Adventist University of Central Africa

E-TUNGO SYSTEM

Case Study: Ministry of Agriculture and Animal Resources

A Final Year Project

Presented in partial fulfillment of the

requirements for the degree of

BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY

Major in

SOFTWARE ENGINEERING

By

VIERRY SHEMA

August, 2023

ABSTRACT

Adventist University of Central Africa

Project for bachelor's degree in Information Technology

Emphasis in Software Engineering

Title: E-Tungo System

Name of the researcher: Vierry Shema

Name of faculty Advisor: Mr. MANIRAHO Laurent

Date Completed: August 2023

The objective of the proposed E-Tungo Management System for the Ministry of Agriculture and Animal Resources (MINAGRI) in Rwanda is to improve and modernize the monitoring, control, and management of animal health and resources in the country. The system aims to enhance disease monitoring, enabling timely detection of outbreaks and effective implementation of control measures. It will facilitate efficient communication and collaboration among veterinarians, sector officers, and farmers, enabling coordinated efforts to control diseases and provide necessary care. The system also seeks to streamline the process of obtaining animal transportation permissions, ensuring smooth movement, and minimizing delays. By providing access to information and resources, it empowers farmers with knowledge on disease prevention and animal health, improving productivity and welfare. Additionally, through digitization and automation, the system aims to enhance overall efficiency, productivity, and resource allocation, contributing to strengthened food security by safeguarding animal health and minimizing disease-related losses.

The technologies we have used in this system are PHP, HTML, CSS, JavaScript, and bootstrap for the design and MySQL as database.

DECLARATION

I, **Vierry SHEMA**, hereby declare that this is my original work except where references have been made and has never been submitted for an award to this Institution, or any other university or institution.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented.

Signature……

Date…….

APPROVAL

I, **Mr. MANIRAHO Laurent** hereby certify that I have supervised this report and it's now ready to be submitted for the award of a Degree in Information Technology Major Software Engineering.

Signature……

Date…….

DEDICATION

To the Almighty God,

My parents

My supervisor,

My adoring siblings,

And all of my friends and coworkers.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AUCA : Adventist University of Central Africa

CD's : Compact Disc

CASE : Computer-Assisted Software Engineering

CSS : Cascading Style Sheet

DBMS : Data Base Management System

HTTP : Hyper Text Transfer Protocol

OOA : Object Oriented Analysis

OOD : Object Oriented Design

OOM : Object Oriented Methodology

OOP : Object Oriented Programming

OMG : Object Management Group

PK : Primary key

FK : Foreign Key

UML : Unified Modeling Language

GUI : Graphic User Interface

UML : Unified Modelling Language

SDP : Software Development Process

OOAD : Object Oriented Analysis and Design

PHP : Hypertext Preprocessor

ACKNOWLEDGMENT

I am grateful to my supervisor Mr. MANIRAHO Laurent who guided me in the accomplishment of this work, especially for his substantial advice, professional assistance, guidance, and precious ideas.

I also thank the Adventist University of Central Africa lectures for the best educational skills that I have gained through them. I thank the Administration of Adventist University of Central Africa as well as the staff of the Department of Information Technology.

A special mark of honor goes to my parents who supported me during academic period in one way or another in the accomplishment of my study, especially for their substantial advice and prayers.

No one is truly self-made, thus my special thanks also goes to my colleagues at AUCA, especially for their encouragement and invaluable assistance.
May the Almighty God bless you All.

CHAPTER 1

GENERAL INTRODUCTION

Rwanda has emerged as a leading advocate for using technology to showcase its economic prowess, resulting in significant advancements in various aspects of daily life. The country has witnessed remarkable progress in areas such as transportation payment systems, with the introduction of smart cards and mobile phone-based transactions for everyday activities. In agriculture, services like the Agriculture Land Information System and Agricultural Market Information System have revolutionized the sector.

However, the animal resources services sector has faced challenges in providing accessible and up-to-date information and resources in Rwanda. This is a matter of concern, especially for farmers who encounter difficulties in accessing certain services related to animal resources. As a student who has benefited from an AUCA-intensive education, I have taken it upon myself to address this issue by leveraging technology to provide a solution. This chapter gives an overview of the background of the study, statement of the problems, objectives of the study, research questions, and scope of the study and significance of the study.

Background of the Study

Animals play a vital role in our daily lives, serving as companions, sources of food, and contributors to the overall ecological balance. Their significance extends far beyond simple companionship and sustenance, encompassing a wide range of essential contributions that impact various aspects of human existence.

First and foremost, animals hold a special place in our hearts as cherished companions. Animals, such Cows, goats, hens, and pigs are specific animals that hold great importance in our daily lives. These animals provide us with essential food resources, contribute to agricultural sustainability, and support various industries. Their role in our daily lives highlights the importance of ensuring their health and well-being through effective disease control measures. Implementing proper animal disease control systems is crucial to safeguarding their welfare, preventing disease outbreaks, and maintaining the quality and safety of the animal-derived products we rely on.

Problem Statement

The Department in charge of animal resources is responsible for promoting the development and sustainable management of animal resources in Rwanda. This includes the protection and control of animal diseases, animal welfare, and the promotion of the animal products industry. Therefore, it is the primary responsibility of this department to ensure that there are adequate systems and resources in place for the effective monitoring and control of animal diseases.

If there are issues related to the provision of animal health services, including delays in disease detection and control, inefficient communication and information sharing, and delays in animal transportation, it is the responsibility of the Department of Animal Resources to address these issues.

This department should ensure that there are adequate resources and systems in place for the effective provision of animal health services, including the implementation of an electronic system for disease monitoring and control.

The current manual system for animal health services in Rwanda, particularly in the Department of Animal Resources, faces several challenges that hinder the effective monitoring and control of animal diseases, leading to inefficiencies, delays, and potential economic losses for farmers. These challenges include:

- Delayed disease detection: Manual processes result in delayed disease detection and response, leading to economic losses for farmers.
- Inefficient communication: Manual communication hampers coordination among veterinarians and timely care for sick animals.
- Lack of data analysis: The absence of a tracking system hinders identifying disease patterns and implementing effective control strategies.
- Time-consuming record-keeping: Manual record-keeping for animal transportation is prone to errors and causes delays.
- Limited visibility and oversight: Inconsistent processes and insufficient oversight result in confusion and delays for farmers.
- High costs and resource challenges: Travel and scheduling costs burden farmers, while limited visibility affects resource allocation.
- Insufficient response to outbreaks: Delays and coordination difficulties lead to inadequate disease control measures.

Choice and Motivation

I am driven by my passion to apply my skills towards solving societal problems. With the education I have received from the Adventist University of Central Africa (AUCA), I aim to create a system that will provide a solution for farmers in Rwanda. The successful implementation of this research project will not only assist farmers in accessing crucial information and services, benefiting both them and the Ministry, but it will also fulfill the academic requirements for me to obtain a bachelor's degree from AUCA. Additionally, the development of this system will enhance my programming abilities, enabling me to excel in various aspects of life beyond my academic studies.

Objectives of the study

My study has both main and specific objectives as they are mentioned below.

General Objective

The general objective of the system is to create a comprehensive and efficient platform that facilitates communication, collaboration, and disease control measures among farmers, veterinarians, and sector officers in Rwanda. It aims to improve the overall management of animal health and disease prevention strategies.

Specific Objectives

- Provide tools for sector officers to monitor disease control and make informed decisions.
- Enable sector officers to grant transport permissions for animals, ensuring regulation.
- Establish a communication platform for veterinarians to share information and coordinate efforts.
- Enable veterinarians to issue disease testing certificates for accurate documentation.
- Allow veterinarians to post and escalate disease cases, facilitating prompt action.
- Provide sector officers with the ability to reject permission requests, ensuring scrutiny.
- Empower farmers to report animal disease cases for early detection.
- Give farmers up-to-date information on animal status and trending diseases.

- Offer farmers access to disease prevention information for proactive practices.
- Streamline farmers' requests for animal transportation permission.
- Enable veterinarians to post and track disease outbreaks for data-driven decisions.
- Facilitate communication among sector officers for coordinated efforts.
- Provide farmers with information on the destination area for animal transportation.
- Create a collaborative platform for veterinarians to communicate and share expertise.

Scope of the research

The scope of the research encompasses the development and implementation of a comprehensive system that addresses the needs of farmers, veterinarians, and sector officers in Rwanda. The research aims to create a platform that facilitates effective communication, coordination, and disease control measures within the animal resources sector.

Challenges

Creating a computerized system, even for a small organization, is a complex undertaking. It necessitates significant expertise, attention to detail, and ample time to fulfill customer requirements adequately.

The primary obstacle encountered during web application development is the insufficient experience in application development.

When it comes to gathering data, various methods such as interviews, observations, and research projects are necessary. However, the research project might not achieve its objectives if an interviewee provides incomplete or inaccurate information.

Likewise, if the observation process does not proceed as planned, the research's execution may suffer.

Methodology and Techniques Used in the Study

Observation

Observation is a method of documenting the actions and interactions of individuals, objects, and events without engaging in direct questioning or communication. This valuable technique allows for a deeper understanding of the current system by closely observing behaviors. However, it is important to note that people tend to alter their behavior when they are aware of

being observed. In the specific case mentioned, the study aimed to analyze how farmers acquire animal transport permissions and how they learn about animal health (Paul,2018).

Interview

This is a conversation between the interviewer and the interviewee where a set of questions are asked to elicit information. This technique was appropriate in order to get in-depth information about how farmers get animal transport permission?

Expected Results

After developing and testing the system, I am expecting the following results:

- Sector officers monitor disease control and make informed decisions.
- Sector officers provide transport permissions for animals.
- Veterinarians communicate, share information, and coordinate efforts.
- Veterinarians issue disease testing certificates.
- Veterinarians post and escalate disease cases.
- Farmers report animal disease cases.
- Farmers receive information on animal status and diseases.
- Farmers access disease prevention and control measures
- Farmers request animal transport permissions.
- Veterinarians post and track animal disease cases.
- Sector officers communicate and coordinate efforts.
- Farmers receive area status for animal transport.
- Veterinarians collaborate on disease control measures.

Organization of Report

The research comprises five chapters:

1. Introduction: Provides background, problem statement, objectives, scope, and contribution.
2. Current system analysis and proposed solution.
3. Requirements analysis and design of the new system using UML diagrams.
4. Implementation: Covers system development, deployment, and evolution.
5. Conclusions, recommendations, and future work.

These chapters structure the research, addressing various aspects from introduction to implementation and concluding with recommendations for future studies.

CHAPTER 2

ANALYSIS OF EXISTING SYSTEM

Introduction

Understanding a problem is crucial for effective problem-solving. Analyzing the current system (as-is system) provides a detailed understanding of its workings and identifies existing problems. This analysis guides the optimization or replacement of the system to address those problems and improve outcomes.

The system enables sector officers to monitor disease control, make informed decisions, and allows veterinarians to communicate, share information, coordinate efforts, and issue certificates. Farmers can report cases, access animal information, request transport permissions, and receive updates. Veterinarians have a platform for communication and collaboration on disease control.

Description of the Current System

The Department in charge of animal resources is responsible for promoting the development and sustainable management of animal resources in Rwanda. This includes the protection and control of animal diseases, animal welfare, and the promotion of the animal products industry. Therefore, it is the primary responsibility of this department to ensure that there are adequate systems and resources in place for the effective monitoring and control of animal diseases. If there are issues related to the provision of animal health services, including delays in disease detection and control, inefficient communication and information sharing, and delays in animal transportation, it is the responsibility of the Department of Animal Resources to address these issues. This department should ensure that there are adequate resources and systems in place for the effective provision of animal health services, including the implementation of an electronic system for disease monitoring and control.
This is how the current system works:

Farmer

- Go to the sector office.
- Ask for an appointment with a veterinary.
- Provide access to the animals for testing.
- Ask for transportation certificate

Veterinary

- Test animals for diseases
- Animal health monitoring
- Prevention of animal diseases
- Communication and education of citizens
- Record-keeping

Sector officer

- Coordination and communication to veterinaries
- Monitoring and reporting to the MINAGRI
- Record-keeping
- Animal transportation permission

Historical Background

The Ministry of Agriculture and Animal Resources (MINAGRI) of Rwanda has played a crucial role in the country's development since its inception. Following Rwanda's independence in 1962, the government established a Ministry of Agriculture and Forestry to oversee the sector's development.

However, during the 1994 genocide, much of the country's infrastructure was destroyed, including its agricultural systems. After the conflict, the government recognized the need for a more focused approach to agriculture and re-established the Ministry of Agriculture and Animal Resources in 1997.

Since then, MINAGRI has implemented various initiatives to improve food security, increase agricultural production, and reduce poverty. These efforts have included promoting the use of high-yield crops, introducing modern farming techniques, and providing training and support to farmers. In recent years, MINAGRI has also placed a strong emphasis on developing the country's

livestock sector, including the promotion of animal health and the implementation of policies to encourage investment in the sector.

Overall, MINAGRI has played a crucial role in Rwanda's economic development, particularly in the agricultural sector, and continues to work towards the country's long-term growth and sustainability.

Mission

The Ministry of Agriculture and Animal Resources (MINAGRI) of Rwanda has the mission to transform agriculture and animal resources into a sustainable, modern, and market-oriented sector that contributes to the country's socio-economic development.

Vision

The vision of the Ministry of Agriculture and Animal Resources (MINAGRI) of Rwanda is to build a modern, resilient, and sustainable agricultural sector that can contribute significantly to the country's economic growth, poverty reduction, and food security.

Modeling of the Current System

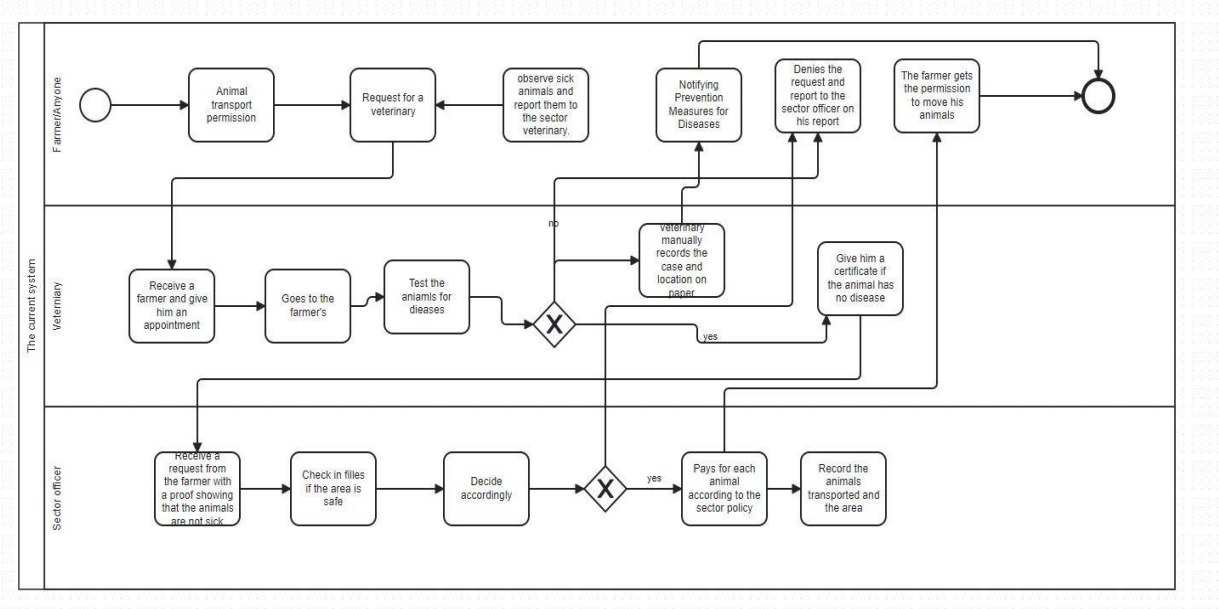


Figure 1: Current system analysis

Farmers currently rely on visiting the sector office to obtain information and services, such as permission certificates for animal transport. Furthermore, sector officers face difficulties in communication when there are disease outbreaks in areas where animal transportation is prohibited. The process of getting a veterinarian to test animals is time-consuming, and obtaining approval for animal transport requests can be challenging and time-consuming as well. Farmers also lack real-time information on animal diseases and prevention measures, and when they need to report a new case, they typically have to visit the sector office and meet with the sector veterinarian. MINAGRI (Ministry of Agriculture) faces challenges in effectively knowing and monitoring animal diseases.

Problems of the Existing System

Performance

Throughput

- The manual process may be prone to delays and bottlenecks, which could affect its overall efficiency and throughput.
- Scheduling appointments with the sector veterinary cause delays due to limited availability or other obligations.
- Certification process may also have its own delays, depending on the workload of the sector office.
- The capacity of the sector office to handle and process requests may also impact on the system's throughput.

Response Time

- Scheduling appointments with the sector veterinary, which may have limited availability or long wait times.
- The time required for the veterinary to physically test the animals for diseases and issue certificates.
- Delay in identifying disease outbreaks. Lack of information for decision-making
- Difficulty in tracking disease spread.
- Difficulty in identifying affected animals in their areas.

Information

Input

- Lack of standardization or consistency in the information provided by farmers seeking permission, which may lead to errors or confusion during the review process.
- Manual data entry and paper-based records, which can be time-consuming and error-prone.
- Lack of digital records or data storage, which may make it difficult to track and analyze data over time.
- Inaccurate information.
- Difficulty in analyzing data.

Output

- Lack of visibility or transparency into the review process, which may make it difficult for farmers to understand why their requests were denied.
- Manual issuance of certificates or permission documents can be time-consuming and may lead to errors or delays.
- Lack of digital records or data storage, which may make it difficult to track and analyze data over time.
- Delayed reporting of disease outbreaks.
- Incomplete data.

Storage

- Papers are kept in the files by sector officers.

Economics

Farmers

- Time and cost associated with traveling to the sector office and scheduling appointments with the sector veterinary.
- Cost of veterinary services and certification process, which may be a burden for smaller farmers.
- Delays in obtaining permission to transport animals, which may impact on their livelihoods and profitability.
- Increased costs.
- Difficulty in transporting animals.

Sector officers

- Time and cost are associated with reviewing and processing animal transportation requests.
- Lack of visibility into the overall workload and demand for animal transportation permission, which may impact resource allocation and planning.
- Potential for errors or delays in the review process, which may impact overall efficiency and customer satisfaction.
- Inaccurate information
- Reduced public trust if disease outbreaks are not addressed effectively or if there is a perception that sector officers are not doing enough to protect animal health, this can

erode public trust in the agricultural sector. This can have negative consequences for the industry.

- Difficulty in identifying disease outbreaks.

Veterinary

- Time and cost associated with traveling to farms to conduct disease testing and issue certificates.
- Potential for high demand for veterinary services during peak seasons, which may impact their ability to serve all customers.
- Cost of certification equipment and supplies, which may be a burden for smaller veterinary practices.
- Reduced effect Inaccurate information activeness of control measures.
- Difficulty in identifying disease outbreaks.
- Reduced effectiveness of control measures

Control

- Lack of standardization or consistency in the review process, which may lead to errors or confusion during the review process.
- Limited oversight or monitoring of the review process, which may make it difficult to identify errors or areas for improvement.
- Potential for human error or bias in the review process, which may lead to inconsistencies or unfair treatment.
- Difficulty in tracking and analyzing data over time, which may make it difficult to identify patterns or areas for improvement.
- Inadequate response
- Difficulty in coordinating response
- Inadequate re Lack of information sources.
- Control measures.

Efficiency

The current system is not efficient for the: farmers, sector officers and the veterinaries, therefore it needs improvement.

Service

The manual approach to animal health monitoring and disease control leads to various adverse effects on services:

- Delays in detecting and responding to disease outbreaks due to time-consuming data collection and analysis. This results in increased disease spread and economic losses for farmers.
- Inability to identify trends and patterns in animal disease outbreaks, making it challenging to provide appropriate preventative care and implement effective control measures.
- Slow and inefficient communication between veterinarians, hindering their ability to coordinate efforts during disease outbreaks and provide timely care to sick animals.
- Time-consuming and error-prone record-keeping for animal transportation, causing delays in transportation and posing potential health risks to animals.

Proposed solutions

The E-Tungo System aims to address the deficiencies of the current manual system by introducing digitalization and automation. The system will have user-friendly online portals enabling real-time communication and information sharing. Standardized data entry and reporting mechanisms will ensure accuracy, and a centralized database with analytics capabilities will enable proactive disease control strategies. Enhanced farmer engagement, increased resources, and collaborations with relevant stakeholders will strengthen the overall system.

Functional Requirements

Functional requirements are basic facilities that the system should provide to end users. Essentially, they are what the system does or must not do, and can be thought of in terms of how the system responds to inputs. Functional requirements usually define if/then behaviors and include calculations, data input, and business processes.

- The system shall allow sector officers to monitor disease control measures and make informed decisions on disease prevention strategies.
- The sector officer shall be able to provide the transport permission for animals.
- The system shall allow veterinarians to communicate and share information with other veterinarians to coordinate efforts to control disease outbreaks and provide appropriate care to sick animals.
- The veterinarians shall be able to provide a certificate after testing an animal from disease.
- The system shall allow farmers to report animal disease cases in their area.
- The system shall provide farmers with information on the status of animals in their area, including information on sick animals and trending diseases.
- The system shall allow farmers to access information on disease prevention measures and control measures.
- The system shall allow farmers to request permission to transport animals from one place to another.
- The system shall allow veterinarians to post animal disease cases and track disease outbreaks.
- The system shall provide farmers with the status of the area they want to transport their animals.
- The system shall provide a platform for veterinarians to communicate and collaborate on disease control measures.
- The system shall allow farmers to access information about their animal vaccination status.

Non-Functional Requirements

Non-functional requirements are the requirements that define the criteria that can be used to judge the performance of the system rather than the specific behaviors. Non-functional requirements describe how the system should work and in which conditions the system should be allowed to work. It defines system attributes such as security, reliability, performance, maintainability, scalability, and usability. Below are the non-functional requirements of the E-Tungo System:

Performance

- the system must support 5,000 users per hour.
- The system must provide 6 seconds or less response time in a chrome desktop browser.

Scalability

- the system must be scalable enough to support 10,000 users at the same time while maintaining optimal performance.

Portability

- the system must be able run on window 10 or window 11 without change in its behavior and on an android phone.

Reliability

- The System must perform without failure in 95 percent of use cases during a month**.**

Maintainability

- the mean time to restore the system (MTTRS) following a failure must not be greater than 10 minutes.

Security

- All data inside the system shall be protected against malware attacks or unauthorized access.

CHAPTER 3

REQUIREMENTS ANALYSIS AND DESIGN OF THE NEW SYSTEM

Introduction

To construct a resilient and prosperous system that aligns with user needs, a solid foundation is essential, necessitating meticulous analysis and design. System development encompasses two primary constituents: thorough analysis and meticulous design. By delving into detailed examination and thoughtful planning, the groundwork for a robust and triumphant system can be established.

System development is made up of 2 main components:

- System analysis
- System Design

System Design

Design involves the systematic process of determining the elements, modules, interfaces, and data of a system, all aimed at meeting specific requirements.

System Analysis

The process of decomposition in design involves breaking down a system into its constituent parts to determine the effectiveness of their interactions in achieving the defined requirements.

Element of a system:

- **Architecture:** this is the concept that defines the structure, behavior, and more views of a system. We can use flowcharts to illustrate the architecture.
- **Modules:** These are components that handle one specific task in a system. A combination of modules makes up the system.
- **Components:** This provides a function or group of related functions. They are made up of modules.
- **Interfaces:** This is the shared boundary across which the components of the system exchange information and relate.
- **Data:** This is the management of the information and data flow.

Analysis and Design Methodology

Object Oriented Methodology (OOM)

Several methodologies exist for developing information systems, including Data Structure-Oriented, Object-Oriented, Prototyping, and more. However, our focus here is on ObjectOriented programming.

Object-Oriented Methodology (OOM) is a software development approach that promotes and simplifies the reuse of program components. By employing this technique, a computer system can be constructed using a modular approach, enabling the efficient reuse of existing components and facilitating their sharing with other systems.

In Object-Oriented Methodology, the fundamental building blocks for system development are Classes and Objects. These elements play a crucial role in constructing a system that adheres to the principles of Object-Oriented Methodology.

Object: is a mix of variables, functions, and data structures that represents a specific instance of a class. A thing, a notion, or an event can all be objects.

Class: is a description of a collection of objects with common attributes and behaviors. A class is divided in three parts as shown below:

Class Name
Attributes
Operations ()

- The upper part holds the name of the class.
- The middle part contains the attributes of the class.
- The last part gives the method or operation the class can take or undertake.

A named property of a class that defines a range of values that instances of the property may have been called an attribute. A method is the implementation of a service that may be requested by any object to impact behavior in the class.

Unified Modelling Language (UML)

The Unified Modeling Language (UML) is a versatile modeling language used in software engineering. Its primary objective is to establish a standardized approach for representing system architecture. UML offers a set of notations that enable the creation of models and is specifically designed to be implemented by CASE (Computer-Aided Software Engineering) tools.

It is important to note that while UML models inherently imply certain techniques, UML itself does not provide any specific modeling methods. Instead, it serves as a visual syntax, providing a standardized way to construct models within the software engineering domain.

UML Model

A model is a representation that serves as an example or guide to follow. Using a model is often easier than referring to the actual reality because it focuses on the essential aspects and disregards unnecessary details.

In the context of UML, various models are available to describe different aspects of a system, such as use case diagrams, class diagrams, and sequence diagrams. However, it is not mandatory to create all the models for a system. Instead, it is recommended to develop the models that effectively visualize the system and convey the necessary information.

During the analysis phase, key questions regarding system usage, functionality, and deployment are answered. The project initiator examines any existing systems explored in the previous chapter, identifies opportunities for improvement, and develops a conceptual plan for the new system.

Requirement Analysis

Requirement analysis refers to the process of identifying the expectations of users for a new or modified product. These expectations, also referred to as criteria, must possess certain qualities. They should be quantifiable, pertinent, and precise. In software engineering, these requirements are often referred to as functional specifications.

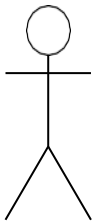
Design of the New System

Use Case Diagram

A use case diagram depicts a system's functionality in terms of actors, their objectives (represented by use cases), and the interactions between actors and use cases. A use case model consists of the following elements:

● Actor

When engaging with a system directly, an actor specifies a role that some external entity takes. It might reflect a user role or another system's role in interacting with a system.



Actor Name

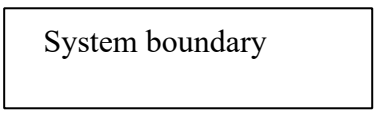
● **Use Case**

A use case is a sequence of activities or event stages that specify how a role (also known as an actor in the Unified Modelling Language) interacts with a system to achieve a goal. It's written like this:

● **Relationship**

An association symbol in UML that represents meaningful links between actors and use cases.

● **System boundary**



The figure below describes the operations of new system and the stakeholders through the use case

E-Tungo Use case diagram.

Figure 2: Use Case diagram

Use case description.

A use case's description explains what it does and what it requires to be properly performed. This is how each use case looks:

- Name: a use case's title
- Description: the goal of a system
- Actor: the user case's actor
- Post-condition: when the use case is completed, the system state
- Normal flow: the use case's real steps
- Alternative flow: processes that may occur if a typical flow is interrupted.

Use case description to Create Veterinary Account

Name: Create account
Actors: Administrator
Description: The Administrator creates a new account for a veterinarian in the system.
Pre-condition: The Administrator has the necessary privileges to create new accounts.
Post-condition: The veterinarian account is successfully created.
Normal Flow: 1. The Administrator accesses the account creation functionality. 2. The Administrator enters the required information for the veterinarian account. 3. The system validates the entered information. 4. The system creates the veterinarian account. 5. The system notifies the Administrator about the successful account creation.

Alternative flow: 3. a. if the information is not valid the system shows the error message. 3. b. The actor can decide to go back to the beginning of the main flow or cancel the login, at that point the use case ends.

Table 1: Create Veterinary Account

Use case description for Provide Transport Permission

Name: View platform categories
Actors: Sector Officer
Description: The Sector Officer reviews and grants permission for animal transportation.
Pre-condition: The sector officer has access to the permission review functionality.
Post-condition: The transport permission is either granted or rejected.
Normal flow: 1. The Sector Officer accesses the permission review functionality. 2. The Sector Officer retrieves the request for transport permission. 3. The Sector Officer reviews the request and associated information. 4. Based on the review, the Sector Officer either grants or rejects the transport permission. 5. The system notifies the farmer about the decision.
Alternative flow: The sector officer may reject the permission due to some factors.

Table 2: Provide Transport Permission

Use case description for Communicate and Share Information

Name: Communicate and Share Information
Actors: Veterinarian
Description: Veterinarians communicate and share information with other veterinarians for coordinated disease control efforts.
Pre-condition: The veterinarian has access to the communication and information sharing functionality.
Post-condition: Relevant information is effectively communicated and shared among veterinarians.
Normal flow: 1. Veterinarian accesses the communication and information sharing platform. 2. Veterinarians send messages to other veterinarians. 3. Veterinarian receives messages from other veterinarians. 4. Veterinarian shares disease updates, control strategies, or treatment protocols. 5. Veterinarians engage in discussions and collaborate on disease control measures.
Alternative flow: If the section is not available, the system shows an error.

Table 3: Communicate and Share Information

Use case description for Access Information of Animal Vaccination Status

Name: Use case description for Access Information of Animal Vaccination Status
Actors: Farmer

Description: Farmers can access information about the vaccination status of their animals.
Pre-condition: The farmer has access to the system and the functionality to access animal vaccination information.
Post-condition: The farmer receives information about the vaccination status of their animals.
Normal Flow: 1. Farmer logs into the system and accesses the vaccination status functionality. 2. Farmer selects specific animals or the entire herd for vaccination information. 3. System retrieves and presents the vaccination records or history of the selected animals. The system presents the farmer with the vaccination status, including the vaccines administered, dates, and any upcoming vaccinations. 4. The farmer can review the information and ensure that their animals are up-to-date with the required vaccinations. 5. Based on the provided information, the farmer can take appropriate actions, such as scheduling future vaccinations or contacting a veterinarian for further guidance.
Alternative flow: 7. a. if the information is not valid the system shows the error message. 7. b. Based on the provided information, the farmer can take appropriate actions, such as scheduling future vaccinations or contacting a veterinarian for further guidance.

Table 4: Access Information of Animal Vaccination Status

Class Diagrams

A class diagram is a visual representation of a system's static structure, showcasing classes, properties, methods, and object relationships. It provides insights into the system's design perspective by displaying classes, interfaces, collaborations, and their connections.

The followings are components of a class diagram:

Classes

A class diagram illustrates classes and their relationships. It uses rectangles with three compartments:

Top: Class name (centered, bolded, capitalized).

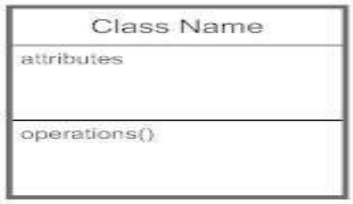
Middle: Attributes (left-aligned, lowercase).

Bottom: Operations (methods or functions).

Associations show connections between classes.

Active Classes

Passive classes retain data and service other classes, whereas active classes begin and regulate the flow of activity. Active courses should have a thicker border.



Active class

Visibility

Visibility markers in class diagrams indicate how information within a class is accessed:
- (Private): Hidden from external classes.
+ (Public): Accessible to all classes.

Visibility

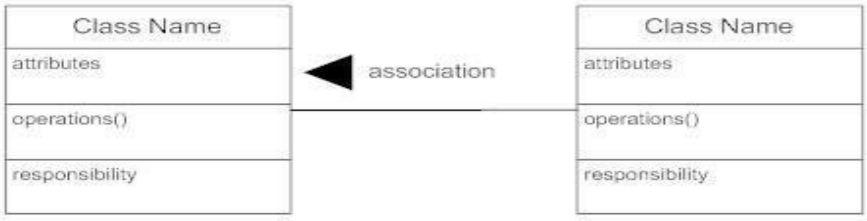
Marker	Visibility
+	public
-	private
#	protected
~	package

(Protected): Accessible to child classes.

Associations

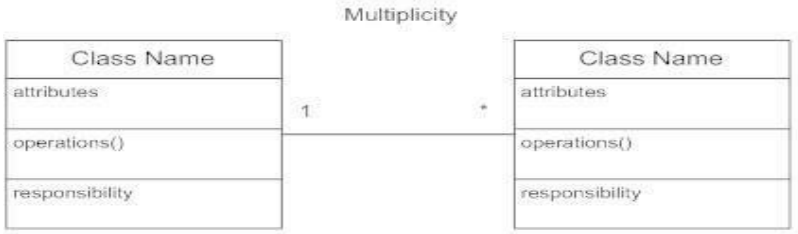
The static connections between classes are represented by associations. Above, above, or below the association line, write the names of the associations.

To show which way the connection is going, use a full arrow. Roles should be placed at the end of a chain. The two classes' perceptions of one other are represented through roles.



Multiplicity (Cardinality)

Place multiplicity notations near the ends of an association. These symbols indicate the number of instances of one class linked to one instance of the other class.



Constraint

Place constraints inside curly braces {}.

Composition and Aggregation

Composition: Strong ownership, Class A (whole) and Class B (component) connected with a filled diamond.

Aggregation: Weaker connection, "whole" class is important, represented by a hollow diamond. Both point to "total" class (aggregation).

Generalization

Inheritance or a "is a" connection are both examples of generalization. It's a connection between two classes in which one is a specialized version of the other.

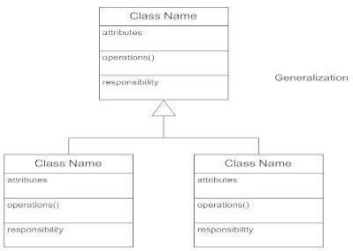


Figure 2: Class Diagram

The diagram below shows the class diagram.

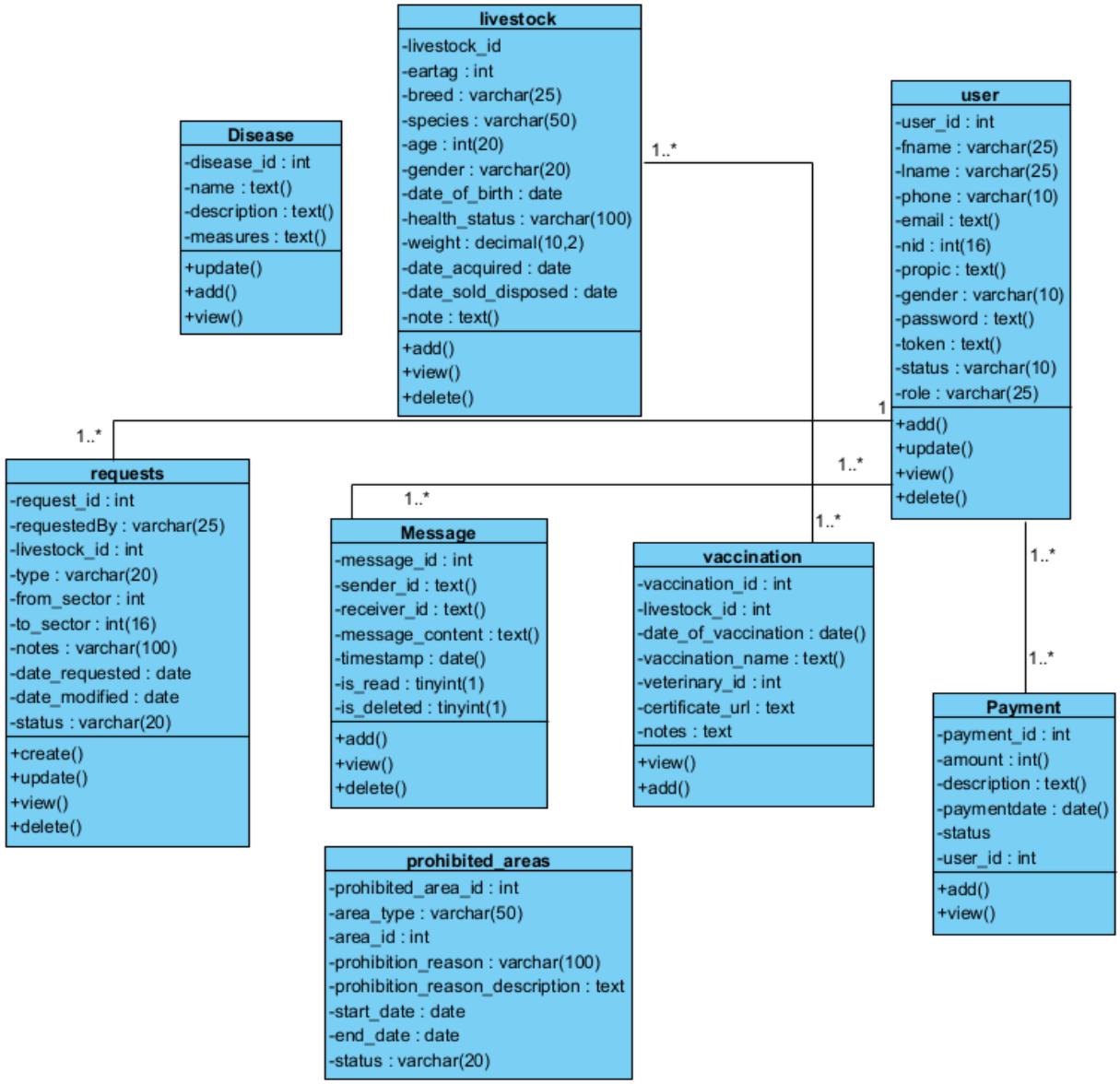


Figure 3: Class diagram

Sequence Diagram

A sequence diagram is a dynamic model that depicts the explicit message sequence that is sent between objects in a defined interaction. Sequence diagrams are highly useful for comprehending real-time requirements and complicated use cases because they stress the time-based ordering of the activity that occurs among a group of objects.

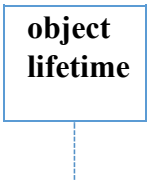
Term and definition	Symbol
An object: It participates in a sequence by sending and/or receiving messages. It is placed across the top of the diagram.	
An activation: Is a lifeline a long, thin rectangle on top? It indicates whether an item is transmitting or receiving messages.	
A message: It is used to transfer data from one item to another. An operation call is shown by a solid arrow and the message being sent, whereas a return is indicated by a dashed arrow and the value being returned.	message call   

Table 5: Symbols used in sequence diagram.

Create Account

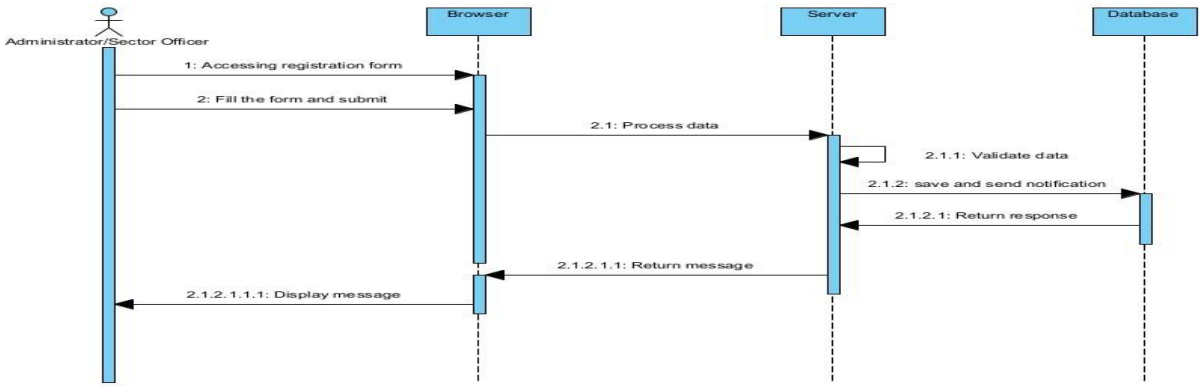


Figure 4: Create account sequence diagram.

Provide permission.

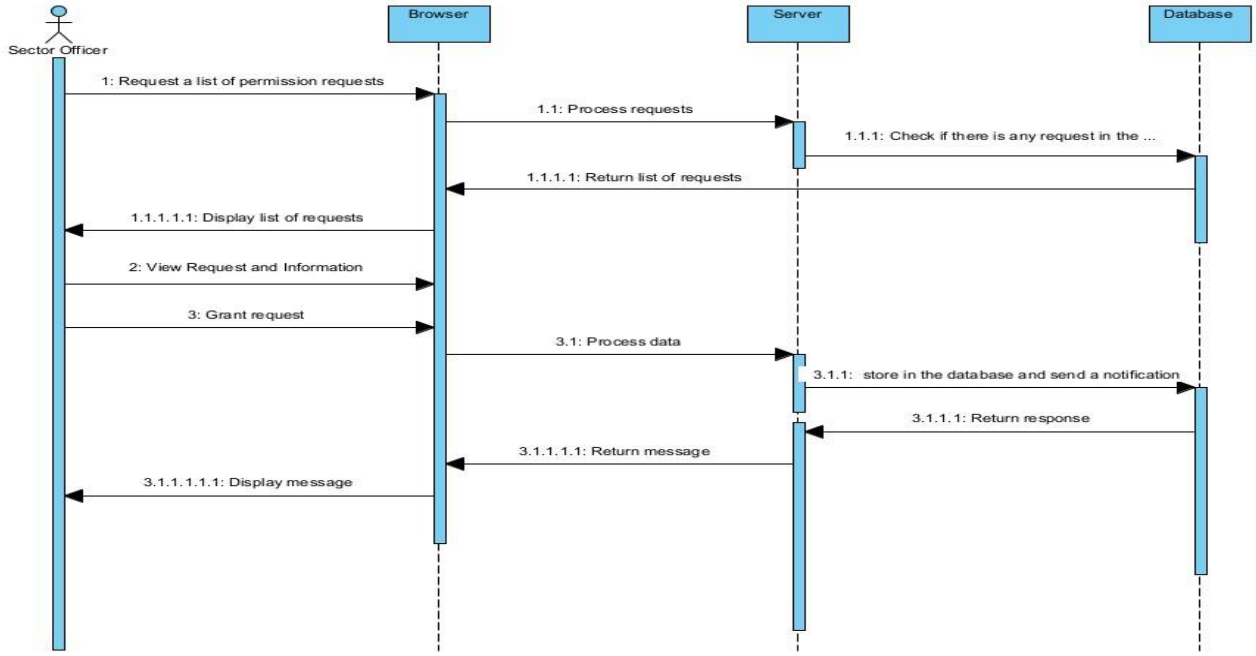


Figure 5: Provide permission sequence diagram.

Access information on animal vaccination status

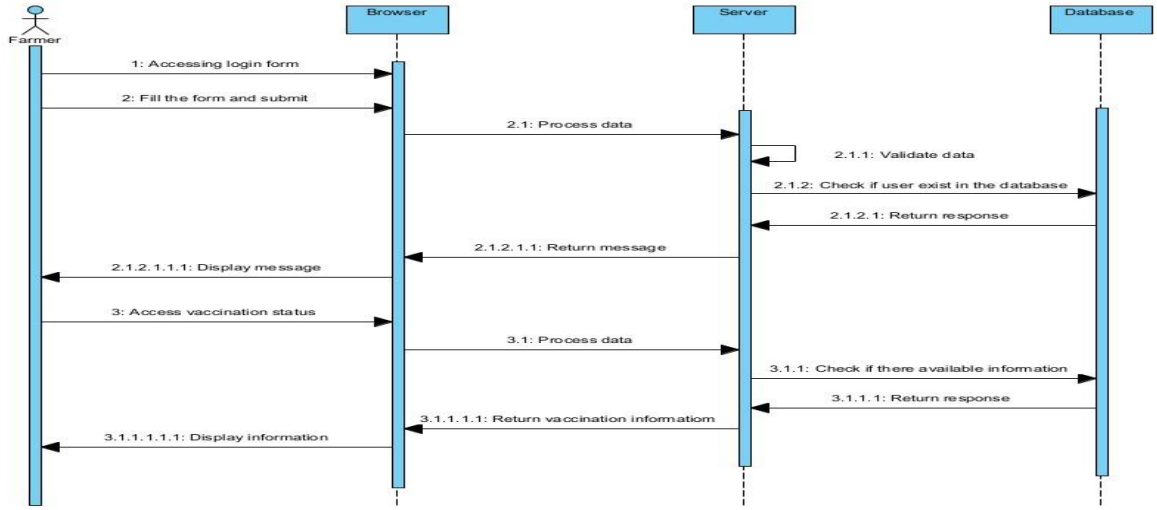


Figure 6: Access information on animal vaccination status sequence diagram

Communicate and share information.

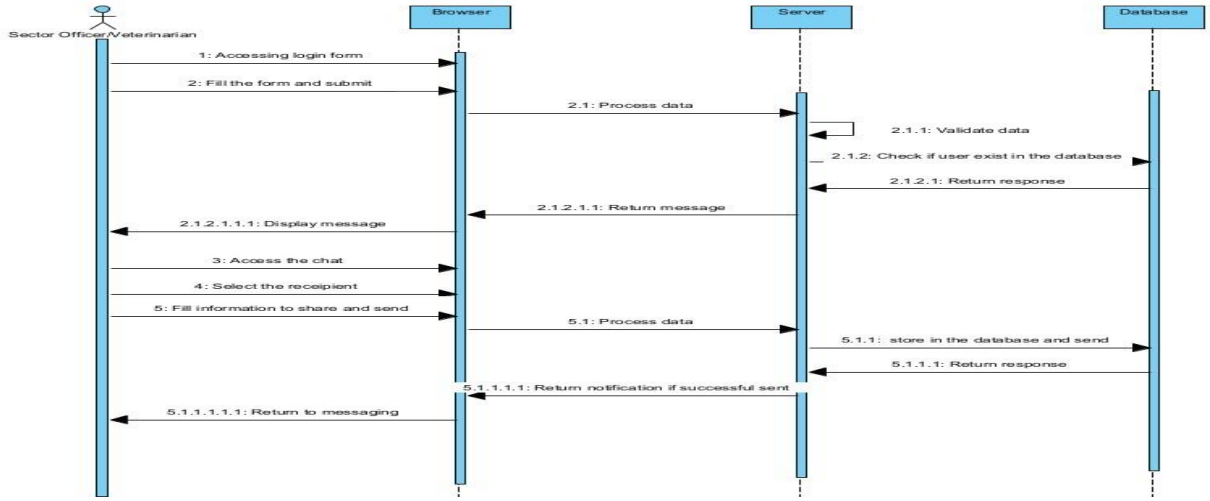


Figure 7: Communicate and share information sequence diagram.

Activity Diagram

An activity diagram is a graphical representation of a series of procedural actions within a system, using shapes connected by arrows in the Unified Modeling Language (UML). It shows concurrent activities, conditional actions, use cases, and system functions. Arrows indicate the order of activities, starting from initiation and ending with completion. Initial states are depicted as black circles, with one circle denoting a goal. Executed actions are in rounded rectangles with descriptions. Decisions are shown by diamond shapes. When an activity finishes, a transition to subsequent actions must be chosen. Activity diagrams capture dynamic system aspects, depicting flow between operations. They manage various flow types using elements like fork and join.

Create account activity.

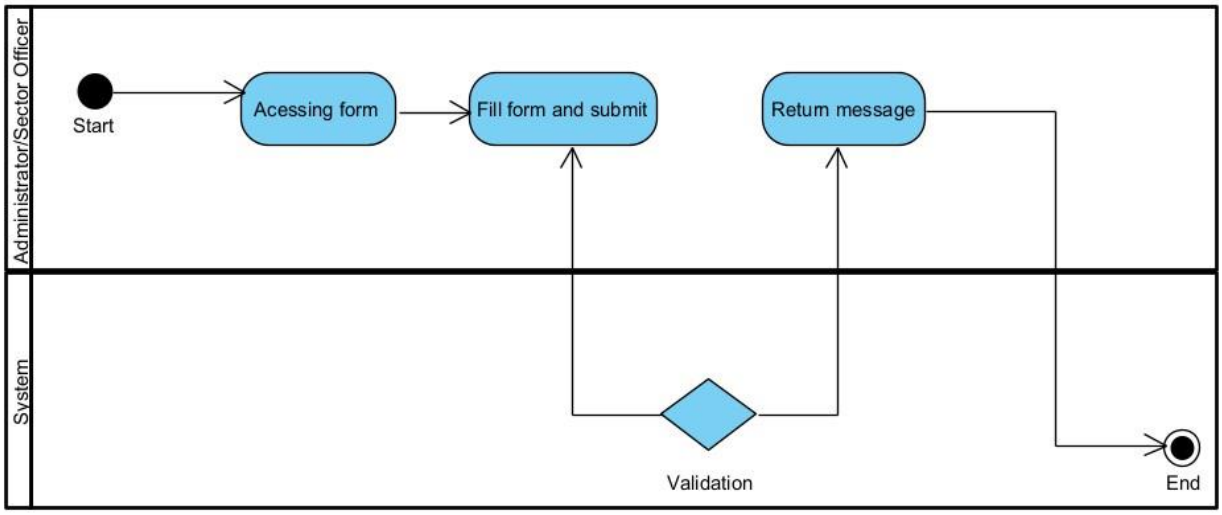


Figure 8: Create account diagram.

Provide Permission activity.

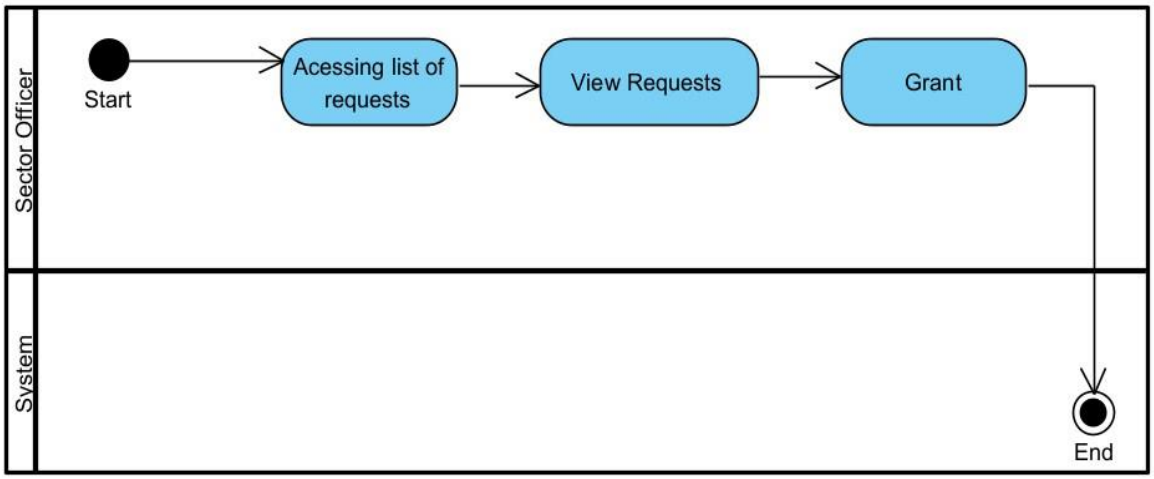


Figure 9: Permission activity diagram

Access vaccination status

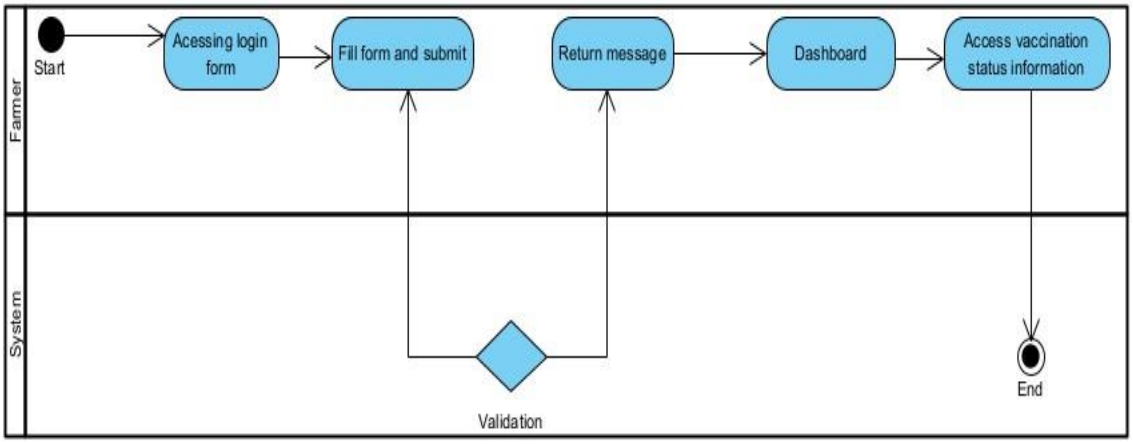


Figure 10: Access vaccination status information

Communicate and share information.

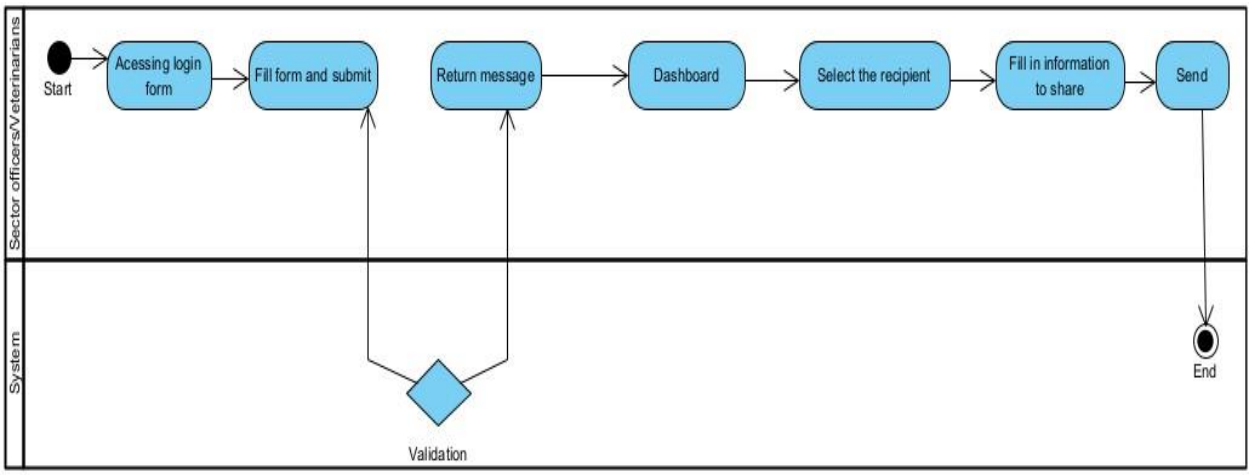


Figure 11: Communicate and share information

Database Diagram

A database schema diagram is a collection of structured data that can be accessed, maintained, and changed simply. Database software solutions that are primarily used for storing, changing, extracting, and searching for information within databases are referred to as database management systems (DBMS).

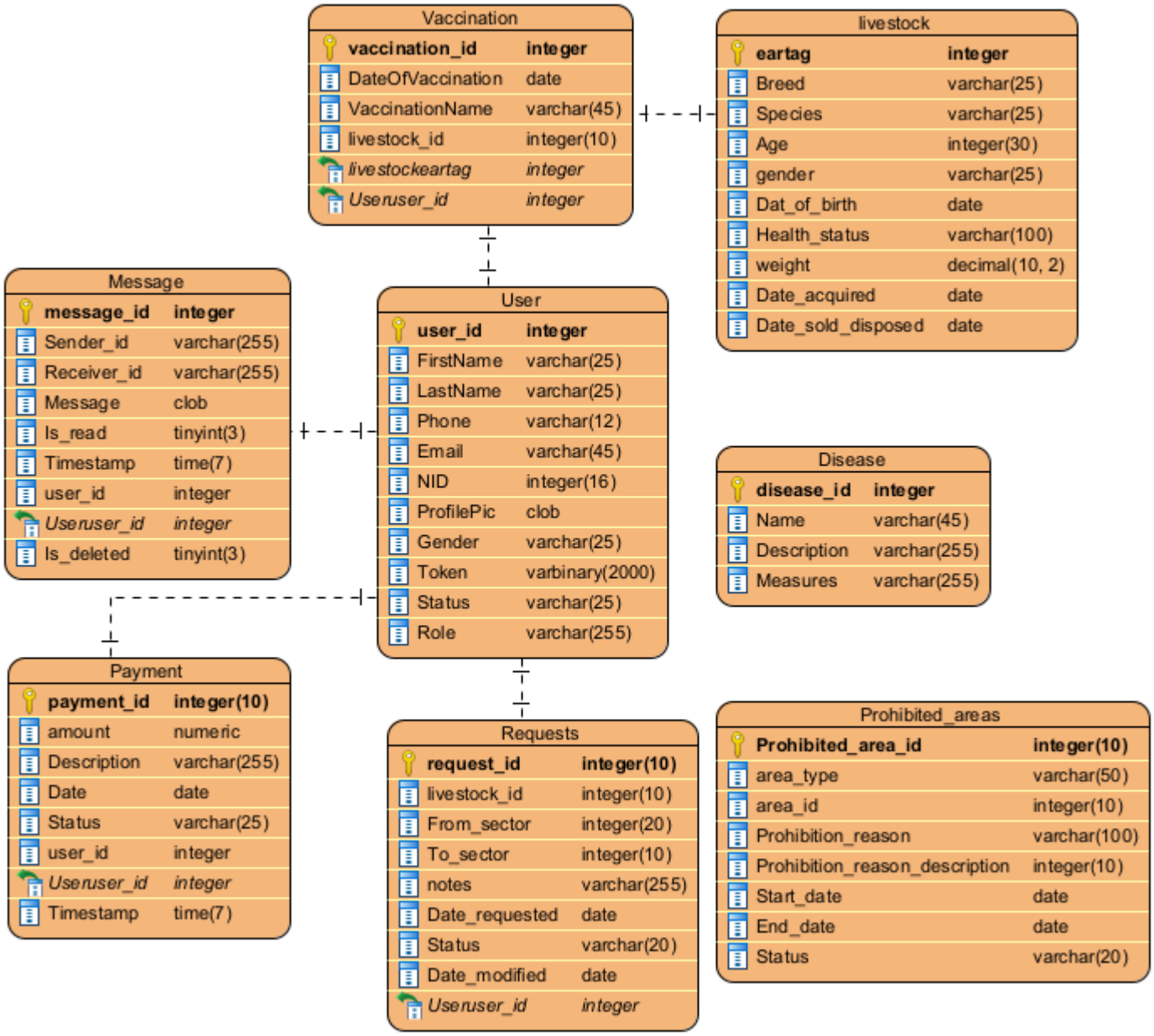


Figure 12: Database schema

System Architecture Design

A system architecture, also known as systems architecture, is a conceptual model that outlines a system's structure, behavior, and many viewpoints. A formal description and representation of a system arranged in a way that facilitates reasoning about the system's structures and behaviors is known as an architectural description.

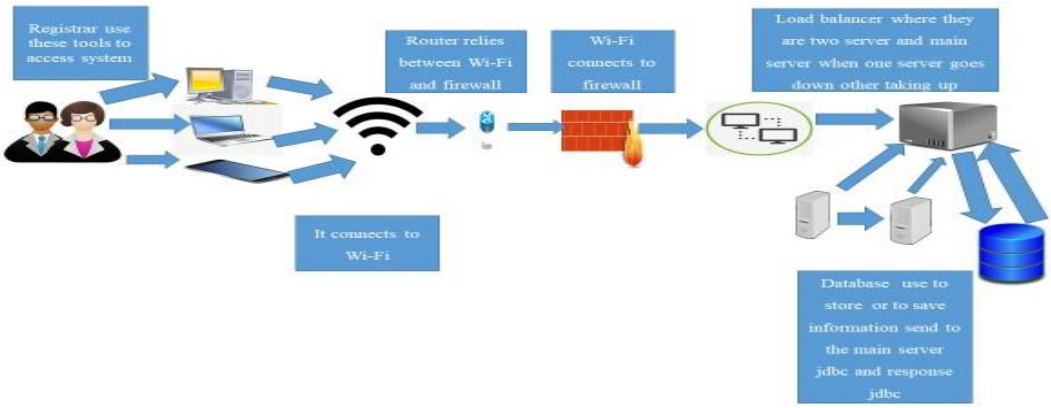


Figure 13: System architecture

CHAPTER 4

IMPLEMENTATION OF THE NEW SYSTEM

Introduction

This chapter describes the development E-Tungo System. It includes a brief explanation of the technology utilized to create this system, as well as the operations that were used. Finally, software and hardware compatibility requirements.

Tools and Technology Used

To successfully implement this project, I used the following technologies and tools namely:

TECHNOLOGY SOFTWARE	PRODUCT
Database	MySQL
Integrated development environment	Visual Studio Code
Web Browser	Google chrome
Programming Language	PHP, JavaScript
Design	Visual paradigm and figma

Table 13: tools and technology used.

MYSQL: MySQL is an open-source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter My, and "SQL", the acronym for Structured Query Language.

Visual Studio Code:

Visual Studio Code, often referred to as VS Code, is a popular open-source code editor developed by Microsoft. It is designed for developers and is widely used for a wide range of programming and scripting tasks.

Google Chrome: is a popular web browser developed by Google. It was first released in 2008 and has since become one of the most widely used web browsers globally.

PHP: PHP, which stands for "Hypertext Preprocessor," is a popular and widely used opensource server-side scripting language. It is primarily designed for web development but can also be used for general-purpose programming.

JavaScript: JavaScript is a versatile and widely used programming language that is primarily known for its role in web development. It allows developers to add interactivity and dynamic behavior to websites and web applications.

Visual Paradigm: features a rich set of Agile and Scrum tools for project management. Project Management Tool. PM process map and roadmap tools.

Figma is a vector graphics editor and prototyping tool which is primarily web-based, with additional offline features enabled by desktop applications for macOS and Windows.

Presentation of the new system

Home Page

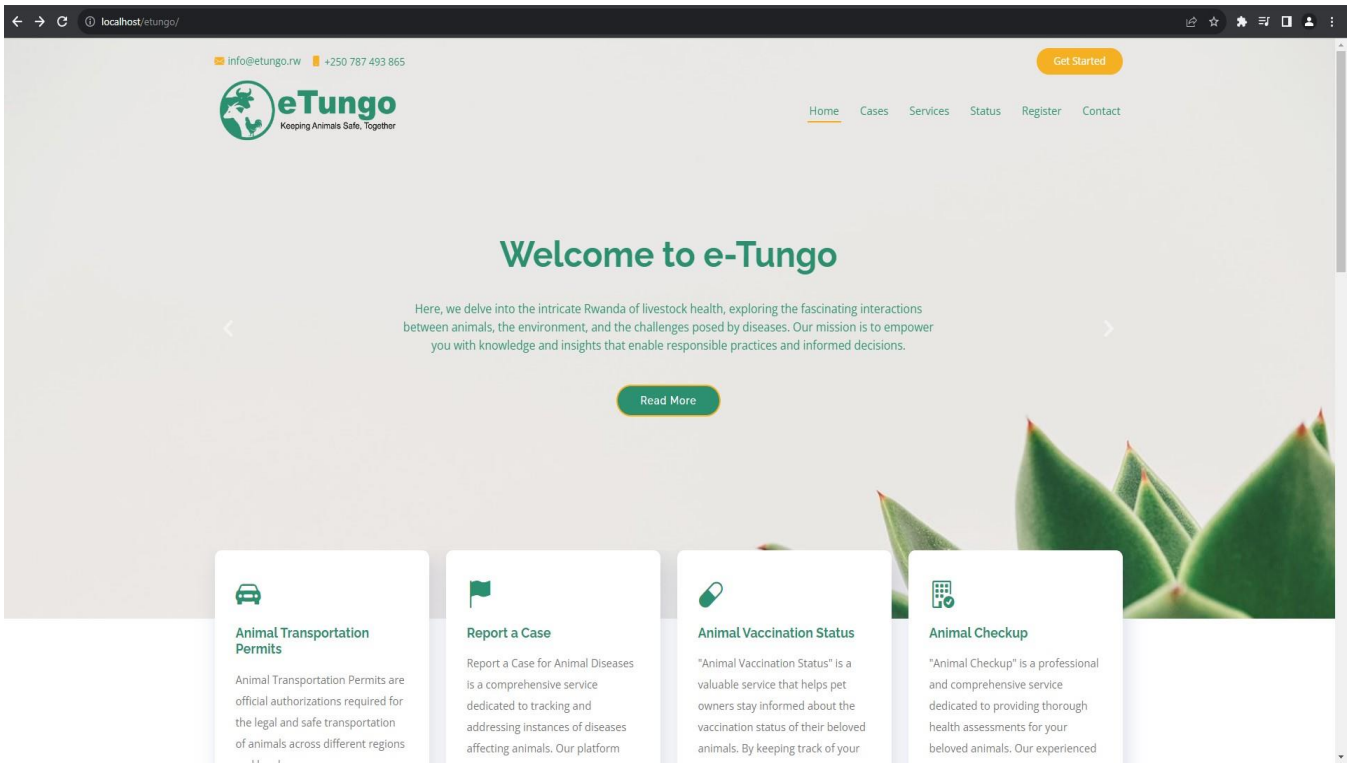


Figure 14: Homepage

Farmers account creation

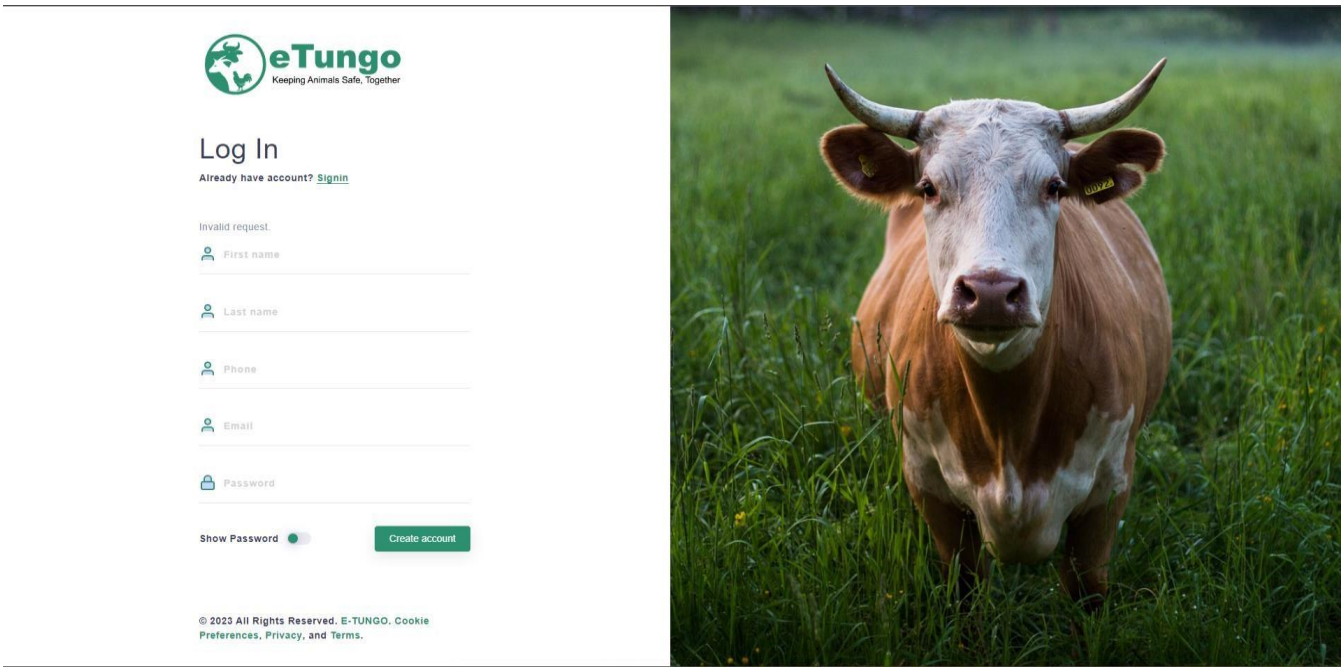


Figure 15: Farmers account creation

Vaccination and transport permission request

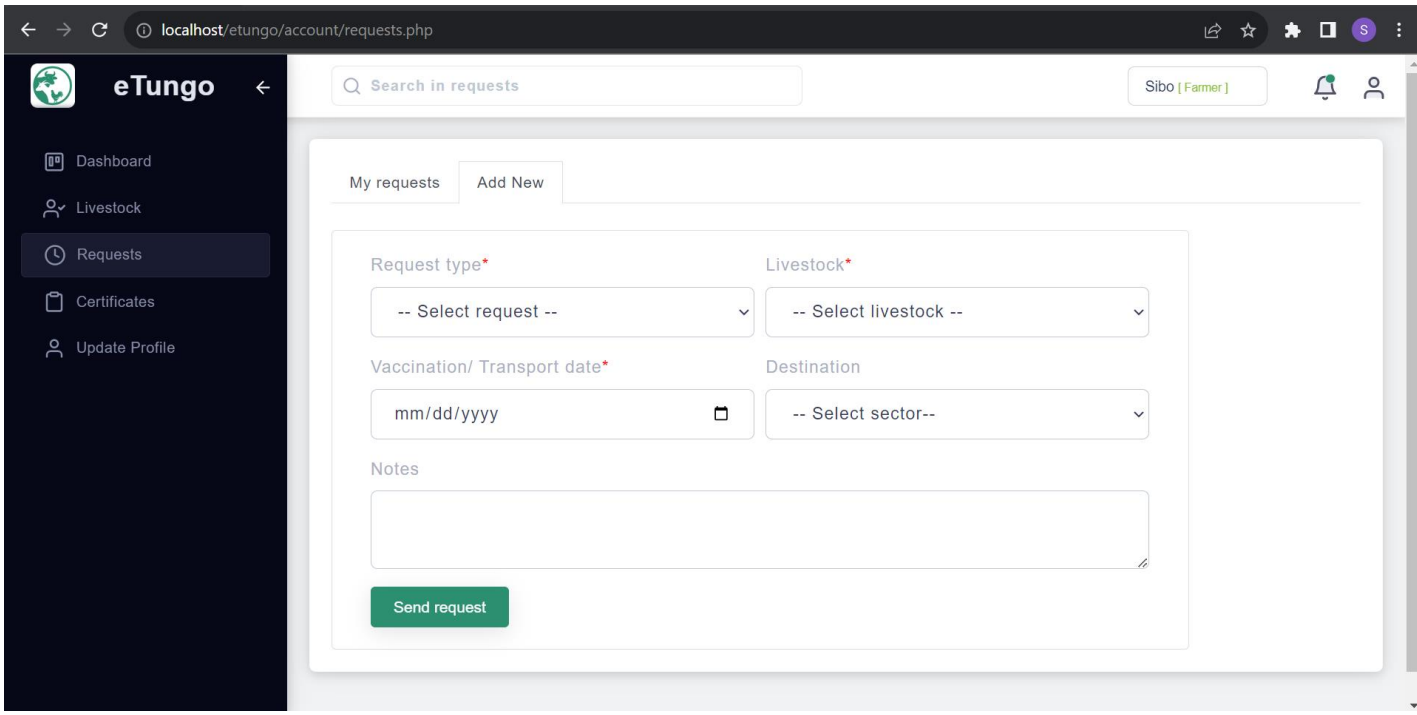


Figure 16: Vaccination and transport permission request

Sector officer registration

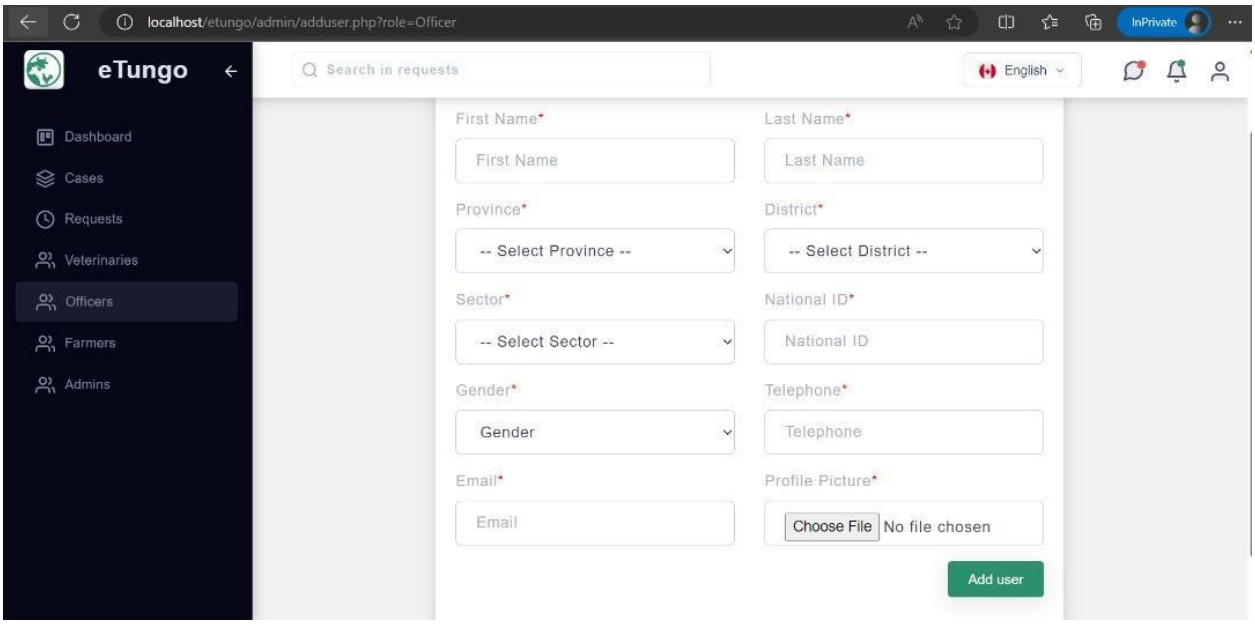


Figure 17: Sector officer account creation

Admin dashboard

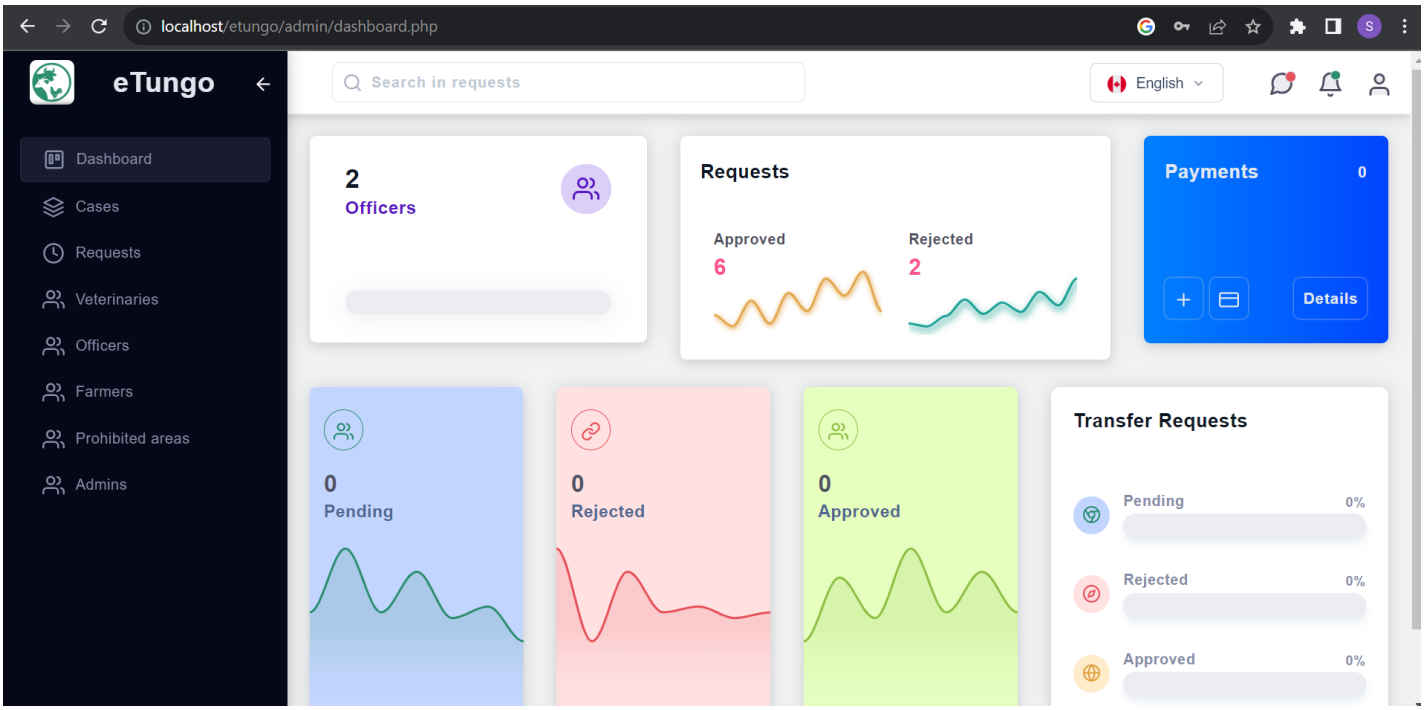


Figure 18: Admin dashboard

Disease Registration

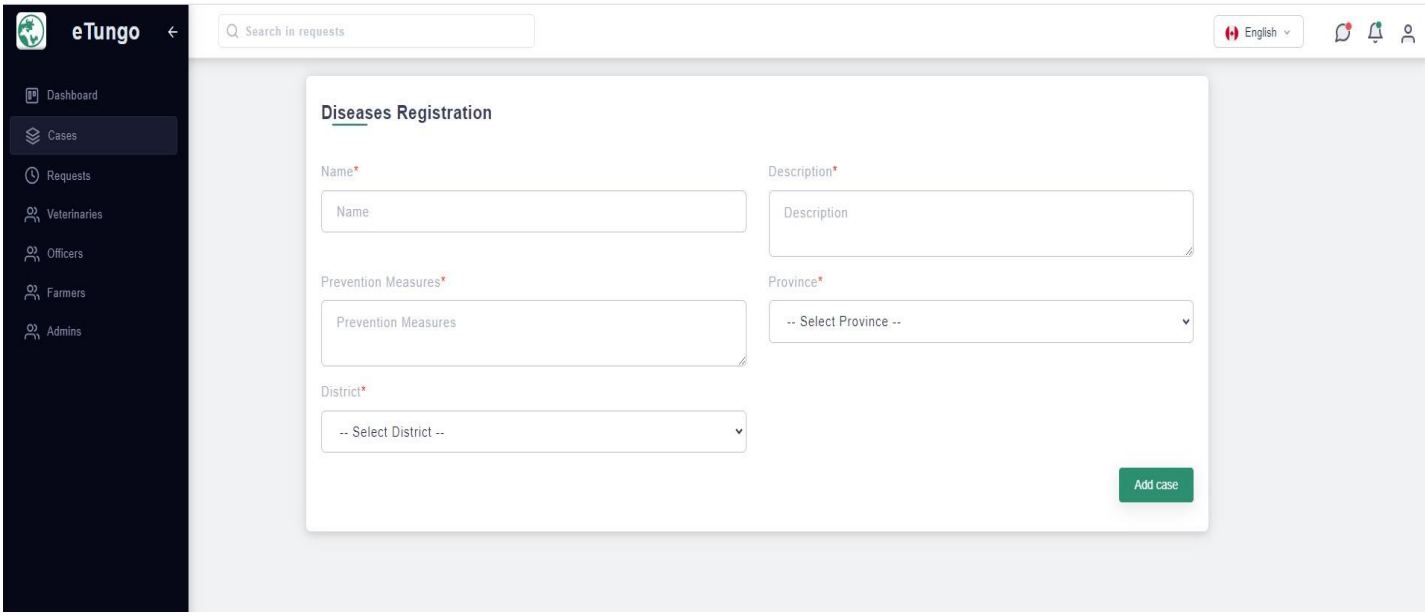


Figure 19: Disease registration

Cases and diseases posts

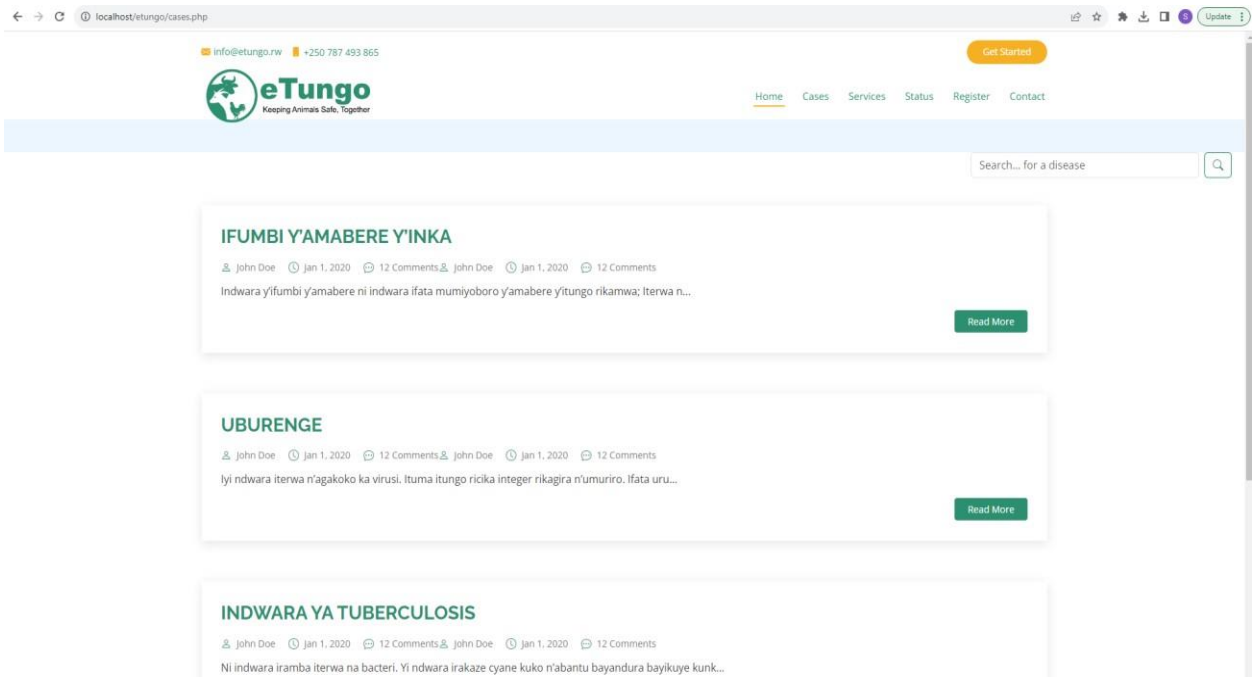


Figure 20: Cases and diseases posts

Livestock certificate

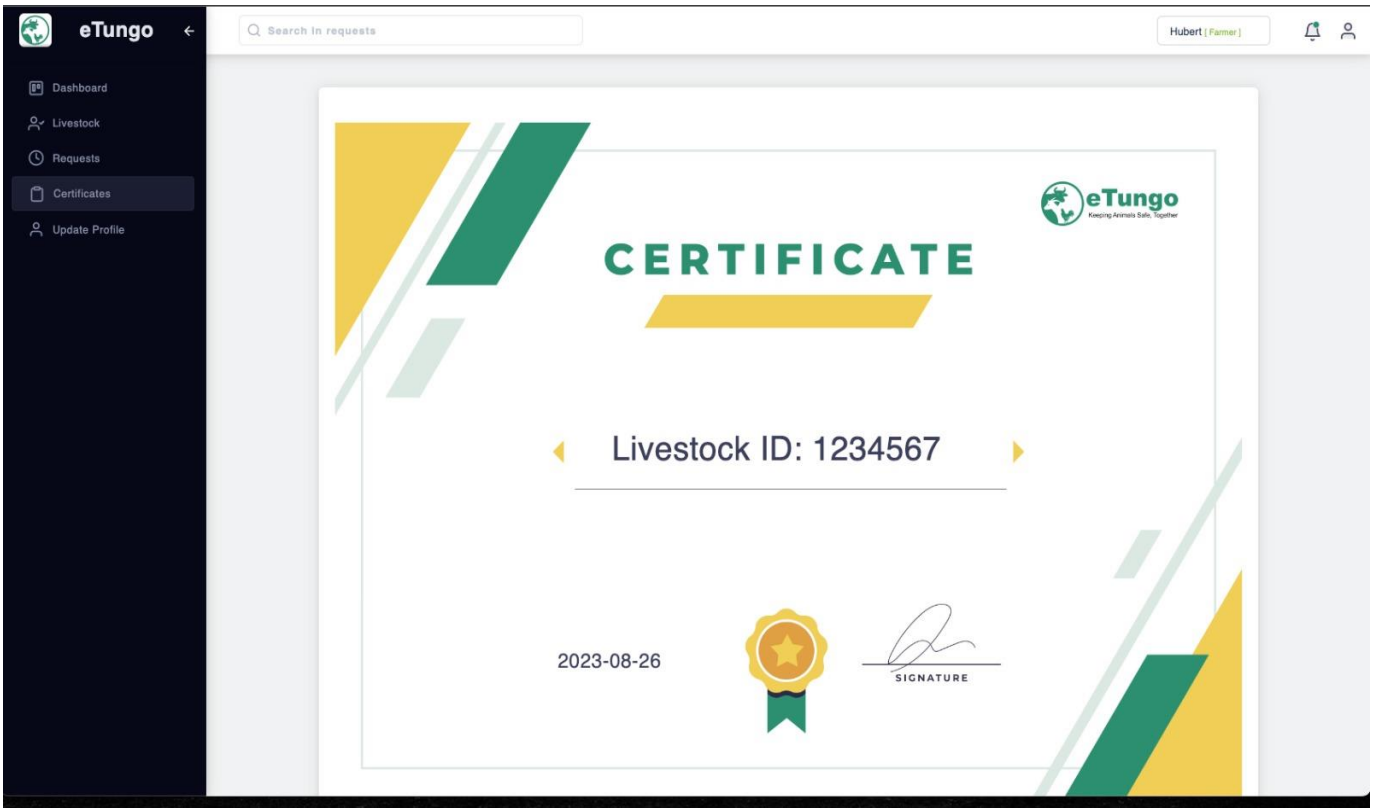


Figure 21:Farmer certificate

Application testing

Application testing is crucial for validating software design. It ensures the software aligns with specifications and functions as intended. Key testing stages include:

Unit Test: Examines individual code components for conformance to specs, detecting logic errors.

Integration Test: Evaluates grouped units for interaction flaws.

Validation Test: Assesses software in its real environment, verifying alignment with initial needs.

Hardware and Software requirements

Requirements:

1. Phone, tablet, computer
2. Internet connection.
3. A web browser (Opera Mini, Google Chrome, etc.)

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

The E-Tungo System was created to provide farmers and stakeholders in the animal resources sector with efficient access to critical resources and services. Our approach involved research, design, and implementation.

Research involved observation, documentation, and interviews to understand stakeholders' challenges. Design, using Visual Paradigm, resulted in a detailed system blueprint. Implementation was done in PHP.

The system now offers diverse learning resources on animal resources and services. Administrators can enhance these resources.

In summary, the E-Tungo System is operational and meets its objectives.

Recommendations

I would suggest this system or platform to all farmers and the Ministry of Agricultural and Animal Resources as it will ease the services and tasks. Investigate the utility of this new application or platform by learning how it works and what it accomplishes, so that it helps to provide good value to all. To wrap up this project, I'd like to recommend that anyone who is interested provide more functions to improve my work.

REFERENCES

Books

Bell, D. (2005). *Software Engineering for Students, Programming Approach, Fourth Edition.* USA.

Bentley, L. D. (1997). *Systems Analysis and Design for the Global Enterprise.* Boston: Publisher of Boston.

Dennis, A, (2012). *System Analysis and Design.* India: John Wiley & Sons, Inc.

Eileen, (2006). *Requirement and Analysis.* USA: Rational software Library.

Rumbaugh, (2006). *Unified Modeling Language.* USA: Rational software Library.

Stumptner, (2007). *Behavior consistent inheritance in UML.* McGill inc.

Wixom, (2011). *System Analysis and Design Fifth Edition.* USA: Don Fowley.

Web Sites

Baudry, B. (2002, August 7). Cascading style sheet Retrieved July 20, 2017, from www.org/tr/2010/wd-css, accessed on June 03,2019

Rouse, M. (2005, April 14). Java-Server-Page-JSP. Retrieved July 20, 2017, from www.tutorialspoint.com/jsp/jsp_iterative_model.htm, accessed on May 11, 2019

John, T. (2012, May 15). JavaScript Retrieved from www.w3schools.com/whatis_js.asp, accessed on: July 17, 2019 https://www.codecademy.com https://www.php.net/ https://www.wikipedia.google.com