Internet of Things and Farm Management Information System for Precision Agriculture: A Proof of Concept Circuit, Simulation and Web App

Emmanuel Adetiba[1,2,3]*, Chiebuka Favour Olumba[1], Ayodele H. Ifijeh[1,2], Etinosa Noma-Osaghae[1], Joy N. Adetiba[4], James Ameh[1], Oluwatosin A. Matthews[5]

[1]Department of Electrical and Information Engineering, Covenant University, Ota, Ogun State, Nigeria
[2]Covenant Applied Informatics and Communication Africa Center of Excellence, Covenant University, Canaanland, P.M.B 1023, Ota, Nigeria
[3]HRA, Institute for Systems Science, Durban University of Technology, P.O. Box 1334, Durban 4000, South Africa
[4]Lagos State College of Nursing, Midwifery and Public Health, Igando, Lagos, Nigeria
[5]IITA Headquarters, PMB 5320, Oyo Road, Ibadan, Oyo State, Nigeria

*Corresponding Author - emmanuel.adetiba@covenantuniversity.edu.ng

ABSTRACT

The rising world population has made it imperative to get rid of time-consuming and non-economical agricultural practices. Thus, advances in Internet of Things (IoT) technology have recently propelled significant advancements in agriculture, similar to several other sectors. Through the combination of IoT and Farm Management Information Systems (FMIS), field data can be automatically collected, stored, analysed and accessed by farmers in real-time. In this paper, we present Proof-of-Concept (PoC) IoT circuit and FMIS web app for precision agriculture. The IoT circuit incorporates sensors, microcontroller and Wi-Fi module, which acquire and transfer field data to the FMIS web app. We designed and simulated the IoT circuit using Proteus, Arduino Uno, and Arduino Integrated Development Environment (IDE). The FMIS PoC web app was modelled with relevant Unified Modeling Language (UML) diagrams and Django (a Python framework) was employed to implement the app. Our simulation result illustrates how essentila field parameters can be monitored remotely and seamlessly by agriculture practitioners. It further provides a blueprint for real-time precision agriculture at scale and could serve as a learning aid for students in engineering and agricultural science.

Keywords: FMIS, Internet of Things, Precision Agriculture, PoC, UML

1. INTRODUCTION

Recent technological advances have resulted in 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). 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. The IoT 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 [1-2].

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, factory farming, animal feeding, agricultural monitoring and many others. Agriculture is an important sector for the application of IoT, especially because of the large areas of land that usually require continuous monitoring, which could prove to be a daunting task for humans due to factors such as the required efforts, harsh weather and environmental conditions [3].

IoT in agriculture has rightfully propelled what is now known as smart farming or precision agriculture. IoT in the agricultural sector involves the use of 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. The data collected by these systems can be used to generate algorithms for machine learning systems and to make predictions. In order to feed the world's growing population, combat adverse weather conditions and climate change, the use of IoT in agriculture is the way forward [4-6].

Some of the problems encountered by commercial agricultural industries include data collection, data accessibility when needed and automatic irrigation. Much time is wasted following traditional agricultural methods, which could include manual record-keeping, physical measurement of agricultural parameters, harvesting and irrigation. A number of research works have shown the necessity of encouraging the improvement and the adoption of smart farming practices across different agricultural domains [7]. In the

information science knowledge domain, Management Information Systems (MIS) are developed to compile past, collect present and predict future information related to certain operations for enhanced decision making. Thus, MIS that are specifically created for farming applications are known as Farm Management Information Systems (FMIS). Variants of FMIS reported in the literature have been explored and leveraged by farmers with reported optimized outputs and profit [8]. For instance, Kaloxylos et al. [9], 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.

Daum et al. [10], investigated the use of smartphones for data collection of agricultural and socio-economic data 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.

Assirelli et al. [11], 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. Zhang [20], also considered the algorithms used to map remote sensing features to particular plant diseases. Colezea et al. [12], 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.

Fogarty et al. [13], 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 [13]. Pathak et al. [14], 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.

In line with these foregoing existing studies, our work provides potential solutions to the disadvantages of traditional agricultural methods by developing a proof of concept platform for automatic and seamless farm data collection, storage, transmission and retrieval. However, one of our contributions in this work is the design, modeling and prototyping of a FMIS rather than connecting our IoT circuit to ThingSpeak, which is the normal trend in the literature. This paper is organized as follows: materials and methods are presented in section two, results and discussion are presented in section three while section four contains the conclusion.

2. MATERIALS AND METHODS

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. Table 1 shows the different components that make up the hardware system and their voltage ratings.

Table 1: Different hardware components and their voltage ratings [15-18]

Electronic components	**Voltage specification (V_{DC})**
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 sensors	3.3V – 5.2V
Water pump	5V

2.1 Hardware Block Diagram and Circuit

The hardware connection diagram is shown in Figure 2. The light and the soil moisture sensors are connected to different analogue pins on the Arduino Uno board. Arduino is an open-source platform that gives anyone a chance to create electronic projects on their own. The Arduino Uno was chosen for its simplicity and wide availability of resources. The hardware comes in different sizes and layouts known as boards. For instance, 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. It has a USB port for powering up the board via a computer or laptop and an ICSP header. The heart of the board, however, is the ATmega328 microcontroller, which is 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 Arduino to interpret analogue signals because the ATmega328 has an inbuilt analogue to digital converter.

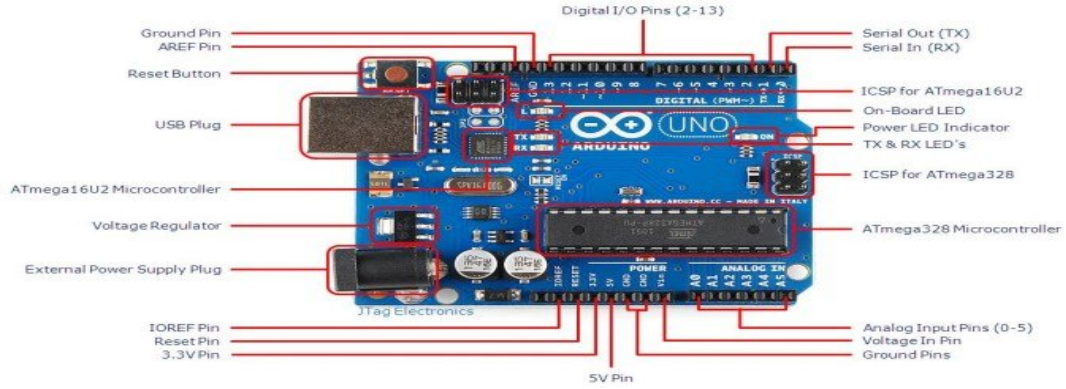

Figure 1: Arduino Uno [17]

The NodeMCU can be connected to digital pins 11, 12, and 13. These are Wi-Fi supported pins and 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 sensors are connected to any of the other remaining digital pins; this is because the buzzer and the DHT sensors 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 a pre-set value, the pump comes on and waters the plant in the plant pots. The plant pots represent a prototype farm to illustrate the concept of IoT for precision agriculture.

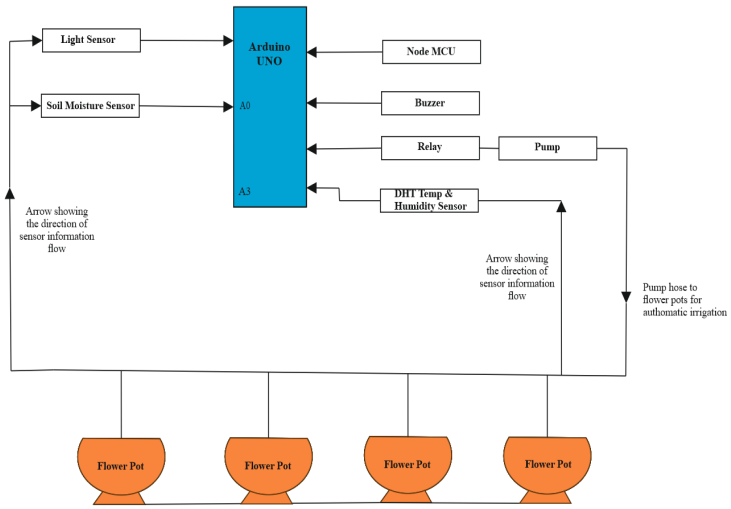

Figure 2: Hardware Connection Block Diagram

The Proteus software is used for electrical/electronic design and to generate schematics for Printed Circuit Boards (PCB) as well as for electrical circuit testing. Proteus design suite version 8.8 was used to develop the circuit diagram and perform circuit simulation. Virtual circuit components that correspond with the real components were selected and configured in Proteus (as shown in Fig. 3) in order to realize the block diagram in Fig. 2.

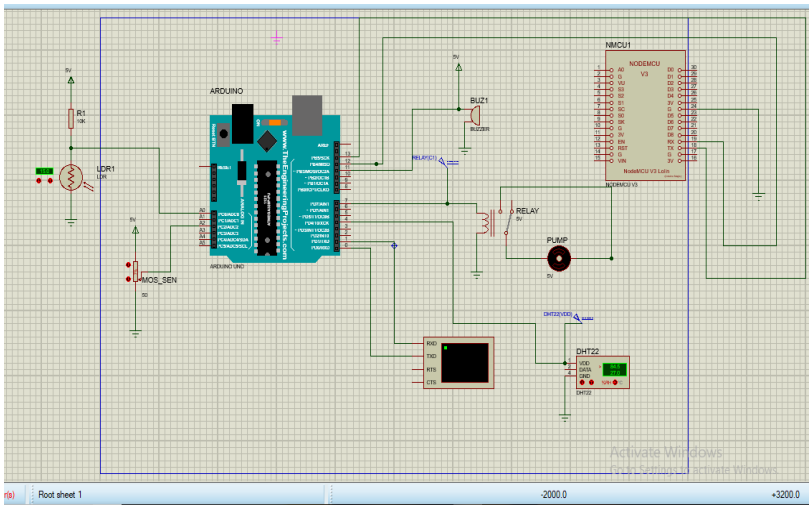

Figure 3: Circuit Diagram

2.2 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 [19]. The objective of this aspect of the project is to develop a web portal. 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 Hypertext Markup Language (HTML) and Cascaded Style Sheet (CSS) were included as the front-ends of websites are usually scripted as 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 [20].

Python is a multipurpose high-level programming language. It was developed in the late 1980s by Guido Van Rossum, a Dutch computer scientist [21]. Django is a Python framework and it saves developers from developing software projects from the scratch. For a web application, instead of the developer creating all the Application Programming Interfaces (APIs) and RSS feeds from the scratch, Django contains libraries of existing codes, which helps in straightforward and faster solution development.

Web applications can be modelled with various viewpoints using various modelling languages, which include Network Description Language (NDL), and Unified Modelling Language (UML). However, the most common object oriented modelling language is the UML. Class diagrams are the most common UML diagrams in use for software modeling available [22]. A class defines the properties and behaviours of objects that they describe. Each class has fields that hold values in the object; the fields are known as properties or attributes [22]. The class diagram for the SmartAgrio web portal in this work is shown in Fig. 4.

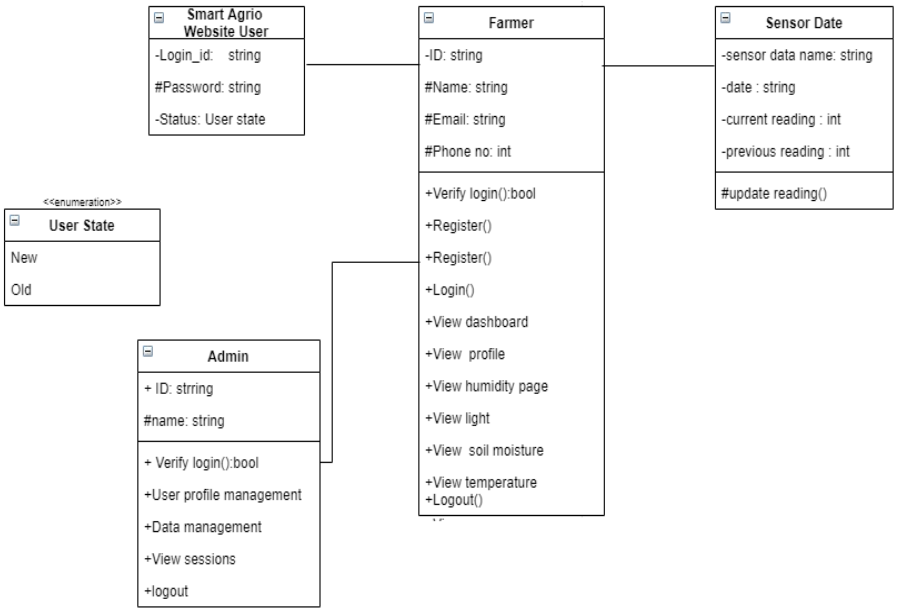

Figure 4: Class Diagram

The activity diagram shown in Fig. 5 represents a series of actions or flow of the control of the system. It is in the form of a flow chart diagram. As a UML behavioural diagram, it shows how the objects in the system can work together [23].

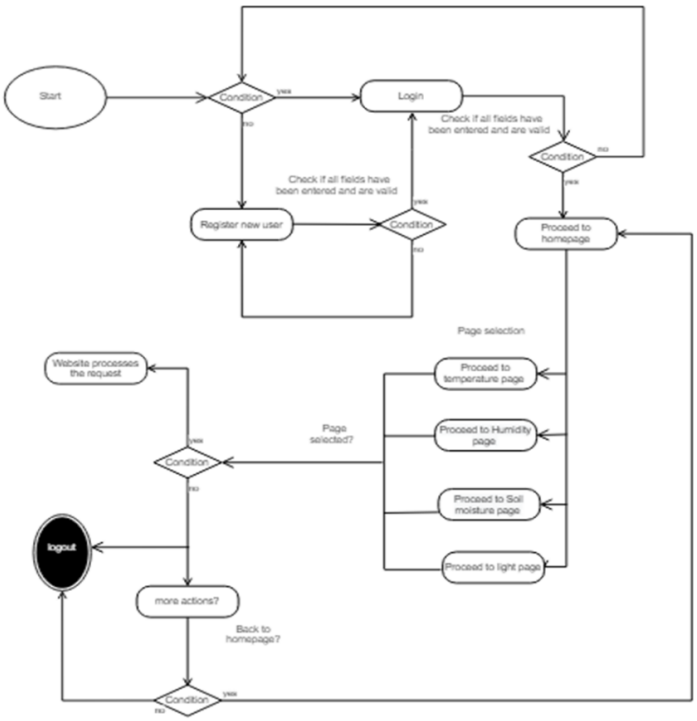

Figure 5: Activity Diagram

The system's components interaction is shown in Fig. 6. 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

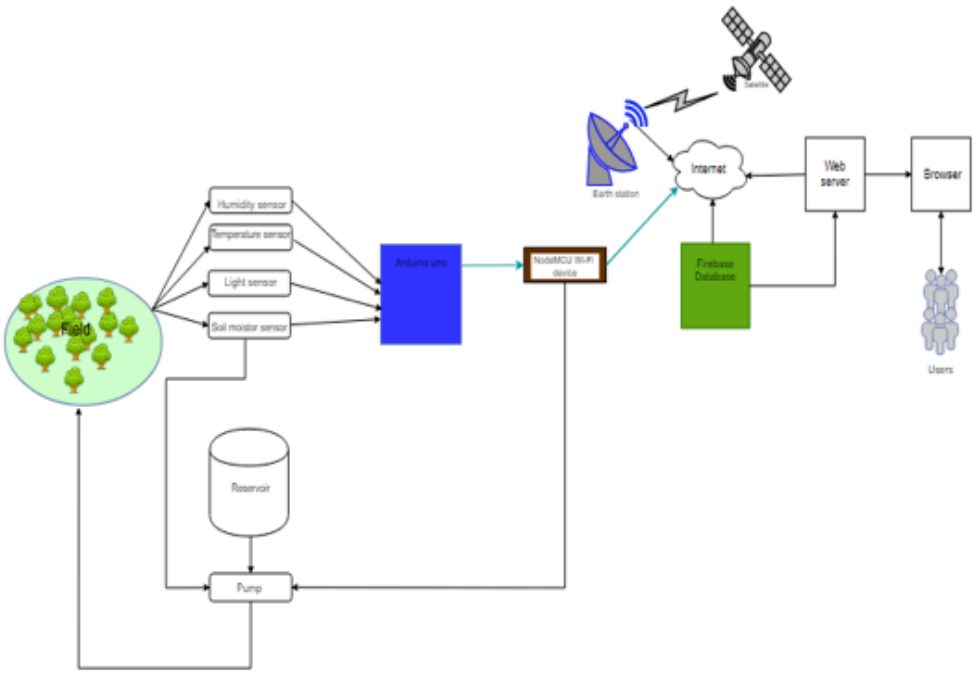

Figure 6: System Components Interaction Diagram

3. RESULTS AND DISCUSSIONS

The hardware circuit 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 emulator in Proteaus 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 output is shown in Fig. 7.

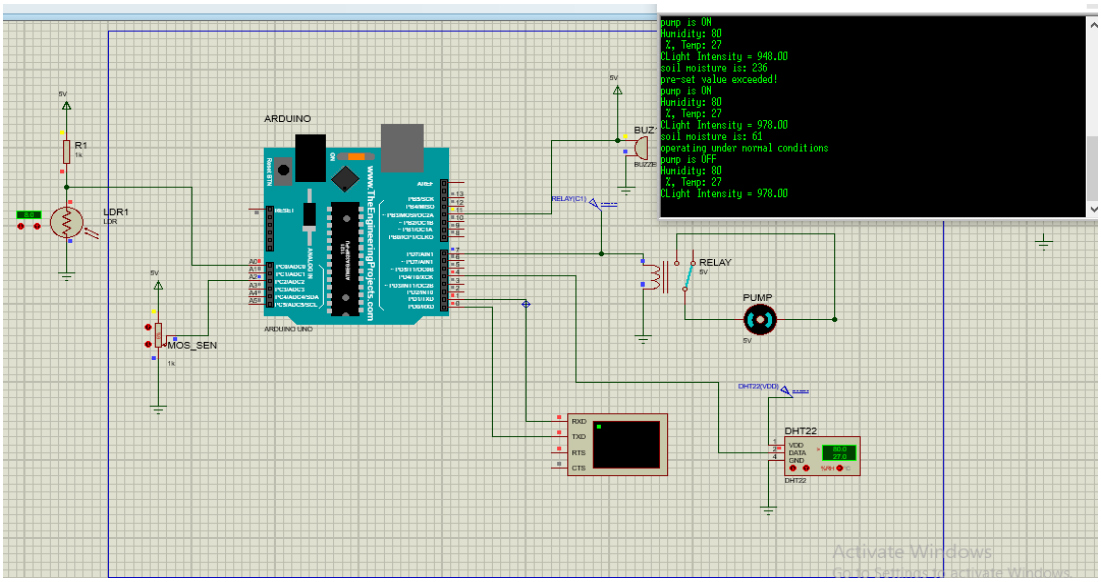

Figure 7: Proteus Simulation of Hardware Circuit

The login page is the first page of the SmartAgrio web portal is in Figure 8, followed by the sign-up page for a new user in Fig. 9. After the user logs in, the first view page is shown in Figure 10. The user can navigate and select the required action from the dashboard. Fig. 11 shows the profile page for users to adjust their personal profiles.

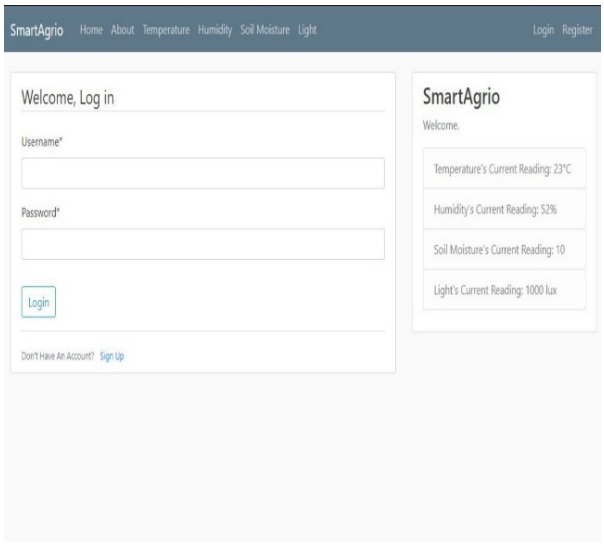

Figure 8: Login Page

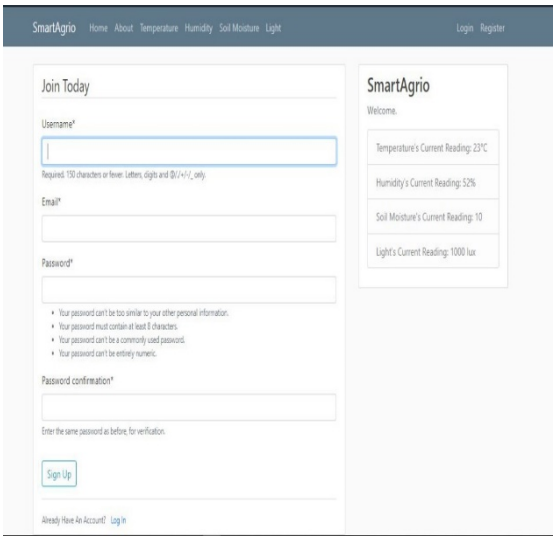

Figure 9: Registration Page for New Users

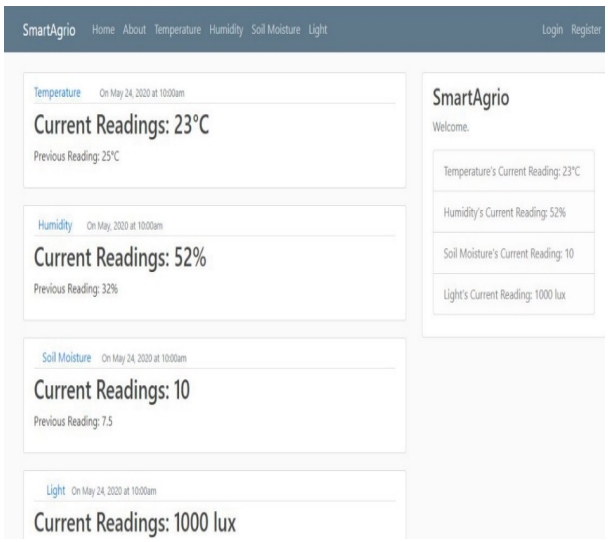

Figure 10: Current Readings Display Page

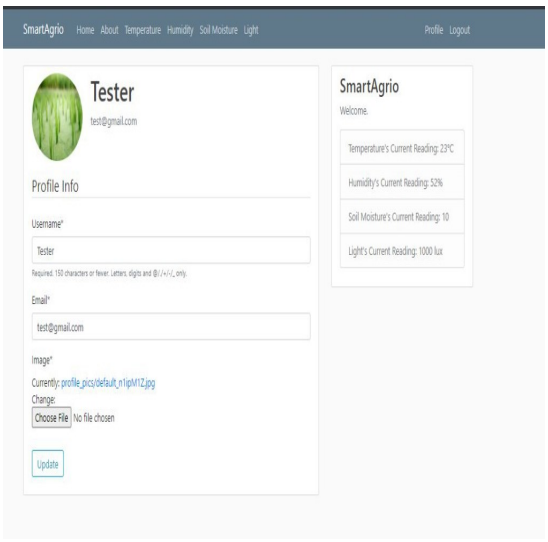

Figure 11: Test User Profile Page

Tests were carried out on the software to ensure the quality and performance of the application. Software testing deals with the verification of the quality of the system to ensure the elicited requirements were satisfied. Testing may or may not involve code execution. The unit testing involves testing each unit of the code and then each unit of the software system. The approach used for the unit testing of the prototype SmartAgrio web portal was a manual approach, which is convenient for the size of the system [24]-[27].

Each element of the front-end code was debugged and verified to ensure they worked correctly. Fig. 12 shows the output (for the temperature page) of the free online tool - the "Nu HTML checker" that we used to check the HTML codes for errors. 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; as shown in Fig. 13.

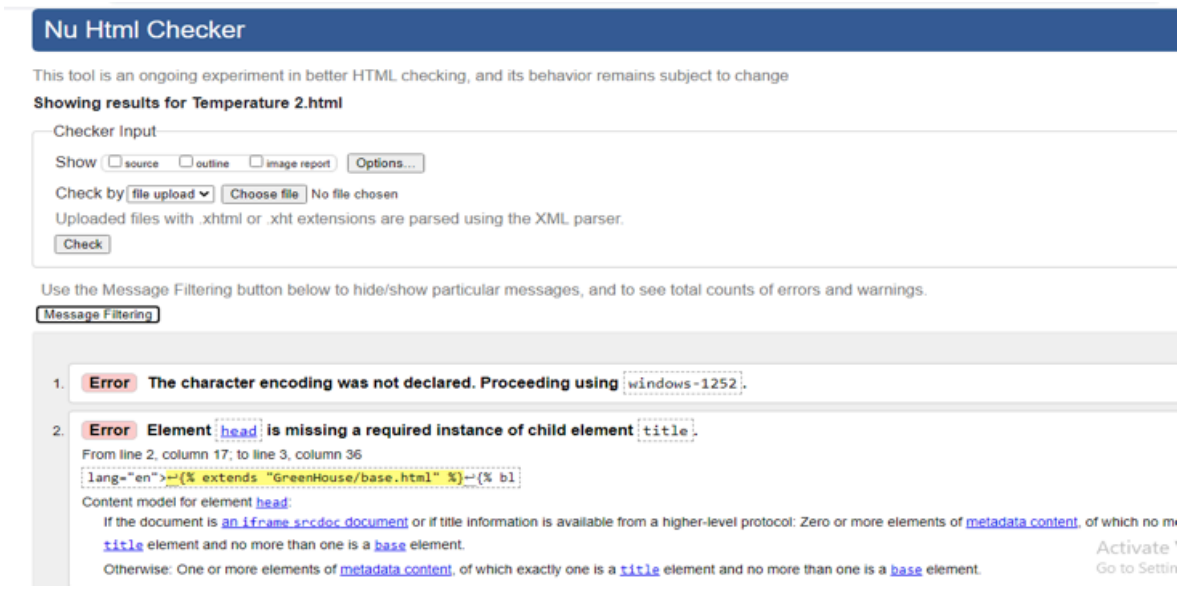

Figure 12: HTML Testing with the Nu HTML Checker before Bug Fixes.

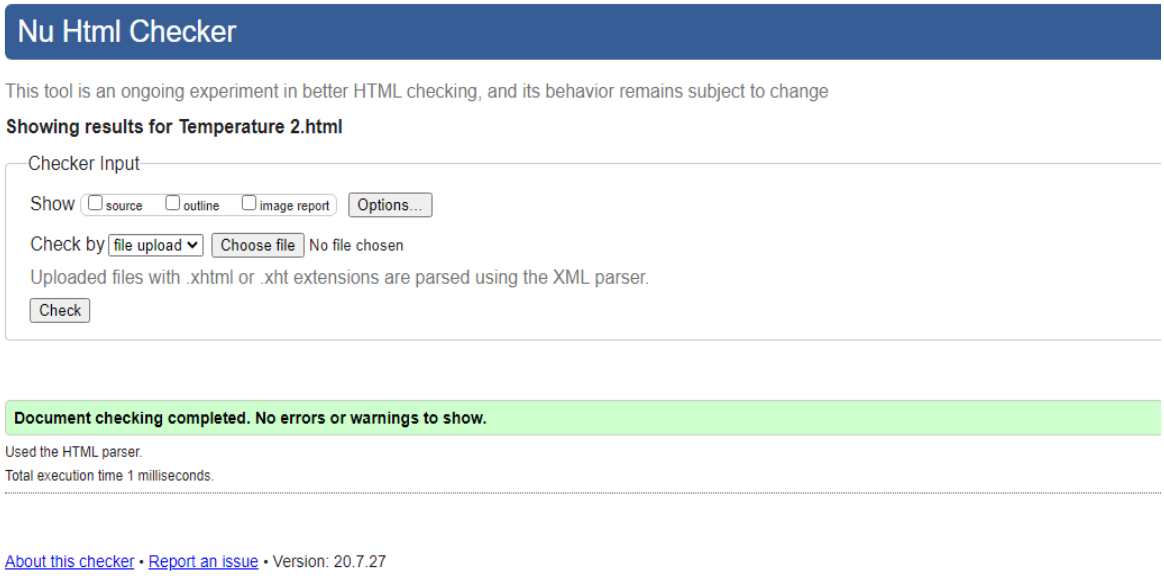

Figure 13: Nu HTML Checker after Bug Fixes

Furthermore, usability tests were carried out to ensure that the web portal is user friendly and easy to use. Responses from users were collected using Google form. The result of this testing showed the users view about their interaction with the web portal. For the user interface test, 15 respondents (9 students, 5 graduates and 1 non-classified respondent) were recruited for the survey to carry out usability testing of the SmartAgrio web portal. The Google form contained closed-ended questions (on a scale of 1 to 5), where 1 is very poor, 2 is poor, 3 is average, 4 is good and 5 is excellent. The categories in the questionaire include, *occupation, overall perception of the web portal user interface, overall user experience*. The responses to each category are graphically represented in Figs 14-16. Based on the statistics on the graphs(Figs 15 and 16), the *overall perception of the web portal user interface* (excellent + good = 86.7%) and *the overall user experience* (excellent + good = 80% *)* are high.

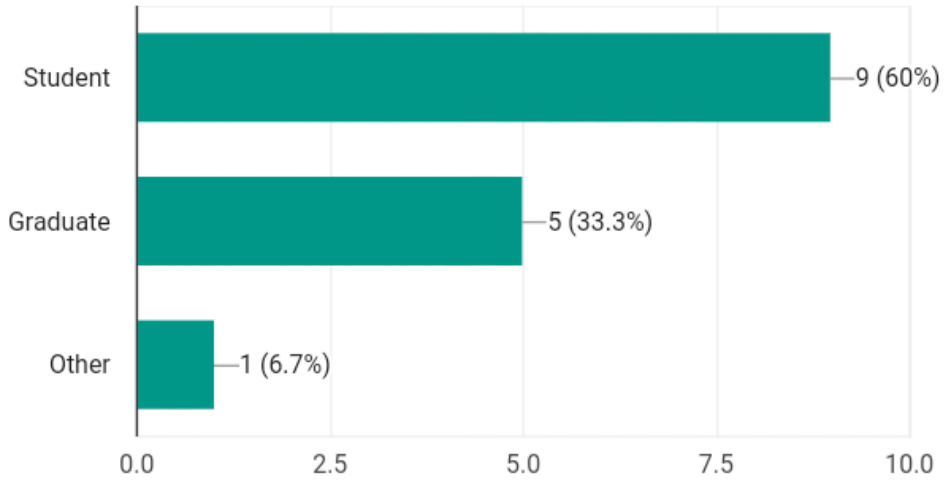

Figure 14: Participants Occupation

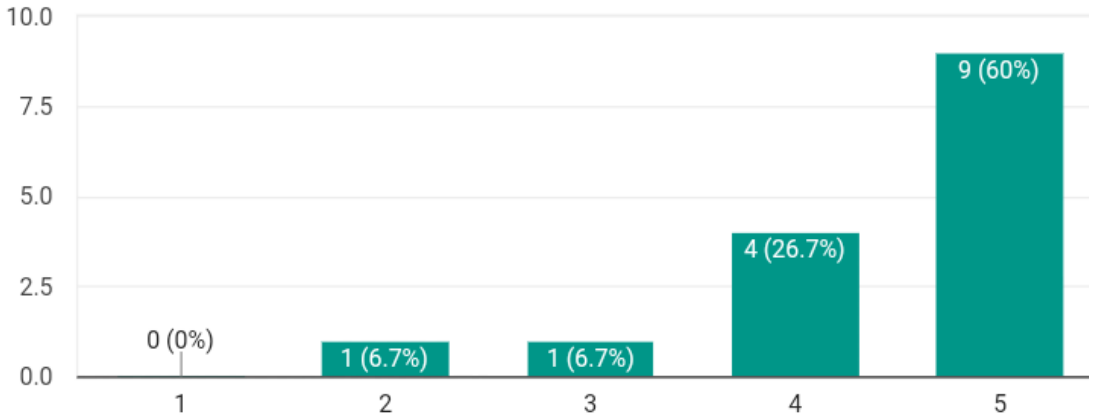

Figure 15: Overall Perception of the Site's User Interface

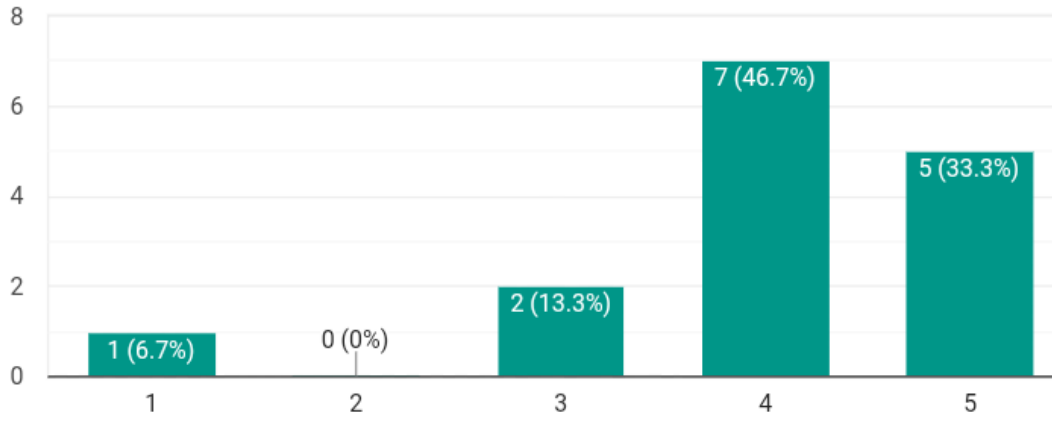

Figure 16: Overall User Experience

The foregoing results obtained through simulation of the designed electronic circuit and testing of the prototype SmartAgrio web portal illustrate the feasibility of precision agriculture. Our achievement in this work is similar to earlier effort by Pathak et al. [14], who automated the allocation of water to farms under different environmental conditions and streamed the sensor data to "Thingspeak". However, the prototype SmartAgrio web portal in our work provides a proof-of-concept alternative to "Thingspeak".

4. CONCLUSION

A proof-of-concept IoT and farm management information system have been presented in this paper. With the system, farms can be irrigated automatically and daily data logs of vital farm parameters (e.g. humidity, temperature, light, and soil moisture) are streamed to the SmarAgrio Web Portal, which is hosted on the Internet. The simulation outputs and prototype web portal testing results have established the feasibility of full scale implementation of precision agriculture. For future works, we hope to fully implement the system for deployment on real farms. We will also incorporate state-of-the-art techniques such as machine learning, stream and big data analytics to derive intelligence from the archived data. The outputs of this endeavor will ultimately help farmers and other agricultural stakeholders to overcome the challenges of traditional agricultural practices.

ACKNOWLEDGEMENTS

This work was carried out at the Advanced Signal Processing and Machine Intelligence Research (ASPMIR) Group laboratory, IoT Enabled Smart and Connected Community Research Cluster, Covenant University.

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] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.
[14] 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.
[15] 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.
[16] 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.
[17] H. Alyuruk, "Chapter 1 - Introduction to R and Python," in *R and Python for Oceanographers*, H. Alyuruk, Ed., ed: Elsevier, 2019, pp. 1-21.
[18] P. B. Kruchten, "The 4+1 View Model of architecture," *IEEE Software,* vol. 12, pp. 42-50, 1995.
[19] 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.
[20] M. J. Chonoles, "Chapter 14 - Behavior: Sequence Diagrams," in *OCUP Certification Guide*, M. J. Chonoles, Ed., ed Boston: Morgan Kaufmann, 2018, pp. 243-257.
[21] H. Alyuruk, "Chapter 1 - Introduction to R and Python," in *R and Python for Oceanographers*, H. Alyuruk, Ed., ed: Elsevier, 2019, pp. 1-21.
[22] M. J. Chonoles, "Chapter 6 - Objects and Classes," in *OCUP Certification Guide*, M. J. Chonoles, Ed., ed Boston: Morgan Kaufmann, 2018, pp. 89-113.
[23] M. J. Chonoles, "Chapter 16 - Behavior: Activity Diagrams," in *OCUP Certification Guide*, M. J. Chonoles, Ed., ed Boston: Morgan Kaufmann, 2018, pp. 271-295.
[24] M. J. Chonoles, "Chapter 14 - Behavior: Sequence Diagrams," in *OCUP Certification Guide*, M. J. Chonoles, Ed., ed Boston: Morgan Kaufmann, 2018, pp. 243-257.
[25] 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.
[26] 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.
[27] 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.
[28] P. B. Kruchten, "The 4+1 View Model of architecture," *IEEE Software,* vol. 12, pp. 42-50, 1995.