Software Requirements Specification

for

Fall Armyworm Diagnosis

Version 1.0 approved

Prepared by Group BSE25-22

13/10/2024

NAME	REGISTRATION NUMBER
BUTERABA LYNATTE JOANITTA	21/U/09727/PS
SSEBATTA ADAM	21/U/0534
NABUKENYA MARIAM	21/U/17130/EVE
GAFABUSA WILLY	21/U/13385/EVE

TABLE OF CONTENTS

LIST OF TABLES

Revision History

Name	Date	Reason For Changes	Version

CHAPTER ONE

Introduction

1.1. Purpose

The purpose of this Software Requirements Specification (SRS) is to document the requirements for the **Fall Armyworm Diagnosis** mobile application. The application is designed to assist farmers in detecting **Fall Armyworm (FAW)** infestations early by using **machine learning (ML)** to analyze images of crops. Based on detection results, the app will offer actionable recommendations on how to control infestations and suggest preventive practices. The ultimate goal is to reduce crop damage, particularly in maize fields, which are heavily affected by FAW, and to ensure food security by providing real-time, science-based assistance to smallholder farmers.

By integrating machine learning, the application will enhance the precision of disease identification and empower users to make data-driven decisions in managing their crops.

1.2. Document Conventions

This document follows these conventions:

- **High priority (H)**: Critical requirements that must be implemented in the first version.
- **Medium priority (M)**: Important features that are necessary but can be postponed to a later version.
- **Low priority (L)**: Optional features that can be implemented based on time and resources.
- **TBD (To Be Determined)**: Items that are yet to be fully defined.

Font Styles:

Bold text indicates section titles and important terms.

Terminology:

- **User**: Refers to farmers, agricultural stakeholders, and any individuals using the mobile application.
- **System**: Refers to the mobile application being developed for Fall armyworm disease detection.
- **Model**: Refers to the machine learning algorithm integrated into the application for disease detection.

1.3. Intended Audience and Reading Suggestions

The primary audience for this document includes:

- **Software developers and engineers** who will design and implement the mobile application.
- **Project managers** who will oversee the development process.
- **Agricultural experts and researchers** who will provide data and insights into Fall Armyworm behavior and control practices.
- **Stakeholders such as farmers' cooperatives, government bodies, and NGOs** involved in pest control and agricultural sustainability.

Readers are advised to start with the **Overall Description** to get an understanding of the project and then move on to **System Features** for detailed technical requirements.

1.4. Product Scope

The Fall Armyworm Diagnosis Project aims to develop a specialized, machine learning-based system for the early detection, diagnosis, and management of Fall Armyworm (FAW) infestations in maize crops. Focused exclusively on maize, which is crucial to the livelihoods of smallholder farmers in countries like Uganda, this project targets the primary crop impacted by FAW, directly supporting Uganda's goals of enhancing food security, reducing crop losses, and improving pest management through accessible technology.

This system will leverage image analysis to identify FAW infestations by processing photographs of maize leaves, either uploaded by users or captured in real time. By offering timely detection and actionable recommendations, the app will support farmers and agricultural experts in:

- Detecting infestations early to minimize crop damage and yield losses.
- Receiving tailored recommendations for treating infestations, using cultural, biological, and chemical control measures.
- Implementing preventive strategies to reduce the risk of future FAW outbreaks in maize crops.

The project will focus on four key areas to ensure the system's effectiveness and scalability:

- **Image Data Collection**: Gathering a diverse dataset of maize crop images, including both healthy and FAW-infested plants, to train the machine learning model effectively.

- **Machine Learning Model Development**: Creating and refining a model capable of accurately identifying FAW infestations based on symptoms such as leaf damage, larvae presence, or frass (excrement).
- **System Validation and Testing**: Validating the model's accuracy and reliability with real-world data from farms and research institutions in high-risk FAW regions.
- **Recommendation Engine**: Providing actionable, context-based pest management recommendations that encompass biological controls, cultural practices, and selective pesticide use.

1.5. References

Fall Armyworm IPM Guide for Africa - A detailed guide on integrated pest management for FAW.

National Agricultural Research Organization (NARO) Guidelines - Parasitoid Distribution and Parasitism of the Fall Armyworm Spodoptera frugiperda (Lepidoptera: Noctuidae) in Different Maize Producing Regions of Uganda

TensorFlow Lite Documentation - Technical reference for integrating machine learning models into mobile applications.

CHAPTER TWO

2.1. Overall Description

2.1.1. Product Perspective

The Fall Armyworm Diagnosis App shall be a standalone mobile application designed to assist smallholder maize farmers in detecting and managing Fall Armyworm (FAW) infestations on their crops. This application will leverage advanced machine learning, image recognition, and predictive modeling techniques to identify FAW infestations and provide actionable insights. Additionally, language APIs will be implemented to make the application accessible to users in multiple languages, ensuring it can be widely adopted in diverse regions.

Context and Origin

This project addresses the growing need for a technology-driven solution to combat the Fall armyworm threat faced by maize farmers in Uganda. Traditionally, farmers rely on visual inspections to detect pest damage, which is inefficient and prone to human error. By using neural networks to analyze crop images, this mobile application provides a faster, more accurate diagnosis, empowering farmers to take timely action. The integration of language APIs also ensures that the app supports multiple languages, enabling farmers from various linguistic backgrounds to use it effectively.

System Overview

The system will consist of a mobile application that incorporates neural network models for precise disease detection, a cloud-based backend for data storage, and language APIs for multilingual support. The app will allow users to upload images of potentially infected crops, which are processed by the neural network model to detect the presence of Fall armyworm. It will then offer pest control recommendations in the user's preferred language.

Major Components:

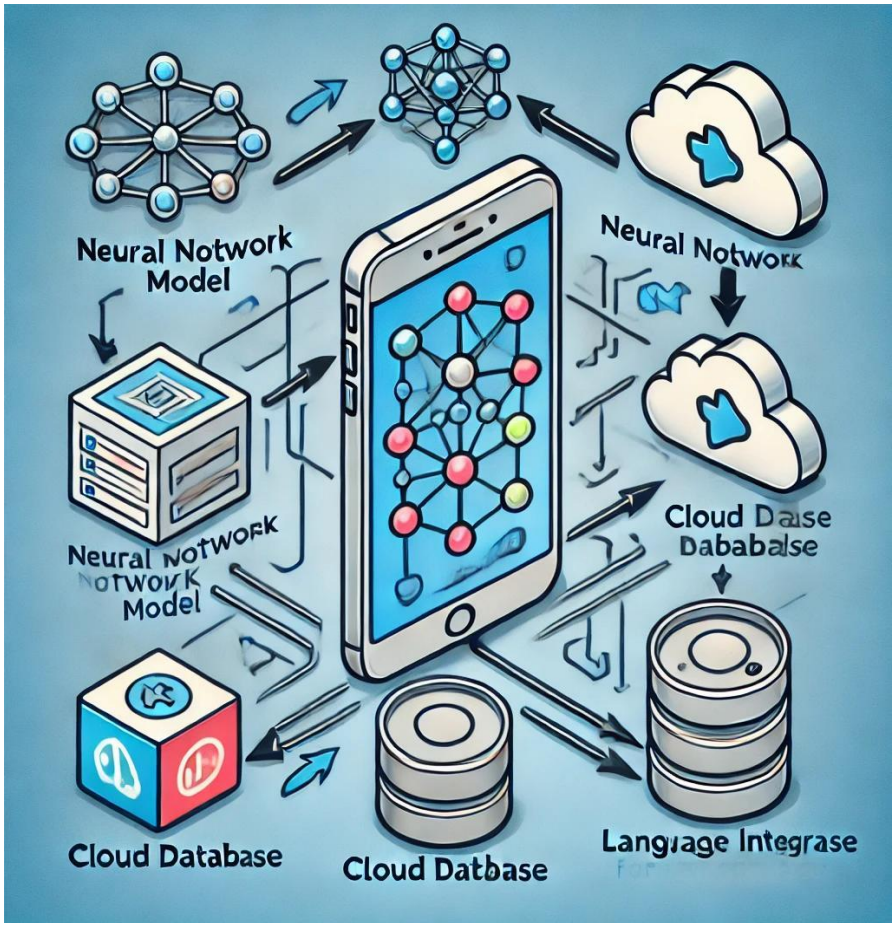
- **Mobile Application**: The user interface where farmers upload images of their crops for analysis, select their preferred language, and receive diagnostic results and recommendations.

- **Neural Network Model**: The machine learning engine, implemented as a convolutional neural network (CNN), used for image classification to detect Fall armyworm disease with high accuracy.
- **Cloud Database**: A remote database that stores uploaded images, results, and historical data for future analysis and to improve model performance.
- **Language APIs**: APIs that facilitate real-time translation, ensuring the app can deliver content in multiple languages to cater to diverse user demographics.
- **External Interfaces**: The app may integrate with external agricultural databases and services for additional pest management insights and regional farming data.

Product Interfaces

- **User Interface (UI)**: A simple and intuitive interface allowing farmers to interact with the app by uploading images, selecting a language, and receiving diagnostic results.
- **API Interface**: The application communicates with cloud-based servers where the neural network models and language APIs are hosted, sending requests and receiving processed data.
- **Language APIs**: The app implements language translation APIs (e.g., Google Cloud Translation) to deliver content in various languages, enabling localization for a wider user base.

The following diagram illustrates the major components and their interactions:



2.2. Product Functions

The Fall armyworm disease detection mobile application will provide the following major functions:

- **Image Capture and Upload**: Users shall have the ability to capture images of maize plants directly within the app, upload images from their gallery, or conduct real-time scanning for immediate FAW diagnosis.
- **FAW Detection Using Machine Learning**: The app shall use a machine learning model to analyze uploaded or captured images and detect signs of FAW, assigning a severity rating based on infestation levels.
- **Treatment and Prevention Recommendations**: Based on the severity of the detected infestation, the app shall provide recommendations on treatment, including biological,

chemical, and cultural practices. The app will also suggest preventive strategies to minimize future infestations.

- **Multi language Support**: The app will support multiple languages through language APIs, ensuring accessibility for users from different regions.
- **User Notifications**: Automated notifications will be sent to users, informing them of relevant updates and pest management techniques.
- **Push Notifications for Disease Risks**: The app will send alerts when environmental conditions, such as humidity or temperature changes, increase the likelihood of disease outbreaks, prompting preventive actions.

2.3. User Classes and Characteristics

- **Farmers**: Primary users of the app, with limited technical expertise, requiring an intuitive and simple-to-navigate interface to access diagnostic and treatment functions.
- **Agricultural Extension Workers**: Users who shall use the app to assist farmers and validate diagnostic results in the field. They may also provide feedback on treatment effectiveness.
- **Researchers and Experts**: Users interested in analyzing diagnostic and outbreak data for research on FAW infestation patterns and pest control measures.

2.4. Operating Environment

The app shall operate in the following environments:

- **Mobile Operating Systems**: Android version 5.0 and above, and iOS version 10 and above.
- **Hardware Requirements**: Smartphones equipped with a camera. While basic internet connectivity is recommended for updates and external data, the core functionality, including image processing and FAW detection, will be accessible offline.
- **Software Dependencies**: TensorFlow Lite for on-device machine learning inference, enabling offline functionality. The app will also integrate an external weather API, used when internet connectivity is available, to provide additional context for pest management recommendations.

This design will ensure that users can access essential features of the app without internet connectivity, allowing farmers in remote areas to detect and manage FAW infestations effectively.

2.5. esign and Implementation Constraints

- **Machine Learning Model Optimization**: The machine learning model shall be optimized for mobile devices using TensorFlow Lite to ensure efficient and accurate image recognition while conserving device resources.
- **Network Dependency**: Although the app will allow offline image capture and diagnosis, internet connectivity shall be required periodically to fetch weather updates, synchronize outbreak data, and receive model updates.
- **Data Privacy and Compliance**: The app shall comply with applicable data privacy regulations, ensuring that user images and diagnostic information are stored securely and only used for the intended purposes.

2.6. User Documentation

The following user documentation shall be provided:

- **User Manual**: A detailed guide outlining app functionality, instructions for image capture, interpretation of results, and treatment recommendations.
- **In-App Help Section**: A FAQ and troubleshooting section accessible within the app, offering step-by-step guides on common tasks and potential solutions for common issues.

2.7. Assumptions and Dependencies

Assumptions:

It is assumed that users will have access to a smartphone with basic internet connectivity for weather updates and other periodic functions.

Lighting and Image Quality: It is assumed that users will capture images under adequate lighting conditions, as poor lighting may reduce the accuracy of FAW identification.

Environmental Conditions: It is assumed that farmers will take images of maize plants primarily in outdoor settings where lighting and environmental conditions are suitable for image-based analysis.

Dependencies:

The app will depend on third-party APIs for weather data, and on regular model updates to improve accuracy and relevance over time.

CHAPTER THREE

3.1. External Interface Requirements

3.1.1. User Interfaces

The app will consist of several primary interfaces to ensure intuitive navigation and easy access to all features. The following describes each interface's layout and functionality, with additional design specifications documented in a separate user interface specification.

Home Screen:

Description: The main interface will welcome users and serve as the central navigation hub.

Features:

- **Image Upload Option**: Allows users to upload an existing photo of a maize plant for FAW analysis.
- **Photo Capture Option**: Enables users to take a new photo directly within the app for instant analysis.
- **Real-Time Scanning Option**: Activates the camera for live scanning of maize crops, offering immediate diagnostic feedback.
- **Forum Access Button**: Provides access to a user forum where users can discuss pest management techniques, share success stories, and seek advice from other farmers and experts.

Layout: A simple, clean layout with large, easy-to-tap buttons for each option, suitable for users of varying technical literacy levels.

Diagnosis Screen:

Description: Displays the diagnostic results after processing an uploaded or captured image.

Features:

- **FAW Detection Result**: Clearly indicates whether FAW has been detected.
- **Severity Level**: Shows the estimated level of infestation severity, helping users assess the urgency.
- **Symptoms Display**: Lists visible symptoms identified in the image, such as leaf damage, larval presence, or frass.

Layout: Results will be displayed prominently, with easy-to-read text and icons indicating FAW presence and severity. Error messages or prompts will appear if image quality is insufficient.

Recommendations Screen:

Description: Provides treatment and preventive measures based on the diagnosis.

Features:

- **Biological Control Suggestions**: Lists natural pest control options, such as introducing specific predators.
- **Chemical Control Guidelines**: Offers safe pesticide options with usage instructions.
- **Cultural Practices**: Recommends practices like crop rotation or intercropping to minimize FAW impact.

Layout: Information will be displayed in expandable sections, allowing users to easily view and implement recommendations based on preference and resource availability.

Forum Screen:

Description: A dedicated space where users can connect, share experiences, and discuss FAW management practices.

Features:

- **Discussion Threads**: Users can start or participate in threads on various topics related to FAW, pest control, and crop management.
- **Expert Interaction**: Agricultural experts may contribute insights or answer user questions within specific threads.
- **User Profiles**: Displays basic profile information for each user, fostering community engagement.
- **Like and Reply Functions**: Allows users to like posts and reply to comments to promote helpful information.

Layout: A scrollable list of discussion topics with the ability to filter by category. Each thread will include options for replying and viewing replies in a nested format.

Standard Interface Elements:

- **Navigation Bar**: Located at the bottom of every screen, allowing quick access to the Home, Diagnosis, Recommendations, and Forum screens.

- **Help Button**: Positioned on each screen to provide users with quick tips and FAQs.
- **Error Messages**: Clearly worded messages will appear if a feature cannot load or if input (e.g., image quality) is insufficient, guiding users on corrective actions.

Authentication Screens (Conditional Access)

Login Screen (Displayed only when accessing Recommendations or Forum):

Description: Appears when a user tries to access the **Recommendations** or **Forum** screen without being logged in.

Features:

- **Username/Email Field**: Allows users to enter their username or email.
- **Password Field**: A secure field for password input.
- **Forgot Password Option**: A link to reset passwords for convenience.

Layout: Clear, straightforward, with login prompts that guide users to log in before accessing certain sections.

Signup Screen:

Description: Presented to new users who want to access **Recommendations** or **Forums** and need to create an account.

Features:

- **Username, Email, Password Fields**: Standard registration fields for creating an account.
- **Terms and Conditions Checkbox**: Ensures users agree to app policies.

Layout: Simple layout with helpful prompts, like password guidelines and clear button navigation, to ease the sign-up process.

Conditional Access Notifications:

- **Notification Prompt**: When a user will want to access **Recommendations** or **Forum** while logged out, they will receive a prompt notifying them that these sections require an account.
- **Options**: Users will select to **Log In** or **Sign Up**, redirecting them to the appropriate screen.

User Profile Screen (Will be accessed once logged in):

Description: Will allow users to view and manage their profile, update preferences, and log out.

Features:

- **Profile Editing**: Users will edit their information and preferences.
- **Logout Button**: Users will log out easily from their account.

Layout: Organized with clear labels for profile options and easy navigation.

3.2. Hardware Interfaces

Camera Access: The app will use the device's camera for real-time image capture and scanning. It will require permission to access the camera for this functionality.

3.3. Software Interfaces

TensorFlow Lite Integration:

- **Description**: TensorFlow Lite will be embedded as the machine learning framework to perform on-device image processing and FAW detection.
- **Data Interaction**: Images captured or uploaded by users will be processed by the TensorFlow Lite model, which will return a diagnosis of FAW presence and severity.
- **Purpose**: Using TensorFlow Lite will ensure efficient and low-latency processing, enabling users to detect FAW even in offline scenarios.
- **Compatibility Requirements**: The app will be compatible with devices that support TensorFlow Lite, ensuring that model inference can happen on-device without constant network connectivity.

Local Database (Optional for Offline Data Storage):

- **Description**: A local database, such as SQLite, will be used to store user-submitted images, diagnostic history, and treatment recommendations.
- **Data Interaction**: Data generated from FAW detection and treatment suggestions will be stored and accessible offline, allowing users to track pest management without internet access.

- **Purpose**: To enhance user experience by retaining data locally and reducing dependency on internet connectivity.
- **Data Sharing Mechanism**: SQLite storage will retain information across sessions, ensuring persistent access to diagnostic history.

3.4. Communications Interfaces

Network Protocols:

- **Description**: HTTPS will be utilized for all interactions with external APIs, ensuring secure data transmission for weather data, model updates, and other periodic communications.
- **Communication Security**: HTTPS will protect data in transit, and any sensitive data will be encrypted to maintain user privacy.

Data Synchronization:

- **Description**: Periodic synchronization will be used to update machine learning models and retrieve fresh weather data.
- **Mechanism**: The app will perform scheduled API calls to sync these updates, requiring intermittent internet connectivity.
- **Synchronization Constraints**: Users will periodically connect to the internet to receive updated models or real-time weather data.

Error Handling & Offline Mode:

- **Description**: Error messages will appear if there's no internet connection for retrieving weather data or syncing model updates. Diagnostic and image analysis features will remain accessible offline due to TensorFlow Lite's on-device capabilities.
- **Formatting & Standards**: Standard error dialogs will inform users when connectivity issues arise, ensuring transparency.

CHAPTER FOUR

4.1. System Features

4.2. Image-Based FAW Detection

4.2.1. Description and Priority

USE-CASE ELEMENT	DESCRIPTION
System Feature	Image-Based FAW Detection
Use Case Name	Detect FAW Infestation in maize leaves
Primary Actor	Farmer
Preconditions	The user must have a device with camera capabilities The app must be installed and accessible.
Goal	To enable users to detect FAW infestations early, providing actionable insights to help manage and mitigate the spread of FAW.
Flow of Events	1. The user opens the app and selects an option to upload or capture an image of the maize leaf. 2. The system displays options for image upload or camera capture, allowing the user to choose.

	3. The user uploads or captures an image of the maize leaf. 4. The system receives the image and processes it using a machine learning model to detect signs of FAW infestation and assess the severity level. 5. The system analyzes the image and generates diagnostic results, including information on whether FAW is present and the severity of the infestation. 6. The system displays the diagnostic results to the user, with severity level and a recommendation to view treatment options.
Postconditions	The app provides a diagnosis of FAW infestation based on the image analysis.

The app's primary feature will enable users to upload, capture, or scan maize leaves images in real-time to detect FAW infestation.

This high-priority feature shall use machine learning to analyze images for FAW symptoms, assess severity, and display diagnostic results.

4.2.2. Stimulus/Response Sequences.

Action: Users will capture or upload an image of the maize leaf.

Response: The app will analyze the image and display a diagnosis indicating the presence or absence of FAW and the severity of the infestation.

Table 1: Use Case: FAW Detection through Image Analysis

4.3. Functional Requirements

REQ-1.1: The app shall provide an option for the user to upload an image from the gallery or capture a new image using the camera when the app is opened.

REQ-1.2: The app shall display both upload and capture options on a single interface, allowing the user to choose their preferred method.

REQ-1.3: The app shall allow the user to select and upload an image from their gallery or capture an image directly using the in-app camera.

REQ-1.4: The app shall receive the uploaded or captured image and process it through a machine learning model to detect signs of FAW infestation.

REQ-1.5: The app's machine learning model shall analyze the image for signs of FAW infestation, identifying any patterns associated with FAW and determining the severity level.

4.4. Treatment Recommendations

4.4.1. Description and Priority

This high-priority feature will provide treatment options based on the detected severity of FAW infestation. The app shall include information on biological, chemical, and cultural treatment methods tailored to the severity level.

4.4.2. Stimulus/Response Sequences

- **Action**: The app diagnoses FAW infestation and assesses its severity.
- **Response**: The app shall display customized treatment recommendations, including details on implementation and safety for each method.

Table 2: Use Case: FAW Treatment Recommendations Based on Infestation Severity

USE-CASE ELEMENT	**DESCRIPTION**
System Feature	Treatment Recommendations
Use Case Name	Provide FAW Treatment Options
Actors	**Primary Actor:** User (farmer, agricultural technician)

Preconditions	FAW infestation severity has been diagnosed and is available in the app.
Goal	To equip users with effective treatment methods based on the severity of FAW infestation, helping manage and reduce pest impact.
Flow of Events	1. The user will view FAW diagnosis results and select "Treatment Options." 2. The system will retrieve and display tailored treatment recommendations based on infestation severity. 3. The user will review treatment details and select a preferred option. 4. The system will provide additional information on the chosen method, including materials and application guidelines. 5. The system will periodically update treatment recommendations to ensure they align with the latest best practices in pest management.
Postconditions	The user receives relevant, severity-based treatment recommendations to manage the infestation effectively.

4.4.3. Functional Requirements

REQ-2.1: The app shall define multiple diagnostic severity levels (e.g., mild, moderate, severe) to classify FAW infestation based on diagnostic data.

REQ-2.2: The app shall categorize each FAW infestation input within a severity level based on diagnostic analysis.

REQ-2.3: The app shall provide biological treatment options specific to each severity level of FAW infestation.

REQ-2.4: The app shall provide chemical treatment options specific to each severity level, with clear instructions for safe and appropriate usage.

REQ-2.5: The app shall provide cultural treatment options specific to each severity level, focusing on sustainable pest management methods.

REQ-2.6: The app shall display treatment options in a user-friendly format, ensuring easy navigation and understanding for all users.

REQ-2.7: The app shall include clear icons or labels to distinguish between biological, chemical, and cultural treatment methods within the user interface.

REQ-2.8: The app shall implement a periodic update mechanism that refreshes treatment information on a predefined schedule (e.g., monthly or quarterly).

REQ-2.9: The app shall fetch new treatment guidelines from verified sources during each update cycle to ensure information is current.

REQ-2.10: The app shall notify users via push notifications when new treatment information is updated or added.

REQ-2.11: The app shall provide users access to a summary of recent updates to treatment information, allowing them to stay informed about changes.

REQ-2.12: The app shall offer a feedback feature for users to report the effectiveness of specific treatments, contributing to the ongoing improvement of recommendations.

4.5. Preventive Measures

4.5.1. Description and Priority

This medium-priority feature will offer preventive strategies that help users mitigate the risk of future FAW infestations. Strategies shall include best practices such as crop rotation, early planting, and natural pest deterrents.

4.5.2. Stimulus/Response Sequences

- **Action**: Users will access preventive tips from the app.
- **Response**: The app shall display preventive measures and explain their significance for FAW management.

Table 3: Use Case: Preventive Measures

USE-CASE ELEMENT	**DESCRIPTION**
System Feature	Preventive Measures
Use Case Name	Provide FAW Preventive Strategies
Actors	**Primary Actor:** User (farmer, agricultural technician)

Preconditions	The app is accessible, and the user is logged in.
Goal	To equip users with strategies to prevent FAW infestations, supporting long-term pest control and crop health.
Flow of Events	1. The user will access the "Preventive Tips" section in the app. 2. The system will retrieve preventive strategies. 3. The system will display each preventive measure with a brief explanation of its effectiveness in reducing FAW risk. 4. The user will review the list and note strategies applicable to their specific agricultural practices.
Postconditions	The user receives preventive measures to reduce the likelihood of future FAW infestations.

4.5.3. Functional Requirements

REQ-3.1: The app shall display a list of preventive measures that users can implement to mitigate the risk of future FAW infestations.

REQ-3.2: The app shall include crop rotation as a recommended preventive measure against FAW.

REQ-3.3: The app shall include early planting as a recommended preventive measure against FAW.

REQ-3.4: The app shall include the use of natural pest deterrents as a recommended preventive measure against FAW.

REQ-3.5: The app shall present preventive measures in a user-friendly format, with brief descriptions for each measure to enhance user understanding.

CHAPTER FIVE

5.0. Other Nonfunctional Requirements

5.1. Performance Requirements

- **Image Processing Speed**: The app shall deliver diagnostic results within 2-3 seconds after an image is uploaded or captured. This speed will be achieved by optimizing the machine learning model using TensorFlow Lite for on-device processing, which reduces latency compared to server-side processing.
- **Predictive Modeling Efficiency**: The app shall fetch and process weather data within 5 seconds to provide timely outbreak alerts. This will involve integrating a lightweight weather API and setting up periodic background syncs, reducing processing delays and improving responsiveness for users in real-time settings.
- **Real-Time Scanning**: For the real-time scanning feature, the app shall maintain a frame rate of at least 15 frames per second (fps) to ensure smooth scanning without lag. This will be achieved by limiting resource-intensive processes during scanning and prioritizing critical image frames for analysis.

5.2. Safety Requirements

- **Data Storage Safeguards**: User data, especially images, shall be stored with safeguards to prevent loss, unauthorized access, or modification. The app will use secure cloud storage solutions with redundant backups to prevent data loss due to system failures.
- **Privacy and Data Usage**: To protect users' information, the app shall request permission before accessing the camera or storage, with a clear explanation of data usage. The app will prevent the transfer of images and personal data to unauthorized entities.
- **User Guidance for Accurate Diagnosis**: The app shall include warnings and user tips to ensure accurate image capture, such as instructions on taking clear photos in adequate lighting and avoiding blurred images. These guidelines will reduce the likelihood of misdiagnosis or inaccurate recommendations.
- **Compliance with Local Policies**: The app will adhere to Uganda's data protection regulations to ensure users' information is safeguarded, especially given its use in agriculture, where specific policies may apply.

5.3. Security Requirements

- **Secure Data Transmission**: All data transmissions, including image uploads, weather data requests, and model updates, shall be encrypted using HTTPS to protect user privacy. This encryption will prevent data interception during transmission between the app and remote servers.
- **User Authentication**: The app shall implement optional user authentication for access to advanced features, such as history tracking and custom outbreak alerts. This will be done through a simple username and password setup, or via OAuth for social media accounts.
- **Data Anonymization**: Collected data, such as user locations and diagnostic results, shall be anonymized and aggregated when shared for research purposes to ensure privacy. Personal identifiable information (PII) will not be included in shared datasets.
- **Security Certifications**: The app shall meet security certification standards relevant to mobile applications and cloud storage to reassure users of data protection and privacy, particularly for storage and transmission of sensitive data.

5.4. Software Quality Attributes

- **Adaptability**: The app will be designed to allow easy updates to machine learning models and treatment recommendations. It shall have a modular architecture so that additional crop types or pests can be integrated in future versions.
- **Availability**: The app shall have an availability rate of 99% to minimize downtime, ensuring reliable service during peak farming periods. This will be achieved through the use of cloud hosting and backup servers.
- **Correctness**: The app shall provide accurate FAW detection, with a minimum accuracy of 90% in diagnosing FAW symptoms based on machine learning tests. Data collection and model training will be continuously refined to maintain this standard.
- **Flexibility**: The app will support multi-language functionality to accommodate users in Uganda with English and Luganda as the initial supported languages.
- **Interoperability**: The app shall be compatible with both Android and iOS platforms, ensuring accessibility to a wide range of users.

- **Maintainability**: The app shall be developed using modular coding practices, allowing for efficient updates and maintenance by developers. Clear documentation and adherence to coding standards will also support maintainability.
- **Portability**: The app shall function effectively on various Android and iOS devices with different screen sizes and hardware capabilities, supporting usage across a range of smartphones.
- **Reliability**: To ensure reliability, the app will have built-in checks for network connectivity before accessing weather data or model updates. Local caching will be used to store diagnostic history, so users can still access information even when offline.
- **Reusability**: Certain core components, like the image recognition module, shall be designed as reusable modules that can be implemented in other pest diagnosis applications.
- **Robustness**: The app will have built-in error handling to manage unexpected issues, such as failed image uploads or connectivity issues. Users shall receive clear error messages and guidance on resolving issues.
- **Testability**: The app will be developed with automated testing capabilities for continuous integration and deployment, ensuring every update maintains system integrity.
- **Usability**: The app interface will be intuitive, with clear icons, large buttons, and accessible language. Tooltips and in-app guidance shall support users in understanding each feature.

5.5. Business Rules

Authenticated User Access:

- **Diagnostic History & Data Saving**: Only authenticated users will save diagnostic results and view their diagnostic history. This feature will enable users to track past assessments, observe trends, and make informed decisions based on historical data.
- **Recommendations & Forum Access**: Access to the **Recommendations** and **Forum** screens will be restricted to authenticated users only. Non-authenticated users will perform diagnostic scans but will not be able to access treatment recommendations, participate in the forum, or receive location-based FAW outbreak alerts.

Data Collection & User Consent:

- **Permissions**: The app shall request user consent before collecting personal or diagnostic data. Authenticated users will be given the option to opt out of data sharing for research purposes while still being able to use diagnostic features.
- **Data Privacy**: Users who opt out of data sharing will still be able to save personal diagnostic history locally, without contributing this data to research databases or cloud storage.

Notifications & Updates:

- **In-App Update Notifications**: Authenticated users will receive in-app notifications for critical updates, such as new treatment recommendations or adjustments to predictive models, ensuring they have the most recent information to address FAW threats effectively.
- **Predictive Alerts for Outbreaks**: Only users who opt into location sharing will receive predictive alerts regarding potential FAW outbreaks based on local weather data, which can help them preemptively protect their crops.

Access Control & User Interface Adjustments:

- **Login Prompt for Restricted Features**: If a non-authenticated user attempts to access restricted features (such as the **Recommendations** or **Forum**), they will see a login prompt with options to log in or register. This prompt will include:
- **Username/Email Field** will be used for login input.
- **Password Field** will be used for secure access.
- **Forgot Password Option** will be for easy password recovery.
- **Clear Layout & Guidance** will be used for ease of use, prompting the user to authenticate to access the additional functionality.

Appendix A: Glossary

- **FAW (Fall Armyworm)**: A significant agricultural pest affecting maize crops, known scientifically as *Spodoptera frugiperda*. The application is designed to assist in its early detection and management.
- **API**: Application Programming Interface, a tool to connect the app with external weather data.
- **Neural Network Model**: A machine learning framework that mimics the human brain's network of neurons to process input data (e.g., images of maize leaves) and produce output (e.g., detection of FAW).
- **Machine Learning (ML)**: A subset of artificial intelligence that enables the application to analyze images and make predictions based on data without being explicitly programmed for specific tasks.
- **Image Capture and Upload**: The feature that allows users to take pictures of maize plants using their smartphone camera or upload images from their device's gallery for analysis.
- **Image Analysis**: The process of examining digital images using machine learning algorithms to identify and assess the presence of FAW infestations in maize crops.
- **Multi-language Support**: The application's capability to provide content in various languages, enhancing accessibility for users from diverse linguistic backgrounds.
- **Real-Time Scanning**: The capability of the application to analyze images immediately upon capture, providing instant feedback on the presence of FAW.
- **User Interface (UI)**: The graphical layout of the application that users interact with, including screens, buttons, and navigation elements.
- **Video and Forum Access**: Features that allow users to engage in discussions and share experiences related to FAW management within the application community.
- **HTTPS**: Hypertext Transfer Protocol Secure, a protocol used for secure communication over a computer network, ensuring that data exchanged between the user and the application is encrypted.
- **Data Synchronization**: The process of updating the application's data by retrieving new information from external sources, such as weather APIs and model updates.
- **Recommendations Engine**: The system component responsible for generating treatment and preventive suggestions based on diagnostic results and severity ratings.

- **Cloud Database**: A storage solution hosted on the internet that allows the application to store and retrieve data, enabling functionalities like user profiles and diagnostic history.

  Acronyms used

  FAW: Fall Armyworm

  SRS: Software Requirements Specification

  HTTPS: Hypertext Transfer Protocol Secure

  API: Application Programming Interface.

  UI: User Interface.

  FAQ: Frequently Asked Questions.

  CNN: Convolutional Neural Networks

  ML: Machine Learning.

Appendix B: Analysis Models

- **System Architecture Diagram**. The system architecture diagram will illustrate the entire setup, including interactions between client devices (e.g., smartphones for field diagnostics), a cloud-based server for processing, and a storage system for data and model management. This diagram will help clarify the layout of the data pipelines and the placement of components (e.g., database, model server) in the overall structure.
- **Data Flow Diagram (DFD)**. This diagram will illustrate the flow of data, starting from image inputs (e.g., photos of potentially infested crops) through stages of pre-processing, model inference, and result generation. It will show how data will be received, processed, and used to produce the final diagnosis.
- **State-Transition Diagram**. This diagram will define the different states in the diagnostic process, such as Image Upload, Preprocessing, Diagnosis in Progress, and Diagnosis Complete. The state-transition model will capture the dynamic behavior of the diagnostic system as it progresses through each phase.

Appendix C: To Be Determined List

- **Weather Data Provider.** The application will use the Google Weather API to integrate real-time weather data. Further configuration details and integration specifics will be finalized as development continues to ensure optimal data accuracy and performance.
- **User Interface Design Details.** Additional details on the user interface elements, design standards, and user experience optimizations are under development. These will be finalized to ensure a cohesive and user-friendly interface that aligns with the project's goals and user needs