Software Requirements Specifications

AI Assistant for Plant Disease Diagnosis

Project Code:	FYP2226G
Internal Advisor:	Dr. Hussam Ali
External Advisor:	N/A
Project Manager:	Prof. Dr. Muhammad Iliyas
Project Team:	M Taha Mukhtar BSCS51F22S004 (Team Lead) Hammad Ahmad BSCS51F22S012 (Team Member) Usama Farooq BSCS51F22S045 (Team Member)
Submission Date:	10th Nov 2025

[illegible]
10~Nov-25

Project Manager's Signature

Document Information

Category	Information
Customer	University of Sargodha
Project	AI Assistant for Plant Disease Diagnosis (Agro Guard)
Document	Requirement Specifications
Document Version	4.0
Identifier	PGBH01-2003-RS
Status	Approved Version
Author(s)	M Taha Mukhtar, Hammad Ahmad, Usama Farooq
Approver(s)	Prof. Dr. Muhammad Iliyas
Issue Date	OCT 20th,2025
Document Location	
Distribution	1. Advisor 2. PM 3. Project Office

Definition of Terms, Acronyms and Abbreviations

Term	Description
AI	Artificial Intelligence
API	Application Programming Interface
XML	Extensible Markup Language
JSON	JavaScript Object Notation
GPS	Global Positioning System
ML	Machine Learning
PNG	Portable Network Graphics
JPEG	Joint Photographic Expert Group

Table of Contents

1. Introduction

1.1 Purpose of Document

This document describes the functionality, objectives, and scope of the Plant Disease Detection System. It describes how the system works, its intended users, and the benefits it aims to deliver.

This document is intended for:

1. **Developers:** To understand the system's architecture, functionality, and technical requirements.

2. **Project Supervisors/Stakeholders:** To evaluate the project's objectives, scope, expected outcomes.

End Users (Farmers): To understand how the system assists in diagnosing plant diseases and managing crop health efficiently.

1.2 Project Overview

The **Plant Disease Detection** is an AI-powered mobile application developed using **Kotlin.** The system is designed to assist farmers in identifying plant diseases through image recognition and provide effective management recommendations.

Key Features

1. **AI-Based Disease Detection:** Farmers can upload images of their plants, and the AI model analyzes them to detect potential diseases accurately.

2. **Disease Information and Treatment Suggestions:** The system provides detailed information about the detected disease along with preventive and treatment guidelines.

3. **Weather Updates:** Offers live weather information to help farmers make better decisions about planting, irrigation, and crop care.

4. **User-Friendly Interface:** The application provides a simple, intuitive interface that makes disease diagnosis easy for all types of users, including those with limited technical experience.

Goals and Benefits

1. Enhance **crop health** through early and accurate disease detection.

2. Reduce **crop losses** caused by unidentified or untreated diseases.

3. Minimize **dependency on agricultural experts** by providing AI-driven recommendations.

Empower farmers with accessible, reliable, and quick diagnostic support.

1.3 Scope

Included

Features

1. Image-based detection of plant diseases using Artificial Intelligence.
2. Display of disease-related information and suitable treatment or preventive measures.
3. Real-time weather updates relevant to agricultural activities.
4. An intelligent conversational agent that allows farmers to ask questions regarding general agriculture, crop health, and disease management using natural language.

Not Included

1. Advanced agricultural automation tools such as drones for field monitoring.
2. Automated irrigation, pest control, or fertilizer application systems.
3. Marketplace or product-selling functionalities.

2. Overall System Description

2.1 User characteristics

The **Plant Disease Detection and Management System** is mainly designed for **farmers** and people involved in agriculture who want an easy way to identify plant diseases and get useful guidance.

User Type
Farmer:
Farmers are the main users of this app. Most of them may not have technical knowledge, so the system is designed with a simple and easy-to-understand interface. They can upload pictures of their plants and get quick feedback about possible diseases and treatments.

Agricultural Experts (optional users):
Experts or agricultural officers may use the system to verify results, provide recommendations, or use it as a support tool in research or advisory work. The app focuses on being **simple, reliable, and helpful** for all users, even those with minimal technical experience.~~,~~

2.2 Operating environment

The system is developed as a **mobile application** that runs on Android devices.
Operating Details:

1. **Platform:** Android smartphones and tablets.

2. **Programming Language:** Kotlin.

3. **Internet Requirement:** The app requires a stable internet to upload images, fetch weather updates, and display results.

4. **AI Model:** The disease detection model processes images on a cloud server or integrated AI module.

The app is designed to work smoothly on most Android devices with moderate specifications and a reliable internet connection.
.

2.3 System Constraints

The following constraint and limitations may affect how the system operates:
Software Constraints:

1. The app is currently developed for Android only; it is not available for iOS or desktop platforms.

2. Requires active internet access for image uploads.

3. The AI model's accuracy depends on the quality and clarity of the uploaded plant images.

Hardware Constraints:

1. Users need a smartphone with a camera and internet connectivity.

2. The device should have enough memory and storage to run the app and handle image files.

Cultural Constraints:

1. The app's language is English, which may be difficult for users who are not familiar with it. Future updates may include local language support.

Legal Constraints:

1. Users must only upload images they own or have permission to use.

Environmental Constraints:

1. The app is designed for outdoor use, but performance may be affected by poor network signals in rural areas.

User Constraints:

1. The app is made for adults involved in farming; therefore, it avoids complex terms and provides simple text and visuals.

Performance Constraint:
The app is designed to detect plant diseases from uploaded images under normal internet conditions.

3. External Interface Requirements.

3.1 Hardware Interfaces

Mobile Device

Requirements

1. **Platform:** Android smartphones
2. **Minimum Android Version:** Android 8.0 or higher
3. **RAM:** Minimum 2 GB, Recommended 4 GB or higher
4. **Storage:** Minimum 100 MB free space for app installation and data caching
5. **Camera:** Rear camera with minimum 8 MP resolution for plant disease image capture
6. **Network:** Wi-Fi or mobile data connectivity (3G/4G/5G) for real-time data synchronization.

GPS/Location Services

1. The application accesses the device's GPS hardware to obtain location data
2. Location information is used to provide localized weather updates

3.2 Software Interfaces

Mobile Application

Platform

1. **Development Framework**: Kotlin programming language
2. **Target Platform**: Android OS
3. **IDE**: Android Studio
4. **Minimum SDK**: API Level 26 (Android 8.0)
5. **Target SDK**: API Level 35 (Android 15) or latest stable version
6. **UI Framework**: Jetpack Compose or XML-based layouts for user interface design

External APIs:
Weather API
1. **Purpose**: Fetch real-time weather data based on user's location
2. **Data Exchange Format**: JSON
3. **Authentication**: API Key
4. **Data Retrieved**:

4.1 Current temperature and conditions

Machine Learning Model Interface

1. **ML Framework**: TensorFlow Lite for Android
2. **Input**: Plant disease images captured or uploaded by users
3. **Output**: Disease classification with treatment recommendations
4. **Inference**: Cloud-based inference using a Python API (e.g., FastAPI/Flask) hosted on a GPU-enabled server to ensure high-accuracy predictions

3.3 User Interfaces

Dashboard with weather updates and quick access to features Disease
detection screen with camera integration

4. Functional Requirements

4.1 Weather Information Display

1. The system shall display current weather information on the user's dashboard, including temperature, humidity and weather conditions.
2. The system shall automatically detect the user's location using GPS to provide localized weather updates.

4.2 Plant Disease Detection

1. The system shall allow users to capture or upload images of diseased plants through the camera interface.
2. The system shall provide the disease name, and description of the disease.
3. The system shall suggest appropriate treatment methods and preventive measures for the identified disease.

4.3 Disease Detection System (Detailed Workflow)

AI Analysis

1. The system shall preprocess the uploaded image (resizing, normalization) before feeding it to the ML model.
2. The system shall run the preprocessed image through a trained AI Model.
3. The system shall handle cases where no disease is detected (healthy plant) by displaying "No Disease Detected."

Results and Recommendations

1. The system shall provide a detailed description of the disease including symptoms and causes.
2. The system shall list recommended treatment methods including organic and chemical solutions.

4.4 AI Chatbot Interaction

1. The system shall provide a chat interface accessible from the dashboard.
2. The system shall allow users to send text-based queries regarding crop care and disease prevention.
3. The system shall generate responses using a trained Large Language Model (LLM) or predefined agricultural knowledge base.

5. Non-functional Requirements

5.1 Performance Requirements

1. The app should be optimized to run efficiently on Android devices running Android 8.0 or later, with minimal memory and CPU consumption.

2. Image preprocessing (compression/resizing) shall be performed on the device to minimize bandwidth usage.

3. Model inference shall occur on the cloud server. The system must handle network latency gracefully and provide loading indicators to the user during the upload/analysis phase.

4. The app should be available 99.9% of the time, except for planned maintenance.

5.2 Safety Requirements

1. The system should ensure that no harmful actions (e.g., system crash, data corruption) occur due to invalid inputs, such as uploading non-image files or corrupted images.

2. The app should handle all exceptions gracefully, providing clear error messages without exposing internal details.

3. The system should not make direct agricultural treatment recommendations (e.g., pesticide use) without proper disclaimers, to avoid user harm.

5.3 Security Requirements

1. All communications between the app and any remote service must use secure **HTTPS encryption**.

2. Any temporary image data stored during processing must be deleted automatically after prediction.

3. Remote APIs or services must use authenticated and protected endpoints to prevent unauthorized use.

4. The app should clearly state its data privacy policy, specifying that it does not collect,

store, or share personal data.

5. The AI model and backend APIs should be protected from unauthorized access or tampering.

User Documentation

Ease to Use

The app's interface should be simple enough for anyone to use without training.

6. Assumptions and Dependencies

Assumptions

1. Users will have a **stable internet connection** to upload images and receive predictions.
2. Users' devices will have a **functional camera** and **sufficient storage** for temporary image handling.
3. The AI model will have been **pre-trained and optimized** for the targeted plant species before deployment.
4. The project will use **publicly available datasets** or **proprietary data collected under consent** for model training.
5. Weather integration (if implemented) assumes availability of **reliable third-party weather APIs**.

Dependencies

1. The application depends on **Google ML Kit**, **TensorFlow Lite**, or equivalent libraries for AI inference.
2. The weather feature depends on integration with APIs such as **OpenWeatherMap** or **WeatherAPI**.
3. Project success depends on **accurate dataset labeling** and **model training quality**.

7. References

Ref. No.	Document Title	Date of Release/ Publication	Document Source
FYP2226G	Project Proposal	Oct 13, 2025	https://github.com/mtahamukhtar2-ui/FYP.git
[1]	PS Venkata Reddy, KS Nandini Prasad, and C. Puttamadappa. Farmer's Friend: Conversational AI Bot for Smart Agriculture. Journal of Positive School Psychology, 6(2):2541–2549	2022	https://journalppw.com/index.php/jpsp/article/view/1833

Ref. No.	Document Title	Date of Release/ Publication	Document Source
[2]	Vandana Nayak, NH Sowmya, et al. Agroxpert – Farmer Assistant. Global Transitions Proceedings, 2(2):506–512	2021	https://www.scribd.com/document/716551511/Agroxpert-Farmer-assistant-2021-Global-Transitions-Proceedings
[3]	Abbas Jafar, Nabila Bibi, Rizwan Ali Naqvi, Abolghasem Sadeghi-Niaraki, and Daesik Jeong. Revolutionising Agriculture with Artificial Intelligence: Plant Disease Detection Methods, Applications, and Their Limitations. Frontiers in Plant Science, 15:1356260	2024	https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2024.1356260/full
[4]	Tejal Yadav, Pooja Sable, and Dhananjay Kalbande. Smart Kisan: A Mobile App for Farmers' Assistance in Agricultural Activities. 2023 International Conference on Smart Systems for Applications in Electrical Sciences (ICSSES)	2023	IEEE Xplore (Available through conference proceedings)

8. Appendices

8.1 Dataset Description

The AI model for plant disease detection was trained on trusted, publicly available datasets to ensure accurate results across a wide variety of plant types.

Dataset Name	Source	Description	Format
Plant Village Dataset	Kaggle	Contains over 2,000 labeled images of healthy and diseased plant leaves from 9 plant species.	JPEG
AI Challenger Dataset	AI Challenger Platform	Provides agricultural images for machine vision applications, used for cross-validation.	PNG

Custom-Collected Dataset	Project Team	Locally collected images of regional crops for enhanced model accuracy and real-world testing.	JPEG