MAKERERE UNIVERSITY

DEPARTMENT OF NETWORKS

SCHOOL OF COMPUTING AND INFORMATICS TECHNOLOGY

BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING

SECOND YEAR RECESS PROJECT

SOFTWARE DESIGN DOCUMENT

Github link: https://github.com/blackbody256/Agricure

Kaggle link: https://www.kaggle.com/code/karagwaanntreasure/cnn-model

Model Github link: https://github.com/Karagwa/Crop-Disease-model

Dataset: https://www.kaggle.com/datasets/karagwaanntreasure/plant-disease-detection

BY: GROUP J (DAY)

Name	Registration Number	Student Number
KARAGWA ANN TREASURE	23/U/09034/PS	2300709034
AYAN MUSTAFA ABDIRAHMAN	23/X/22948/EVE	2300722948
WANYANA SELINA MASEMBE	23/U/1529	2300701529
AMABE JUNIOR HUMPHREY	23/U/05942/PS	2300705942
AKANGA ANDREW	23/U/05555/PS	2300705555

Table of Contents

List of Figures

List of Tables

1.INTRODUCTION

1.1 Document Overview

This document outlines the project overview, system requirements, scope, system architecture and user interface of the Agricure system. It serves as a reference point for stakeholders involved in system development, implementation, deployment, and maintenance.

1.2 Project Overview

Agriculture is the backbone of many developing countries, yet millions of rural farmers continue to face significant challenges in managing crop health. One of the most pressing issues is the timely and accurate diagnosis of crop diseases. Many rural farmers lack access to up-to-date information on disease management practices. As a result, diseases often go undetected or are misdiagnosed, leading to severe yield losses, increased poverty, and food insecurity. Additionally, climate change and unpredictable environmental conditions have introduced new disease patterns that traditional farming knowledge cannot easily address. To address these issues, there is an urgent need for a scalable, accessible, and intelligent system that empowers farmers with timely, data-driven insights to protect and improve their crop yields.

1.3 Justification

The Agricure System is an AI-powered, data-driven application designed to support farmers, agronomists and agricultural extension workers in diagnosing crop diseases and receiving actionable treatment recommendations. By leveraging machine learning and predictive analytics, the system aims to enhance crop health management, improve yields, and reduce losses caused by plant diseases.

1.4 Purpose

The purpose of the Agricure system is to provide a smart agricultural support platform that diagnoses crop diseases through image analysis using AI/ML models and provide real-time recommendations for disease management.

1.5 Goals and Objectives

The Agricure system is built with the following key objectives:

i. To facilitate secure user registration and authentication for farmers, agronomists and administrators.
ii. To allow users to upload images of affected crops via a web interface.
iii. To accurately diagnose crop diseases using a trained AI/ML model.
iv. To generate treatment recommendations based on diagnosis and environmental conditions.

v. To track historical disease reports and provide analytics on disease trends.
vi. To enable administrators to manage users, update AI models, and maintain system content.
vii. To enable agronomists to label newly uploaded datasets and provide personalized feedback to farmers who opt for the in-app chat support option.

2. SYSTEM REQUIREMENTS

2.1 Functional Requirements (FR)

Functional requirements describe what the Agricure system should do, the features and services it must provide.

2.1.1 User Management

i. FR1: The system shall allow users (farmers, agronomists, extension workers) to register and create personal accounts.
ii. FR2: The system shall authenticate users through a secure login system.
iii. FR3: The system shall allow users to reset their passwords.
iv. Image Upload and Disease Diagnosis
v. FR4: The system shall enable users to upload images of affected crops web applications.
vi. FR5: The system shall process the uploaded image using a trained AI/ML model to detect and classify crop diseases.
vii. FR6: The system shall display the diagnosis results to the user, including disease name, severity, and affected plant part.

2.1.2 Recommendation and Advisory System

i. FR7: The system shall generate data-driven recommendations for disease management and treatment based on diagnosis results.
ii. FR8: The system shall integrate environmental data (e.g., temperature, humidity, soil type) to enhance the accuracy of recommendations.
iii. FR9: The system shall notify users of preventive measures for common regional diseases based on predicted environmental risks.

2.1.3 Historical Data and Analytics

i. FR10: The system shall maintain a history of users' crop diagnoses and recommendations.
ii. FR11: The system shall visualize trends and analytics for farmers, indicating disease patterns and management outcomes over time.

2.1.4 IoT Integration

i. FR12: The system shall collect environmental data from connected IoT devices (e.g., soil moisture, temperature sensors).
ii. FR13: The system shall use real-time IoT data to support disease prediction and advisory recommendations.

2.1.5 Admin and Model Management

i. FR14: The system shall allow administrators to manage users, reported issues, and system content.
ii. FR15: The system shall enable administrators to update and retrain AI models with new image datasets.

2.2 Non-Functional Requirements (NFR)

Non-functional requirements describe how the Agricure system performs, its qualities and operational constraints.

2.2.1 Performance Requirements

i. NFR1: The system shall process disease detection and generate recommendations within 10 seconds of image upload.
ii. NFR2: The system shall support at least 1,000 concurrent users without performance degradation.

2.2.2 Security Requirements

i. NFR3: The system shall ensure secure user authentication using hashed passwords and secure session management.
ii. NFR4: The system shall encrypt sensitive data transmissions using HTTPS/SSL protocols.
iii. NFR5: The system shall protect AI model access and administrative controls via role-based access control (RBAC).

2.2.3 Usability Requirements

i. NFR6: The system shall offer a user-friendly, multilingual interface tailored for farmers with limited digital literacy.
ii. NFR7: The mobile application shall be compatible with Android versions 8.0 and above.

2.2.4 Reliability and Availability

i. NFR8: The system shall maintain 99.5% uptime.
ii. NFR9: The system shall provide an offline mode for diagnosis history access when network coverage is unavailable.

2.2.5 Scalability Requirements

i. NFR10: The system shall be scalable to accommodate future integration of additional crop types, regions, and IoT devices.

2.2.6 Maintainability

i. NFR11: The system's AI models and datasets shall be updatable without requiring system downtime.

2.2.7 Compliance

i. NFR12: The system shall comply with data protection regulations (e.g., GDPR or applicable national laws) to safeguard user and farm data.

3. SCOPE

The system encompasses the following functionalities:

i. User Management: Account registration, login, password reset, profile management.
ii. Image-Based Diagnosis: ML-based detection of crop diseases from uploaded images.
iii. Advisory System: Treatment suggestions and recommendations.
iv. Historical Analytics: Diagnosis records, trend visualization, and analytics dashboards.
v. Admin Dashboard: User and content management

3.1 User Roles and Access

Table 1: Table shows the user roles and access rights

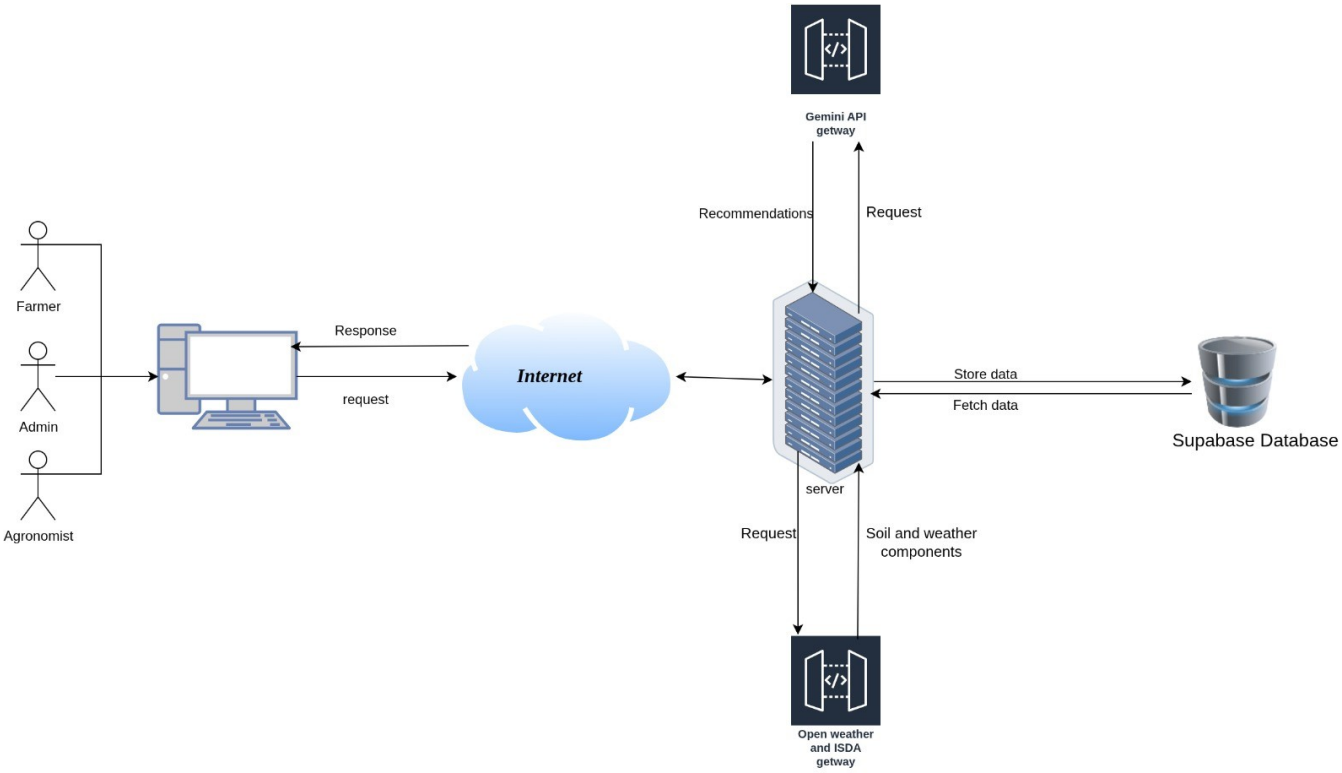
User role	Description	Access Rights
Farmer	This is the primary user who uploads crop images and receives advice.	i. Register, log in and manage their profiles ii. Upload crop images iii. View diagnosis and recommendations iv. Access personal diagnosis history
Agronomist	This is an expert who can monitor trends, provide manual feedback, or validate AI results.	i. Access diagnosis trends ii. Provide feedback to farmers iii. Label new datasets for retraining
Administrator	This is a super user who manages other users, models, content, system configurations, and uploads new datasets for retraining the model.	i. Manage user accounts and roles ii. View all system data iii. Retrain and update AI model iv. Handle reported issues

4. SYSTEM ARCHITECTURE

4.1 System Overview

This section shows the high-level architecture of the Agricure system.

Figure 1: Figure shows the architecture of the Agricure System



4.2 Decomposition Description

This section decomposes the Agricure system into figurative models explaining user interactions and actions.

4.2.1 Activity diagrams

These are UML diagrams that are used to model the workflows in the Agricure application.

Figure 2: Figure shows the Activity Diagram of the farmer.

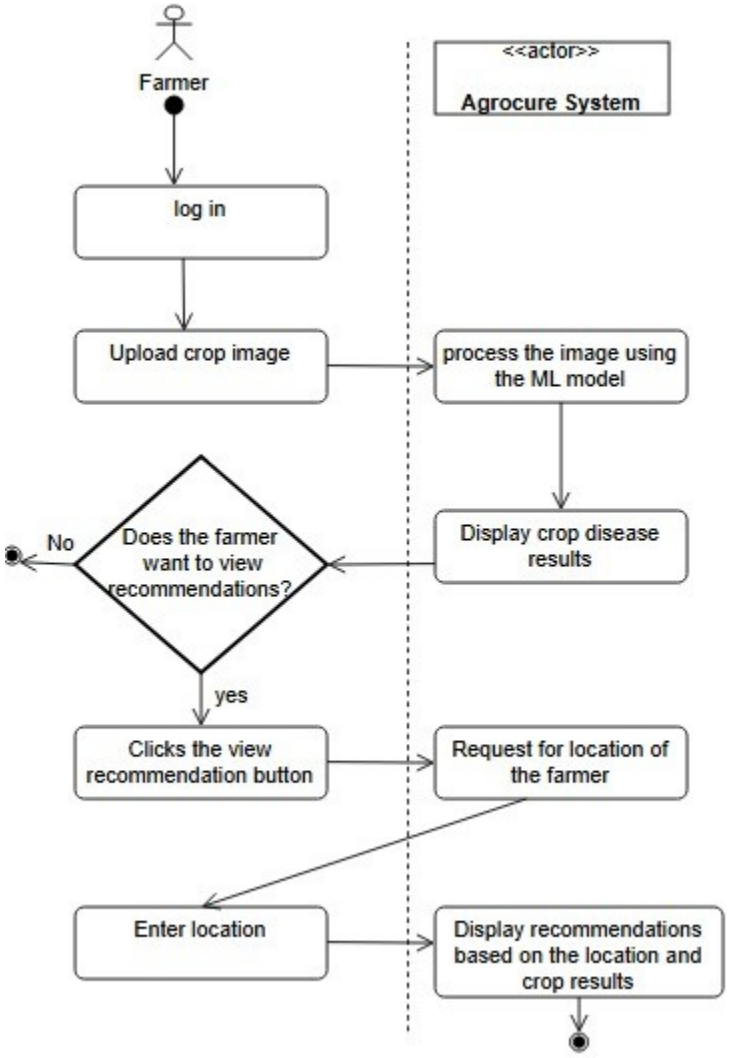
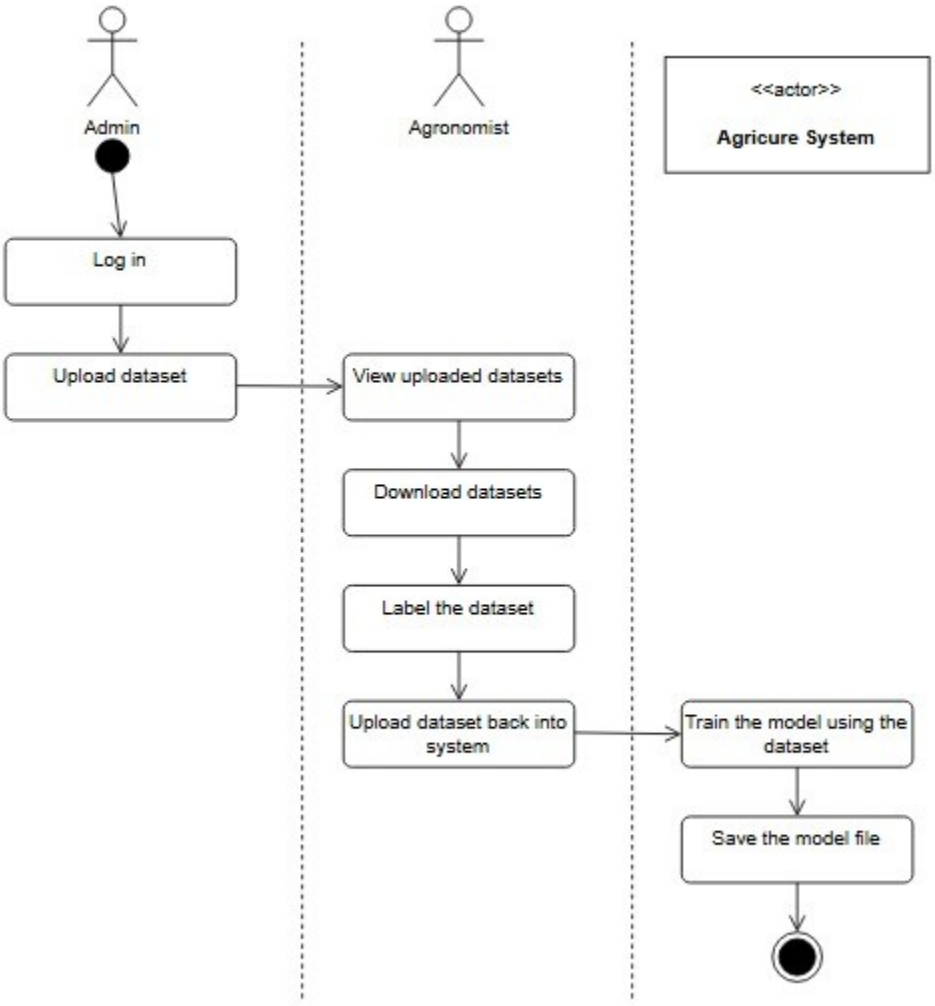


Figure 3: Figure shows the Activity diagram of the Administrator



4.2.2 Sequence Diagrams

These are UML diagrams that are used to show how objects interact in the Agricure system.

Figure 4: Figure shows the Sequence Diagram for the farmer interactions

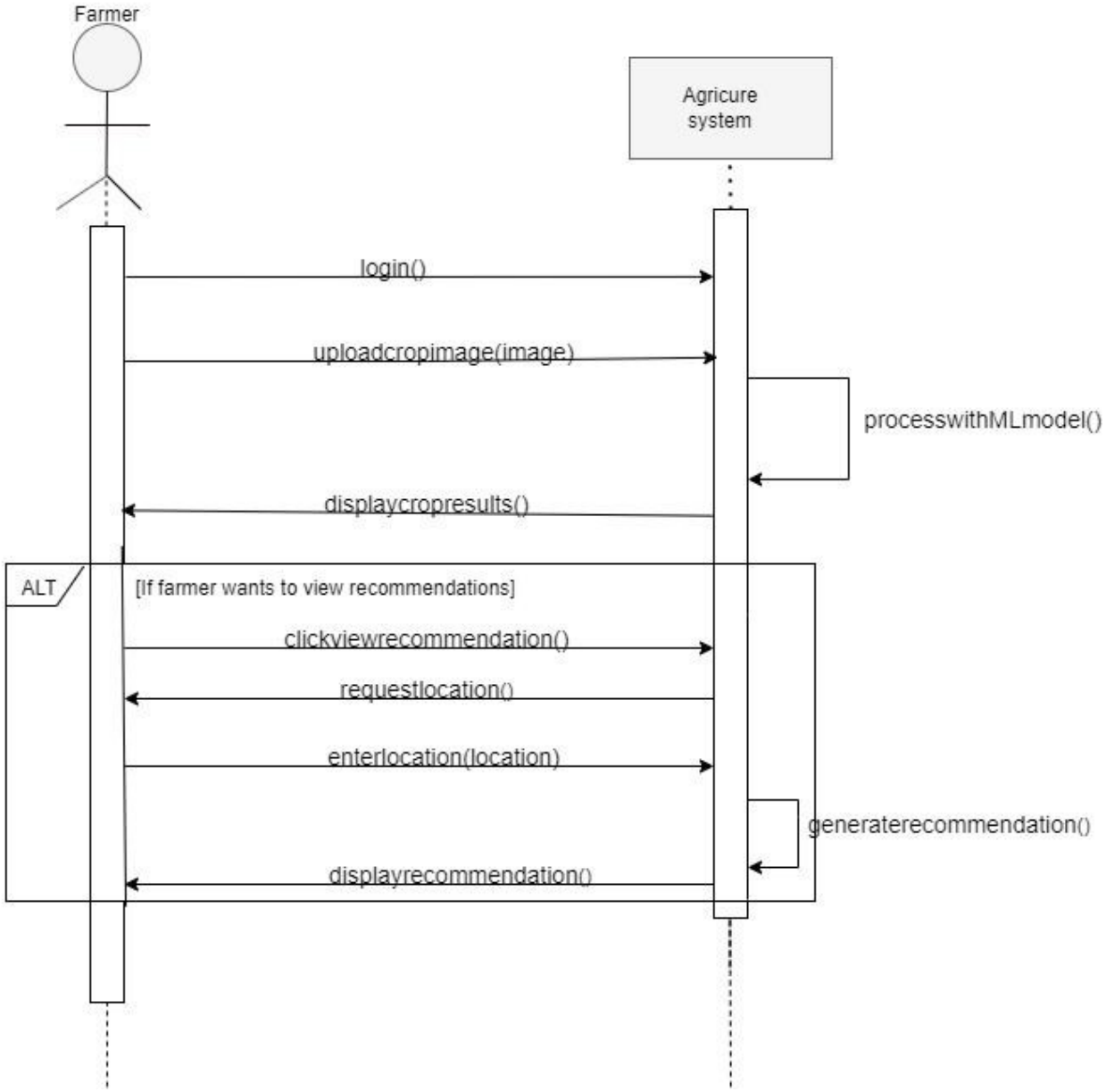
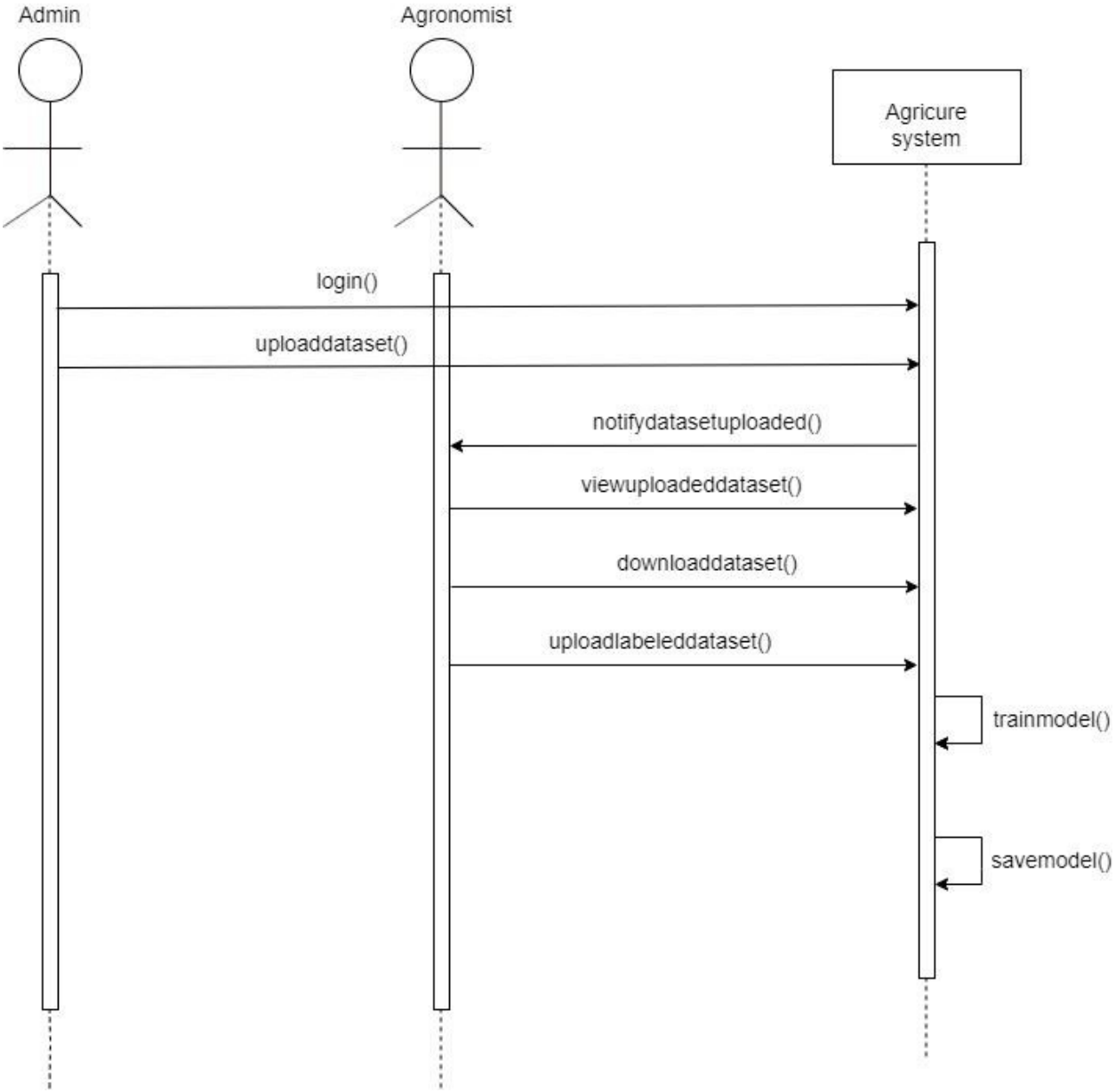


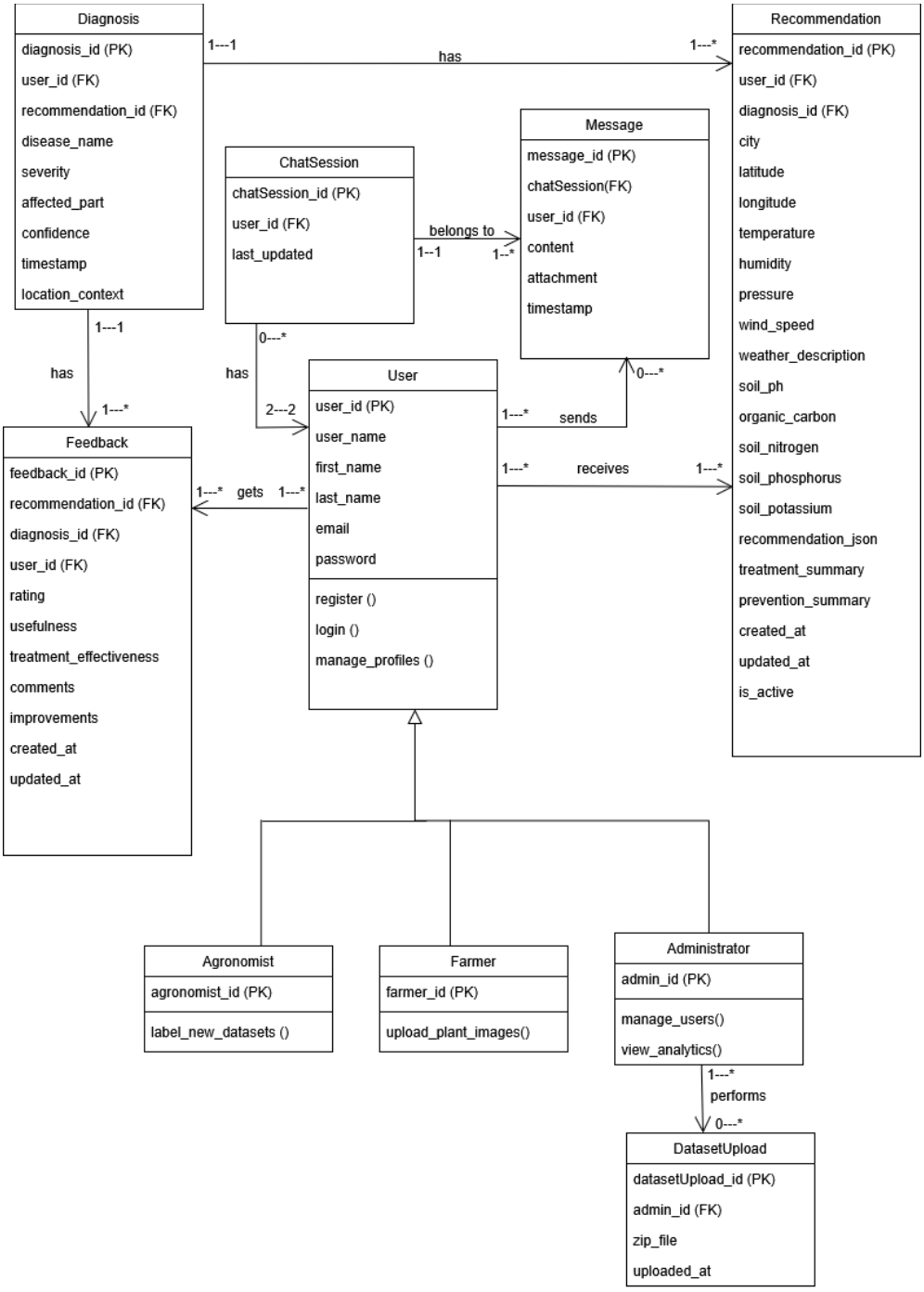
Figure 5: Figure shows the Sequence Diagram for the administrator interactions



4.2.3 Class Diagram

This UML diagram shows the structure of the Agricure system by outlining its classes, attributes and methods.

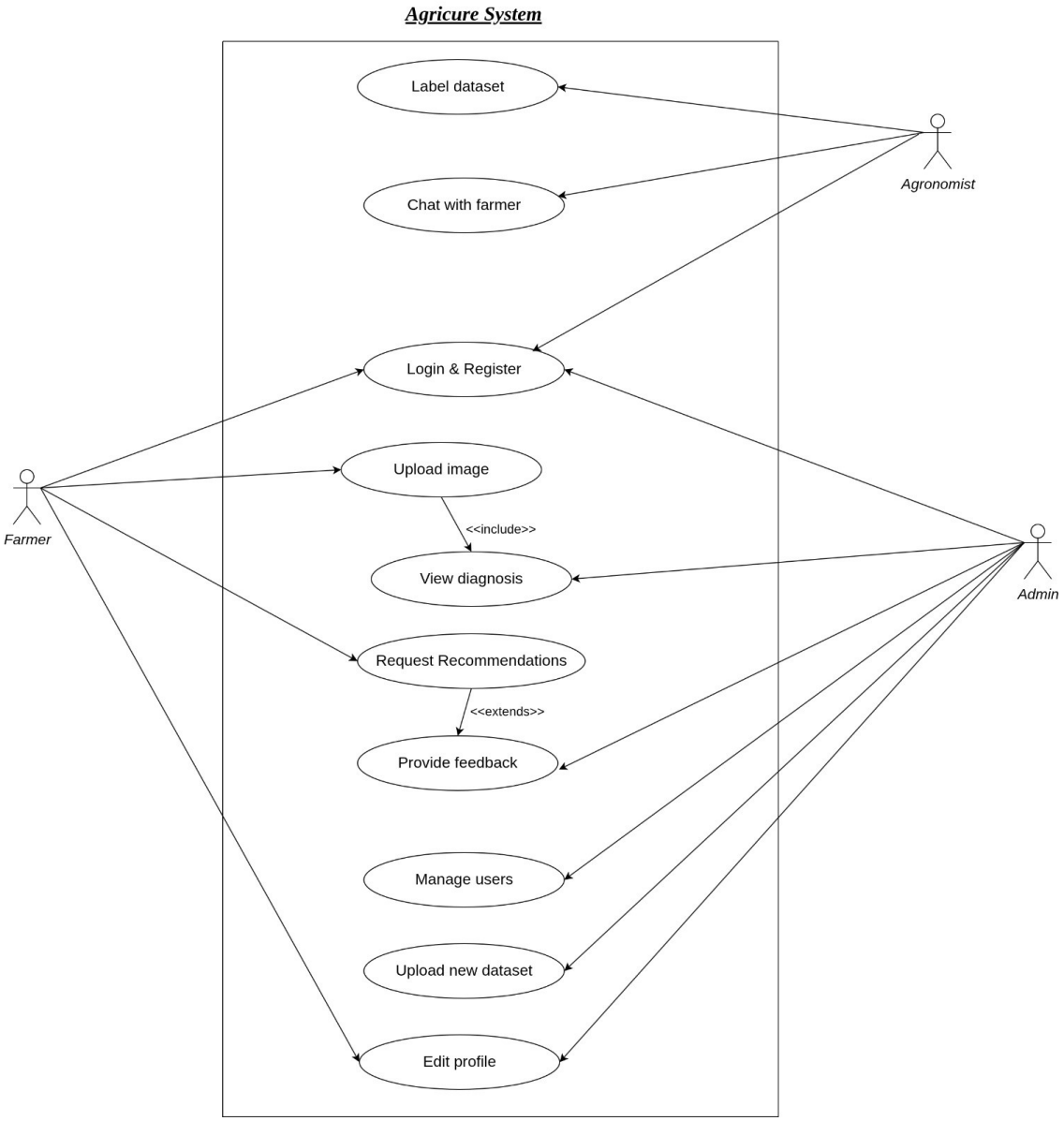
Figure 6: Figure shows a Class Diagram of the Agricure System



4.2.4 Use case diagram.

This is a UML diagram that shows interactions between actors in the Agricure system.

Figure 7:Figure shows Use-Case Diagram of the Agricure System.



4.3 Model Architecture

i. Transfer Learning Foundation: EfficientNetB0 is used as the pre-trained feature extraction backbone, with its layers frozen (base_model.trainable = False).

ii. Custom Classification Head: Appended to the base model, it consists of GlobalAveragePooling2D, two Dropout layers (0.5 and 0.3), and two Dense layers (256 units with 'relu' and num_classes units with 'softmax').

iii. Model Compilation: Optimized with Adam (learning rate=0.001), sparse_categorical_crossentropy loss, and accuracy metric.

Figure 8: Figure shows the Modal Architecture

Layer (type)	Output Shape	Param #
input_layer_3 (InputLayer)	(None, 224, 224, 3)	0
efficientnetb0 (Functional)	(None, 7, 7, 1280)	4,049,571
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 1280)	0
dropout_2 (Dropout)	(None, 1280)	0
dense_2 (Dense)	(None, 256)	327,936
dropout_3 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 23)	5,911

5. USER INTERFACE

5.1 Farmer Interactions

Figure 9: Figure shows Farmer dashboard

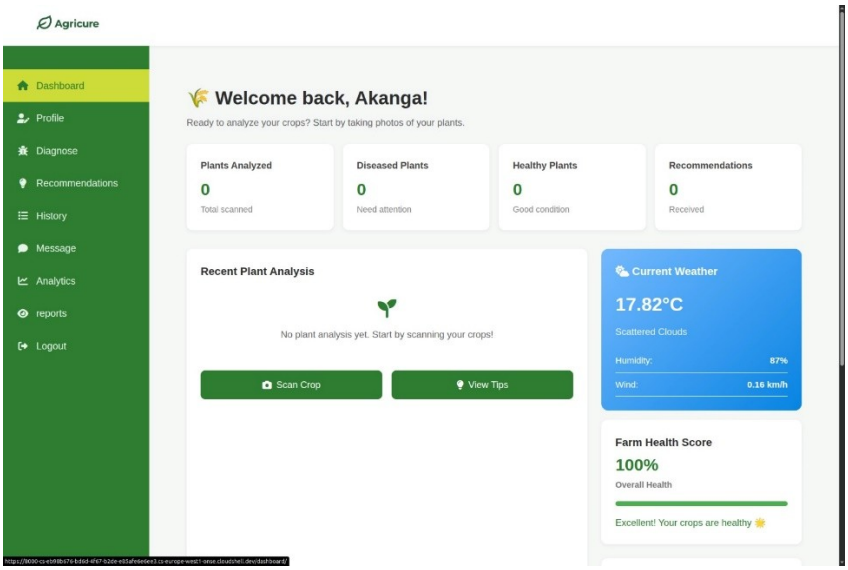


Figure 10: Figure shows the Farmer Diagnosis page

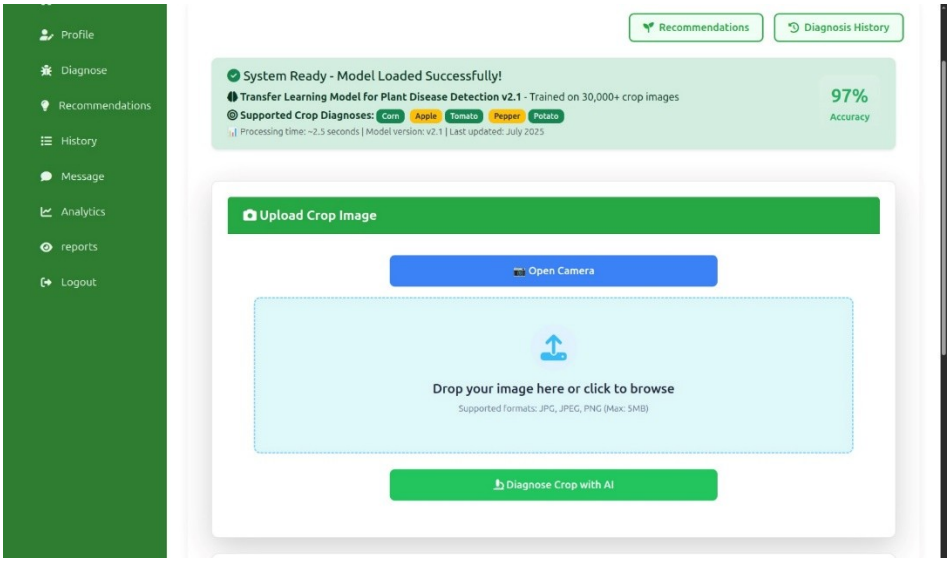
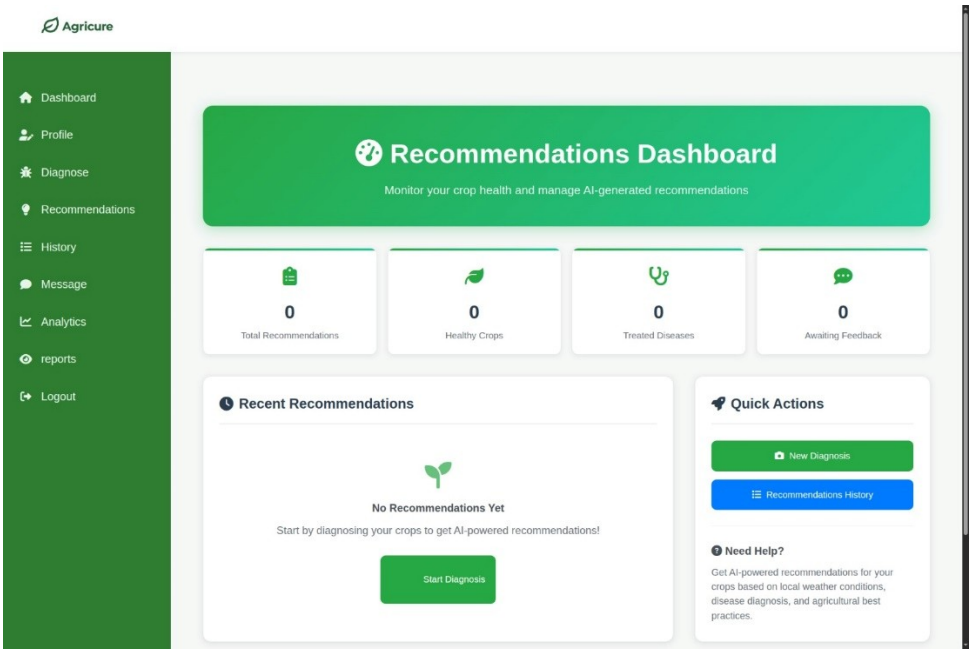


Figure 11:Figure shows the Recommendation Page for the farmer



5.2 Agronomist Interactions

Figure 12: Figure shows the Agronomist dashboard

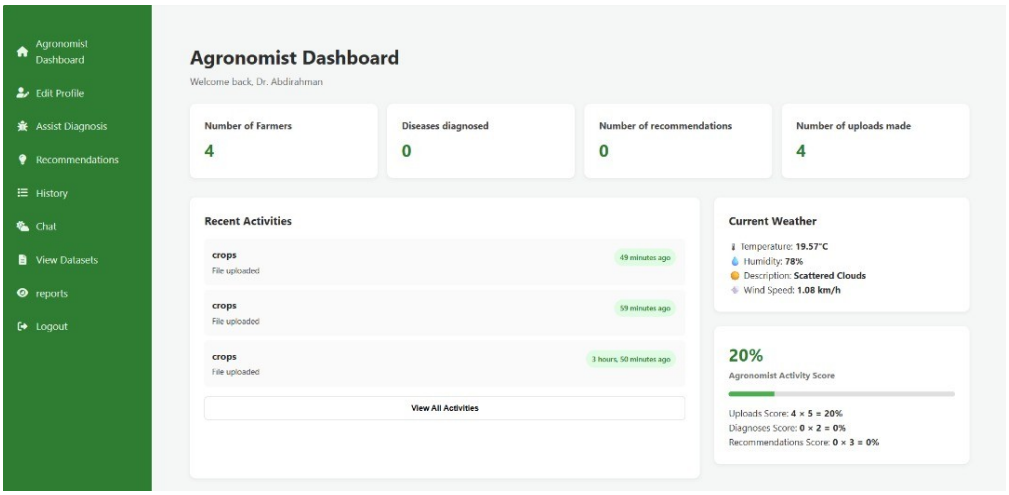
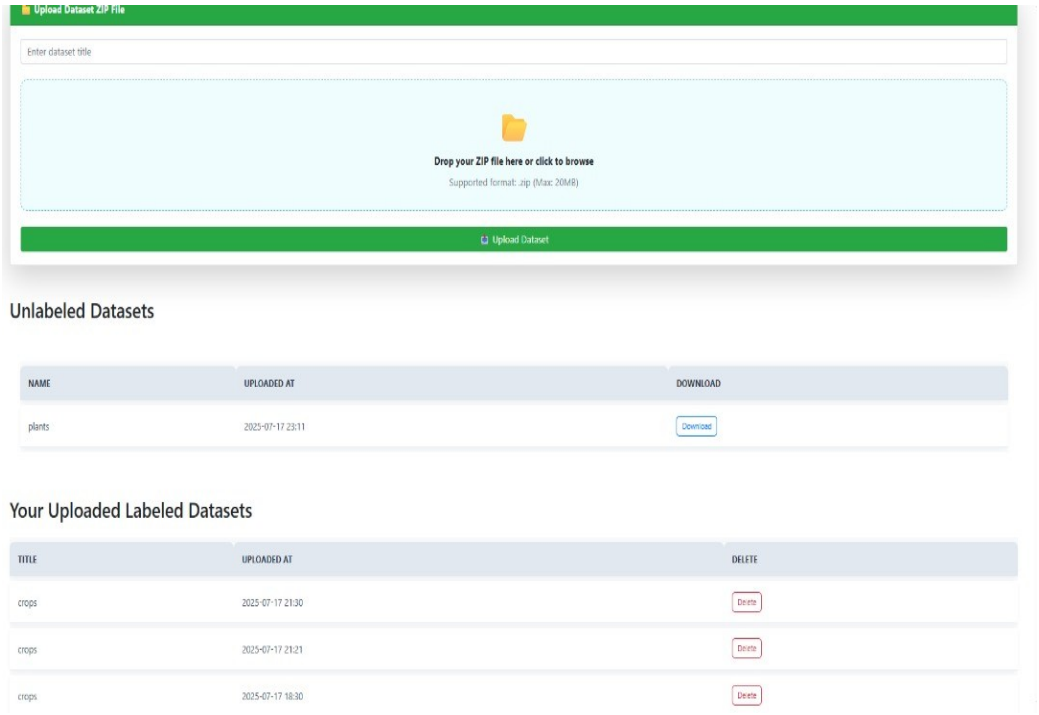


Figure 13: Figure shows the Agronomist page for labelling datasets



5.3 Administrator Interactions

Figure 14: Figure shows the Administrator dashboard

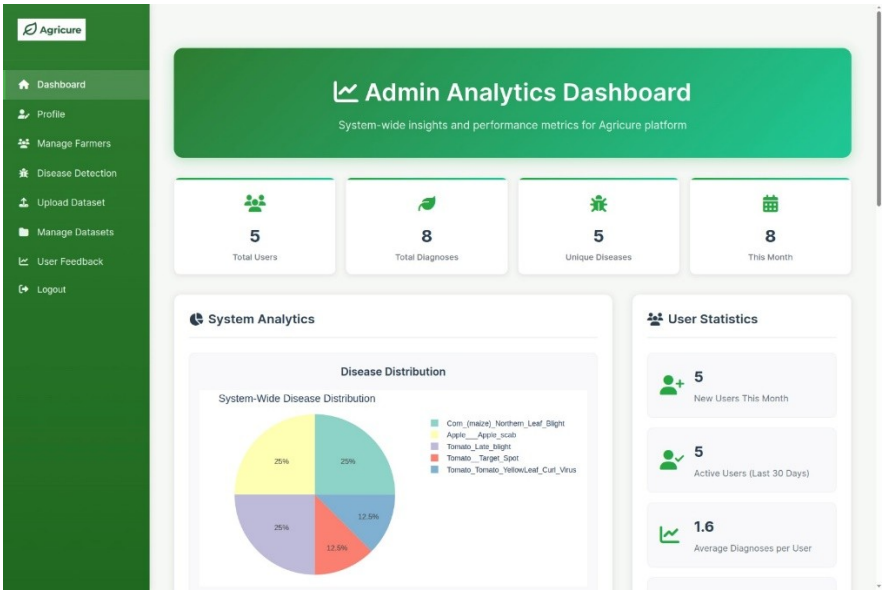
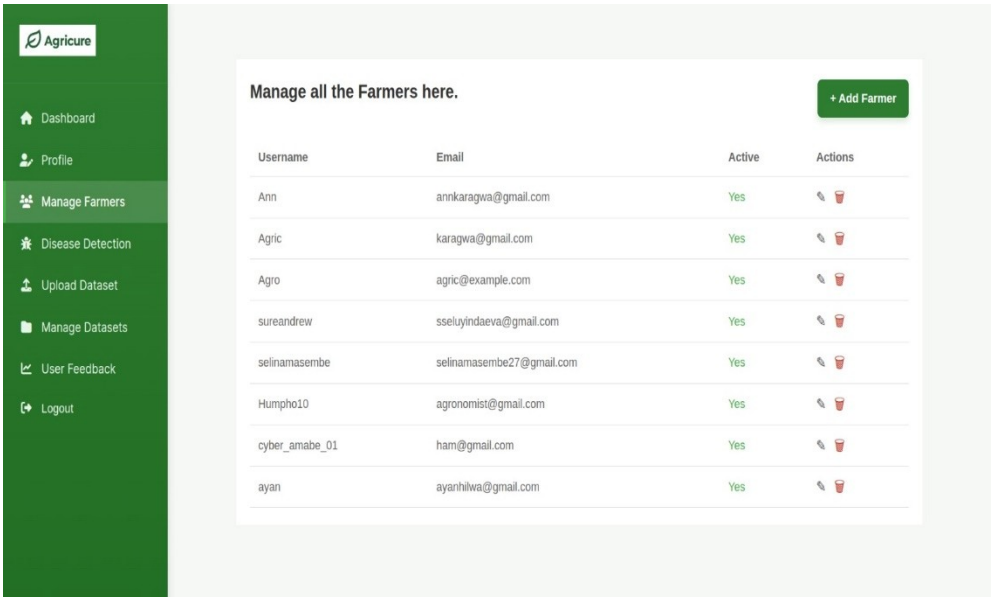


Figure 15: Figure shows the manage farmer's page.



5. CONCLUSION

The Agricure System offers a powerful solution to modernize rural agriculture by bridging technology and traditional practices. Its AI-powered disease detection and advisory features make it a valuable tool in the fight against food insecurity and crop loss.

5.1 Glossary

5.1.1 Definitions and Acronyms

Table 2: Table shows definitions and acronyms

Acronym	**Definition**
UML	Unified Modeling Language
HTTP/SSL	Hypertext Transfer Protocol/ Secure Sockets Layer
RBAC	Role-Based Access Control
ML	Machine Learning
AI	Artificial Intelligence
FR	Functional Requirement
NFR	Non-Functional Requirement