Design and implement a Crop Management System for farmers in Chongwe District of Zambia.

Gregory Kakoma Chikanwa: ENG2100013

Supervisor: Dr. Simon Tembo

This research proposal is submitted in fulfilment of the academic requirements for the degree of Master of Engineering in ICT in the School of Engineering UNIVERSITY OF ZAMBIA, LUSAKA.

2024

As the candidate’s supervisor, I have approved this research for submission.

Sign……………………………………… Date ……………………. January 2024

Dr Simon Tembo
Lecturer (Supervisor)
University of Zambia (School of Engineering)

DECLARATION

I declare that this research is my work. Where collaboration with other people has taken place or material generated by other researchers is included, the parties and/ or materials are explicitly stated with references as appropriate.

This work is being submitted for the master's in information and communication technology at the University of Zambia. It has not been submitted to any other university for any other degree or examination.

Sig………………………………………….Date……………….January 2024

Name: Gregory Kakoma Chikanwa

Student ID: ENG21000136

CERTIFICATE OF APPROVAL

This dissertation by Gregory Kakoma Chikanwa is approved as fulfilling the partial requirements for the award of the degree in Master of Engineering in ICT at the University of Zambia.

………………………	……………………..	…………………
Internal Examiner 1	Signature	Date
………………………	……………………..	…………………
Internal Examiner 2	Signature	Date
………………………	……………………..	…………………
Internal Examiner 3	Signature	Date
………………………	……………………..	…………………
Chairperson Board of Examiner	Signature	Date

DEDICATION

I dedicate this work as a tribute to the scholars who paved the way in the field of System Development through their prior research and publications. Additionally, I extend this dedication to my beloved wife and children, acknowledging their unwavering support and understanding during the duration of my Master of Engineering in Information and Communication Technology studies.

ABSTRACT

This article highlights the important role of agricultural extension services for Zambian farmers, despite challenges in reaching remote areas. The government of the republic of Zambia has been using ICT to provide rural and remote farmers with e-Extension services through the Ministry of Agriculture website to help in agricultural planning. This application supplements traditional methods, offering a comprehensive solution to enhance crop production in Chongwe District.

The growth of Zambia's agriculture sector relies on timely detection and treatment of crop diseases, with the support of agriculture extension officers but the services were hindered by impact COVID-19's. Zambia's government is countering this by adopting electronic extension services, supported by provision of e-Extension services. Insufficient extension officers and reduced funding lead to inadequate support for farmers in Chongwe District. Additionally, the existing e-extension portal currently being used lacks user-friendliness and comprehensive data, which affects crop management. The purpose of this study was to develop and implement a crop management system that will support agriculture production in Chongwe District, providing expert information on crop production and management.

The study examined how extension officers in Chongwe district deliver services to remote farmers, assessing current e-extension services' effectiveness and seeking ways to improve user experience. The literature review provided a historical overview of farming practices and explored web-based extension services' emergence, comparing various mobile applications being used by farmers. However, limitations in existing systems, such as lack of comprehensive information and focus on specific crops, were identified. The efforts to enhance extension services in Zambia using technology, that addresses issues like crop management and disease were identified.

Studies identified the integration of video tutorials into e-extension services was for agricultural education and training. Videos offer visual demonstration, efficient learning, and standardization of information, improving adoption of best practices among farmers. Embracing digital solutions like mobile

applications and video tutorials enhances agricultural extension services, increasing crop production and supporting farmers' livelihoods.
Recent developments in Precision Agriculture (PA) sparked academic interest in Farming Management Information Systems (FMIS), aiming to boost efficiency. Research focuses on Decision Support Systems (DSS), computational models, and systems which are user friendly. Usability is key for adoption, however designing user interfaces faces challenges due to dispersed end-users with limited IT expertise. Microservice architecture suggests use of agile methodology to develop FMIS. Agile methodologies enable swift adaptation to dynamic conditions, fostering communication and collaboration among stakeholders.

Developers use Agile methodology to create applications in short bursts, focusing on developing, coding, and testing features to deliver a usable product at the end of each sprint.
In the System Development Process, requirements analysis involves meetings with agricultural experts to understand existing systems and define scope. In this study the researcher held meetings with the agriculture extension experts from the ministry of to develop a new system for the farmers in Chongwe with features user-friendly interfaces, multimedia integration, mobile accessibility, and tailored content for local farmers, bridging the gap between tradition farming and the use of ICT to enhance productivity and farmers livelihoods.

Keywords: Agricultural extension services, Information and Communication Technology, E-Extension services, Chongwe District, Zambia

ACKNOWLEDGMENT

I express my sincere appreciation to Dr. Simon Tembo, my supervisor, and Mr. IR Donat M. Ngendo, my co-supervisor, for their invaluable advice and contributions to my project. I am also grateful for the guidance offered by several system developers throughout the system development process. Lastly, I would like to acknowledge the cooperation and assistance extended by the extension officers from the Ministry of Agriculture Extension Office Department in defining system requirements and conducting User Acceptance Testing (UAT). Their support has been crucial to the successful completion of this project.

Table of Content

LIST OF TABLES

Table of Figures

LIST OF ACRONY

ICT	: Information and Communication Technology
COVID-19	: Coronavirus Disease 2019
UTA	: User Acceptance Testing
IAPRI	: Indaba Agricultural Policy Research Institute
UML	: Unified Modelling Language
HTML	: Hyper Text Markup Language
PC	: Personal Computer
UNZA	: University of Zambia
CMS	: Content Management System
GDP	: Gross Domestic Product
USD	: United States Dollar
RAM	: Random-Access Memory
DFD	: Data Flow Diagram
SQL	: Structured Query Language
URL	: Uniform Resource Locator
PA	: Precision Agriculture
FMIS	: Farming Management Information System
DSS	: Decision Support Systems
TAM	Technology Acceptance Model
SMS	: Short Message Service
JSON	; Javascript Object Notation
DBMS	: Database Management System
API	: Application Programming Interface
DOM	: Document Object Model
MVC	: Model View Controller
SDLC	: System Development Life Cycle
RAM	: Radom Access Memory
CMD	: Command Prompt

Chapter 1: Introduction And Background

1.0: Introduction

Zambian farmers mostly depend on the assistance of agriculture extension officers to enhance their crop production. In spite of the anticipation of regular visits made by extension officers from the ministry of agriculture offices throughout the agricultural calendar and guidance given on the use of e-Extension portal provided on the ministerial website, challenges still arise on how to reach those in remote and rural areas. This article highlights the crucial need for agricultural extension services in supporting farmers. Studies indicate that providing farmers with knowledge in crop management leads to the production of healthier crops even in the face of inadequate workforce and limited funding [1]. To address these challenges, I introduce an e-agriculture extension application designed to alleviate the issues faced by extension officers. This innovative application aims to empower remote and rural farmers with essential crop management knowledge and facilitate the development of successful agricultural calendars. Extension officers typically rely on an agriculture calendar provided by the agriculture methodologist at the Ministry of Agriculture's headquarters as their primary tool for crop management and planning. [1]. This article offers a comprehensive overview encompassing the background, objectives, project scope, methodology employed in system development, results, and a conclusive summary.

1.1 Background

The growth of the agriculture sector in Zambia is dependent on the capacity to detect and cure crop diseases at the earliest possible time. Farmers need to constantly monitor the crops at all times for any symptoms of diseases. In the wake of the Coronavirus Disease 2019 (COVID-19), farmers experienced difficulties in identifying crop diseases due to the absence of agriculture extension officers as a result of COVID-19 [2].

The government of the Republic of Zambia has adopted the use of electronic extension services by embracing the use of information and communication technology (ICT) to cover areas that do not have sufficient agriculture extension officers. This has been demonstrated by the use of the e-Extension portal provided on the ministerial website and this has been supported by the construction of the

two thousand (2,000) communication towers that the Ministry of Transport and Communication planned to erect countrywide [3].
In the United States of America, crop diseases are said to cause a loss in the agriculture output of about 20 to 40 percent annually. That is the reason why farmers are advised to promptly and timely diagnose all types of diseases affecting their crops to reduce the spread of diseases in their fields. Most of the time farmers try to observe and diagnose these diseases using manual system visually through the plant leaves which may be too complex and if not handled well may result in the use of wrong fertilizers and pesticides [4].

Farmers are required to be visited by the agriculture extension officers in all eight (8) Zones located in each agriculture camp at least twice (2) a month to help them manage the crops and diseases in their fields and recommend pesticides suitable for their crops, even though they are knowledgeable about their environment and farming system. However, in the wake of COVID-19, farmers found it difficult to get in touch with the agriculture extension officers. [1].

Due to a lack of extension officers and resources for extension services, farmers may not always be able to receive assistance from agricultural extension staff regarding crop illness and diagnosis; therefore, when extension workers are available, they are required to travel long distances to reach out to farmers [5]. Figure 1.1 below shows the allocation of the agriculture budget towards the extension services from 2010 to 2022 as produced by the Indaba Agricultural Policy Research Institute (IAPRI).

The graph in Figure 1.1 below which was analysed by the Policy Research Institute shows how funding for the agriculture extension services under the ministry of agriculture activities decreased from 2014 to 2017 and again from

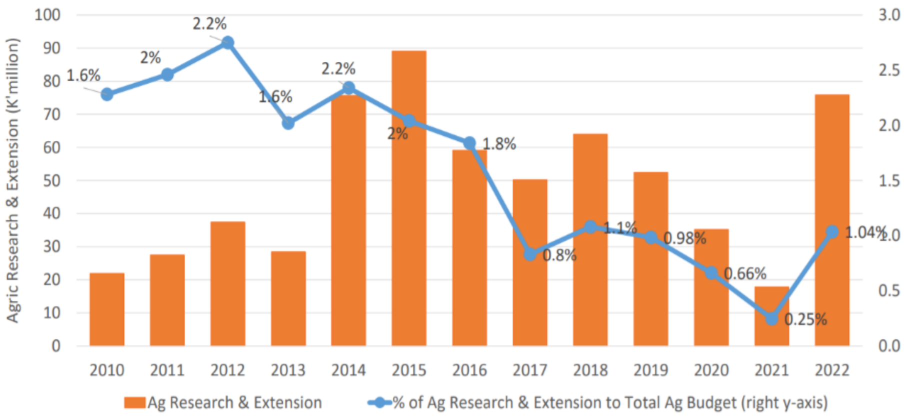


Figure 1.1 Indaba Agricultural Policy Research Institute (IAPRI) [5].

2018 to 2021. This proves beyond a doubt that the extension budget is unstable.

Based on statistical data, the Ministry of Agriculture oversees 2,701 agricultural camps with around 2.5 million farmers under the supervision of approximately 2,523 agricultural extension staff. [6]. According to the Chongwe extension office, the district is composed of four farming blocks. Each block comprises 20 camps, and within each camp, there are 8 zones. In total, there are 20 camp officers and 8 zone officers, resulting in approximately 160 extension officers for each agricultural camp in Chongwe District. On average, each agricultural camp caters to between 800 to 4000 farmers. This suggests that one extension officer serves between 800 to 4000 farmers in Chongwe District, whereas the recommended ratio is 1-500 extension officers to farmers.

The research that was done by Somanje [7] indicates that 72% of farmers had a good perception of agriculture extension officers. [6] In 2019, only 2% of the total agriculture budget was apportioned for research and extension services. In 2020, the budget was further reduced to 1.50%. [8, p. 12] and, In October 2020, farmers reported a decrease in the extension services to 56%. While in March 2021, the services had further decreased to 38%. [5]. In 2022, there was an increase in the budget as can be seen from the graph above in figure 1 compared to the past three (3) years. this clearly shows instability in the allocation of funds towards the services in Zambia, hence the development and enhancement of the already existing extension systems.

1.2 Problem Statement

Farmers in Chongwe District are expected to receive regular visits from agriculture extension officers across all eight (8) Zones within each Agriculture Camp, ensuring at least two visits per month to help in effective crop management. Unfortunately, the insufficient number of agriculture extension Officers and a reduction in funding for extension services have resulted in a significant gap in the required support for most farmers [1] .

The Ministry of Agriculture currently provides e-services to farmers through the Ministry website's e-extension portal. However, navigating multiple modules to access downloadable materials presents challenges, and the system's limited data types hinder user-friendliness. To address these issues, a crop management system has been developed and implemented for farmers in Chongwe District. This system is a web-based and aims to enhance the support provided to farmers by agriculture officers and overcome the limitations of existing extension services.

1.3 Purpose of the study

To design and implement a crop management system that will support agriculture production in Chongwe District of Zambia. The application will provide the necessary expert information on crop production and management as prepared by the department of extension service under the Ministry of Agriculture.

1.4 Research Objectives

The following are the main objectives of the project: -

(i) To identify and examine the information and communication technology (ICT) systems being used by agriculture extension officers to support agriculture extension services in Chongwe.

(ii) To analyse the ICT system used in Crop Management.

(iii) To Design and implement a crop management system for Chongwe. District.

1.5 Research Questions

The following are the research questions of the study.

(i) Examine crop production and management system currently used in Chongwe.

(ii) Analyse information and communication technologies and information systems used in crop management.

(iii) What ICT system should be designed and implemented for crop management in Chongwe.

1.6 Ethical Considerations

Participants and farms remain anonymous in this document as their names are not revealed. Also, participation in the User Acceptance Testing (UAT) of the Crop Management System was voluntary, ensuring that individuals engaged in the process willingly and with full understanding.

1.7 Significance of the study

The residents of Chongwe district in Zambia would utilize the developed application when seeking access to agriculture extension services, particularly in the face of geographical challenges like inadequate road infrastructure leading to their locations.

1.8 Scope of the Project

The research study sought to explore the system being utilised by agriculture extension officers to deliver extension services to farmers residing in the remote and rural regions of Chongwe district. It aimed to assess the effectiveness of the existing e-extension services accessed by farmers and identify opportunities for enhancing the user experience within the Chongwe district farming community in Zambia.

1.9 Theoretical and Conceptual framework

1.9.1 Knowledge Transfer Theory

This theory suggests that effective knowledge dissemination and utilization are important for improving practices and outcomes within a specific domain. In the context of agricultural extension services, knowledge transfer theory emphasizes the importance of transferring expertise from extension officers to farmers, enabling them to adopt best practices in crop management. The application of this theory in the development of the e-agriculture extension application facilitates the dissemination of essential crop management knowledge to remote and rural farmers, thereby enhancing their capacity to improve crop production [9] [10].

1.9.2 Technology Acceptance Model (TAM)

TAM suggests that the perceived usefulness and ease of use of a technology influence its adoption and usage. In the case of the e-agriculture extension application, farmers' perception of its usefulness in providing timely and relevant information on crop management, as well as its ease of use, will determine their acceptance and adoption of the technology. By considering the factors outlined in TAM during the design and implementation of the application, its effectiveness in supporting agriculture production in Chongwe District can be enhanced [11].

1.9.3 Conceptual Framework

(i) Agriculture Extension Services

This encompasses the traditional role of agriculture extension officers in providing advisory services, disseminating information, and offering technical support to farmers. The existing challenges in reaching remote and rural farmers due to limited resources and the COVID-19 pandemic underscore the need for innovative solutions such as the e-agriculture extension application [12].

(ii) Information and Communication Technology (ICT) Systems

ICT systems, including the e-Extension portal and the proposed crop management system, serve as platforms for delivering agricultural information and services to farmers. These systems leverage technology to bridge the gap between extension officers and farmers, providing access to expert knowledge and resources regardless of geographical constraints [13] [14].

(iii)Farmer Empowerment

Empowering farmers with knowledge and tools for crop management enables them to make informed decisions and adopt best practices. The e-agriculture extension application aims to empower remote and rural farmers by providing them with essential crop management information and facilitating the development of successful agricultural calendars, thereby enhancing their productivity and livelihoods [1].

(iv) Institutional Support and Funding

The Ministry of Agriculture's support and funding allocations for extension services play a crucial role in the effectiveness and sustainability of agricultural extension programs. Instability in funding, as evidenced by fluctuations in budget allocations, underscores the need for innovative solutions to enhance extension services and address resource constraints [15].

(v) User Acceptance and Adoption

The acceptance and adoption of the e-agriculture extension application by farmers depend on factors such as perceived usefulness, ease of use, and accessibility. By considering these factors and addressing user needs and preferences, the application can effectively support agriculture production in Chongwe District and contribute to the overall empowerment of farmers.

By integrating these theoretical perspectives and conceptual elements, the development and implementation of the e-agriculture extension application can address the challenges faced by extension services in Zambia and contribute to the enhancement of agricultural productivity and sustainability in Chongwe District.

According to S. Bekoe, e-Agriculture in Ghana has gained recognition due to its ability to support farmers and improve food security. It serves as a crucial tool for disseminating agricultural information, underlining the significance of its usage. Bekoe also mentioned the use of the Technology Acceptance Model (TAM) to comprehend factors affecting its adoption, aiming to offer insights into effectively applying technology in agriculture. [16].

1.10 Operational definitions

Agriculture Extension Officers: Trained professionals employed by the Ministry of Agriculture who provide advisory services, disseminate information, and offer technical support to farmers.

Agriculture E-Extension Portal: This is a digital platform provided by the Ministry of Agriculture, designed to deliver agricultural extension services, information, and resources to farmers.

Remote and Rural Areas: Geographical locations which is characterised by limited access to infrastructure and services.

Crop Management: The practices and techniques employed by farmers to cultivate, monitor, and protect crops from diseases, pests, and environmental stresses.

Crop Diseases: Pathological conditions affecting the health and productivity of crops which includes viral, bacterial, fungal, and pest-related ailments.

Crop Management System: A digital application developed and implemented to support agriculture production, to provide expert information on crop production, management, and planning.

1.11 Chapter Summary

Chapter 1 emphasizes the importance of agriculture extension services for Zambian farmers, especially those in remote areas. Despite efforts like regular visits by extension officers and agriculture e-Extension portals, challenges remain in reaching these communities. Providing farmers with crop management knowledge is key for improving crop health, especially with limited resources. An e-agriculture extension application has been introduced to empower remote farmers and help them in agricultural planning. The chapter discusses how extension officers rely on agriculture calendars from the Ministry of Agriculture and the instability of extension service budgets. Specific challenges in Chongwe District, like insufficient visits from extension officers and funding reductions, are explored. The chapter outlines the development of a crop management system to address these issues, aiming to better support farmers and improve the accessibility of extension services. Finally, it presents research objectives, research questions, ethical considerations, significance of the study, and scope, along with theoretical frameworks guiding the project.

Chapter 2: Literature Review

2.0: Introduction

The literature review describes a brief history of the topic, identifying breakthrough studies and publications cantered on the advantages and disadvantages of the specific study. It demonstrates the major issues concerning the study by identifying the gaps to be considered in the study. [17]

This research focused on the new era of farming which uses web application as a knowledge base for the provision of extension services to farmers. Similar applications have been developed by different developers for farmers. The review will try to compare different mobile applications that have been developed by different developers.

2.1 Related Work on e-Extension services

Through their close interactions with nature and the natural resources essential for food and fibre production, ancient farmers developed a manual system, as clarified by Chhetry and colleagues. Employing visual methods, farmers carefully managed their crops and manually identified diseases, drawing upon their past knowledge of the land and its resources. [18].

The COVID-19 restrictions that began in 2020 caused an unusual shock to most developing countries' farming activities, especially those in sub-Saharan Africa and South Asia. However, to reduce some of the effects caused by COVID-19, agricultural extension and advisory personnel which involved Anthony and friends created an avenue that helped farmers in these developing countries by using unconventional ways such as mobile phones, in providing agriculture extension services to the farmers. This unconventional method involved sending of Short Message Service (SMS), WhatsApp, and video recordings messages between the farmer and the extension officers. This means that farmers are required to continue communicating with the officers. This process may be very difficult if applied in Zambia looking at the ratio of Farmer to Extension officer. The developed system has all the information relating to the crops been cultivated in the area [19].

The main problem faced by farmers is dependence on the manual and traditional way of managing Crops production. As a result of this problem, the University of Zambia, Smart Zambia Institute, and the Ministry of Agriculture worked together to create a system that gives farmers in Zambia access to e-extension services [20]. The only weakness with this system is that famers can only access text and image-based agriculture information service modules. The system does not have an agriculture activity calendar for each month. [21]Annual agriculture calendars play a very important role in crop management planning for farmers. That's why the Ministry of Agriculture's Integrated Cropping Calendar Information System has developed a planting calendar for food crops in Indonesia. This calendar estimates the initial planting time based on yearly rainfall conditions, categorising them as wet, neutral, or dry. It helps farmers plan which crops to plant according to the prevailing weather conditions. [22]. The crop management system includes a feature that enables farmers to access information about which crops to plant in specific months. This system not only provides access to textual materials but also allows farmers to access information in video format. The system is user friendly and does not need farmers to navigate through various modules.

Social media and web 2.0 applications developed by Thomas and friends helped to open up a platform that was used by farmers, agricultural extension officers, agricultural institutions, and nongovernmental organisations to disseminate and exchange agricultural information. The study aimed at assessing the use of social media as a source of information for the farmers in the Kesses District of Kenya. Most farmers used social media in seeking agricultural information, However the use of social media comes with its challenges of misinformation among other things. Information may not be authentic, and the web 2.0 application may not have systematically arranged information as per agriculture calendar used in Zambia by the extension officers [23]. The crop management system is updated with the authentic information that is used by the extension officers in the dissemination of information to the farmers.

The use of web applications in Agriculture contributed to an increase in agriculture production. It was a core component that helped in increasing the productivity of crops and indirectly increase the Gross Domestic Product (GDP) of India that in turn reduced poverty. Koli and Raut developed the application which was user-friendly and was able to be used by farmers and agriculture institutes to determine the average ph. of the soil, soil type, and average temperature. In so doing, a farmer knew exactly the type of crops that was to be cultivated on a particular land with a precise soil type [24].

Raikar and his colleagues developed a system that was able to identify diseases and crop yield prediction. The application was in form of a website and was able to be accessed on mobile phones using an internet connection. The system was built to suit the soils, crops, and climate of India. Web applications help farmers to access information while the experts are not available. Similarly, the developed application has information on the crop management and production [25].

Shiferaw and his colleagues developed a system that was only able to manage potato crops for the farmers in Ethiopia. The system was aimed at examining two types of disease namely wilt and late blight. This system was used as a survey, to ascertain the knowledge that farmers have concerning the two diseases compared to what the system was able to identify. The only drawback with this system is that it was only able to deal with one specific crop and two specific diseases that would have reduced the yield of potatoes in the selected religion. The developed system will cover a wide range of crops mostly cultivated by the farmers in Zambia [26].

Mahapatra reports that, the use of mobile application technologies as a tool in agriculture is gradually becoming common. He developed a system that was used to reduce poverty in India. The system was used by the farmers during COVID-19 lockdowns as they were not able to interact with the agriculture extension officers. Internet and web applications acted as an important tool for agricultural information dissemination during the lockdown. This system was just seen to be an important tool during the lockdown period. The system helped in bridging the gap of not having enough agriculture extension officers who are vested with expert knowledge in agriculture. The developer only targeted the COVID-19 lockdown period [27].The Crop Management system will be used even after the COVID-19 pandemic.

The COVID-19 pandemic affected more than 4.3 million and killed more than 2,090,000 deaths worldwide. This gave rise to fears of an economic crisis and possible recession in some countries. Social distance, self-isolation, and travel restrictions led to a reduction in the workforce across all economic sectors and it also contributed to loss of many jobs. In Malaysia, the COVID-19 pandemic tested the resilience of the agriculture sector [28].

The software that Shubham created in response to farmers' errors in selecting the incorrect crops, fertilizers, and pesticides helped India's sustainable agricultural development [29]. Though this was implemented in India, some of the ideas can be used in Zambia.

The state Agriculture University of India creates systems that offer farmers extension services. These technologies are critical to the nation's agricultural prosperity since they form its foundation. The lack of extension officers is one of the justifications given for the promotion of these technologies [30].

The right agriculture information is important to farmers as it plays an important role in gaining maximum productivity with effective utilisation of available resources. Farmers are guaranteed fair and prompt access to the highest quality extension services by the extension officers. The conventional door-to-door extension method is still important for disseminating agricultural information, but it is currently facing punitive criticism for its inability to guarantee timely and equitable access to such information. The only advantage with electronic information systems they can be accessible from anywhere at any time. One may argue that e-agricultural is the informational future of agriculture [31].

The cost of providing extension services to individual farmers was observed by Saravanan Raj to have decreased by 3.6 times from Rs. 2,400 (USD 53) under the e-agriculture prototype compared to the conventional extension system. However, this article argues that areas that are less developed require a hybrid form of extension services to handle the post-harvest reviews of India in order to reduce poverty [32]. The hybrid form of extension services can still be applicable to Zambian setup.

2.2 Integration of video tutorial into e-Extension services

Tian Cai and friend started that videos offer several advantages over text-based learning materials in agricultural education and training. Firstly, they can be highly persuasive, making complex concepts and technologies easier to understand when demonstrated visually.

Additionally, videos allow for the compression of lengthy agricultural processes into short segments, thereby increasing training efficiency. The flexibility of video content enables it to be accessed anywhere and at any time, making it convenient for farmers to access information when needed. Moreover, videos can standardise the information provided to farmers, ensuring consistency in messaging.

Localised videos that are tailored to the social context of the target audience are more likely to be accepted, as they provide evidence that recommended practices are effective in the local environment.

Furthermore, when used for training purposes, videos are typically shown to small groups of farmers living near each other. Group participation in training sessions encourages learning, adoption, and innovation through peer pressure, and it can even strengthen social bonds within communities [33].

Overall, videos offer a dynamic and effective tool for agricultural education and training, facilitating learning and adoption of best practices among farmers. Local facilitators with agricultural training are often employed to conduct training sessions and monitor the progress and adoption rates of recommended practices [33]. This is contrary to the current system used by the agriculture extension officers in Chongwe to disseminate e-extension to the farmers.

The study highlighted the fundamental role of video-based agricultural education in disseminating crucial knowledge and practices among farmers. Compared to conventional training methods, video reached a significantly higher number of women, transcending barriers imposed by participant selection biases and elite monopolies. Moreover, video-based training helped overcome local power structures and reduced community-level conflicts, fostering a more inclusive learning environment. The adoption rates of improved practices, such as rice drying techniques and equipment utilization, were substantially higher among participants exposed to video-based training. The widespread translation and dissemination of rice videos in multiple African languages further underscored the scalability and effectiveness of this approach. Overall, the study underscores the transformative impact of farmer-to-farmer video in agricultural extension, providing valuable insights into its role in improving farmers' practices and attitudes towards agricultural technology dissemination [34]. This article emphasises the importance of including video tutorials in e-agriculture Extension Services in order to increase crop production.

2.3 Use of ICT technology in agriculture

According to Ugwuishiwiu, computer technology initially emerged as a powerful tool for calculation and document preparation, while communication technologies concurrently evolved for information transfer across locations. Over time, these technologies converged, giving rise to Information and Communication Technology (ICT), sometimes referred to simply as Information Technology (IT). Agriculture and rural development are experiencing significant shifts due to globalisation and transformative technologies altering production organisation. ICT, however, lacks a universally accepted definition due to its constantly evolving concepts, methods, and applications. ICT encompasses products capable of electronically storing, retrieving, manipulating, transmitting, or receiving information in digital form, including personal computers, digital television, email, and robots. E-Agriculture involves the innovative utilisation of ICT in rural settings, particularly in agriculture, covering conceptualisation, design, development, evaluation, and application. In many developing countries, ICTs represent rapidly growing industries of increasing macroeconomic importance. The lack of a robust technological infrastructure, particularly in rural areas, hinders sustainable development and exacerbates poverty issues in Africa. Despite this, ICT penetration remains largely limited to urban centers, despite approximately 70% of citizens in developing countries residing in rural areas [35].

According to Chavula, while agriculture and natural resources are expected to remain key drivers of Africa's economic growth, it is the application of modern technologies that is anticipated to have the most substantial impact on the growth trajectories of most African economies. The researcher emphasizes that technology offers significant opportunities in agriculture. Consequently, countries have identified ICT as a crucial component in transitioning from subsistence-based economies to service-sector driven ones [36].

In the context of India, the increase of mobile telephony, both in rural and urban areas, offers a promising avenue to disseminate pertinent information on a broader scale. This has the potential to enhance the efficiency of agricultural markets and address various challenges encountered by the sector. Numerous studies within the Indian context have examined the effects of information and communication technology (ICT) as well as mobile-enabled agricultural services on the delivery of extension services. These investigations consistently highlight the role of mobile telephony in fostering awareness among farmers [37].

The author highlights the advantages of ICT in e-agriculture, emphasizing its role in promoting innovation and economic growth globally. According to the author, e-agriculture enhances agricultural and rural development through improved information and communication processes. Hussaini underscores key benefits such as increased productivity, efficient resource management, timely access to information, and support for decision-making [38].

2.4 Use of ICT in Chongwe for agriculture purpose

In Zambia, Information and Communication Technology (ICTs) have become widespread across various sectors, including agriculture, leading to increased productivity. However, the adoption of ICTs in agriculture is still limited, especially among rural farmers in Africa, notably in Zambia. Challenges in accessing and utilising these technologies persist, preventing many farmers from benefiting fully. Research suggests that access to information through mobile phones and mobile internet can help agriculturalists mitigate risks and adapt to climate change [39]. The Crop Management System advocates for the adoption of ICT to assist farmers residing in remote areas of Chongwe district, particularly in managing the crops cultivated in the region.

As per the Ministry of Agriculture's e-Extension department, they are currently using the Ministry's website to deliver e-extension services to farmers in the Chongwe district of Zambia [1] though the system has a challenge as it can only provide two data type that is text and picture and not video.

The application that has been developed is intended to be accessible through both mobile smartphones and computers, given that it is a web-based application. In 2012, Chongwe had approximately 70,000 Airtel subscribers [40] .

In 2016, research conducted by [41] revealed that the number of mobile phone subscribers in Chongwe district had risen to approximately 127,817 across all mobile networks. With this figure steadily increasing each year, the outlook for e-extension services appears promising.

2.5 System Design Process

According to Arthur M. Langer, process flow diagrams (PFDs) are a part integral in examining the completeness and sequence of a process, a perspective that data flow diagrams (DFDs) alone may not capture. While DFDs provide insights into data flow within a system, they lack the depiction of sequential steps inherent in processes. PFDs, in contrast, outline the sequence of steps, which assist analysts in identifying any omissions or gaps in the system.

The sequential representation offered by PFDs proves particularly valuable when confronted with potentially inaccurate or incomplete information from users. By guiding users through each phase of the system in its actual sequence, PFDs help uncover overlooked details, thus serving as a means to verify the thoroughness of interviews and comprehension of the system's logical environment (LE).

While some methodologies advocate for the supremacy of PFDs over DFDs due to this advantage, Langer suggests that both models can harmoniously coexist and complement each other. While PFDs shed light on process sequence and time, DFDs focus on the net inputs and outputs of flows and the disposition of data at rest.

PFDs offer a more streamlined and adaptable approach compared to traditional flowcharts, rendering them more suitable for structured modelling tasks. Despite disparities, both PFDs and DFDs encompass processes, externals, and data flows. Their disparity lies in emphasis: PFDs prioritize the temporal aspect of processes, while DFDs concentrate solely on data flows and the state of data at rest [42].

2.5.1 Role Activity Model in System design

In system development, Role Activity Diagrams (RADs) and Data Flow Diagrams (DFDs) are essential for capturing and understanding complex interactions and behaviours within processes.

RADs, according to Ould [43], focus on roles within a process, representing tasks and collaborations, thus making them ideal for organizational contexts. They visually depict roles, activities, interactions, and logical elements, aiding in communication and understanding of process aspects. RADs also highlight activity sequences and interactions between roles, enhancing clarity and identification of key process components. DFDs complement RADs by illustrating the flow of data within processes, showcasing interactions and dependencies between processes and external entities.

By adopting both RADs and DFDs, organizations can derive system models from business process models, ensuring alignment between system functionality and business processes. This approach involves developing a Business Process Model using RAD, identifying automated activities, and creating a Functional Specifications Document, followed by DFD development based on these requirements [44]. One of an example of an activity diagram in system development is the login process for the user.

2.6 Methodology and software consideration for Crop Management System

2.6.1 System Methodology

(i) Agile

According to [45] recent developments in agriculture, particularly in Precision Agriculture (PA), have spurred academic interest in Farming Management Information Systems (FMIS). These systems aim to enhance efficiency in agricultural practices. Academic research in this field has focused on three key criteria: Decision Support Systems (DSS), computational biological models, and system usability. Usability is important for adoption; if a system isn't user-friendly, it won't be adopted. However, designing user interfaces for FMIS presents challenges, as end-users are often geographically dispersed and lack Information Technology (IT) expertise. The researchers suggested the use of microservice architecture for agile FMIS in precision agriculture.

Hasan emphasizes that the distinctive feature of applying an Agile mindset in farm management is the inherent unpredictability and complexity of the agriculture industry. Agile methodologies empower farmers to adeptly navigate this complexity by swiftly adapting to dynamic conditions and making informed decisions grounded in data and feedback. Moreover, Agile enhances communication and collaboration among stakeholders, which is paramount in an industry characterized by numerous participants engaged in crop production and distribution [46].

Agile is one of the top software developments mostly used by developers as can be seen by the pie chart below in figure 2.1 below [47].

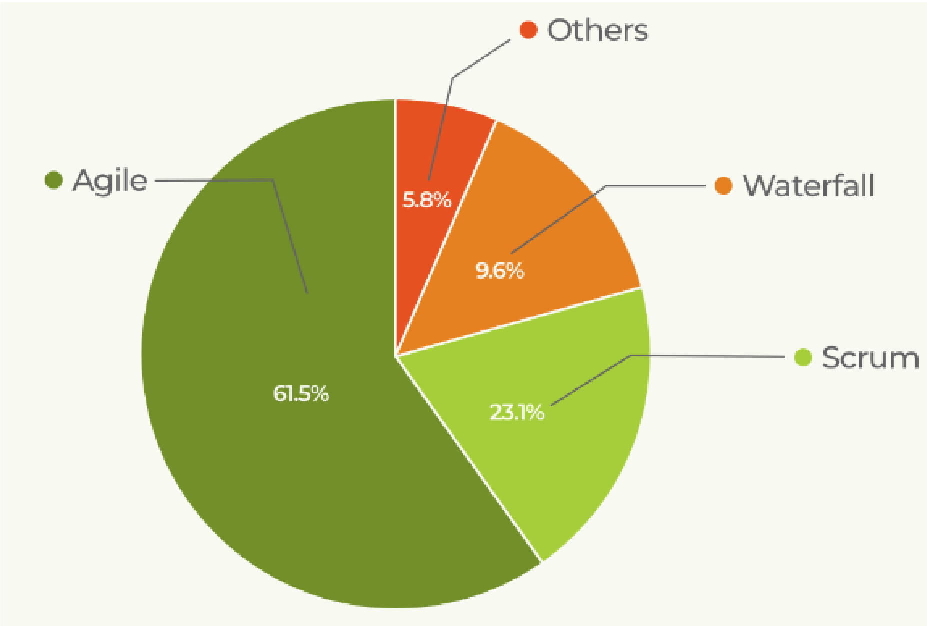


Figure 2. 1 Agile methodologies compared to others.

Using the Agile technique, a developer can create an application in short bursts, known as sprints, which can range from one (1) to four (4) weeks. Every sprint is developed, coded, and tested in the same manner as a finished feature of a functional product. Creating a usable product is the sprint's goal. A developer could design and provide a functional product, for instance, if they wanted to create a login page, it can be designed, developed, and implemented. Figure 2.2 shows Agile development process using sprints.

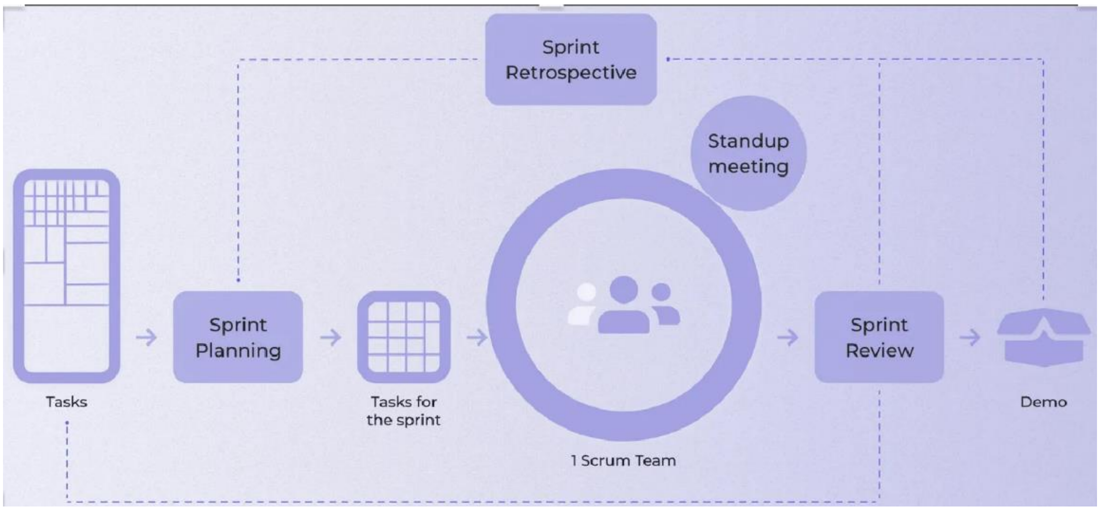


Figure 2. 2: Agile development process using sprints.

a) Components of Agile Methodology [48]

- **Task:** This refers to the project being undertaken by the developer, in this instance, it pertains to the Crop Management System.

- **Sprint Planning:** In this phase, the developer collaborates with the user or client to prioritize the features to be included in the application.

- **Standup Meetings:** These daily meetings serve to update development team members on the progress made on each task.

- **Sprint Retrospective Meetings:** These meetings involve the team discussing issues encountered during sprints and brainstorming improvements.

- **Demo:** The project manager meets with the client to showcase the accomplishments achieved by the development team.

According Khudadad and friends [49] agile stands as a widely preferred in software development for its ability to efficiently complete small projects and which breaks down larger systems into manageable modules.

2.6.2 Software

(i) MongoDB as a NoSQL Database Management System (DBMS)

[50]MongoDB is a powerful open-source NoSQL database that offers numerous advantages for modern data management. Initiated by 10gen Company, MongoDB stores documents in binary encoded Javascript Object Notation (BSON) format, a binary form of JSON, which supports various data types such as strings, integers,

dates, Booleans, floats, and binaries. One of the key features of MongoDB is its schema-less nature, allowing users to insert new fields into documents or update existing structures without constraints. Unlike relational databases, MongoDB does not use joins; instead, it utilizes embedded documents for efficient data access.

MongoDB offers robust replication features, including Master-Slave replication for backups and reads, and supports cluster configurations comprising Shard nodes, Configuration Servers, and Mongos instances. Shard nodes store data across one or more shards, while Configuration Servers manage routing information, and Mongos instances serve client requests by collecting and merging data from different shards. Key features of MongoDB include flexibility in storing data in JSON format, a rich query language with dynamic queries for linear scalability, for easy installation and maintenance, high performance with reduced I/O activity and indexing for faster queries, and high availability through replica sets for automatic failover and redundancy.

Moreover, MongoDB supports multiple storage engines such as Wired Tiger and MMAPv1, along with a pluggable storage engine API for third-party development. These features make MongoDB an ideal choice for modern applications requiring flexibility, scalability, ease of use, high performance, and high availability in data management [51].

(a) MongoDB Algorithm for system development

- **Define the Data Model:** Identify the entities and their attributes that need to be stored in the database. Design the schemeless document structure using BSON format to represent the data model.
- **Install MongoDB:** Download and install MongoDB on your system or server. Set up the necessary configurations such as data directory, log file, etc.
- **Start MongoDB Service:** Start the MongoDB service using the appropriate command for your operating system. Verify that the service is running correctly.
- **Connect to MongoDB:** Establish a connection to the MongoDB server from your application code. Use the MongoDB client or driver compatible with your programming language.

- **Create a Database:** Use the MongoDB client to create a new database for your application. Choose a meaningful name for the database based on your application's purpose.
- **Insert Data:** Write code to insert documents (data records) into the MongoDB collections (similar to tables in relational databases). Ensure that the data inserted conforms to the defined data model.
- **Query Data:** Implement queries to retrieve data from MongoDB collections based on various criteria. Use MongoDB's query language and operators to filter, sort, and limit the results as needed.
- **Update Data:** Implement update operations to modify existing documents in the MongoDB collections. Use update operators such as $set, $inc, $push, etc., to perform specific modifications.
- **Delete Data:** Implement delete operations to remove documents from MongoDB collections. Exercise caution when deleting data to avoid unintentional data loss.
- **Indexing:** Create indexes on fields that are frequently used in queries to improve query performance. Consider the types of queries performed in your application and create appropriate indexes.
- **Error Handling and Security:** Implement error handling mechanisms to gracefully handle exceptions and errors that may occur during database operations. Follow security best practices to secure your MongoDB deployment, including authentication, authorization, and encryption.
- **Testing and Optimisation:** Test your MongoDB implementation thoroughly to ensure its functionality, reliability, and performance. Profile and optimize database operations as needed to improve overall system performance.
- **Monitoring and Maintenance:** Set up monitoring tools to monitor the health and performance of the MongoDB deployment. Perform regular maintenance tasks such as backups, index optimizations, and software updates.
- **Documentation:** Document your MongoDB implementation, including the data model, database schema, query patterns, and any custom configurations. Provide clear documentation for other developers who may work on the project in the future.

(ii) Angular.js

AngularJS is considered the best programming language for several reasons outlined in the research. Firstly, it offers versatility and ease of use, making it suitable for a wide range of web development tasks. Secondly, it has strong community support, which means developers can access resources and assistance when needed.

One of AngularJS's key advantages is its implementation of two-way data binding, which allows for seamless synchronization between the model and the view. This feature simplifies development by automatically updating the UI when the data changes, enhancing user experience and productivity.

Comparative evaluations with other frameworks like jQuery and BackboneJS highlight AngularJS's superior performance in terms of handling Document Object Model (DOM) interactions. Despite potential delays in managing multiple entries simultaneously, AngularJS demonstrates efficiency in this aspect.

AngularJS also boasts a well-structured Model-View-Controller (MVC) architecture, facilitating clean and maintainable code. Its dependency injection mechanism and directives further enhance development productivity and code quality.

In addition to these technical advantages, AngularJS consistently ranks high in performance evaluations, considering factors such as code quality, performance, and security. Overall, the research recommends AngularJS as the preferred JavaScript framework for its robust features, performance, and ease of development, supported by thorough evaluations and comparisons with other options [52].

(iii) Node.js

According to Lei and friends [53], in the dynamic landscape of web development, challenges such as accommodating multiuser requests and managing high concurrency have surfaced. JavaScript, renowned for its dynamic scripting capabilities, has become a cornerstone in web development, particularly for client-side tasks. Node.js, a technology built upon Chrome's JavaScript runtime, stands as a significant advancement in this realm. It offers a robust platform for constructing fast and scalable network applications, capitalizing on an event-driven, non-blocking I/O model to deliver lightweight and efficient performance. Node.js is particularly well-suited for data-intensive real-time applications across distributed devices. The growing popularity of Node.js is evident, with average downloads surpassing 35,000 since its version 0.10 release in March 2013, coupled with major Platform as a Service (PAAS) providers embracing it [53] [54].

(iv) GitHub

Borges and friends describe GitHub as a top platform for open-source software development with lots of features and a huge user base. It has 9 million users and 17 million public repositories. Besides being a git-based version control system, it also lets developers interact socially. They can copy repositories, work on code, and ask for changes to be merged. Also, users can show interest in projects by starring them, and the number of stars shows how popular a project is.

Understanding what makes a GitHub project popular is very important for developers. By looking at things like programming language, what the project does, who owns it, how old it is, how often code changes, how many people contribute, and how many copies are made, developers can see if their projects are doing well. This helps them attract new users, see if updates are successful, and stay competitive [55].

(v) Visual Code Navigator

Lommerse and friends state that Software maintenance involves understanding complex codebases, which can be challenging given their size and complexity. Studies have shown that developers spend a significant portion of their time understanding code. To address this, software visualization tools are used to help developers gain insights into program structure and processes, such as reverse engineering and process recovery.

In this paper, the authors introduce the Visual Code Navigator (VCN), a toolset designed to provide multiple views of software source code. VCN aims to assist developers in understanding the structure and changes of large software systems. The toolset comprises three interconnected views: the syntactic view, the symbol view, and the evolution view. These views visualize different aspects of the codebase hierarchy, such as syntactic constructs, symbols available after compilation, and changes over time [56].

(vi) Strapi

Adservio experts explains Strapi as a headless Content Management System (CMS), separating content management from presentation. Unlike platforms like WordPress, Strapi lacks a built-in frontend, offering an API for content creation and retrieval. This decoupling grant flexibility, enabling developers to implement their preferred frontend technology. It allows content reuse across various platforms and facilitates integration with other systems through its standardized API. The difference between Strapi and traditional CMS like WordPress lies in the separation of frontend and backend functionalities [57]. Figure 2.3 below shows the difference between traditional CMS and Strapi.

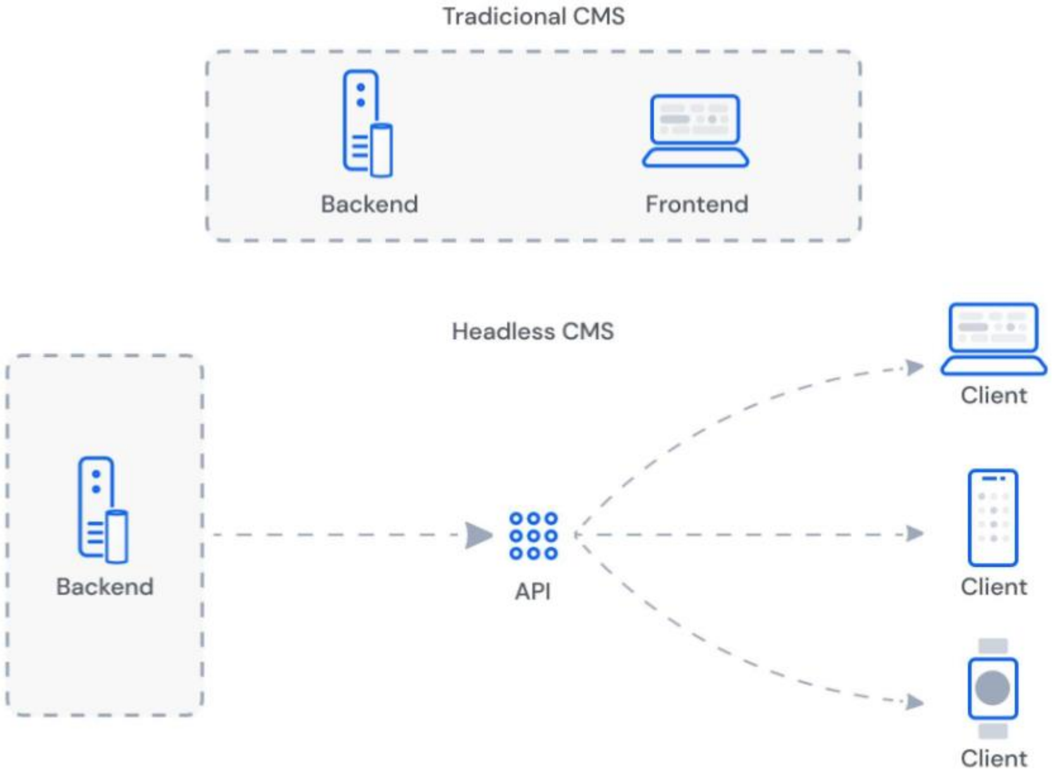


Figure 2. 3: Illustrates the difference between traditional CMS and Strapi [57].

The following are some of the advantages of using strapi [57]

- Strapi supports multiple database architectures including PostgreSQL, MySQL, MongoDB, and SQLite, offering easy setup and configuration.
- Built on the open-source Koa framework, Strapi provides accessible and extensible source code.

- Easily tailored to suit project requirements, Strapi allows for custom data. types of creation via the admin panel. Its plugin system facilitates the addition of extra features.
- Strapi offers a REST API for consumption by various client frameworks like VueJS or Angular, ensuring flexibility in development.
- Strapi handles authorization, allowing granular control over user access to endpoints. It supports OAuth authentication for integration with third-party. tools like Google or GitHub.
- Strapi can scale effortlessly with Docker, Kubernetes, and cloud providers, enabling it to handle high traffic loads effectively.

2.6.3 Data (Video, Text, and Picture) Processing

The system uses REST API to process the videos accessed by the system from various databases using JSON. According to W3schools JSON serves as a data storage and transport format, commonly utilized for transmitting data from a server to a web page [58]. Figure 2.4 below shows data processing using REST APIs.

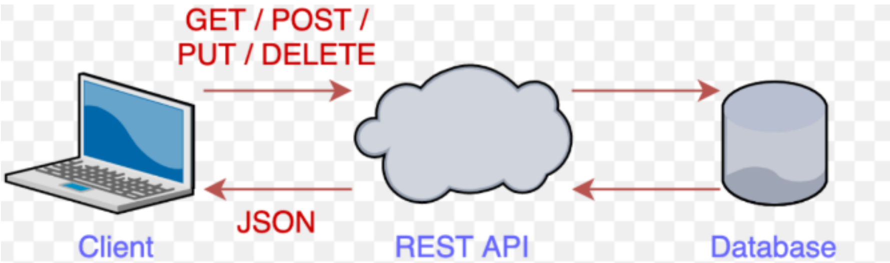


Figure 2. 4: Data Processing using REST API [59]

Navraj explains that REST API enables software applications to communicate over the internet by exposing specific endpoints for client access. These endpoints allow programs to share and exchange data, with clients sending structured requests to retrieve desired information. The API responds with data usually in JSON or XML format [60].

2.6.4 Summary

Chapter 2 covers how technology is changing agriculture, focusing on e-extension services, video tutorials, system design, and software tools. It discusses how agriculture services, like those in Zambia, are adapting to challenges such as COVID-19 by using mobile phones and social media for information sharing. Video tutorials are highlighted as a useful tool for educating farmers and improving agriculture practices. The chapter also explores the broader impact of ICT in agriculture globally, discussing its role in innovation and productivity. In Chongwe, Zambia, there's a focus on how ICTs can help farmers, especially in remote areas. The technical side of the system design is explained, including diagrams and methodologies used. Lastly, the chapter explains the importance of using software tools like MongoDB, Angular.js, Strapi, and Node.js in system development showing how data is processed and managed in the systems. Overall, Chapter 2 provides a thorough overview of how technology is reshaping agriculture, from practical applications to theoretical concepts.

Chapter 3: Methodology

3.0 Introduction

Research is a fundamental aspect of any scientific investigation, employing a structured approach known as research methodology. The methodology serves as a guiding framework for developers, facilitating efficient work, effective collaboration, and adherence to project deadlines and budgets. It provides a structured approach to organizing, planning, and overseeing the development process of an information system.

3.1 System Development Methodology

Satzinger and others, describes an information system as a group of connected parts that gather, process, store, and deliver the output data required for a full-fledged commercial endeavour. An application is a software that can be run on a computer device to perform a particular task. [61].

According to Tran [62], system development methodology is important for project success, guiding developers in planning, executing, and validating projects. Adopting a methodology helps developers work efficiently, collaborate effectively, and meet project deadlines and budgets.

Software development methodology provides a structured framework for organizing, planning, and overseeing the development process of an information system. [63]

3.3 System Development

Systems development involves defining, designing, testing, and implementing a new software application or program. This process may encompass internal development of customised systems, creation of database systems, or acquisition of third-party developed software [64].

The diagram in figure 3.1 below illustrates the workflow followed for system development life cycle (SDLC) for the Crop management System.

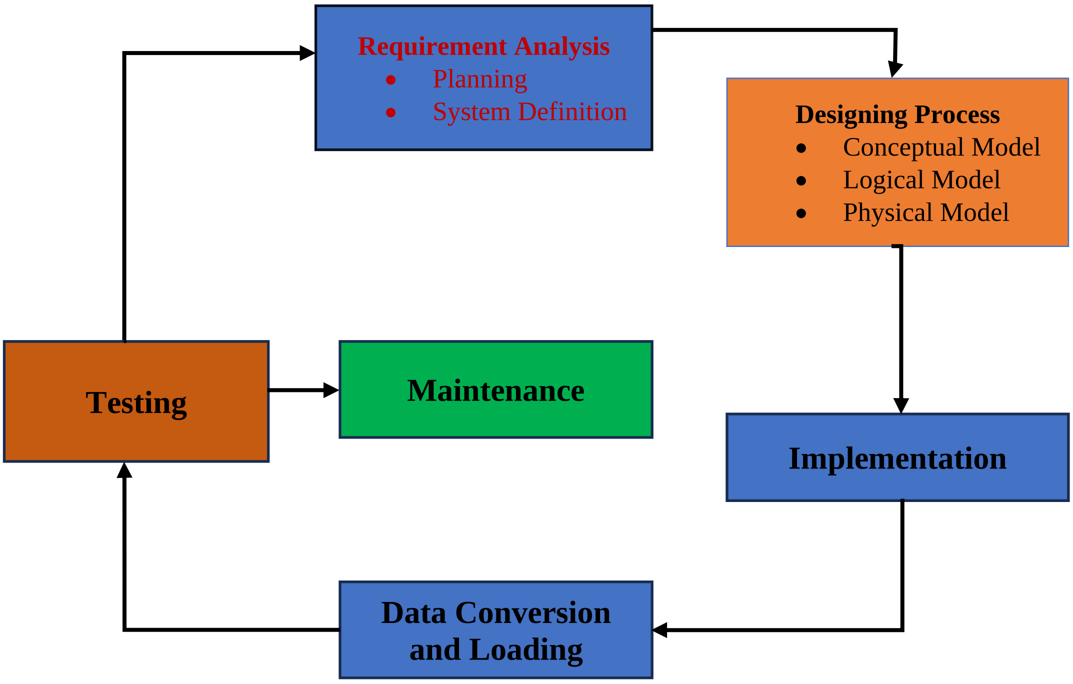


Figure 3. 1: Crop Management System Development Process

3.3.1 Requirements Analysis

The researcher held meetings with the agricultural methodologists at the Ministry of Agriculture and Extension Officers in Chongwe to discuss and understand the existing system employed for delivering e-extension services to farmers in the Chongwe. Before commencing with the design of a database, the scope and the boundary of the system was defined along with its interfaces and other components of the information system as guided by [65].

(i) Research Instruments

These are tools and methods that are used to collect data for the research being carried out [66]. In this project, the researcher used interviews method and the current annual agriculture calendar in collecting data for the project from the methodologist at the Ministry of Agriculture.

The researcher also conducted interviews with the Ministry of Agriculture Extension Officers in Chongwe District to determine the methods being utilised in providing extension services to farmers in the area.

(ii) Documentation

According to the agricultural experts, the ministry of agriculture uses an annual agriculture calendar with outlined activities for each month which farmers are required to follow. The extension officers conduct meetings with Farmers at least twenty-four (24) times a year and some of the activities in the calendar are outlined in the table 3.1 below: -

Table 3. 1: Annual Agricultural Calendar

General Topic	Objectives	Key Message	Timeline per Quarter
Planning of Agriculture Calendar	Create awareness on the importance of planning in agriculture	User the agriculture calendar to select plants to grow	Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sept, Oct, Nov, Dec,
Land Preparation and Soil Nutrient Status	Create awareness on land preparation, soil testing, soil fertility improvement, and early planting preparedness	Start your land preparation early i.e. just after harvesting and use basins and ripping where possible	Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sept, Oct, Nov, Dec,
Crop residue management	Create awareness of crop residue management	Do not burn crop residues except those for cotton	Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sept, Oct, Nov, Dec,
Planting	Create awareness on the importance of planting on time	Plant with the first effective rainfall to ensure crops have enough rain periods to reach maturity.	
Timely weeding	Create awareness on timely weed management.	Use manual, chemical, or mechanical methods for weed control.	Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sept, Oct, Nov, Dec,

(iii) Interaction diagrams

The research also used interaction diagram to further understand the requirements of the system as shared with the agriculture extension Officers. These stand as part of the functional requirements of the system. Interaction diagrams outline the software functionalities that developers need to include in system development to assist users in the provision of information, thereby fulfilling the business requirements [67]. The extension officers suggested the following requirements for the system: -

The system administrator should be able to:

(i) Login

(ii) Manage Crop

(iii) View/Update Crop

(iv) Add Camp

(v) Add Zone

(vi) Create Report

(v) Manage Calendar

The system user (Farmer) should be able to:

(i) View crop

(ii) View Calendar

The following are some the interaction tools that was used during system requirement analysis.

- **Use Case Diagram**

According to E. Wiegers, and Beatty, a use case diagram illustrates a series of interactions between a system and an external actor, to achieve valuable outcome. Use case names consistently follow a verb-object format [67] as indicated in figure 3.2 below. The figure shows the use case diagram for the admin and the farmer(user).

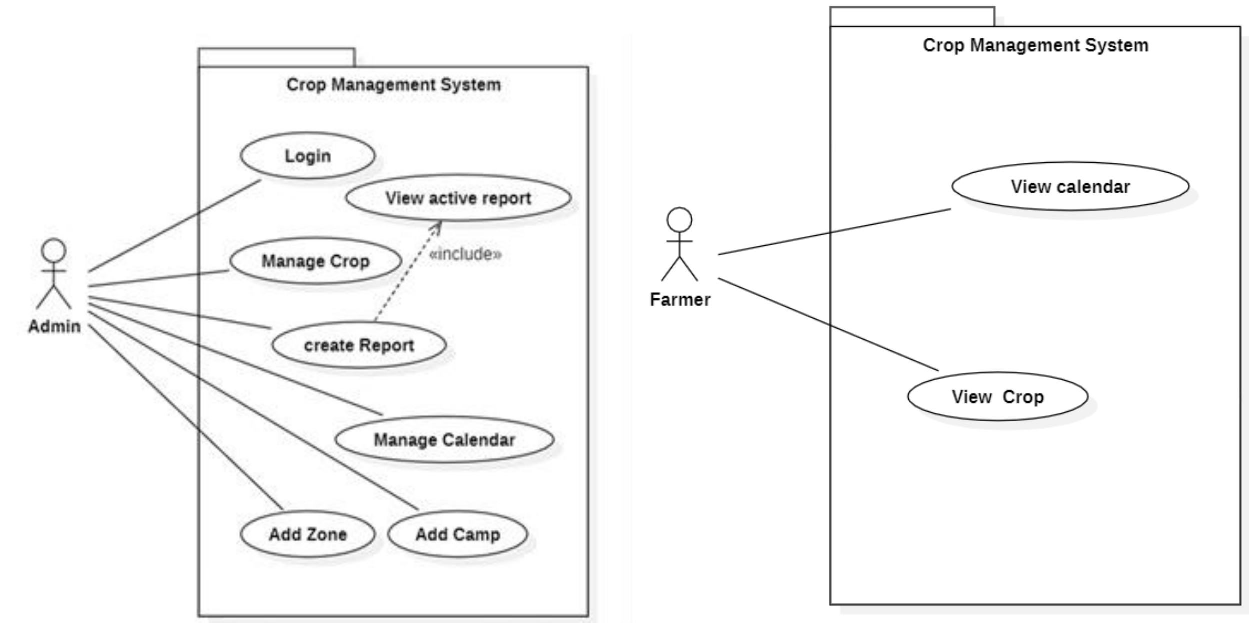


Figure 3. 2: Use Case Diagram for both Farmer and Admin

- **Sequence Diagram**

A Sequence diagram also called an interactive diagram, it shows a communication comprising a set of objects and their relationship as well as the messages that may be dispatched among them. [68, p. 65]

The diagram in figure 3.3 below shows a sequence diagram that emphasises the time ordering of messages. Graphically it shows objects arranged along the X-axis and messages ordered in increasing time along the Y-axis.

the sequence diagram in figure 3.3 below shows the interaction and time ordering of messages between the users (admin and farmer) and the system.

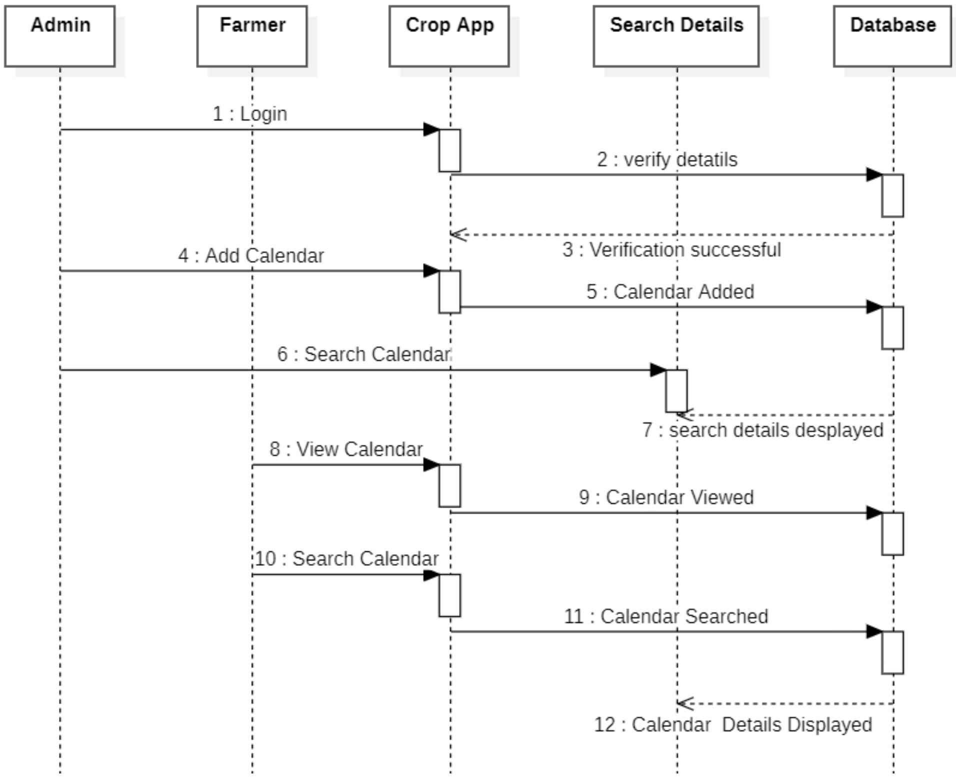


Figure 3. 3: Sequence Diagram for an Agriculture Calendar

Furthermore, a Sequence diagram shows a complete flow of specific use case. It shows the communication between different objects in their sequence [69]. A Sequence diagram represents the behaviour of a series of sequential steps over time depicting the workflow of messages being passed and how elements achieve the results. They capture the flow of information throughout the system [70].

- **Collaborate Diagram**

This is also called a communication diagram; it illustrates the relationships and interactions among software objects in the Unified Modelling Language (UML). In this case, it illustrates the relationship between the admin, the database, and the users of the application [71] as illustrated in figure 3.4 below; -

figure 3.4 below shows the relationship between the admin, the database, and the users of the application as exaplained above.

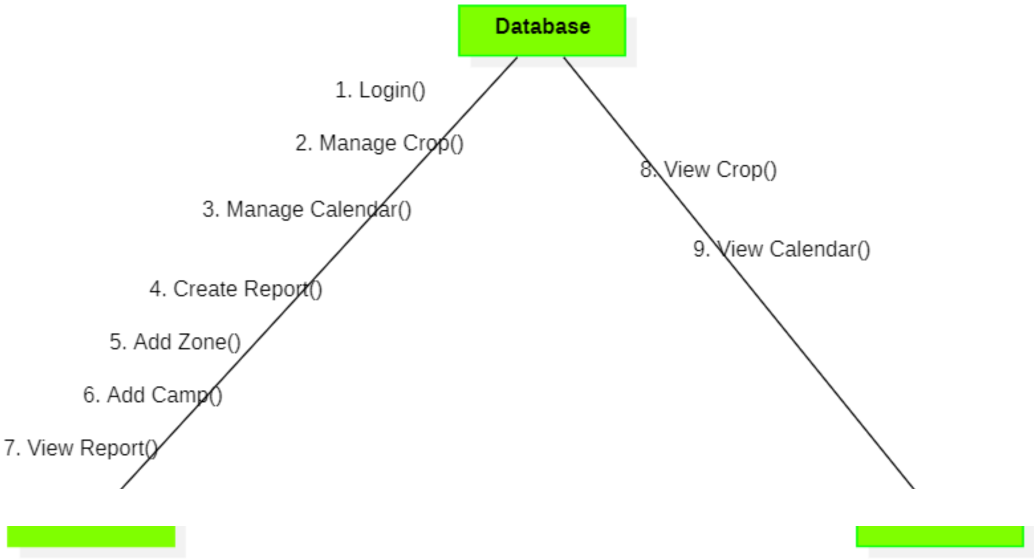


Figure 3. 4: Collaboration diagram for the Crop Management System

- **Activity Diagram**

A UML activity diagram provides a detailed visualization of a use case, similar to a flowchart but with specialized symbols. It models system behaviours clearly, helping in understanding and analysis, especially in software development. This visual tool simplifies complex systems for both technical and non-technical audiences, helping to depict business processes, workflows, and event flows, while also provides support in identifying process flow and requirements. Figure 3.5 below shows the activity diagram for the system administrator for the crop management system [72].

In view of the explanation above find below figure 3.5 an activity diagram for the Crop Management System.

Activity Diagram for user Login Process

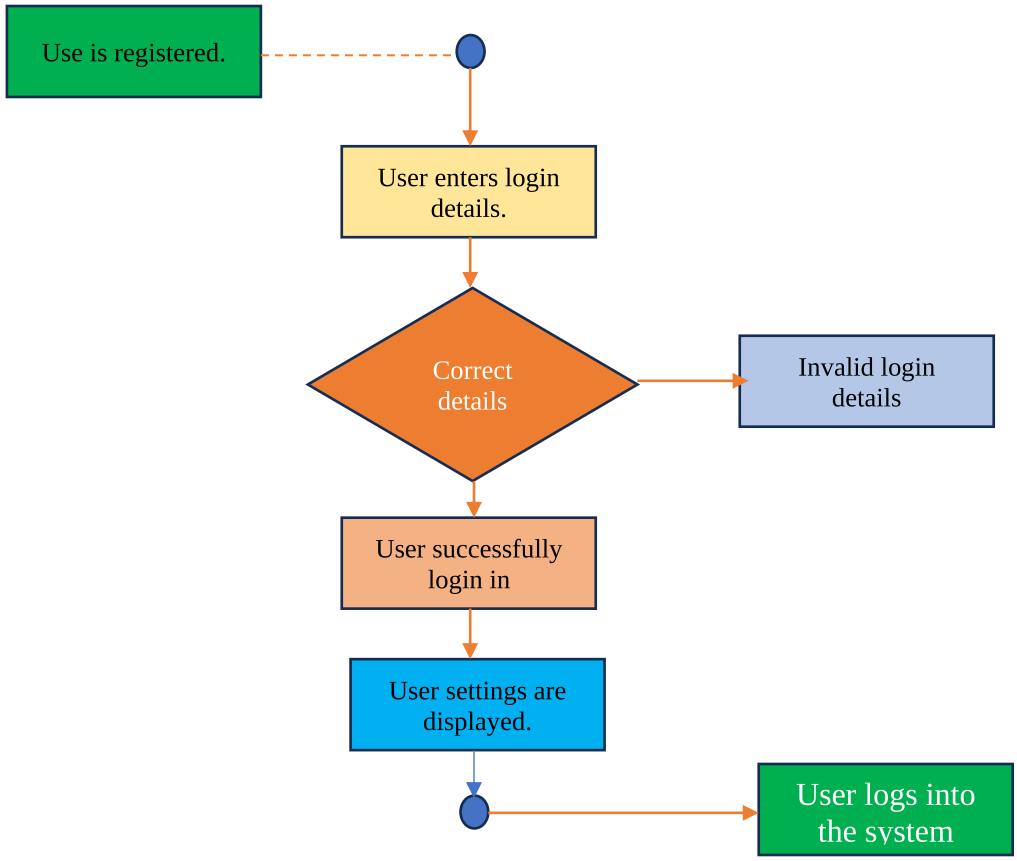


Figure 3. 5: Activity diagram for admin login process

3.3.2 Research Design

Research design is a framework or a plan that guides in the collection and analysis of data. It works as a blueprint for the collection measurement and analysis of data. It is like a map that is usually developed to guide researchers in software development [73].

Figure 3.6 below shows a workflow that was used as a guide to develop the Crop management System

Figure 3.6 shows the framework or a plan for the development of the Crop Management System

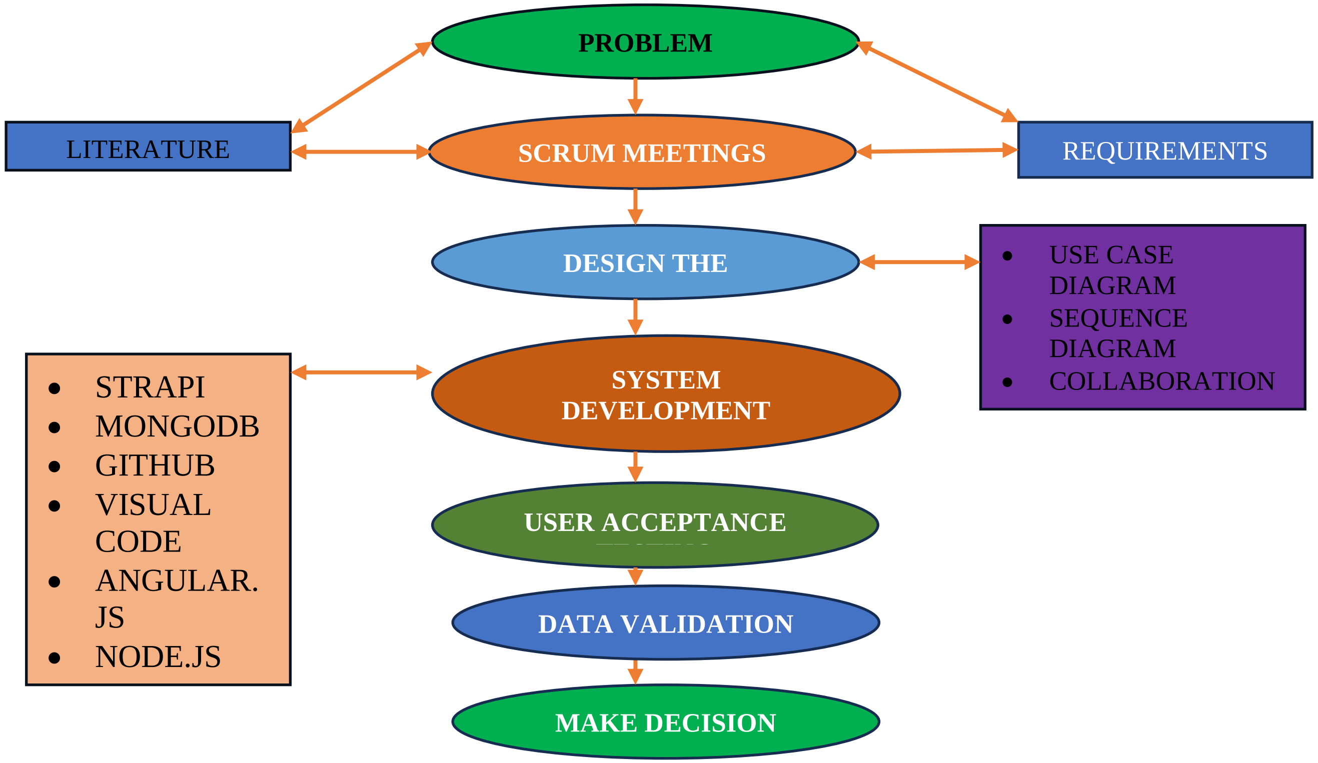


Figure 3. 6: Crop Management System workflow diagram

(i) Conceptual Model

Conceptual data modelling is a fundamental process that involves a series of key activities aimed at bridging the gap between our understanding of the real world and the elements of a data model. It serves as the initial stage in the data modelling process, where the developer has to conceptualise and define the essential components and relationships within the domain of interest [74]. The entities and their relationships are identified during the conceptual design stage.

The diagram below in figure 3.7 describes the conceptual model for the Crop Management System. An entity is a collection of objects sharing identical attributes, recognised by either a user or an organisation as possessing autonomous existence [65].

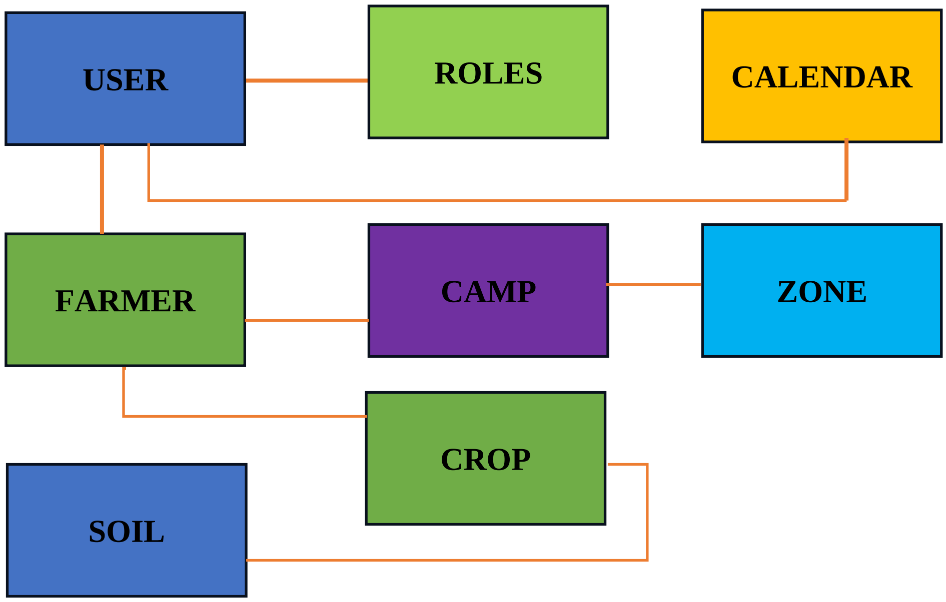


Figure 3. 7: Conceptual Diagram for the Crop Management System

(ii) Logical Design

Logical design encompasses an abstract representation of the system's data flow, inputs, and outputs. It specifies the sources of inputs, destinations of outputs, data stores, and data flows to align with user requirements [75]. Some of the logical diagram that are used to illustrate data flow in a system are as follows: -

- Data flow Diagram (DFD)
- Entity Relationship Diagram (E-RD)
- Class Diagram

(iii) Data flow Diagram

A data flow diagram (DFD) is a diagram that shows how information moves through a system. DFDs help in improving efficiency, identifying possible issues, and creating better processes by providing a deeper understanding of system or process activities [76]. Figure 3.8 below shows data low diagram for a Crop and Management System.

Data flow Diagram for the Crop Management System

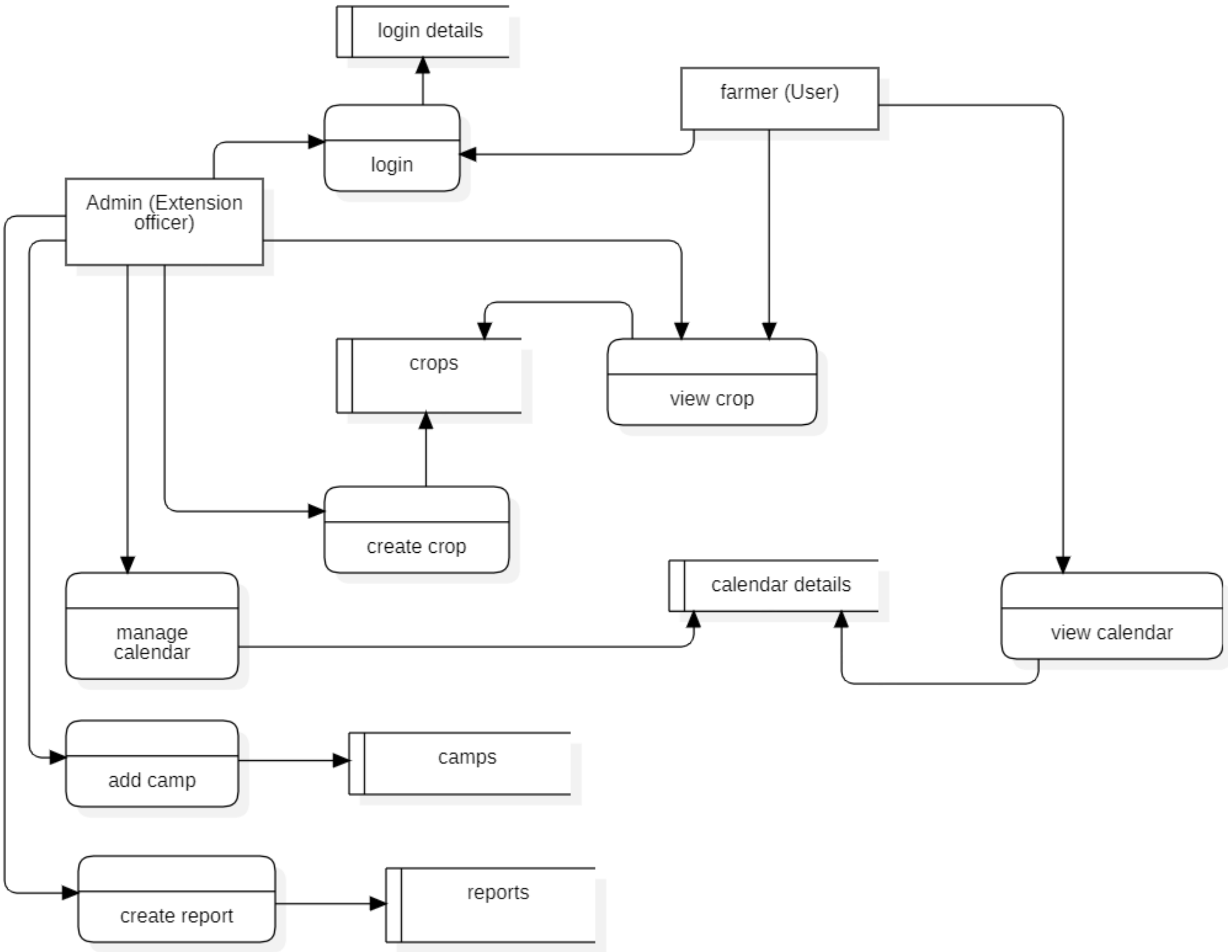


Figure 3. 8: Data Flow Diagram for the Crop Management System

3.3.3 Physical Design

Brad Dayley states that, MongoDB, collections and documents play essential roles in database design and implementation. According Dayley, data is organized into collections, which act similarly to tables in traditional SQL [77].

The following are the collections and documents used during the implementation of the Crop management System.

<u>The User Collection</u>

```
{
  "_id": ObjectId,
  "username": "String",
  "email": "String",
  "password": "String",
  "role": "String",
  "camp": "ObjectId", // Reference to the camp the user belongs to
  "phoneNumber": "String"
}
```

<u>Camp Collection</u>

```
{
  "_id": ObjectId,
  "name": "String",
  "location": "String",
  "description": "String",
  "users": [ObjectId] // References to users who belong to this camp
}
```

<u>Farmer Collection</u>

```
{
  "_id": ObjectId,
  "user_id": ObjectId, // Reference to the corresponding User document
  "name": "String",
  "farm": "String",
  "date_of_birth": "Date",
  "phoneNumber": "String",
  "gender": "String",
  "crops": ["String"],
  "role": "String"
}
```

Zone Collection

```
{
  "_id": ObjectId,
  "camp_id": ObjectId, // Reference to the corresponding Camp document
  "name": "String",
  "location": "String"
}
```

Crop Collection

```
{
  "_id": ObjectId,
  "farmer_id": ObjectId, // Reference to the corresponding Farmer document
  "name": "String",
  "description": "String",
  "process": "String",
  "product": "String"
}
```

Product Collection

```
{
   "_id": ObjectId,
   "crop_id": ObjectId, // Reference to the corresponding Crop document
   "product_name": "String"
}
```

Soil Collection

```
{
  "_id": ObjectId,
  "crop_id": ObjectId, // Reference to the corresponding Crop document
  "name": "String",
  "description": "String",
  "management": "String",
  "image": "String"
}
```

Disease Collection

```
{
  "_id": ObjectId,
  "crop_id": ObjectId, // Reference to the corresponding Crop document
  "name": "String",
  "description": "String",
  "monitor": "String",
  "control": "String"
```

}

Explanation for relationships between collections (Entities)

User Collection:

- Contains information about users. Each user has a unique
- _id, along with their username, email, password, role, phoneNumber, and a
- reference (camp) to the camp they belong to. The camp field holds the
- ObjectId of the corresponding camp document in the Camp Collection.

Camp Collection:

- Contains information about camps. Each camp has a unique _id, along with
- its name, location, description, and an array of users which holds the
- ObjectIds of users who belong to this camp.

Farmer Collection

- _id: A unique identifier for each farmer document.
- user_id: ObjectId referencing the corresponding User document. This establishes a relationship between the Farmer and User collections.
- name: The name of the farmer.
- farm: The name or identifier of the farm the farmer works on.
- date_of_birth: Date of birth of the farmer.
- phoneNumber: Contact number of the farmer.
- gender: Gender of the farmer.
- crops: An array of strings representing the crops the farmer cultivates.
- role: Role of the farmer, if applicable (e.g., manager, owner, worker).

Crop Collection

- _id: A unique identifier for each crop document.
- farmer_id: ObjectId referencing the corresponding Farmer document. This establishes a relationship between the Crop and Farmer collections.
- name: The name of the crop.
- description: Description or details about the crop.
- process: Information about the cultivation process or techniques used for the crop.
- product: Information about the product obtained from the crop (e.g., fruits, grains, etc.).

Disease Collection

- _id: A unique identifier for each disease document.
- crop_id: ObjectId referencing the corresponding Crop document. This establishes a relationship between the Disease and Crop collections.
- name: The name of the disease.

- description: Description or details about the disease.
- monitor: Information about monitoring techniques or strategies for the disease.
- control: Information about disease control measures or strategies

Soil Collection

- _id: A unique identifier for each soil document.
- crop_id: ObjectId referencing the corresponding Crop document. This establishes a relationship between the Soil and Crop collections.
- name: The name or type of soil.
- description: Description or details about the soil.
- management: Information about soil management practices or techniques.
- image: URL or path to an image representing the soil type (optional).

Zone Collection

- _id: A unique identifier for each zone document.
- camp_id: ObjectId referencing the corresponding Camp document. This establishes a relationship between the Zone and Camp collections.
- name: The name of the zone.
- location: Location or description of the zone.

3.3.4 Software and Hardware System Requirement

According to the expert from Computer Hope [78], system requirements refer to the necessary specifications a device must possess to operate certain hardware or software. This could entail a specific I/O port for a computer to function with a peripheral device or a particular operating system for a smartphone to run a specific application. These system requirements are software and hardware requirements. Table 3.2 and Table 3.3 below shows the system requirements (Desktop/Laptop) for the Crop Management System.

Table 3. 2: Hardware System Requirements

S/N	Hardware	Specifications
1	PC	➢ Edition Windows 10 Enterprise or better ➢ Version 21H2 or Better ➢ OS build19044.2364 ➢ RAM 8GB ➢ System type 64-bit operating system, x64-based processor

Table 3. 3: Software Requirements

S/N	Software	Specification
1	Visual Studio	Version 1.83 or above
2	Github	Version 3.3.3 or above
3	MongoDB	Version 4.4
4	Microsoft package	2019 or better
5	Strapi	Version 3
6	Node.js	Version 14.1
7	Angular.js	Version 7.2

3.3.5 Summary

Chapter 3 looks at system development, particularly focusing on developing a Crop Management System. It covers the importance of methodologies in project planning, execution, and validation, ensuring efficiency and meeting deadlines and budgets. The chapter introduces SDLC phases such as defining, designing, testing, and implementing software, discussing various development approaches. It explains the process of gathering system requirements through methods like interviews and diagrams, using various diagrams to understand and document requirements. The chapter also discusses research design as a framework for data collection and analysis, illustrating the research design for the Crop Management System with a workflow diagram. It outlines the stages of conceptual, logical, and physical design, defining components, data flow, and implementation details like database design. Lastly, it discusses the necessary software and hardware requirements for system implementation, including tools like Visual Studio and MongoDB. Overall, Chapter 3 provides a comprehensive overview of system development methodologies, processes, and tools, using the Crop Management System as an example.

Chapter 4: Results and Discussion

4.0 Introduction

This chapter is the heart of the study, where we analyse the data and talk about what we found from after executing the Crop Management System. We'll look at the results and discuss them in relation to what we set out to achieve in the study.

4.1 Results

The following are the results that were generated both from the backend and the frontend.

Frontend and Backend Implementation results

The software described in chapter 2.6.2 were used to design and implement both backend and the frontend of the system.

Figure 4.1 below describes the process of executing both the frontend and backend of the Crop Management System. The code was executed in visual code editor.

Step 1. Frontend

- The first step is to locate the project on the local Computer and open using Visual Studio Code editor. The project is opened it will appear as shown in figure 4.1.

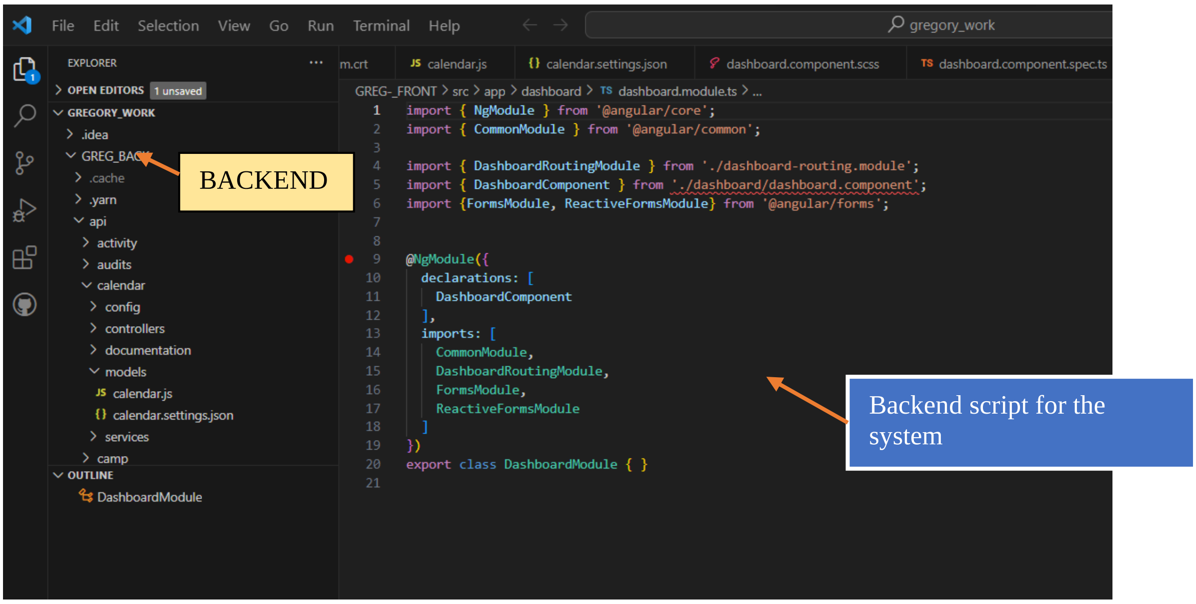


Figure 4. 1: Backend

The project has both the backend and Frontend which are named after the author, GREG BACK and GREG FRONT as can be seen in the figure 4.1 above. These two are opened by right clicking either GREG_FRONT or GREG_BACK using integrated Terminal and run the following the ‘npm run start’ command.

Step 2 Code Execution

After running the ‘npm run start’ command for the frontend and backend as can be seen in figure 4.2 and figure 4.3 below, the links are generated for both frontend and backend.

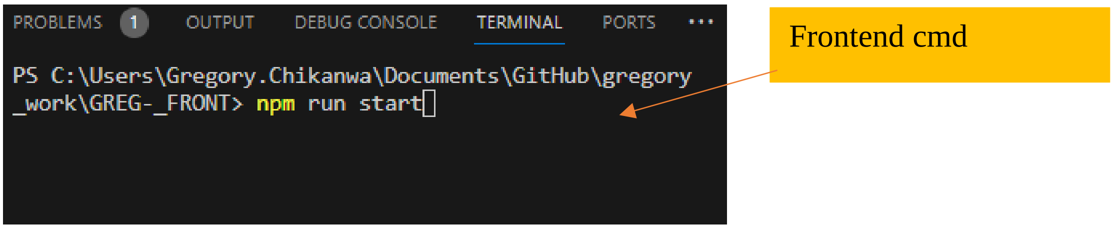


Figure 4. 2: Frontend Execution Code

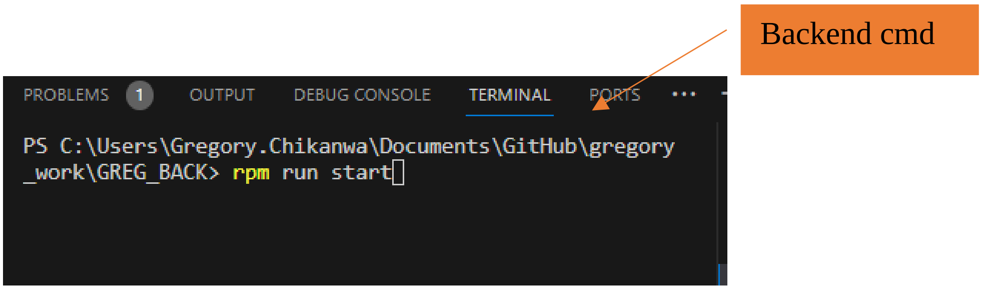


Figure 4. 3: Backend Execution Code

The generated links for the backend is **http://localhost:1338/admin** and **http://localhost:1338**. The links allows the system administrator to login on the the backend and able to configure the system accessibility by different users according to their user roles.

Figure 4.4 below shows the link to the backend for the system administrator. When this link is clicked it will automatically open the login page for the backend of the system.

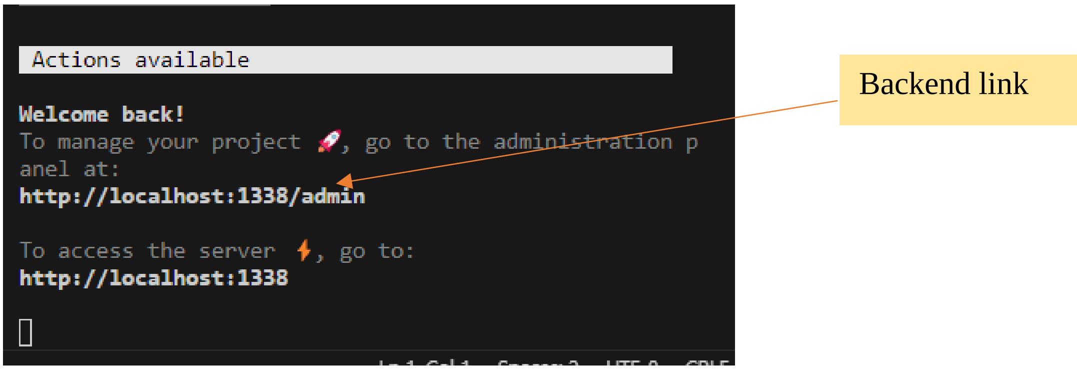


Figure 4. 4: Backend URL link

Furthermore, when the link http://localhost:4200/ ** shown below in figure 4.5 is clicked, it opens the frontend of the system for the users according to their roles.

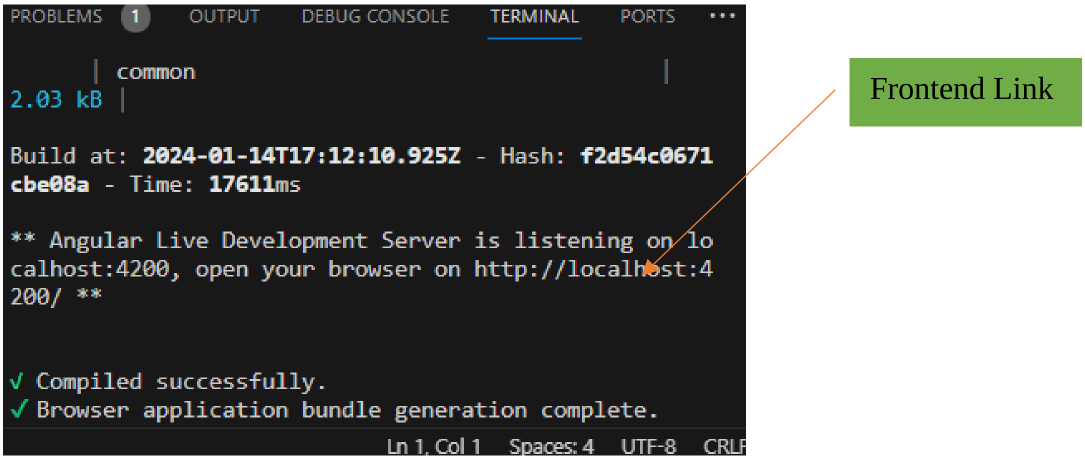


Figure 4. 5: Frontend URL link

The login page shown below in figure 4.6 is for the backend accessed by the system administrator for the Crop Management System. It allows the system administrator to login, configure the system according to roles and can register other super users to edit and upload data for the farmers to access.

the figure below shows the login page for system administrators.

The login page is only for system administrators allowed to upload data on the system Farmers access the system through the web.

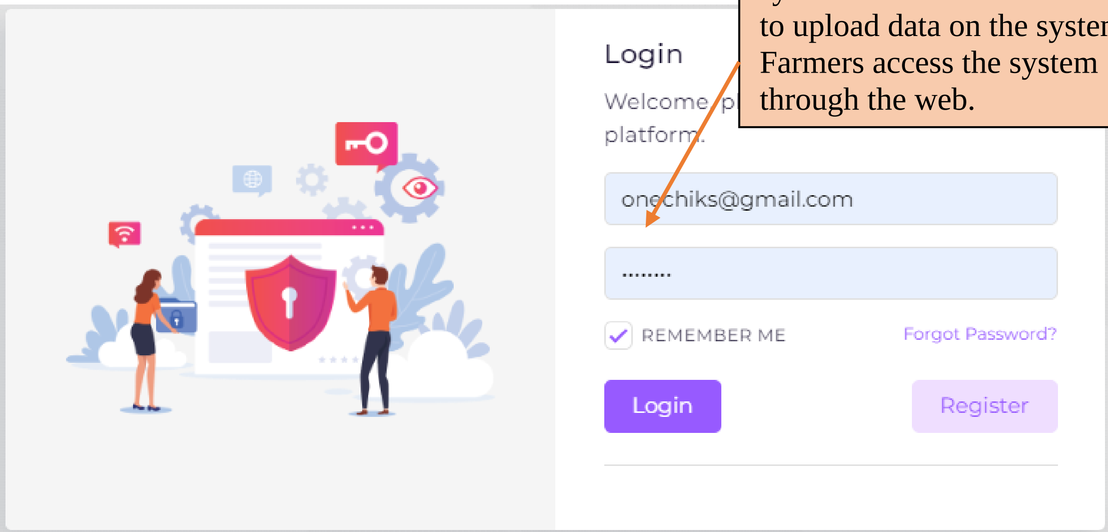


Figure 4. 6: Login interface for the frontend

Figure 4.7 below shows the frontend for other users with different roles ranging from editing, viewing and system configuration. These users are also able to upolad data for the farmers to access through the url of the system. This is necessited by the content management system running on the backend of the system powered by strapi.

The system administrator can add crops, add calendar, add Zones and Camps these statistical data can be viewed through the system dashboard.

Figure4.7 below shows the frontend of the system as described above.

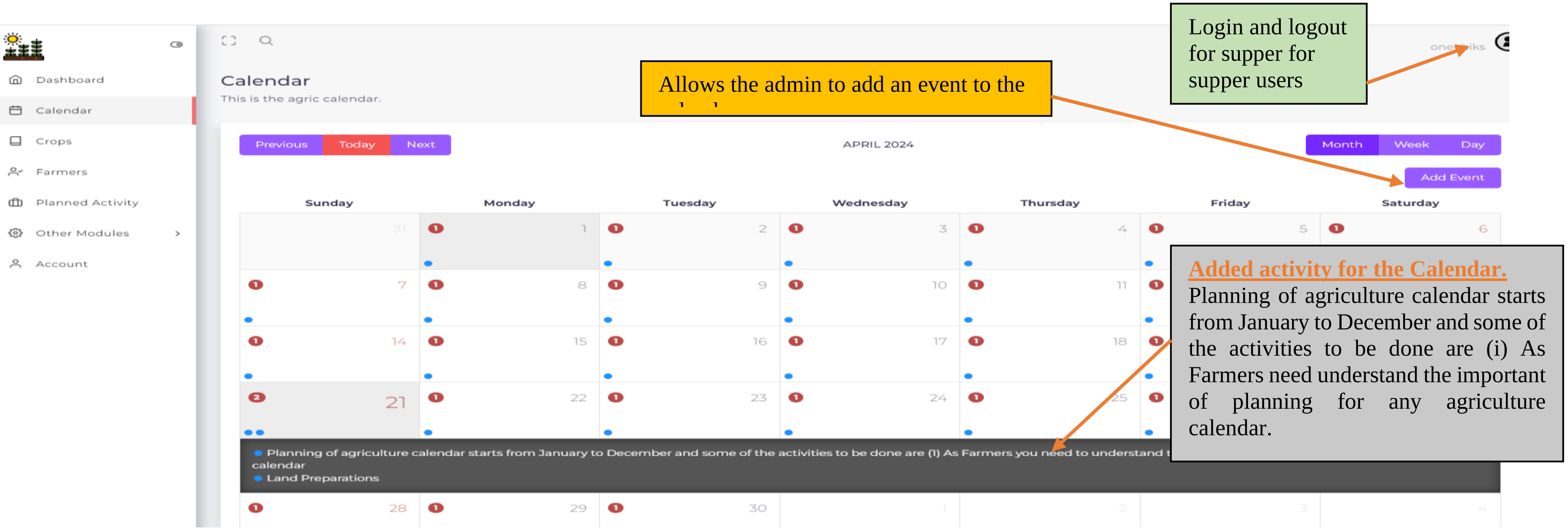


Figure 4. 7: Frontend interface for the Crop management System

Syeda Famita Amber [79]describes a dashboard as a central hub of information. It shows real-time data in easy-to-understand pictures to help business people make smart decisions based on facts. It pulls together lots of different data sources and turns them into clear graphs and charts. Dashboards are great for showing important performance stats and helping teams keep track of how they're doing towards their goals. The dasboard for the crop management system is as shown below in figure 4.8 it shows all the statistical data in the system that has been uploaded into the system in order to make proper decisions on the system data.

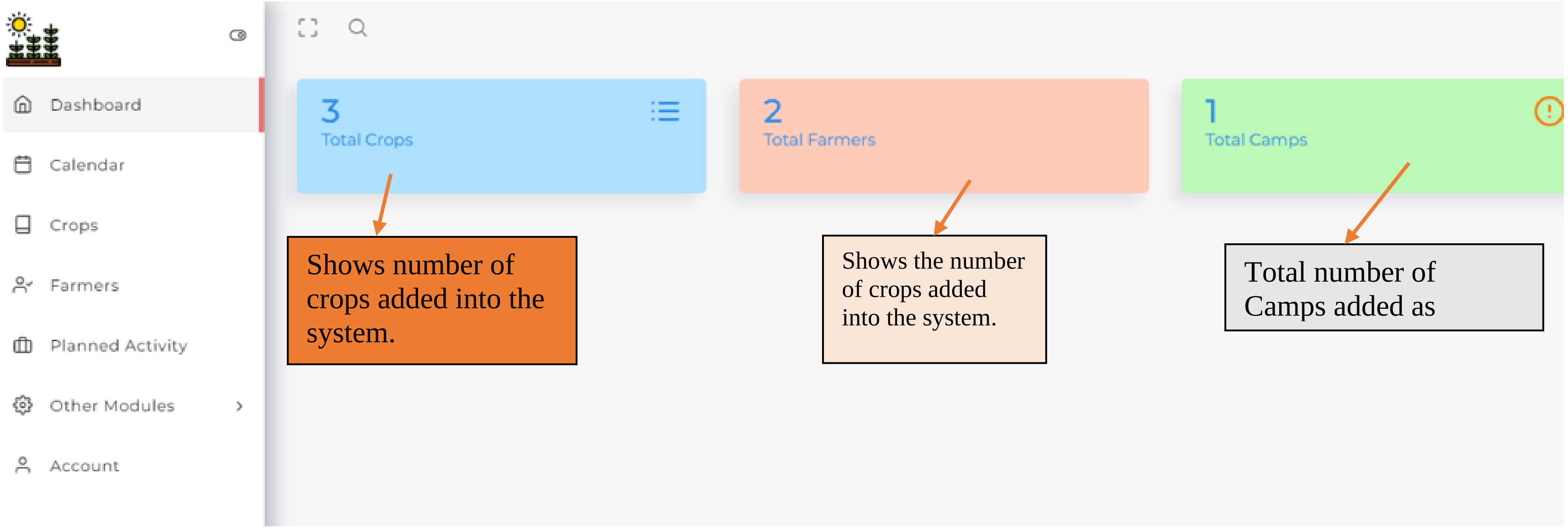


Figure 4. 8: Dashboard for the Crop management System

The diagram below shows in figure 4.9 shows the data that is accessed by users (Farmers/admin) depending on the user roles through the frontend of crop management system showing collections with its documents.

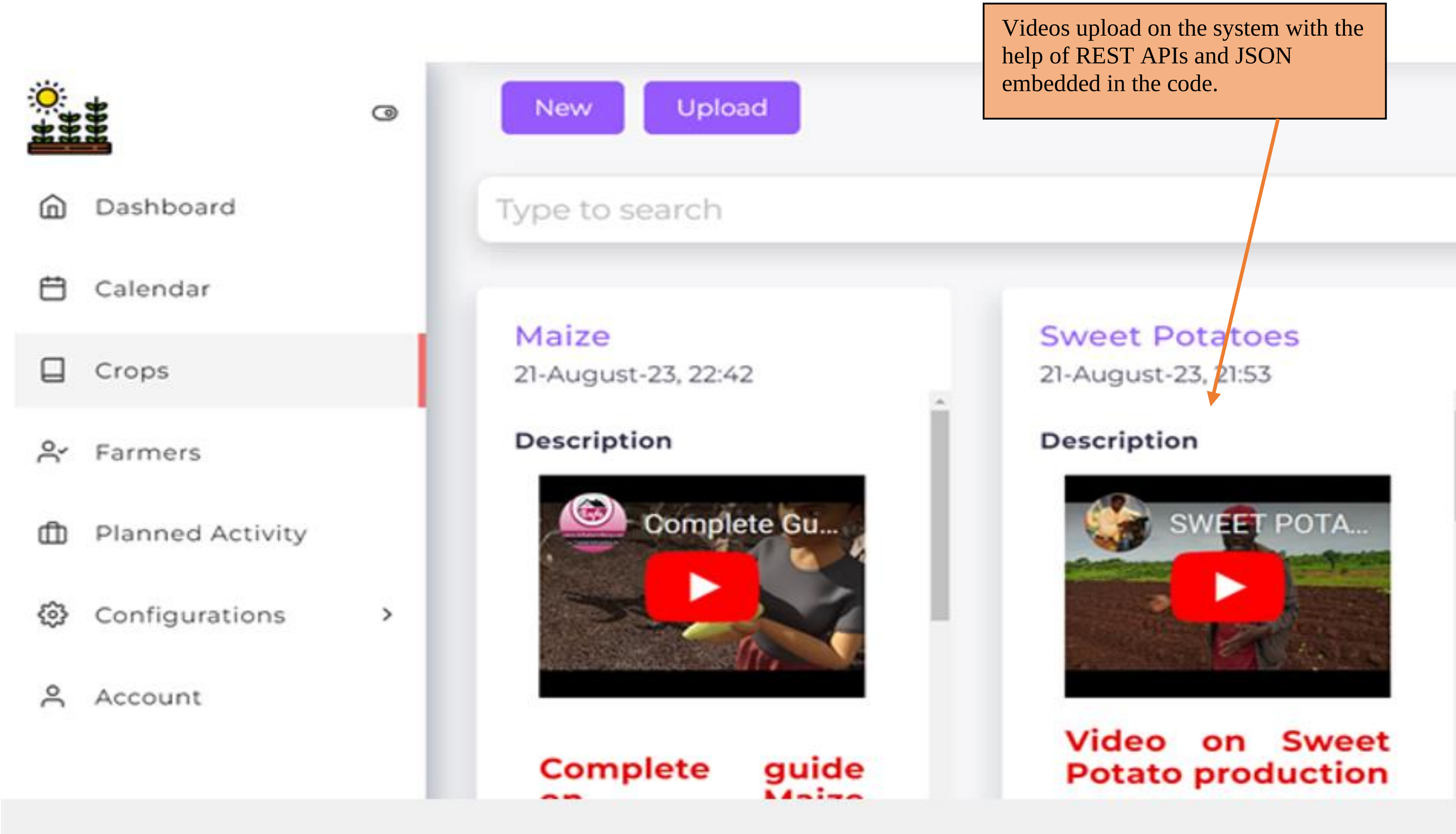


Figure 4. 9: Frontend showing Crops uploaded on the System

Strapi, a headless CMS, powers websites and apps with its back end and admin panel. The back end, functioning as an HTTP server, manages requests and interacts with databases for content management. The admin panel offers a graphical interface for content management. The User Guide explains admin panel usage, and customization options are detailed separately. Strapi's back end, powered by Koa, operates as an HTTP server, enabling data manipulation via REST or GraphQL APIs. [80]. Figure 4.10 below show the backend and frontend for strapi CMS.

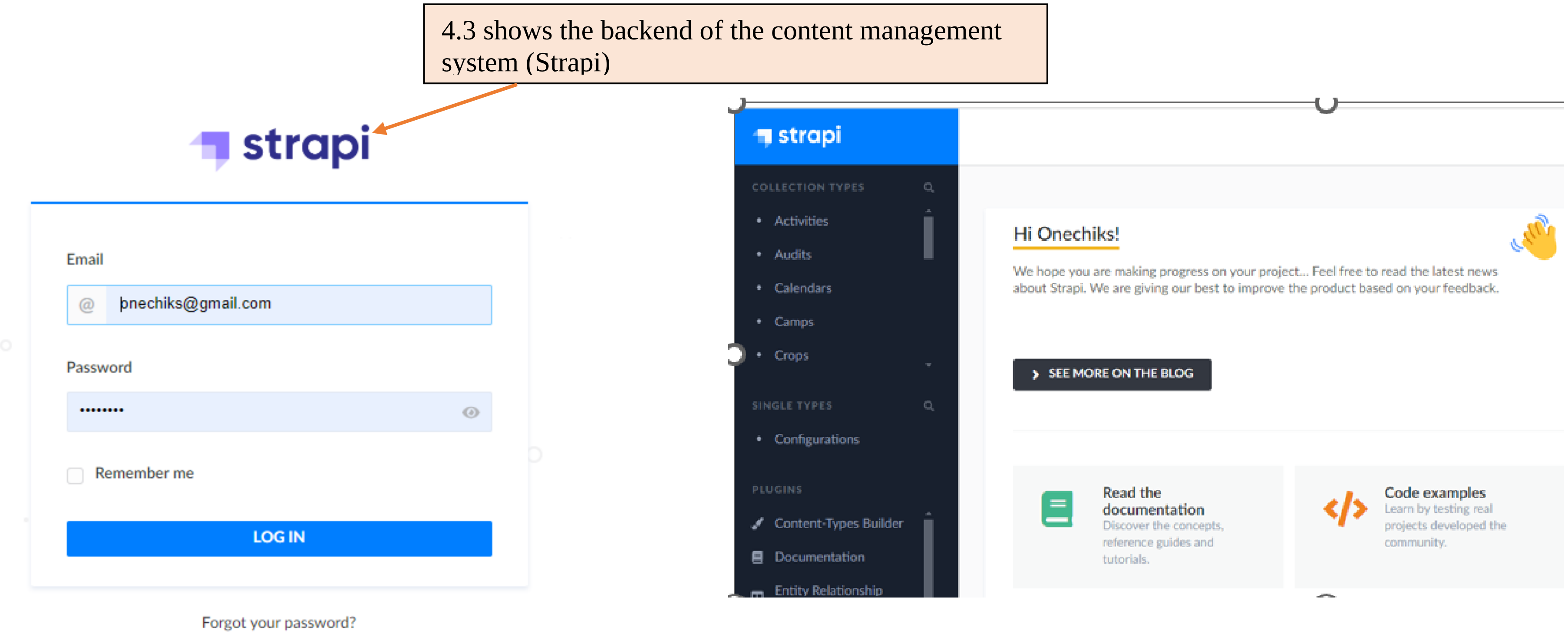


Figure 4. 10: Shows the login and login success for the backend administrator

Figure 4.11 and figure 4.12 shows the process that is followed by the admin to add data on to the system.

Adding a Videos to the application.

- Run the application and selecting a collection for example Crop.
- Click the collection and add new document.

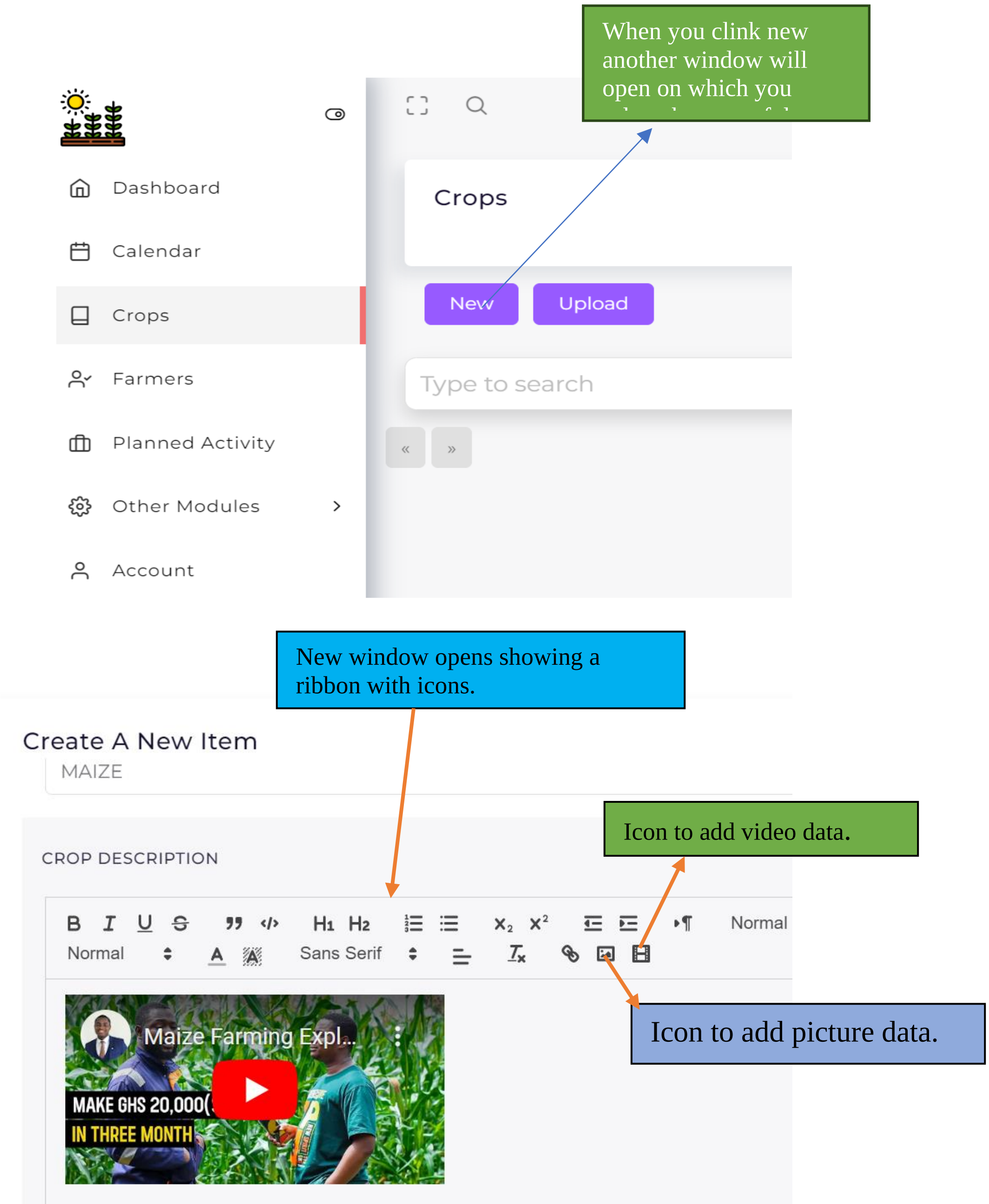
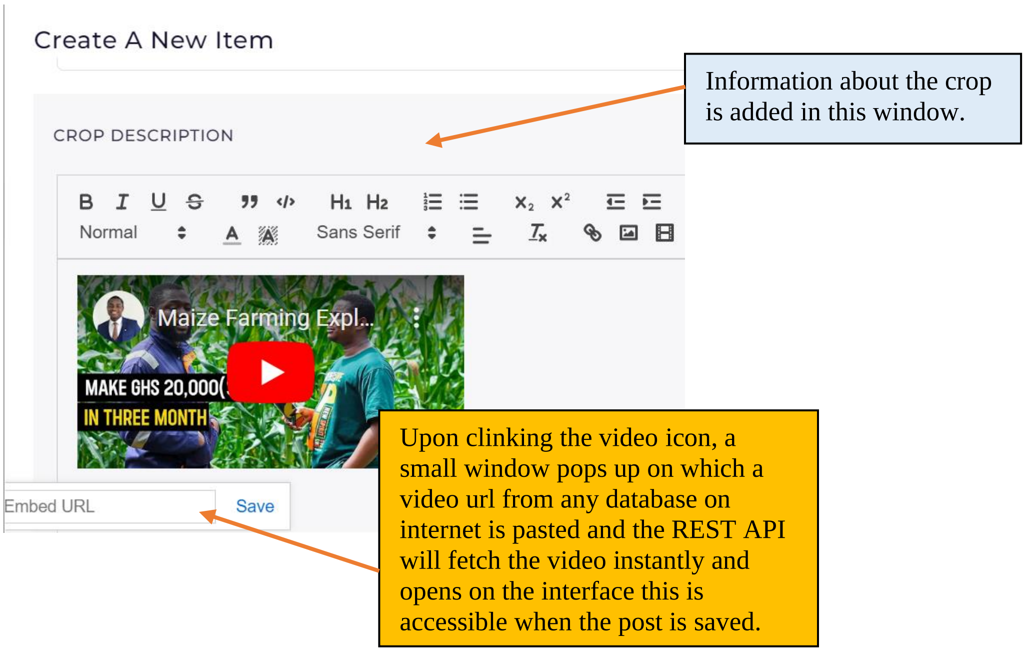


Figure 4. 11: showing the process of adding new material to the system



The figure below shows before and after adding another video tutorial.

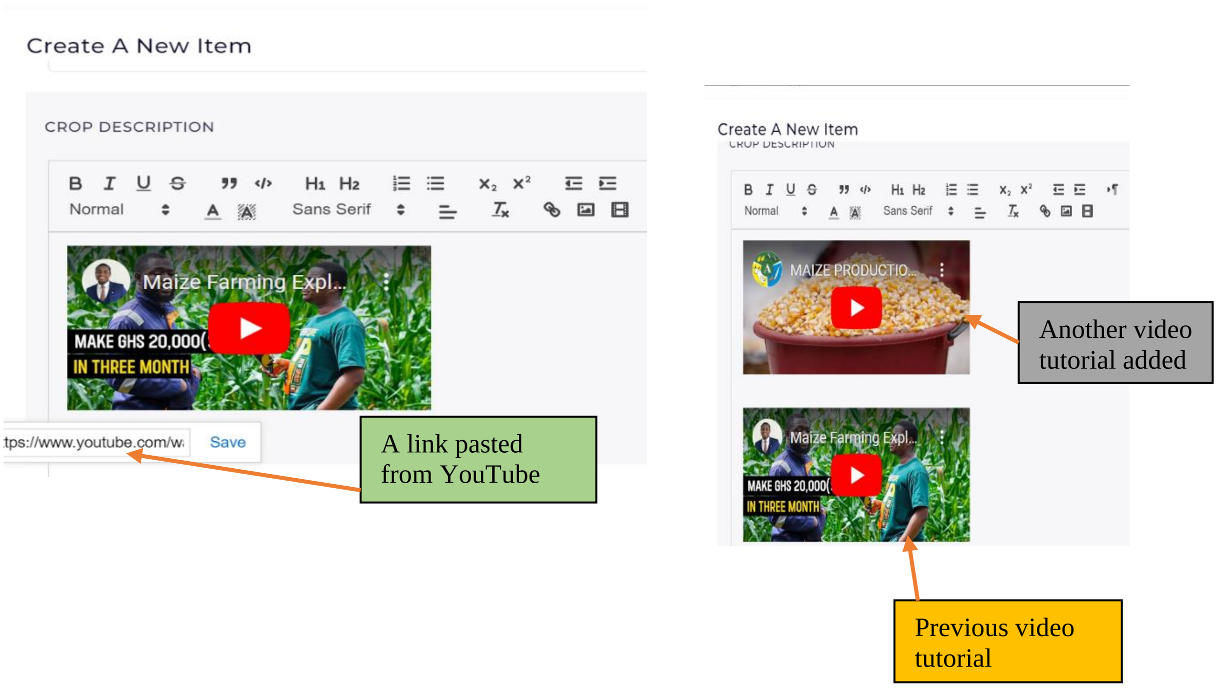


Figure 4. 12: shows a second video tutorial added by the system administrator.

This process happens seamlessly with the help of the REST API and JSON file embedded on the backend code of the system. This is done during the programming stage.

4.2 Discussion

In Chongwe district, the current crop production and management system primarily rely on traditional methods supplemented by limited ICT adoption. While Information and Communication Technology (ICT) has seen widespread use across various sectors in Zambia, its integration into agriculture remains constrained, particularly among rural farmers.

The purpose of the study was to design and implement a Crop Management System for farmers in Chongwe District of Zambia.

The Ministry of Agriculture's e-Extension department is leveraging ICT by utilizing the Ministry's website to deliver e-extension services to farmers in Chongwe. However, this system has limitations, notably the inability to provide video content, restricting it to text and picture formats. Despite this constraint, the web-based application developed by the ministry is accessible via both mobile smartphones and computers, aiming to enhance accessibility and reach among farmers in remote areas.

Given the current context in Chongwe district, where traditional methods still dominate crop production and management, and ICT adoption is limited dual to users not accepting the use of the current e-Extension services being provided by the e-Extension portal under the ministry of Agriculture website because of its limitation on data type. To address this problem a Crop management system was designed with the following attributes: -

User-Friendly Interface: it has a user-friendly interface that is intuitive and easy to navigate, especially for farmers who may have limited experience with technology. This will ensure that the system is accessible to a wide range of users with mobile smart phones.

Multimedia Integration: The developed system has addressed the limitation of the existing system by incorporating multimedia capabilities, including the ability to provide video content alongside text and pictures. This will enhance the effectiveness of information delivery and improve farmers' understanding of crop management practices.

Mobile Accessibility: The new system is compatible with mobile smartphones, as this is a common means of accessing information, particularly in rural areas where access to computers may be limited.

Tailored Content: Crop management system provide content that is tailored to the specific needs and challenges faced by farmers in Chongwe district. This may include information on locally relevant crops, climate-specific recommendations, pest and disease management strategies, and soil health management techniques.

Research limitations.

Though the developed system has provided the solution and reduced the distance required by Ministry Agriculture extension officers the following are the limitations.

- The research might to have provided solutions for the farmers who may not have smart phones as this require using a phone that is able access internet connectivity.
- Farmers may not be able to chat with the extension officers in real time but depend only on the activity posted by the extension officers in each month.

Future Research

1. Exploring alternative methods to reach farmers without smartphones, like SMS-based or voice-based systems.
2. Developing real-time communication channels between farmers and extension officers within the Crop Management System.
3. Understanding barriers to technology adoption among farmers and devising strategies to promote system uptake.
4. Assessing the impact of the system on crop productivity, income, and resilience to challenges like climate change.
5. Exploring opportunities to scale up the system to other regions and ensure its sustainability through partnerships.
6. Continuously evaluating and improving the system based on user feedback to meet evolving farmer needs.

Chapter 5: Recommendations and Conclusion

5.0 Introduction

This chapter looks at the recommendations based on the on the limitations that the system is likely to face and the conclusion of the research.

5.1 Recommendations

To address the limitations identified in the research above, the following recommendations have been proposed:

Offline Accessibility Solutions: Develop alternative methods for farmers without smartphones to access essential information. This could involve implementing SMS-based services or utilizing community centers equipped with internet-enabled devices where farmers can access the platform.

Real-Time Interaction Channels: Introduce additional channels for real-time interaction between farmers and extension officers. This could include integrating instant messaging features within the platform, allowing farmers to ask questions and receive immediate assistance, supplementing the monthly postings.

By implementing these recommendations, the system can enhance accessibility and foster more dynamic engagement between farmers and extension officers, thereby maximizing the impact of agricultural extension services.

5.2 Conclusion

In conclusion, this research has shed light on the current state of crop production and management systems in Chongwe District, Zambia, emphasizing the reliance on traditional methods alongside limited adoption of Information and Communication Technology (ICT).

The primary aim of this study was to design and implement a Crop Management System tailored to the needs of farmers in the district. Despite the Ministry of Agriculture's efforts to leverage ICT through their e-Extension services, significant limitations were identified, notably the lack of video content and real-time communication channels.

To address these challenges, a new Crop Management System was developed with several key attributes. It boasts a user-friendly interface, multimedia integration, mobile accessibility, and tailored content to enhance its effectiveness and accessibility among farmers.

However, it's important to acknowledge the limitations of this research. Not all farmers may have access to smartphones, limiting their ability to benefit from the system. Additionally, the lack of real-time communication with extension officers poses a constraint on the system's utility.

Moving forward, future research should focus on overcoming these limitations by exploring alternative methods for reaching farmers without smartphones and establishing real-time communication channels. By addressing these challenges, we can ensure the widespread adoption and effectiveness of ICT-enabled solutions in agricultural development, ultimately contributing to improved crop productivity and livelihoods in Chongwe District and beyond.

References

[1] N. Dominic, Agriculture Methodologies, Lusaka: Ministry of Agriculture, 2022.

[2] Chapwa Kasoma, Hussein Shimelis & Mark D. Laing, Fall armyworm invasion in Africa: implications for maize production and breeding, vol. 35, Journal of Crop Improvement, 2019.

[3] National Assembly of Zambia, COMMITTEE ON MEDIA, INFORMATION AND COMMUNICATION, LUSAKA: National Assembly of Zambia, 2017.

[4] N. Petrllis, A Smart Phone Image Processing Application for Plant Disease Diagnosis, Tripoli, 2018.

[5] A. Chapoto and A. Kuteya, 2022 Agriculture Fisheries, & Livestock Budget Analysis, Lusaka: INDABA AGRICULTURAL POLICY RESEARCH INSTITUTE, 2021.

[6] WRITER and STAFF, DATA SHOWS EXTENSION SERVICES THE MOST NEGLECTED IN AGRICULTURE, Lusaka: Nakanday: Center for investigative journalism, 2020.

[7] Somanje, A.N., Mohan, G. & Saito, Evaluating farmers' perception toward the effectiveness of agricultural extension services in Ghana and Zambia, Agriculture & Food Security , 2021.

[8] Hichaambwa, C. Matenga and Munguzwe, A MULTI-PHASE ASSESSMENT OF THE EFFECTS OF COVID-19 ON FOOD SYSTEMS AND RURAL LIVELIHOODS IN ZAMBIA: APRA COVID-19 Country Report, LUSAA: APRA, 2021.

[9] Khushboo Yadav, Lauvdya Bangari,Preethi Manukonda,Amtul Waris, Role of Agricultural Extension in Knowledge Transfer, Stella International Publication, 2023.

[10] M. N. Wulystan P. Mtega, Strengthening the Flow of Agricultural Knowledge among Agricultural Stakeholders: The Case of Morogoro Region in Tanzania, Morogoro, 2018.

[11] Ross Flett, Fiona M Alpass, Steve Humphries, Claire Massey, The technology acceptance model and use of technology in New Zealand dairy farming, New Zealand: Agriculture Systems, 2004.

[12] Petros Chavula, Belay Teressa,Marie Grace Ntezimana,Yusuf Umer, An overview of Zambia's Agricultural Extension and Advisory System, vol. 6, Zambia: International Journal of Academic and Applied Research (IJAAR), 2022.

[13] Pallavi, D.T. Santosh,Ashoka. N, E-Extension for Agriculture Development: ICT Tools, Implementation, and Impacts, New Delhi: Dilpreet Publishing House, 2023.

[14] Food and Agriculture Organisationof the united nations, USE OF ICT IN AGRICULTURE AND ENHANCING EXTENSION, Lusaka, 2020.

[15] National Agricultural Information Services call centre, GOVT ROLLS OUT E-EXTENSION TO BOOST AGRICULTURAL PRODUCTIVITY, LUSAKA.

[16] S. BEKOE, Assessing the usefulness of the e-Agriculture programme in Ghana using the Technology Acceptance Model, Accra: Ghana Jnl Agric. Sci, 2023.

[17] J. Jonker and B. Pennink, The Essence of Research Methodology: A Concise Guide for Master and Ph.D Students, Heidelberg: Springer, 2010.

[18] Chhetry, Gopal Kumar Niroula; Belbahri, Lassaad, Indigenous pest and disease management practices in traditional farming systems in northeast India. A review, vol. 1(3), Imphal: Academic Journals, 2009.

[19] Anthony Baffoe-Bonnie, David Tanner Martin , Frank Mrema, Agricultural extension and advisory services strategies during COVID-19 lockdown, H. Zhang, Ed., Mississippi: Lorman, 2021.

[20] Smart Zambia Institute, www.agriculture.gov.zm, LUSAKA, LUSAKA, 2022.

[21] Ministry of Agriculture, Smart Zambia, E-Extension, Lusaka: Minstry of Agriculture, Smart Zambia, 2022.

[22] E. S. Yayan Apriyana, The Integrated Cropping Calendar Information System: A Coping Mechanism to Climate Variability for Sustainable Agriculture in Indonesia, Indonesia, 2021.

[23] Thomas Kipkurgat, Michael Onyiego, Silahs Chemwaina, IMPACT OF SOCIAL MEDIA ON AGRICULTURAL EXTENSION IN KENYA: A CASE OF KESSES DISTRICT, vol. 3, UK: European Centre for Research Training and Development, 2016.

[24] S. D. Rachana P.Koli, Android Application Agriculture Decision Support System, vol. Vol. 4, Int. Journal of Engineering Research and Applications , 2014, pp. pp.63-66.

[25] K. Raikar, S. Gawade and V. Turkar, Usability Improvement with Crop Disease Management as a Service, Bhopal: IEEE, 2017.

[26] S. Tafesse, E.Damtew, v. B. Mierlo, R. Lie, B. Lemaga, K. Sharma, C. Leeuwis and P. C. Struika, Farmers' knowledge and practices of potato disease management in Ethiopia, vol. 86–87, Wageningen: NJAS - Wageningen Journal of Life Sciences, 2018.

[27] Mahapatra and S. Kumar, Smartphone Apps for Agri-information dissemination during Covid19 lockdown, Vols. 2(5): 116-119, Odisha: Biotica Research Today, 2020.

[28] resolution, N. Adnan and S. M. N. o. g. f. a. potential, How COVID 19 affect Malaysian paddy industry? Adoption, Malaysia: Springer, 2020.

[29] S. Sharma, V. Patodkar, S. Simant, C. Shah and P. S. Godse, E-Agro Android Application (Integrated Farming Management Systems for sustainable development of farmers), vol. 3, Maharashtra: International Journal of Engineering Research and General Science, 2015.

[30] Prakash, Dr. Om, E-EXTENSION INITIATIVES IN AGRICULTURAL SECTOR, Odisha: S P Publishing, 2023.

[31] E. S. Brindha Gopalakrishnan, Improving agricultural information and extension services to increase small-scale farmer productivity, J-PAL Policy Insights, 2023.

[32] S. Raj, e-Agriculture Prototype for Knowledge Facilitation among Tribal Farmers of North-East India: Innovations, Impact and Lessons, vol. 19, 2023.

[33] Tian Cai, Eric Abbot, Using Video as a Replacement or Complement for the Traditional Lecture/Demonstration Method in Agricultural Training in Rural Uganda, vol. 97, Journal of Applied Communications,, 2013.

[34] E. Zossou, P. Van Mele, S. D. Vodouhe and J. Wanvoeke, Comparing Farmer-to-Farmer Video with Workshops to Train Rural Women in Improved Rice Parboiling in Central Benin, vol. 15, Benin: Journal of Agricultural Education and Extension, 2009.

[35] Ugwuishiwu, Application of ICT in Crop Production, vol. 2, Nsukka: International Journal of Soft Computing and Engineering (IJSCE), 2012.

[36] H. K. Chavula, The role of ICT in Agriculture Production, vol. 6, Addis Ababa: Journal of Development and Agricultural Economics, 2013.

[37] R. J. Atulya Narayan, Performance Measurement Model for Agriculture Extension Services for Sustainable Livelihood of the Farmers: Evidences from India, vol. 9, Shillong: Scientific Research, 2019.

[38] Hussaini, Potential of E-Agriculture in Extension Service Delivery Towards Accelerating Agricultural Productivity and Efficiency in Nigeria: A Review, Zaria: International Conference on Computing and advances in InformationTechnology, 2021.

[39] AKTC, USE OF SMART PHONES IN AGRICULTURE, LUSAKA: ZAMBIA - GERMAN Agricultural Knowledge and Training Center (AKTC), 2021.

[40] Communication Africa, Airtel to upgrade towers and increase sites in Zambia, Alain Charles Publishing Ltd, 2012.

[41] P. K. Banda, A Study on Mobile Penetration Rate in a Multi-Simming Environment: The Case of Zambia, 2016.

[42] Langer, Arthur M., Analysis and Design of Information Systems, Third Edition ed., London: Springer, 2008.

[43] Ould, Martyn A., Business Process Management: A Rigorous Approach, United Kingdom: British Computer Society, 2005.

[44] FAISAL ABURUB, SOUD MOHAMMAD ALMAHAMID, A METHOD FOR DERIVING SYSTEM MODELS BASED ON BUSINESS PROCESS MODELS, vol. 2, JORDAN: Journal of Information Technology Management, 2010.

[45] Pradeep Hewage, Mark Anderson ,Hui Fang, An Agile Farm Management Information System Framework for Precision Agriculture, Barcelona: Association for Computing Machinery, 2017.

[46] Al jafa Hasan, Using Agile in Implementing Agriculture AI Projects and Farm Management, vol. 14, Journal of Agricultural Informatics, 2023.

[47] D, Maria, Popular software Development Methodologies and Frameworks: Full Comparison, Saudi Arabia, 2022.

[48] E. Altynpara, Popular Methodologies Comparison: Full overview, Oslo: Cleverroad, 2023.

[49] Muniba Khudadad, Yasir Hafeez Motla,Sohail Asghar,Syeda Ayesha Anwar,Zahid Iqbal, A scrum based framework for e-agriculture system, Karachi: IEEE, 2014.

[50] Divya Chauhan, K. L. Bansal, Using the Advantages of NOSQL: A Case Study on MongoDB, vol. 5, Shimla: International Journal on Recent and Innovation Trends in Computing and Communication, 2017.

[51] Arora, Rupal ,Rinkle Rani Aggarwal, An Algorithm for Transformation of Data from MySQL to NoSQL (MongoDB), vol. 2, International Journal on Recent and Innovation Trends in Computing and Communication, 2013.

[52] A. M. Sasidharan, Which is the better option, when it comes to building highly responsive web applications that need to process very large number of HTTP request per second, AngularJS or jQuery?, Dublin, 2016.

[53] K. Lei, Performance Comparison and Evaluation of Web Development Technologies in PHP, Python and Node.js, Chengdu, China: Institute of Electrical and Electronics Engineers, 2014.

[54] S. L. Bangare, S. Gupta,M. Dalal,A. Inamdar, Using Node.Js to Build High Speed and Scalable Backend Database Server, International Journal of Research in Advent Technology, 2016.

[55] Hudson Borges, Andre Hora, Marco Tulio Valente, Understanding the Factors that Impact the Popularity of GitHub Repositories, 2016 IEEE International Conference on Software Maintenance and Evolution, 2016.

[56] Gerard Lommerse, Freek Nossin,Lucian Voinea,Alexandru Telea, The Visual Code Navigator: An Interactive Toolset for Source Code Investigation, IEEE Symposium on Information Visualization 2005, USA.

[57] Adservio, Introduction to Strapi.

[58] 3schools.

[59] Tree Web Solutions.

[60] N. khanal, Build a REST API with Node JS & Express, 2022.

[61] J. W. Satzinger, R. B. Jackson and S. D. Burd, Systems Analysis,and Design In a Changing World, Seventh Edition ed., Boston: Cengage Learning, 2012, p. 4.

[62] T. Tran, What Is a Software Development Methodology? - A Quick Tour to All the Prevalent Types, Ho Chi Minh City: Orient Software Development Corp, 2023.

[63] TatVaSoft, Top 12 Software Development Methodologies, Ahmedabad, 2020.

[64] FARM CREDIT ADMINISTRATION, SYSTEM DEVELOPMENT: Essential Practices for Information Technology, Virginia, 2007.

[65] Thomas M. Connolly, Carolyn E. Begg, Database Solutions: A step-by-step approach to building databases, second, Ed., England: Pearson Education Limited, 2004.

[66] P. B. David Wilkinson, Using Research Instruments: A guide for Researchers, London: RoutledgeFalmer, 2003.

[67] J. B. Karl E. Wiegers, Software Requirements, Third Edition, Washington: Microsoft Press, 2013.

[68] M. Popovic, Communication Protocol Engineering, Second Edition ed., Boca Raton: Taylor & Francis Group,CRC Press, 2018.

[69] D. Bell, UML basics: An introduction to the Unified Modeling Language, Rational Software, 2003.

[70] S. System, ENTERPRISE ARC, 2021.

[71] Petraq J. Papajorgji, Panos M. Pardalos, Software Engineering Techniques Applied to Agricultural Systems: An Object-Oriented and UML Approach, Second Edition ed., D. Du, Ed., New York: Springer, 2014.

[72] Creately, Activity Diagram Tutorial | Easy Guide with Examples, Melbourne, 2022.

[73] Dr. Prabhat Pandey, Dr. Meenu Mishra Pandey, RESEARCH METHODOLOGY: TOOLS AND TECHNIQUES, Romania: BRIDGE CENTER, 2015.

[74] Ljubica Kazi, Biljana Radulovic,Ivana Franja Berkovic,Zoltan Kazi, Integration of conceptual data modeling methods: Higher education experiences, Metalurgia International, 2014.

[75] tutorialpoint, ystem Analysis & Design - System Design.

[76] C. Chi, A Beginner's Guide to Data Flow Diagrams, Canada: HubSpot, 2023.

[77] B. Dayley, NoSQL with MongoDB in 24 Hours, Sams Teach Yourself, Pearson Education, 2014.

[78] Computer Hope, System requirements, Salt Lake City: Computer Hope, 2021.

[79] S. F. Amber, Understanding Data Dashboards: A Comprehensive Guide, SAN FRANCISCO: Hevo, 2021.

[80] Strap Doc v4.

[81] S. Nagashima, FY2017 Ex-Post Evaluation of Technical Cooperation Project "Rural Extension Services Capacity Advancement Project – Through PaViDIA Approach-", Tokyo: INTEM Consulting Inc, 2017.

[82] S. G. G. Santosh G. Karkhile, A Modern Farming Techniques using Android Application,Engineering and Technology, vol. 4, Maharashtra: International Journal of Innovative Research in Science,, 2015, p. 1.

[83] C. R. Kothari, Research Methodology: Methods and Techniques, illustrated, reprint, revised ed., New Delhi: New Age International, 2004.

[84] RAD, Main Types of Application Development Methodologies, Kissflow Low-Code, 2015.

[85] S. E. S. S. TIMOTHY C. LETHBRIDGE, Studying Software Engineers: Data Collection Techniques for Software Field Studies, M. Zelkowitz, Ed., Netherlands: Springer Science + Business Media, Inc, 2005.

[86] Mendix, Model-Driven Development, Boston: Mendix.

[87] Chaudhuri.B.A, Flowchart and Algorithm Basics : The Art of Programming, Boston,: Mercury Learning and Information LLC., 2020.

[88] A. A. Ahmed and G. H. Reddy, A Mobile-Based System for Detecting Plant Leaf Diseases Using Deep Learning, vol. 3, B. Xu, Ed., Texas: AgriEngineering, 2021.

[89] P. Oakley and C. Garforth, Guide to extension training, United Kingdom: Food and Agriculture Organization of the United Nations, 1985.

[90] W. C. Richardson, D. Avondolio, J. Vitale, S. Schrager, M. W. Mitchell and J. Scanlon, Professional Java, Fifth Edition ed., Canada: John Wiley & Sons, 2005.

[91] D. Avison and G. Fitzgerald, Information System Development: methodology, techniques, and tool., New York: McGraw-Hill Companies Inc, 2006.

[92] Abbas and Hadi, An introduction to different types of study design, 2021.

[93] M. Brambilla, J. Cabot and M. Wimmer, Model-Driven Software Engineering in Practice: Synthesis Lectures on Software Engineering, California: Morgan & Claypool, 2012.

[94] S. Vaupel, G. Taentzer, R. Gerlach and M. Guckert, Model-driven development of mobile applications for Android and iOS supporting role-based app variability, Berlin: Springer, 2016.

[95] Felici and Massimo, Use Cases, Edinburgh: School of Informatics, 2011.

[96] K. Bittner and I. Spence, Use Case Modeling, Boston: Pearson Education, Inc., 2003.

[97] Opinaldo and Norlyn, Ultimate Data Flow Diagram Tutorials with Explanation Examples, GitMind, 2021.

[98] Vaati and Esther, What Is the Android SDK and How to Start Using It, Envatotuts, 2020.

[99] Helmers and S. A., Microsoft Visio 2013: Step by Step, M. Yarbrough, Ed., United States of America.: Practice file plus ebook, 2013.

[100] S. Steve, Introducing System designs, Second Edition ed., Cambridge: Blackwell Publishers Inc., 1996.

[101] University of Zambia, REGULATIONS FOR POSTGRADUATE TRAINING:DIRECTORATE OF RESEARCH AND GRADUATE STUDIES, LUSAKA: UNZA, 2015.

[102] T. Cox, What Is Rapid Application Development (RAD), Virginia: Capterra, 2019.

[107] R. Bernhard, Agile Modeling with UML:Code Generation, Testing, Refactoring, Switzerland: Springer International Publishing, 2017.

[108] C. E. B. THOMAS M. CONNOLLY, Database System:A Practical Approach to Design,Implementation, and Management, Fourth Edition ed., England: Pearson Education Limited, 2005.

[109] R. E. Sikha Bagui, Database Design Using Entity Relationship Diagrams, Second Edition ed., New York: CRC Press, 2012.

[110] H. Gomaa, Software Modeling and Design - UML, Use Cases, Patterns, and, New York: cambridge university press, 2011.

[111] L. D. B. Jeffrey L. Whitten, System Analysis and Design Methods, seven edition ed., New York: McGram-Hill/Irwin, 2007.

[112] S. Brown, Software Architecture for Developers;, Leanpub, 2014.

[113] F. T. Ali Mili, Software Testing; Concepts and Operations, New Jersey: John Wiley & Sons, Inc, 2015.

[114] S. P. Adrian W. West, Practical PHP 7, MySQL 8, and MariaDB Website Databases:A Simplified Approach to Developing Database-Driven Websites, Second Edition ed., California: Apress, 2018.

[115] Simplilearn, Why Use MongoDB: What It Is and What Are the Benefits, Atlanta, USA, 2023.

[116] S. 3. Team, The MongoDB Basics: Databases, Collections & Documents, Berlin, Berlin: 3T Software Labs Ltd, 2024.

[117] Ministry of Agriculture. Ministry of Fisheries andLivestock, Extension Messages / Bulletins, LUSAKA, LUSAKA: ministry of Agriculture, Ministry of Fisheries and Livestock, 2020.

APPENDIX 1: SYSTEM CODE

Login page

```
<! --Login Page Starts-->
<section id="login">
 <div class="row auth-height full-height-vh m-0">
 <div class="col-12 d-flex align-items-center justify-content-center">
 <div class="card overflow-hidden">
        <div class="card-content">
          <div class="card-body auth-img">
            <div class="row m-0">
              <div class="col-lg-6 d-none d-lg-flex justify-content-center align-items-center
auth-img-bg p-3">
                <img src="assets/img/gallery/login.png" alt="" class="img-fluid"
width="300" height="230">
              </div>
              <div class="col-lg-6 col-12 px-4 py-3">
                <h4 class="mb-2 card-title">Login</h4>
                <p>Welcome, please login to access the platform.</p>
                <ngb-alert dismissible="false" type="light-danger" class="mb-2"
*ngIf="isLoginFailed">
                  <p class="mb-0">Login failed!</p>
                </ngb-alert>
                <form [formGroup]="loginForm">
                  <div class="form-group">
                    <input type="text" formControlName="username" class="form-control"
placeholder="Username"
                      [ngClass]="{ 'is-invalid': loginFormSubmitted && lf.username.invalid, 'is-
valid': loginFormSubmitted && !lf.username.invalid }"
                      required>
                    <div *ngIf="loginFormSubmitted && (lf.username.invalid ||
lf.username.errors?.required)"
```

```
                    class="help-block mt-1 text-danger"> <i class="ft-alert-circle align-
middle"></i> This is
                    required</div>
                  </div>
                  <div class="form-group">
                    <input type="password" formControlName="password" class="form-
control" placeholder=" Password"
                    [ngClass]="{ 'is-invalid': loginFormSubmitted && lf.password.invalid, 'is-
valid': loginFormSubmitted && !lf.password.invalid }"
                    required>
                    <div *ngIf="loginFormSubmitted && (lf.password.invalid ||
lf.password.errors?.required)"
                    class="help-block mt-1 text-danger"> <i class="ft-alert-circle align-
middle"></i> This is required
                    </div>
                  </div>
                  <div class="d-sm-flex justify-content-between mb-3 font-small-2">
                    <div class="remember-me mb-2 mb-sm-0">
                      <div class="checkbox auth-checkbox">
                        <input type="checkbox" formControlName="rememberMe"
class="form-control" id="rememberMe">
                        <label for="rememberMe"><span class="font-small-2 mb-3 font-weight-
normal">Remember Me</span></label>
                      </div>
                    </div>
                    <a [routerLink]="['/pages/forgotpassword']">Forgot Password?</a>
                  </div>
                  <div class="d-flex justify-content-between flex-sm-row flex-column">
                    <a  class="btn btn-primary" (click)="onSubmit()">Login</a>
                    <a [routerLink]="['/pages/register']" class="btn bg-light-primary mb-2 mb-
sm-0">Register</a>
                  </div>
                </form>
                <hr>
```

```
<!--              <div class="d-flex justify-content-between align-items-center">-->
<!--                <h6 class="text-primary m-0">Or Login With</h6>-->
<!--                <ngx-spinner></ngx-spinner>-->
<!--                <div class="login-options">-->
<!--                <a class="btn btn-sm btn-social-icon btn-facebook mr-1"><span class="fa fa-facebook"></span></a>-->
<!--                  <a class="btn btn-sm btn-social-icon btn-twitter mr-1"><span class="fa fa-twitter"></span></a>-->
<!--                </div>-->
<!--              </div>-->
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
</section>
<!--Login Page Ends-->
```

Calendar Code

Modules

```
import { NgModule } from '@angular/core';
import { CommonModule } from "@angular/common";
import { FormsModule } from '@angular/forms';

import { NgbModalModule, NgbDatepickerModule, NgbTimepickerModule } from '@ng-bootstrap/ng-bootstrap';
import { CalendarModule, DateAdapter } from 'angular-calendar';
import { adapterFactory } from 'angular-calendar/date-adapters/date-fns';
import { CalendarRoutingModule } from "./calendar-routing.module";

import { CalendarsComponent } from "./calendar.component";
import { DateTimePickerComponent } from './date-time-picker.component';
```

```
@NgModule({
    imports: [
        CommonModule,
        CalendarRoutingModule,
        CalendarModule.forRoot({
            provide: DateAdapter,
            useFactory: adapterFactory
          }),
        NgbModalModule,
        NgbDatepickerModule,
        NgbTimepickerModule,
        FormsModule
    ],
    declarations: [
        CalendarsComponent,
        DateTimePickerComponent
    ]
})
export class CalendarsModule { }
```

HTML CODE

```
<!--Calendar Starts-->
<section id="calendar">
  <div class="row">
    <div class="col-sm-12">
      <div class="content-header">Calendar</div>
      <p class="content-sub-header">This is the agric calendar.</p>
    </div>
  </div>
  <ng-template #modalContent let-close="close">
    <div class="modal-header">
      <h5 class="modal-title">{{ modalData?.action }}</h5>
      <button type="button" class="close" (click)="close()">
        <span aria-hidden="true">&times;</span>
      </button>
    </div>
    <div class="modal-body">
```

```
          <form action="#" class="form form-horizontal">
            <div class="form-body">
              <div class="form-group">
                <label>Event Title:</label>
                <input type="text" name="event-title" class="form-control"
[(ngModel)]="modalData?.event.title"
                  (keyup)="refresh.next()">
              </div>
              <div class="row">
                <div class="col-lg-6 col-12">
                  <div class="form-group">
                    <label>Primary Color:</label>
                    <input        type="color"        name="primary-color"
[(ngModel)]="modalData?.event.color.primary"
                      (change)="refresh.next()">
                  </div>
                </div>
                <div class="col-lg-6 col-12">
                  <div class="form-group">
                    <label>Secondary Color:</label>
                    <input       type="color"       name="secondary-color"
[(ngModel)]="modalData?.event.color.secondary"
                      (change)="refresh.next()">
                  </div>
                </div>
              </div>
              <div class="row">
                <div class="col-lg-6 col-12">
                  <div class="form-group">
                    <label>Starts At:</label>
                    <mwl-demo-utils-date-time-picker       name="starts-at"
[(ngModel)]="modalData?.event.start"
                      (ngModelChange)="refresh.next()"     placeholder="Not
set">
                    </mwl-demo-utils-date-time-picker>
                  </div>
                </div>
                <div class="col-lg-6 col-12">
                  <div class="form-group">
                    <label>Ends At:</label>
```

```
                    <mwl-demo-utils-date-time-picker         name="ends-at"
[(ngModel)]="modalData?.event.end"
                      (ngModelChange)="refresh.next()"     placeholder="Not
set">
                    </mwl-demo-utils-date-time-picker>
                  </div>
                </div>
              </div>
            </div>
          </form>
        </div>
        <div class="modal-footer">
          <button    type="button"    class="btn    btn-outline-secondary"
(click)="close()">Cancel</button>
          <button    type="button"    class="btn    btn-outline-secondary"
(click)="saveEvent(); close()">Save</button>
        </div>
      </ng-template>

      <div class="card">
        <div class="card-content">
          <div class="card-body">
            <div   class="d-sm-flex   justify-content-between   align-items-
center mb-3">
              <div class="text-center text-sm-left mb-2 mb-sm-0">
                <div class="btn-group d-none d-lg-block">
                  <a    class="btn    btn-primary"    mwlCalendarPreviousView
[view]="view" [(viewDate)]="viewDate">
                    Previous
                  </a>
                  <a      class="btn      btn-danger"      mwlCalendarToday
[(viewDate)]="viewDate">
                    Today
                  </a>
                  <a     class="btn     btn-primary"     mwlCalendarNextView
[view]="view" [(viewDate)]="viewDate">
                    Next
                  </a>
                </div>

                <div class="btn-group d-block d-lg-none">
```

```
                  <a      class="btn      btn-primary      ft-chevron-left"
mwlCalendarPreviousView [view]="view" [(viewDate)]="viewDate">
                    <i class="icon icon-arrows-left"></i>
                  </a>
                  <a  class="btn  btn-danger  ft-calendar"  mwlCalendarToday
[(viewDate)]="viewDate">
                    <i class="icon icon-arrows-sign-down"></i>
                  </a>
                  <a      class="btn      btn-primary      ft-chevron-right"
mwlCalendarNextView [view]="view" [(viewDate)]="viewDate">
                    <i class="icon icon-arrows-right"></i>
                  </a>
                </div>
              </div>
              <div class="text-center mb-2 mb-sm-0">
                <h6    class="text-uppercase    mb-0">{{    viewDate    |
calendarDate:(view + 'ViewTitle'):'en' }}</h6>
              </div>
              <div class="text-center text-sm-right">
                <div  class="btn-group  d-none  d-sm-none  d-md-none  d-lg-
block">
                  <a  class="btn  btn-primary"  (click)="view  =  'month'"
[class.active]="view === 'month'">
                    Month
                  </a>
                  <a  class="btn  btn-primary"  (click)="view  =  'week'"
[class.active]="view === 'week'">
                    Week
                  </a>
                  <a  class="btn  btn-primary"  (click)="view  =  'day'"
[class.active]="view === 'day'">
                    Day
                  </a>
                </div>

                <div  class="btn-group  d-block  d-sm-block  d-md-block  d-lg-
none">
                  <a class="btn btn-primary btn-icon-icon" (click)="view =
'month'" [class.active]="view === 'month'">
                    <i class="fa fa-th"></i>
                  </a>
```

```
                  <a class="btn btn-primary btn-icon-icon" (click)="view =
'week'" [class.active]="view === 'week'">
                    <i class="fa fa-columns"></i>
                  </a>
                  <a class="btn btn-primary btn-icon-icon" (click)="view =
'day'" [class.active]="view === 'day'">
                    <i class="fa fa-th-list"></i>
                  </a>
                </div>

              </div>
            </div>
            <div class="row mb-2">
              <div class="col-12">
                <button      class="btn      btn-primary      float-right"
(click)="addEvent()">
                  Add Event
                </button>
              </div>
            </div>

            <div [ngSwitch]="view">
              <mwl-calendar-month-view             *ngSwitchCase="'month'"
[viewDate]="viewDate" [events]="events" [refresh]="refresh"
                [activeDayIsOpen]="activeDayIsOpen"
(dayClicked)="dayClicked($event.day)"
                (eventClicked)="handleEvent('Clicked',      $event.event)"
(eventTimesChanged)="eventTimesChanged($event)">
              </mwl-calendar-month-view>
              <mwl-calendar-week-view               *ngSwitchCase="'week'"
[viewDate]="viewDate" [events]="events" [refresh]="refresh"
                (eventClicked)="handleEvent('Clicked',      $event.event)"
(eventTimesChanged)="eventTimesChanged($event)">
              </mwl-calendar-week-view>
              <mwl-calendar-day-view                 *ngSwitchCase="'day'"
[viewDate]="viewDate" [events]="events" [refresh]="refresh"
                (eventClicked)="handleEvent('Clicked',      $event.event)"
(eventTimesChanged)="eventTimesChanged($event)">
              </mwl-calendar-day-view>
            </div>
          </div>
```

```
    </div>
  </div>
</section>
```

Calendar.Services.ts

```
import { Injectable } from '@angular/core';
import {HttpsService} from "../../shared/services/https.service";
import {NotifyUserService} from "../../shared/services/notify-user.service";

@Injectable({
  providedIn: 'root'
})
export class CalendarService {

  constructor(private http: HttpsService, private info: NotifyUserService ) { }
  async makePaginatedCall({page, limit, rootApi, search, sort= 'createdAt:desc'}) {
    const _start = !page ? page : (page * limit) - 1;
    const api = `/${rootApi}?_q=${search}&_start=${_start}&_limit=${limit}&_sort=${sort}`
    const data: any = await this.http.call({api});
    return data
  }
  async makeCountTotalCall({rootApi}) {
    const api = `/${rootApi}/count`
    const data: any = await this.http.call({api});
    return data
  }

  async createEvent(data) {
    const api = `/calendars`;
    const method = 'POST';
    let response = null;
    try {
       response = await this.http.call({api,method,data});
       if(response) {
        this.info.success('Event created successfully');
```

```
        } else{
         this.info.error('Event creation failed.');
        }

      } catch (error) {
        this.info.error('Event creation failed.');
      }
      return response;
    }
  }
```

Date Time Picker Component

```
import {ChangeDetectorRef, Component, forwardRef, Input} from
'@angular/core';
import {
  getSeconds,
  getMinutes,
  getHours,
  getDate,
  getMonth,
  getYear,
  setSeconds,
  setMinutes,
  setHours,
  setDate,
  setMonth,
  setYear
} from 'date-fns';
import {NgbDateStruct, NgbTimeStruct} from '@ng-bootstrap/ng-
bootstrap';
import {ControlValueAccessor, NG_VALUE_ACCESSOR } from
'@angular/forms';

export const DATE_TIME_PICKER_CONTROL_VALUE_ACCESSOR: any = {
            provide: NG_VALUE_ACCESSOR,
 useExisting: forwardRef(() => DateTimePickerComponent),
            multi: true
};

      @Component ({
```

```
                    selector: 'mwl-demo-utils-date-time-picker',
                    template: `
            <form class="form-inline">
          <div class="form-group">
            <div class="input-group">
              <input
                readonly
                class="form-control"
                [placeholder]="placeholder"
                name="date"
                [(ngModel)] ="dateStruct"
                (ngModelChange)="updateDate()"
                ngbDatepicker
                #datePicker="ngbDatepicker">
                <div class="input-group-append">
                  <div                             class="input-group-text"
(click)="datePicker.toggle()" >
                    <i class="fa fa-calendar"></i>
                  </div>
                </div>
            </div>
          </div>
            </form>
        <ngb-timepicker
          [(ngModel)] ="timeStruct"
          (ngModelChange)="updateTime()"
          [meridian]="true">
        </ngb-timepicker>
              `,
              styles: [
        `
            . form-group {
          width: 100%;
            }
              `
                    ],
              providers: [DATE_TIME_PICKER_CONTROL_VALUE_ACCESSOR]
      })
      export class DateTimePickerComponent implements ControlValueAccessor
{
      @Input () placeholder: string;
```

```
          Date: Date;

          dateStruct: NgbDateStruct;

          timeStruct: NgbTimeStruct;

          datePicker: any;

          private onChangeCallback: (date: Date) => void = () => {};

          constructor (private cdr: ChangeDetectorRef) {}

          writeValue (date: Date): void {
this.date = date;
this.dateStruct = {
  day: getDate(date),
  month: getMonth(date) + 1,
  year: getYear(date)
};
   this.timeStruct = {
  second: getSeconds(date),
  minute: getMinutes(date),
  hour: getHours(date)
};
   this.cdr.detectChanges();
          }

          registerOnChange(fn: any): void {
   this.onChangeCallback = fn;
          }

          registerOnTouched(fn: any): void {}

   updateDate (): void {
const newDate: Date = setYear (
  setMonth (
   setDate(this. Date, this.dateStruct.day),
   this.dateStruct.month - 1
  ),
  this.dateStruct.year
```

```
            );
                  this.onChangeCallback(newDate);
            }

                  updateTime(): void {
        const newDate: Date = setHours(
          setMinutes(
            setSeconds(this.date, this.timeStruct.second),
            this.timeStruct.minute
          ),
          this.timeStruct.hour
        );
        this.onChangeCallback(newDate);
                  }
      }
```

Dashboard Code

Dashboard Components (HTML)

```
<div class="row" (resized)="onResized($event)">
<div class="col-xl-3 col-lg-6 col-md-6 col-12">
<div class="card bg-info bg-lighten-3">
  <div class="card-content">
    <div class="card-body py-0">
      <div class="media">
        <div class="media-body info text-left">
          <h3 class="font-large-1 info mb-0">{{totalCrops | number}}</h3>
                   <span>Total Crops</span>
                 </div>
                 <div class="media-right info text-right">
                   <i class="ft-list font-large-1"></i>
                 </div>
               </div>
             </div>
             <div       id="Widget-line-chart"       class="WidgetlineChart
WidgetlineChartShadow mb-3">

             </div>
           </div>
            </div>
                  </div>
```

```
                  <div class="col-xl-3 col-lg-6 col-md-6 col-12">
        <div class="card bg-danger bg-lighten-3">
          <div class="card-content">
            <div class="card-body py-0">
              <div class="media">
                <div class="media-body info text-left">
                  <h3   class="font-large-1   info   mb-0">{{totalFarmers   |
number}}</h3>
                  <span>Total Farmers</span>
                </div>
                <div class="media-right warning text-right">
                  <i class="ft-info-circle font-large-1"></i>
                </div>
              </div>
            </div>
            <div      id="Widget-line-chart2"      class="WidgetlineChart1
WidgetlineChart1Shadow mb-3">

            </div>
          </div>
        </div>
                        </div>

                        <div class="col-xl-3 col-lg-6 col-md-6 col-12">
           <div class="card bg-success bg-lighten-3">
          <div class="card-content">
            <div class="card-body py-0">
              <div class="media">
                <div class="media-body info text-left">
                  <h3   class="font-large-1   info   mb-0">{{totalCamps   |
number}}</h3>
                  <span>Total Camps</span>
                </div>
                <div class="media-right warning text-right">
                  <i class="ft-alert-circle font-large-1"></i>
                </div>
              </div>
            </div>
            <div      id="Widget-line-chart2"      class="WidgetlineChart1
WidgetlineChart1Shadow mb-3">
```

```
            </div>
          </div>
                  </div>
                  </div>

                        <div class="col-xl-3 col-lg-6 col-md-6 col-12">
                  <div class="card bg-warning bg-lighten-3">
          <div class="card-content">
            <div class="card-body py-0">
              <div class="media">
                <div class="media-body info text-left">
                  <h3    class="font-large-1    info    mb-0">{{totalZones|
number}}</h3>
                  <span>Total Zones</span>
                </div>
                <div class="media-right warning text-right">
                  <i class="ft-alert-circle font-large-1"></i>
                </div>
              </div>
            </div>
            <div      id="Widget-line-chart2"      class="WidgetlineChart1
WidgetlineChart1Shadow mb-3">

            </div>
          </div>
           </div>
                        </div>

      </div>




<section id="minimal-statistics-bg">
                        <div class="row">
<div class="col-12 mb-1">
<div class="content-header"></div>
<p class="content-sub-header mb-1"></p>
</div>
</div>

</section>
```

Dashboard services

```
import { Injectable } from '@angular/core';
import {HttpsService} from '../../shared/services/https.service';

        @Injectable ({
        providedIn: 'root'
})
 export class DashboardService {

        constructor (private http: HttpsService) { }


        totalCrops() {
     const method = 'get';
     const api = '/crops/count';
     return this.http.call({method, api})
         }

        totalFarmers() {
     const method = 'get';
     const api = '/farmers/count';
     return this.http.call({method, api})
        }

              totalCamps() {
const method = 'get';
const api = '/Camps/count';
return this.http.call({method, api})
       }

              totalZones() {
     const method = 'get';
     const api = '/Zones/count';
     return this.http.call({method, api})
              }

     searchDB(search) {
     const method = 'get';
     const api = '/customers/search?_q=' + search;
```

```
    return this.http.call({method, api})
  }


}
```

Appendix 2: Certificate of Publication

Scientific & Academic Publishing

Computer Science and Engineering

Certificate of Publication

Date: December 25, 2023

Dear Colleague,

Thank you very much for your contribution to Scientific & Academic Publishing.

We are pleased to inform you that your paper has been published online.

Paper ID:	108100232
Paper Title:	Design and Implement a Crop Management System
Authored by	Gregory Kakoma Chikanwa, Simon Tembo, Ir. Donat M. Ngendo

It is published in *Computer Science and Engineering (Volume 13, Number 1, 2023).*

We look forward to your continued support.

Thank you again.

Sincerely yours,

Charles Duke

Journal manager

Scientific & Academic Publishing (http://www.sapub.org/), USA

SCIENTIFIC & ACADEMIC PUBLISHING CO. LTD
CORPORATE SEAL
USA
CALIFORNIA

Appendix 3: Published article

Computer Science and Engineering 2023, 13(1): 1-7
DOI: 10.5923/j.computer.20231301.01

Design and Implement a Crop Management System

Gregory Kakoma Chikanwa*, Simon Tembo, Ir. Donat M. Ngendo

Department of Electrical Engineering, University of Zambia, School of Engineering, Lusaka, Zambia

Abstract Farmers in Zambia of Chongwe District rely primarily on services provided by agricultural extension officers to boost crop production. These officials aim to visit farmers throughout the agricultural calendar covering all eight (8) zones within each agricultural camp at least twenty-four (24) times a year and twice (2) a month to impart farmers with knowledge on crop management. They face various challenges, in terms of reaching remote and rural areas. This project highlights the necessity and importance of agricultural extension services in supporting farmers. To solve this problem, an e-agriculture extension application has been designed and developed to alleviate the challenges faced by the field workers. The e-agriculture extension application aims to provide basic knowledge in crop management and how to plan for a successful agricultural calendar, especially for remote and rural farmers. Extension workers typically rely on the agricultural calendar produced by the Department of Agricultural Methodologists. The application was built using an agile methodology that allows each sprint to be carefully planned and approved by the client before moving on to the next.

Keywords Extension Officer, Agile, Scrum, Agriculture Camp, Agriculture Zone, System, ICT

1. Introduction

Farmers in Zambia primarily rely on the services provided by agriculture extension officers to bolster their crop production. While it is expected that farmers should receive regular visits from extension officers throughout the agricultural calendar, they often encounter various challenges in reaching remote and rural areas [1]. This article underscores the imperative need and significance of agricultural extension services in supporting farmers. In response to these challenges, we introduce an e-agriculture extension application designed and developed to alleviate the issues faced by extension officers.

This innovative application is aimed at equipping remote and rural farmers with fundamental knowledge of crop management and facilitating the creation of successful agricultural calendars. Extension officers typically employ an agriculture calendar prescribed by the agriculture methodologist at the Ministry of Agriculture's headquarters as their primary tool for crop management and planning.

This article will provide an overview of the introduction, objectives, project scope, literature review, the methodology employed in system development, results, challenges, and conclusive summary.

The government of the Republic of Zambia has adopted the use of electronic extension services by embracing the use of information and communication technology (ICT) to cover areas that do not have sufficient agriculture extension officers. The adoption of this technology is meant to benefit from the 2,000 towers that the Ministry of Transport and Communication planned to erect countrywide [2]. Amid the Coronavirus Disease 2019 (COVID-19) pandemic, farmers encountered challenges in crop management due to the unavailability of extension officers as a result of COVID-19. Farmers possess knowledge about their environment and farming system. However, they are mandated to receive visits from agricultural extension officers at least twenty-four (24) times a year, twice (2) a month in each of the eight (8) Zones within each agriculture Camp. These visits assist in crop and disease management and the recommendation of suitable pesticides for specific crops. Unfortunately, the funding reduction due to COVID-19 made it difficult for farmers to access these extension officers [1].

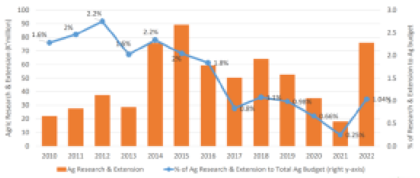


Figure 1. Budget allocation towards Extension services for 2022

The agriculture extension officers are not always available to assist farmers in identifying the diseases affecting their crops in the fields. The extension officers whenever

* Corresponding author:
onechiks@gmail.com (Gregory Kakoma Chikanwa)
Received: Nov. 13, 2023; Accepted: Dec. 1, 2023; Published: Dec. 13, 2023
Published online at http://journal.sapub.org/computer

Appendix 4: Approval of Study

THE UNIVERSITY OF ZAMBIA

DIRECTORATE OF RESEARCH AND GRADUATE STUDIES

Great East Road Campus | P.O. Box 32379 | Lusaka 10101 | **Tel:** +260-290 258/291 777
Fax: (+260) 211 290 258/253 952 | **Email:** director.drgs@unza.zm | **Website:** www.unza.zm /directorates/drgs

APPROVAL OF STUDY

IORG No. 0005376
NASREC IRB No. 00006465

8th December, 2022

REF NO. NASREC-2022-JUL-003

Mr. Gregory Chikamwa
The University of Zambia
School of Engineering,
P.O. Box 32379,
LUSAKA.

Dear Mr. Chikamwa,

RE: "DESIGN AND IMPLEMENT A CROP MANAGEMENT SYSTEM FOR THE FARMERS IN CHONGWE DISTRICT OF ZAMBIA"

Reference is made to your protocol dated as captioned above. NASREC resolved to approve this study and your participation as Principal Investigator for a period of one year.

REVIEW TYPE	ORDINARY REVIEW	APPROVAL NO. NASREC-2022-JUL.-003
Approval and Expiry Date	Approval Date: 8th December, 2022	Expiry Date: 8th December, 2023
Protocol Version and Date	Version - Nil.	28th November, 2023
Information Sheet, Consent Forms and Dates	• English.	To be provided
Consent form ID and Date	• Version - Nil	To be provided
Recruitment Materials	• Nil	Nil
Other Study Documents	• Questionnaire.	

Towards Improving Service and Excellence in High Education Beyond Fifty Years

Specific conditions will apply to this approval; As Principal Investigator it is your responsibility to ensure that the contents of this letter are adhered to. If these are not adhered to, the approval may be suspended. Should the study be suspended, study sponsors and other regulatory authorities will be informed.

CONDITIONS OF APPROVAL

- No participant may be involved in any study procedure prior to the study approval or after the expiration date.
- All unanticipated or Serious Adverse Events (SAEs) must be reported to NASREC within 5 days.
- All protocol modifications must be approved by NASREC prior to implementation unless they are intended to reduce risk (but must still be reported for approval). Modifications will include any change of investigator/s or site address.
- All protocol deviations must be reported to NASREC within 5 working days.
- All recruitment materials must be approved by NASREC prior to being used.
- Principal investigators are responsible for initiating Continuing Review proceedings. NASREC will only approve a study for a period of 12 months.
- It is the responsibility of the PI to renew his/her ethics approval through a renewal application to NASREC.
- Where the PI desires to extend the study after expiry of the study period, documents for study extension must be received by NASREC at least 30 days before the expiry date. This is for the purpose of facilitating the review process. Documents received within 30 days after expiry will be labelled "late submissions" and will incur a penalty fee of K500.00. No study shall be renewed whose documents are submitted for renewal 30 days after expiry of the certificate.
- Every 6 (six) months a progress report form supplied by The University of Zambia Natural and Applied Sciences Research Ethics Committee as an IRB must be filled in and submitted to us. There is a penalty of K500.00 for failure to submit the report.
- When closing a project, the PI is responsible for notifying, in writing or using the Research Ethics and Management Online (REMO),both NASREC
- and the National Health Research Authority (NHRA) when ethics certification is no longer required for a project.
- In order to close an approved study, a Closing Report must be submitted in writing or through the REMO system. A Closing Report should be filed when data collection has ended and the study team will no longer be using human participants or animals or secondary data or have any direct or indirect contact with the research participants or animals for the study.
- Filing a closing report (rather than just letting your approval lapse) is important as it assists NASREC in efficiently tracking and reporting on projects. Note that some funding agencies and sponsors require a notice of closure from the IRB which had approved the study and can only be generated after the Closing Report has been filed.
- A reprint of this letter shall be done at a fee.

- All protocol modifications must be approved by NASREC by way of an application for an amendment prior to implementation unless they are intended to reduce risk (but must still be reported for approval). Modifications will include any change of investigator/s or site address or methodology and methods. Many modifications entail minimal risk adjustments to a protocol and/or consent form and can be made on an Expedited basis (via the IRB Chair). Some examples are: format changes, correcting spelling errors, adding key personnel, minor changes to questionnaires, recruiting and changes, and so forth. Other, more substantive changes, especially those that may alter the risk-benefit ratio, may require Full Board review. In all cases, except where noted above regarding subject safety, any changes to any protocol document or procedure must first be approved by NASREC before they can be implemented.

Should you have any questions regarding anything indicated in this letter, please do not hesitate to get in touch with us at the above indicated address.

On behalf of NASREC, we would like to wish you all the success as you carry out your study.

Yours faithfully,

Dr. E. M. Mwanaumo

CHAIRPERSON
THE UNIVERSITY OF ZAMBIA NATURAL AND APPLIED SCIENCES RESEARCH ETHICS COMMITTEE - IRB

cc: Director, Directorate of Research and Graduate Studies
Assistant Director (Research), Directorate of Research and Graduate Studies
Assistant Registrar (Research), Directorate of Research and Graduate Studies