**DESIGN AND IMPLEMENTATION OF A FARM MANAGEMENT INFORMATION SYSTEM**

**BY**

**OLUMBA CHIEBUKA FAVOUR**

**15CK02972**

**A PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING, COLLEGE OF ENGINEERING IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF ENGINEERING DEGREE IN ELECTRICAL AND ELECTRONICS ENGINEERING.**

**SUPERVISOR: PROF. E. ADETIBA**

**AUGUST 2020**

certification

This is to certify that the project titled “Design and implementation of a Farm Management Information System” by Olumba Chiebuka Favour, meets the requirements and regulations governing the award of the Bachelor of Engineering, B.Eng. (Electrical and Electronics Engineering) degree of Covenant University and is approved for its contribution to knowledge and literary presentation.

Student: Sign: ………………………. …………………………..

Name: Olumba Chiebuka Favour Date:

Supervisor: Sign: …… …..01/09/2020…..

Name: Prof. E. Adetiba Date

Internal Examiner: Sign: ……………………… …………………………..

Name: ……………………. Date

HOD: Sign: ………………………. …………………………..

Name: Prof. A. Adoghe Date

External Examiner Sign: ……………………… ……………………………

Name: …………………… Date

# 

dedication

This project is dedicated to God Almighty, for His grace and guidance throughout the period of this project. I also dedicate this project to my friends and family who contributed to the success of this project.

acknowelgements

My sincere gratitude goes to God Almighty for enabling me to complete this work. I would like to thank my family and friends for their support and financial assistance.

I would like to express my appreciation to my project supervisor, Prof. E. Adetiba and Mr. O. T. Ajayi, for all their assistance during this project.

abstract

Advances in IoT technology have propelled significant advancements in agriculture. Farm data can be sent to farmers in real-time through Farm Management Information Systems, which integrates data collection, data transfer, storage systems and a few other components that provide a great user experience. This project encompasses the design of a Farm Management Information System that transfers sensor field data to the Internet, via a microcontroller and a Wi-Fi module. With the rising population of the world today, it is imperative to get rid of time-consuming and non-economical agricultural practices; a major problem, which this project addresses. In this project, a Farm Management Information System was designed and implemented using Arduino Uno, Proteus design suite, and software development. Python programming language with the Django framework was used to develop a SmartAgrio farm management web portal. The significance of this project is evident, as it enables the farmers to run their farms without the need of being physically present. This project also elicits requirements for better improvements on the Farm Management Information System.

Table of contents

certification ii

dedication iii

acknowelgements iv

abstract v

Table of contents vi

List of figures ix

List of tables x

chapter one 1

INTRODUCTION 1

1.1 Background of the study 1

1.1.1 Data in Farm Management 2

1.1.2 Farm Management Information Systems 2

1.1.3 Indoor Farming 3

1.2 problem statement 4

1.3 motivation of the study 4

1.4 aim and objectives 4

1.5 Significance of the project 4

1.6 METHODOLOGY 5

1.7 project organisation 5

1.8 SUMMARY 5

chapter two 7

LITERATURE REVIEW 7

2.1 Introduction 7

2.2 definItion of key terms 7

2.2.1 Problems Facing the Wide Adoption of Farm Management Information Systems 8

2.3 Review of related works 9

2.3.1 Review of the Definition and Formulation of the Research Problem 9

2.3.2 Review of Automated Solutions in the Agricultural Industry 10

2.3.3 Review of the Application of Intelligent Monitoring Systems in Agriculture 11

2.3.4 Review on Automated Irrigation Solutions 15

2.4 SUMMARY 16

CHAPTER THREE 17

SYSTEM ANALYSIS AND DESIGN 17

3.1 INTRODUCTION 17

3.2 system hARDWARE 17

3.2.1 Arduino UNO 17

3.2.2 TSL2561 Digital Luminosity Sensor 19

3.2.2 Resistive Corrosion-Resistant Soil Moisture Sensor 21

3.2.3 DHT22 Temperature and Humidity Sensor 21

3.2.4 NodeMCU 22

3.2.5 Hardware Block Diagram 24

3.2.6 Circuit Simulation 25

3.3 System software 26

3.3.1 Python 27

3.3.2 Django 28

3.3.3 Unified Modelling Language (UML) 28

3.3.4 Use Case Diagram 30

3.3.5 Class Diagram 31

3.3.6 Activity Diagram 33

3.3.7 Sequence Diagram 35

3.4 system interaction 36

3.5 SUMMARY 36

chapter 4 37

SYSTEM IMPLEMENTATION AND TESTING 37

4.1 INTRODUCTION 37

4.2 Circuit Simulation results 37

4.2.1 Arduino Programming 38

4.3 Web portal implementation 38

4.3.1 Python Software 39

4.3.2 Farmer Section of the SmartAgrio Web portal 39

4.3.3 Administration Section of the SmartAgrio Web portal 44

4.4 Software testing 44

4.4.1 Unit Testing 44

4.4.2 Deployment Testing 46

4.4.3 Usability Testing 46

4.4.4 User Interface Testing 46

4.4.5 Questionnaire 47

4.5 SUMMARY 49

CHAPTER 5 50

Conclusions and recommendations 50

5.1 SUMMARY 50

5.2 ACHIEVEMENTS 50

5.2.1 Challenges Encountered 50

5.2.2 Limitations 51

5.3 CONCLUSION 51

5.4 RECOMMENDATION 51

REFERENCES 52

List of figures

[Figure 3.1: Arduino Uno 20](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc48419063)

[Figure 3.2: Front-view of a Digital Luminosity Sensor 21](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc48419064)

[Figure 3.3: Back-view of a Digital Luminosity Sensor 21](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc48419065)

[Figure 3.4: Resistive Corrosion-Resistant Soil Moisture Sensor 22](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc48419066)

[Figure 3.5: DHT22 Temperature and Humidity Sensor 23](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc48419067)

[Figure 3.6: NodeMCU Device 24](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc48419068)

[Figure 3.7: Hardware Connection Block Diagram 26](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc48419069)

[Figure 3.8: Circuit Diagram 27](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc48419070)

[Figure 3.9: Use Case Diagram 31](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc48419071)

[Figure 3.10: Use Case Diagram for Web Administrator 32](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc48419072)

[Figure 3.11: Class Diagram 33](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc48419073)

[Figure 3.12: Activity Diagram 35](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc48419074)

[Figure 3.13: Sequence Diagram 36](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc48419075)

[Figure 3.14: System Interaction Diagram 37](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc48419076)

[Figure 4.1: Proteus Simulation of Hardware Circuit 39](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc49333095)

[Figure 4.2: Snippet of Arduino Code Highlighting the ‘.elf’ file in Green 40](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc49333096)

[Figure 4.3: Snippet of Python Code 41](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc49333097)

[Figure 4.4: SmartAgrio Login Page 42](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc49333098)

[Figure 4.5: Registration Prompt on the SmartAgrio Web Portal 42](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc49333099)

[Figure 4.6: Current Readings Display Page on the SmartAgrio Webpage 43](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc49333100)

[Figure 4.7: Light Readings 43](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc49333101)

[Figure 4.8: Temperature Readings 44](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc49333102)

[Figure 4.9: Humidity Readings 44](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc49333103)

[Figure 4.10: Soil Moisture Readings 45](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc49333104)

[Figure 4.11: Test User Profile Page 45](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc49333105)

[Figure 4.12: Administrator Page 46](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc49333106)

[Figure 4.13: HTML Testing with the Nu Html Checker before Bug Fixes 47](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc49333107)

[Figure 4.14: Nu Html checker after Bug Fixes 48](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc49333108)

[Figure 4.15: Participants Occupation 50](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc49333109)

[Figure 4.16: Participant Perception of the Web portal 50](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc49333110)

[Figure 4.17: User Experience Response 51](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc49333111)

List of tables

[Table 3.1: Table showing the different hardware components and their voltage ratings 18](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc48419447)

[Table 3.2: The terminals on the device 20](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc48419448)

[Table 3.3: The terminal type either an input or output or none 21](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc48419449)

[Table 3.4: Table showing parameters and their sensitivity 23](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc48419450)

[Table 3.5: Table showing DHT pins and their description 23](file:///C:\Users\favour\Desktop\15CK02972_EEE_ADETIBA_Edited_NotYetApproved.docx#_Toc48419451)

chapter one

INTRODUCTION

1.1 Background of the study

Recent technological advances have seen to the explosion of information concerning various subjects in the IT world. Advances in sensor technology, wireless and Internet communication has led to the unlimited possibilities that can be achieved with the Internet of Things (IoT) [1]. The first mention of IoT was in 2006, and since then, the subject has led to tremendous technological advancement in different fields, for example, engineering, economics, finance, medicine and so on [2]. The Internet of things provides a platform to link everything around us to the Internet. This is important because these devices could operate on their own, without human input, and they can be controlled remotely.

Thus, it is evident that the IoT links people to devices and devices to devices. All of these proceeds through the Internet. Agriculture has not been left out in the recent developments; in fact, the IoT plays a significant role in modern agriculture in so many ways such as smart irrigation, early disease prediction for plants, in factory farming facilities, animal feeding, agricultural monitoring and many others. Agriculture is an important sector for the use of IoT controlled systems, especially because of the large areas of land that usually require continuous monitoring, which could prove to be a daunting task for humans to perform considering the amount of labour and sometimes harsh weather and environmental conditions [3].

IoT in agriculture has rightfully propelled what is now known as smart farming [4]. IoT in the agricultural sector involves the use of the sensors and actuators that detect physical quantities and convert these quantities to signals, these signals are harnessed by information and communication devices which transmit data to the Internet. It relies basically on data [5]. The data collected by these systems can be used to generate algorithms for machine learning systems, to make certain predictions [6]. In order to feed the world’s growing population, to combat adverse weather conditions and climate change, the use of the IoT in agriculture is the way forward.

1.1.1 Data in Farm Management

Although traditional methods of data collection and analysis still exist in some farms (mostly subsistence farms). In farm management, data keeping in recent times has greatly evolved from traditional methods of paper record-keeping and physical measurements of agricultural parameters to vast Farm Management Information Systems that comprise of data capture and transfer components [7]. This data could be used for various purposes, including some of the aforementioned. Data in smart farming is always diverse [8]; for example, these data could include information on soil types, the number of crops harvested, agricultural parameters (e.g. humidity) [9].

There are three types of data that can be generated; these include:

1. Process-mediated: This involves agricultural business data such as purchasing, harvesting etc.
2. Human Sourced Data: Human sourced information is information gotten from human interaction such as books, pictures, video etc.
3. Machine-generated Data: This includes data from sensors [9].

This project will deal with the use of machine-generated data.

1.1.2 Farm Management Information Systems

Farm management information systems are IoT agricultural systems. It is a category of management information system, but it is used for agricultural purposes. A management information system can then be defined as a system that brings together past and present information relating to the internal operations of the system, use this information to make future predictions as well as output the information collected to the external environment. They consist of data collection media, transmission media and also data output platforms such as web-based portals or mobile applications. This data can be used for different purposes, as required by the farmer [10].

The data collection media are sensors or a network of sensors known as wireless sensor networks. These sensors could include humidity, soil moisture, temperature, soil pH sensors and a few others. These sensors main objective is to be able to obtain their respective data without the intervention of a human operator [3]. These sensors can be used for a variety of agricultural monitoring purposes. Examples include animal monitoring, soil monitoring and plant monitoring which is involved in this project.

Sometimes data collection devices are linked up with certain control devices; this is where actuators come into use [3]. They are quite different from sensors which handle information in one direction alone. Actuators are bi-directional; this means that commands can be sent back and forth from the field. A pre-programmed algorithm is in charge of collecting sensor data and triggering the actuators when the pre-set condition or value is reached. Actuator devices include alarms, valves and pumps. These actuators help to regulate and control the supply of water, fertilisers, pesticides, light and temperature [3].

The transfer of data across the devices involved is very crucial to the success of the farm; therefore, the transmission medium consists of a network layer. This network carries the information gotten from all the sensors using different gateways and protocols[3]. For the output of the data, a web or mobile application can be used. More advanced farm management and information systems can perform functions of displaying geographical areas using GPS location trackers, show weather and climate patterns and more advanced functions [11].

1.1.3 Indoor Farming

For farmers to grow certain crops from other geographical regions and to avoid adverse temperature and harsh weather conditions, which can reduce harvests and increase plant mortality rate [12], indoor farming is practised, which includes:

1. Factory farming, where plants are grown without soil, and the sunlight is replaced by artificial light using light-emitting diodes (LEDs).
2. Greenhouses are controlled environments where plants are grown under controlled conditions for optimal development.

Indoor farming requires that agricultural parameters such as temperature, humidity, light etc. be maintained at optimal conditions; this ensures quality agricultural produce [13].

1.2 problem statement

Some of the problems encountered by commercial agricultural industries include data collection, data accessibility when needed and irrigation. Much time is wasted following traditional agricultural methods which could include manual record-keeping, physical measurement of agricultural parameters, harvesting and irrigation. This project provides solutions to save time lost engaging in traditional agricultural methods by providing a platform for data collection, transmission and data retrieval when needed.

**1.3 MOTIVATION OF THE STUDY**

With the rapid rise in the population of the world today, there is also an alarming increase in the demands of food supply to feed the ever-growing population. Advancements in agricultural engineering have made possible hybrids with the sole aim of bringing out the most desirable properties in plants and animals and also higher yields. Also, most crops are harvested by mechanical harvesters, although a few crops still require harvest by hand.

From the advancements mentioned above, it can be seen that food needs to be grown, monitored and harvested faster than before. Most commercial farms are so large that it becomes a daunting task for individuals to monitor, water and harvest these farms. Thus there is a need for a solution to these challenges.

1.4 aim and objectives

This project aims to design and build an accurate farm management information system that collects data from the field through the help of sensors that sense soil and environmental conditions. It displays the sensor data on a web portal, where registered users can log in and view data. The objectives of this study are to:

1. Design and implement an agricultural data collection system.
2. Develop a farm management web application that connects farmers with the farm sensor datasets.
3. Evaluate the performance of the developed data collection and web application.

1.5 Significance of the project

The results of this project will be beneficial to the following groups of people

1. Home Growers: The data that is collected could be accessed from any location, and in the event of the absence of the farmer, their farm will be fully autonomous.
2. Commercial Farmers: Larger spaces can be adequately monitored and controlled as it will reduce the cost of hiring workers, as well as save time.
3. Students: The project provides more information for further works and research on the subject area.
4. Researchers and Scientists: The data collected can be useful in determining optimal growing conditions for plants, and it will also help them in studies of disease activity in specific plant species.

1.6 METHODOLOGY

Four agricultural parameters, which are temperature, humidity, light intensity and soil moisture were obtained using their respective sensors. These sensors interface with the Arduino Uno. The Arduino microcontroller was programmed using the Arduino Integrated Development Environment (IDE). This was linked to the Internet via the NodeMCU device and the firebase server to receive the data. Python programming language was used in the development of a web-based portal, which makes agricultural parameters available to farmers and farm administrators. It enables users to sign in and log in to be able to view each of the agricultural parameters and data.

1.7 project organisation

Chapter one: Contains a general overview on the project, the background information, the aim and objectives of the project, what problems the project seeks to solve as well as a brief outline of the methodology.

Chapter two: Gives the literature review, it discusses the past related works on the project subject area. It contains theoretical background and other concepts necessary to make the project to be well understood.

Chapter three: Contains the methodology of the project, how the project is going to be carried out, design of the project, it contains all components required for the design and block diagrams and software design.

Chapter four: This chapter describes the implementation phase of the project as well as testing. The functional system design will be shown in detail as well as real pictures of the project undergoing testing. The results of the project will be analysed in this chapter.

Chapter Five: This is the last chapter of the project report, and it contains conclusions and recommendations for the project. It also shows the results the project was able to achieve.

**1.8 SUMMARY**

This chapter aims to introduce the concept of the Internet of things and their application in agriculture. Farm Management Information Systems, agricultural data and sensors have been explained.

The project’s problem statement, motivation, significance, methodology, as well as the aim and objectives of the project, are given here. The layout of the project organisation is given in brief. The introductions given in this chapter provide the necessary foundation for understanding the rest of this work.

chapter two

LITERATURE REVIEW

* 1. Introduction

To propel the realisation of smart cities of the future, every industry has to have its fair share of automation. This will be evident in the application of smart monitoring systems and automated replacements in all industries, including the agricultural sector. Over the years, intelligent and automated solutions in farming have witnessed extensive development. These continue to penetrate several applications from complex subjects like genetically modified crops and animals to simpler aspects like irrigation and monitoring. Due to the lack of sufficient irrigation provisions in various countries, irrigation solutions have been among the crucial points of interest for several studies. This study seeks to explore a new solution to monitoring the conditions of the soil and providing automated irrigation based on the observed conditions. Recent studies have explored different thematic areas in this domain which includes, but not limited to, automated solutions in the agricultural industry, application of intelligent soil monitoring systems in agriculture and automated irrigation solutions.

* 1. definItion of key terms

Some key terms related to farm management information systems are described below.

1. **Farm Management Information Systems:** They are a type of management information systems. These systems as their name implies are used for agricultural purposes, and they could cover horticulture, aquaponics, aquaculture, animal husbandry and all other forms of agriculture. A management information system can be described as a system that handles information concerning the operational capabilities of an organisation; it performs the necessary decision-making processes [14]. Thus, Farm management and information systems handle information and perform necessary decision making in the agricultural sector.
2. **Delivery Modes:** Farm management information systems have different delivery modes. These modes are the methods with which the software aspect meets the user. There are three of these; web, desktop, mobile or the combination of the three or two of them [15].
3. **Field Monitoring:** Consists of devices that help to capture the different agricultural parameters.
4. **Data Processing:** This is the process whereby data is converted to a form that is easily understandable by the farmer.
5. **Sensors:** These are devices that detect physical quantities and convert them to electrical signals [5]. A sensor would respond to changes in its environment. Examples include a gas sensor which responds when some specific gases are present in the environment, a light sensor which responds to changes in light intensity, a proximity sensor which responds to movement and so on. There are several types of sensor classifications. However, in this project, more emphasis is made on the analog and digital classification. The two main groups of sensors are described below.
6. Analogue sensors are a group of sensors which return an analogue output. Sensors that output values within a range of possible values are analogue sensors. An example is the soil moisture sensor.
7. Digital sensors are sensors that provide a digital output. An example is a proximity sensor.
8. **Actuators:** These are devices that regulate and control systems and activities. Actuator devices include alarms, valves, pumps etc.

2.2.1 Problems Facing the Wide Adoption of Farm Management Information Systems

A lot of factors hamper the adoption of farm management information systems, some of these factors are explained below.

1. Lack of understanding of the technology: A lot of farmers are not literate and do not know how to fully integrate their farms with modern systems. As a result, these farmers tend to shy away from modern technology. This, therefore, can pose a problem.
2. Communication difficulties: Although most farmers today are knowledgeable in the lingua franca of the nation they live in, some farmers in some rural communities only know how to speak their local languages, this could cause problems. Sometimes a system may not be available in some languages.
3. Cost: The cost of some of these systems may discourage farmers from acquiring them.
4. Internet connectivity: A farm management information system may require an internet connection. It may be impossible for farmers in rural areas to get internet accessibility.
5. Data management: A farm management system needs to be able to accommodate lots of data; therefore, a system needs to be efficient in data management. Non-efficient systems are not effective and will discourage further adoption of farm management information systems.

* 1. Review of RELATED WORKS

2.3.1 Review of the Definition and Formulation of the Research Problem

A number of research works have shown the necessity of encouraging the improvement and the adoption of smart farming practices across different agricultural platforms [16]. Management and Information System (MIS) is used to compile past, collect present and predict future information related to certain operations [14]. Such systems help in decision making. MIS for agriculture have been developed and can be used to compile data obtained from various aspects of a farm. MIS specifically created for farming applications are known as Farm Management and Information Systems (FMIS). FMIS helps to maximise the profit generated by farms [14].

Farm Management and Information Systems (FMIS) include features such as [14] :

i. Reporting features viii. Financial management

ii. Irrigation management ix. Labour management

iii. Data acquisition x. Data processing

iv. Condition management xi. Crop management

v. Field management xii. Decision support

vi. Collect produce information xiii. Planting management

vii. Risk analysis xiv. Technology management

A good number of FMIS in use have not received sufficient research considerations. In as much as further research should be encouraged on already studied FMIS, more attention should be given to already commercialised systems. Kaloxylos et al. [17], described a functional farm management system that incorporates future internet characteristics. The characteristics of the system included the integration of several services and the interconnection of these services with the networked devices.

Some of the setbacks faced by FMISs are given below:

i. Market penetration ix. Cost

ii. Data size x. Extra skill requirement for the farmer

iii. Data complexity xi. Scalability

iv. Time consumption xii. Data accuracy

v. Data management xiii. Data timeline

vi. Slow information upload xiv. Excess information

vii. Performance assurance xv. Privacy concerns

viii. Availability

Daum et al. [18], investigated the use of smartphones for data collection of agricultural and socio-economic data in small farms in rural Zambia. The study developed an application for smartphones that enables the user to collect different kinds of data from small farms and is compatible with devices that are affordable to those in rural Zambia. This application collects and analyses agronomic and socio-economic data on small farming systems. The investigation showed that smartphone apps can be used for data collection in small farm systems [18].

2.3.2 Review of Automated Solutions in the Agricultural Industry

Regardless of how healthy a soil sample is, weeds are in constant competition with plants, thereby reducing the nutrients available to plants and in turn, the productivity of the farm [19]. Weeding is therefore very crucial and can be done in various methods. Chemical destruction of weeds might be effective and fast, but it is also harmful to plant and animal communities. Besides, the excessive use of herbicide-based agriculture has led to the mutation of some weeds into Glyphosate-resistance agriculture [19].

Automatic weeding systems are already in use in the industry. These systems utilise crop detection and weed detection mechanisms. However, such systems lack a certain level of intelligence and end up harming weak plants in the rows where they are applied. Assirelli et al. [19], tested two sensors used to locate relatively fragile cuttings (such as tender poplar cuttings). The solution employed the divergence between the actual position and the response of the cuttings in question, to ascertain the accuracy of the detecting system. The study showed that the photoelectric system generated a response of lower amplitude in a row with poplar cutting than the capacitive sensor. In other words, the testing machine detected the cuttings with a high degree of accuracy [19].

One of the top concerns in the forestry and agricultural industry is plant diseases and pests [20]. Conventional solutions to plant disease and pests such as field scouting are expensive, ineffective and liable to bias. For this reason, remote sensing techniques are employed. These techniques date back to the 1980s. Different types of remote sensing solutions have been applied to detect plant health and disease, which engenders better productivity [20]. It goes without saying that although a good number of symptoms are common to several plant diseases, still, there are certain distinguishable symptoms observable in some disease cases [20]. Zhang [20], also considered the algorithms used to map remote sensing features to particular plant diseases. Colezea et al. [21], proposed a web platform to help increase the quality of farm produce, and it also supports business development. The platform has a social side in order to allow user interaction.

2.3.3 Review of the Application of Intelligent Monitoring Systems in Agriculture

Sensors are used in monitoring systems. In a review, [22] showed that there are several possible applications of sensors in agriculture which are undergoing feasibility proof stages. This precedes the launch of such technology into commercialisation and integration with already existing technologies [22]. However, more disruptive application of sensors awaits more progress in the sensor hardware technology. There have been a good number of advancements in sensor technology.

Lots of improvements have been channelled to boost the lifetime of sensors which cover better battery systems or battery charging systems (for instance solar energy collection system for charging the batteries) [23]. Fogarty et al. [22], reviewed the on-animal sensor and its use in sheep research. The review verified that the studies on on-animal sensor technology are increasing. Nevertheless, as expected, their actual application in real farming systems are still in the seeding phase [22].

Sensing systems are classified thus [20]:

1. Visible and near-infrared spectral sensors (VS-NIR)
2. Synthetic aperture radar sensors (SAR)
3. Fluorescence and thermal sensors
4. Ranging and light detection sensors (Lidar)

The sensors operate by observing various features as required. These features that are remotely sensed include:

1. Habitat suitability features
2. Fluorescence and optical features
3. Thermal features
4. Image-based features

Studies on monitoring systems have grown increasingly popular recently. There is an industry drive (as well as a global drive) towards greater degrees of autonomy. Among the several works towards this end, [24] proposed a monitoring system for micro-climate horticulture. The authors implemented a device for capturing images from the plants and created a database platform to store their results. This was done using an electronic sensing board to monitor parameters such as soil condition, water and air in the particular area.

Thakur et al. [25], designed a low-cost system to monitor the growth of Valeriana Jatamansi plant in a greenhouse. The system has the capability of sensing soil moisture and insect detection. It also has provisions for data transmission back to the farmer using a Wi-Fi module. This is achieved using an Arduino and a sensor device. Musat et al. [26], developed a system to enable farmers to manage their greenhouses effectively and also interact with other farmers. The system notifies farmers in case of any problems encountered, and it collects information. Zamora-Izquierdo et al. [27], proposed a flexible platform to deal with soilless recirculation greenhouses that use moderately saline water. This is achieved using low-cost hardware, supported by an open-source software platform that has three layers; local, cloud and the edge planes.

Ferentinos et al. [28], carried out an analysis of Wireless Sensor Networks (WSNs) on greenhouses and developed a monitoring system using wireless sensor networks in commercial greenhouses, the authors also performed an analysis of data collected to check for problems caused by climatic variations inside a greenhouse. The authors performed this by developing a prototype wireless sensor network to look into the consequences of environmental conditions to the reliability of WSNs, and analyse the performance and practicality of its operation in a commercial greenhouse.

Karimi et al. [29], designed a wireless sensor network for real-time collection and monitoring of micrometeorological data in vineyards. Monitoring the fumigation process of Sulphur (IV) oxide gas can also be done using the system developed. Among the several monitoring capabilities of this system, it was equipped to monitor the atmospheric parameters inside the grape drying structure. Harun et al. [13], developed a remote monitoring system to control indoor climatic conditions using light-emitting diodes parameter control and manipulation. Using Brassica Chinensis as a test crop, the authors observed that light of varying wavelengths influenced its development and phytochemical properties. This was done by developing an embedded system for automatic LED control and utilisation.

Srbinovska et al. [30], proposed a wireless sensor network system for vegetable greenhouses to lower costs. The authors performed this by designing a low-cost monitoring system for greenhouses based on WSN technology to monitor some environmental parameters, which included temperature, light and humidity. Trilles et al. [31], came up with a model used in vineyards to detect the best time to apply control treatments for downy mildew in grape plants. They achieved this using a low-cost sensor platform, which can monitor environmental data in order to use them for detection and treatments of disease in vineyards. Karim et al. [32], proposed a prototype for soil moisture level assessment using a wireless sensor network.

Evidently, lesser sophisticated methods like image capturing and so on will not provide connected solutions for future smart cities. Hence, IoT integrations are necessary to proffer even better monitoring options. In light of this, a lot of studies have been conducted to provide monitoring solutions with IoT capabilities. Dachyar et al. [2], reviewed advancements and major quantum leaps in the growth and development of the Internet of things during the last twelve years. In a study that highlighted the potential of Wireless sensor network technologies and the Internet of things in agriculture and challenges faced when incorporating these technologies to traditional farming techniques, [33] showed current IoT technologies available today.

Farooq et al. [34], investigated the current network technologies used in IoT based agriculture; it involves a close look at network topologies, layers, protocols and architecture. The study also looked at the interconnection of Internet of things agricultural systems with modern technologies such as big data and cloud computing, as well as investigated current security issues in using IoT. Khattab et al. [35], developed an IoT monitoring system linked up to the Internet that made use of wireless sensor networks which predicts outbreak of plant epidemic disease and also has the ability to store soil information in a database. The authors use artificial intelligence and algorithms to develop a system that emulates human decision-making ability. It sends warning messages to the concerned parties before the outbreak of the disease [35].

Kyaw and Ng [36], designed an IoT aquaponics system that creates an interdependent system which simultaneously runs plant cultivation alongside fish production. The system monitors the combined farm and notifies the farmer of any abnormal condition. It is also equipped to provide automatic solutions to any problem as they arise. This innovative solution was supported by a data collection capability. The authors also came up with web and mobile applications to display data and send notifications in the form of emails, and SMS in case of abnormalities. They combined the use of several types of sensors, microcontrollers, actuators and microprocessors, to control and monitor certain parameters. The aquaponics system also made use of emails, push notifications and short messages, sent to the user automatically when abnormal growing conditions are noticed by the sensors. There are also inbuilt cameras that allow the farmers to monitor the farm in real-time.

2.3.4 Review on Automated Irrigation Solutions

Agriculture is dependent on water supply (moisture). Irrigation is one of the few ways crops in arid regions can be provided with the moisture they require for growth [37]. Globally, it has been estimated that 20% of cultivated land is irrigated and produces 40% of the total food supply in the world. Human activity continues to lead to environmental crisis that depletes the freshwater reserves used for agriculture.

Irrigation is also subject to certain problems such as water stress. Without intelligent systems, crop water stress is monitored with measurements of soil moisture and other variables to approximate the water loss in the given soil sample; water balance calculations and measurement of plant water status [37]. This method is time-consuming and filled with idealistic assumptions (like the uniformity of plant density and equality in transpiration rate). Water stress control is very important for successful irrigation scheduling. Studies have focused on establishing the use of remotely collected data as a replacement to the archaic ways of measurement.

Weather changes also affect water conditions. This is predicted using evapotranspiration models. Plant water conditions are measured using Spectral vegetation indices [37]. At a wavelength of 900 nm, there is no absorption by water. Wavelengths as high as 1300-2500 nm are the water absorption bands showing the best sensitivity to water concentration in leaves. Lower wavelengths of 950-970 nm show weaker absorption [37].

In order to properly exploit out of the box intelligent irrigation systems, innovations in the monitoring of water stress are very key [37]. The authors pointed out advancements in detecting crop water stress. Muangprathub et al.[38], proposed a system for automatic irrigation using wireless sensor networks. The system can be manually controlled if the user so wishes. A web-based application was also designed to get field information as well as crop data.

Evidently, equipping farmers with the ability to monitor the moisture content and irrigation requirements of their farms, and providing customised nearly autonomous irrigation solutions for such farms will benefit in countless number of ways. Several studies have focused on the benefits of such autonomy in agriculture. Pathak et al. [39], developed an algorithm that allows allocation of water for farming under different environmental conditions. The system uses “Thingspeak” to display the sensor data on the cloud platform.

Irrigation monitoring is just one of the many faces of the applications of autonomous systems in irrigation. Irrigation prediction studies have also been conducted. These studies, although having different methods and specific objectives, all aim at predicting certain requirements of the farm in terms of irrigation. They make use of sensors, and most times, prediction is based on historical data already collected. Goap et al. [40], developed an open-source-based system to enable prediction of irrigation requirements of a farm area using soil sensors and environmental conditions along with the weather forecast data. The irrigation scheme was created to have a high level of autonomy, which was achieved by using closed-loop control of the water. The system uses a smart algorithm which takes into account sensed data as well as parameters such as precipitation, humidity temperature etc. for future use [40].In the study, the major focus is to develop a smart app-based soil monitoring system and integrate the system to an automated irrigation solution that will provide better autonomy to farmers in these areas for more advanced applications and opening doors to increased scale farming.

**2.4 SUMMARY**

In this chapter, a brief overview of intelligent soil monitoring, irrigation systems and other intelligent agricultural monitoring techniques have been discussed to provide a more rounded understanding of the processes involved and their expected results. Furthermore, previous studies in this regard as related to the application of intelligent monitoring and irrigation systems have been reviewed. From the studies reviewed above, proper control of soil conditions is of great concern in the industry. Also, productivity and efficiency in farming are greatly dependent on soil conditions and irrigation. The general opinion is that the best solutions for soil condition improvement lie in smart systems. Hence, human error and limitations can be overcome. Several soil properties and environmental conditions have to be monitored, and data stored. To improve the conditions of the soil, the present study will focus on improving the solution to soil condition monitoring and automated irrigation.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.1 INTRODUCTION

This chapter addresses the process involved in the development of farm management and information system. The various components used to build the hardware system are discussed. The schematics, as well as the different components used, are reviewed and the relevant diagrams shown, where necessary. The software aspect for the web portal is also discussed in this chapter, with appropriate UML diagrams.

3.2 SYSTEM HARDWARE

The hardware design was implemented using the Arduino Uno microcontroller. Other microcontrollers could also be used to implement this project; some of them include Evie, bolt, Raspberry pi and a few others. The Arduino Uno was chosen for its simplicity and wide availability of resources. Below is a table showing the different components that make up the hardware system and their voltage ratings.

Table 3.1: Table showing the different hardware components and their voltage ratings [41-44]

|  |  |
| --- | --- |
| Electronic components | Voltage specification (VDC) |
| Arduino Uno | 5V |
| Soil moisture sensor | 3.3V – 12V |
| Light sensor | 2.7V – 3.6V |
| NodeMCU | 3V – 3.6V |
| Buzzer | 3V – 5V |
| Temperature and humidity sensor | 3.3V – 5.2V |
| Water pump | 5V |

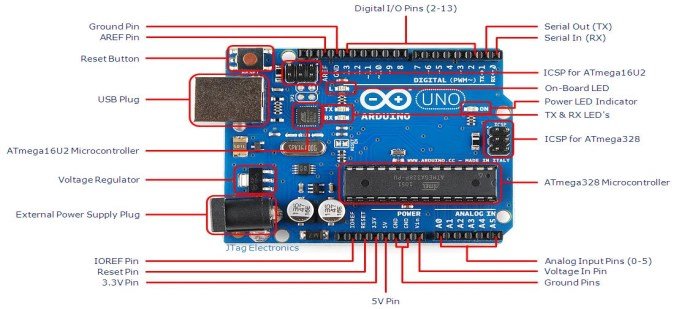
3.2.1 Arduino UNO

Arduino is an open-source platform that gives anyone a chance to create electronic projects on their own. The hardware comes in different sizes and layouts known as boards. Some of the famous boards available are the Uno, Mega, and Yun; each board, has some differences. Boards can also be used with other support hardware known as shields. Shields perform a more specific function. Examples of standard Arduino shields include the Ethernet shield and the Wi-Fi shield. The shield is usually made such that it can fit on the top of the Arduino board.

The Arduino open-source is not restricted to the hardware alone; it also has an Integrated Development Environment (IDE) which uses a variant of the C language for the code. This software aspect allows the user to program their boards to perform specific actions. This project makes use of an Arduino Uno board. The Arduino Uno is a type of Arduino board. It has been used for a variety of projects by hobbyists and professionals around the globe due to its simplicity and ease of use.

The Arduino Uno has 14 digital pins and 6 analogue pins. The digital pins can be used for both input and output, but they have only two states, HIGH or LOW. A buzzer and an LED are examples of devices that return a digital output; to turn on an LED, for instance, the digital pin should be set to HIGH, likewise to turn off the LED the digital pin should be set to LOW. Therefore, such devices that return a digital output should be connected to the digital pins of the Arduino Uno. Analogue pins return a value from 0 to 1023. An example of an analogue sensor is a temperature sensor, and it displays temperature values.

The Arduino Uno has a USB port for powering up the board via a computer or laptop and an ICSP header. It has a power supply port which serves as an additional method for powering up the board. The board has a small red button, which is known as the reset button and performs the task of enabling the user to reset the board from the previous code. The Uno board has an LED that comes on when the board is powered and blinks during new code upload. Two voltage pins support two different voltages, 3.3V and 5V. The heart of the board, however, is the Atmega328 microcontroller - the brain behind the computing. The most important advantage of having the Atmega328 on the Arduino board is that the microcontroller makes it possible for the Arduino to interpret analogue signals because the Atmega328 microcontroller has an inbuilt analogue to digital converter.

Figure 3.1: Arduino Uno [43]

3.2.2 TSL2561 Digital Luminosity Sensor

It is an advanced type of light sensor that converts light intensity to a digital output; it finds its use in a variety of projects. The TSL2561 digital luminosity sensor is quite different from other types of photodiode sensors in that it contains infrared diodes and full spectrum diodes. Other types of light sensors only have one type of diode and only detect either infrared light or full-spectrum light waves. The standard operating voltage of the device is from a minimum of 2.7V to a maximum voltage of 3.6V. The device contains two integrating Analogue to Digital Converters (ADCs); these perform the function of integrating currents from the two photodiodes included in the device.

Table 3.2: The terminals on the device[42]

|  |  |
| --- | --- |
| Terminals | Description |
| Vcc | Supply voltage |
| GND | Ground |
| SCL | SMBus serial clock input terminal |
| SCA | SMBus serial data input/output terminal |
| INT | Open drain |

Table 3.3: The terminal type either an input or output or none[42]

|  |  |
| --- | --- |
| Terminals | Type (Input or Output) |
| Vcc | N/A |
| GND | N/A |
| SCL | Input |
| SCA | Input/output |
| INT | Output |

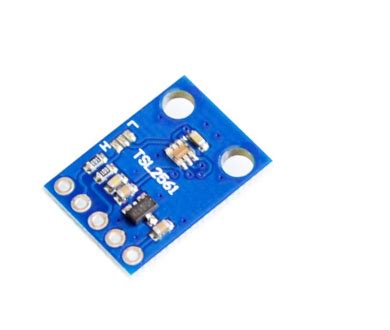


Figure 3.2: Front-view of a Digital Luminosity Sensor [45]

****

Figure 3.3: Back-view of a Digital Luminosity Sensor [45]

3.2.2 Resistive Corrosion-Resistant Soil Moisture Sensor

Soil plays a significant role in agriculture. The type of soil used in farming could affect crop productivity and crop yield. Thus, measuring soil parameters is key to success in the agricultural sector. The soil moisture sensor measures the volumetric content of water contained in the soil sample and presents the moisture level as an output. It is equipped with both analogue and digital outputs, and therefore it can be used in both modes.

The probes of the soil moisture sensor allow an electric current to move through the soil, and then it receives the soil resistance value to measure the moisture content. It is standard knowledge that water is a good conductor of electricity. Therefore, when the soil has more water, it conducts more electricity. Soil moisture would be higher with lower resistance. The reverse is the case for dry soil as the dryer the soil, the less the electrical conduction and the more the resistance. The soil moisture sensor makes use of the LM393 comparator the potentiometer on the comparator can be adjusted to increase or reduce the sensitivity of the probes. The sensor operates at a voltage of 3.3V to 12V.

Figure 3.4: Resistive Corrosion-Resistant Soil Moisture Sensor [46]

3.2.3 DHT22 Temperature and Humidity Sensor

The DHT22 temperature and humidity sensor is composed of a thermistor and capacitive humidity sensor enclosed in a plastic case. It contains four pins. It also has an analogue to digital converter on the board, within the plastic enclosure. The sensor measures the atmospheric air condition and sends a digital signal via the digital pin. It has a drawback of being 2 seconds slow; that is, it takes approximately two seconds for the sensor to get new values. Operating voltage is from a minimum of 3.3V to 5.5VDC max.

Table 3.4: Table showing parameters and their sensitivity [44]

|  |  |
| --- | --- |
| Parameter | Sensitivity |
| Temperature | -40 c to 80 c |
| Humidity | 0 to 100% RH |

Table 3.5: Table showing DHT pins and their description [44]

|  |  |
| --- | --- |
| Pins | Description |
| Vcc | Power supply |
| SDA | Serial data port |
| NULL | Null Pin |
| GND | Ground |

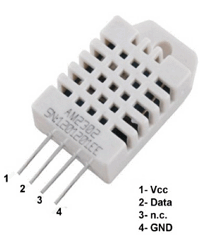
**

Figure 3.5: DHT22 Temperature and Humidity Sensor [44]

3.2.4 NodeMCU

The NodeMCU is an open-source IoT platform manufactured by Esspressif systems; it runs on the ESP8266 Wi-Fi SoC (Wi-Fi system-on-chip)[41]. The abbreviation MCU stands for Microcontroller unit. It is a small, compact device ideal for small systems like mobile phones and robots. Its main purpose is connecting these systems to the Internet.

The NodeMCU has the following features[41]:

1. Compactness
2. High durability
3. Low power consumption
4. Integrates a tensilica L106 32-bit RISC processor

****

Figure 3.6: NodeMCU Device [47]

3.2.5 Hardware Block Diagram

The Hardware connection diagram is shown in Figure 3.7. The light and the soil moisture sensors are connected to different analogue pins on the Arduino Uno board. This is because these sensors return an analogue output that can be read by the user. The analogue pins can return a value from 0 to 1023. The NodeMCU can be connected to digital pins 11, 12, and 13. These are Wi-Fi supported pins; the Arduino communicates with the Wi-Fi shield using the SPI bus.

Digital pin 7 is not used because it serves as a handshake pin between the Arduino and the device. The buzzer and the DHT temperature, and humidity sensor are connected to any of the other remaining digital pins; this is because the buzzer and the DHT sensor are digital devices. A HIGH turns on the buzzer and the LOW turns off the buzzer. The pump is connected via a relay to one of the Arduino Uno’s digital pins. A HIGH turns on the pump and a LOW, turns it off. The principle is that when the soil moisture level falls below the pre-set value, the pump comes on and waters the plant.

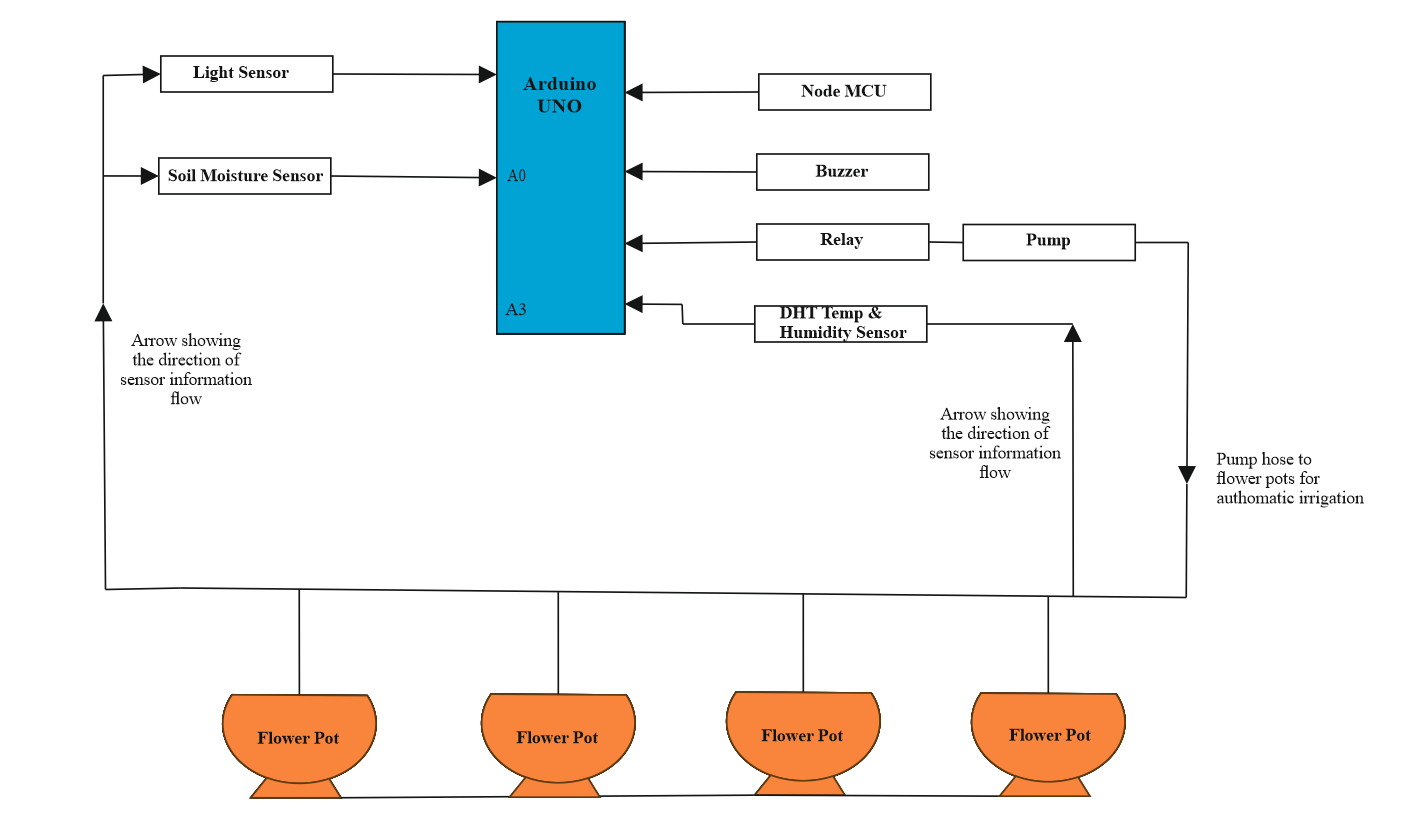
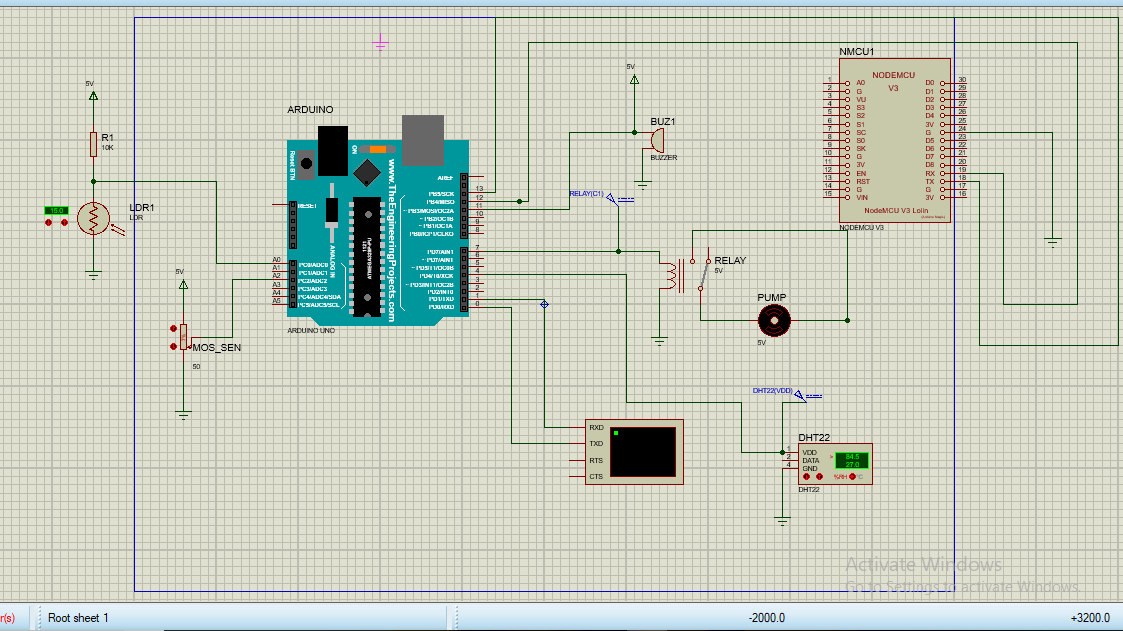


Figure 3.7: Hardware Connection Block Diagram

3.2.6 Circuit Simulation

From the hardware block diagram in Figure 3.7, the circuit diagram can be obtained, as shown in Figure 3.8 below. Proteus design suite version 8.8 was the tool used to develop the circuit diagram and perform the circuit simulation. The Proteus software application is an electrical/electronic design software used to design and generate schematics for printed circuit boards (PCB). It is used for electrical circuit testing. It saves time and resources that could be lost in manual components connection. For example, if a component is connected wrongly, it would only produce errors, unlike a physical component that might get damaged. The Proteus Arduino library and the Esp8266 library were added Proteus, as these libraries are not inbuilt in the Proteus software. The various components of the hardware system were selected form the component menu on the Proteus development package. Each component is selected and dragged to an appropriate part of the development space. The connection then proceeds, as shown in the hardware connection block diagram in Figure 3.7.

The central component the Arduino Uno is connected to the light sensor on the analogue pin ‘A0’ and the soil moisture sensor on analogue pin ‘A2’ on the Arduino Uno shown on the left side of Figure 3.8. The temperature and humidity sensor was connected to digital pin 4 since it is a digital sensor. The buzzer which has 2 states (HIGH or LOW) is connected to digital pin 11; ‘HIGH’ turns the pump ON while a ‘LOW’ turns it OFF. The water pump represented by the motor symbol is connected to digital pin 7, via a relay. It is a digital device because it also has 2 states (HIGH or LOW). ‘HIGH’ turns the pump ON while a ‘LOW’ turns it OFF. The NodeMCU device is connected to digital pin 12. The virtual monitor is connected to pins 0 and 1 to provide an interface for the values to be read during the simulation.

Figure 3.8: Circuit Diagram

3.3 System SOFTWARE

In recent times, software systems have gotten more extensive and complicated due to a variety of factors such as advancements in technology and increasing customer demands [48]. The objective of this aspect of the project is to develop a web portal. The elements needed for the development of the web portal are discussed in this section. The application developed for this project is called the SmartAgrio web portal. For its development, the python programing language was used with the Django framework. Elements of HTML and CSS were included as websites consist of HTML and CSS files. The text editor and IDE used was Pycharm, which is the most popular Python IDE. Other python IDEs include Spyder and IDLE [49]. Firebase is the database used to receive data from the Arduino and send it to the web portal.

3.3.1 Python

This is a multipurpose high-level programming language. It was developed in the late 1980s by Guido Van Rossum, a Dutch computer scientist [50]. The first version of python was released in 1991. The python language consists of three main entities.

1. Variables
2. Functions
3. Programs

Python is a very popular programming language used by developers. It has the following advantages, which makes it desirable:

1. Its several libraries meet user’s specific need.
2. It can use shorter lengths of code
3. It has a vast ecosystem when compared to other programming languages; these include frameworks and libraries.
4. It is cross-platform
5. It does not have issues with memory management.
6. It is open-source and beginner-friendly

Python programming language has the following features [49];

1. It is concise and easy to use.
2. It makes use of modules, which are pieces of code that can be imported to another program.
3. It has a straight forward approach to object-oriented programming.
4. It has a lot of built-in data types.
5. It can be used in conjunction with other programming languages. For example C and C++.

3.3.2 Django

Django is a python framework. It saves developers from performing software projects from scratch. Django is used mostly for developing web applications.. For a web application, instead of the user to create all the APIs and RSS feeds from scratch, Django is a more straightforward, faster solution. Compared to other frameworks, Django can handle more users, and it has unique features such as API connectivity and User login. Also, it does not make use of structured query language (SQL).

3.3.3 Unified Modelling Language

To reduce software design complexity, software architecture has been brought up to separate complex, large software systems into smaller components [48]. Kruchten [51], proposed that software architecture can be viewed from different perspectives. The different viewpoints postulated by [43] are;

1. Logical
2. Process
3. Development
4. Physical

From the logical viewpoint, software system is separated into components and their relationships. From the process viewpoint, the interaction of the elements of the system is the main focus. Development focuses on issues related to the implementation. From the physical viewpoint, the main focus is on the hardware and how they are connected. Software can be modelled with various viewpoints using various modelling languages, which include Network Description Language (NDL), and Unified Modelling Language (UML). The most common software language, however, is the UML.

UML is a third-generation object-oriented modelling language. Its initial version was released in November 1997 [52]. The UML diagrams aid in the representation of complex software systems in visual form. UML diagrams can be grouped into two main categories. They are static diagrams and dynamic diagrams [52]. Static diagrams are concerned with the structure of the system. Some examples include object diagram, deployment diagram, profile diagram and class diagram. A dynamic diagram is concerned with how the system changes. Examples include timing diagrams, activity diagrams, and use case diagrams.

There are four basic types of UML diagrams, and they are as follows:

1. The first kind of UML diagram is structural diagrams. The main focus is its purpose. Structural diagrams include class diagrams, structure diagrams, object diagrams, package diagrams, component diagrams and deployment diagrams.
2. Functionaldiagrams focus on functionality, not on the behaviour or structure. Examples are case diagram and information flow diagram.
3. Interaction diagrams focus on how the elements of the system relate together; examples of interaction diagrams include sequence, collaboration and timing diagrams.
4. Behavioural diagram focuses on the behaviour of the system. Examples of behavioural diagrams include state machine and activity diagrams.

However, complex systems can be modelled with three major diagrams. These are; Class, State, and Sequence diagram. [52]. It should also be noted that different UML diagrams describe different views of a system and are dependent on each other; thus, they need to be consistent with each other. This project makes use of the activity diagram, sequence diagram and the use case diagram.

The advantages of UML include;

1. It is quite easy to learn, and the notation is of a graphical nature.
2. It is a well-defined language
3. It offers an extensive tool support

3.3.4 Use Case Diagram

The use case diagram shows the system or application and the basic flow of events that happen in the system. The lines show the interaction between the actors and the system; the actors are the users; they initiate the use case [53], which is always represented with an oval shape.

In the use case diagram, there are two types of actors:

1. The primary actor initiates the use of the system, and they are always positioned at the left-hand side of the system.The primary actors in this system are those who make use of the web portal; in this case, the farmers.
2. The secondary actors create a user-friendly environment. The secondary actor is always positioned at the right-hand side of the system.

Each actor interacts with the different use cases as indicated in figure 3.9. From the use case diagram in figure 3.9, the primary actor can log in or register, view the dashboard to access the corresponding agricultural data and log out.

Figure 3.9: Use Case Diagram

The administrative personnel’s use case interaction is shown in Figure 3.10. The administrative personnel can log in and manage user profiles, data, view user sessions and logout.

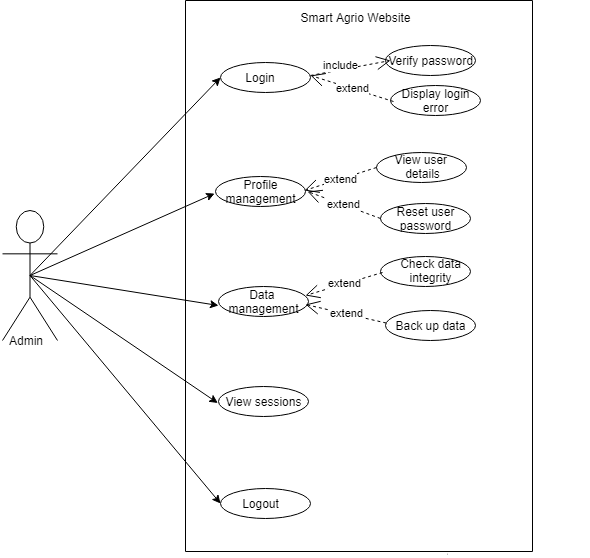


Figure 3.10: Use Case Diagram for Web Administrator

3.3.5 Class Diagram

Class diagrams are one of the most common UML diagrams available [54]. A class defines the properties and behaviours of objects that they describe [54]. Each class has fields that hold values in the object; the fields are known as slots [54]. The class diagram for the web portal is shown in figure 3.11 below.

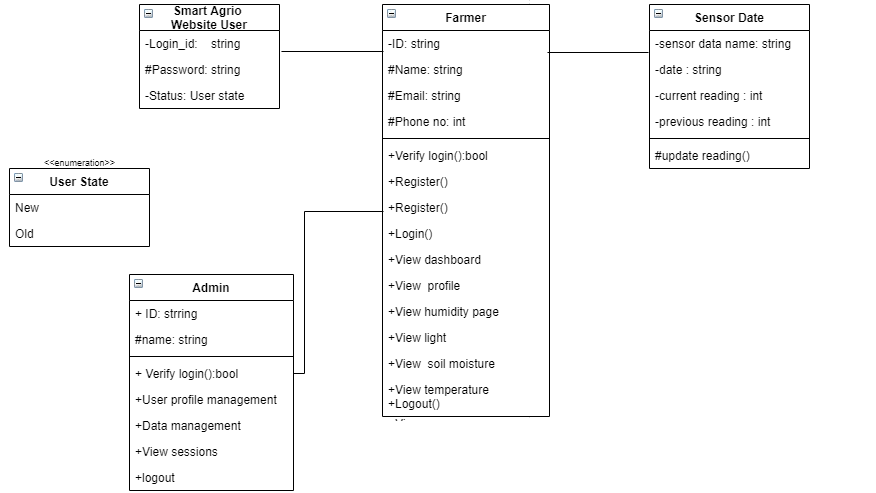


Figure 3.11: Class Diagram

3.3.6 Activity Diagram

The activity diagram shown in figure 3.12 represents a series of actions or flow of the control of the system. It is in the form of a flow chart diagram [55]. As a UML behavioural diagram, it shows how the objects in the system can work together [55]. From the diagram above, at the start of the process, the user is authenticated for registration, then there is a branch which contains two events that happen simultaneously. These events are condition-based and would occur if a particular condition is met. The user can either login in if he/she is registered or register as a new user. Another branch occurs in the new user registration. Once the user can log in, they are then able to access the dashboard and proceed to get the information they need. The branch is then merged, and then another branch begins. If no more actions are necessary, the user can safely log out. If more actions are required, the user is redirected back to the dashboard.

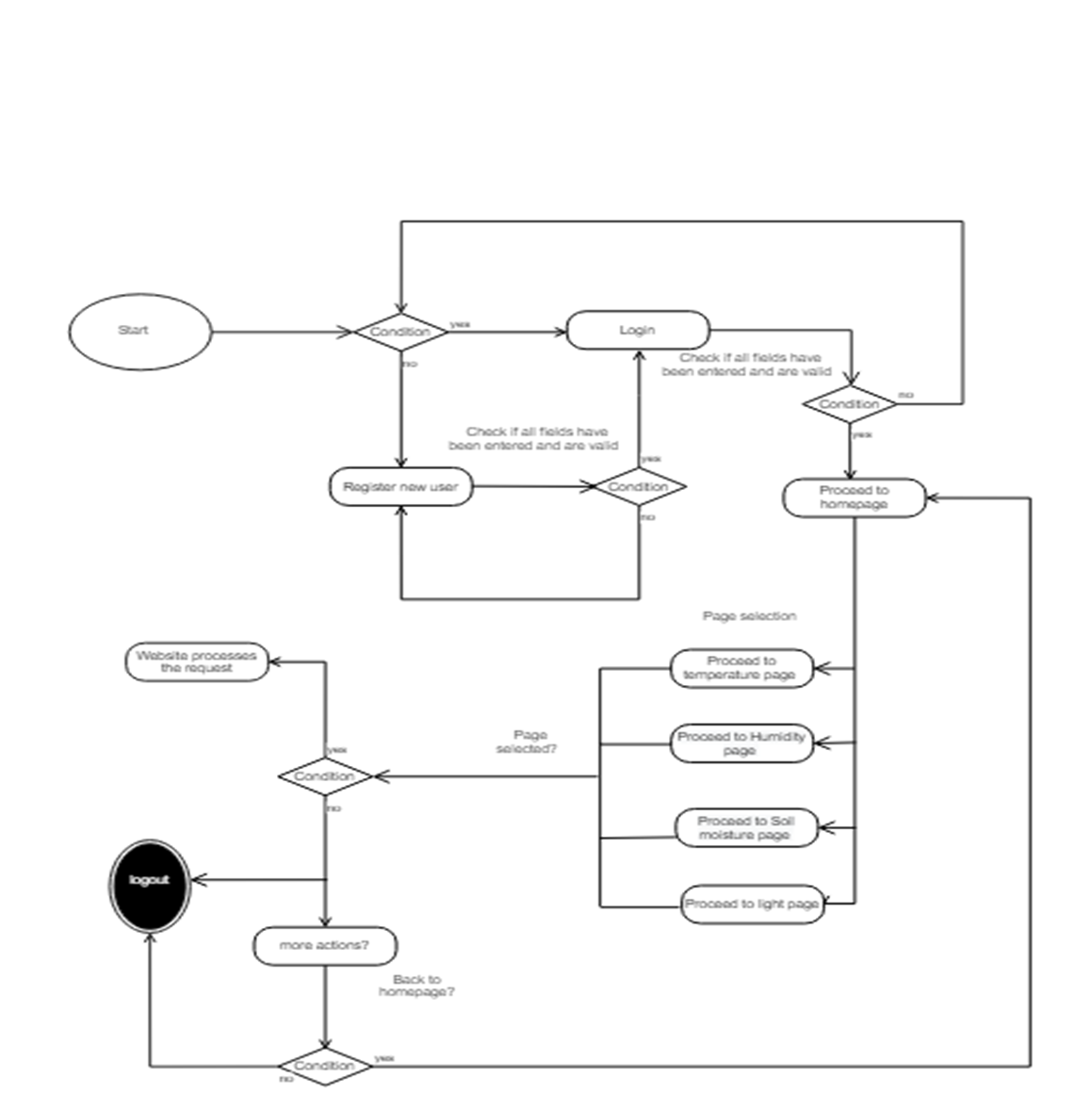


Figure 3.12: Activity Diagram

3.3.7 Sequence Diagram

The sequence diagram describes how the elements of the system communicate, that is, the exchange of messages between them [56]. The diagram is read from left to right and from top to bottom in the sequence in which the events happen. The actor (the user) is represented by the figure on the top left corner of the diagram. The elements involved in the interaction are placed side by side. From the diagram, we can see the actors’ or objects’ existence over a period of time; when each event represented by a line ends, another event begins. The figure below shows the communication between the user and the elements of the system. The user logs in, and the web portal validates the data. This is represented with a solid line; the elements performing the actions are represented in [56]. The dashed line however represent the necessary reactions [56].

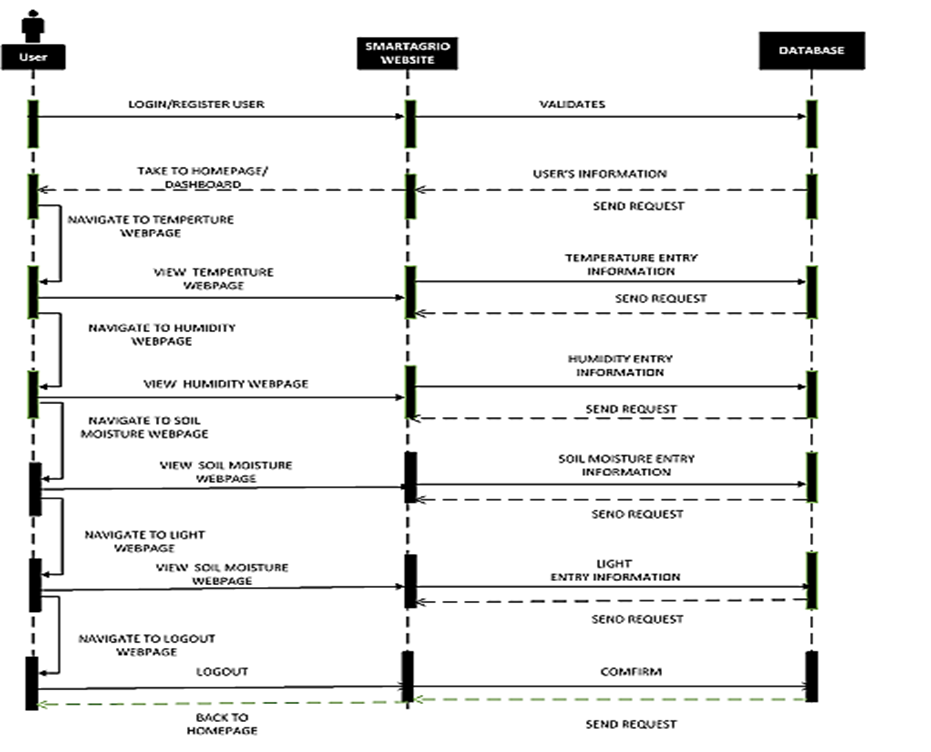


Figure 3.13: Sequence Diagram

3.4 system interaction

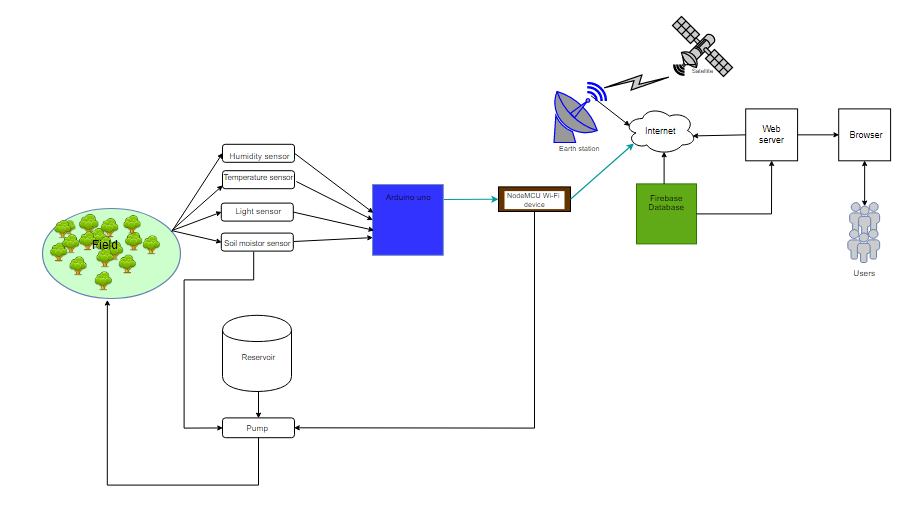
The system interaction is shown in figure 3.14 below. The sensor data collected from the field is passed to the microcontroller, through the Wi-Fi device to the firebase database on the Internet. The web portal collects data from the database and displays the results of the sensor field data, which can be accessed by the users when they log in to the web portal.

Figure 3.14: System Interaction Diagram

**3.5 SUMMARY**

The various components used to create this system were discussed in this chapter. Both hardware and software designs were presented. For the hardware; the electronic components, the block diagram and the circuit diagram were explained. For software development; the programming language used was explained, and the software architectural models were explored.

chapter 4

SYSTEM IMPLEMENTATION AND TESTING

4.1 INTRODUCTION

This chapter highlights the work that has been done during the development process of this project. The implementation of every building block that makes up this system and performance of the system as a whole and its components are tested. This chapter shows the results of the work that has been carried out at various sections of the project.

4.2 Circuit Simulation results

The hardware block diagram design was used to determine the correct placement of components and their connections and also determine the feasibility of the pre-drawn design. Before the simulation could take place the Arduino Uno was uploaded with the code from the Arduino IDE. This was done by attaching the “.HEX” or “.ELF” at the bottom of the verbose output. It was then run and the necessary debugging done. The Proteus simulation is shown in figure 4.1 below.

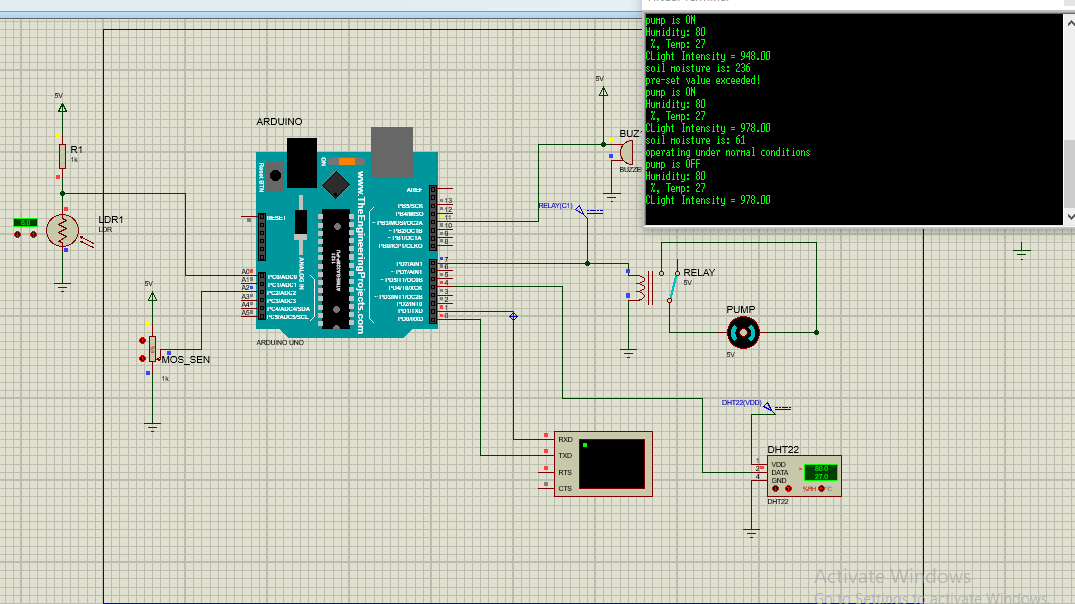


Figure 4.1: Proteus Simulation of Hardware Circuit

4.2.1 Arduino Programming

The Arduino IDE was used for this work. The Arduino IDE already installed on the system was updated to the latest version, Arduino 1.8.13. All the necessary libraries were installed such as the DHT, Adafruit and Esp8266 libraries. The code development workspace and the setup were used to run the initial code, while the loop was used to run the code repeatedly. The Arduino code was run, compiled and the errors debugged. The libraries were included where necessary using the ***# include*** function. Figure 4.2 shows a snippet of the Arduino code.

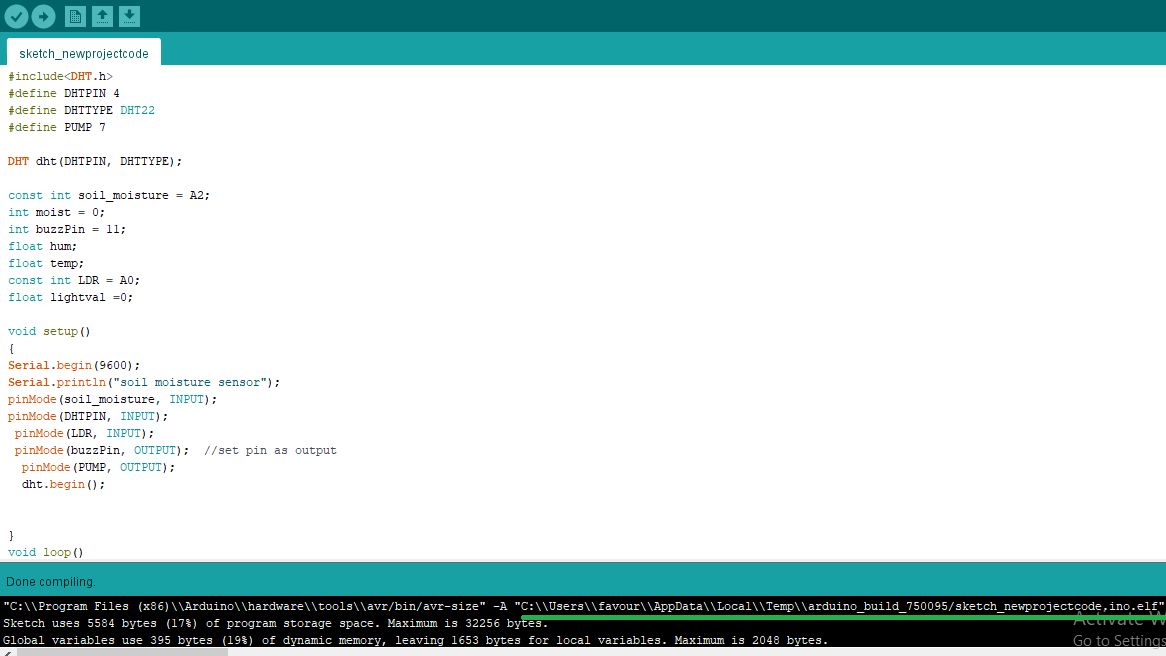


Figure 4.2: Snippet of Arduino Code Highlighting the ‘.elf’ file in Green

4.3 Web Portal implementation

The SmartAgrio web portal is a web portal that allows the farmer to check the values of certain agricultural parameters. It has an authentication system allowing users to log in with a username and a password, and log out. From the registration tab, new users can create a new account. Once logged in, the users can then proceed to access their required information. The users can view the current readings as well as previous readings of the required parameters.

4.3.1 Python software

Python with the Django framework was used for the creation of the web portal. The following were installed on the system:

1. Python
2. Pycharm text editor
3. Django

Django packages were installed via the command’ *pip install Django’* on the command line. Then the *django\_admin* was used to find the sub-commands available. The new project was created by selecting the start project sub-command, i.e. *django\_admin* start project.

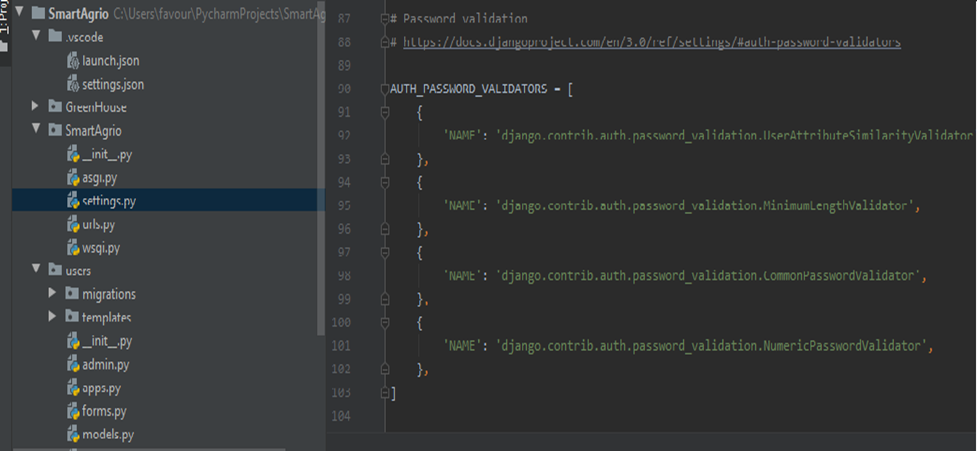


Figure 4.3: Snippet of Python Code

4.3.2 Farmer Section of the SmartAgrio Web Portal

The first page of this section of the web portal is the login page in Figure 4.4, followed by the sign-up page for a new user in Figure 4.5. After the user logs in, the first view page is shown in Figure 4.6. The user can navigate and select the required action from the dashboard. The figure shows the contents of the dashboard. Figures 4.7, 4.8, 4.9 and 4.10 show the corresponding agricultural parameter pages (light, temperature, humidity and soil moisture). Figure 4.11 shows the user profile page.

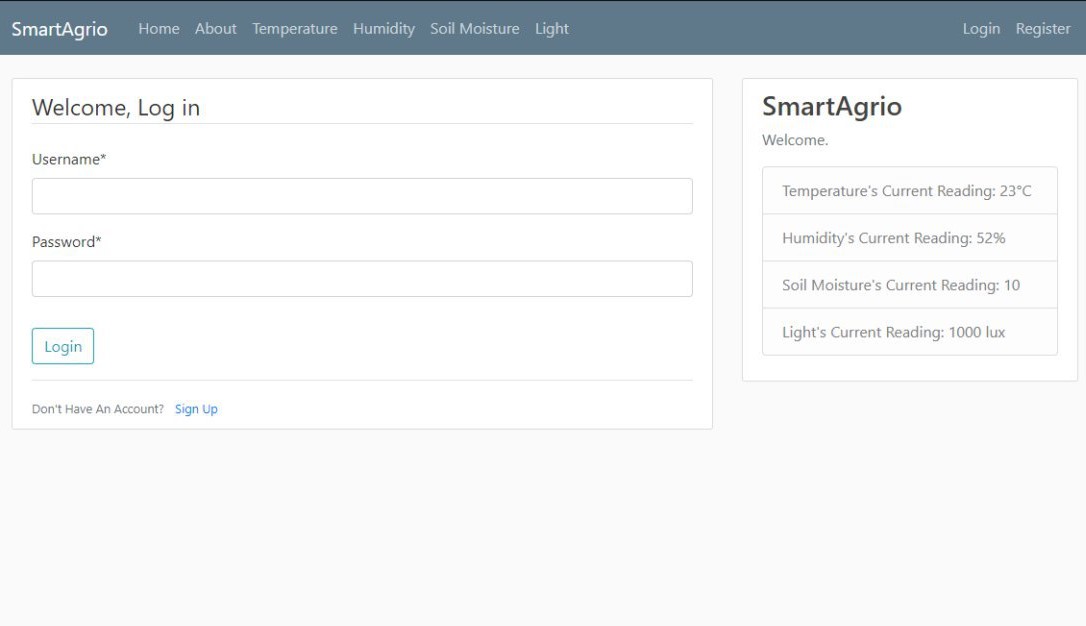


Figure 4.4: SmartAgrio Login Page

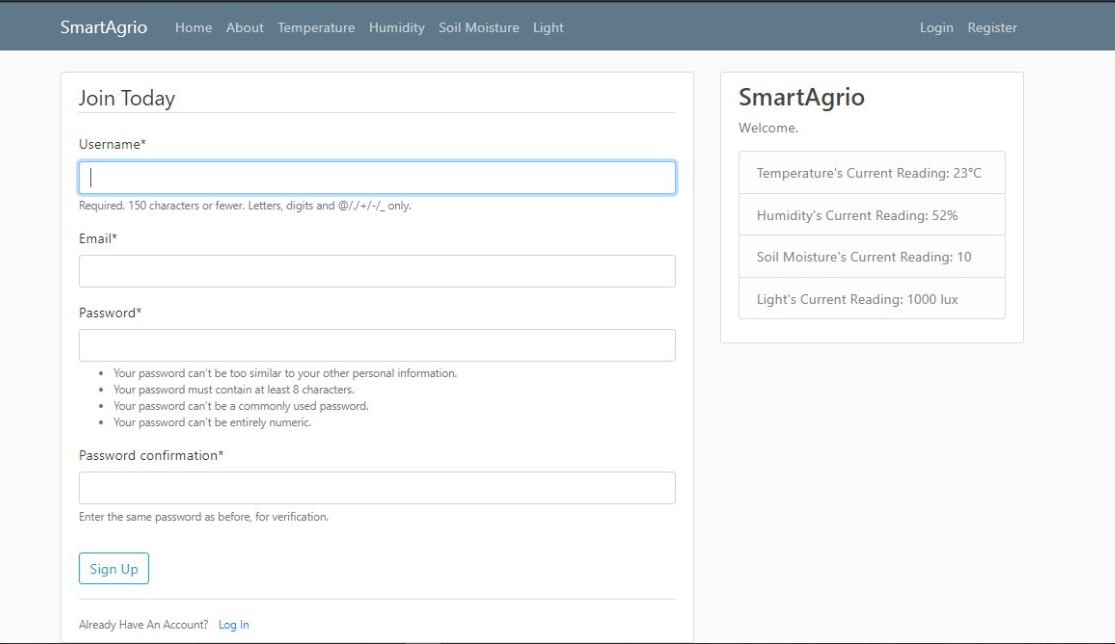


Figure 4.5: Registration Prompt on the SmartAgrio Web Portal

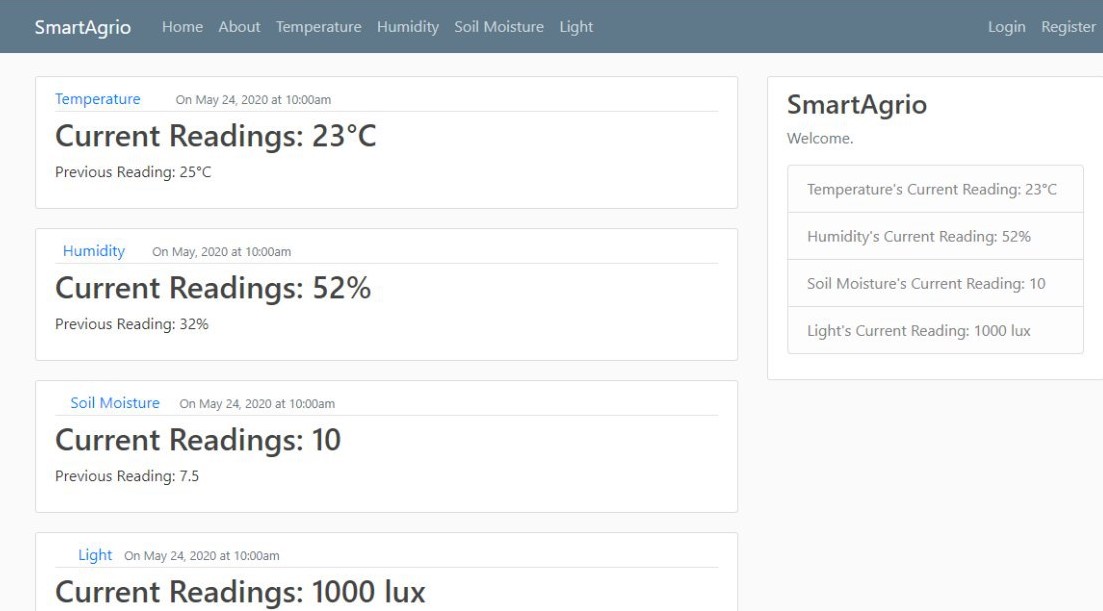


Figure 4.6: Current Readings Display Page on the SmartAgrio Webpage

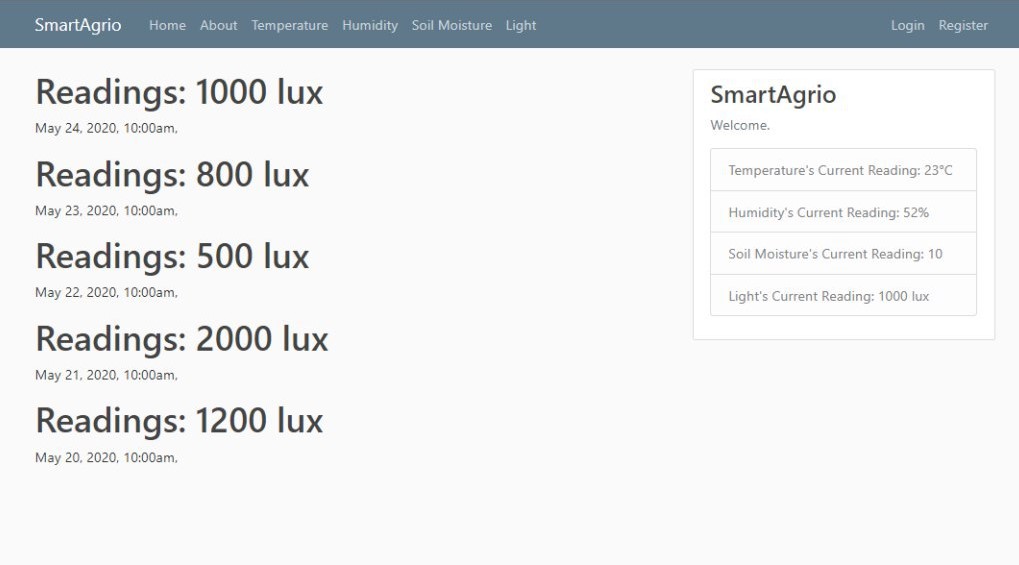


Figure 4.7: Light Readings

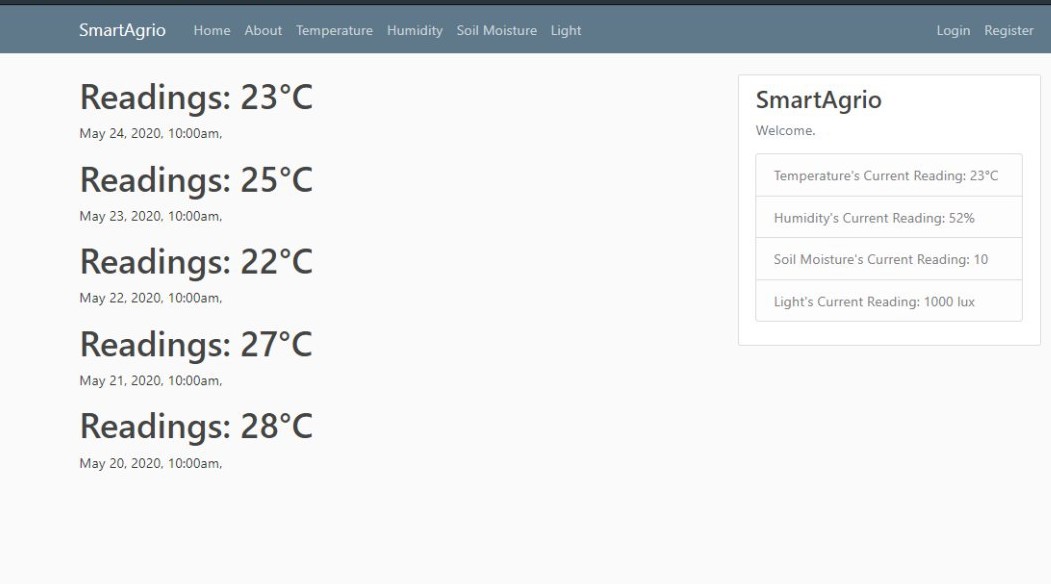


Figure 4.8: Temperature Readings

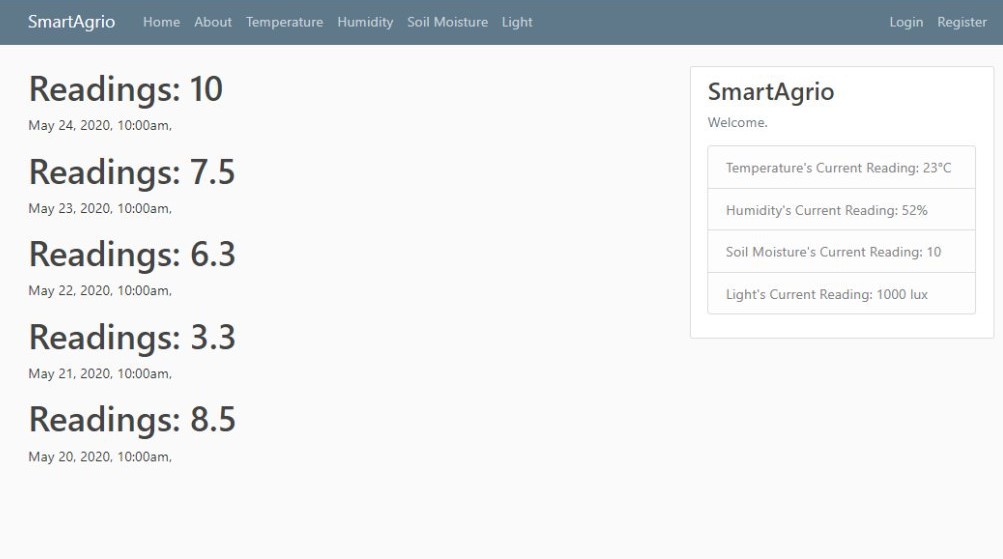


Figure 4.9: Humidity Readings

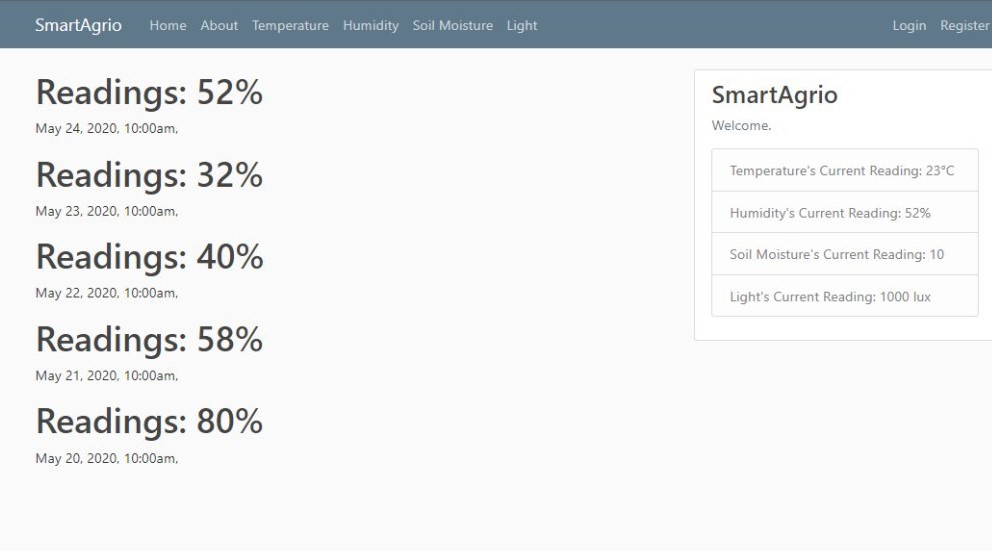


Figure 4.10: Soil Moisture Readings

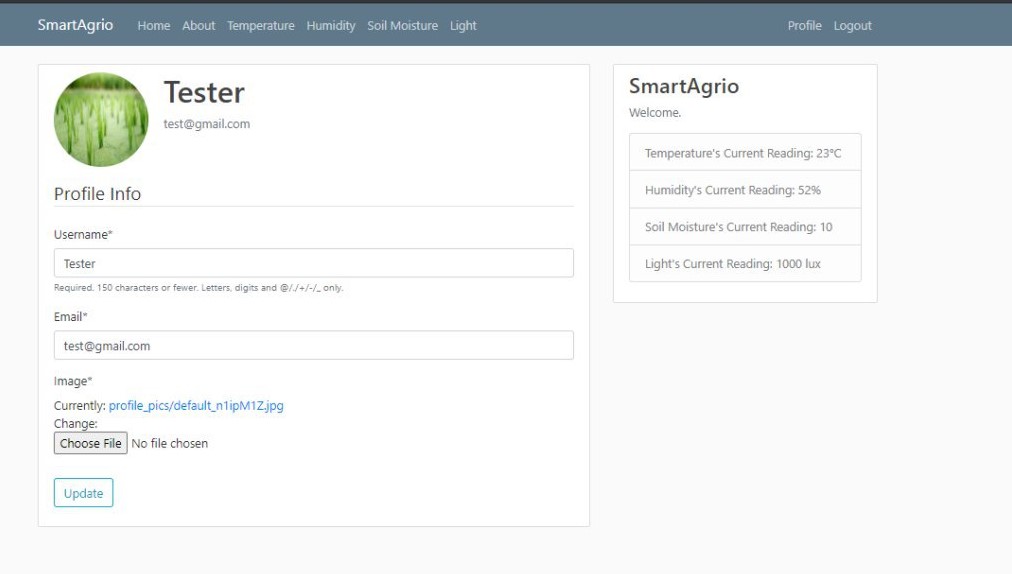


Figure 4.11: Test User Profile Page

4.3.3 Administration Section of the SmartAgrio Web Portal

The Administration page of the web portal is shown in Figure 4.12. The administrator can log in and can perform user profile management; reset passwords, check user-profiles and user activity. The administrator can also perform data management and check the integrity of the data sent from the sensors in the field.

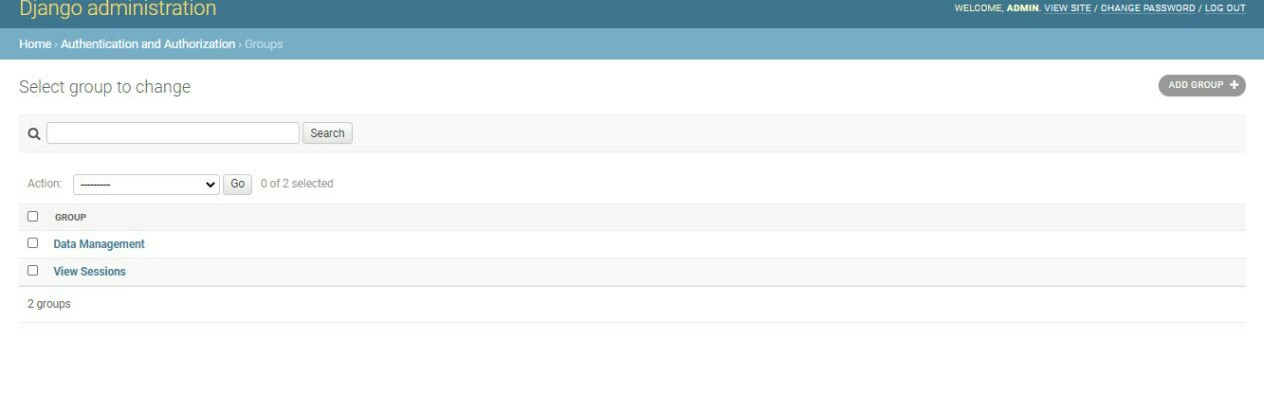


Figure 4.12: Administrator Page

4.4 Software testing

Tests were carried out on the software to ensure the quality and performance of the software [57]. These tests involve processes carried out to ensure it works as intended by the developers [58]. Software testing deals with the verification of the quality of the system [59], it may or may not involve code execution [58]. The following are some reasons the software is tested [57]:

1. To check the performance
2. To check for the reliability
3. To do quality assurance checks.

4.4.1 Unit Testing

The unit testing involves testing each unit of the code and then each unit of the software system; as the unit testing employs a bottom-up approach [57], each unit tested is integrated with the other test units. The approach used for the unit testing of this software system was a manual approach, which is convenient for the size of this system [51]. Each element of code was debugged and verified to ensure they worked correctly. Each section of the web portal was also tested. The user session, as well as the admin section, was tested. Figure 4.13 below shows the free online tool - the “Nu HTML checker” used to check the HTML codes for errors. Figure 4.13 shows the temperature page when first checked with the “Nu HTML checker”. The Nu HTML checker performs checks on HTML code against the correct HTML code standards and produces errors if the code does not meet up with these standards. The resulting checks produced errors; hence bug fixes were performed until there was no error; this can be seen in Figure 4.14.

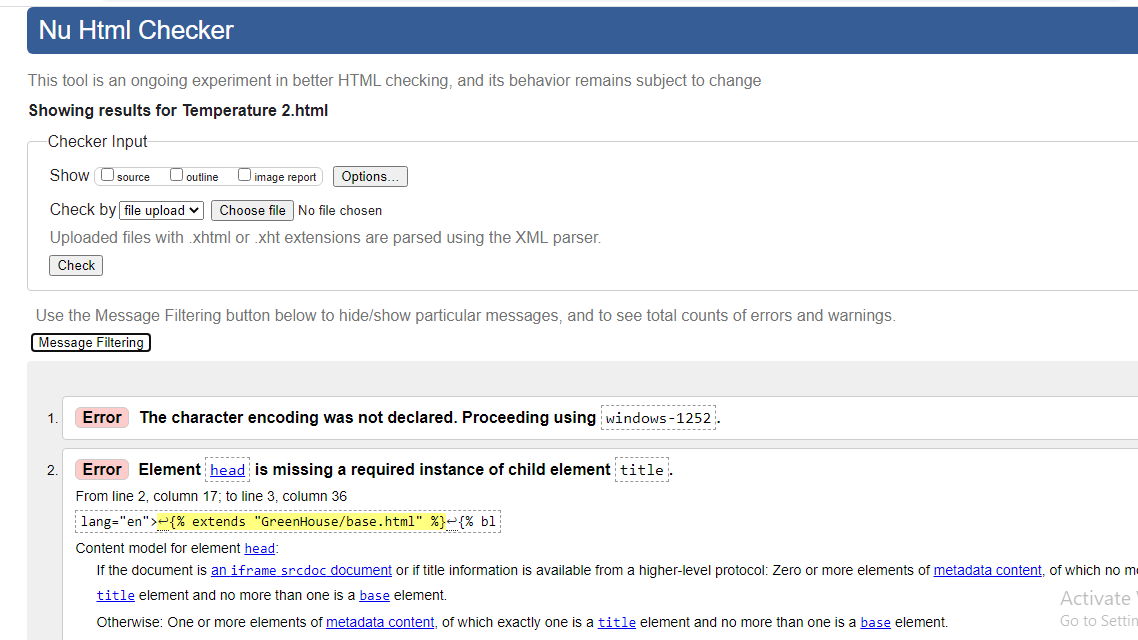


Figure 4.13: HTML Testing with the Nu Html Checker before Bug Fixes



Figure 4.14: Nu Html checker after Bug Fixes

4.4.2 Deployment Testing

Before a system or application is deployed, certain checks are carried out to ascertain that the software would perform well when eventually deployed. Such tests include unit testing, user interface testing, and functionality testing, amongst others. The locally hosted site was tested on different browsers, Microsoft Edge, Google Chrome and Mozilla Firefox to ensure it runs efficiently on all the browser platforms. Other tests, such as grammatical and spelling checks, were performed.

4.4.3 Usability Testing

Usability tests were carried out to ensure that the web portal is user friendly and easy to use. The response was collected using Google forms. The results of this testing showed the users view about their interaction with the web portal.

4.4.4 User Interface Testing

The user interface of any system is one of the most important parts of the system, as it is the part of the system that the user directly interacts with, thus user feedback should be an essential part of user interface testing. For the user interface test, 15 respondents were asked to partake in the survey to test the web portal’s user interface and give suggestions on what could be improved upon in future works. A Google form was created for this purpose in the form of a mixed questionnaire that contained open-ended and closed-ended questions. The questionnaire is discussed in details in the next section.

4.4.5 Questionnaire

The questionnaire was created using Google forms. The following questions were asked;

1. Occupation
2. The overall perception of the web portal user interface
3. Overall user experience.
4. Suggestions for improvement.

The results of the questionnaire are shown in the graphs below. The respondent’s categories are as follows: 9 students, 5 graduates, and 1 non-classified respondent, shown in figure 4.15. The overall perception of the web portal, user interface and user experience are shown in figure 4.16 and figure 4.17. It can be gathered from the results that most of the user reviews were positive. The questions asking respondents for their ideas for improvement was an open-ended question which received suggestions mostly on graphics, animations and a chat-bot. Other respondents were indifferent. For the open-ended question on the participant’s user experience, respondents asked for android application versions and graphs. The user interface testing using Google forms met its objective.

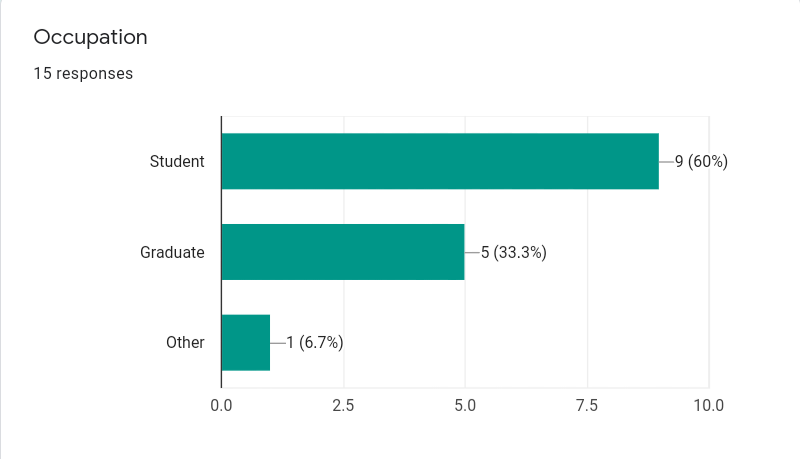


Figure 4.15: Participants Occupation

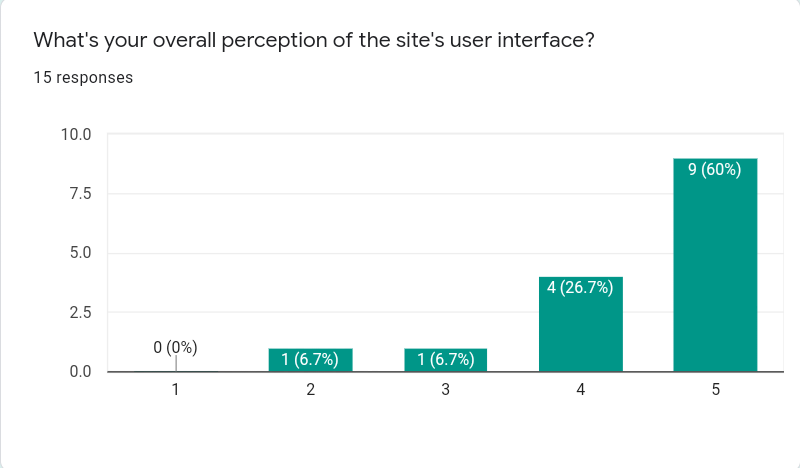


Figure 4.16: Participant Perception of the Web portal

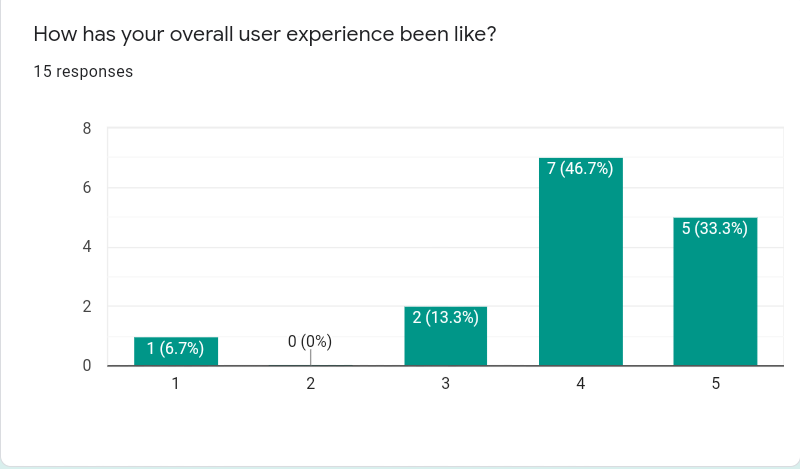


Figure 4.17: User Experience Response

**4.5 SUMMARY**

The results of this project are contained in this chapter. Circuit simulation of the hardware components and an overview of the code development was shown. The images from the web portal developed were shown here. Software testing was carried out to ensure quality and to test the performance of the web portal.

CHAPTER 5

Conclusions and recommendations

5.1 SUMMARY

The farm management information system in this project was designed to help farmers remotely run their farms. The system helps farmers with irrigation and daily data logs of 4 agricultural parameters (humidity, temperature, light, and soil moisture) on the Internet, which the farmer can check once he/she logs in to the web portal. The hardware system can be installed on a farm (enclosed in such a way to prevent damage from environmental conditions) or a greenhouse. The sensors contained in the system sense the agricultural parameters and transmit this data to a web portal that users can log in daily to view data. Also, with soil moisture, for values greater than or equal to 200, the alarm and pump are turned ON. The alarm and pump go OFF after soil conditions have been restored to normal.

5.2 ACHIEVEMENTS

This project implemented the design of a farm management information system with the aim of remotely running the farm without the farmer needing to be physically present. The hardware system comprises of sensors and a microcontroller; it also contains a pump and a relay for protection and control. When the water content of the soil is low, the soil moisture sensor detects it, and this triggers the buzzer alarm, and the pump comes ON and waters the plants. The pump goes OFF after the soil returns to normal. The NodeMCU device transmits the sensor data to the Internet, where the farmer or user can log in to monitor the data daily via a web portal.

5.2.1 Challenges Encountered

The following challenges were encountered during the implementation of this project:

1. Unavailability of components to build the hardware system due to the coronavirus pandemic.
2. Interrupted power supply.
3. Simulation software works better on a more powerful operating system.

5.2.2 Limitations

Every system is subject to limitations as no system is 100% efficient. The following are certain limitations associated with the system.

1. The device must be housed in such a way as to prevent damage to the electronic components due to environmental factors, such as weather.
2. It should be protected to prevent damages from animals such as rodents.
3. Sensor errors, leading to incorrect data.
4. Network failure that might lead to users not being able to access the required data for a period of time.
5. Corrosion of metallic parts of the components.
6. A constant supply of electricity is required.

5.3 CONCLUSION

The implementation of a farm management information system will be useful in residential, industrial and commercial applications. Certain challenges faced by farmers would be greatly reduced if the methods employed in this project are put into place. It would also help in the conservation of time and resources.

5.4 RECOMMENDATION

The project was able to meet its objectives. There is a need for improvement on the functionality of the system. Some additional features that were not implemented in this project that could be added to later works are:

1. A field on-site camera module to record real-time images.
2. A mobile application, where users can check data from their devices.
3. Automatic daily SMS message that shows the data log for the day.
4. The web portal could be integrated with a chat-bot system.

REFERENCES

[1] W. Yong, L. Shuaishuai, L. Li, L. Minzan, L. Ming, K. G. Arvanitis*, et al.*, “Smart Sensors from Ground to Cloud and Web Intelligence,” *IFAC-PapersOnLine,* vol. 51, pp. 31-38, 2018/01/01/ 2018.

[2] M. Dachyar, T. Y. M. Zagloel, and L. R. Saragih, “Knowledge growth and development: internet of things (IoT) research, 2006–2018,” *Heliyon,* vol. 5, p. e02264, 2019/08/01/ 2019.

[3] J. M. Talavera, L. E. Tobón, J. A. Gómez, M. A. Culman, J. M. Aranda, D. T. Parra*, et al.*, “Review of IoT applications in agro-industrial and environmental fields,” *Computers and Electronics in Agriculture,* vol. 142, pp. 283-297, 2017/11/01/ 2017.

[4] H. Sundmaeker, C. Verdouw, S. Wolfert, and L. Pérez Freire, “Internet of food and farm 2020,” *Digitising the Industry - Internet of Things Connecting Physical, Digital and Virtual Worlds,* vol. 2, pp. 129-151, 2016.

[5] D. Pivoto, P. D. Waquil, E. Talamini, C. P. S. Finocchio, V. F. Dalla Corte, and G. de Vargas Mores, “Scientific development of smart farming technologies and their application in Brazil,” *Information Processing in Agriculture,* vol. 5, pp. 21-32, 2018/03/01/ 2018.

[6] R. Saville, K. Hatanaka, M. Sano, and M. Wada, “Application of information and communication technology and data sharing management scheme for the coastal fishery using real-time fishery information,” *Ocean & Coastal Management,* vol. 106, pp. 77-86, 2015/03/01/ 2015.

[7] S. Fountas, G. Carli, C. G. Sørensen, Z. Tsiropoulos, C. Cavalaris, A. Vatsanidou*, et al.*, “Farm management information systems: Current situation and future perspectives,” *Computers and Electronics in Agriculture,* vol. 115, pp. 40-50, 2015/07/01/ 2015.

[8] K. Ishii, “Big data analysis medical, agriculture, in the environmental field,” *Seibutsu-kogaku Kaishi,* vol. 92, pp. 92-93, 2014.

[9] S. Wolfert, L. Ge, C. Verdouw, and M.-J. Bogaardt, “Big Data in Smart Farming – A review,” *Agricultural Systems,* vol. 153, pp. 69-80, 2017/05/01/ 2017.

[10] C. G. Sørensen, S. Fountas, E. Nash, L. Pesonen, D. Bochtis, S. M. Pedersen*, et al.*, “Conceptual model of a future farm management information system,” *Computers and Electronics in Agriculture,* vol. 72, pp. 37-47, 2010/06/01/ 2010.

[11] R. Nikkilä, I. Seilonen, and K. Koskinen, “Software architecture for farm management information systems in precision agriculture,” *Computers and Electronics in Agriculture,* vol. 70, pp. 328-336, 2010.

[12] S. Niu, Y. Luo, D. Li, S. Cao, J. Xia, J. Li*, et al.*, “Plant growth and mortality under climatic extremes: An overview,” *Environmental and Experimental Botany,* vol. 98, pp. 13-19, 2014/02/01/ 2014.

[13] A. N. Harun, N. Mohamed, R. Ahmad, A. R. A. Rahim, and N. N. Ani, “Improved Internet of Things (IoT) monitoring system for growth optimization of Brassica chinensis,” *Computers and Electronics in Agriculture,* vol. 164, p. 104836, 2019/09/01/ 2019.

[14] J. Tummers, A. Kassahun, and B. Tekinerdogan, “Obstacles and features of Farm Management Information Systems: A systematic literature review,” *Computers and Electronics in Agriculture,* vol. 157, pp. 189-204, 2019/02/01/ 2019.

[15] A. Wasserman, *Software engineering issues for mobile application development*, 2010.

[16] M. Bacco, P. Barsocchi, E. Ferro, A. Gotta, and M. Ruggeri, “The digitisation of agriculture: A survey of research activities on Smart Farming,” *Array,* p. 100009, 2019/11/05/ 2019.

[17] A. Kaloxylos, R. Eigenmann, F. Teye, Z. Politopoulou, S. Wolfert, C. Shrank*, et al.*, “Farm management systems and the Future Internet era,” *Computers and Electronics in Agriculture,* vol. 89, pp. 130-144, 2012/11/01/ 2012.

[18] T. Daum, H. Buchwald, A. Gerlicher, and R. Birner, “Smartphone apps as a new method to collect data on smallholder farming systems in the digital age: A case study from Zambia,” *Computers and Electronics in Agriculture,* vol. 153, pp. 144-150, 2018/10/01/ 2018.

[19] A. Assirelli, P. Liberati, E. Santangelo, A. Del Giudice, V. Civitarese, and L. Pari, “Evaluation of sensors for poplar cutting detection to be used in intra-row weed control machine,” *Computers and Electronics in Agriculture,* vol. 115, pp. 161-170, 2015/07/01/ 2015.

[20] J. Zhang, Y. Huang, R. Pu, P. Gonzalez-Moreno, L. Yuan, K. Wu*, et al.*, “Monitoring plant diseases and pests through remote sensing technology: A review,” *Computers and Electronics in Agriculture,* vol. 165, p. 104943, 2019/10/01/ 2019.

[21] M. Colezea, G. Musat, F. Pop, C. Negru, A. Dumitrascu, and M. Mocanu, "CLUeFARM: Integrated web-service platform for smart farms," *Computers and Electronics in Agriculture,* vol. 154, pp. 134-154, 2018/11/01/ 2018.

[22] E. S. Fogarty, D. L. Swain, G. Cronin, and M. Trotter, “Autonomous on-animal sensors in sheep research: A systematic review,” *Computers and Electronics in Agriculture,* vol. 150, pp. 245-256, 2018/07/01/ 2018.

[23] H. Sharma, A. Haque, and Z. A. Jaffery, “Maximization of wireless sensor network lifetime using solar energy harvesting for smart agriculture monitoring,” *Ad Hoc Networks,* vol. 94, p. 101966, 2019/11/01/ 2019.

[24] E. R. Kaburuan, R. Jayadi, and Harisno, “A Design of IoT-based Monitoring System for Intelligence Indoor Micro-Climate Horticulture Farming in Indonesia,” *Procedia Computer Science,* vol. 157, pp. 459-464, 2019/01/01/ 2019.

[25] D. Thakur, Y. Kumar, A. Kumar, P. Kumar, and V. Singh, “Real Time Monitoring of Valeriana Jatamansi Plant for Growth Analysis,” *Procedia Computer Science,* vol. 132, pp. 507-517, 2018/01/01/ 2018.

[26] G.-A. Musat, M. Colezea, F. Pop, C. Negru, M. Mocanu, C. Esposito*, et al.*, “Advanced services for efficient management of smart farms,” *Journal of Parallel and Distributed Computing,* vol. 116, pp. 3-17, 2018/06/01/ 2018.

[27] M. A. Zamora-Izquierdo, J. Santa, J. A. Martínez, V. Martínez, and A. F. Skarmeta, "Smart farming IoT platform based on edge and cloud computing," *Biosystems Engineering,* vol. 177, pp. 4-17, 2019/01/01/ 2019.

[28] K. P. Ferentinos, N. Katsoulas, A. Tzounis, T. Bartzanas, and C. Kittas, “Wireless sensor networks for greenhouse climate and plant condition assessment,” *Biosystems Engineering,* vol. 153, pp. 70-81, 2017/01/01/ 2017.

[29] N. Karimi, A. Arabhosseini, M. Karimi, and M. H. Kianmehr, “Web-based monitoring system using Wireless Sensor Networks for traditional vineyards and grape drying buildings,” *Computers and Electronics in Agriculture,* vol. 144, pp. 269-283, 2018/01/01/ 2018.

[30] M. Srbinovska, C. Gavrovski, V. Dimcev, A. Krkoleva, and V. Borozan, “Environmental parameters monitoring in precision agriculture using wireless sensor networks,” *Journal of Cleaner Production,* vol. 88, pp. 297-307, 2015/02/01/ 2015.

[31] S. Trilles, J. Torres-Sospedra, Ó. Belmonte, F. J. Zarazaga-Soria, A. González-Pérez, and J. Huerta, “Development of an open sensorized platform in a smart agriculture context: A vineyard support system for monitoring mildew disease,” *Sustainable Computing: Informatics and Systems,* 2019/01/24/ 2019.

[32] F. Karim, F. Karim, and A. frihida, “Monitoring system using web of things in precision agriculture,” *Procedia Computer Science,* vol. 110, pp. 402-409, 2017/01/01/ 2017.

[33] M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour, and E. M. Aggoune, “Internet-of-Things (IoT)-Based Smart Agriculture: Toward Making the Fields Talk,” *IEEE Access,* vol. 7, pp. 129551-129583, 2019.

[34] M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem, “A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming,” *IEEE Access,* vol. 7, pp. 156237-156271, 2019.

[35] A. Khattab, S. E. D. Habib, H. Ismail, S. Zayan, Y. Fahmy, and M. M. Khairy, “An IoT-based cognitive monitoring system for early plant disease forecast,” *Computers and Electronics in Agriculture,* vol. 166, p. 105028, 2019/11/01/ 2019.

[36] T. Y. Kyaw and A. K. Ng, “Smart Aquaponics System for Urban Farming,” *Energy Procedia,* vol. 143, pp. 342-347, 2017/12/01/ 2017.

[37] S. O. Ihuoma and C. A. Madramootoo, “Recent advances in crop water stress detection,” *Computers and Electronics in Agriculture,* vol. 141, pp. 267-275, 2017/09/01/ 2017.

[38] J. Muangprathub, N. Boonnam, S. Kajornkasirat, N. Lekbangpong, A. Wanichsombat, and P. Nillaor, “IoT and agriculture data analysis for smart farm,” *Computers and Electronics in Agriculture,* vol. 156, pp. 467-474, 2019/01/01/ 2019.

[39] A. Pathak, M. AmazUddin, M. J. Abedin, K. Andersson, R. Mustafa, and M. S. Hossain, “IoT based Smart System to Support Agricultural Parameters: A Case Study,” *Procedia Computer Science,* vol. 155, pp. 648-653, 2019/01/01/ 2019.

[40] A. Goap, D. Sharma, A. K. Shukla, and C. Rama Krishna, “An IoT based smart irrigation management system using Machine learning and open source technologies,” *Computers and Electronics in Agriculture,* vol. 155, pp. 41-49, 2018/12/01/ 2018.

[41] Components101. (2020). *NodeMCU ESP8266*. Available: <https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet>

[42] A. D. Sheet. (2009). *TSL2561 Datasheet*. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/view/203054/TAOS/TSL2561.html>

[43] S. Sarkar, G. Ghosh, A. Mohanta, A. Ghosh, and S. Mitra, *Arduino based foot pressure sensitive smart safety system for industrial robots*, 2017.

[44] Pinout. (2016). *DHT22 Datasheet*. Available: <http://www.datasheetcafe.com/dht22-datasheet-pdf/>

[45] R. Parts. (2013). *TSL2561 Images*. Available: <https://roboparts.ru/catalog/datchiki/datchiki_sveta/gy-2561-tsl2561/>

[46] Ebay. (2020). *Soil Humidity Hygrometer Moisture Detection Sensor Corrosion Resistance Probe*. Available: <https://www.ebay.com/p/2179956927?iid=362072686633>

[47] Components101. (2020). *Nodemcu Image*. Available: <https://components101.com/sites/default/files/inline-images/NodeMCU-ESP8266.jpg>

[48] M. Ozkaya and F. Erata, “A survey on the practical use of UML for different software architecture viewpoints,” *Information and Software Technology,* vol. 121, p. 106275, 2020/05/01/ 2020.

[49] M. Rocha and P. G. Ferreira, “Chapter 2 - An Introduction to the Python Language,” in *Bioinformatics Algorithms*, M. Rocha and P. G. Ferreira, Eds., ed: Academic Press, 2018, pp. 5-58.

[50] H. Alyuruk, “Chapter 1 - Introduction to R and Python,” in *R and Python for Oceanographers*, H. Alyuruk, Ed., ed: Elsevier, 2019, pp. 1-21.

[51] P. B. Kruchten, "The 4+1 View Model of architecture," *IEEE Software,* vol. 12, pp. 42-50, 1995.

[52] B. P. Douglass, “Chapter 1 - Introduction to UML,” in *Real-Time UML Workshop for Embedded Systems (Second Edition)*, B. P. Douglass, Ed., ed Oxford: Newnes, 2014, pp. 1-32.

[53] M. J. Chonoles, “Chapter 12 - Use Cases,” in *OCUP Certification Guide*, M. J. Chonoles, Ed., ed Boston: Morgan Kaufmann, 2018, pp. 207-225.

[54] M. J. Chonoles, “Chapter 6 - Objects and Classes,” in *OCUP Certification Guide*, M. J. Chonoles, Ed., ed Boston: Morgan Kaufmann, 2018, pp. 89-113.

[55] M. J. Chonoles, “Chapter 16 - Behavior: Activity Diagrams,” in *OCUP Certification Guide*, M. J. Chonoles, Ed., ed Boston: Morgan Kaufmann, 2018, pp. 271-295.

[56] M. J. Chonoles, “Chapter 14 - Behavior: Sequence Diagrams,” in *OCUP Certification Guide*, M. J. Chonoles, Ed., ed Boston: Morgan Kaufmann, 2018, pp. 243-257.

[57] M. Pitchford, “9 - Embedded Software Quality, Integration, and Testing Techniques,” in *Software Engineering for Embedded Systems (Second Edition)*, R. Oshana and M. Kraeling, Eds., ed: Newnes, 2019, pp. 269-338.

[58] M. Pitchford, “Chapter 15 - Embedded Software Quality, Integration and Testing Techniques,” in *Software Engineering for Embedded Systems*, R. Oshana and M. Kraeling, Eds., ed Oxford: Newnes, 2013, pp. 441-510.

[59] F. Lonetti and E. Marchetti, “Chapter Three - Emerging Software Testing Technologies,” in *Advances in Computers*. vol. 108, A. M. Memon, Ed., ed: Elsevier, 2018, pp. 91-143.