# Hamdard University
# Department of Computing
# Final Year Project


# Plant Disease Detection Using Mobile Application
**FYP-028/FL24**

## Software Requirements Specifications

## Submitted by

Syed Hasnain Imran
(BSAI/1653-2021)
Khadija Rashid Mughal
(BSSE/1814-2021)
Zainab Jamil
(BSAI/2179-2021)

## Supervisor
Sir Waqas Pasha

## 2025

**Faculty of Engineering Sciences and Technology**

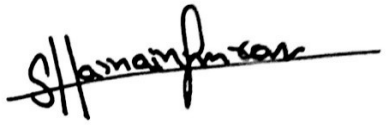
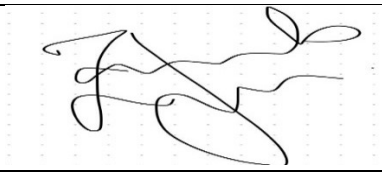
Hamdard Institute of Engineering and Technology

Hamdard University, Main Campus, Karachi, Pakistan

# Document Sign off Sheet

## Document Information

| | |
|---|---|
| **Project Title** | <Plant Disease Detection Using Mobile Application> |
| **Project Code** | <FYP-028/FL24> |
| **Document Name** | Software Requirements Specifications |
| **Document Version** | <1.0> |
| **Document Identifier** | FYP-028/FL24-SRS |
| **Document Status** | Final Draft |
| **Author(s)** | Syed Hasnain Imran<br>Khadija Rashid Mughal<br>Zainab Jamil |
| **Approver(s)** | Sir Waqas Pasha |
| **Issue Date** | 30-10-2024 |

| Name | Role | Signature | Date |
|---|---|---|---|
| Syed Hasnain Imran | Team Lead |  | 17-jan-2025 |
| Khadija Rashid Mughal | Team Member 2 |  | 17-jan-2025 |
| Zainab Jamil | Team Member 3 |  | 17-jan-2025 |
| Sir Waqas Pasha | Supervisor | | 17-jan-2025 |
| | Project Coordinator | | |

# Revision History

| Date | Version | Description | Author |
| --- | --- | --- | --- |
| 30-10-2024 | 0.1 | First Draft | Khadija Rashid |
| 26-11-2024 | 0.2 | Revised Draft | Zainab Jamil |
| 10-12-20241 | 1.0 | Final Draft | Khadija Rashid, Zainab Jamil |

# Definition of Terms, Acronyms, and Abbreviations

| Term | Description |
| --- | --- |
| SRS | Software Requirement Specification |
| AI | Artificial Intelligence |
| ANN | Artificial Neural Networks |
| DL | Deep Learning |
| CNN | Convolutional Neural Networks |
| CV | Computer Vision |
| UI | User Interface |
| MobileNetV2 | Mobile Net Version 2 |

# Table of Contents

Contents

# 1. Introduction

Detection of plant diseases is crucial for maintaining healthy plants and improving nursery productivity. This project aims to develop a mobile application specifically designed for nursery workers to efficiently detect plant diseases. By utilizing advancements in deep learning and image processing, the application will analyze captured or uploaded images of plant leaves to identify potential diseases. The application will offer a user-friendly interface, making it accessible even for individuals with minimal technical expertise. By focusing on reliable disease detection, this tool aims to streamline plant health assessment in nurseries and reduce manual effort.

In recent years, deep learning methods have greatly improved plant disease detection capabilities. One study highlights the effectiveness of convolutional neural networks (CNNs) in identifying plant diseases through image analysis. The study demonstrates how CNN models can accurately classify diseases by analyzing leaf images, facilitating early detection and timely intervention in agricultural practices. This approach surpasses traditional manual methods, as it can quickly process large volumes of plant images with enhanced accuracy and efficiency. By efficiently analyzing numerous images in a short period, CNN-based systems reduce the need for expert involvement and manual inspections, positioning them as a vital tool for modern agricultural practices.

## 1.1 Purpose of Document

The purpose of this Software Requirement Specifications (SRS) document is to provide detailed insights into the design considerations, architecture, and functionality of the Plant Disease Detection Using Mobile Application (Nursery Base)**.** This document serves as a blueprint for the development team, outlining the technical requirements and specifications necessary for the successful implementation of the application. The application leverages deep learning and computer vision techniques to accurately identify plant diseases and aims to assist users, such as nursery owners and farmers, in managing plant health effectively.

## 1.2 Motivation

The question of why develop a software solution that already exists and is deployed in the industry is valid and can be addressed by highlighting the unique aspects and advantages of this project. While various plant disease detection systems are available in the market, many of them may have limitations such as high cost, complex interfaces, or limited accessibility. This project aims to address these gaps and provide a more practical, user-friendly, and accessible solution for plant disease detection.

The primary motivation behind this project is to develop a mobile-based plant disease detection system that is cost-effective, easy to use, and accessible to a wider range of users. By leveraging the power of deep learning, particularly Convolutional Neural Networks (CNNs) such as MobileNetV2, this system aims to provide accurate disease detection while overcoming challenges faced by traditional systems.

## 1.3 Intended Audience

The intended audience for this project includes nursery workers, who will directly use the plant disease detection application for improved plant health management. It also target developers, academic supervisors involved in guiding and building the system and also researchers studying plant disease who can use the application. Additionally, this document serves as a guide for project team members, including designers, developers, and researchers, to understand the system's objectives, requirements, and deliverables.

# 2. Overall System Description

## 2.1 Project Charter

| | | | |
|---|---|---|---|
| **Project Name** | **Plant Disease Detection Using Mobile Application** | | |
| **Project Start Date** | 18-July-2024 | **Project End Date** | 15-june-2025 |
| **Project Members** | Syed Hasnain Imran<br>Khadija Rashid Mughal<br>Zainab Jamil | | |
| **Version** | 1.0 | | |
| **Project Overview** | Plants play an essential role in creating a healthy environment, whether in nurseries, gardens, or homes. However, plant care can be challenging for nursery workers and casual users alike. This project aims to develop a mobile application to detect plant diseases by analyzing images and also give tell you symptoms and recommendations. | | |
| **Project Objective** | 1. Develop a mobile application for plant disease detection.<br>2. Implement an AI model for accurate plant disease detection from images.<br>3. Ensure compatibility with common smartphones.<br>4. Make the app accessible to both nursery workers and home users. | | |
| **Scope** | 1. Detect plant diseases through uploaded or captured images.<br>2. To provide symptoms description for the detected disease.<br>3. To generate recommendations for managing or preventing the identifying plant disease.<br>4. Focus on a user-friendly interface for both technical and non-technical users. | | |
| **Stakeholders** | **Project Supervisor**: Sir Waqas Pasha | | |
| | **Development Team:** Front-End Developer(s), Back-End Developer(s), Database Administrator, QA Tester(s) | | |
| | **Secondary:** Nursery owners, home gardeners. | | |
| **Deliverables** | 1. A mobile application for plant disease detection.<br>2. A detailed SRS and SDS.<br>3. A presentation demonstrating the system to the evaluation panel. | | |
| **Milestone** | **Project Planning:** 18 July – 15 Aug 2024<br>Deliverables: Project plan, initial requirements analysis.<br>**Project Analysis & Design:** 16 Aug – 25 Sep 2024<br>Deliverables: Project budget, SRS/SDS, WBS.<br>**Data Collection: 26 Sep – 10 Oct 2024**<br>Collect plant disease images and corresponding data (types of plants, disease details, etc.).<br>**Front-End Development:** 11 Oct – 15 Dec 2024 | | |

| | | |
| --- | --- | --- |
| | Deliverables: Interface design and front-end application.<br>**Back-End Development:** 16 Dec – 15 Mar 2025<br>Deliverables: Database integration, model deployment.<br>**System Testing & Refinement 16 Mar – 30 Apr 2025**<br>Deliverables: Functional and performance testing.<br>**Final Deployment Preparation:** 01 May – 30 May 2025<br>Deliverables: Executable software, FYP 2 report, proposed system.<br>**Final Submission:** 31 May – 15 June 2025<br>Deliverables: Complete software package, code stored on CD/USB. | |
| **Assumptions** | 1. Users will have basic knowledge of using mobile apps.<br>2. Internet access is available for app updates and database synchronization.<br>3. Stakeholder participation and feedback.<br>4. App performance depends on the quality of users mobile devices. | |
| **Constraints** | Time | 11 Months |
| | Budget | 76,500 Rs |
| | Quality | Prioritize time and budget over quality. |
| **Resources** | Hardware Resources | High-performance servers for web hosting and database management.<br>Flash drives and external storage for data backup. |
| | Human Resources | Development team with expertise<br>in front-end, back-end, database and testing. |
| | Software Tools | Python, Flutter (mobile app), Firebase (database), OpenCV (image processing).<br>Android Studio, Jupiter Notebook, VS code, Anaconda, Emulator |

## 2.2 Project Background

Plant diseases significantly affect the productivity of nurseries, leading to economic losses and the wastage of valuable resources. Traditional disease detection methods such as manual observation often demands expert knowledge that nursery workers may lack. However, these methods can be time-consuming, labor-intensive, and may not be readily accessible to all nursery workers, particularly those in remote locations.

Artificial Neural Networks (ANNs) are effective tools for managing complicated data, drawing inspiration from the analytical and processing powers of the human brain. Because they can

recognize spatial patterns and hierarchies in visual data, Convolutional Neural Networks (CNNs) are the best of these for image-related applications. For this project, the MobileNetV2 architecture a lightweight and mobile-optimized version of CNN is employed to ensure accurate and efficient plant disease detection.

Recent advancements in computer vision and deep learning, specifically the emergence of Convolutional Neural Networks (CNNs), have demonstrated remarkable potential in image-based disease detection. This project leverages these challenges by developing a mobile application that uses AI-powered image processing techniques to detect plant diseases.

## 2.3 Problem Statement

The detection of plant diseases plays a vital role in modern agriculture, yet traditional approaches are often slow and rely heavily on expert knowledge. Recent progress in deep learning, particularly with Convolutional Neural Networks (CNNs), has proven to significantly enhance the precision and speed of plant disease detection. A comparative evaluation of various detection methods indicates that while traditional techniques face challenges with scalability and real-time processing, deep learning models present a promising solution to these issues.

This project aims to address these challenges by developing a mobile application that utilizes deep learning techniques, specifically CNNs to automate plant disease detection. By providing nursery workers with a rapid and accurate diagnostic tool, this application will enhance disease management practices, minimize crop losses, and contribute to a more sustainable and resilient agricultural system.

## 2.4 Project Scope

- Focuses on detecting and classifying plant diseases from images captured through mobile devices.
- Utilizes Convolutional Neural Networks (CNNs), specifically optimized models like MobileNetV2.
- The project will integrate lightweight models like MobileNetV2 for efficient and accurate processing on mobile devices.
- The app will feature an easy-to-use interface for nursery owners, Includes features like image upload, real-time detection, and logout functionality.

## 2.5 Not In Scope

- Doesn’t include external farming tools or equipment like drones and sensors.
- No advanced editing features for uploaded images like manual adjustments or filters.

## 2.6 Project Objectives

The primary objectives of plant disease detection is to identify the disease accurately and quickly by utilizing deep learning Convolutional Neural Networks (CNNs) model. This is important to facilitate early disease detection to minimize crop losses and promote sustainable agriculture practices.

## 2.7 Stakeholders & Affected Groups

- Nursery workers
- Home Gardeners
- Software Development Team
- Supervisor

## 2.8 Operating Environment

The system will run on both Android and iOS devices, leveraging Flutter's cross-platform capabilities, using optimized frameworks for efficient deep learning on mobile devices. It will be designed for smartphones with sufficient processing power for real-time detection, offering offline functionality, with an internet connection required for cloud updates.

## 2.9 System Constraints

- Mobile devices must have a functioning camera.
- Detection accuracy relies heavily on image quality, including lighting and focus.
- An internet connection is required for updates, though offline functionality is supported for image uploads.

## 2.10 Assumptions & Dependencies

System relies on the availability and utilization of the PlantVillage dataset as a fundamental resource for the development and evaluation of the deep learning model. This open-access repository of plant health images offers a vast and diverse range of categorized images essential for effectively training the CNN model. The system assumes that users will utilize mobile

devices to capture and upload images of plant leaves, with the real-time detection and disease classification process designed to operate smoothly on these mobile platforms.

# 3. External Interface Requirements

## 3.1 Hardware Interfaces

- Mobile phone with a functionally camera for capturing images.
- No other external hardware interface needed.

## 3.2 Software Interfaces

- **OpenCV:** OpenCV is utilized for various image processing tasks such as resizing, cropping, and improving the quality of plant leaf images before they are fed into the deep learning model for classification.
- **TensorFlow:** TensorFlow is employed to create and implement deep learning models, with TensorFlow Lite being used to deploy the trained model on mobile devices for real-time plant disease detection. It ensures efficient model inference on mobile platforms with constrained resources.
- **PlantVillage Dataset:** The PlantVillage dataset offers images of plant diseases, which are used for training, validating, and testing the deep learning model. It contains labeled images of plant leaves, categorized into different disease types and healthy conditions.
- **Firebase:** Firebase is utilized for storing images, user information, and disease diagnosis outcomes. It also offers authentication services for user management and ensures smooth synchronization between the app and the cloud backend.
- **External Libraries (NumPy and Pandas):** Provide essential functionality for data preprocessing and analysis.

## 3.3 Communications Interfaces

- The internet connection plays an important role which is updating the data in real time.

# 4. System Functions / Functional Requirements

## 4.1 System Functions

- **Signup Form:** For new users to create account
- **Login Form:** For existing users to login to the system
- **Select Camera:** Users can capture images using the camera.
- **Select Gallery:** Users can upload images of plants for disease detection.
- **Detect Disease:** AI processes the image and identifies the disease.
- **Logout Form:** To Logout from app.

### Function Categories

| Ref # | Functions | Category | Attribute | Details & Boundary Constraints |
| --- | --- | --- | --- | --- |
| R1.1 | Upload Image | Evident | System Response time | Uploads plant images within 5 seconds on a stable internet connection. |
| R1.2 | Capture Image | Evident | System Response time | Camera opens instantly to capture clear plant images. |
| R1.3 | Detect Disease | Evident | Accuracy | Identifies diseases using the CNN model (MobileNetV2). |
| R1.4 | Logout | Evident | Data Security | Ensure secure user logout |
| R1.5 | Download Disease Report | Frill | Future Feature | Enables users to generate and download detailed disease reports (planned for FYP-2). |

## 4.2 Use Cases

### 4.2.1 List of Actors

1. **User:** The primary user who interacts with the mobile app to upload images and receive disease detection results.
2. **System/Server:** Handles backend processing like disease detection and fetching disease details.

### 4.2.2 List of Use Cases

1. **Upload Image:** The user uploads a plant image for analysis.
2. **Capture Image:** The user captures an image using the device camera.
3. **Preprocess image:** The system preprocesses the uploaded image to ensure it is ready for disease detection.
4. **Train model:** The CNN model is trained using labeled plant disease images to enable accurate disease classification.
5. **Fetch Disease Details:** The system retrieves and displays detailed information about the detected disease
6. **Detect Disease:** The system processes the image and detects potential plant diseases.
7. **View Disease Details:** Displays detected disease information.
8. **Logout:** Allows the user to securely log out of the application.

### 4.2.3 Use Case Diagram


### 4.2.4 Description of Use Cases

| Use Case Name | Detect Disease |
|---|---|
| Actors | Nursery Worker, system/server |
| Purpose | To identify plant diseases based on uploaded or captured images. |
| Preconditions | The user must be logged into the application and have an image ready for upload or capture. |
| Postconditions | The system displays the disease details. |
| Normal Flow | 1. User logs into the app.2. Uploads or captures an image.3. Clicks the "Detect" button.4. System analyzes the image and displays results. |

# 5. Non - Functional Requirements

## 5.1 Performance Requirements

The application should analyze uploaded images and generate results within 5 seconds to provide a smooth user experience. This fast processing time is crucial for nursery workers who may require immediate information to act promptly. Achieving this will require optimized algorithms and efficient use of hardware or cloud resources.

## 5.2 Safety Requirements

User data must be securely stored and managed to prevent loss, unauthorized access, or corruption. This includes implementing encryption, access controls, and regular backups. Ensuring data safety is essential for building user trust and adhering to data protection laws.

## 5.3 Security Requirements

If your system involves any data collection or online components, security features such as data encryption, user authentication, and secure data storage need to be mentioned. This might also involve protecting sensitive user data if applicable. Login credentials provided by users should be encrypted to safeguard against cyberattacks and unauthorized access. Proper security measures are necessary to protect sensitive user data and minimize system vulnerabilities.

## 5.4 Reliability Requirements

The system must ensure reliability, making the application consistently available for users. Achieving high reliability requires regular maintenance, backup systems, and robust testing to avoid unexpected outages. This is critical for users who rely on the app for timely results.

## 5.5 Usability Requirements

The app’s interface must be simple and user-friendly, catering to non-technical users like nursery workers. Responsive design to ensure accessibility on various devices. A well-designed interface ensures the app is accessible and encourages consistent use.

## 5.6 Maintainability Requirements

The system must be designed for easy maintenance to ensure long-term functionality and adaptability. This includes writing clean, modular, and well-documented code to facilitate updates or bug fixes. Additionally, a robust error-logging mechanism should be in place to quickly identify and resolve issues, reducing downtime and maintenance efforts.

## 5.7 User Documentation

A detailed user manual should be created to describe the technical and functional details of the project. This manual must outline the system's architecture, core features, development process, and the technologies used. It should also include system requirements, and maintenance guidelines to provide a clear understanding of the project.

# 6. References

**[1] Deep Learning for Plant Disease Detection,** *International Journal of Mathematics, Statistics, and Computer Science, 2024*

- **Authors**: M. M. Khalid and O. Karan
- **Summary**: Explores the application of CNNs for plant disease detection, supporting the use of MobileNetV2 for efficient disease identification.

**[2] Using Deep Learning for Image-Based Plant Disease Detection,** *Frontiers in Plant Science, 2016*

- **Authors**: S. P. Mohanty, D. P. Hughes, and M. Salathé
- **Summary**: Highlights deep learning methods for image-based plant disease detection, directly supporting the classification methods.

**[3] A Mobile-Based System for Detecting Plant Leaf Diseases Using Deep Learning,** *Journal of Agricultural Technology and Innovation, 2023*

- **Authors**: A. A. Ahmed and G. H. R. Reddy
- **Summary**: Discusses mobile-optimized implementations of CNNs for real-time plant disease detection on smartphones.

**[4] Deep Learning and Content-Based Filtering Techniques for Improving Plant Disease Identification,** *Computers and Electronics in Agriculture, 2020*

- **Authors**: Isinkaye, F. O., Olusanya, M. O., & Singh, P. K.
- **Summary**: Reviews techniques combining deep learning and content-based filtering for plant disease detection and treatment recommendations.

**[5] An Open Access Repository of Images on Plant Health to Enable the Development of Mobile Disease Diagnostics**

- **Authors:** David P. Hughes, Marcel Salathé

- **Summary:** This paper presents the PlantVillage dataset, an open-access resource of plant health images, aimed at advancing plant disease detection research and enabling mobile diagnostic tool development.

**[6] Plant disease detection and classifcation techniques,** *A comparative study of the performances, International Journal of Computer Applications*, vol. 182, no. 15, pp. 1-9, 2021.

- **Authors:** Wubetu Barud Demilie
- **Summary:** This paper provides a comprehensive comparative analysis of various techniques for plant disease detection and classification, evaluating the performance of each method and highlighting the strengths and weaknesses of traditional and modern approaches.

**[7]** C. Adams, "Role of Use Case Diagram in Software Development," 26 May 2022. [Online]. Available: https://www.researchgate.net/publication/322991847_role_of_use_case_diagram_in_software_development