  //Pluviometer Code
  Agriculture.sleepAgr("00:00:00:00", RTC_ABSOLUTE, RTC_ALM1_MODE4, SENSOR_ON,
SENS_AGR_PLUVIOMETER);
  point =  weather.readPluviometerCurrent();
  newResult = point - secondMinus;
  secondMinus = point;
```

Appendix B: Code for converting data to hexadecimal format

The code that follows is just for the temperature sensor data, but the same format was applied to all the acquired data. The data is being converted to hex format and a line space is added as well.

```
 char temperature[15];
 dtostrf(temp, 4, 2, temperature);
 char *out;
 if ((out = (char *)malloc(strlen(temperature) + strlen("\n") + 1)) != NULL)
 {
   strcpy(out, temperature);
   strcat(out, "\n");
 }
```

Appendix C: Code for sending data over LoRaWAN

```
char buffer[120];
 error = LoRaWAN.sendRadio(buffer);
 if (error == 0)
 {
   USB.println(F("--> Packet sent OK"));
 }else
 {
   USB.print(F("Error waiting for packets"));
 }
```

Appendix D: Code for receiving data over LoRaWAN

```
void loop()
{
  delay(15000);
  //USB.println(F("\nListening to packets..."));
  error = LoRaWAN.receiveRadio(10000);

  // Check status
  if (error == 0)
  {
    //USB.println(F("--> Packet received"));
    //USB.print(F("packet: "));

    size = Utils.str2hex((char*) LoRaWAN._buffer, buffer2, sizeof(buffer2));
    USB.println(buffer2, size);

    //USB.println((char*) LoRaWAN._buffer);
    //USB.print(F("length: "));
    //USB.println(LoRaWAN._length);

    // get SNR
    LoRaWAN.getRadioSNR();
    //USB.print(F("SNR: "));
    //USB.println(LoRaWAN._radioSNR);

  }
  else
  {
    //USB.print(F("Error waiting for packets. error = "));
    //USB.println(error, DEC);
  }
}
```

Appendix E: Java program for reading data from the serial port

```
try {
    // open serial port and use class name for the appName.
    serialPort = (SerialPort) portId.open(this.getClass().getName(),
            TIME_OUT);

    // set port parameters
    serialPort.setSerialPortParams(DATA_RATE,
            SerialPort.DATABITS_8,
            SerialPort.STOPBITS_1,
            SerialPort.PARITY_NONE);

    // open the streams
    input = new BufferedReader(new InputStreamReader(serialPort.getInputStream()));
    output = serialPort.getOutputStream();

    // add event listeners
    serialPort.addEventListener(this);
    serialPort.notifyOnDataAvailable(true);
} catch (Exception e) {
    System.err.println(e.toString());
}
```

Appendix F: Continuation of Java program. Preparing the data for ThingSpeak

```
if (counter == 0) {

        System.out.println("Temperature");
        totals[counter] = inputLine;
        one = inputLine;

    } else if (counter == 1) {

        System.out.println("Humidity");
        totals[counter] = inputLine;
        two = inputLine;

    } else if (counter == 2) {

        System.out.println("Pressure");
        totals[counter] = inputLine;
        three = inputLine;

    } else if (counter == 3) {

        System.out.println("Watermark");
        totals[counter] = inputLine;
        four = inputLine;

    } else if (counter == 4) {
        System.out.println("Leaf");
        totals[counter] = inputLine;
        five = inputLine;

    }
```

```
else if (counter == 5) {

        System.out.println("Anemometer");
        totals[counter] = inputLine;
        six = inputLine;

    } else if (counter == 6) {

        System.out.println("Vane");
        totals[counter] = inputLine;
        seven = inputLine;

    } else if (counter == 7) {

        System.out.println("Pluviometer");
        totals[counter] = inputLine;
        eight = inputLine;
    }

    counter += 1;

    if (counter > 7) {
        MQTT mqtt = new MQTT();
        mqtt.publish("field1=" + one + "&field2=" + two + "&field3=" + three + "&field4=" +
four + "&field5=" + five + "&field6=" + six + "&field7=" + seven + "&field8=" + eight);
        counter = 0;
    }
```

Appendix G: Code for sending the data to ThingSpeak over MQTT

```
try {
        // Construct the connection options object that contains connection parameters
        conOpt = new MqttConnectOptions();
        conOpt.setCleanSession(true);
        client = new MqttClient(brokerUrl, MqttClient.generateClientId(), dataStore);

        // Set this wrapper as the callback handler
        client.setCallback(new MqttCallback() {

          @Override
          public void messageArrived(String arg0, MqttMessage arg1) throws Exception {
            // TODO Auto-generated method stub
            String time = new Timestamp(System.currentTimeMillis()).toString();
            System.out.println("Time:\t" + time
                + "  Topic:\t" + arg0
                + "  Message:\t" + new String(arg1.getPayload())
                + "  QoS:\t" + arg1.getQos());
          }

          @Override
          public void deliveryComplete(IMqttDeliveryToken arg0) {
            // TODO Auto-generated method stub

            try {
              System.out.println(arg0.getMessageId() + " " + arg0.getMessage());
            } catch (MqttException e) {
              // TODO Auto-generated catch block

            }
          }
```

```
        @Override
        public void connectionLost(Throwable arg0) {
            // TODO Auto-generated method stub

            System.out.println("Connection to " + brokerUrl + " lost!");

            System.out.println("Reconnecting..");
            try {
                client.connect(conOpt);
                //System.out.println("Connected1");
            } catch (MqttException e) {
                // TODO Auto-generated catch block

            }
        }
    });
    if (client.isConnected()) {

        System.out.println("was already Connected");
        client.disconnect();
    }
    client.connect(conOpt);

} catch (MqttException e) {
    System.out.println("Unable to set up client: " + e.toString());
}
```

Appendix H: Code in Android app for retrieving data over MQTT and HTTP

```
        String topic = "channels/438939/subscribe/************";
        int qos = 1;
        try {
          IMqttToken subToken = client.subscribe(topic, qos);
          subToken.setActionCallback(new IMqttActionListener() {
            @Override
            public void onSuccess(IMqttToken asyncActionToken) {
              // The message was published
            }

            @Override
            public void onFailure(IMqttToken asyncActionToken,
                                  Throwable exception) {
              // The subscription could not be performed, maybe the user was not
              // authorized to subscribe on the specified topic e.g. using wildcards

            }
          });
        } catch (MqttException e) {
          e.printStackTrace();
        }

      }

      @Override
      public void onFailure(IMqttToken asyncActionToken, Throwable exception) {
        // Something went wrong e.g. connection timeout or firewall problems
        System.out.println("Fail");

      }
    });
} catch (MqttException e) {
    e.printStackTrace();
}
```

```
  JSONObject json1 = new JSONObject(IOUtils.toString(new
URL("http://api.thingspeak.com/channels/438939/feed.json"), Charset.forName("UTF-8")));
  jsonarr = json1.getJSONArray("feeds");

  int length = jsonarr.length();

  //Retrieving last time that data was sent
  timeRetrieval = jsonarr.getJSONObject(length - 1).getString("created_at");
  timeRetrieval = timeRetrieval.replaceAll("[^-:\\d.]", " ");
  lastUpdate = findViewById(R.id.lastUpdated);
```

Appendix I: Data Formulation

```
//Temperature
   tempFinal = findViewById(R.id.temp);

   //Watermark
   watermarkSensor = findViewById(R.id.soil);
   waterMark = findViewById(R.id.soilValue);

   if (watermarkDouble <= 0) {

      watermarkDecision = "Soil is dangerously dry for production";

   } else if (watermarkDouble > 0 && watermarkDouble <= 500) {

      watermarkDecision = "Need for irrigation (heavy clay soil)";

   } else if (watermarkDouble > 500 && watermarkDouble <= 1000) {

      watermarkDecision = "Need for irrigation (except heavy clay soil)";

   } else if (watermarkDouble > 1000 && watermarkDouble <= 5000) {

      watermarkDecision = "Soil is adequately wet (coarse sands lose water))";

   } else if (watermarkDouble > 5000 && watermarkDouble <= 7000) {

      watermarkDecision = "Saturated soil)";

   } else if (watermarkDouble > 7000) {

      watermarkDecision = "Oversaturated soil)";
   }

//Retrieving atmospheric pressure
   atmoPressure = Double.parseDouble(atmopressure);
   compareAP = findViewById(R.id.comparePressure);

   if (atmoPressure > settings.spinnerAltitude) {

      finalAP = "The A.P. is higher than expected";

   }

  else if (atmoPressure == settings.spinnerAltitude) {

      finalAP = "The A.P. is normal";

   }
```

```java
else {

    if (temperatureFinal >= temp30) {

        finalAP = "Expect an increased water loss";

    } else {

        finalAP = "The A.P. is lower than expected";

    }

  }

  //Wind direction
  windDirected = findViewById(R.id.windDirection);

  //Pluviometer data
  waterPluviometer = findViewById(R.id.waterPluvio);

  //Wind speed
  windSpeeds = findViewById(R.id.windspeed);
  characterization = Float.valueOf(windSpeed);
  windCharacterize = findViewById(R.id.windCharacterization);

  if (characterization < 1) {
    windCharacter = "(Calm)";
  } else if (characterization >= 1 && characterization < 6) {
    windCharacter = "(Light Air)";
  } else if (characterization >= 6 && characterization < 12) {
    windCharacter = "(Light Breeze)";
  } else if (characterization >= 12 && characterization < 20) {
    windCharacter = "(Gentle Breeze)";
  } else if (characterization >= 20 && characterization < 29) {
    windCharacter = "(Moderate Breeze)";
  } else if (characterization >= 29 && characterization < 39) {
    windCharacter = "(Fresh Breeze)";
  } else if (characterization >= 39 && characterization < 49) {
    windCharacter = "(Strong Breeze)";
  } else if (characterization >= 49 && characterization < 61) {
    windCharacter = "(Near Gale)";
  } else if (characterization >= 61 && characterization < 74) {
    windCharacter = "(Gale)";
  } else if (characterization >= 74 && characterization < 88) {
    windCharacter = "(Strong Gale)";
  } else if (characterization >= 88 && characterization < 102) {
    windCharacter = "(Storm)";
  }else if (characterization >= 102 && characterization < 117) {
    windCharacter = "(Strong Storm)";
  } else if (characterization >= 117) {
    windCharacter = "(Hurricane)";}
```

```java
//Calculating Avg temperature & Leaf Diseases
counter = 0;
while (counter < length) {

    double temp = Double.parseDouble(field1);
    totalTemperatureSum += temp;
    counter++;

}
totalTMP = totalTemperatureSum/length;
DecimalFormat df = new DecimalFormat("#.##");
double totalSumTMP = Double.parseDouble(df.format(totalTMP));
System.out.println(totalSumTMP);


count = length-1;
while (count>=0){

    String field5 = jsonarr.getJSONObject(count).getString("field5");
    double leafWetness = Double.parseDouble(field5);
    System.out.println(leafWetness);

    if(leafWetness>0){
        leafCount+=1;
        count-=1;

    }else{
        break;
    }


}


if(leafCount>=10 & totalSumTMP>30){
    leafDecision = "Check crops for diseases";
}else{
    leafDecision = "Crops are healthy";
}
```

```
leafWetnessFinal = (TextView)findViewById(R.id.cropsHealth);

//Humidity – Temperature Part
double tempVPD = Double.parseDouble(fieldTemp);

//humidity
double humVPD = Double.parseDouble(fieldHum);

if( (tempVPD>=15 && tempVPD<19 && humVPD>=35 && humVPD<40) || (tempVPD>=15 &&
tempVPD<20 && humVPD>=40 && humVPD<45)|| (tempVPD>=15 && tempVPD<22 &&
humVPD>=45 && humVPD<50) || (tempVPD>=15 && tempVPD<23 && humVPD>=50 &&
humVPD<55) || (tempVPD>=15 && tempVPD<25 && humVPD>=55 && humVPD<60) ||
(tempVPD>=15 && tempVPD<27 && humVPD>=60 && humVPD<65) || (tempVPD>=15 &&
tempVPD<29 && humVPD>=65 && humVPD<70) || (tempVPD>=15 && tempVPD<32 &&
humVPD>=70 && humVPD<75) || (tempVPD>=15 && tempVPD<35 && humVPD>=75 &&
humVPD<80) || (tempVPD>=18 && tempVPD<36 && humVPD>=80 && humVPD<85) ||
(tempVPD>=23 && tempVPD<36 && humVPD>=85 && humVPD<90) || (tempVPD>=29 &&
tempVPD<36 && humVPD>=90 && humVPD<95)){

    vpd = "Ideal circumstances";

}else{

    vpd = "Not ideal circumstances";

}
```