2021-07

Internet of things based system for environmental conditions monitoring in poultry house: A case of Tanzania

Lufyagila, Beston

NM-AIST

https://doi.org/10.58694/20.500.12479/1341

INTERNET OF THINGS BASED SYSTEM FOR ENVIRONMENTAL CONDITIONS MONITORING IN POULTRY HOUSE: A CASE OF TANZANIA

Beston Lufyagila

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of Master's In Information and Communication Science and Engineering of the Nelson Mandela African Institution of Science and Technology

Arusha, Tanzania

July, 2021

ABSTRACT

Poultry health is imperative for the continued growth of poultry and increased production. Environmental conditions such as temperature, humidity, and Ammonia gas impact the health of the poultry; whereby, they can affect the respiratory system and eventually cause death. In Tanzania, most farmers use charcoal and kerosene stoves to control these parameters as they have limited access to economical, secure, and user-friendly poultry house monitoring systems. However, such traditional methods are not environmentally friendly, unreliable, difficult to manage, and inaccurate. In addition to that, many smart poultry monitoring platforms proposed by previous studies are not centralized, thus making the system being not scalable and costly. For example a farmer with three distributed coops needs three monitoring platforms to monitor the coops while only single platform could suffice. Therefore, this study proposes an Internet of Things (IoT) based system for environmental conditions monitoring in the poultry house to address the aforementioned challenges. The survey was conducted in Arusha and Kilimanjaro regions respectively to determine the strength and weakness of the traditional methods used by poultry farmers, perception towards the proposed system and identify the requirements needed to develop the proposed an IoT based system. The study used Jupyter Notebook powered by Anaconda package manager and the Python language for data analysis. Based on the requirements gathered from the survey, the prototype that is divided into two four parts; sensing, aggregation, transmission and monitoring part was developed, where the hardware components were selected based on the cost, open source, availability in local market, Quality of Service (QoS), throughput and latency criteria. The developed prototype was deployed in the field to verify and validate its performance. This activity was done in seven days consecutively and the results indicated that the system would save both small scale and large scale farmers in terms of time and labor costs as Farmer can monitor and control the poultry house conditions securely, reliably, and remotely. The study also proposes an algorithm to allow the system to work online, and offline (i.e., synchronizing with the cloud server when the Internet access is available).

DECLARATION

I, Beston Lufyagila, do hereby declare to the Senate of Nelson Mandela African Institution of Science and Technology that this dissertation is my own original work and that it has neither been submitted nor being concurrently submitted for degree award in any other institution.

Beston Lufyagila

Name and Signature of Candidate Date

The above declaration is confirmed by:

Dr. Dina Machuve

Name and Signature of Main Supervisor Date

Prof. Thomas Clemen

Thomas Clemen

Name and Signature of Co Supervisor Date

COPYRIGHT

CERTIFICATION

The undersigned certify that, have read and hereby recommends for acceptance by the Nelson Mandela African Institution of Science and Technology, a dissertation entitled, "*Internet of Things Based System for Environmental conditions Monitoring in Poultry House*" in partial fulfilment of the requirements for award of the degree of masters in Information Communication Science and Engineering.

Dr. Dina Machuve

Name and Signature of Main Supervisor   Date

Prof. Thomas Clemen

Name and Signature of Co Supervisor   Date

ACKNOWLEDGEMENT

I thank the almighty God for this life and keeping me away from problems that could hinder my studies.

I am very grateful to my supervisors Dr. Dina Machuve and Prof. Thomas Clemen for the endless close supervision and advice throughout the period of my studies. Whenever I faced any challenge, they were there to support me morally and materially. Their efforts in reviewing and shaping my research towards the right direction were significant. I will always remain thankful for their efforts they have put to shape me in career and personal wise.

My sincere appreciation to my Employer Mbeya University of Science and Technology (MUST), thank you for granting me study leave that enabled me to focus on my studies. I would also extend my gratitude to my sponsor DAAD for supporting my studies.

I am too thankful to the community of the NM-AIST for the generosity support and encouragement. The NM-AIST community is of its kind that normally seeks cooperation than competition nor conflict. I will always remember this community.

Special thanks to my colleagues Francis Ruambo, Evelyn Mshana, and Tunu Ngajilo for devoting their time despite their tight schedule in my research whenever I asked them for any technical assistance. In fact, their contribution is forever reserved in my memory.

Lastly, I would be greedy for kindness if I did not thank the following; my mother Anjelica Mnyagala, uncle Japhet Mnyagala, and his family, my three sisters (Ajentina, Jerumana, and Agness), my lovely wife Siaeli Moshi and my children Ivan and Evans with their Sister Evelyn for their prayers, and social and economic activities that they did in my absence but for my benefit.

Thank you all.

DEDICATION

To my lovely mother Angelica Mnyagala, my lovely wife Siaeli Moshi, and my children Ivan and Evans.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS AND SYMBOLS

ARM	Advanced RISC Machine
CPU	Central Processing Unit
CSS	Cascading Style Sheet
DBMS	Data Base Management System
DED	District Executive Officer
DFD	Data Flow Diagram
DHT 11	Humidity and Temperature Sensor
FAO	Food and Agriculture Organization
FARMESA	Farm Level Applied Research for Eastern and Southern Africa
FR	Functional Requirement
GDP	Gross Domestic Product
GND	Ground
GPIO	General Purpose Input Output
GPRS	General Packet Radio Services
GSM	The Global System for Mobile Communications
GUI	Graphical User Interface
HDMI	High-Definition Multimedia Interface
HTML	Hyper Text Mark-up Language
HTTP	Hyper Text Transfer Protocol
ICT	Information Communication Technologies
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
JQ	J- Query library
JS	Java Scripts
KENCHRIC	The Northern Tanzania Health Research Ethics Committee
LAN	Local Area Network
LDR	Light Dependent Resister
LPWAN	Low power Wide Area Network
MCU	Microcontroller unit
MFA	Multi-Factor Authentication
MQ 135	Ammonia Sensor

NBS	National Bureau of Statistics
NFR	Non Functional Requirements
ODK	Open Data Kit
OS	Operating System
PC	Personal Computer
PHP	Hypertext Pre-Processor
PK	Primary Key
PPM	Parts Per Million
RAM	Random Access Memory
RP	Raspberry Pi
SDLC	System Development Life Cycle
SOC	System On Chip
SQL	Structured Query Language
SSL	Secure Socket Layer
TCRA	Tanzania Communication Regulation Authority
TLMI	Tanzania Livestock Modernization Initiative
TLMP	Tanzania Livestock Master Plan
UART	Universal Asynchronous Receiver/Transmitter
UML	Unified Modelling Language
USB	Universal Serial Bus
WAN	Wide Area Network
WAP	Wireless Access Point
WHO	World Health Organization

CHAPTER ONE

INTRODUCTION

1.1 Background of the Problem

In Tanzania, agriculture is the pillar of the economy, contributing about 30% to the Gross Domestic Product (GDP) and employing 67% of the labor force, with women accounting to over 70% of the workforce (National Bureau of Statistics [NBS], 2017). Livestock keeping, a sub-sector of agriculture, plays an important role in achieving objectives of the National Development and poverty alleviation (Tanzania Livestock Modernization Initiative [TLMI], 2015). The livestock sector contributes 4.7% of the National GDP and 18% of Agricultural GDP. Livestock keeping involves keeping animals like goats, cattle, pigs, swine, and poultry (TLMI, 2015).

Poultry farming is the sub-sector of livestock and deals with keeping of chickens, ducks, turkeys, gees, and pigeons; where by, chicken is prevalent in Tanzania (Goromela *et al.*, 2007). Chickens accounts for 86% of animals kept by families in Tanzania. Out of 4.6 million domesticated animals keeping families, 48% just farm chicken (NBS, 2017). In addition to that, chicken as source of food and earnings to subsistence farmers, the global demand for chicken meat and eggs will be more than 40% by year 2050 (Astill *et al.*, 2020). These measurements exhibit the focal significance of advancing poultry production in Tanzania to poverty alleviation and household health improvement in terms of nutrition from the meat and eggs (TLMP, 2018).

Poultry health is important for continued growth and productivity of the poultry sector (Edmond, 2018; TLMP, 2018). However, environmental conditions such as temperature, humidity, ammonia gas and luminosity seriously affect the health and welfare of the flocks (Islam *et al.*, 2019; Pereira *et al.*, 2020). When the temperature in the poultry house is above the threshold, the flock suffers from heat stress and takes less food, this in turn causes poor health of the bird or death (Mahale, 2016). On the other hand, when the temperature is below the threshold, the bird consumes more feed to compensate for the heat lost due to coldness, this increases costs as farmers have to buy more feeds to the chicken’s requirements (Hitimana *et al.*, 2018; Mahale, 2016). Humidity and ammonia gas, if not maintained at optimal level, damage the respiratory system of the bird that eventually leads to loss of weight or death (Najafi *et al.*, 2015).

In Tanzania, monitoring and controlling of environmental conditions in the poultry house is still a challenge among farmers. Farmers use traditional systems like charcoal stoves and kerosene lamps to control temperature and humidity inside the poultry house (Edmond, 2018; FAO SHFS, 2015). Furthermore, many farmers use common sense to monitor conditions inside the poultry house. These tradition systems still result into low poultry productivity, and they are environmentally unfriendly as compared to using Internet of Things (IoT) based systems with sensors, and heater and fan to monitor and control the parameters respectively (Edmond, 2018; FAO SHFS, 2015; Phiri *et al.*, 2018).

The Internet of Things (IoT) is a potential technological solution for smart remote monitoring of environmental conditions. The IoT is the interconnection of devices that communicate and exchange data over the internet (Abdul-qawy *et al.*, 2015; Phiri *et al.*, 2018). The IoT uses sensor networks, in the automation and monitoring fields such as home automation, health, industrial, and environmental monitoring (Miorandi *et al.*, 2012). Sensor networks have sensor nodes that are incorporated into the environment to gather data at regular intervals of time defined by the user. These nodes are designed to operate unattended for periods of weeks to years (Baronti *et al.*, 2007).

1.2 Statement of the Problem

Monitoring temperature, humidity and ammonia parameters that affect poultry health conditions inside the poultry house is still a challenge for poultry farmers in Tanzania. Currently, poultry farmers use charcoal stoves and kerosene lamps to control temperature and humidity in poultry houses. In addition, many farmers use common sense to monitor conditions inside the poultry house. These tradition methods still result into low poultry productivity, unreliable, and the use of charcoal is not environmentally friendly (Edmond, 2018; FAO SHFS, 2015; Phiri *et al.*, 2018). Moreover, the commercial off the shelf monitoring systems are expensive, and those (Thermometer and Hygrometer) that are affordable by the farmer operate manually, that is, the farmer has to visit the farm in order to read the temperature and humidity of the day from the Thermometer and Hygrometer (Pereira *et al.*, 2020).

Furthermore, many studies describe poultry monitoring systems, but few have reported security, reliability, and scalability issues as discussed by Farooq *et al.* (2020). Also, many automated poultry systems, their IoT Platforms are not centralized i.e. they lack a single IoT Platform (dashboard) to monitor and manage more than 100s distributed Poultry farms. Also, little has been reported on the algorithm to allow a farmer to monitor and manage the coop in both offline, and online modes (Manshor *et al.*, 2019; Pereira *et al.*, 2020; Phiri *et al.*, 2018; Islam *et al.*, 2019a; Mahale & Sonavane, 2016). Therefore, this study proposes an IoT system that fills the gaps in the previous studies while proposing an algorithm for a system to work in both online and offline mode to accommodate the local context.

1.3 Rationale of the Study

Internet of Things approach for environmental monitoring in various industries have increased in recent years (Abraham *et al.*, 2017). Many researchers have been using this approach in automation in health, smart cities, smart homes, and agriculture sectors (Abraham *et al.*, 2017; Pereira *et al.*, 2020). However, Poultry farmers in Tanzania, have been using traditional methods to monitor environmental conditions which seriously affect the health of poultry because the commercial off the shelf monitoring systems are expensive, and those (Thermometer and Hygrometer) that are affordable by the farmer operate manually (Edmond, 2018; FAO SHFS, 2015). Therefore, the developed IoT based system will address farmer's needs by incorporating the following:

(i) A tool to monitor and manage environmental conditions remotely and securely.

(ii) A feature that allows Farmers to monitor and control the farm in offline and online mode- when they are on premise, and off premise respectively. This feature supports both small scale and large scale farmers.

(iii) A scalable system- farmers can monitor and manage as many coops as they own in real-time from a single dashboard (IoT platform), switch a system to automatic or manual control, set threshold values based on the age of chicken, and set time intervals for the sensor to send data.

(iv) A secure system-that secures data during transmission from the sensor to the consumer (cloud server and mobile app) using Secure Socket Layer (SSL) algorithm, and integrates Multi-Factor Authentication security mechanism to secure access to the system.

1.4 Research Objectives

1.4.1 General Objective

The main objective of this research is to develop a secure, reliable and low-cost Internet of Things based system for monitoring environmental conditions (Temperature, Humidity and Ammonia gas) in poultry house.

1.4.2 Specific Objectives

The specific objectives of this research were:

(i) To identify and establish requirements for an IoT based system for environmental conditions monitoring in poultry house.

(ii) To design and implement an IoT based system for environmental conditions monitoring in poultry house.

(iii) To validate the IoT based system developed.

1.5 Research Questions

(i) What are the requirements for developing an IoT based system for environmental conditions monitoring in the poultry house?

(ii) How can an IoT based system for environmental conditions monitoring in the poultry house be designed and implemented?

(iii) Does the developed IoT based system satisfy the users' requirements?

1.6 Significance of the Study

The proposed low-cost solution for monitoring poultry farms in Tanzania is expected to improve the farmers' productivity. This is due to the fact that poultry health as well as good yield entirely are highly influenced by environmental conditions of the poultry house (Kocaman *et al.*, 2006; Pereira *et al.*, 2020).

Using charcoal to control temperature and humidity is not environmentally friendly as it is currently practiced by poultry farmers (Edmond, 2018). However, the proposed solution is expected to control temperature and humidity using Fan and Heater. Furthermore, the proposed solution will be powered using micro-solar panel, rechargeable batteries, and adopting algorithms that saves power. Therefore, the study is expected to promote environmental conservation that would be destroyed for harnessing charcoal.

According to the World Health Organization (WHO), it is estimated that the population is expected to grow to 8.6 billion by 2030. To feed this population, food security is the major area of concern. In this regard, the study will contribute to improved production of poultry that will serve the world demand of food.

Also, monitoring and controlling the poultry farm remotely and securely will motivate employed people from the public and private sector to practice poultry farming as a source of passive income.

1.7 Delineation of the Study

Environmental conditions monitoring in poultry house is a broad field. This study focuses on monitoring temperature, humidity, and ammonia parameters; control of temperature and humidity only using heater and fan. Other parameters such as Luminosity, oxygen, carbon dioxide, intruders and the like have not been addressed.

CHAPTER TWO

LITERATURE REVIEW

2.1 Environmental Conditions in Poultry House

Environmental conditions in poultry house play an important role to poultry health and productivity. Temperature, Humidity, Ammonia, Luminosity, Carbon dioxide, Hydrogen gases and others are examples of environmental conditions in poultry house. Temperature, humidity and ammonia are the major environmental conditions that seriously affect the poultry health and Pereira *et al.* (2020) suggest that they should be kept in optimal for continued growth and productivity of poultry.

2.1.1 Effects of Temperature to Poultry Health

Temperature is the degree of hotness or coldness of an object. The study conducted by Daghir (2008) found that the optimal temperature of depends on the age of the poultry as shown in Table 1. Generally the range of temperature should be around 20–26.5 ^{o}C. Above this, the flock suffers from heat stress; below this, the flock consumes more food in order to generate internal body heat, which is costly to farmer (Daghir, 2008; Holik, 2015). Heat stress due to high temperature results into under-nutrition, stunted growth, reduction in egg production and size, laying of premature eggs and even death of the chicken (Gebregeziabhear & Ameha, 2015).

Table 1: Recommended temperature for chicken based on age of the flock

Age (Weeks)	Temperature (^{o}C)
1	32 - 30
2	30 -26
3	26- 23
4	23 -20
5	20

2.1.2 Effects of Humidity to Poultry Health

Humidity is defined as the amount of water vapor present in air. In general relative humidity should be around 50–70%. Low relative humidity causes airways and mucosal dryness due to increased rate of heat dissipation by evaporation. On the other hand, high relative humidity associated with high temperature may worsen chicken performance. Relative humidity when combined with temperature also may influence the growth of bacteria causing diseases such Avian Influencer, and increase of ammonia gas in the house (Oloyo, 2018; Zhao *et al.*, 2012).

2.1.3 Effects of Ammonia to Poultry Health

Ammonia is the toxic gas which is present in the air. In poultry house, ammonia gas is produced when micro-bacteria in the litter under favorable temperature and humidity decompose the uric acid produced in the droppings of the chicken. Studies show that the recommended ammonia level in the poultry house should be kept less 10 parts per million (ppm) (Oloyo, 2018; Pereira *et al.*, 2020).

2.2 Poultry Monitoring Systems in Tanzania

In Tanzania, monitoring and control of temperature, humidity and ammonia gas in the poultry house is still a challenge among farmers (Edmond, 2018; TLMP, 2018). Farmers use charcoal stoves and kerosene lamps to control temperature and humidity inside the poultry house (FAO SHFS, 2015; TLMP, 2018). Moreover, Farmers use common sense or devices such as thermometer and hygrometer in attempt to monitor conditions inside the poultry house (TLMP, 2018).

2.3 Information Communication Technologies Usage in Tanzania

The usage of ICT in Tanzania is evidenced by the quarterly reports published by Tanzania Communication Regulation Authority (TCRA). The statistical report of September 2020 (Fig. 2) indicates that Tanzania has 47.7 million mobile subscribers (corresponding to 88%) of which 25.8 million (corresponding to 46%) have access to internet (Tanzania Communication Regulation Authority [TCRA], 2019). From Fig. 1, Internet usage has been increasing with an average of 2.8% annually from 2014 to 2019. With this significant technological advancement, low-cost monitoring systems are significantly becoming an attractive option for monitoring environmental conditions remotely in Tanzania.

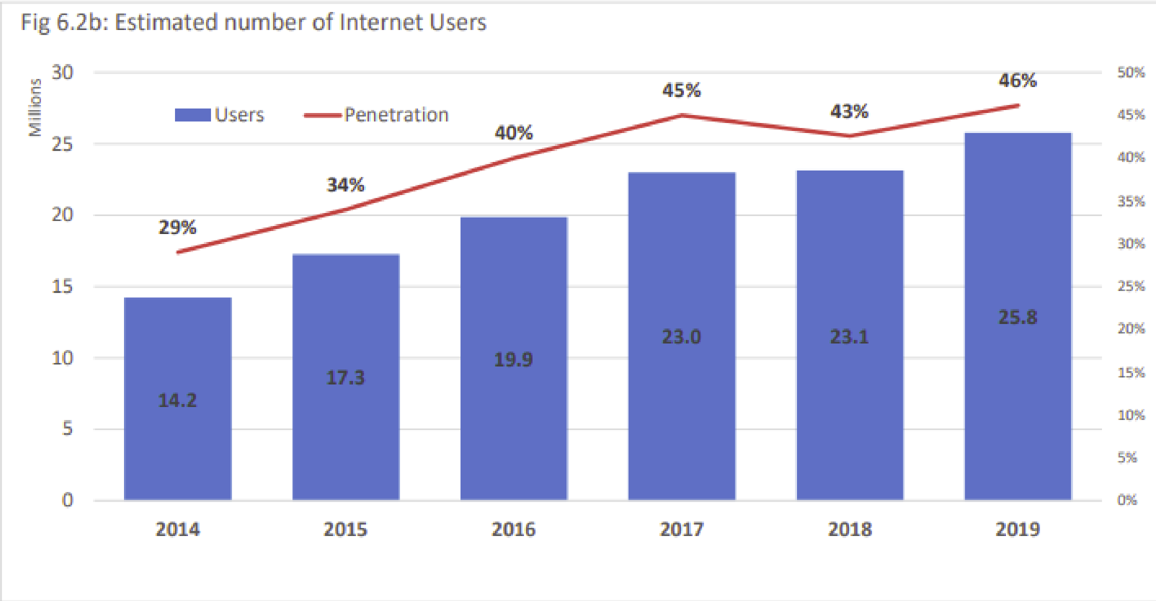

Figure 1: Estimated number of Internet Users in Tanzania (TCRA, n.d.)

2.4 Internet of Things

The term Internet of Things (IoT) was devised by Kevin Ashton in 1999 when he wanted to attract the top management in a company he was working on the Radio-Frequency Identification (RFID) technology for supply chain management (Chin *et al.*, 2019). The RFID is one approach of realizing the concept of IoT where by, it works by putting tags to an objects that will then be tracked (Landaluce *et al.*, 2020). Another IoT enabling technology is a Wireless Sensor Network (WSN)– network of sensor nodes incorporated into the field to gather data at regular intervals of time defined by the user (Baronti *et al.*, 2007). For any device to qualify as an IoT device, it should have three fundamental characteristics; first, it should be able to identify itself, communicate and exchanging information with other IoT devices in the world through internet (Abdul-qawy *et al.*, 2015).

2.5 Internet of Things Architecture

The fundamental IoT architecture consist of three layers as shown in Fig. 2. These layers are perception layer, network layer, and Application layer (Phiri *et al.*, 2018;Talavera *et al.*, 2017).

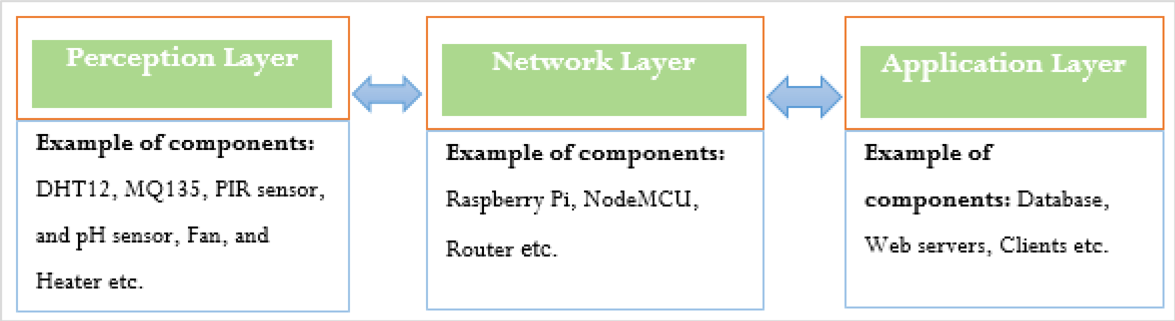

Figure 2: Internet of Things Architecture

2.5.1 Perception Layer

This layer consists of sensor nodes for gathering information about the environmental conditions. Examples of sensors are; DHT sensors – for sensing humidity and temperature; PH sensors – for sensing the pH of the soil; PIR sensors- for sensing movement/motion of an object and the like. Normally this layer senses environmental conditions or detects other smart objects in the environment (Phiri *et al.*, 2018; Talavera *et al.*, 2017).

2.5.2 Network Layer

This layer is used for receiving, processing, and transmitting the sensed data to different parts of the network. This layer consist of gateways, routers and Micro-controller devices like Arduino, Esp 8266 NodeMCU, and Raspberry pi, and communication protocols. Communication protocols like Hyper Text Transfer Protocol (HTTP), MQ Telemetry Transport (MQTT), Zig bee and Wi-Fi provide the capability of the Network layer devices to communicate with Perception Layer as well as Application Layer (Phiri *et al.*, 2018).

2.5.3 Application Layer

In this layer, all data collected are managed, processed, analyzed and presented in a user-friendly manner. Furthermore, Application layer provides an interface to communicate with sensor layer for changing environmental conditions using actuators like fan and heater. Generally, this layer consists of Webservers, Databases, and Clients (Phiri *et al.*, 2018; Talavera *et al.*, 2017).

2.6 Multi-Factor Authentication

Multi-Factor Authentication (MFA) is a zero trust security mechanism used to identify a user by requiring additional credentials (Mathew & Thomas, 2013). For instance, instead of asking a user name and password, the system may go further to require a code generated from the user's phone, facial recognition, voice recognition, retina or iris scanning, behavioral analysis, or asking the question. The appropriate use of the MFA features enhances protection from different attacks, for example, Phishing, Denial of Service (DoS), eavesdropping, and many more (Griffin, 2017; Mathew & Thomas, 2013). Along this line, MFA has been integrated into our work to enhance security.

2.7 Communication Technologies in Internet of Things

Communication technologies plays an important role in IoT arena. In IoT, sensors exchange information with other devices using wired or wireless protocols (Al-Fuqaha *et al.*, 2015). However, wireless technology is becoming popular than wired technology due to the fact that wireless sensors are cost effective, and are possible for installation even in areas where cabling is not easy. For devices to communicate wirelessly requires wireless protocols. Examples of these wireless protocols are Wi-Fi, Zig bee, Bluetooth, and Low Power Wide Area Network (Sohraby *et al.*, 2006).

2.7.1 Low Power Wide Area Network

Low Power Wide Area Network (LPWAN) is the family of technologies that is built to support large-scale IoT networks sprawling over vast industrial and commercial campuses. LPWAN comprises of the technologies such as LoRA, Sigfox and Narrow Band IoT (NB-IoT) (Ikpehai *et al.*, 2019). However, LoRa is significantly used in embedded systems and IoT than other technologies of LPWAN family. The LoRa has high noise immunity and ability to transmit data over long distance while using licence –free spectrum, this makes LoRa cost – effective solution for building IoT systems (Georgiou & Raza, 2017; Khan & Chen, 2019).

2.7.2 Wi-Fi

The term Wi-Fi was created in 2000 by Wireless Ethernet Compatibility Alliance which later become Wireless Alliance. The Wi-Fi is the wireless Local Area Network (LAN) technologies that utilize the IEEE 802.11 standards for communications (Li *et al.*, 2011). The Wi-Fi products use radio waves to transmit data from a client device to either an access point, which includes a router, and the router completes a connection to other devices on the LAN, Wide Area Netwrok (WAN) or the internet. Initially the technology used the 2.4 GHz frequency, but has since expanded to 5 GHz frequency bands. The Wi-Fi technology is widely used in various IoT products because it is ubiquitous -most of the devices come with Wi-Fi embedded in the devices (Ahmed *et al.*, 2018).

2.7.3 Zig Bee Protocol

Zig Bee is a wireless communication protocol which is used for data communication between devices requiring low power and low data transfer rate (Yi *et al.*, 2011). It is defined by an Institute of Electrical and Electronics Engineers (IEEE) 811.15.4 as a standard of wireless communication protocol that is capable to transmit data to maximum rate of 250 Kbps at operating frequency of 2.4 GHz ISM-Band and a bandwidth of 1.3 Mbps (Li *et al.*, 2017). The protocol is recommended in the areas with unreliable power source, and in applications that require low bandwidth like in home automation and medical equipment (Han & Lim, 2010).

2.7.4 Bluetooth

Bluetooth (IEEE 802.15.1) is wireless protocol that is used for short-range communication and considered a cable replacement for mobile devices (Wang *et al.,* 2006). The maximum coverage range of Bluetooth signals between the devices communicating is 10 meters. Above this range, the protocol suffers from poor performance. Hence, Bluetooth is suitable for applications requiring data at high speed but within a given short distance of travel. Bluetooth is highly used in mobile phones, radio and Televisions for data transfer such as files, music's, images and videos between one device and the other (Ruiz-Garcia *et al.*, 2009).

2.8 Related Works

Researchers have investigated recent developments in environmental conditions monitoring in poultry house using Internet of Things technology. Recent topics in poultry sector include on improving the health of poultry to increase productivity and growth.

Pereira *et al.* (2020) described a low cost, IoT system for monitoring temperature, humidity, and ammonia and luminosity environmental parameters in poultry farms. Sensors DHT 22, MQ 137 and LDR for measuring these parameters were connected to Wemos Mini D1. The system was able to send data to cloud server where a client is able to access them through android mobile application. The model showed good outcomes, demonstrating that its execution is feasible. However, the proposed solution does neither take into account multi-factor authentication, lacks a feature to enable a farmer to set reminders such as vaccination, cleaning house. Also the system does not work offline and the author suggests improvement of software engineering of the system.

Manshor *et al.* (2019) proposed an IoT solution to monitor temperature, humidity and electricity presence in the cage with poultry under research. He used Raspberry Pi to connect to DHT 11 sensor to read temperature and humidity values. The system uses google firebase database for storing real-time data and status from the sensor devices. Moreover, a mobile application was developed to access the data and alert notification to users. However, the system is not scalable as google fire-base database provides small storage of data. In addition, the system does not provide mechanism to control them.

Phiri *et al.* (2018) proposed a low-cost model that can be used to monitor conditions in the environment of a broiler house and send the values to the farmer in real-time. In the proposed model, Arduino Uno boards are used for sensor nodes and gateway. The Zig bee, GSM and GPRS are used to provide connectivity. The proposed system monitors only temperature, humidity and intruders using DHT11 and PIR and Ultrasonic sensors. However, one of the limitation is that, the system neither monitor Ammonia gas nor works in offline mode when there is no internet connectivity.

Paputungan *et al.* (2020) propose a prototype to monitor temperature, and humidity within the Poultry house/Cage. The prototype use DHT11 and Arduino DUE microcontroller. The system collects data and forwards to Blynk server, which shows the monitoring results. However, the prototype does not provide a mechanism to control those parameters, does not monitor Ammonia toxic gas, and the Blynk server architecture does not provide single dashboard to monitor more than 100s poultry farms (coops).

Islam *et al.* (2019) proposed design and implementation of a prototype of an automated poultry farm system which takes responses from the microcontroller and informs the user about any necessity of the maintenance through the GSM and IoT system. The system monitors humidity, temperature, light intensity and ammonia gas. These data are send to the farmer owner mobile phone through Short Message Services (SMS). The limitation to this is, the proposed prototype has no single dashboard (platform) to monitor more than 100s chicken coops.

Mahale and Sonavane (2016) proposed system of monitoring poultry farm by using Wireless Sensor and GPRS based Network. The system monitors the water, food level as well as poultry house conditions which are temperature and humidity. The data collected by the wireless sensor are then shared to the person in charge of poultry farm with the assistance of GPRS network. The system does not monitor the presence of toxic gas such as Ammonia which highly affect the production of chicken.

Thus, this study aim to develop a reliable, manageable and secure IoT system to remotely monitor and control temperature, humidity and ammonia gas in poultry house by in-cooperating Multi-factor authentication and Secure Socket Layer protocols for protecting data during access and transmission respectively. The proposed system incorporates common features found from the previous studies while introducing new features to accommodate the Tanzania environment. The new features include the ability of the system to be accessible in online and offline modes and a centralized monitoring platform where a farmer can monitor many distributed coops using single platform.

CHAPTER THREE

MATERIALS AND METHODS

3.1 Study Area

This study was carried out in Arusha and Kilimanjaro Regions located in the north eastern part of Tanzania. Arusha region lies south of the equator between latitude 2°S and 6°S While Kilimanjaro lies south of the equator between latitude 20° 25' and 40°15' S, and 36° 25' 30" and 38° 10' 45" E east of Greenwich. In the Eastern part, Arusha is boarded with Kilimanjaro Region, to the south is boarded with Manyara and it is boarded with Simiyu and Mara in the west and North-West respectively. In the north part, Kilimanjaro is boarded with Kenya, to the west is boarded with Arusha region, to the East is boarded with Kenya, to the South is boarded with Tanga while in the south west is boarded with Manyara as shown in Fig. 3.

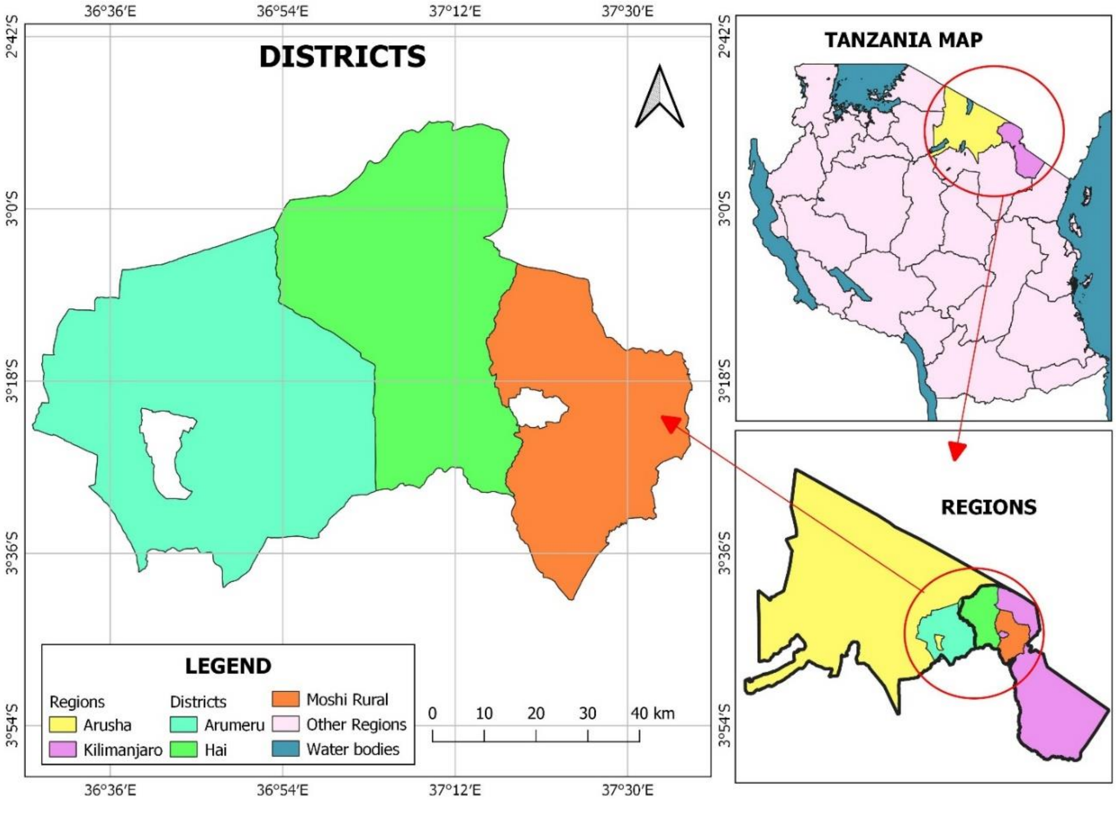

Figure 3: The Map indicating the study area

Arusha and Kilimanjaro were selected because there are several poultry farmers practicing in house farming. Furthermore, the two regions have good tourists attractive natural resources that makes them being among the leading regions consuming poultry products (NBS, 2017).

In addition to that, the two regions have a good number of people who possess smart phones capable of accessing internet and other ICT services (TCRA, 2019).

3.2 Sampling Techniques

Convenience and Snowball non-probability sampling methods were used to reach 120 participants based on the availability and willingness to participate in the study. Furthermore, the selected participant was used to refer the researcher to another individual within the population whom can participate in the study. A sample size of 120 participants was selected because according to Farm Level Applied Research for Eastern and Southern Africa (FARMESA) programme, the sample size of around 80 to 120 participants is satisfactory for most applied research studies in Sub-Saharan Africa (Matata, 2001; Mkunda *et al.*, 2020).

3.3 Data Collection

The research employed both qualitative and quantitative questionnaires to capture data related to research objectives in three districts, namely: Arumeru, and Hai and Moshi village districts of Arusha and Kilimanjaro Region respectively. The questionnaires were prepared in excel, converted to XML using online XlsForm Converter and then were uploaded on the Open Data Kit (ODK) collector. The ODK collector is an open source android platform, which is used by researchers to collect data using electronic forms (Abdulgadir *et al.*, 2014).

The questions prepared are grouped into five categories: The first category is the demographic survey – which consists of age, gender, and education level. The second category is general information about poultry farming. The third category is about strength and weakness of the methods used by farmers in maintaining conditions in poultry house. The fourth category is the perception of farmers towards IoT powered monitoring systems for poultry house and reasons of not using them. And the last category of the questions is about features/requirements farmers anticipate in the proposed system. To achieve the last category of questions, design thinking approach was used to gather requirements of the proposed system by creating mockups and visualizing the system to farmers (Kumar *et al.*, 2020). Almost, 80% of the questions were closed questions. Closed questions structure was adopted to collect only relevant information and save time.

3.4 Data Analysis

The study used Jupyter Notebook powered by Anaconda package manager and the Python 3.8.0 language for data analysis. Since during data collection the ODK collector submitted the survey to a spread sheet stored in Google Drive, the survey excel sheet was downloaded into local computer. The investigator, cleaned the data by renaming the columns and handling any missing values before uploading to Jupyter Notebook for analysis. Figure 4 shows sample data before uploading to Jupyter Notebook for analysis. Similarly, Fig. 5 shows the data loaded to Jupyter Notebook during analysis. Pandas and Multi-plotlib libraries were used for data manipulation and graph plotting respectively.

	A	B	C	D	E	F	G	H
1	survey_date	gender	education_level	age_range	chicken_number	chicken_type	farm_size(sqrm)	cleanliness_frequ
2	03/03/20	female	secondary	20-29	between_20_50	chotara	4	on_demand
3	03/03/20	female	primary	30-39	between_50_100	chotara	16	weekly
4	03/03/20	female	university	40-49	between_100_500	broiler	30	daily
5	03/03/20	male	university	40-49	between_500_1000	broiler, chotara	54	weekly
6	03/04/20	female	secondary	30-39	between_50_100	chotara	9	weekly
7	04/03/20	female	primary	30-39	between_100_500	chotara	12	on_demand
8	04/03/20	male	secondary	50-59	between_100_500	chotara, kienyeji	45	on_demand
9	04/03/20	male	primary	30-39	between_20_50	chotara	16	daily
10	04/03/20	male	university	60+	between_500_1000	kienyeji, chotara	16	daily
11	04/03/20	male	university	60+	between_50_100	chotara	9	monthly

Figure 4: Data from excel file before loading to Jupyter Notebook for Analysis

In [49]:
```
#Load Data set with pandas
df = pd.read_csv('../data/poultry_farmer.csv', delimiter=',')
```

In [7]:
```
df.head()
```

Out[7]:

urvey_date	gender	education_level	age_range	chicken_number	chicken_type	farm_size(sqrm)	cleanliness_frequency	temp_monitoring_method
3/3/2020	female	secondary	20-29	between_20_50	chotara	4	on_demand	common_sense,open_windows
3/3/2020	female	primary	30-39	between_50_100	chotara	16	weekly	open_windows, chacoal,kerosine_lamp
3/3/2020	female	certificate	40-49	between_100_500	broiler	30	daily	chacoal, open_windows
3/3/2020	male	secondary	40-49	between_500_1000	broiler, chotara	54	weekly	common_sense,chacoal, kerosine_lamp
3/4/2020	female	secondary	30-39	between_50_100	chotara	9	weekly	common_sense,open_windows,chacoal

s × 23 columns

Figure 5: Data loaded in Jupyter Notebook for Analysis

The researcher used inductive approach to analyze qualitative data. Data collected from sources other than questionnaires such as observation, unstructured interview and literature review were organized and their content was analyzed by observing, reasoning and drawing inference (Dudovskiy, 2019). Results from analysis were then compared with other secondary sources to obtain suitable methods for developing the proposed system.

3.5 Ethical Consideration

Ethical considerations are vital for successful investigation in the research. The ethical considerations include personal identity protection, individual privacy, honest, confidentiality, integrity and ensuring the rule of law is observed in any circumstance (Arifin, 2018). In this case, the researcher asked for a permission from the school of Computational and Communication Science and Engineering (COCSE) at the NM-AIST, Arusha. Since the research involved poultry animals, the researcher requested for the permission from The Northern Tanzania Health Research Ethics Committee (KENCHRIC) before conducting the research. However, before approaching the respondents in the respective districts, the investigator sought for permission from the District Executive Directors (DED) to use the research devices like camera for photo taking and recording from respondents for study purpose.

3.6 System Development Approach

Different software methodologies exist in developing software. In this study, Agile development methodology was used to develop the system because of the product's early delivery due to a timing constraint, and customer- developer interaction (Al-Zewairi *et al.*, 2017; Chandra, 2015). In Agile methodology (Fig. 6), instead of the product being delivered as a whole, it is delivered incrementally (iterations). Each iteration follows the System Development Life Cycles (SDLC), where by, a complete product increment is deployed for every iteration. Scrum is one of the agile development frameworks used in this study. In scrum, all requirements (user stories) are piled from time to time to form a Product Backlog. A few requirements (Sprint backlog) are chopped from the Product Backlog based on the highest priority. All the requirements in the sprint backlog are then planned, designed, implemented, tested and delivered as a shippable product increment in given time box (Srivastava *et al.*, 2017; Vallon *et al.*, 2018).

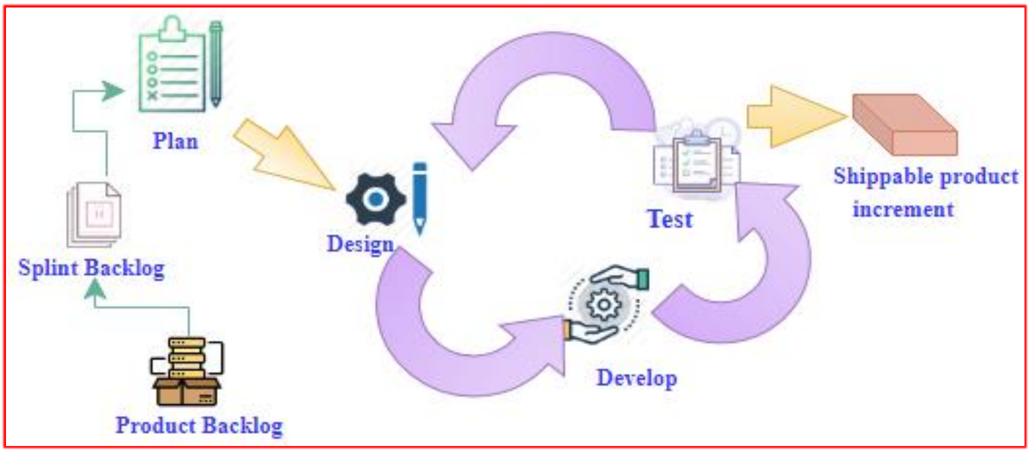

Figure 6: Agile-Scrum Software Development Model

3.7 System Modelling

Effective and efficiency systems are the results of system modelling. System Modelling is the process which is used to represent the real system using diagrams and other visualization tools for the purpose of conceiving and documenting the design of the real system. In this study, Unified Modelling Language (UML) was used to model the system. The UML diagrams used include Use case diagrams, Data Flow Diagrams (DFD), Context Diagram and Flow charts. The researcher also used My Structured Query Language (MySQL) Workbench for data base modelling. The UML and MySQL Workbench are the good computer aided software for systems modelling and design.

3.8 System Implementation

The system is divided into two parts: Hardware and software. The researcher identified hardware components required to transform the design of the system into a real system. Also, for the software implementation, different software tools and languages were identified and used to implement the web application. The choice of the components, software tools and programming languages are described in the following subsection:

3.8.1 Hardware Components

The hardware components considered are Sensors for detecting temperature, humidity and ammonia, Microcontroller unit (NodeMCU); Gateway in this case Raspberry Pi, Router, Fan and heater. These hardware components of the system were selected based on the criteria established during the collection of the system requirements using the Design Thinking approach. The Design Thinking Approach is defined in section 3.9 of this chapter. The criteria were the cost, open source, device compatibility, availability in the local market, and the efficiency and accuracy of components themselves.

3.8.2 Database Server

Database plays an important role in software systems. Letkowski (2014) defines database as an electronic shelf for storing data that are accessed by Server-Side applications. In this study, My Structured Query Language (MySQL) database management system was used for data definition, storage and manipulation. The MySQL was used because it is an open source, free, platform (operating system) independent and has many technical supporters (Letkowski, 2014).

3.8.3 Server- Side Applications

Server-Side Applications are also referred to as Back-end Applications. They are normally written using compiled or interpreted languages such as C++, Hypertext Preprocessor (PHP), and Java. Server-Side Applications are used to handle client requests to interact with the database residing in the server. In this study, PHP was used as a server-side programming language to develop a web application that the client side will use to interact with the database. The PHP was selected out of the many server – side scripting languages because of easy integration and compatibility with MySQL database (Crawford & Hussain, 2017). Furthermore, PHP is an open source framework and rich in documentation and has got a large community support that helps to improve the language and libraries for free (Crespo, 2013).

3.8.4 Client-Side Applications

Client–Side Applications are also called Front-End Applications. They are used for graphical user interfaces (GUI) to the client interacting with the web application. Client side applications are developed using Hyper Text Mark-up Language (HTML), Cascading Style Sheet (CSS) and Java Script (JS) language. However, in this study, the GUIs were developed using Bootstrap framework since it is an open source HTML, CSS and JS front end framework for building responsive client- side applications (Otto & Thornton, 2015). Moreover, JQuery and Ajax libraries were used for manipulating HTML elements and loading data dynamically- without refreshing the page respectively (Crespo, 2013).

3.8.5 Embedded Software

Embedded Software resides on the micro-chip to perform specific function and for the hardware device to interact with other devices (Heath, 2002). Unlike application software, embedded software is designed for specific chip and serves as Operating System (OS) itself. In this study, the researcher developed an embedded software to run on Microcontroller chip for reading and transmitting data from sensors to gateway. In addition to that, the software controls fan and heater in the poultry house depending on the housing conditions. The embedded software was written using C++ language and Arduino Integrated Development Environment (IDE) (Dhingra *et al.*, 2019).

3.8.6 System Security Implementation

Data/information could be tampered during transmission or accessing (data at rest). During transmission, data could be vulnerable to man-in-middle attacks that could mislead the client's data. For example, an attacker could report low temperatures all the time while in the field, the temperature is high. Tampering to the system in this way can cause the death of flocks and a significant loss to the farmer. Also, during accessing, a brute force attack could affect the confidentiality, Availability, and Integrity of the data. In this work, MFA and SSL have been used to ensure data security while accessing the system and during transmission, respectively.

multifactor authentication was implemented to enhance data security while accessing the system. In this study, codes generated from the user's smartphone were used to implement a multifactor authentication. When the user logs in using username and password, the system simultaneously verifies the device, generates the code, and requests the user to enter the code sent to them and then granting the user to access the system upon meeting the predefined criteria.

Secure Socket Layer (SSL) Security algorithm was used to secure data during transmission. The SSL is widely used in cryptography for encrypting and decrypting messages because of its enhanced security. The SSL certificates contain data about the entity the certificate was delivered for, digital signature, and Public key (PK) for encryption and digital signature proof (Nakhuva & Champaneria, 2017). The SSL certificates was installed in the webserver; the same keys are uploaded to the NodeMCU and Gateway when uploading the firmware software to these devices. The node and gateway use these certificates to encrypt and decrypt the message sent or received from the cloud server.

3.8.7 Algorithm for a System to Work Offline and Online

Two algorithms were developed, the main algorithm for the system to work in both offline and online, and the synchronization subroutine algorithm that runs continuously to synchronize data between the local server and cloud server. The main algorithm was implemented using PHP language, while the synchronization algorithm was implemented using JavaScript language and Node.js framework. The synchronization algorithm runs continuously by checking if there is any data in the local server that need to be synchronized or deleted. The data is deleted locally if it has stayed over two days. The flow charts and algorithms are described in detail in section 4 of results and discussion.

3.9 System Design Architecture

Design thinking approach was applied in realization of system design architecture and development. The design thinking approach is the paradigm which allows individuals to focus on people's needs when designing and implementing the products (Kumar *et al.*, 2020). In realization of this concept, the researcher created mock-ups, and poultry farmers in Tanzania were engaged to give the requirements on how the system should be. From the requirements, a high-level system architecture diagram in Fig. 7 was developed, and the devices were identified based low cost, open source, device compatibility, availability in the local market, and the efficiency and accuracy of components themselves.

From Fig. 7, a node (with sensor, fan, and heater) collects sensor data from the poultry house and sends them to a local server (RPi gateway). The local server aggregates the data from different nodes, forwards to the cloud server using Secure Hyper Text Transfer Protocol (HTTPs) messaging protocol, the cloud server receives, processes data, sends back a response to the local server, and ends the connection. A web portal platform facilitates farmers to monitor and control the farm conditions in real-time. The designing of the system allows farmers to access it online or offline, as described clearly in results and discussion section.

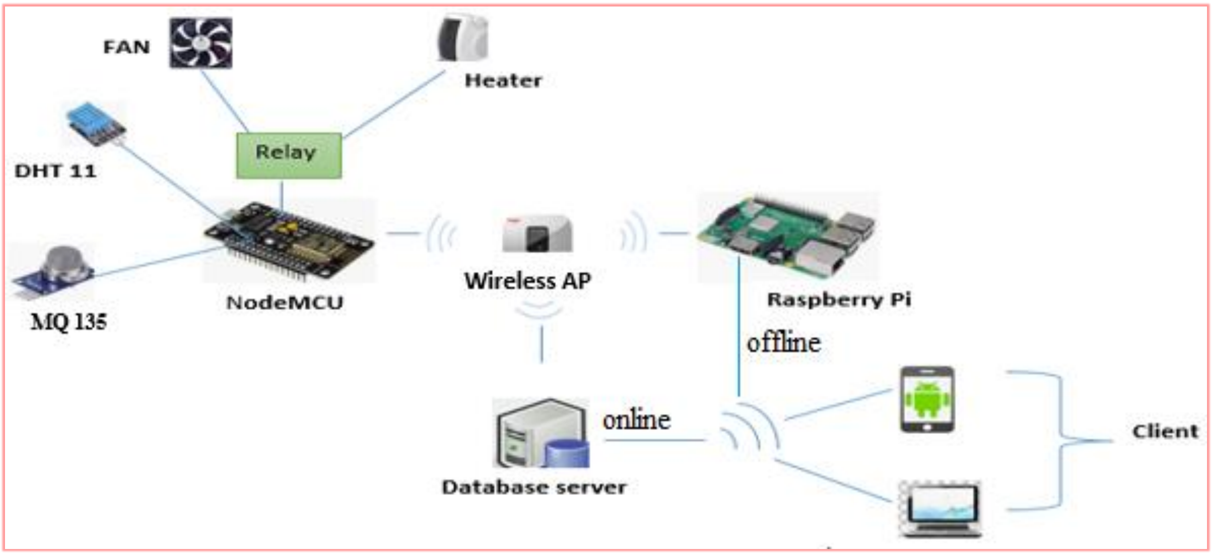

Figure 7: A high level System Design Architecture

The system design architecture consists of three parts which are sensing unit, transmission unit, and monitoring and control unit. These parts are described as follows:

3.9.1 Sensing Unit

The part describes each component in the sensing unit part. The components described are NodeMCU, Sensors, and Relay. The researcher then presents, the general circuit diagram (Fig. 14) that shows how these components were interconnected together.

(i) NodeMCU

The NodeMCU ESP8266 was used in our work. The NodeMCU is an open-source IoT development board for developing or prototyping IoT systems. The kit has a RAM of 128KB and 4MB of flash memory for data and program storage. Also, it has Wi-Fi and Bluetooth microchips built-in and deep sleep operating features that make it suitable for use in our work. The firmware of the NodeMCU that runs on the ESP8266 Wi-Fi microchip uses the Lua scripting language. The node uses 3.3V power and can be driven directly from the adaptor or PC through Micro USB Port or from the external power source through Vin Pin. Loading of NodeMCU microcontroller with software is necessary for its operation (i.e., collecting and forwarding the sensed data to the local server (Prayogo *et al.*, 2019). To achieve this, we developed a C++ program using the Arduino IDE and uploaded it to NodeMCU.

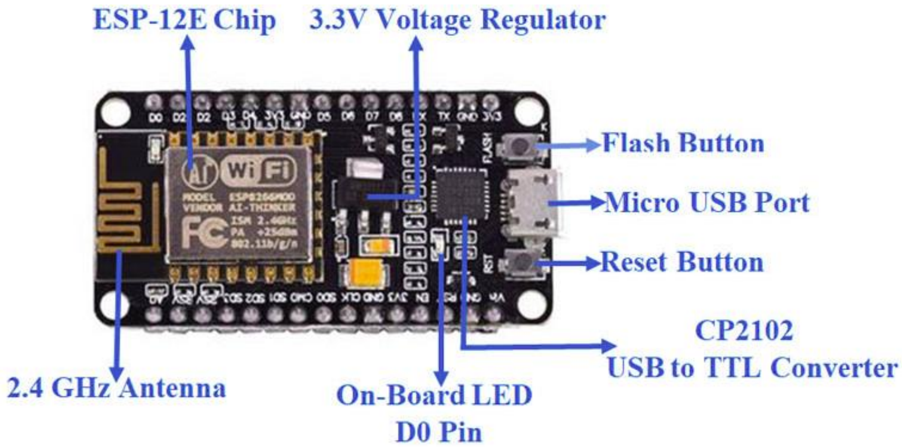

Figure 8: The key parts of NodeMCU (NodeMCU ESP8266 Pinout, Specifications, Features & Datasheet, n.d.)

(ii) Sensors

These are devices that are capable of responding to external changes in the environment (Islam *et al.*, 2019b). In this work, we used the DHT11 sensor for reading temperature and humidity. The DHT11 sensor (Fig. 9) reads the accurate temperature, hence does not need calibration. The MQ135 sensor (Fig. 10) was used to read Ammonia gas in the poultry house. The detection range of MQ135 is 10 ppm – 300 PPM. The MQ135 sensor is highly sensitive to ammonia and other harmful gases such as benzene and carbon dioxide.

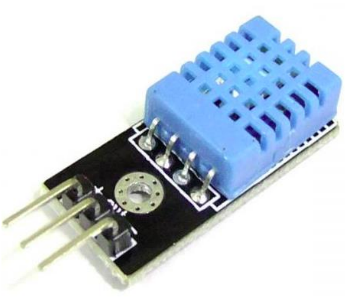

Figure 9: DHT11 Sensor module

Figure 10: MQ135 gas sensor

(iii) Relay Switch

Relay switch is an electromagnetic switch which operates under minimum voltage of 3.3V – 5V and allows a user to control high voltage consuming devices by using a low voltage consuming microcontroller. It has four input pins VCC, IN and GND and the three output pins Normal Closed (NC), Common (COM) and Normally Open (NO). The VCC, IN and GND pins were connected to VIN, D6, and GND pins of the NodeMCU respectively. Figure

11 shows a 5V 1 channel relay module which is used to drive devices that operates under 10A@250VAC, and 15A@125VAC.

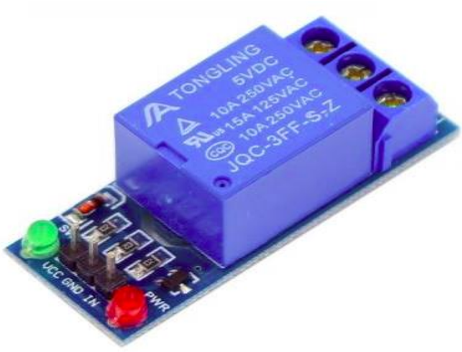

Figure 11: A 5V 1Channel Relay Module

(iv) Circuit diagram for a sensing unit

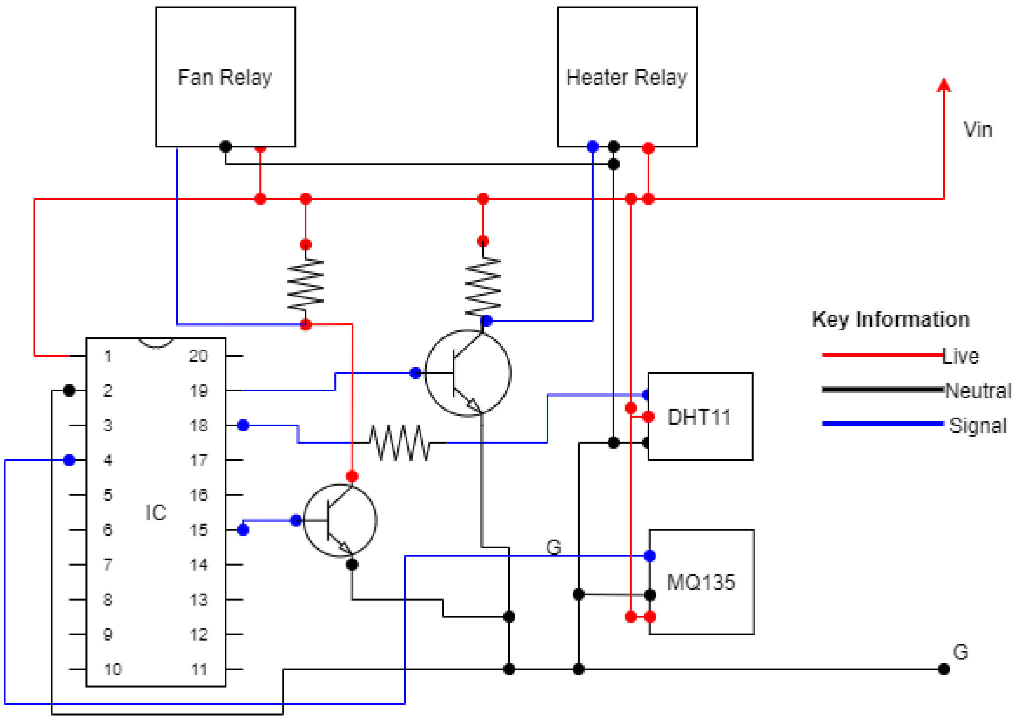

Figure 12: Circuit diagram for sensing unit

3.9.2 Transmission Unit

This part consist of Raspberry Pi and Wireless Access Point. Various communication protocols for device to device communications are also described.

(i) Raspberry Pi

Raspberry Pi (RPi) is a small, low-cost system on chip (SOC) device with all capabilities of a standard computer. In this work, the Raspberry Pi 3 Model B+ was used as a local server to allow our system to work offline as well. Raspberry Pi 3 Model B+ (Handigolkar *et al.*, 2018) is the latest version with the following key features; it has built-in support for 2.4/5GHz 802.11b/g/n/ac Wi-Fi connectivity, 4 USB 2.0 ports, Low energy Bluetooth 4.1 module , and 300 Base T Ethernet port. Other features encompass ARM Cortex-A53 CPU of 1.4GHz speed, Micro SD card, HDMI audio, and video output, has 40 General Purpose Input Output (GPIO) pins for digital input and output operating at 5V/2.5A.

(ii) Wireless Access Point (WAP)

Wireless Access Point (WAP) facilitates the connection of NodeMCU ESP8266 to the RPi gateway through 2.4 GHz WI-Fi protocol. Also WAP connects a gateway to the cloud server.

(iii) Communication Protocols

Various communication protocols exist for device-device, device- processor, or inter-processors communication. The Inter-integrated Circuit (I^2C) protocol is being to facilitate the communication between the NodeMCU processor and sensors. Unlike UART and SPI, I^2C uses two pins, which are CL and DAL, to communicate with the processor (Leens, 2009) and (Circuit Basics, 2017).

The Wi-Fi protocol was also used in our work to enable wireless communication between nodes and gateway as well as a cloud server. The Wi-Fi is a good, reliable, and secure protocol for sending and receiving data between devices. The Wi-Fi is now ubiquitous and the cheapest wireless network protocol after Bluetooth. Near all smart devices come with Wi-Fi embedded inside (Camps-Mur *et al.*, 2013). Such smart devices include smartphones, laptops, smart watches, et cetera.

3.9.3 Monitoring and Control Unit

This is application software that allows the user to monitor and control the sensing unit remotely. Use case diagram and database schema were developed before implementing the actual web portal. The diagrams saves as reference when implementing the system.

(i) Use case Diagram

Use case diagrams shows the interaction between the actor and the system. They save as the reference during User Interface implementation. This study identified two primary actors (Farmer and Admin). The core functions of the farmer are to monitor, control, and manage the farm conditions. While the core functions of Admin are to register and manage a farmer, coop, Gateway, and nodes plus all activities a farmer performs as shown in Fig. 13.

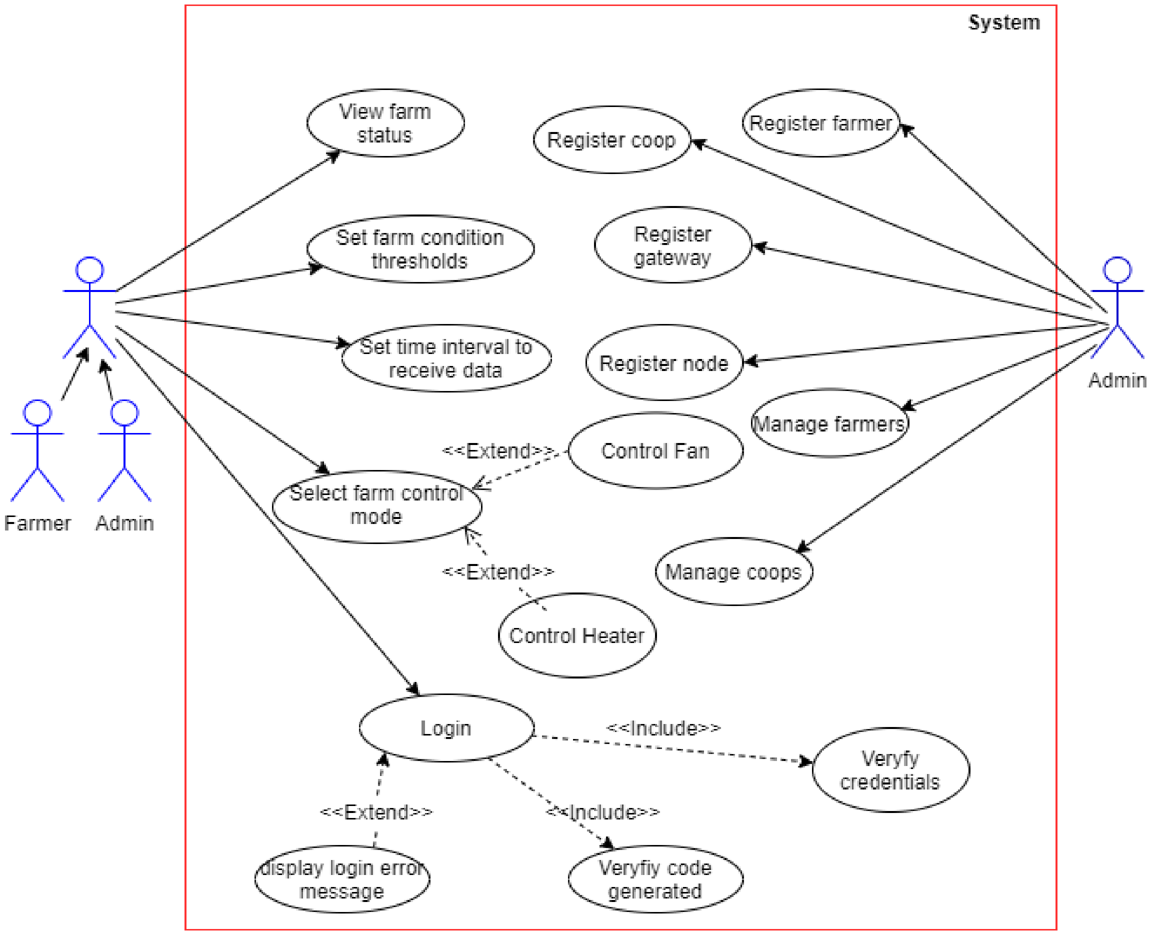

Figure 13: Use case diagram for Farmer and Admin

(ii) Database Schema

Database schema shows the structure to which data are organized in Data base management system. The UML modeling language was used to design the use cases diagram, while Database schema (Fig. 14) was designed using MySQL Workbench. The key tables identified are Farmers which keep track of registered farmers, coops which keep tracks of registered coops, Gateways that records registered gateways, coops_data that stores parameters value in the field, coop thresholds table holds of parameter thresholds set by the user, and the Nodes table holds of information of every registered node.

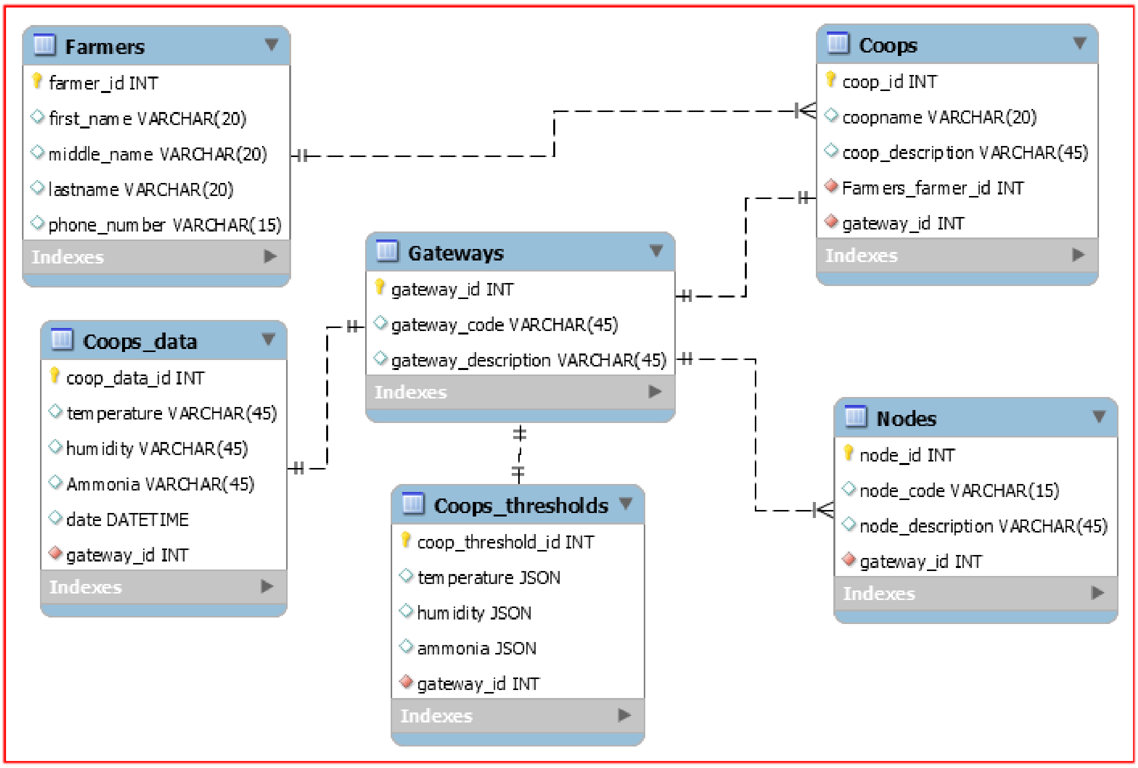

Figure 14: Data base schema

3.10 System Testing and Validation

System testing was done in phases, where the first face was unit testing, the second integration testing, and the last phase was system testing. During Unit testing each module was tested independently before integrating with other modules. Integration testing was performed after integrating one module and the other. This was done to eliminate any logical or syntax errors that could arise due to modules compatibility. When all modules were successful integrated, a black–box testing methodology was used to verify the functionalities that were identified during requirement gathering. Black–box testing methodology is one of the testing approach in which the tester tests the system without having the knowledge of the internal codes.

Lastly, validation testing was done by deploying the system to a representative of farmer and let them interact for a period of seven days.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Results

4.1.1 Findings from Poultry Farmers

A survey was done to find out the strength and weakness of the current methods farmers use to maintain poultry house conditions. Furthermore, the study was conducted to determine requirements of the proposed system that would be an appropriate solution to both small scale and large scale poultry farmers.

(i) Poultry Farmers' Demographic Information

In this study, out of 120 respondents, 75(62.5%) were male while 45 (37.5%) were female as indicated in Table 2. The maximum age group was 60 years and above while the minimum age group was 20 to 29 years. The majority of respondents were aged between 30 to 39 years, where they constituted 35% of the whole sample size. Also, the findings indicated that many poultry farmers have primary education level followed by those with secondary education.

Table 2: Respondents 'demographic information

Demographic Characteristics		**Respondents**	**Percentage (%)**
Gender	Male	75	62.5
	Female	45	37.5
Age (in Years)	Less than 20	0	0.0
	20 -29	21	17.5
	30 -39	42	35.0
	40 -49	24	20.0
	50 -59	21	17.5
	60 and above	12	10.0
Education	non- formal	0	0.0
	Primary	35	29.2
	Secondary	31	25.8
	Certificate	13	10.8
	Diploma	17	14.2
	University	24	20.0

(ii) Respondents' House Conditions Monitoring Methods

Monitoring methods refers to ways the farmers use to detect or determine the status of the poultry housing conditions that plays a significant role to the health of flocks. During the study it was revealed that about 86.5% of poultry farmers use common sense to monitor the status of temperature, humidity and ammonia in the poultry house while 13.5% use thermometer to monitor only temperature in the poultry house as indicated in Fig. 15.

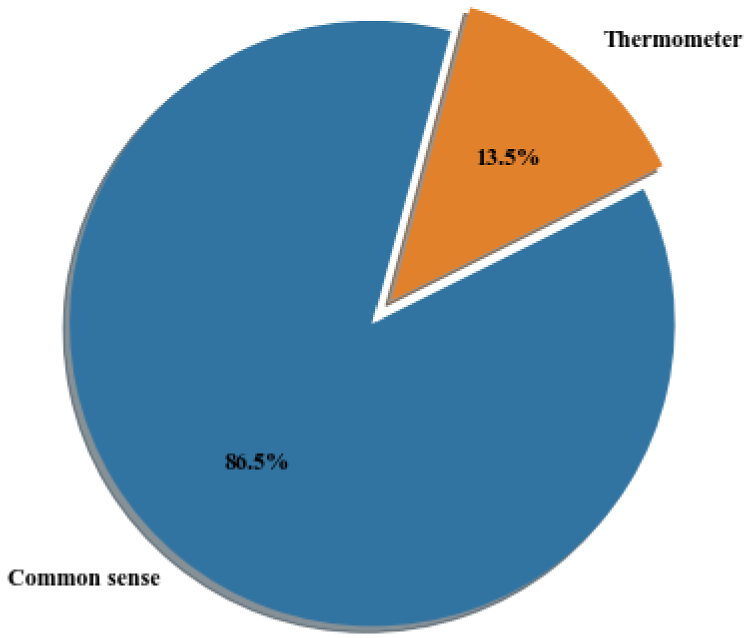

Figure 15: Respondents' house conditions monitoring methods

(iii) Respondents' Methods to Maintain Poultry House Conditions

The study found that there are various methods poultry farmers use to maintain conditions in poultry house. These methods encompasses charcoal, kerosene lamps, closing/opening of curtains and electricity. About 34.5% use charcoal to maintain the temperature, humidity and ammonia in the poultry house while 47.0% use curtains. Furthermore, the study found that, 13.4 % of the farmers use kerosene lamps while very few about 4.5% use electricity bulb to maintain the conditions inside the poultry house. Farmers reported that using electricity is costly than alternative source of energy such as charcoal. These findings are summarized in Fig. 16 and reveals that there is a need to develop a system that would be a potential solution to farmers instead of using charcoal and kerosene lamps which are environmentally unfriendly.

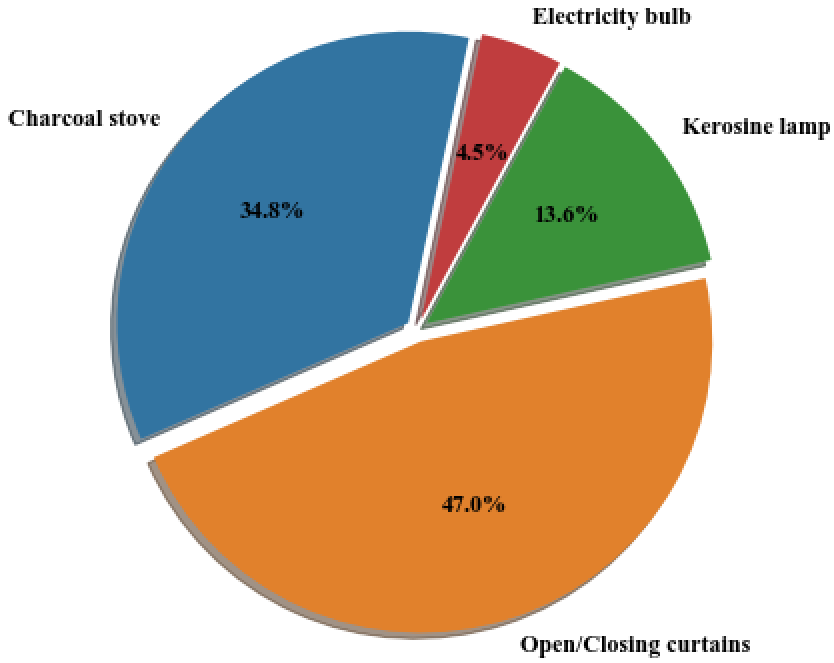

Figure 16: Respondents' methods to maintain Poultry house conditions

(iv) Respondents' Awareness to Smart Poultry Monitoring Systems

The study focused on knowing the probability of respondents 'general awareness of Smart poultry Monitoring Systems. As it is presented in Fig 17, out of 120 participants 62.5% are not aware of smart poultry monitoring systems that are developed for maintaining house conditions in farm. Moreover, 37.5% of the respondents said they are aware of the monitoring systems, particular the incubator which is used for hatching eggs and regulating temperature for chick's growth.

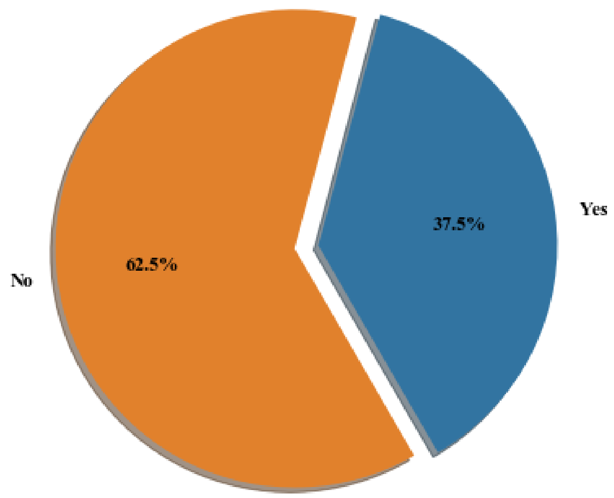

Figure 17: Respondents' awareness to Smart Poultry Monitoring Systems

(v) Factors Affecting Adoption of Smart Poultry Monitoring Systems in Tanzania

The study revealed that, out of many factors, 43.0% of respondents reported cost as the leading factor which hinder the adoption of automated systems among poultry farmers followed by lack of knowledge, which is 38.0% while the least of respondents (7.3%) reported that the systems lack some features to suit the local demand as shown in Fig. 18.

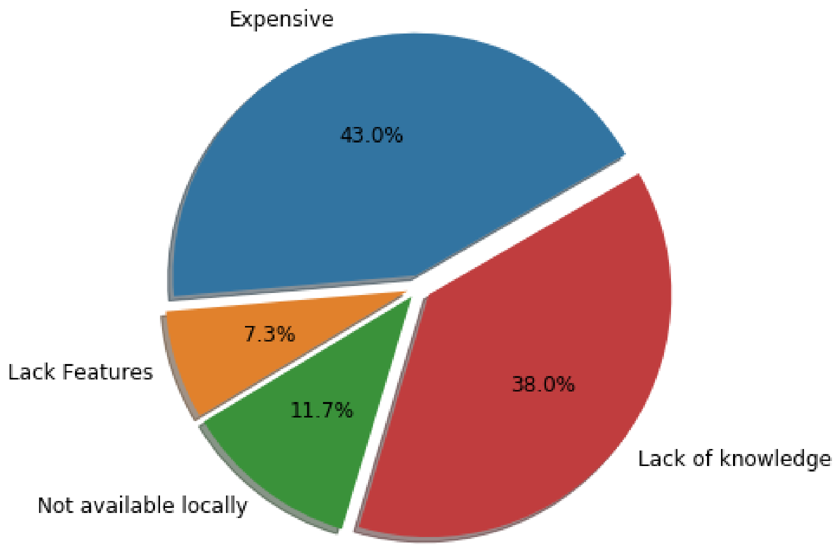

Figure 18: Factors affecting adoption of smart poultry monitoring systems in Tanzania

(vi) Respondents' Perspective on the Need for a Proposed IoT based System

The study sought to find the perception of the respondents on the proposed IoT based system. The data was collected after having explained fully on the anticipated system using mockups and the initial prototype. Of the 120 participants, 72.5% accepted that the proposed IoT based system would be supportive in monitoring and controlling the poultry house conditions while 17.5% rejected and 10.0% were neutral in a sense that they neither accepted nor rejected. The results are summarized in Fig. 19.

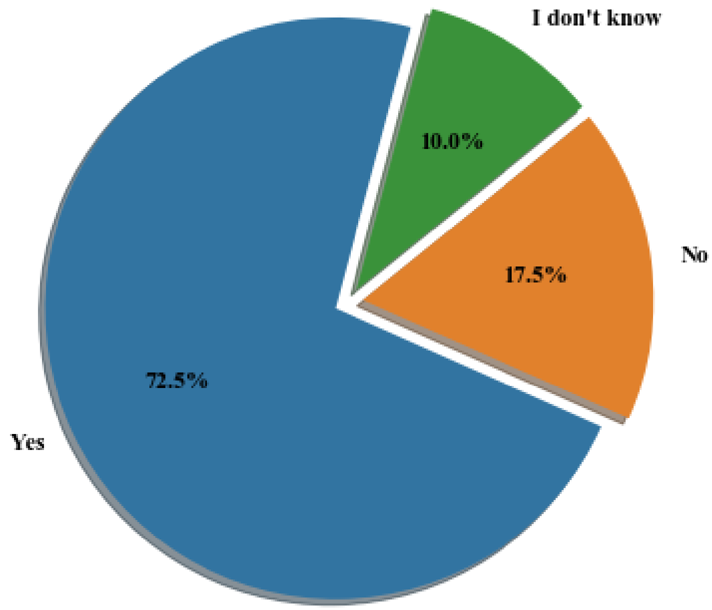

Figure 19: Respondents' perspective on the need for a proposed IoT based System

4.1.2 Requirements for the Proposed Internet of Thing Based System

The requirements are grouped into functional and non-functional requirements. Functional requirements describes what the system should do to deliver the service intended by the user while Non-functional requirements describes the attributes of the system for successful operation. For example, availability, reliability, scalability and security issues are non-functional requirements. This study used both primary and secondary sources to determine the requirements for the proposed IoT based system for Environmental conditions monitoring in poultry house called ***Tanzakuku***. The primary source involved interviewing the poultry farmers by describing the key features of the system and leaving them rate those features. Secondary source involved reading from documents, manuals and other retrievable documents. Using these techniques, Table 3 and Table 4 summarize the functional and non-functional requirements for ***Tanzakuku***.

Table 3: Proposed Functional requirements (FR) for Tanzakuku

FR#	Description
1.	The system should allow a farmer to monitor temperature, humidity and ammonia remotely in both graphical and textual format
2.	The system should be able to work in off line mode provided the client is within the Local Area Network (LAN) environment. This is especially when there is no internet connection; there should be a local server that will synchronize with online server when internet connection is back.
3.	The system should allow a farmer to switch on/off the fan when the temperature is low or high respectively.
4.	The system should allow the user to enable automatic monitoring and control of the temperature and humidity.
5.	The system should allow a farmer to monitor and control temperature and humidity parameters of different distributed coops using the same dashboard.
6.	The system should authenticate the farmer before using the system.
7.	The system should allow a farmer to view monthly trend of temperature, ammonia and humidity in graphical format (line graph).
8.	The system should allow the system administrator to create, suspend or remove a user account
9.	The system should allow admin to add new nodes (Sensor units) to the system.
10.	The system should allow admin to view the status of all nodes with their respective clients/farmers.
11.	The system should allow admin to enable/disable any client's node for maintenance.
12.	The system should allow admin to register new user/client to the system with the respective coops they own.
13.	The system should be able to be powered by micro solar panels for cost- effective power.

Table 4: Proposed Non- Functional Requirements (NFR) for Tanzakuku

NFR#	Description
1.	The system contents should be easy to learn and use.
2.	The interface of the system should be aesthetic, that is attractive and interactive.
3.	The web system should be compatible with all browsers and mobile application should run on android OS.
4.	The system should be reliable, scalable and affordable i.e. One can get the system and pay it in installments.
5.	The system should be secured at both data access and data transmission levels.
6.	The application should be available at all times.
7.	The system should be easy to maintain. When one service is under maintenance should not hinder other services from operation.

4.1.3 The developed System Results

This section describes the system developed based on the analysis and design done in section 3.9. The IoT based system ***(Tanzakuku)*** comprises hardware and software as described in section 4.1.3 (i) and 4.1.3 (ii) respectively.

(i) The Hardware System

The hardware setup (Fig. 20a) consists of nodes (with sensors, a heater, and a fan), RPi gateway, and WAP. The setup was for testing purpose before deploying to the field. Each node was then placed in a low cost- water proof container (Fig. 20b), then deployed in the field.

Figure 20 (a): Hardware setup

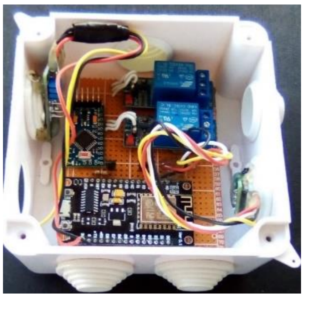

Figure 20 (b): Sensor Node

(ii) The Web Application

Figure 21 shows an authentication panel where user supplies username and password to login into the system. They then supply the code (One-Time Password) sent to their mobile phone for verification. The user is then presented with the main dashboard (Fig. 22) upon successful login. The main dashboard contains functionalities that allows the user to view the status of parameters in the house both graphically and textually, add new coop, add new node and the like.

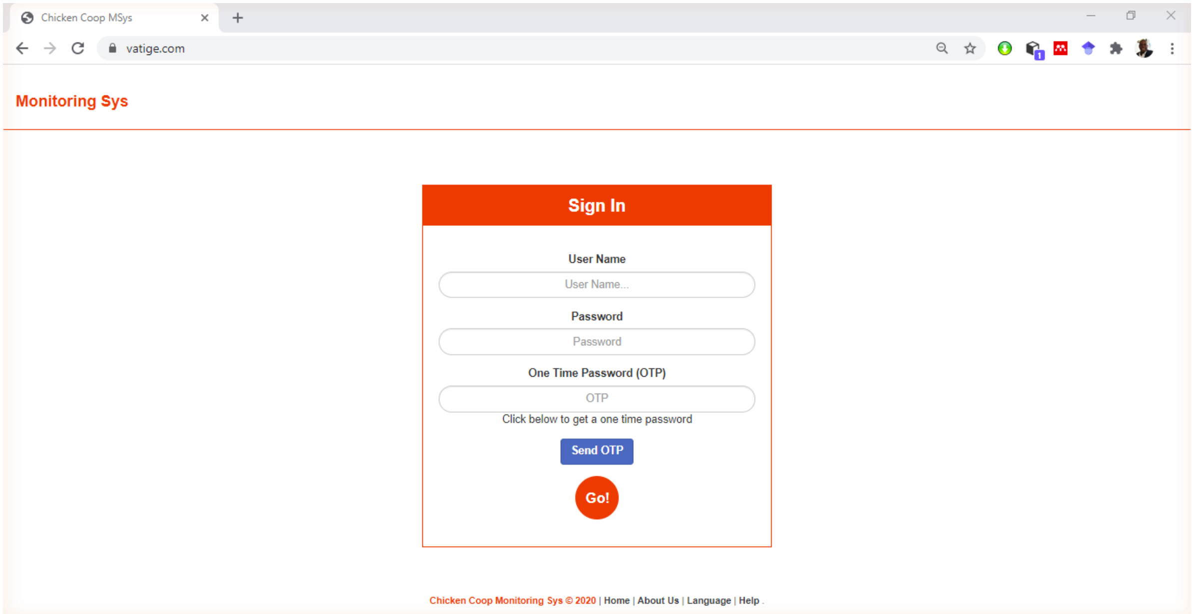

Figure 21: Authentication panel

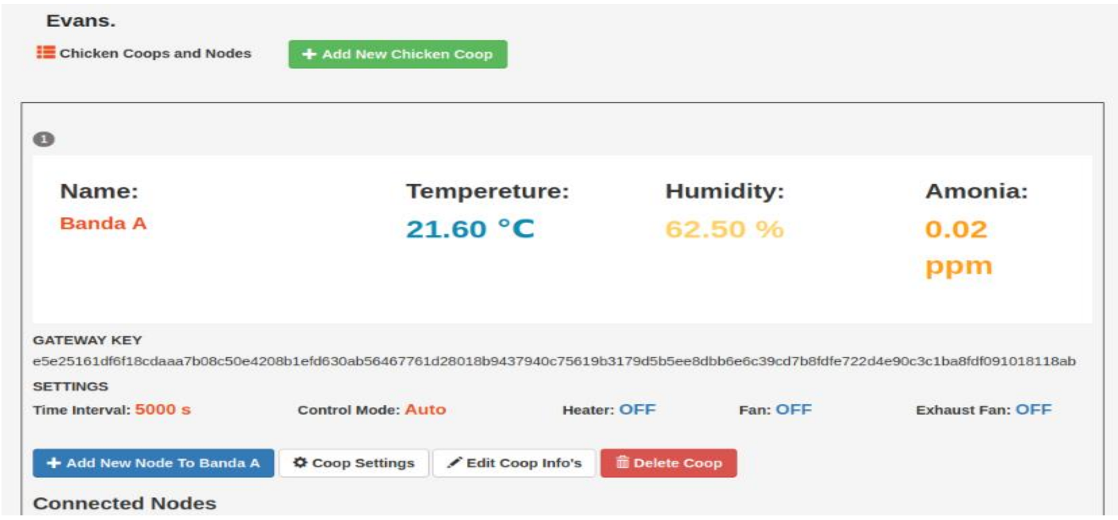

Figure 22: Main Dashboard of Tanzakuku

4.1.4 Algorithm for a System to Work Offline and Online

By default, unless configured by the farmer, the node forwards data to local server after every 15 seconds. When there is no Internet connectivity, the node still send data to the local server (Raspberry Pi) since both devices are in the same LAN network. A client still will access the information and control the parameters provided they are in the same LAN environment. Figure 23 shows a flow chart for a system to work in both offline and online mode with its equivalent algorithm in Table 5. In this algorithm, the local server forwards data to cloud server provided that the online mode is enabled and there is internet connectivity, otherwise the system stores data locally until processed by a synchronization subroutine whose flow chart and algorithm are shown in Fig. 24 and Table 6 respectively. The synchronization subroutine forwards data to cloud portal provided that there is data to be synchronized, the online mode is enabled, and the internet is available. Thereafter, the data is deleted should it be stayed more than two days in order to prevent the local server to run out of memory. Otherwise the system keeps monitoring the locally-stored data.

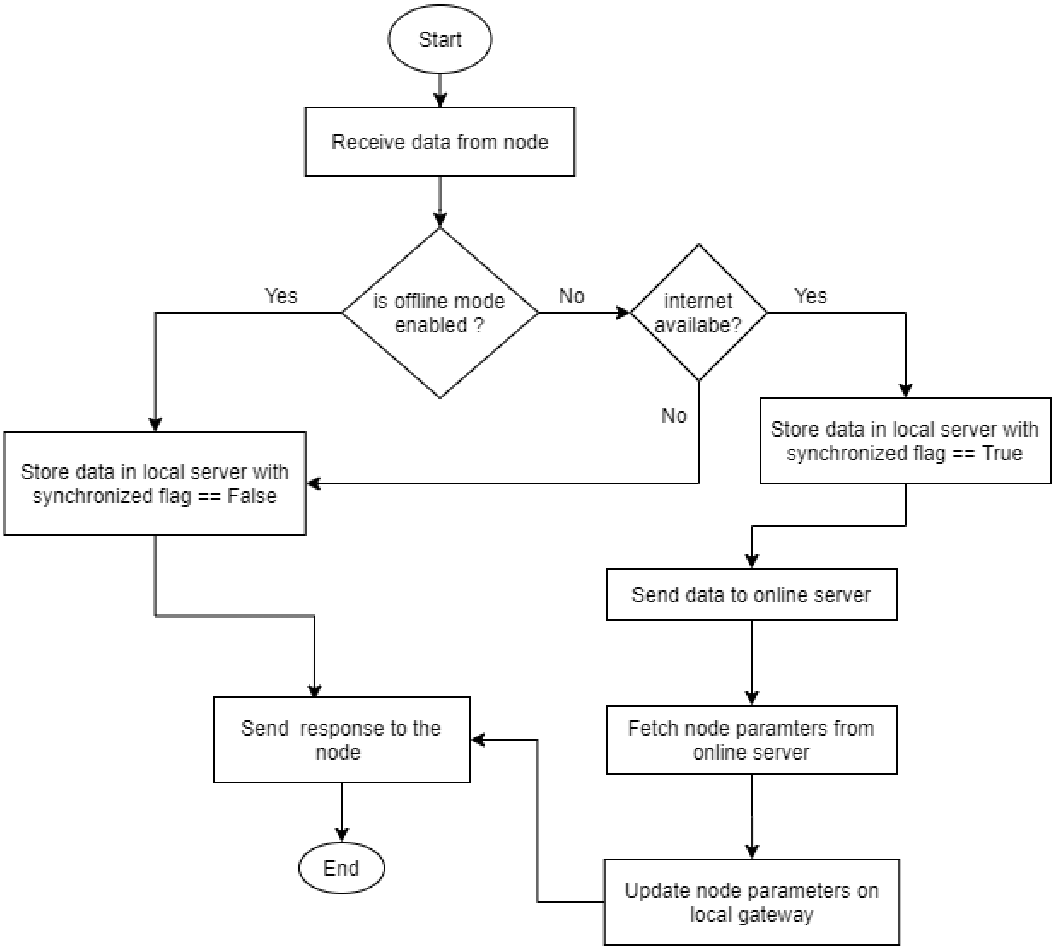

Figure 23: A flow chart for a system to work in both offline and online mode

Table 5: An algorithm for a system to work offline and online

Algorithm 1-System to work offline and online

Receive data from a node - *after every predefined time interval*
If *is_offline_mode == True* || *is_internet_available ==False* **Then**
    Data_synchronized_flag ← False
    ***S**tore data in local DB server*
    Send response to the Node
Else
    Data_synchronized_flag ← True
    Store data in local DB server
    Send data to online DB server
    Fetch node parameters from online server
    Update node parameters on local server
    Send response to the Node
End If

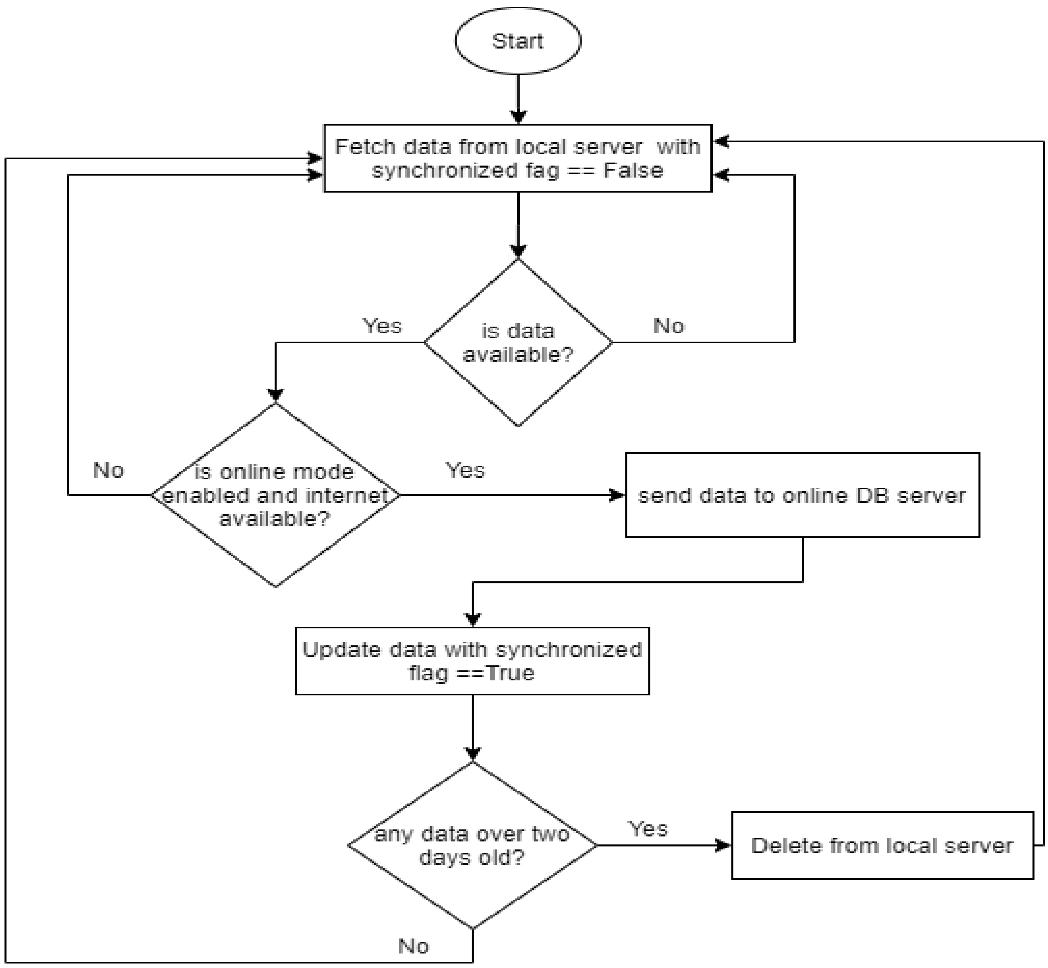

Figure 24: A synchronization flowchart to data between the local and online server

Table 6: A synchronization algorithm for data between the local and online server

Algorithm 2 – A subroutine to synchronize data between the local and online server

```
1.  Do While ( is_data_available == False)
2.        µ ← Fetch data with data_synchronized_flag == False
3.        If µ ! = Null Then
4.            is_data_available ←True
5.        Else
6.            is_data_available ← False
7.        End If
8.  End While
9.  If is_offline_mode==False && is_internet_available ==True Then
10.     Send data to online DB server
11.     Data_synchronized_flag ← True
12. End If
13. If is_data_over_two_days == True Then
14.     Delete the data in the local DB server
15.     Call step 1
16. Else
17. Call step 1
18. End If
```

4.1.5 Reliability Testing Results

Reliability is the statistical measure that defines a degree of the system to operate correctly over a given time of operation (Menčík, 2016). To determine and improve the reliability of our proposed solution, we set up an experiment and recorded observed reliability and failure for seven days, as tabulated in Table 7. However, we made the following assumptions in the experiment:

(i) The system reliability is observed every two hours from 0800 – 0000 hours in a day for seven days. In this case, we have 16 hours of operation per day.

(ii) In every observation, the system is marked “1” if it performs all of the following core functionalities correctly: (a) monitor parameters, (b) control parameters, and (c) works in offline, and online mode. Otherwise, it is marked “0”. In summary, 1 = reliable and 0 = failure.

(iii) If the system is marked “0” then, the failure plus fixing time is regarded as to occur within one hour.

Table 7: Observed operation of the system for seven days

Time	day 1	day 2	day 3	day 4	day 5	day 6	day 7
08:00	1	0	1	1	1	1	1
10:00	1	1	1	1	1	1	1
12:00	0	1	1	1	1	1	1
14:00	1	0	1	1	1	1	1
16:00	0	1	0	1	1	1	1
18:00	1	1	1	1	1	1	1
20:00	1	1	1	1	1	1	1
22:00	0	1	1	1	1	1	1
00:00	1	1	1	1	1	1	1

Based on the data in Table1, we used the following equations to estimate the system reliability.

Total operating hours (Nh) = 16hrs/day x7 days = 112 hours (1)

$$\text{Down time (Nf)} = \text{Number of cells with } 0's = 6\text{hrs} \quad (2)$$

$$\text{Reliability } R(t) = 100 * (Nh - Nf) / Nh \quad (3)$$

The computation of the system's reliability for seven (7) days operation using Equation (1) to (3) reached 94.6%. However, it was learnt from the experiment that the system's reliability was mainly affected by the System source of power, physical safety of sensors due to harsh environmental conditions and position of the nodes as discussed in detail in section 4.8.

4.1.6 Time Cost Evaluation

It was evaluated at what percentage, this proposed system saves time in monitoring and controlling poultry farm conditions compared to the traditional system. To achieve this, assumptions, mathematical Equations, and data from the survey were developed. Data from the survey indicated that a farmer uses an average time of eight minutes to visit the site (poultry house), five minutes to monitor the conditions and 12 minutes to control the parameters (e.g., opening/closing the curtains) as summarized in Table 8. Furthermore, we assumed the system setting up for both tradition and the proposed system is constant. Also, it was assumed that this proposed system uses two minutes to send data to the saver for simplicity, although by default, unless configured by the farmer, it is 15 seconds.

Table 8: Average time required in Traditional system and proposed system

Task	Average time (minutes)	
	Tradition system	Proposed System
Visiting the Site	8	0
Monitoring conditions	5	2
Controlling conditions	12	2

Mathematical Equations:

$$T_o = V_s + M_c + C_c \quad (4)$$

$$T_1 = M_c + C_c \quad (5)$$

$$T_s = \left[\frac{T_o - T_1}{T_o}\right] * 100\% \quad (6)$$

Where, T_o = total time used by the traditional system, T_1 = Total Time used by the proposed system, T_s =Time in Percentage saved by the proposed system compared to the traditional system, V_s = time to visit the site, M_c = Time to monitoring conditions, and C_c = Time to control conditions. By using equations (4), (5), and (6), it was found that the proposed system saves 84% of the time compared to the traditional system.

4.1.7 Labour Cost Evaluation

We assessed at what percentage does this proposed system saves labour costs compared to the traditional system. To achieve this, mathematical Equation (7) with the following assumptions was developed; both systems require labour to operate, and a farmer owns three (3) distributed coops. Table 9 summarizes the task and number of labourers required in tradition and proposed system.

Table 9: Number of labourers required in Traditional system and Proposed system

Task	Number of Labourers	
	Tradition system	**Proposed System**
Monitoring and controlling conditions in three distributed coops	3	1

$$C_l = \left[\frac{N_t - N_p}{N_t}\right] * 100\% \quad (7)$$

Where, C_l = Percentage labour cost the proposed system saves, N_t = number of labourers in the traditional system, and N_p= Number of Labourers in the Proposed system.

Using Equation (7), the proposed system saves 66.7% of labour costs than the traditional system. However, 66.7% labour cost saved depends on the number of coops the farmer owns. For instance, the proposed system would save up to 75% of the traditional system's labour cost for a farmer owning four (4) coops.

4.1.8 Scalability

We developed a scalable system that allows farmers to monitor as many coops as they own in real-time, switch a system to automatic or manual control, set threshold values based on the age of chickens, and set time intervals for the sensor to send data. For instance, when the user enables automatic mode, the device can spontaneously monitor and control the parameters. The system automatically checks if the temperature is out of the acceptable range and controls by switching on a fan or heater. When manual mode is enabled, the tool monitors only the parameters and waits for the user to issue the command to control those parameters. Generally, the same software system can be utilized by the farmer to keep chicken of different ages and add new coops as they wish without affecting the software system.

4.2 Discussion

A comparison graph between the numbers of the hours the system was in operational against the down time is shown in Fig. 25. From the graph it appears that in day1 and 2 the system went down three times in each day. However, the down time decreased as the number of days increased. This is because the following factors identified to hinder reliability were corrected as follows:

4.2.1 Source of Power

Some failures happened during the night because of a power cut off. To rectify the power issue, both Micro- solar panels and LiFePO4 rechargeable batteries were both used instead of depending only to Micro-solar panels. Also a deep sleep feature in NodeMCU was implemented to allow nodes to sleep deeply when waiting for time to send data. Furthermore, 2.4 GHz Wi-Fi technology was opted for, which uses low power compared to 5 GHz Wi-Fi technology.

4.2.2 Position of the Nodes

Initially, we put the sensor on the ground. However, they were disturbed by chicken. Therefore, the nodes were repositioned by hanging them from the roof using a reliable sword.

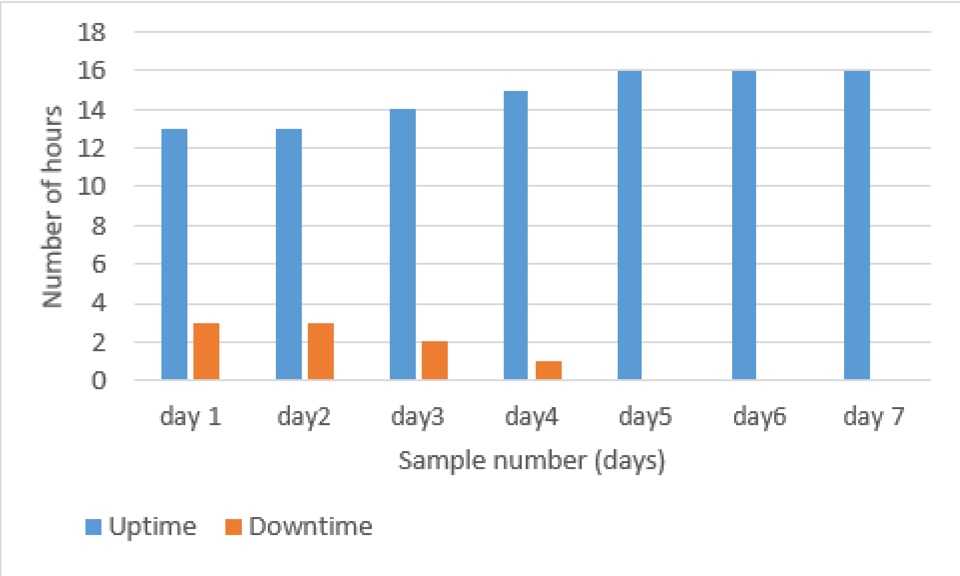

Figure 25: A graph of number of hours the system was in operation and down time

Furthermore, there are two ways to access the system: online and offline mode. The online mode is used when a farmer is distant from the coop, where they are disconnected from Wi-Fi signals. The offline mode is used when the farmer is only at home and does not have the internet to access data. However, the detection of the LAN signals depends on the coop's distance from the farmer's house. The distance should be less than 100 m since 2.4 GHz Wi-Fi signals range up to 100 m. In this regard, the WAP should be positioned in safe environment free from interference, diffraction and attenuation.

Likewise, the proposed system has not been evaluated for its efficiency in increasing poultry productivity against traditional methods practiced by poultry farmers. Therefore, further experiments are needed to compare the extent to which the proposed system increase poultry productivity against using the traditional methods. This can be done by setting up two similar poultry houses, one utilizing the proposed system and the other house utilizing traditional methods at constant feed, number of chicken, and chicken age.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This study focused to develop IoT based system for environmental conditions monitoring in poultry house. Environmental conditions such as temperature, humidity and ammonia seriously affects the health of the poultry thus decreasing its productivity. A survey was carried out in Arusha and kilimanjaro regions to specifically determine the major factors that affect the adoption of smart poultry monitoring systems in Tanzania, and how can the IoT based system be developed to suit the Tanzania context by addressing gaps in previous researches and weakness from traditional systems used by poultry farmers. The findings indicated that lack of awareness, lack of features to suit the local contexts and costs of available automated systems are among key factors that affect the adoption of smart poultry monitoring systems in Tanzania.

The study then developed an IoT based monitoring system that was found to save both time (84%) and labor cost (66.7%) as compared to traditional system. Specifically the study contributes the following: a tool to monitor environment conditions remotely and securely; A tool to control the parameters remotely; a scalable platform that allows a farmer to monitor and manage more than 100s distributed poultry farms using single dashboard (IoT platform); and lastly the study has proposed an algorithm that allows a system to work both on-line and offline. We believe that the proposed solution would be potential to poultry farmers in Tanzania.

5.2 Recommendations

To improve security in our proposed system, incorporating emerging security technologies for distributed networks such as block chain could be an exciting area for future works. Also, incorporate Artificial Intelligent (AI) models to data collected for diseases prediction. Moreover, the study can be extended to add camera for live view which will add some value and confidence. In addition, incorporating sensors for Luminosity, and sensor for water and food refilling would add value to this study. Lastly, developing a mobile application with additional features such as a reminder scheduler for vaccination can also significantly contribute to the current work.

REFERENCES

Abdul-qawy, A. S., Pramod, P. J., Magesh, E., & Srinivasulu, T. (2015). *The Internet of Things (IoT): An Overview*. *International Journal of Engineering Research and Applications*, *5*(12), 71–82.

Abdulgadir, M. J., Hoshi, T., Hirayama, K., & Kaneko, S. (2014). *Digitized data from rural Africa: A field application of open data kit (ODK) to collect baseline health data in eastern Sudan, New Halfa locality*. https://www.google.com

Abraham, S., Beard, J., & Manijacob, R. (2017). *Remote environmental monitoring using Internet of Things (IoT). GHTC 2017 - IEEE Global Humanitarian Technology Conference, Proceedings*. https://doi.org/10.1109/GHTC.2017.8239335

Ahmed, N., De, D., & Hussain, I. (2018). Internet of Things (IoT) for Smart Precision Agriculture and Farming in Rural Areas. *IEEE Internet of Things Journal*, 5, 4890-48-99. https://doi.org/10.1109/JIOT.2018.2879579

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *Communications Surveys and Tutorials*, 17(4), 2347-2376, https:// doi. org/10. 1109/ COMST. 2015.2444095

Al-Zewairi, M., Biltawi, M., Etaiwi, W., & Shaout, A. (2017). Agile Software Development Methodologies: Survey of Surveys. *Journal of Computer and Communications*, 5, 74-97. https://doi.org/10.4236/jcc.2017.55007

Arifin, S. R. M. (2018). Ethical considerations in qualitative study. *International Journal of Care Scholars*, 1(2), 30–33.

Astill, J., Dara, R. A., Fraser, E. D. G., Roberts, B., & Sharif, S. (2020). Smart poultry management: Smart sensors, big data, and the internet of things. *Computers and Electronics in Agriculture, 170*(February), 1-9. https:// doi. org/10. 1016/j. compag. 2020. 105291

Baronti, P., Pillai, P., Chook, V. W. C., Chessa, S., Gotta, A., & Hu, Y. F. (2007). *Wireless sensor networks : A survey on the state of the art and the 802 . 15 . 4 and ZigBee standards*. *Computer Communications*, *30*, 1655–1695. https:// doi. org/10. 1016/j.

comcom. 2006.12.020

Camps-Mur, D., Garcia-Saavedra, A., & Serrano, P. (2013). Device-to-device communications with WiFi direct: Overview and experimentation. *Wireless Communications*, 20(3), 97-104. https://doi.org/10.1109/MWC.2013.6549288

Chandra, V. (2015). Comparison between Various Software Development Methodologies. *International Journal of Computer Applications*, 131(9), 7-10 https:// doi. org/10. 5120/ijca2015907294

Chin, J., Callaghan, V., & Ben Allouch, S. (2019). The Internet-of-Things: Reflections on the past, present and future from a user-centered and smart environment perspective. *Journal of Ambient Intelligence and Smart Environments*, 11(1), 45-69. https:// doi. org/10. 3233/AIS-180506

Circuit Basics. (2017). *Basics of UART Communication*. *Electronics Hub*. https://www.google.com

Crawford, T., & Hussain, T. (2017). A Comparison of Server Side Scripting Technologies. *Proceedings of the 2017 International Conference on Software Engineering Research and Practice*. https://www.google.com

Crespo, G. (2013). *Responsive Web Design with jQuery: In Web Design*. https://www.google.com

Daghir, N. J. (2008). *Nutrient requirements of poultry at high temperature: Poultry Production in Hot Climate*. https://www.google.com

Dhingra, S., Madda, R. B., Gandomi, A. H., Patan, R., & Daneshmand, M. (2019). Internet of things mobile-air pollution monitoring system (IoT-Mobair). *Internet of Things Journal*, 6(3), 5577–5584. https://doi.org/10.1109/JIOT.2019.2903821

Dudovskiy, J. (2019). *Inductive Approach (Inductive Reasoning): In Research-Methodology.Net*. https://research-methodology.ne

Edmond, R. (2018). *Poultry Subsector in Tanzania: A Quick Scan Study commissioned by Embassy of the Kingdom of the Netherlands in Dar es Salaam, Tanzania*. https://www.google.com

FAO SHFS. (2015). *White Meat Value Chain Analysis in Southern Tanzania*. https://www.google.com

Farooq, M. S., Riaz, S., Abid, A., Umer, T., & Zikria, Y. Bin. (2020). Role of iot technology in agriculture: A systematic literature review. *Electronics (Switzerland), 9*(2), 1-42. https://doi.org/10.3390/electronics9020319

Gebregeziabhear, E., & Ameha, N. (2015). The effect of stress on productivity of animals: A review. *Journal of Biology, Agriculture and Healthcare,* 5(3), 1-9.

Georgiou, O., & Raza, U. (2017). Low Power Wide Area Network Analysis: Can LoRa Scale? *Wireless Communications Letters,* 6(2), 162-165. https:// doi. org/10. 1109/LWC. 2016.2647247

Goromela, E. H., Kwakkel, R. P., Verstegen, M. W. A., & Katule, A. M. (2007). Identification, characterisation and composition of scavengeable feed resources for rural poultry production in Central Tanzania. *African Journal of Agricultural Research,* 2*2*(August), 380–393.

Griffin, P. H. (2017). *Secure authentication on the Internet of Things. Conference Proceedings.* https://doi.org/10.1109/SECON.2017.7925274

Han, D. M., & Lim, J. H. (2010). Design and implementation of smart home energy management systems based on ZigBee. *Transactions on Consumer Electronics*, 56(3), 1417-1425. https://doi.org/10.1109/TCE.2010.5606278

Handigolkar, L. S., Kavya, M. L., & Veena, P. D. (2018). Iot Based Smart Poultry Farming using Commodity Hardware and Software. *Bonfring International Journal of Software Engineering and Soft Computing*, 6, 171-175. https://doi.org/10.9756/BIJSESC.8269

Heath, S. (2002). *What is an embedded system? In Embedded Systems Design*. https://www.google.com

Hitimana, E., Bajpai, G., Musabe, R., & Sibomana, L. (2018). Remote Monitoring and Control of Poultry Farm using IoT Techniques. *International Journal of Latest Technology in Engineering, Management & Applied Science, VII*(V), 87–90.

Holik, V. (2015). *Management of laying hens under tropical conditions begins during the rearing period*. Lohmann Information. https://www.google.com

Ikpehai, A., Adebisi, B., Rabie, K. M., Anoh, K., Ande, R. E., Hammoudeh, M., Gacanin, H., & Mbanaso, U. M. (2019). Low-power wide area network technologies for internet-of-things: A comparative review. *Internet of Things Journal*, 6(2), 2225-2240. https://doi.org/10.1109/JIOT.2018.2883728

Islam, M., Tonmoy, S. S., Quayum, S., Sarker, A. R., Hani, S. U., & Mannan, M. A. (2019a). *Smart Poultry Farm Incorporating GSM and IoT. 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (pp. 277–280).* https://www.google.com

Khan, J. Y., & Chen, D. (2019). Low-Power Wide-Area Networks. In J. Y. Khan, & M. R. Yuce (Eds), *Internet of Things (IoT)*. https://doi.org/10.1201/9780429399084-4

Kocaman, B., Esenbuga, N., Yildiz, A., Laçin, E., & Macit, M. (2006). Effect of environmental conditions in poultry houses on the performance of laying hens. *International Journal of Poultry Science*, *5*(1), 26–30. https:// doi. org/10. 3923/ijps. 2006.26.30

Kumar, K., Zindani, D., & Davim, J. P. (2020). *Introduction to Design Thinking. In SpringerBriefs in Applied Sciences and Technology*. https://doi.org/10.1007/978-3-030-31359-3_1

Landaluce, H., Arjona, L., Perallos, A., Falcone, F., Angulo, I., & Muralter, F. (2020). A review of iot sensing applications and challenges using RFID and wireless sensor networks. *Sensors*, *20*(9), 1-18. https://doi.org/10.3390/s20092495

Leens, F. (2009). An introduction to I2C and SPI protocols. *Instrumentation and Measurement Magazine*, 12(1), 8-13. https://doi.org/10.1109/MIM.2009.4762946

Letkowski, J. (2014). Doing database design with MySQL. *Journal of Technology Research*, 6, 1-5.

Li, L., Hu, X., Chen, K., & He, K. (2011). *The applications of WiFi-based Wireless Sensor Network in Internet of Things and Smart Grid. Proceedings of the 2011 6th Conference on Industrial Electronics and Applications, ICIEA 2011*. https://doi.org/10.1109/ICIEA.2011.5975693

Li, Z., Wang, J., Higgs, R., Zhou, L., & Yuan, W. (2017). *Design of an Intelligent*

Management System for Agricultural Greenhouses Based on the Internet of Things. Proceedings - 2017 IEEE International Conference on Computational Science and Engineering and IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, CSE and EUC 2017. https://doi.org/10.1109/CSE-EUC.2017.212

Mahale, R. B., & Sonavane, S. S. (2016). Smart Poultry Farm Monitoring Using IOT and Wireless Sensor Networks. *International Journal of Advanced Research in Computer Science*, 7(3), 187–190.

Manshor, N., Rizaan, A., Rahiman, A., & Yazed, M. K. (2019). *IoT Based Poultry House Monitoring. 2019 2nd International Conference on Communication Engineering and Technology (pp.72–75)*. https://www.google.com

Matata, J. B. (2001). *Farming systems approach to technology development and transfer*. https://www.google.com

Mathew, G., & Thomas, S. (2013). A Novel Multifactor Authentication System Ensuring Usability and Security. *International Journal of Security, Privacy and Trust Management*, 2(5), 1-10. https://doi.org/10.5121/ijsptm.2013.2503

Menčík, J. (2016). *Reliability of Systems. In Concise Reliability for Engineers*. InTech. https://doi.org/10.5772/62358

Miorandi, D., Sicari, S., Pellegrini, F. De, & Chlamtac, I. (2012). Ad Hoc Networks Internet of things : Vision , applications and research challenges. *Ad Hoc Networks*, *10*(7), 1497–1516. https://doi.org/10.1016/j.adhoc.2012.02.016

Mkunda, J. J., Lassen, J., Chachage, B., Kusiluka, L. J. M., & Pasape, L. (2020). Analysis of marketing efficiency of processed sardine products of Lake Victoria: Case of Tanzania. *African Journal of Science, Technology, Innovation and Development*, (12)1, 85-96. https://doi.org/10.1080/20421338.2019.1609783

Msami, H. M. (2000). *Studies on the Structure and Problems of Family Poultry Production in Tanzania with Suggestions on Improvements and Interventions. 2nd Research Coordination Meeting of the Co-Coordinated Research Programme on" Improvement of Health and Management of Family Poultry Production in Africa" from*. https://www.google.com

Najafi, P., Zulkifli, I., Amat Jajuli, N., Farjam, A. S., Ramiah, S. K., Amir, A. A., O'Reily, E., & Eckersall, D. (2015). Environmental temperature and stocking density effects on acute phase proteins, heat shock protein 70, circulating corticosterone and performance in broiler chickens. *International Journal of Biometeorology*, *59*(11), 1577–1583. https://doi.org/10.1007/s00484-015-0964-3

Nakhuva, B. N., & Champaneria, T. A. (2017). *Security provisioning for RESTful web services in Internet of Things. 2017 4th International Conference on Advanced Computing and Communication Systems, ICACCS 2017*. https:// doi. org/10. 1109/ICACCS.2017.8014642

NBS. (2017). *National Bureaural of Statistics*. https://www.google.com

NodeMCU ESP8266 Pinout. (n.d.). *Specifications, Features & Datasheet.* Retrieved January 25, 2021, from https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet

Oloyo, A. (2018). The use of housing system in the management of heat stress in poultry production in hot and humid climate: A review. *Poultry Science Journal*, *6*(1), 1–9. https://doi.org/10.22069/psj.2018.13880.1284

Otto, M., & Thornton, J. (2015). *Bootstrap· The world's most popular mobile-first and responsive front-end framework.* Getbootstrap.Com. https://www.google.com

Paputungan, I. V, Faruq, A. A., Puspasari, F., Hakim, F. A., Fahrurrozi, I., Oktiawati, U. Y., & Mutakhiroh, I. (2020). *Temperature and Humidity Monitoring System in Broiler Poultry Farm. Conference Series: Materials Science and Engineering*. https://doi.org/10.1088/1757-899x/803/1/012010

Pereira, W. F., Fonseca, L. Da S., Putti, F. F., Góes, B. C., & Naves, L. De P. (2020). Environmental monitoring in a poultry farm using an instrument developed with the internet of things concept. *Computers and Electronics in Agriculture*, *170*(January), 1-7. https://doi.org/10.1016/j.compag.2020.105257

Phiri, H., Kunda, D., & Phiri, J. (2018). An IoT Smart Broiler Farming Model for Low Income Farmers. *International Journal of Recent Contributions from Engineering, Science and IT, 6*(3), 95–110.

Prayogo, S. S., Mukhlis, Y., & Yakti, B. K. (2019). *The Use and Performance of MQTT and CoAP as Internet of Things Application Protocol using NodeMCU ESP8266. Proceedings of 2019 4th International Conference on Informatics and Computing, ICIC 2019.* https://doi.org/10.1109/ICIC47613.2019.8985850

Ruiz-Garcia, L., Lunadei, L., Barreiro, P., & Robla, J. I. (2009). A review of wireless sensor technologies and applications in agriculture and food industry: State of the art and current trends. *Sensors*, 9(6), 4728–4750. https://doi.org/10.3390/s90604728

Sohraby, K., Minoli, D., & Znati, T. (2006). *Wireless Sensor Networks: Technology, Protocols, and Applications.* https://www.wiley.com

Srivastava, A., Bhardwaj, S., & Saraswat, S. (2017). *SCRUM model for agile methodology. Proceeding - International Conference on Computing, Communication and Automation, ICCCA 2017.* https://doi.org/10.1109/CCAA.2017.8229928

Talavera, J. M., Tobón, L. E., Gómez, J. A., Culman, M. A., Aranda, J. M., Parra, D. T., Quiroz, L. A., Hoyos, A., & Garreta, L. E. (2017). Review of IoT applications in agro-industrial and environmental fields. *Computers and Electronics in Agriculture*, 142(Part A), 283-297. https://doi.org/10.1016/j.compag.2017.09.015

TCRA. (n.d.). *Tanzania Communications Regulatory Authority.* Retrieved December 8, 2020, from https://www.tcra.go.tz/publication-and-statistics/statistics

TCRA. (2019). *Tanzania Communication and Regulatory*. https:// www. tcra. go. tz/images/TelCom_ Statistics_June_2019.xlsx.pdf on 17th, September,2019 at 15:00hrs

TLMI. (2015). *Tanzania Livestock Modernization Initiative, Ministry of Livestock and Fisheries Development: United Republic of Tanzania*. https://www.google.com

TLMP. (2018). *Tanzania livestock master plan*. https://www.google.com

Vallon, R., Da Silva Estácio, B. J., Prikladnicki, R., & Grechenig, T. (2018). Systematic literature review on agile practices in global software development. *Procedia Computer Science*, 138, 775-782. https://doi.org/10.1016/j.infsof.2017.12.004

Wang, N., Zhang, N., & Wang, M. (2006). Wireless sensors in agriculture and food industry - Recent development and future perspective. *Computers and Electronics in Agriculture*, 50, 1-14. https://doi.org/10.1016/j.compag.2005.09.003

Yi, P., Iwayemi, A., & Zhou, C. (2011). Developing ZigBee deployment guideline under WiFi interference for smart grid applications. *Transactions on Smart Grid*. 2(1),110-120. https://doi.org/10.1109/TSG.2010.2091655

Zhao, Y., Aarnink, A. J. A., Dijkman, R., Fabri, T., de Jong, M. C. M., & Groot Koerkamp, P. W. G. (2012). Effects of temperature, relative humidity, absolute humidity, and evaporation potential on survival of airborne Gumboro vaccine virus. *Applied and Environmental Microbiology*,78(4), 1048-54. https://doi.org/10.1128/AEM.06477-11

APPENDICES

Appendix 1: Data collection Questionnaire

A questionnaire was prepared using Microsoft Excel and Open Data Kit Collector software. The excel document has three sheets named Survey, Choices, and Settings sheet as shown in Table 8, 9, and 10 respectively. The excel file was converted to XML file using an online XLS Form converter. The researcher stored the xml file in google drive and downloaded remotely for ODK collector to build the electronic form.

Table 10: Sample content of Survey sheet

type	**name**	**label**
begin group	page1	Page 1
note	introduction	I am Beston Lufyagila, pursuing masters in Information Communication Science and Engineering (ICSE) at Nelson Mandela African Institution of Science and Technology. I humbly request you to fill in this questionnaire that aims at studying the needy and requirements for developing an Internet of Things based system for environmental conditions monitoring in poultry house, a case of Tanzania. The system will improve poultry productivity at affordable price and in a user-friendly manner. I assure you that all information shall be strictly used for research purposes only, and kept confidential and anonymous.
note	sectionA	**Section A: Background Information**
date	q_1	1. Date of survey
select_one gender	q_2	2. Gender of respondent
select_one edu_level	q_3	3. Education level of respondent
select_one age_range	q_4	4. Age of respondent
end group		
begin group	page2	Page 2
note	sectionB	**Section B: Poultry Information**
select_one chicken_number	q_6	6. How many chicken do you have?
select_multiple chicken_type	q_7	7. What breed of chicken do you keep?
text	q_7_other	Specify
integer	q_8	8. What is the size of poultry house in square meter?
end group		
begin group	page3	Page 3
note	sectionC	**Section C: Poultry house conditions monitoring methods**
select_one cleanliness_frequency	q_11	11. How often do you clean your poultry house?
text	q_11_other	Specify
select_multiple temp_monitoring_method	q_14	14. How do you monitor and control the temperature?

text	q_14_other	Specify
select_multiple humidity_monitoring_method	q_18	18. How do you monitor and control humidity of the poultry house?
text	q_18_other	Specify
select_multiple ammonia_monitoring_method	q_22	22. How do you monitor and control ammonia in the poultry house?
text	q_22_other	Specify
select_multiple poultry_security	q_24	24. How do you ensure security of poultry against intruders and thieves?
end group		
begin group	page4	Page 4
note	sectionD	**Section D: Awareness of existing remote monitoring systems**
select_one systems_awareness	q_25	25. Are you aware of remote monitoring systems for poultry house conditions?
	q_26	26. Do you use in poultry house?
select_multiple unusage_reasons	q_27	27. Which factors affect adoption of smart poultry monitoring systems in Tanzania?
text	q_27_other	Specify
select_one yes_no	q_28	28. Based on the prototypes and mockups, do you need the proposed system?
text	q_29	What other features do you suggest the proposed system should have?
end group		
begin group	page5	Page 5
note	sectionE	**Section E: ICT tools awareness**
integer	q_31	31. Phone number
select_multiple phone_type	q_32	32. What type of mobile phone do you own?
select_multiple	q_33	33. How often do you access internet in your smartphone?
image	q_34	34. Take photo of the farm
geopoint	q_35	35. Capture the longitude and latitude of the location
end group		

Table 11: Choices sheet content

list_name	name	label
age_range	below_20	Below 20 years old
age_range	20-29	20 - 29 years old
age_range	30-39	30 - 39 years old
age_range	40-49	40 - 49 years old
age_range	50-59	50 - 59 years old
age_range	60+	60 and above
gender	male	Male
gender	female	Female
edu_level	primary	Primary
edu_level	form_4	Form IV
edu_level	form_6	Form VI
edu_level	university	University
edu_level	informal	Informal
edu_level	certificate	Certificate
edu_level	diploma	Diploma
chicken_number	between_2_20	Between 2 and 20
chicken_number	between_20_50	Between 20 and 50
chicken_number	between_50_100	Between 50 and 100
chicken_number	between_100_500	Between 100 and 500
chicken_number	between_500_1000	Between 500 and 1000
chicken_number	above_1000	Above 1000
chicken_type	layers	Layers
chicken_type	chotara	Cross breed (Chotara)
chicken_type	kienyeji	Local breed (Kienyeji)
chicken_type	broiler	Broiler
chicken_type	other	Others
cleanliness_frequency	daily	Daily
cleanliness_frequency	2_perweek	Twice per week
	weekly	Weekly
cleanliness_frequency	3_perweek	Thrice per week
cleanliness_frequency	other	Others
temp_monitoring_method	common_sense	Common sense
temp_monitoring_method	themometer	Themometer
temp_monitoring_method	chacoal	Charcoal stove
temp_monitoring_method	kerosine_lamp	Kerosine lamp
temp_monitoring_method	electricity	Electricity heater and fans
temp_monitoring_method	open_windows	Open/Close windows
temp_monitoring_method	other	Others
humidity_monitoring_method	common_sense	Common sense
humidity_monitoring_method	hygrometer	Hygrometer
humidity_monitoring_method	cleaning	cleaning

humidity_monitoring_method	open_windows	Opening of windows
humidity_monitoring_method	other	Others
ammonia_monitoring_method	common_sense	Common sense
ammonia_monitoring_method	cleaning	Cleaning
ammonia_monitoring_method	other	Others
poultry_security	dogs	Dogs
poultry_security	security_guard	Security guard
poultry_security	gates	Gates
poultry_security	patrol	Making patrol
poultry_security	other	Others
systems_awareness	yes	Yes
systems_awareness	no	No
unusage_reasons	costiful	Costiful
unusage_reasons	lack_of_features	Lack of features
unusage_reasons	unavilable_locally	Unavailable locally
unusage_reasons	not_user_friendly	Not user friendly
unusage_reasons	knowledge_shortage	Lack of knowledge on the system
unusage_reasons	other	others
phone_ownership	yes	Yes
phone_ownership	no	No
phone_type	basic_phone	Basic phone
phone_type	feature_phone	Feature phone
phone_type	smartphone	Smart phone

Table 12: Settings sheet content

form_title	**form_id**	**submission_url**
Poultry_Farmer_1	Poultry_Farmer_1	https://docs.google.com/spreadsheets/d/12_uwncVQrgbCmcGYIItB2WM--C4KBxMlplI9nBbGNPA/edit?usp=sharing

Appendix 2: Sample Python source codes for data analysis

```
#############################################
#By: Beston Lufyagila            ###
#Place:  NM-AIST, Arusha, Tanzania ###
# Date: 22/07/2020 Time: 2200hrs EAT    ###
#############################################

#import all the libraries
%matplotlib inline
#library for plotting graphs
import matplotlib.pyplot as plt
#plt.rcParams["font.family"] = "Times New Roman"
#library for reading data
import numpy as np
#library for reading data and creating data frames
import seaborn as sns
import pandas as pd
#import the regular expression for use in removing trailing spaces
import re
import csv
#color codes to be used
beston_colors = ['#3274a1', '#e1812c','#3a923a','#c03d3e','#5dbcd2','#9372b2']
#A function to be used through out to plot a pieChart
def plot_pieChart(values, labels, title_str, explode, fsize, startangle):
    if startangle == '':
        startangle = 140
    else:
        startangle = startangle
    plt.pie(values, explode=explode, labels=labels, colors=beston_colors,
    autopct='%1.1f%%', shadow=True, startangle=startangle,textprops={'fontsize': 12})
   # plt.title(title_str, fontsize=fsize)
    plt.axis('equal')
# A function to be used to draw the bar graph
def plot_barChart(y_pos, yvalues, xvalues, title_str,x_label, y_label):
    rects1 = plt.bar(y_pos, yvalues, align='center',color=beston_colors)
    plt.xticks(y_pos, xvalues,rotation=45,
    horizontalalignment='right',
    fontweight='light',
    fontsize='x-large')
    plt.title(title_str, fontsize= 16)
    plt.xlabel(x_label,fontsize=16)
    plt.ylabel(y_label,fontsize=16)
# used to draw the bar graph with percentages on each bar
def plot_barChart_2(y_pos, yvalues, xvalues, title_str,x_label, y_label):
    rects1 = plt.bar(y_pos, yvalues, align='center',color=beston_colors)
    plt.xticks(y_pos, xvalues,rotation=45,
    horizontalalignment='right',
    fontweight='light',
    fontsize='x-large')
    for rect in rects1:
        height = rect.get_height()
```

```
        percent = (float(100) * float(height))/float(120)
        print("the percentage is {}".format(percent))
        ax.text(rect.get_x() + rect.get_width()/2., 0.99*height,
           '%1.1f' % percent + "%", ha='center', va='top')
    plt.title(title_str, fontsize= 16)
    plt.xlabel(x_label,fontsize=16)
    plt.ylabel(y_label,fontsize=16)
# A function to save and display figure
def save_displayFigure(file_name):
    str_path = '../figures_120/'+ file_name + '.png'
    plt.savefig(str_path)
#Load Data set with pandas
df = pd.read_csv('../data/poultry_farmer_120.csv', delimiter=',')
df.shape[0]
#Analysis of Poultry farmers gender
Me = 0
Fe = 0
total= df.shape[0]
for index, row in df.head(120).iterrows():
    #check if sex is male
    if row['gender'] == 'male':
        Me += 1
    else:
        Fe += 1
print("The number of males are {}".format(Me))
print("The number of females are {}".format(Fe))
print("The number of respondents are {}".format(total))
#Configuring graph
gender_labels = ['Male', 'Female']
values = [Me,Fe]
explode = (0.05, 0.05)  # explode 1st slice

#Plot a pie chart graph
f, ax = plt.subplots(figsize=(6,6)) # set the size that you'd like (width, height)
plot_pieChart(values,gender_labels,'Gender',explode, 16,75)
save_displayFigure('gender')

#analysis of methods used by poultry farmers to monitor/regulate temperature in farm
arr = [] #initialize an array to hold the values
for index, rzn in df.head(120).iterrows():

#add values in the array
   arr.append(rzn['temp_monitoring_method'])
new_val = ','.join(arr) #convert array to string

#get the values for each category
common_sense = new_val.count("common_sense")
open_windows= new_val.count("open_windows")
chacoal = new_val.count("chacoal")
kerosine_lamp = new_val.count("kerosine_lamp")
thermometer = new_val.count("thermometer")
electric_bulb = new_val.count("electric_bulb")
```

```
#display the results
print("Summary of the results:")
print("charcoal : {}".format(chacoal))
print("open_windows : {}".format(open_windows))
print("kerosine_lamp : {}".format(kerosine_lamp))
print("electric_bulb : {}".format(electric_bulb))
#plot a graph to show this

x_lab = ('Charcoal stove','Open/Closing curtains','Kerosine lamp','Electricity bulb') #get the
labels
y_pos = np.arange(len(x_lab))
values = [chacoal,open_windows,kerosine_lamp,electric_bulb]

explode = (0.04, 0.04, 0.04, 0.04)  # explode 1st slice
# Plot
f, ax = plt.subplots(figsize=(10,10)) # set the size that you'd like (width, height)

#plot_barChart(y_pos,values,x_lab,'Temperature Regulation Methods', 'Regulation Method',
'No of Farmers')
plot_barChart_2(y_pos,values,x_lab,' ', ' ', 'No of Farmers')
save_displayFigure('Temperature_regulation_method')
```

Appendix 3: Sample source codes for NodeMCU ESP8266 microcontroller

```
/*
 * Programmer: Beston Lufyagila
   Place: NM-AIST, COCSE LAB
   Date: 22/09/2020
   Phone: +255782616341
 */
#include <ArduinoJson.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include "Arduino.h"
#include <DHT12.h>
#include "MQ135.h"
const int ANALOGPIN = 0;
MQ135 gasSensor = MQ135(ANALOGPIN);
// Set dht12 i2c communication on default Wire pin
DHT12 dht12;

/////////////////////////////////////////////////////

String Node_KEY = "z1mMEtH"; // Node Key is a unique 7 Digits Node identifier
const char* ssid = "root"; //Wi-Fi Name here
const char* password = "root_root"; //Wi-Fi passwd here
//Gateway Local IP adress here
const char* serverName = "http://192.168.1.113/28736api.php";

/////////////////////////////////////////////////////
// the following variables are unsigned longs because the time, measured in
// milliseconds, will quickly become a bigger number than can be stored in an int.
unsigned long lastTime = 0;
unsigned long timerDelay = 1000;
unsigned long timer_d;
//sensor Variables
float hif12;
float hic12;
float dpf12;
float dpc12;

//pin varibales
const int fan = 16;
const int heater = 12;
//status LED
```

```
//const int wifi_led = 2 ;  // Orange LED; S3 changed to exhaust fan
const int exhaust_fan = 2;
const int device_status = 0; //Red LED  S2
//http response

//threshold values (Thresshold is to be cahnged here)
float T_min = 22.00;
float T_max = 26.00;
float H_min = 75.00;
float H_max = 30.00;
float A_min = 30.00;
float A_max = 70.00;
//mode
String Mode = "A";
String fan_s = "D";
String heater_s = "D";
String E_fan_s = "D";
//amonia
float amonia;


void setup() {
  Serial.begin(115200);
  //setup pin here
  pinMode(16,OUTPUT);
  pinMode(12, OUTPUT);
  //device status LED's
  pinMode(exhaust_fan, OUTPUT);
  pinMode(device_status, OUTPUT);

  //tun OFF all status pin
  digitalWrite(device_status, LOW);


  //turn all pin LOW
  digitalWrite(fan, LOW);
  digitalWrite(heater, LOW);
  digitalWrite(exhaust_fan, LOW);

  // Start sensor
  dht12.begin();

  WiFi.begin(ssid, password);
  WiFi.setSleepMode(WIFI_NONE_SLEEP);
  WiFi.mode(WIFI_STA);
```

```
  Serial.println("Connecting");
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");

  }
  Serial.println("Connected");

}

void loop() {
  //read amonia
  amonia = gasSensor.getPPM();
  //Send an HTTP POST request every timerdelay
  if ((millis() - lastTime) > timerDelay) {
    //read data
    // Read temperature as Celsius (the default)
    float t12 = dht12.readTemperature();
    // Read temperature as Fahrenheit (isFahrenheit = true)
    float f12 = dht12.readTemperature(true);
     //read humidity
    float h12 = dht12.readHumidity();

    bool dht12Read = true;
    // Check if any reads failed and exit early (to try again).

    if (isnan(h12) || isnan(t12) || isnan(f12)) {
      Serial.println("Failed to read from DHT12 sensor!");

      dht12Read = false;
    }

    if (dht12Read){
      // Compute heat index in Fahrenheit (the default)
      hif12 = dht12.computeHeatIndex(f12, h12);
      // Compute heat index in Celsius (isFahreheit = false)
      hic12 = dht12.computeHeatIndex(t12, h12, false);
      // Compute dew point in Fahrenheit (the default)
      dpf12 = dht12.dewPoint(f12, h12);
      // Compute dew point in Celsius (isFahreheit = false)
      dpc12 = dht12.dewPoint(t12, h12, false);
    }
    //Check WiFi connection status
    if(WiFi.status()== WL_CONNECTED){
      HTTPClient http;
```

```
    // Your Domain name with URL path or IP address with path
    http.begin(serverName);

    // Specify content-type header
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");
    //Converting Float to String here
    String hif = String(hif12);
    String hic = String(hic12);
    String dpf = String(dpf12);
    String dpc = String (dpc12);
    String temp = String (t12);
    String hum = String (h12);
    String Amonia = String(amonia);
    // Data to send with HTTP POST
    String httpRequestData =
"api_KEY="+Node_KEY+"&hum="+hum+"&temp="+temp+"&hif="+hif+"&hic="+hic+"&
dpf="+dpf+"&dpc="+dpc+"&amonia="+Amonia+"";

    // Send HTTP POST request
    int httpResponseCode = http.POST(httpRequestData);
    const String& payload = http.getString(); //http response
    //Serial.println(httpResponseCode);
    //Serial.print(payload);
    String code = String(httpResponseCode);
    //String httpResponse = code+","+ payload;
    String httpResponse = payload;
    Serial.println(httpResponse); // Uncomment for Debuging purposly
    //Serial.println(httpRequestData);
    //processing response here
    //response format (httpResponseCode, device_status, fan_status, heater_status)
    //trim the white space from a string
    //httpResponse = httpResponse.trim();
    int string_len = httpResponse.length();
    //deserialize JSON data
    char json[] = "httpResponse";
    // Deserialize the JSON document
    const size_t capacity = JSON_OBJECT_SIZE(3) + JSON_ARRAY_SIZE(2) + 400;
    DynamicJsonDocument doc(capacity);
    DeserializationError error = deserializeJson(doc, httpResponse);

    // Test if parsing succeeds.
    if (error) {
      Serial.print(F("deserializeJson() failed: "));
      Serial.println(error.c_str());
```

```
    return;
  }
  /*
    a = fan_status
    b = heater_status
    c = exhaust_fan_status
    d = min_tempreture
    e = max_tempreture
    f = max_humidity
    g = min_humidity
    h = min_amonia
    i = max_amonia
    j = data_interval
    k = operation_mode
    l = device_status

*/

  // Extract values
  //Serial.println(F("Response:"));
  //Serial.println(doc["a"].as<char*>());
  //Serial.println(doc["b"].as<char*>());
 //Serial.println(doc["c"].as<char*>());
 //Serial.println(doc["d"].as<char*>());
   //The right way
//const char* sensor = root["sensor"];
//const char* sensor = root["sensor"].as<char*>();
  //conditions
     //Get Temp
    T_min = doc["d"].as<float>();
    T_max = doc["e"].as<float>();
     //Get Hum
    H_min = doc["g"].as<float>();
    H_max = doc["f"].as<float>();
     //Get Amonia
    A_min = doc["h"].as<float>();
    A_max = doc["i"].as<float>();

    //data interval
    //timer_d = doc["j"].as<int>();
    //timerDelay = timer_d * 1000;

    //Auxilliary operation
    Mode = doc["k"].as<char*>();
    fan_s = doc["a"].as<char*>();
```

```
    heater_s = doc["b"].as<char*>();
    E_fan_s = doc["c"].as<char*>();

   //conditions here
   //MAX CONDITION
   if (Mode == "A"){
     if (t12 >= T_max ){
       //if thempreture is greater than threshold
       digitalWrite(heater, LOW); //tur off heter
       digitalWrite(fan, HIGH); //keep circulating air
       digitalWrite(exhaust_fan, HIGH); //turn on exhaust fan to remove excess tempreture
out
       if (amonia > A_max){
         digitalWrite(exhaust_fan, HIGH); //Turn on exhaust fan to remove excess amonia
       }else{
         digitalWrite(exhaust_fan, LOW); //Turn off exhaust fan to preserve internal
tempreture for a long time
       }
     }
     //NORMAL CONDITION
     if (t12 >= T_min && t12 <= T_max ){
       digitalWrite(heater, HIGH); // turn off heater as tempreture is normal
       digitalWrite(fan,HIGH); //turn on inside fan for Air circulation
       //also check amonia level
       if (amonia >= A_max){
         digitalWrite(exhaust_fan, HIGH); //Turn on exhaust fan to remove excess amonia
       }else{
         digitalWrite(exhaust_fan, LOW); //Turn off exhaust fan to preserve internal
tempreture for a long time
       }


     }
     //MIN CONDITION
     if (t12 < T_min ){
       digitalWrite(heater, HIGH); // turn off heater as tempreture is normal
       digitalWrite(fan,HIGH); //turn on inside fan for Air circulation
       if (amonia >= A_max){
         digitalWrite(exhaust_fan, HIGH); //Turn on exhaust fan to remove excess amonia
       }else{
         digitalWrite(exhaust_fan, LOW); //Turn off exhaust fan to preserve internal
tempreture for a long time
       }
     }
     //HUMIDITY
   } //Auto Mode ends here
```

```
     //Manual Mode
     if (Mode == "M"){
      //fan
      if (fan_s == "D"){
        digitalWrite(fan, LOW);
      }else if(fan_s == "E"){
        digitalWrite(fan, HIGH);
      }
      //heater
      if (heater_s == "D"){
        digitalWrite(heater, LOW);
      }else if(heater_s == "E"){
        digitalWrite(heater, HIGH);
      }
      // Exhaust fan
      if (E_fan_s == "D"){
        digitalWrite(exhaust_fan, LOW);
      }else if(E_fan_s == "E"){
        digitalWrite(exhaust_fan, HIGH);
      }
     }
    // Free resources
    http.end();
   }
   else {
    Serial.println("WiFi Disconnected");
   }
   lastTime = millis();
  }
}
```

Appendix 4: Sample JavaScript source code for the Synchronization Algorithm

```
var sync_KEY; // syncronization key here
// LIBRARIES
var mysql      = require('mysql');
const request = require('request');
var connection = mysql.createConnection({
  host     : 'localhost',
  user     : 'root',
  password : 'root',
  database : 'chicken_coop'
});

//variables
var id;
var temp;
var hum;
var amonia ;
var date;
var node_KEY;
var node_id;
var coop_id;
var hif;
var hic;
var dpf;
var dpc;

connection.connect();
// Infinite loop here
function logEvery2Seconds() {
  //cycle for every one second (2000 ms)
  setTimeout(() => {
    //query for server link
    connection.query('SELECT * FROM storage_option WHERE id = 1', function (error,
results, fields) {
      if (error) throw error; //checeks for error
      //option
      var mode = results[0].option_v
      //check if it is online mode or offline mode
      if (mode == 'online'){
        //obtain a sync API KEY
        connection.query('SELECT * FROM sync WHERE id = 1', function (error, syncData,
fields) {
          if (error) throw error;
          //syncronization key
          sync_KEY = syncData[0].API_KEY;
        });
        //online mode proceed on syncronization
        var online_server = results[0].onlineserver //online server adress
        //query for node data
        connection.query('SELECT * FROM node_data WHERE sync = "N"', function (error,
nodeData, fields) {
```

```
//check for error
if (error){
  console.log(error);
} else if(nodeData.length == 1){
  //Start Synchronization
    var id        = nodeData[0].id;
    var temp      = nodeData[0].temp;
    var hum       = nodeData[0].hum;
    var amonia    = nodeData[0].amonia;
    var date      = nodeData[0].date;
    var node_KEY  = nodeData[0].node_KEY;
    var node_id   = nodeData[0].node_id;
    var coop_id   = nodeData[0].coop_id;
    var hif       = nodeData[0].hif;
    var hic       = nodeData[0].hic;
    var dpf       = nodeData[0].dpf;
    var dpc       = nodeData[0].dpc;
    //prepare a post request for submitting data through API
    var server = online_server+'/sync.php';
      const options = {
        url: server,
        json: true,
        body: {
            // data to be sent
            id: id,
            temp: temp,
            hum: hum,
            amonia: amonia,
            date: date,
            node_KEY: node_KEY,
            node_id: node_id,
            coop_id: coop_id,
            hif: hif,
            hic: hic,
            dpf: dpf,
            dpc: dpc,
            sync_KEY: sync_KEY

        }
    };
    request.post(options, (err, res, body) => {
      if (err) {
          return console.log(err);
      }else{
        //success
        //update data value
        connection.query('UPDATE node_data SET sync = "Y" WHERE id = "'+id+'"
AND node_KEY = "'+node_KEY+'" AND node_id = "'+node_id+'"', function (error, results,
fields) {
          if (error){
            console.log(error);
          }else{
```

```
                //data updated
                console.log('Success Updated');
                //check date and time
                function dateManipulation(date, days, hrs, mins, operator) {
                  date = new Date(date);
                  if (operator == "-") {
                    var durationInMs = (((24 * days) * 60) + (hrs * 60) + mins) * 60000;
                    var newDate = new Date(date.getTime() - durationInMs);
                  } else {
                    var durationInMs = (((24 * days) * 60) + (hrs * 60) + mins) * 60000;
                    var newDate = new Date(date.getTime() + durationInMs);
                  }
                  return newDate;
                }
                var today = new Date();
                var storageDay = 2; //change here for no of day the data are allowed to stay
                var newDate = dateManipulation(today, storageDay, 0, 0, "-");
                var year = newDate.getFullYear();
                var day = newDate.getDay();
                //add zero on day
                if (day <10){
                  day = '0'+day;
                }
                var month = newDate.getMonth();
                if (month <10){
                  month = '0'+month;
                }
                var fullDate = year+'-'+month+'-'+day;
                //console.log(fullDate);
                connection.query('DELETE FROM node_data WHERE data_date =
'"+fullDate+"'', function (error, results, fields) {
                  if (error) throw error;
                  console.log('Affected Rows: ', results.length);
                });


              }

            });

          }
          //console.log(`Status: ${res.statusCode}`);
          //console.log(body);

        });
      }else if(nodeData.length == 0){
        //console.log('No Data');
        function dateManipulation_2(date, days, hrs, mins, operator) {
          date = new Date(date);
          if (operator == "-") {
            var durationInMs = (((24 * days) * 60) + (hrs * 60) + mins) * 60000;
            var newDate = new Date(date.getTime() - durationInMs);
```

```
            } else {
              var durationInMs = (((24 * days) * 60) + (hrs * 60) + mins) * 60000;
              var newDate = new Date(date.getTime() + durationInMs);
            }
            return newDate;
          }
          var today = new Date();
          var storageDay = 2; //change here for no of day the data are allowed to stay
          var newDate = dateManipulation_2(today, 2, 0, 0, "-");
          var year = newDate.getFullYear();
          var day = newDate.getDay();
          //add zero on day
          if (day <10){
            day = '0'+day;
          }
          var month = newDate.getMonth();
          if (month <10){
            month = '0'+month;
          }
          var fullDate = year+'-'+month+'-'+day;
          connection.query('DELETE FROM node_data WHERE data_date = "'+fullDate+'"',
function (error, results, fields) {
            if (error) throw error;
            console.log('Deleting data | '+fullDate+' | Data..');
          });
        }
      });

    }else if( mode == 'offline'){
      console.log('offline mode');
    }else{
      console.log(Error);
    }
  });
    logEvery2Seconds(); //call a function here
 }, 1000)
}
logEvery2Seconds(); //calling a function
//connection.end();
```

Appendix 5: Sample PHP source code for Web application development

```
<?php
//API file for receiving data from nodes
//2020

//include DB connection here
include_once"db_conn.php";
//check for storage option
$storage_id = 1;
$remort_server = '';
$select_storage = $conn->prepare("SELECT * FROM storage_option WHERE id = :id");
$select_storage->execute([
	"id" => $storage_id
]);
$storage_data = $select_storage->fetch(PDO::FETCH_ASSOC);

//include DB connection here
include_once"db_conn.php";
//check for storage option
$storage_id = 1;
$remort_server = '';
$select_storage = $conn->prepare("SELECT * FROM storage_option WHERE id = :id");
$select_storage->execute([
	"id" => $storage_id
]);
$storage_data = $select_storage->fetch(PDO::FETCH_ASSOC);
	//check for storage type
	if($storage_data['option_v'] == 'offline'){
		//include file for local handling
		 include_once '28736api_offline.php';

	}else if($storage_data['option_v'] == 'online'){
		//online server
		$remote_server = 'http://192.168.0.105/28736234325api.php';
		//include file for local handling
		 include_once '28736api_online.php';
	}else{
	//update error
		$info['state'] = false;
		$info['msg'] = "Error, API failed";
	}
?>
```

RESEARCH OUT PUTS

(i) Publication

(ii) Poster Presentation