UNIVERSITY OF NAIROBI

SCHOOL OF COMPUTING AND INFORMATICS

MSc. DCT PROJECT: Application Development

A COGNIFIED DISTRIBUTED SYSTEM FOR LIVESTOCK DISEASES: CASE OF PASTORAL COMMUNITIES IN KAJIADO COUNTY, KENYA.

BY

SALIM JUMA KINYIMU

P53/36130/2019

Supervisor: Mr. Eric Masiga Ayienga

This thesis is a requirement for the award of Masters of Science in Distributed Computing Technology at the University of Nairobi.

DECLARATION

I, Salim Kinyimu, declare that this research project is entirely my original work, and where there is work for other individuals or organizations, it has been well referenced and acknowledged.

To my knowledge, similar work has never been carried out before or submitted to any university.

NAME: SALIM JUMA KINYIMU

REGISTRATION NUMBER: P53/36130/2019

……………………………………… 4th August 2022……………………………….

Signature *Date*

This research project has been submitted for examination with my approval as university supervisor.

MR. ERIC MASIGA AYIENGA

……………………………………… 4th August 2022……………………………….

Signature *Date*

DEDICATION

I dedicate this thesis to my dear mum Popo Suleiman and my lovely wife, Haulat Nasur, the two women who always push me to the pinnacles. To my son Jumaa Salim, you are my special friend with an IQ beyond your age! To my daughter Yusra Salim, you're the most intelligent girl I have ever seen!

I love you all, and thank you so much for always believing in me.

ACKNOWLEDGEMENT

First and foremost, I would give special thanks to the Almighty God for the gift of good health and courage throughout the journey. I can attest that the journey needed hard work, dedication, patience, and persistence.

The research work is a product of support from different institutions and individuals. The entire study was made possible through financial support from the World Bank under the Kenya Climate Smart Agriculture Project (KCSAP). My special appreciation to Kenya Agricultural & Livestock Research Organization (KALRO) scientists Dr. Harrison Lutta and Dr. Yatinder Binepal for their inputs in modeling climate factors in vector-borne diseases.

I appreciate my supervisor's valuable inputs, guidance, and suggestions from the proposal stage to the complete thesis. I further appreciate Prof. Robert Oboko for his input in this work.

I received lots of support from the KALRO data center team led by the Data Scientists Mr. John Kisangau and Mr. Albert Agoya. I am very grateful for your help.

ABSTRACT

In Kenya, the livestock sector experience an unexpected outbreak of contagious diseases that led to the loss of animals and decreased productivity in live cattle. This impacts the livelihoods of pastoral communities, mainly the youth and women, who solemnly depend on this sector as their primary source of income through selling live cattle, extra meat, and milk. Traditional methods used to predict the reoccurrence of contagious diseases are no longer accurate due to unpredicted weather patterns caused by the effect of climate change and associated risks. This calls for accurate, timely, and location-specific advisories on priority livestock diseases such as Rift Valley Fever (RVF), Bovine Ephemeral Fever (BEF), and Capripox virus to prevent losses incurred by farmers. The objective of this study was to test cognified distributed technology in handling data-driven models to generate data-based evidence used to predict the subsequent chances of disease reoccurrence.

The study was done in Kajiado county with a sample size of sixty-five (65) livestock farmers. A constructive research approach was used to develop custom-made surveillance and reporting prototype that leverages high-performance computing resources and real-time weather forecast data from remote satellites. Results from the prototype show a tandem between the number of infections reported and the predicted chances of occurrence generated by the model. Cognified distributed system can handle massive volumes of data coming in different types, formats, magnitudes, and locations. When the incoming data is well formatted and compared with the historical data pattern, the computing resources can perform pattern and matching analysis to determine the chances of disease reoccurrence. Kenya as a country stands to gain by adopting this technology in veterinary epidemiology. The technology will guide agricultural stakeholders, including policymakers, on early response mechanisms and the prioritization of vaccination programs. This further extends to improving food security, a pillar of the Government's *Big Four agenda*.

KEYWORDS

Cognified, predictive, occurrence, traceability, surveillance, data-driven, High-performance computing (HPC)

Table of Contents

Table of Figures

LIST OF ABBREVIATION

ASAL – Arid and Semi-Arid Lands

ASTGS - Agricultural Sector Transformation and Growth Strategy

BEF - Bovine Ephemeral Fever

DAT - Disruptive Agricultural Technology

DSS - Decision Support System

FAO - Food Agricultural Organization

FMD - Foot and Mouth Disease

GIS - Geographic Information System

GPS - Global Positioning System

HPC – High-Performance Computing

IoT - Internet of Things

KALRO - Kenya Agricultural and Livestock Research Organization

KCSAP - Kenya Climate Smart Agriculture Project

KCSAS - Kenya Climate Smart Agriculture Strategy

MoALFC - Ministry of Agriculture, Livestock, Fisheries and Co-operatives

PPR - Peste des Petis Ruminants

RVF - Rift Valley Fever

WHO – World Health Organization

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

The livestock sector partially or entirely relies on the livelihood of people in all regions of the world (FAO, 2017). Livestock supports the food supply and livelihoods of 1.3 billion people worldwide and contributes 40% of the global agricultural outputs. In Kenya, pastoral communities lack accurate, timely, and location-specific advisories on priority livestock diseases such as Rift Valley Fever (RVF), Bovine Ephemeral Fever (BEF), and Capripox virus (Matere *et al*., 2009; Perry & Grace, 2019). The lack of adequate information on these diseases leads to a decrease in red meat and milk production. This further impairs farmers' livelihoods in Arid and Semi-Arid Lands (ASALs), mostly youth and women who rely on the livestock subsector as their primary source of income through the sale of live cattle, extra meat, and milk (Makokha & Witwer, 2013).

A surveillance tool is a systematic information-based procedure within the population that collects, analyzes, and interprets large volumes of data from various sources (WHO, 2006). The collated information is then used to evaluate the effectiveness of control and preventative measures. Reporting systems generate reports using a management system that consists of a human interface, procedures, and processes. In a reporting system, messages are distributed to relevant stakeholders from the management system. Cognification technology has an intelligent way of creating traditional techniques such as surveillance and reporting tool to be smarter by linking them to remote sensors and other Artificial Intelligent software (Babaoglu & Sîrbu, 2018).

Climate prediction in Kenya indicated a rise in temperature of 1^{0} C by the end of 2020 and a possible rise of 2.3^{0} C by 2050 (KCSAS, 2017), which might accelerate incidences of livestock diseases (Carew-Reid *et al*., 2013). Early warning systems are needed to mitigate the re-emergence of priority livestock diseases such as RVF, BEF, and Capripox Virus in livestock farming (Matere *et al*., 2009; Perry & Grace, 2019).

1.2 PROBLEM STATEMENT

Disruptive Agricultural technologies (DAT) in the digital space have proven to handle Africa's food security challenges caused by climate change (Kim *et al.*, 2019). Climate change in agriculture contributes to the high cost of inputs, low farm yields and productivity, limited use of new agricultural technologies, and poor commercialization strategies. It is the primary cause of increasing food insecurity and acceleration of livestock diseases within the pastoral communities. Geographical Information System (GIS) has proven relevant when plotting areas susceptible to disease, thus encouraging early warning mechanisms and disease control strategies. In Kenya, disease reporting methods are inefficient since many smallholder farmers make phone calls and physical visits to veterinary offices while others do not report at all (Peninah *et al.*, 2016). This lead to poor disease containment and management.

As GIS is signifying a better technological approach in veterinary epidemiology that can be adopted in reporting animal diseases and modeling specific disease problems, this does not provide models to predict the occurrence or reoccurrence of the diseases (Fayisa, 2020). This challenge is caused by a lack of relevant data for analyzing and interpreting changes in the environmental variation of pathogens and vectors, and a lack of enough computing capacity to utilize climate models for disease (Bett *et al.*, 2016).

In a cognified distributed environment, agricultural intelligence uses data analytics solutions that leverage on data evidence and infrastructure, mapping technologies, remote sensing capabilities, precision agriculture software tools, and data center computing resources to inform decisions. This allows agricultural policymakers and private service providers to use data-driven decision-making in the agriculture sector (Kim *et al*., 2019). The analytics solutions can be applied in predicting the occurrence of livestock disease. According to Fayisa (2020), there is a need for further study on the use of GIS technologies to map the spread of livestock diseases.

1.3 RESEARCH QUESTIONS

1. "What are the distributed technological options to provide accurate data evidence that can predict disease occurrence in areas affected by climate change?"

2. "What are the available models employed in agricultural research to reduce the spread of contagious livestock diseases?"
3. "How sustainable are these models to support the easy distribution of accurately, timely, and location-specific advisories to the agricultural stakeholder?"

1.4 RESEARCH OBJECTIVES

1. To analyze field data and predict disease reoccurrence using the Predictive analytics tool.
2. To establish a community-based surveillance and reporting system for livestock diseases (BEF, RVF, Capripox Virus).

1.5 SPECIFIC OBJECTIVES

1. To assess the existing similar system to review strengths and weaknesses.
2. To evaluate and monitor the outcomes and impact of the surveillance system to identify gaps.
3. To use computer models to predict the reoccurrences of contagious livestock diseases.
4. To develop a dynamic map of high-risk areas and livestock disease using cognified technology.

1.6 JUSTIFICATION AND SIGNIFICANCE OF THE STUDY

This project developed a viable solution to prevent pastoralists from incurring the massive loss of livestock due to reoccurrences of contagious diseases. The study sought to find an accurate and real-time reporting model by applying new computer technologies that utilized data models and patterns to predict the future reoccurrence of livestock diseases. The study findings are useful to academic researchers, disruptive technology accelerators, pastoral communities, government policymakers, and institutions of higher learning.

Kajiado county was the ideal location to conduct the study. It is one of the counties in the ASALs region where most communities are pastoralists affected by climate change. Kajiado is also one of the counties supported by the Kenya Climate Smart Agriculture Project, which sponsors this study.

The study helps to strengthen the development process by reviewing existing systems and filling out the gaps caused by the lack of capacity to avail real-time and relevant information to interpret the data and predict the next disease reoccurrence. The developed system will be used by veterinary stakeholders as a centralized planning resource for location-specific on animal vaccination programs and informing policymakers on where to prioritize resources for extension services.

Using a cognified distributed system makes timely detection of the diseases possible, thus reducing the costs of response and loss incurred. From the prototype system developed, this project has set the foundation to predict future reoccurrence of other livestock diseases caused by changes in weather patterns.

1.7 SCOPE

The study concentrated on three (3) diseases; RVF, BEF, and Capripox Virus, in areas affected by climate change. The surveillance system was used to monitor the disease pattern in Kijiado county, located in Kenya. The technology used was limited to what Kenya Agricultural & Livestock Research Organization (KALRO) has in the primary data center.

CHAPTER 2: LITERATURE REVIEW

2.1 INTRODUCTION

Many pastoral farmers in Kenya are smallholders who rely on livestock production as the primary source of livelihood. Mitigating contagious diseases is not a priority as their primary focus is on pasture and water for their herds. For many pastoralists, issues with livestock diseases can only be handled during an outbreak.

In most cases, whenever there is an outbreak of contagious diseases, it becomes practically impossible to contain the situation due to a lack of veterinary knowledge and other farm resources to quarantine the infected animals. This further accelerates the spread of the diseases. Very few farmers have access to professional veterinary officers and the capacity to follow the guidelines given to them by the authorities.

2.2 BACKGROUND OF STUDY

Despite the guidelines issued by the Ministry of Agriculture Livestock Fisheries and Cooperatives (MoALFC) on mitigating the spread of contagious livestock diseases, many farmers are not observing the guidelines due to the inaccessibility of agricultural extension services. To these farmers, animal vaccination is also not a priority.

In Kenya, many livestock farmers are disadvantaged in accessing County extension services, leading to decrease in animal productivity due to frequent diseases. In developing countries, livestock diseases contribute to 20% of ruminant mortality and 50% of mortality for non-ruminant. This further extends to an annual loss in global revenue of $300 billion from the livestock sector (Delia *et al*., 2015). Based on historical records, contagious diseases such as BEF, Capripox Virus, Foot and Mouth Disease (FMD), RVF, and Peste des Petis Ruminants (PPR) have recorded the highest numbers.

In Kajiado county, common contagious diseases are vector-borne caused by pathogens that emerge whenever a weather pattern changes. The main weather patterns are either Short-Term Normal or Long-Term Normal. There is a need to avail to pastoral farmers accurate information on the occurrence and reoccurrence of priority diseases. This is vital information to guide the pastoral community on prevention measures (Alhaji *et al*., 2018). It can also inform policymakers like government agencies on where to prioritize animal vaccination or buy off the herds (Lemon *et al*., 2007).

2.3 CLIMATE CHANGE AND LIVESTOCK DISEASES

Climate change leads to an increase of vector pathogens that spread vector-borne diseases. This leads to the reoccurrence and spreading of livestock diseases such as Capripox virus, BEF, RVF, and PPR (Alhaji *et al.*, 2020). Changes above normal precipitation increase the frequency of RVF epidemics while increasing temperature would cause shifts in the spatial distribution of BEF.

According to Alhaji *et al*. (2020), animal vaccination program in the developing country lacks quality due to inadequate control at the manufacturing stage and lack of proper handling mechanisms by incubators during distribution. Some distributors offer substandard dosages to carter for the massive demand for vaccination in the regions. African countries, which are heavily affected by the change in climate, are reducing the resources for public veterinary services. Yet, there are enormous demands for additional extension officers in rural areas. This is evidently by a lack of proper information channels to capacitate farmers on how to mitigate, control and report contagious diseases in remote areas.

In rural areas with limited or no government extension service, E-extension tends to be the only solution that can assist farmers who mostly own large cattle. For a surveillance tool to work effectively in these areas, it requires an understanding and modeling of climate diseases before inferential controls and measures are given out.

2.4 TRADITIONAL TECHNIQUES FOR PREDICTING LIVESTOCK DISEASES

Apart from the apparent diseases caused by prolonged heavy rainfall, traditionally it is almost impossible to predict future incidences of livestock diseases. Communities focus on curing rather than preparing for disease outbreaks.

2.5 MODERN TECHNIQUES FOR PREDICTING LIVESTOCK DISEASES

Modern technologies have made it possible to predict the future reoccurrence of diseases based on analysis derived from observed data sets from the environment and compared against some predefined disease conditions.

2.5.1 Forecast Model of Predictive Analytics

Predictive analytics is a set of procedures and techniques applied to a well-formatted dataset to produce the subsequent chances of events or outcomes (White *et al*., 2018). Big data analytics is used in many aspects of business sectors to improve insights and knowledge required to make an informed decision. The agricultural industry also can benefit from these analytical techniques. Livestock farmers collect adequate and relevant farm data that are enough to perform predictive analytics. Using predictive models such as forecasts, the data generated from the farms can be formatted and used by an automated early warning system to replace human extension services in areas with limited access to veterinary services (FAO, 2011).

Forecast Model operates by learning the recorded data and predicting metric and numeric values for new data. The figure below conceptualizes a forecasting model used to calculate disease risks in crops.

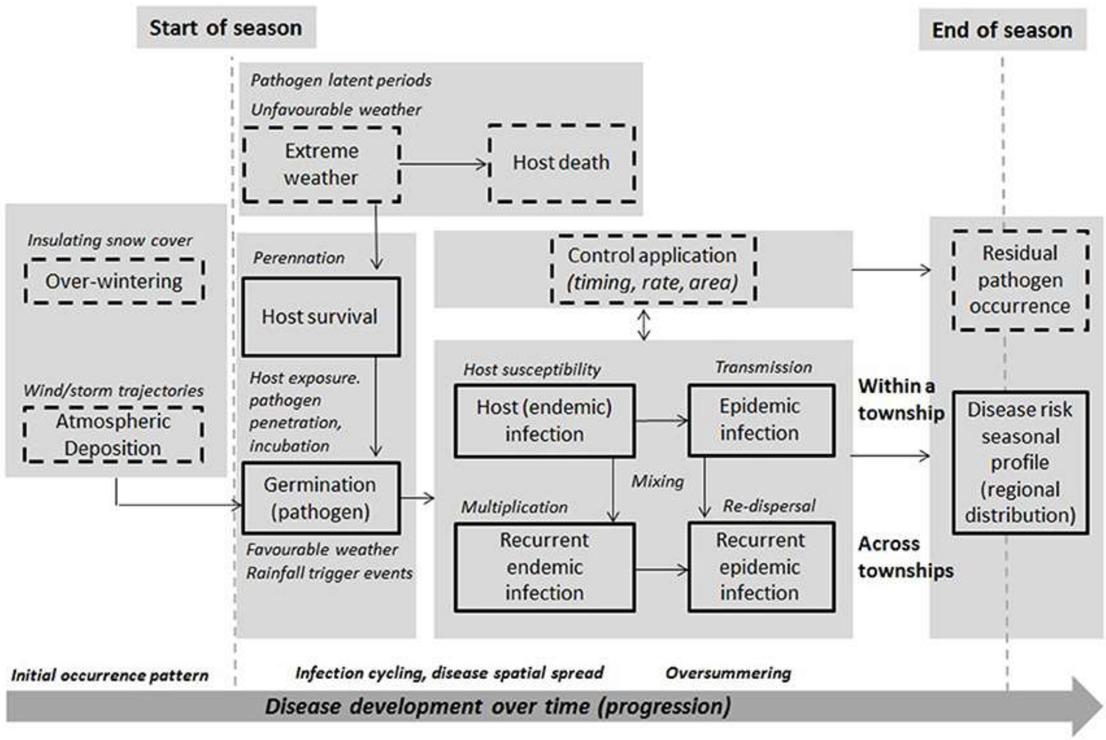

Figure 1: Forecast model for crop disease risk

Source: Model-Based Forecasting of Agricultural Crop Disease Risk at the Regional Scale, Integrating Airborne Inoculum, Environmental, and Satellite-Based Monitoring Data. Nathaniel, K. N. (2018)

2.5.2 Decision Support System

Decission Support System (DSS) is an interactive computer system that uses data manipulation techniques and models in an easy user interface. It can incorporate decision-making thinking to provide an informed and optimal choice (Fotis & Maria, 2018). It is an approach to support decision-making using unpredictable analysis that uses less or no programming effort (Randall & Sauter, 2002). According to Fotis & Maria (2018), DSS informs and supports decisions in a non-formal environment where an ordinary human being cannot tell what choice is better for a particular situation. DSS module contains a database and a data model that feed the system with processed data and information that are presented to the user in a simplified form (Randall & Sauter, 2002).

Randall & Sauter (2018) categorize DSS into the following classifications;

- Data-driven DSS - contains data warehousing and information on data analysis systems that guide the handling of structured datasets.

- Model-driven DSS - contains specialty knowledge in a specific discipline.
- Knowledge-driven DSS - contains recommendations and the best solution to a given situation.

2.5.3 Cognified Distributed Technology

Cognified distributed technology converts a conventional distributed computing node, process or service into smarter and intelligent equivalents through data science, Machine Learning, and Artificial Intelligence (Ozalp and Alina, 2018). It is a revolutionary technology transforming many aspects of businesses and agricultural fields. Traditional computing systems are linked with machine learning, and data science features to enhance service delivery and improve reliability. Adopting this technology is accelerated by quickly accessing large volumes of data in different data types and manipulating them using analytical tools in a high-performance computing (HPC) environment. Modern datacenters environments are ready to accommodate cognified technologies by introducing on-premises HPC systems and supplying reliable clean power.

2.5.4 Disruptive Agricultural Technology

DAT is a new technology in agriculture that allows farmers and agribusiness entrepreneurs to use digital and technical innovations to advance existing methods to increase farm productivity, efficiency, market access, competitiveness, improve nutritional outcomes and enhance resilience to climate change (Kim *et al*., 2019). DAT is different from other agri-tech solutions such as mobile apps. It allows farmers to access weather events and patterns, future market demand, soil characteristics, and other variables.

When farmers are limited to this information, they make decisions based on intuition which is often inefficient. Inequalities in access to technologies, information, and market create marginalized groups in Africa evidently by low-skilled farmers in rural areas of African countries (Kim *et al*., 2019). DAT can address these challenges by helping farmers make accurate, timely, and location-specific decisions and reducing the cost of linking agricultural actors (Kim *et al*., 2019).

2.5.4.1 Application of GIS in Veterinary Epidemiology

According to Fayisa (2020), GIS is a computer system that manipulates and represents topographical referenced data in the form of maps. The strength of GIS in a surveillance system is in integrating Global Positioning System (GPS) data, satellite images, and dynamic maps in an associated database. Using GIS, one can realize the pattern of spreading disease and contain it from further spreading (Mengistu & Haile, 2017).

GIS applications can prevent and manage animal diseases in veterinary epidemiology by mapping the disease information (Fayisa, 2020). Spatial analysis can be used in climate modeling to provide early warning alerts to diseases that are caused by climate change. GIS functions can be integrated into the disease surveillance system and simulate disease spreading (Mengistu & Haile, 2017). This further extends to assist in improving response time to contain the conditions.

2.5.4.2 Remote Sensors in Agriculture

Remote sensing is detecting and monitoring an area's physical characteristics from a distance using independent devices such as the Internet of Things (IoT). IoT devices fuel the application of remote sensors in agriculture. Ultimately, IoT will connect all interacting human objects on earth to the internet. The goal is to enhance remote management of this object from any location (Khanna & Kaur, 2019). Agricultural data can be collected through various remote devices with sensors that are connected to wired or wireless network technologies. The devices are then linked to a central management system where data is harvested in real-time (Jayaraman *et al.*, 2016). According to Khanna & Kaur (2019), remote sensor application is in precision agriculture. In the new farming era, there is a massive shift in techniques, methodologies, and concepts on how to farm. Farmers are now automating their farming practices to reduce input costs and maintain the same productivity (Khaana & Kaur, 2019).

A study by Jayaraman et al. (2016) proposed an IoT platform dubbed "SmartFarmNet" capable of automatically collecting data from various parameters, including environmental, soil, fertilization, and irrigation. Their study suggested a framework that analyzes and cleans the incoming data from the remote sensors in real-time as they stream into the centralized system. This framework immediately generates data insights and meanings, and sends them to stakeholders in real-time.

2.5.4.3 Decision Tree

In data science, decision trees are used to illustrate complicated data models in the form of a tree-like structure. When this structure is followed, it can predict a value or likelihood of an event and inform the subsequent decision to be decided. The data models are created from the ingested datasets in a controlled machine learning process.

While data mining extract knowledge from non-traditional large data sets, decision trees helps to discover data prediction, forecast, and trend. The figure below shows a decision tree analysis used to evaluate the chances of disease infection for drying off cows.

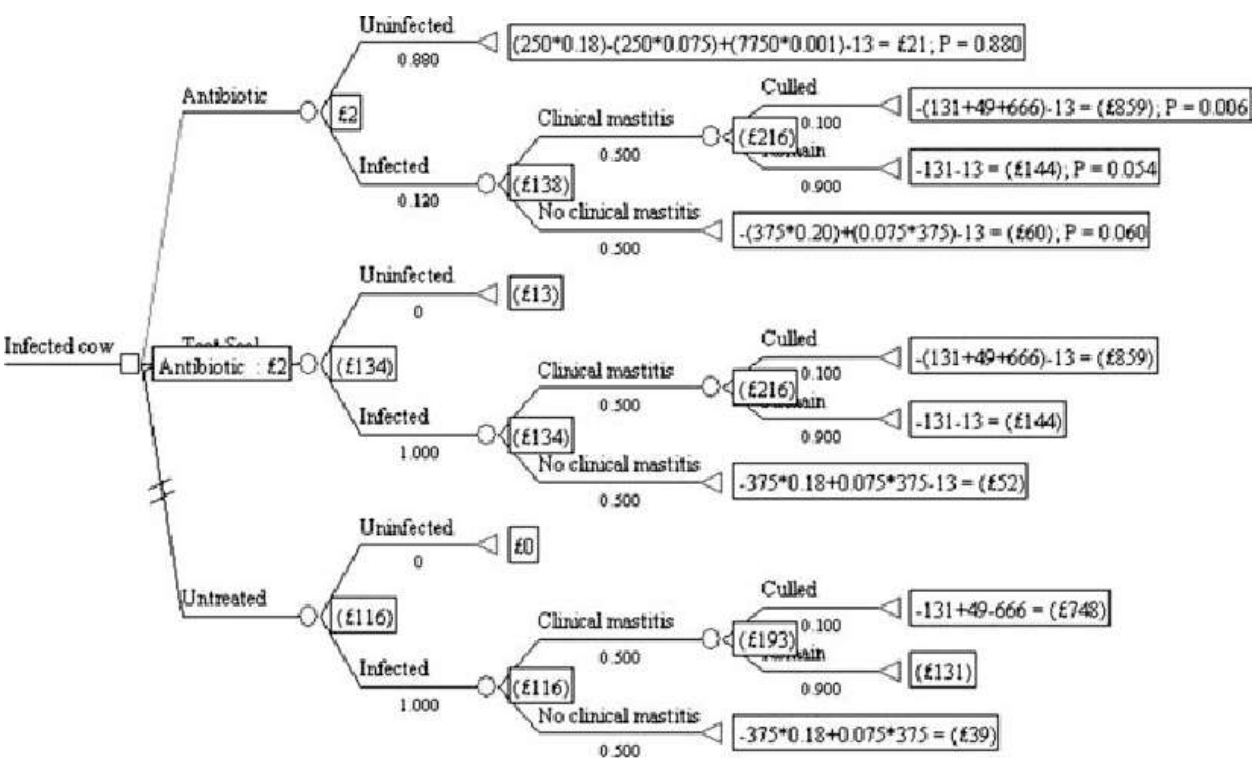

Figure 2: Evaluation of infection chances for drying off cows using a decision tree

Source: Decision tree analysis to evaluate dry cow strategies under UK conditions. Elizabeth A. B., Henk H., & Hillerton J. E

2.6 OTHER WAYS TO PREDICT LIVESTOCK DISEASE

2.6.1 Livestock Identification and Traceability Systems in Botswana

In Botswana, livestock disease handling, control, and management are done using a traceability system and animal identification program. Animal records taken on the farms, during transportation, and processing plants are uploaded to the centralized national database. The database contains information for particular animal diets, parent herd location, health and vaccination records.

The national data are then used to enhance animal traceability and decision-making in case of a disease outbreak. From the database, the consumer can track a particular cattle product throughout the chain. In the likelihood of a product or by-product found with disease infection, the agencies can trace the entire production chain and contain the parent herd where the disease originates. Further contact tracing is done to identify animals with close contact with the quarantined herds. Information is then spread to farmers, warning them of the current disease outbreak.

In the figure below, the traceability of cattle products in Botswana starts from the farm level; all farm records are uploaded into the national cattle database accessible by all value chain input /output providers.

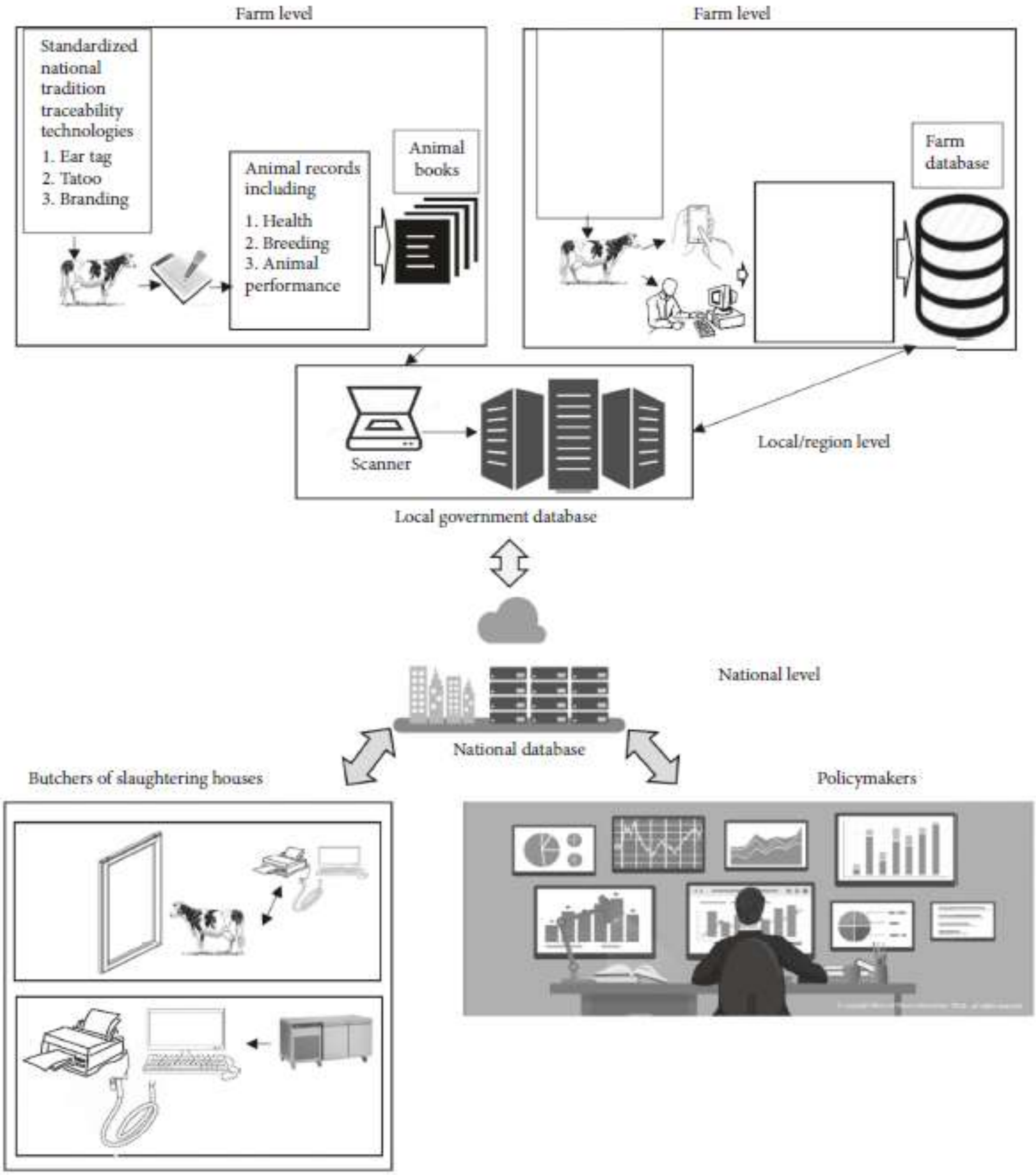

Figure 3: Botswana national framework for a traceability system

Source: How Information Communication Technology Can Enhance Evidence-Based Decisions and Farm-to-Fork Animal Traceability for Livestock Farmers. Mwanga G. et al., (2020)

2.7 EXISTING WORK ON LIVESTOCK SURVEILLANCE & REPORTING SYSTEM

According to WHO (2016), livestock disease surveillance must record the daily disease events and communicate the alerts to the relevant authorities. The authorities are responsible for sending early warning advisories and control measures to the nearby communities in case a disease is reported.

2.7.1 Emergency Prevention System Intelligent (EMPRES-i)

When the livestock mortality rate increased globally due to the spread of contagious diseases, an international consortium was formed through Food Agricultural Organization (FAO) with the mandate to provide regional and national statistics on infection rates and possible deaths caused by conditions. EMPRES-i was developed as an online dashboard that could be accessed by member countries and other veterinary stakeholders to monitor disease spread and receive alerts whenever there are outbreaks.

The figure below shows a screenshot of the EMPRES-i homepage.

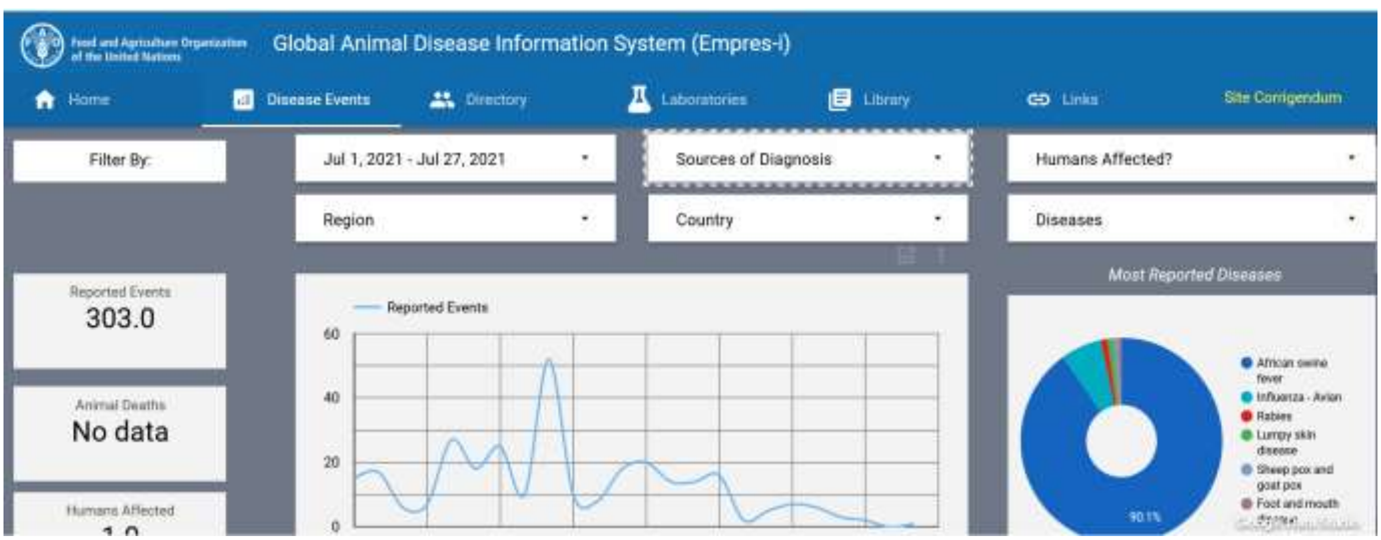

Figure 4: Landing page of EMPRES-i web portal

2.8 CONCEPTUAL MODEL

This study is conceptualized along two layers; the access layer that interacts with data sources and data access, and the core layer that is hosted in the cognified environment and contains the data preparation mechanisms, storage, and the analytic engine.

The figure below shows the conceptual model adopted by this study to implement a cognified distributed system for livestock surveillance and reporting.

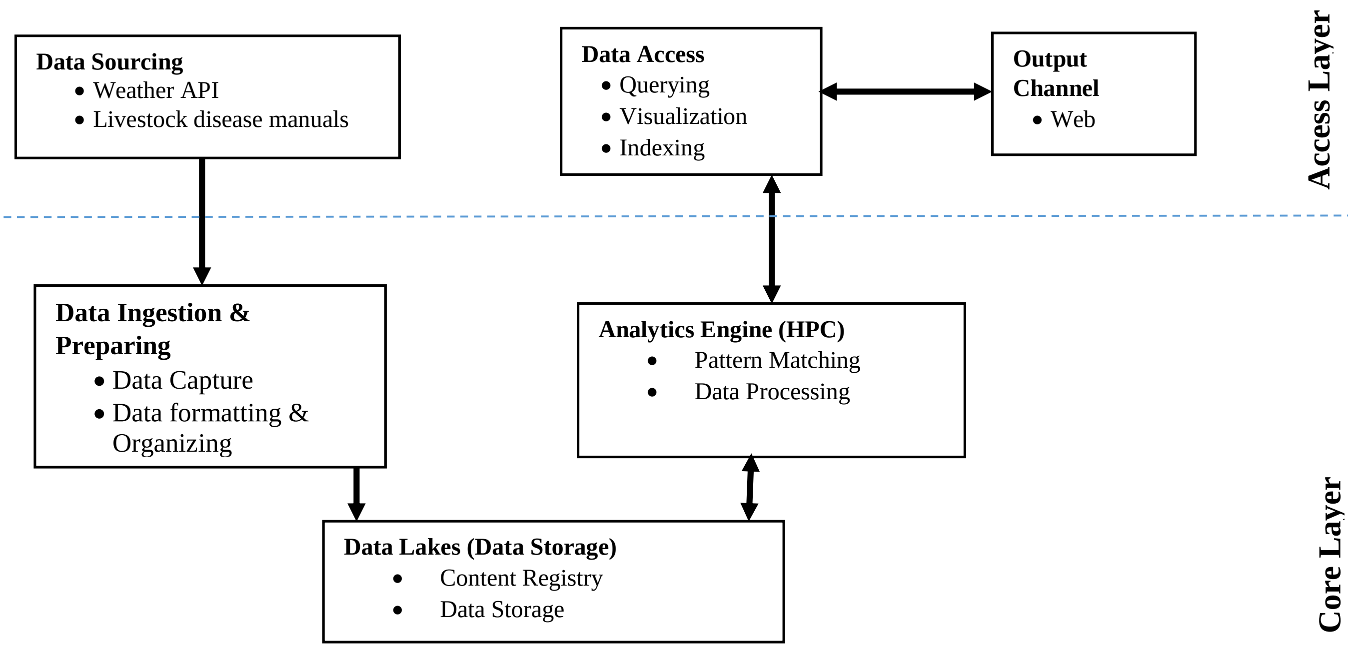

Figure 5: Conceptual framework

2.9 CRITIQUE OF PREVIOUS WORKS

2.9.1 Lack of relevant information

The biggest challenge with the existing surveillance system is the lack of timely and inadequate relevant ecological information on climate diseases (FAO,2017). The current methods can detect a small fraction of livestock diseases, leaving out the bigger picture. This further extends to making the whole system to be unreliable and makes it impossible to be adopted by stakeholders.

2.9.2 Lack of Capacity to Predict the Reoccurrence of the Diseases

The existing surveillance system lacks the capacity to handle massive amounts of datasets from various formats to perform predictive analytics. Farmers are faced with the challenges of not knowing when the priority diseases will re-emerge. Re-emerging of these diseases is usually not detected in an accurately, timely, and location-specific manner (ASTGS, 2019).

CHAPTER 3: METHODOLOGY

3.0 INTRODUCTION

The research study entailed developing a prototype that was used as a system test case covering all functional and non-functional requirements.

3.1 CONSTRUCTIVE RESEARCH

The constructive research method is a qualitative research paradigm that evaluates the "construct" being developed against some predefined criteria as a benchmark test to solve a persistent problem. The construct can be a new theory, algorithm, model, software, or framework developed to solve a specific ongoing problem. In this case, the construct is a prototype, and the persistent problem is predicting the future reoccurrence of contagious livestock diseases to prevent massive loss incurred by pastoral farmers. The diagram below highlights the steps used to formulate the study.

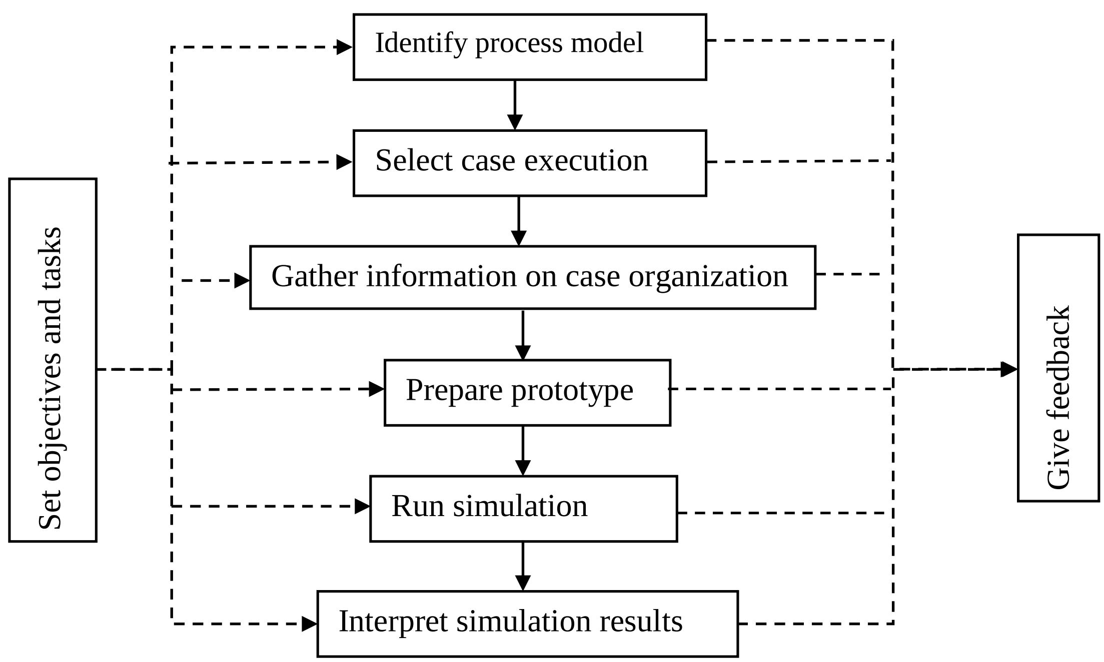

Figure 6: Steps in Constructive Research

3.2 TREND STUDY

This project studies the trend in the occurrence of livestock diseases. Trend analysis tend to answer what will occur in future by interpreting the historical pattern of what happened in the past and what is happening in the present (Kumar, 2019). The design of this project involved choosing data points from the past observed datasets and comparing them with current data to inform assumptions about future trends based on the data pattern.

The project used forecast modeling of predictive analytics and data-driven models to understand trends and patterns of the diseases.

3.3 SYSTEM DEVELOPMENT METHODOLOGY

This section outlines the method that was used to develop the surveillance and reporting system. The development process used agile methodology because of its flexibility to accommodate change and ability to produce end products faster.

The system has a farmer-facing interface module that interacts with the farmers and other stakeholders. This module was rapidly changing as the user demands increased. Thus, the agile methodology was appropriate.

3.4 AGILE METHODOLOGY

Agile methodology was appropriate for this project because of its ability to rapidly adapt to the dynamic needs of user requirements. The developed system was broken into separate independent modules implemented separately and later merged to form a complete system. The modules included reporting interface, surveillance dashboard, weather forecasting, and predictive dashboard. Changes done in one module were not affecting the operation of the other modules. In such a development environment, the agile development methodology is the most suitable (Rahimian & Ramsin, 2008).

Figure 7 below demonstrates the steps to develop the prototype using agile software development methodology.

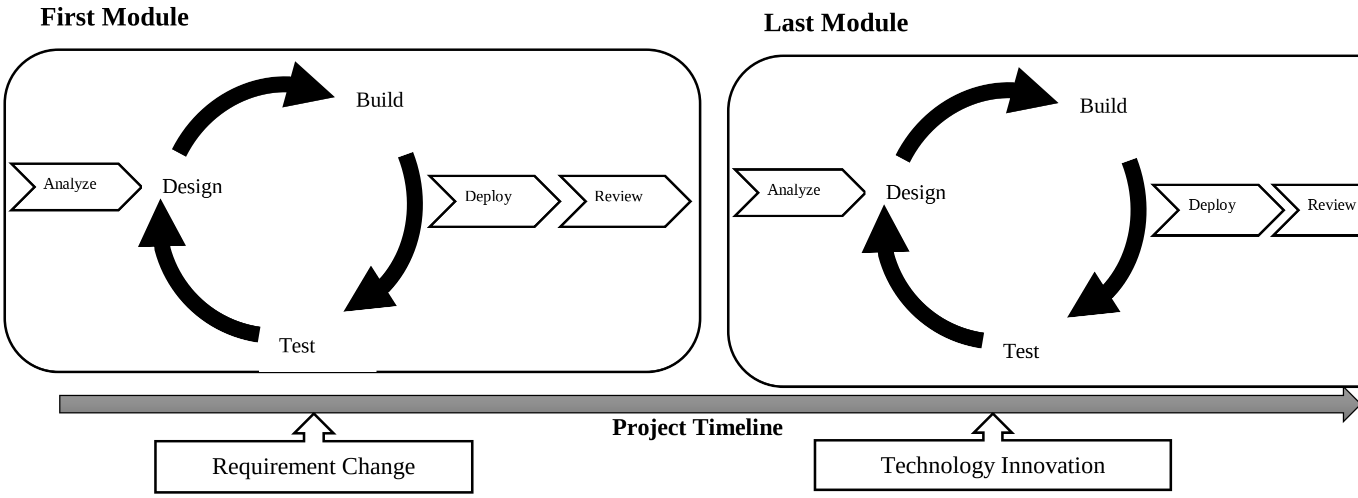

Figure 7: Design Phases for Agile Development

3.5 REQUIREMENT GATHERING

At this stage, needs assessments were gathered from the previous literature and informed the technical and functional requirements of the system. From this phase, the following were identified;

- Application services to be integrated.
- The user interface features to be used.
- Server requirements that would support the environment
- Usability of the system to different stakeholders

3.6. SYSTEM REQUIREMENTS SPECIFICATIONS

The objective of the developed prototype was to disseminate accurate early warning advisories on contagious livestock diseases in Kajiado county. The requirement specifications were derived from a combination of both functional and non-functional requirements.

3.6.1 Functional Requirements

a) Ability to accurately predict RVF, BEF, and Capripox virus reoccurrence in Kajiado county.
b) Ability to visualize disease-reported data in the past six months of each disease at the ward level.
c) Ability to analyze recorded field data to predict disease reoccurrence.
d) Ability to visualize the dynamic Map of high-risk areas and livestock disease.

3.6.2 Non-Functional Requirements

a) Scalability: the system was broken down into independent modules. Future requirements to increase additional diseases or counties can be incorporated by adding a new module that will be easily integrated into the overall system.
b) Usability: user interfaces were designed to suites all classes of stakeholders, including farmers, policymakers, and private stakeholders.
c) Reliability and availability: the system is hosted in a data center with a backup power supply and redundant computing resources for higher availability.

3.7 SYSTEM FEATURES AND SYSTEM COMPONENTS

a) Interactive Map: the system has a dynamic map of Kajiado county, showing the disease chances at the ward level.
b) Surveillance dashboard: the surveillance dashboard is the landing page of the system where current events on disease infection are recorded.
c) Mobile Application: this is the reporting tool in the form of a mobile application that community reporters use in the field.
d) Risk meter: based on the number of live disease events, the risk meter shows the intensity of a particular disease infection.
e) Prediction module: it displays the chances of future disease reoccurrence.

3.8 SYSTEM ARCHITECTURAL DESIGN

The expected system functionality was mapped from the system requirements. The requirement specifications were then used in designing the prototype, which enforced the pairing system components and sub-components. An overall system architecture was then derived, showing how different modules interact with one another.

Figure 8 below shows the architectural system design with different components and how they interact with one another

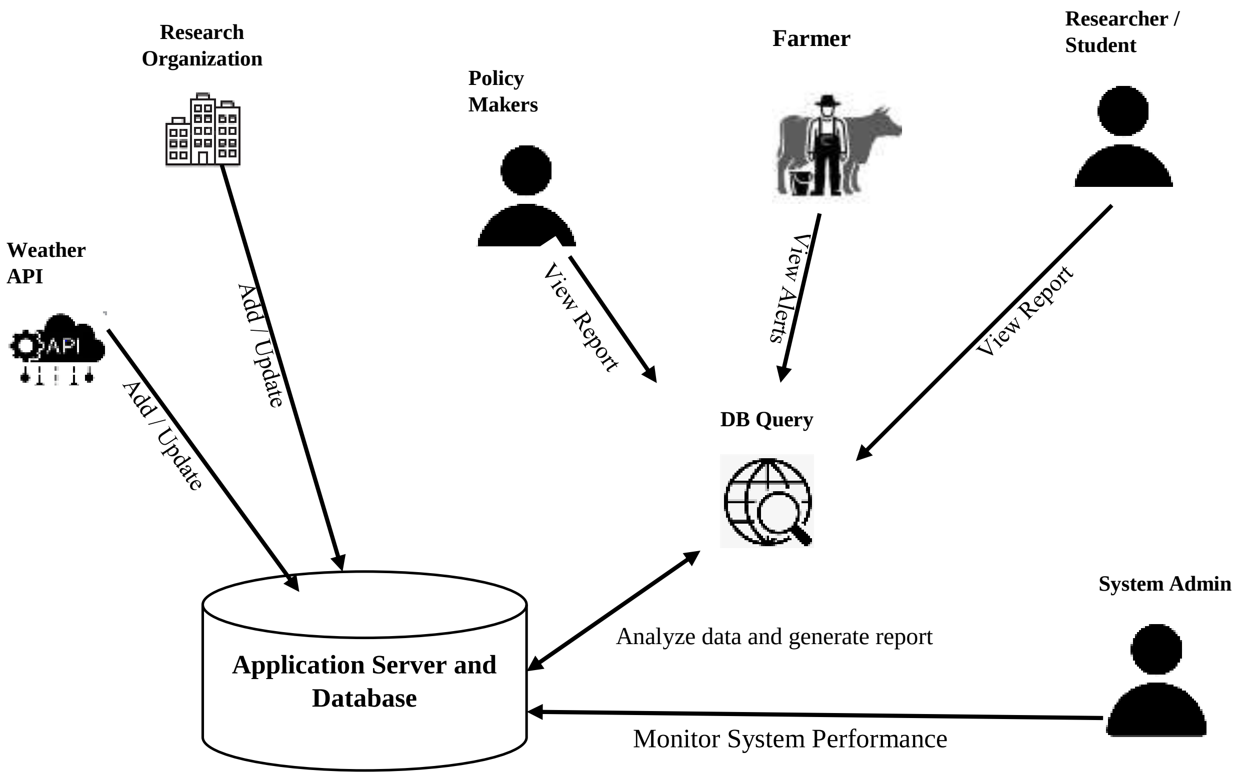

Figure 8: System Design

3.9 USE CASE DIAGRAM

Four types of system actors were identified with their roles in the system's operation. The figure below visualizes the system along with its main actors and their functions.

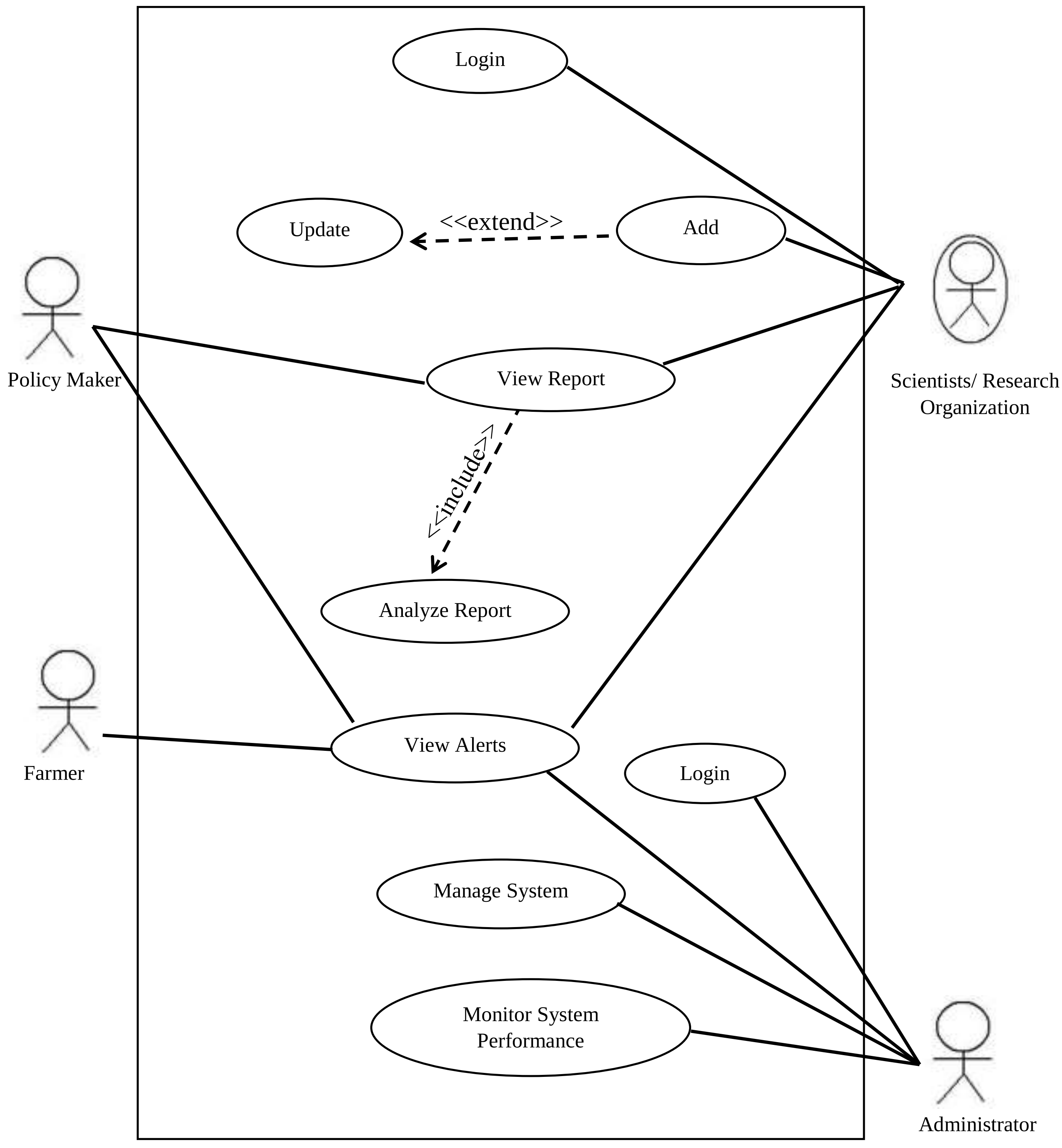

Figure 9: Unified Modeling Language (UML) diagram

3.10 STORY BOARDING

The user interfaces were designed to ensure new user interacting with the system for the first time can understand and use the system on their own in the shortest time possible. The figures below show a high-level graphical overview of the interfaces used in the system development.

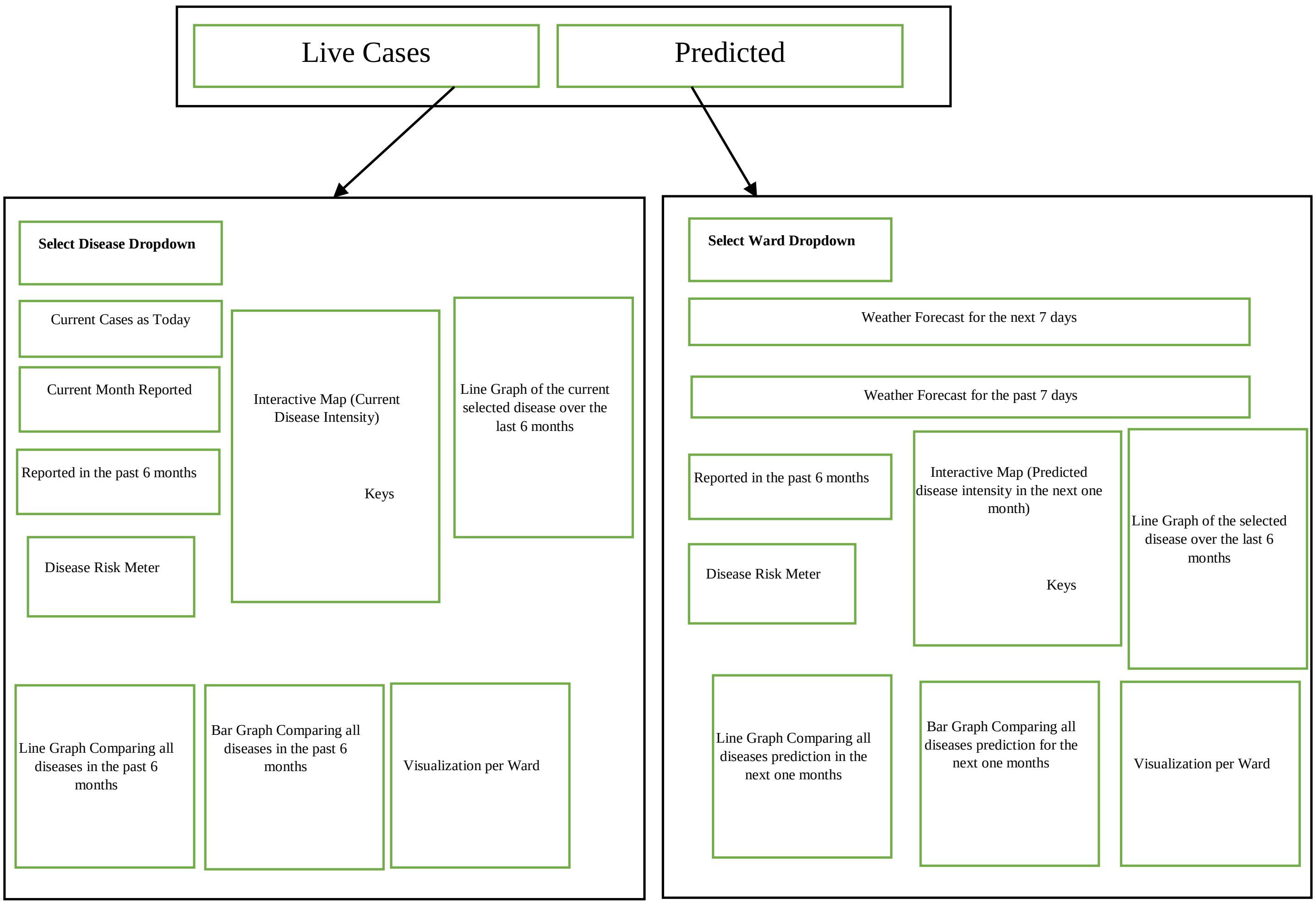

Figure 10: Story boarding of surveillance system

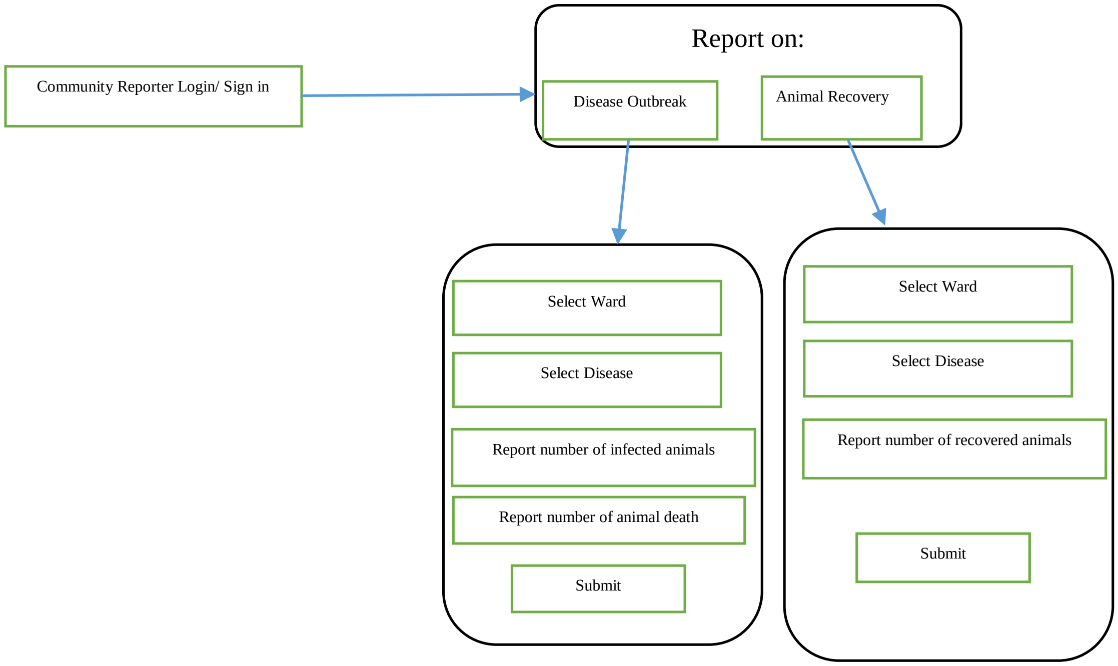

Figure 11: Story boarding of reporting system

3.11 DESIGN DECISION AND IMPLEMENTATION

From intervening in the system requirements, different technological frameworks and designs were prepared and implemented using the following software technologies;

- Surveillance System Dashboard – JavaScript, Hypertext Markup Language (HTML), and Cascading Style Sheets (CSS)
- Database – Postgre Sequential Query Language (PostgreSQL)
- Forecast Model - R
- Data Lakes – Django built on Python for Representational State Transfer (REST) API in the backend, JavaScript (React Library), HTML and CSS in the front end
- Reporting Application – Java, JavaScript

3.12 DATA PREPARATION FOR THE FORECAST MODEL

At the data ingestion, the model is fed with formatted data that contain all possible matches of the weather variables in a comma-separated value (.csv) file. The figure below shows some predefined data patterns that were ingested into the prototype.

Disease Name	Past Precipitation	Past Radiation	Past Temperatu	Temperature	Precipitation	Chances of Occurences
RVF	High	Low/ Medium	Low	Either	Either	High
RVF	High	Low/ Medium	Medium	Either	Either	Medium
RVF	High / Medium	Low/ Medium	High / Medium	Either	Either	Medium
BEF	High	Low/ Medium	Low	Either	Either	High
BEF	High	Low/ Medium	Medium	Either	Either	Medium
BEF	High / Medium	Low/ Medium	High / Medium	Either	Either	Medium
Capripox virus	High	Low/ Medium	Low	Low	High	High
Capripox virus	High / Medium	Low/ Medium	Low	Medium	Either	High
Capripox virus	High / Medium	Low/ Medium	High / Medium	Either	Either	Medium

Figure 12: Predefined data parameters for forecast model

3.13 USER INTERFACE MODEL IMPLEMENTATION

From the developed prototype, it was vital to have an active graphical interface model to enhance system usability. The screenshots below show the interactive interfaces for the surveillance and reporting system.

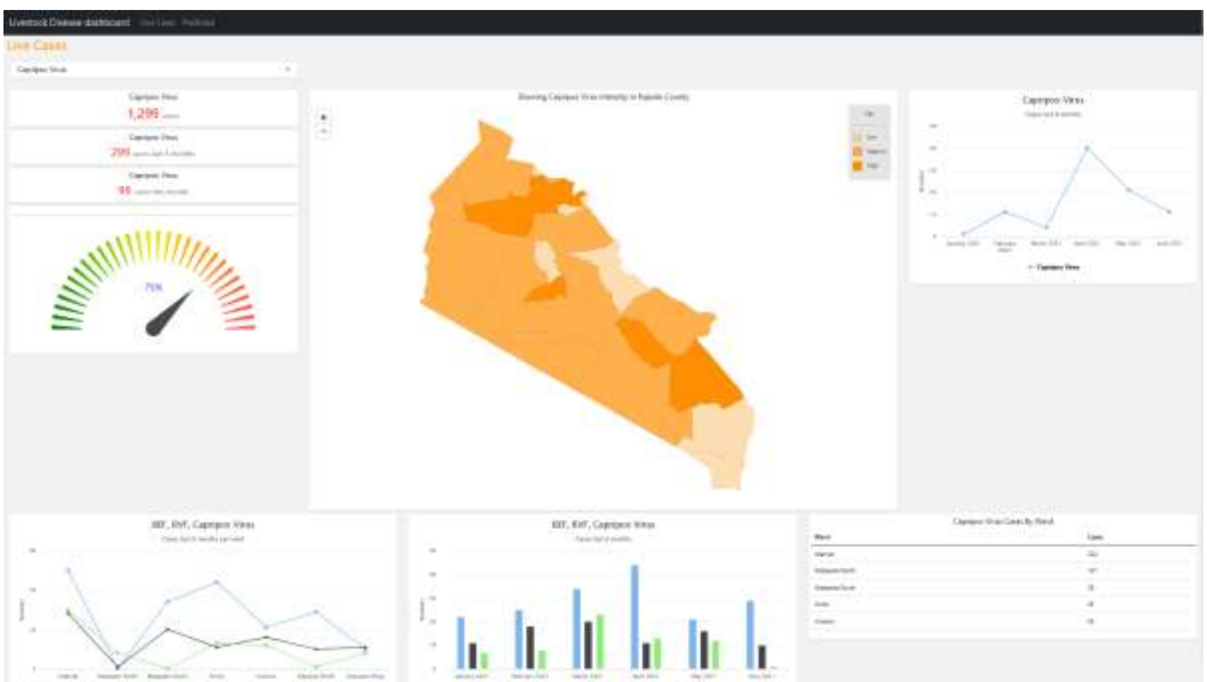

Figure 13: Surveillance Landing page

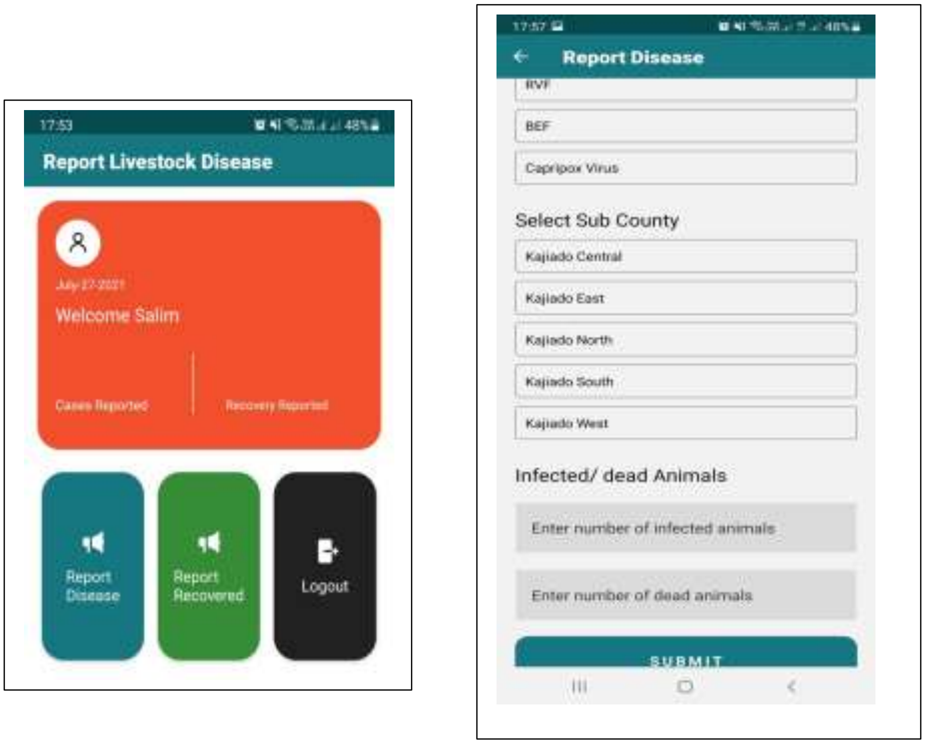

Figure 14: Livestock disease reporting tool

CHAPTER 4: RESULTS AND DISCUSSION

1.1 INTRODUCTION

This section showcases the results as per the prototype developed. Having analyzed the short-term weather pattern in the previous 30 days in Kajiado County, where the majority of citizens are livestock farmers, the results from the community reporters are in tandem with the observed data in the weather pattern.

The figure below shows a corresponding between the increased number of infected animals and the rapid shift in the weather pattern.

Data Lake Weather Reported

county	subcounty	ward	latitude	latitude	date	temperatures_max	temperatures_min	precipitation_amount	solar_
KAJIADO	Kajiado North Sub County	Olkeri Ward	-1.402358638	36.71319036	23/06/2021	21.18000031	11.47999954	0	3397.2
KAJIADO	Kajiado North Sub County	Olkeri Ward	-1.402358638	36.71319036	24/06/2021	22.88999939	12.43000031	0	4121.4
KAJIADO	Kajiado North Sub County	Olkeri Ward	-1.402358638	36.71319036	25/06/2021	20.87000084	12.43999958	0	3202.1
KAJIADO	Kajiado North Sub County	Olkeri Ward	-1.402358638	36.71319036	26/06/2021	22.11000061	11.78999996	0	3715.5
KAJIADO	Kajiado North Sub County	Olkeri Ward	-1.402358638	36.71319036	27/06/2021	16.55999947	12.51000023	0	3185.6
KAJIADO	Kajiado North Sub County	Olkeri Ward	-1.402358638	36.71319036	28/06/2021	22.21999931	12.64000034	0	3503.2

Reported

Phone	User Name	Sub County	Ward	Disease	Number infected	Dead	Date
0791836987	Daniel Albert	Kajiado North	Olkeri	RVF	76	4	2021-07-25T05:11:31Z
0791836987	Daniel Albert	Kajiado Central	Olkeri	BEF	83	1	2021-06-30T18:08:00Z
0791836987	Daniel Albert	Kajiado North	Olkeri	Capripox Virus	54	3	2021-06-30T05:22:28Z

Figure 15: Rapid shift in temperature causes significant increase in disease outbreak in Olkeri ward

The weather-based vector pathogens accelerate whenever there is an abrupt shift in the weather pattern. Apart from rainfall, farmers are not keen to observe other weather variables such as temperature; hence are caught unaware whenever there is an outbreak of diseases that are caused by the sudden change in the weather pattern. Disease unawareness results in subjecting the vulnerable herds to a longer period of infection before the disease is diagnosed. By extension, it increases the cost of curing and sometimes leads to death.

4.1 DATA-DRIVEN EVIDENCE TO PREDICT DISEASE OCCURRENCE

From the developed prototype, data is used as evidence to inform the likelihood of disease occurrence by accessing massive data sets from weather API, using data manipulation tools and High-Performance Computing (HPC) resources. The prototype uses a data-driven model housed in a cognified environment.

In the figure below, data-driven evidence is demonstrated by using different data sources from weather and remote sensor APIs to generate an informed decision on disease likelihood.

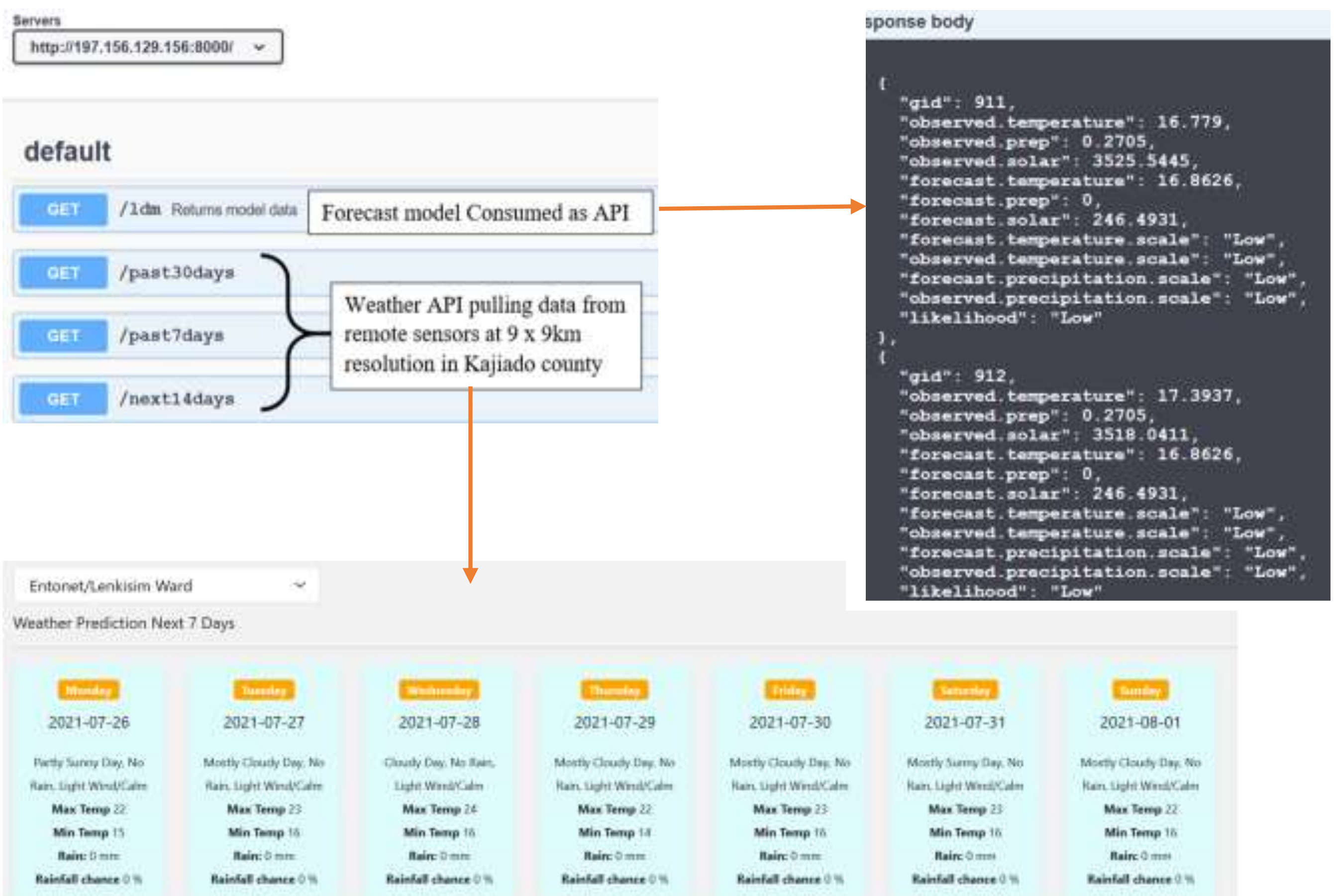

Figure 16: Distributed technology for data driven decision system

4.2 DYNAMIC MAP OF HIGH DISEASE RISK AREA

The prototype uses the predicted data generated from the forecast model and reported data from the community reporters to generate a map of high-risk areas. The figure below shows a screenshot of the interactive Map in the system.

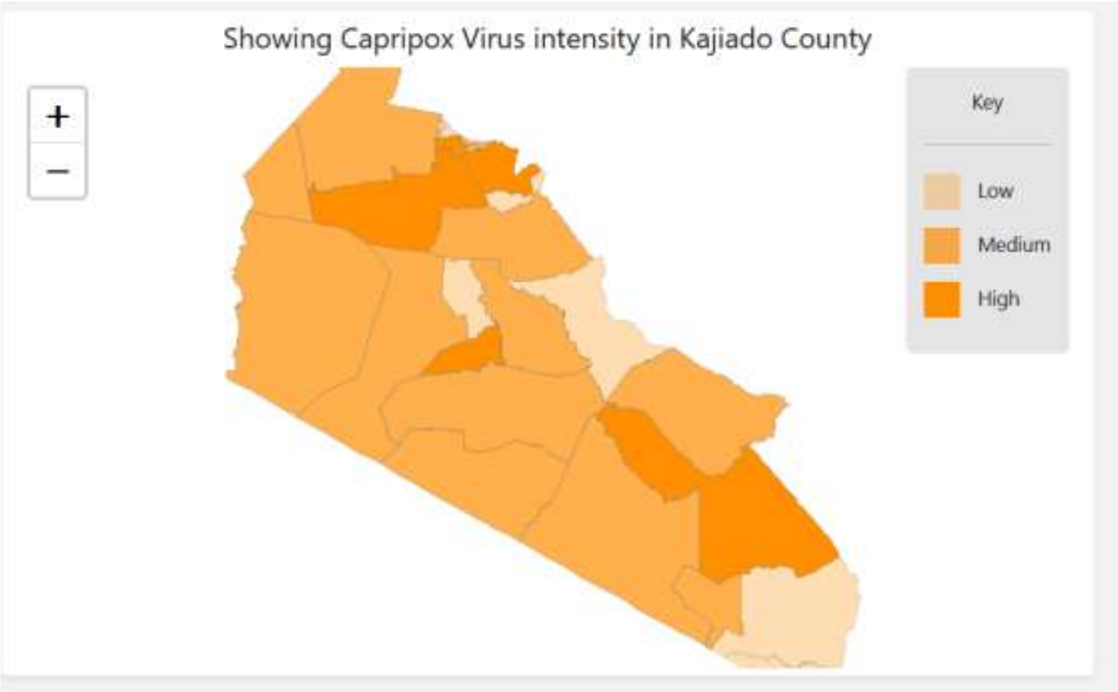

Figure 17: Interactive Map of Kajiado County

CHAPTER 5: SUMMARY, CONCLUSION, AND RECOMMENDATION

5.1 INTRODUCTION

This section contains a summary of key findings, challenges, and limitations encountered during the research, the conclusion reached, and the final recommendation on the research findings.

5.2 SUMMARY

The project focused on using cognified distribution technology to support a data-driven model for predicting diseases caused by changes in the weather pattern. The report from community reporters shows a spike in the number of infections in the same period the prototype under study had demonstrated a shift in the weather pattern.

From the prototype, Cognified distributed technology can provide easy access and manipulation of massive data sets from remote sensors such as virtual weather stations. It offered a framework to analyze massive real-time data in the High-Performance Computing (HPC) environment and be able to understand patterns and trends within the data.

In the forecast model, Real-time data fed from weather API is searched for unique data trends corresponding to weather patterns that accelerate disease outbreaks. Once there is a match, a flag is raised. When the model output is compared with reported data from the field, a similarity is observed in the increase in the number of infected animals over the same period.

The prototype is hosted in a data center with on-premise High Computing resources served with clean, stable power. The data center infrastructure was bought and configured once with perpetual software licenses.

5.2 CONCLUSION

Cognified distributed technology can handle massive datasets coming in different formats and magnitudes to generate insightful data trends that can inform decision-making. When the incoming data is well-formatted, data trends and patterns can be identified and applied in data-driven models to produce data evidence that can be used in agricultural research to predict the next reoccurrence of contagious diseases accurately. Accurately and timely prediction of the disease events reduces losses incurred by farmers and disease response costs by policymakers.

Developing countries can embrace and sustain cognified distributed technologies in their existing data center by applying the element of data science and machine learning to the existing computing resources.

5.3 RECOMMENDATION

1. The study recommends using cognified technology in the existing data center to be applied in veterinary epidemiology. This approach could remedy the challenges faced by livestock farmers by providing the following;
    a) Accurately, timely and location-specific advisory on when the next disease event will re-occur.
    b) Mitigating the effects caused by the contagious disease outbreak improves productivity and reduces disease response costs.
2. Use of data-driven models in agricultural research to generate data evidence for accurate prediction of the disease events.
3. The study also recommends using robust models to capture all environmental factors that accelerate disease incidences. These factors include nearness to water bodies, traceability of the herd, and other vector pathogen factors such as pests that can transmit disease from one animal to another. The incorporation of all the variables will increase the accuracy level. Robust computing resources are also needed to run long-term disease models by comparing disease patterns from the wider past duration of observed data. Predicting long-term chances requires a large volume of stored data that can go back to 35 years. This will require more computing resources to analyze.

REFERENCES

Agricultural Sector Transformation and Growth Strategy (ASTGS), 2019 – 2029.

Alhaji N. B., Ankeli P.I, Ikpa L.T., & Babalobi O.O. (2020) Contagious Bovine Pleuropneumonia: Challenges and Prospects Regarding Diagnosis and Control Strategies in Africa. *Vet Med (Auckl)*. 2020; 11:71-85.

Alhaji N. B., Babalobi O.O., & Isola T. O. (2018). A quantitative exploration of nomadic pastoralists' knowledge and practices towards Rift Valley fever in Niger State, North-central Nigeria: The associated socio-cultural drivers

Babaoglu O., & Sîrbu A. (2018). Cognified Distributed Computing. IEEE 38th International Conference on Distributed Computing Systems (ICDCS), Vienna, 2018, pp. 1180-1191, DOI: 10.1109/ICDCS.2018.00118

Carew-Reid, J., Prescott-Allen, R., Bass, S., & Dalal-Clayton, B. (2013). Strategies for National Sustainable Development (1st ed.). Taylor and Francis.

Charan J., & Biswas T. (2013). How to Calculate Sample Size for Different Study Designs in Medical Research?

Delia G., Bernard B., Johanna L., & Timothy R. (2015). Climate and Livestock Disease: assessing the vulnerability of agricultural systems to livestock pests under climate change scenarios.

FAO (2011). Challenges Of Animal Health Information Systems And Surveillance For Animal Diseases And Zoonoses. Proceedings of the international workshop organized by FAO, 23-26 November 2010, Rome, Italy. FAO Animal Production and Health Proceedings, No. 14. Rome, Italy.

FAO (2017). The future of food and agriculture – Trends and challenges. ISBN 978-92-5-109551-5

Fayisa W. (2020). Review on The Importance of Geographic Information System in Epidemiology: In Prevention and Control of Animal Disease. Int J Vet Sci Res 6(1): 047-051. DOI: 10.17352/ijvsr.000053

Fotis K., & Maria K. (2018). Decision support systems and strategic planning: information technology and SMEs' performance. International Journal of Decision Support Systems. 3. 53. 10.1504/IJDSS.2018.094260.

Gichamba A., Waiganjo P., Orwa D., Wario R., & Ngari B. (2016). Prototyping mAgriculture applications among smallholder farmers

Internet of things platform for smart farming: experiences and lessons learnt. Sensors 16 (11), 1884.

Jayaraman, P.P., Yavari, A., Georgakopoulos, D., Morshed, A., & Zaslavsky, A. (2016).

Kabir S. M. S. (2016). Formulating and Testing Hypothesis. Book Zone Publication, Chittagong-4203, Bangladesh

Kaliyaperumal S. K., Manoj K., & Arumugam S. (2015). Labeling Methods for Identifying Outliers

Kenya Climate Smart Agriculture Strategy (KCSAS), 2017-2026

Ketelsen J. C., Koponen J., Vinh M., Tilleard S., Quang T., & Joffre O. ((2013). Climate Change Impact and Adaptation Study on Livestock: Usaid Mekong Arcc Climate Change Impact and Adaptation Study. Prepared for the United States Agency for International Development by ICEM - International Centre for Environmental

Kaplan B. & Maxwell J. (2005). Qualitative Research Methods for Evaluating Computer Information Systems. 10.1007/0-387-30329-4_2.

Khanna A., & Kaur S. (2019). Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture

Kim J., Shah P., Gaskell J., Prasann A., & Musinga J. (2019) Scaling Up disruptive agricultural technologies in Africa. World Bank Group, Dalberg, Kore World Bank Partnership Facility

Kothari, C.R. (2004). Research Methodology: Methods and Techniques. 2nd Edition, New Age International Publishers, New Delhi.

Kumar, R. (2019). Research methodology: A step-by-step guide for beginners. Sage Publications Limited.

Lemon S. M., Hamburg M. A., Sparling P. F., Choffnes E. R., & Mack A. (2007). Ethical and Legal Considerations in Mitigating Pandemic Disease. Institute of Medicine; Board on Global Health; Forum on Microbial Threats, Rapporteurs ISBN-13: 978-0-309-10769-3ISBN-10: 0-309-10769-5

Loo M. P. J., & Jonge E. (2020) Data Validation

Makokha S., & Witwer M. (2013). Analysis of incentives and disincentives for live cattle in Kenya. Technical notes series, MAFAP, FAO, Rome.

Matere J., Simpkin P., Angerer J., Olesambu E., Ramasamy S., & Fasina F. (2009). Predictive Livestock Early Warning System: Monitoring for age condition and implications for animal production in Kenya

Mbaluka, D., & Okeyo, G. (2015). Application of Geographical Information System (GIS) and Mobile Application in Livestock Disease Management in Developing Countries. International Journal of Science and Research, 2015, 2319-7064.

Mengistu T., & Haile A. (2017). Review on the Application of Geographical Information Systems (GIS) in Veterinary Medicine. School of Veterinary Medicine, Wolaita Sodo University, Ethiopia.

Mugenda, O.M., & Mugenda, A.G. (2003). Research Methods: Quantitative and Qualitative Approaches (Revised Ed). Nairobi: African Centre for Technology Studies (ACTS)

Muhammad M., & Risma W. (2018). Prediction of Layer Chicken Disease using Fuzzy Analytical Hierarchy Process. International Journal of Engineering & Technology, 7 (2.26) (2018) 90-94

Ozalp B., & Alina, S (2018). Cognified Distributed Computing. IEEE 38th International Conference on Distributed Computing Systems

Peninah M., Austine B., Eric O., Emily P., & Josephat M., Athman M., & Samuel T. (2016). Prioritization of Zoonotic Diseases in Kenya, 2015. PLOS ONE. 11. 10.1371/journal.pone.0161576.

Perry B., & Grace D. (2019). The impacts of livestock diseases and their control on growth and development processes that are pro-poor

Rahimian, V., and Ramsin, R.(2008). Designing an agile methodology for mobile software development: A hybrid method engineering approach, second International Conference on Research Challenges in Information Science, 2008,RCIS 2008.vol. no., pp.337.

Randall E., & Sauter V. (2002). Decision Support Systems. University of Missouri St. Louis

Schultze, U., & Avital, M. (2011). Designing interviews to generate rich data for information systems research. Information and Organization, 21(1), 1-16.

White B. J., Amrine D. E., & Larson R. L. (2018) Big Data Analytics and Precision Animal Agriculture Symposium: Data to Decisions. Journal of Animal Science, Volume 96, Issue 4, April 2018, Pages 1531–1539,

World Health Organization (2006). Communicable disease surveillance and response systems Guide to monitoring and evaluating. WHO/CDS/EPR/LYO/2006.2

APPENDIX

1. Observed Weather Forecast and Next 14 days Forecast

```
kajiado_wards = read.csv('data/kajiado_wards.csv')
forecast.final = data.frame(gid = integer(),
                        temp = double(),
                        prep = double(),
                        solar =double())

observed.final = data.frame(gid = integer(),
                        temp = double(),
                        prep = double(),
                        solar =double())
# Load key
apiCredentials = aWhereAPI::load_credentials("credentials/Credentials.txt")
apiKey <- '******************************'
apiSecret <- '**************************'

#for (i in 1:nrow(kajiado_wards)){

no_cores <- detectCores() - 1
cl <- makeCluster(no_cores, type='PSOCK')
registerDoParallel(cl)

forecast.data = foreach::foreach(i=c(1:nrow(kajiado_wards))
                        ,.packages = c("aWhereAPI" ,'data.table','dplyr')
                        ,.export = c('awhereEnv75247')
                        ,.init = forecast.final
                        ,.combine = rbind) %dopar% {
                         apiCredentials =
aWhereAPI::load_credentials("credentials/Credentials.txt")
                         tokenToUse <- apiCredentials$token
                         lat = kajiado_wards[i,'lat']
                         lon = kajiado_wards[i,'lon']
                         gid = kajiado_wards[i,'gid']
                         #Pulling the forecast data
                         forecast = aWhereAPI::forecasts_latlng(lat,
                                        lon,
                                        day_start = as.character(Sys.Date()),
                                        day_end = as.character(Sys.Date()+13),
                                        block_size = 1
                                        ,keyToUse = apiKey
```

```
                                        ,secretToUse = apiSecret
                                        ,tokenToUse = tokenToUse)
                    forecast$gid = gid


                    #forecast = data.frame(gid = gid, temp = forecast$temp,
prep=forecast$prep, solar = forecast$solar)
                    #forecast = data.frame(gid = gid, latitude = forecast$latitude,
longitude=forecast$longitude, date = observed$date
                    #  ,temperatures.max = forecast$temperatures.max,
temperatures.min=forecast$temperatures.min, precipitation.amount =
forecast$precipitation.amount
                    # ,solar.amount = forecast$solar.amount,
relativeHumidity.max=forecast$relativeHumidity.max, relativeHumidity.min =
forecast$relativeHumidity.min
                    # ,wind.morningMax = forecast$wind.morningMax,
wind.dayMax=forecast$wind.dayMax, wind.average = forecast$wind.average)

                    # forecast.final = rbind(forecast.final,forecast)

                    return(forecast)
                   }
stopImplicitCluster()


#summarise
forecast = forecast.data %>% dplyr::select(gid,startTime,temperatures.value,
precipitation.amount,
                         solar.amount) #%>%

forecast = forecast %>% mutate(startTime = substr(startTime, 1,10)) %>% as.data.table()

forecast[, startTime :=  as.Date(startTime, format = "%Y-%m-%d")]


forecast = forecast %>% group_by(startTime,gid) %>%
  summarise(temp = mean(temperatures.value),
        prep = sum(precipitation.amount),
        solar = mean(solar.amount))

forecast = forecast %>% group_by(gid) %>%
  summarise(temp = mean(temp),
        prep = mean(prep),
```

```
          solar = mean(solar))

no_cores <- detectCores() - 1
cl <- makeCluster(no_cores, type='PSOCK')
registerDoParallel(cl)


observed.data = foreach::foreach(i=c(1:nrow(kajiado_wards))
                        ,.packages = c("aWhereAPI" ,'data.table','dplyr')
                        ,.export = c('awhereEnv75247')
                        ,.init = observed.final
                        ,.combine = rbind) %dopar% {
                         apiCredentials =
aWhereAPI::load_credentials("credentials/Credentials.txt")
                         tokenToUse <- apiCredentials$token
                         lat = kajiado_wards[i,'lat']
                         lon = kajiado_wards[i,'lon']
                         gid = kajiado_wards[i,'gid']

                         #Pulling the observed data

                         obs_startdate = as.character(Sys.Date()-31)
                         obs_enddate = as.character(Sys.Date()-1)

                         observed = aWhereAPI::daily_observed_latlng(latitude = lat,
                                                   longitude = lon,
                                                   day_start = obs_startdate,
                                                   day_end = obs_enddate
                                                   ,keyToUse = apiKey
                                                   ,secretToUse = apiSecret
                                                   ,tokenToUse = tokenToUse)
                         observed$gid = gid

                         #observed = data.frame(gid = gid, latitude = observed$latitude,
longitude=observed$longitude, date = observed$date
                         #          ,temperatures.max = observed$temperatures.max,
temperatures.min=observed$temperatures.min, precipitation.amount =
observed$precipitation.amount
                         #          ,solar.amount = observed$solar.amount,
relativeHumidity.max=observed$relativeHumidity.max, relativeHumidity.min =
observed$relativeHumidity.min
                         #          ,wind.morningMax = observed$wind.morningMax,
wind.dayMax=observed$wind.dayMax, wind.average = observed$wind.average)
```

```
                    # observed.final = rbind(observed.final,observed)

                    return(observed)
                   }
stopImplicitCluster()

observed = observed.data %>% dplyr::select(gid,temperatures.max,temperatures.min,
precipitation.amount,
                              solar.amount) %>%
  mutate(temperatures.value = (temperatures.max+temperatures.min)/2) %>%
  group_by(gid) %>%
  summarise(temp = mean(temperatures.value),
          prep = mean(precipitation.amount),
          solar = mean(solar.amount))


weather = merge(observed,forecast, by = 'gid')

weather = rename(weather
            ,observed.temperature= temp.x
            ,observed.prep = prep.x
            ,observed.solar = solar.x
            ,forecast.temperature= temp.y
            ,forecast.prep = prep.y
            ,forecast.solar = solar.y)

kajiado_wards = kajiado_wards %>% dplyr::select(gid,county,subcounty,ward)
kajiado_wards$subcounty = str_replace(kajiado_wards$subcounty, 'Sub County','')
kajiado_wards$ward = str_replace(kajiado_wards$ward, 'Ward','')

observed.data = merge(kajiado_wards, observed.data, by = 'gid')
forecast.data = merge(kajiado_wards, forecast.data, by = 'gid')


observed.data = observed.data %>%
dplyr::select(gid,county,subcounty,ward,date,temperatures.max, temperatures.min
                                  ,precipitation.amount,solar.amount)

forecast.data = forecast.data %>%
dplyr::select(gid,county,subcounty,ward,startTime,endTime,temperatures.value,temperatures.ma
x, temperatures.min
                                  ,precipitation.chance,precipitation.amount,solar.amount)
```

```
observed.data = observed.data %>%
dplyr::rename(temperatures_max=temperatures.max,temperatures_min=temperatures.min,precip
itation_amount=precipitation.amount
                                    ,solar_amount=solar.amount)

forecast.data = forecast.data %>% dplyr::rename(temperatures_value =
temperatures.value,temperatures_max=temperatures.max,temperatures_min=temperatures.min

,precipitation_chance=precipitation.chance,precipitation_amount=precipitation.amount
                                    ,solar_amount=solar.amount)
```

2. Forecast Model Algorithm

```
setwd('C:/Users/salim.kinyimu/Programming/LivestockModel')
library(plumber)
#setwd('/home/ldm')
#source('C:/Users/salim.kinyimu/Programming/LivestockModel/scripts/weather.R')


#* @apiTitle Model API and Raw Datasets
#* @apiDescription API for accessing model and weather data


#* Returns model data
#* @param disease Select the type of disease to model
#* @get /ldm


ldm = function(disease){
  source('C:/Users/salim.kinyimu/Programming/LivestockModel/scripts/weather.R')
  disease = as.character(disease)
## Temperature scale
weather$forecast.temperature.scale = with(weather, ifelse(forecast.temperature < 17,
'Low',
                        ifelse(forecast.temperature >=17 & forecast.temperature
<=20,'Medium', 'High')))
```

```
weather$observed.temperature.scale = with(weather, ifelse(observed.temperature < 17, 'Low',
                          ifelse(observed.temperature >= 17 & observed.temperature<=22,'Medium', 'High')))

## Add precipitation scale

weather$forecast.precipitation.scale = with(weather, ifelse(forecast.prep < 1, 'Low',
                                   ifelse(forecast.prep >= 1 & forecast.prep <= 3,'Medium', 'High')))

weather$observed.precipitation.scale = with(weather, ifelse(observed.prep < 1, 'Low',
                                           ifelse(observed.prep >= 1 & observed.prep <= 3,'Medium', 'High')))
if (disease == 'RVF'){
weather$likelihood = with(weather,ifelse(observed.precipitation.scale == 'High' & observed.temperature.scale == 'Low', 'High',
                             ifelse(observed.precipitation.scale %in% c('High','Medium') & observed.temperature.scale %in% c('High','Medium'), 'Medium', 'Low' )))
} else if (disease == 'BEF'){
  weather$likelihood = with(weather,ifelse(observed.precipitation.scale == 'High' & observed.temperature.scale == 'Low', 'High',
                              ifelse(observed.precipitation.scale %in% c('High','Medium') & observed.temperature.scale %in% c('High','Medium'), 'Medium', 'Low' )))
} else if (disease == 'Capripox virus'){
  weather$likelihood = with(weather,ifelse(observed.precipitation.scale == 'High'
                              & observed.temperature.scale == 'Low'
                              & forecast.precipitation.scale == 'High'
                              & forecast.temperature.scale == 'Low'
                              | observed.precipitation.scale %in% c('High','Medium')
                              & observed.temperature.scale == 'Low'
```

```
                                        & forecast.temperature.scale == 'Medium', 'High',
                                        ifelse(observed.precipitation.scale %in% c('High','Medium') &
observed.temperature.scale %in% c('High','Medium'), 'Medium', 'Low')))
} else
  weather = paste0(disease, ' Not available')
weather
}

#* @get /past30days

historical = function(){
  #source('/home/ldm1/scripts/weather.R')
  observed.data
}
#* @get /past7days

hist7 = function(){
  #source('/home/ldm1/scripts/weather.R')
  observed.data = as.data.table(observed.data)
  observed.data[, date :=  as.Date(date, format = "%Y-%m-%d")]

  observed.data = observed.data %>% filter(date >= as.character(Sys.Date()-7))
  observed.data
}
#* @get /next14days

forecast = function(){
  #source('/home/ldm1/scripts/weather.R')
  forecast.data
}
```

3. **Dashboard**

```
import { createContext, useState } from "react";
import allCase from "../../Data/allCases.json";
import befCases from "../../Data/cbppCases.json";
import rvfCases from "../../Data/pprCases.json";
import capripoxCases from "../../Data/fmdCases.json";

import BEFMapCase from "../../Data/CBPP.json";
import RVFMapCase from "../../Data/PPR.json";
import CapripoxapCase from "../../Data/FMD.json";

export const LivestockDiseaseContext = createContext();

export const LivestockDiseaseContextProvider = (props) => {
  const [diseaseData, SetDiseaseData] = useState(BEFMapCase);
  const [allCasesDataDisease, SetAllCasesDataDisease] = useState(befCases);
  const [focusedDisease, SetFocusDisease] = useState("");
  const [percentageDisease, SetPercentageDisease] = useState(0.9);

  const changeMapDisease = (val) => {
    // console.log(val)
    if (val === "BEF") {
      SetDiseaseData(BEFMapCase);
      SetAllCasesDataDisease(befCases);
    } else if (val === "RVF") {
      SetDiseaseData(RVFMapCase);
      SetAllCasesDataDisease(rvfCases);
    } else if (val === "Capripox Virus") {
      SetDiseaseData(CapripoxapCase);
      SetAllCasesDataDisease(capripoxCases);
    }
```

```
    // else{
    //     SetDiseaseData(BEFMapCase)
    //     SetAllCasesDataDisease(befCases)
    // }
  };
  return (
    <LivestockDiseaseContext.Provider
      value={{
        diseaseData,
        changeMapDisease,
        allCasesDataDisease,
        percentageDisease,
      }}
    >
      {props.children}
    </LivestockDiseaseContext.Provider>
  );
};
```

4. Disease Prediction

```
import { createContext, useState, useEffect, useContext } from "react";
import befCases from "../../Data/predictedData/cbppCases.json";
import rvfCases from "../../Data/predictedData/pprCases.json";
import capripoxCases from "../../Data/predictedData/fmdCases.json";

import BEFMapCase from "../../Data/predictedData/CBPP.json";
import RVFMapCase from "../../Data/predictedData/PPR.json";
import CapripoxapCase from "../../Data/predictedData/FMD.json";
import { AxiosGetService } from "../../Services/AxiosServices/AxiosServices";
import { GeocoodinatesContext } from "../GeoLocationContext/GeoLocationContext";
import { SubDivisionsContext } from "../SubDivisions/SubDivisionsContext";
import {
```

```
  last_1_day,
  last_seven_day,
  next_seven_days,
  today,
} from "../../Services/AxiosServices/common";
// import { authPromise, getJobResults, request2 } from "../../Constants/awhere";

export const PredictedDiseaseContext = createContext();

export const PredictedDiseaseContextProvider = (props) => {
  const { wardLat, wardLong } = useContext(SubDivisionsContext);
  const { latitude, longitude } = useContext(GeocoodinatesContext);
  const [diseaseData, SetDiseaseData] = useState(BEFMapCase);
  const [allCasesDataDisease, SetAllCasesDataDisease] = useState(befCases);
  const [focusedDisease, SetFocusDisease] = useState("");
  const [percentageDisease, SetPercentageDisease] = useState(0.9);
  const [foreCast, setForeCast] = useState([]);
  const [foreCastLast7Days, setForeCastLast7Days] = useState([]);

  const [isLoading, setIsLoading] = useState(null);
  const queryObservation = "observations";
  const queryWeather = "forecasts";

  const lat = wardLong;
  const long = wardLat;

  const getWeatherForeCast = async () => {
    setIsLoading(true);
    try {
      const get = await AxiosGetService(
```

```
`https://selector.kalro.org/api/awhere/predictionFuture/?queryType=${queryWeather}&la
t=${lat}&long=${long}&startDate=${today}&endDate=${next_seven_days}`
      );
      // console.log(await get)
      setForeCast(await get.data.forecasts);
      setIsLoading(false);
    } catch (e) {
      setIsLoading(false);
    }
  };

  const getWeatherForeCastLast7Days = async () => {
    setIsLoading(true);
    try {
      const get = await AxiosGetService(

`https://selector.kalro.org/api/awhere/predictionFuture/?queryType=observations&lat=${l
at}&long=${long}&startDate=${last_seven_day}&endDate=${last_1_day}`
      );
      console.log(await get);
      setForeCastLast7Days(await get.data.observations);
      setIsLoading(false);
    } catch (e) {
      setIsLoading(false);
    }
  };

  const changeMapPredDisease = (val) => {
    // console.log(val)
    if (val === "BEF") {
```

```
    SetDiseaseData(BEFMapCase);
    SetAllCasesDataDisease(befCases);
  } else if (val === "RVF") {
    SetDiseaseData(RVFMapCase);
    SetAllCasesDataDisease(rvfCases);
  } else if (val === "Capripox Virus") {
    SetDiseaseData(CapripoxapCase);
    SetAllCasesDataDisease(capripoxCases);
  }
  // else{
  //     SetDiseaseData(BEFMapCase)
  //     SetAllCasesDataDisease(befCases)
  // }
};

useEffect(() => {
  getWeatherForeCast();
  getWeatherForeCastLast7Days();
  //   effect
  // console.log(request2)
  // request2()
  // getJobResults('/v2/weather/locations/1.2,36/norms/07-07')
  return () => {
    //   cleanup
  };
}, [wardLat, wardLong]);
return (
  <PredictedDiseaseContext.Provider
    value={{
      diseaseData,
      changeMapPredDisease,
```

```
        allCasesDataDisease,
        percentageDisease,
        foreCast,
        isLoading,
        foreCastLast7Days,
      }}
    >
      {props.children}
    </PredictedDiseaseContext.Provider>
  );
};
```