

# **Software Requirements Specifications**

**Project Title: Potato Leaf Disease Classification App**

**Project Code:**

**Internal Advisor:** Prof. Dr. Saad Razzaq

**External Advisor:**

**Project Manager:** Prof. Dr. Muhammad Ilyas

**Project Team:** Team Leader:  
Bilal Ahmed (BSCS51F21S082)  
Members:  
Syed Turab Haider Naqvi (BSCS51F21S065)  
Muhammad Bilal (BSCS51F21S081)

**Submission Date:** 26-11-2024

---

**Project Manager's Signature**

## Document Information

Category	Information
Customer	UOS-CS
Project	Potato Leaf Disease Classification App
Document	
Document Version	
Identifier	
Status	
Author(s)	Bilal Ahmed, Turab Haider, Muhammad Bilal
Approver(s)	Prof. Dr. Muhammad Ilyas
Issue Date	Sept. 10, 2024
Document Location	
Distribution	1. Advisor (Prof. Dr. Saad Razzaq) 2. Project Manager (Prof. Dr. Muhammad Ilyas) 3. Project Office

## Definition of Terms, Acronyms and Abbreviations

*This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly.*

Term	Description
LLM	Large Language Model
RAG	Retrieval Augmentation Generation
CNN	Convolutional Neural Network
GCP	Google Cloud Platform
UI	User Interface

---

## Table of Contents

---

1. INTRODUCTION .....	4
1.1 <i>Purpose of Document</i> .....	4
1.2 <i>Project Overview</i> .....	4
1.3 <i>Scope</i> .....	4
2. OVERALL SYSTEM DESCRIPTION .....	4
2.1 <i>User characteristics</i> .....	4
2.2 <i>Operating environment</i> .....	4
2.3 <i>System constraints</i> .....	4
3. EXTERNAL INTERFACE REQUIREMENTS .....	5
3.1 <i>Hardware Interfaces</i> .....	5
3.2 <i>Software Interfaces</i> .....	5
3.3 <i>Communications Interfaces</i> .....	5
4. FUNCTIONAL REQUIREMENTS .....	5
5. NON-FUNCTIONAL REQUIREMENTS .....	6
5.1 <i>Performance Requirements</i> .....	6
5.2 <i>Safety Requirements</i> .....	6
5.3 <i>Security Requirements</i> .....	6
5.4 <i>User Documentation</i> .....	6
6. ASSUMPTIONS AND DEPENDENCIES.....	6
7. REFERENCES .....	7
8. APPENDICES .....	8

# 1. Introduction

## 1.1 Purpose of Document

*This document defines the functional, non-functional, and technical requirements for the Potato Leaf Disease Classification App. The app leverages deep learning and Retrieval-Augmented Generation (RAG) to detect potato plant diseases, provide detailed information, and assist users in disease prevention and treatment. The document serves as a foundation for development, implementation, and testing.*

## 1.2 Project Overview

*The Potato Leaf Disease Classification App is designed for farmers, agricultural experts, and researchers. The core functionalities include:*

- *Capturing or uploading images of potato plants.*
- *Classifying diseases using a CNN-based model.*
- *Providing disease-specific information via a knowledge base integrated with a Retrieval-Augmented Generation (RAG) model.*
- *Offering treatment suggestions and preventive measures.*
- *The application aims to enhance agricultural productivity by reducing crop losses caused by plant diseases.*

## 1.3 Scope

1. *Platform: Android, iOS, and web.*
2. *Audience: Farmers, agricultural researchers, and agricultural extension workers.*
3. *Core Features: Image analysis, disease classification, knowledge retrieval, and actionable insights.*
4. *Goals: To provide a user-friendly and scalable solution for managing potato crop health.*

# 2. Overall System Description

*Describe the environment, in which the system will be developed and used, the anticipated users of the system and the known constraints, assumptions and dependencies.*

## 2.1 User characteristics

1. *Primary Users: Farmers with basic smartphone literacy.*
2. *Secondary Users: Agricultural researchers and extension workers with moderate technical knowledge.*
3. *Skill Level: Varying levels of technical proficiency, ranging from minimal to moderate.*

## 2.2 Operating environment

1. *Mobile Platforms: Android 8.0+ and iOS 11.0+.*
2. *Web Platforms: Chrome, Firefox, and Edge (latest versions).*
3. *Backend Services: Hosted on Google Cloud Platform (GCP) with API integrations for model inference and data retrieval.*

## 2.3 System constraints

- *Limited access to high-speed internet in rural areas.*
- *Hardware limitations of user devices (e.g., low memory, low processing power).*

- Scalability to accommodate up to 10,000 users in the future..

### 3. External Interface Requirements

*This section is intended to specify any requirements that ensure that the new system will connect properly to external components. Place a context diagram showing the external interfaces at a high level of abstraction.*

#### 3.1 Hardware Interfaces

1. *User Devices: Smartphone or desktop/laptop with a camera.*
2. *Server Requirements: Google Cloud virtual machines for model deployment, Vector database for knowledge base indexing.*

#### 3.2 Software Interfaces

1. *Frontend Frameworks: React Native (mobile), React.js (web).*
2. *Backend Services: Tensor-Flow / Keras models for disease detection, Fast-API for API development.*
3. *Databases: Vector database for RAG model retrieval, Firebase for user authentication and data storage.*

#### 3.3 Communications Interfaces

1. *HTTPS protocol for secure communication.*
2. *REST APIs for interaction between the frontend and backend.*
3. *Web-Sockets for real-time notifications and updates*

## 4. Functional Requirements

#### 1. Real-Time Image Capture and Processing

- Users can capture images in real time using the app and process them immediately to detect potato diseases.  
(Proposal, Section 4: "Features for capturing images of potato plants.")

#### 2. Disease Detection and Classification

- A CNN-based model identifies whether the plant is healthy or diseased and provides specific disease classifications with confidence scores.  
(Proposal, Section 3: "Develop a CNN architecture tailored for image classification.")

#### 3. Knowledge Base Integration for RAG

- The app retrieves disease-specific information from a pre-indexed knowledge base using the RAG model.
- This ensures dynamic, context-aware insights tailored to user queries.  
(Proposal, Section 3: "Implement a retrieval system to search the knowledge base.")

#### 4. Disease Information Summarization via LLM

- A pre-trained LLM generates detailed disease summaries based on the information retrieved by the RAG model.  
(Proposal, Section 3: "Use a pre-trained LLM to generate a response.")

#### 5. Comprehensive Disease Information

- Provide detailed insights about detected diseases, including symptoms, treatment options, and preventive measures.  
(Proposal, Section 4: "Provide real-time disease predictions and integrate RAG to offer comprehensive information.")

## 6. Preprocessing of Data for Model Training

- Perform data cleaning, augmentation, and preprocessing of potato plant images to ensure the CNN model is trained with high-quality data.  
(Proposal, Section 3: "Perform data cleaning, augmentation, and preprocessing.")

## 7. Admin Management of Knowledge Base

- Admin users can manage the knowledge base by adding, updating, or removing disease-related entries.  
(Proposal, Section 3: "Gather and preprocess a diverse dataset.")

## 8. Cloud Deployment on GCP

- Host backend services, including the CNN model, RAG system, and knowledge base, on Google Cloud Platform (GCP) to ensure scalability, reliability, and secure API integrations.  
(Proposal, Section 4: "Deploy the application on GCP.")

## 9. Testing and Validation

- Conduct rigorous testing to ensure the CNN and RAG models perform accurately and meet user expectations.
- Perform system validation during deployment to avoid errors in disease detection and information retrieval.  
(Proposal, Section 4: "Conduct thorough testing.")

## 10. Disease Severity Analysis

- Analyze the severity of the detected disease (e.g., mild, moderate, or critical) and provide severity-based recommendations.

## 11. Notifications and Alerts

- Send notifications about farming tips, disease outbreaks, and prevention measures.  
(Proposal, Section 4: "Regular farming tips and disease prevention alerts.")

# 5. Non-functional Requirements

## 5.1 Performance Requirements

- Disease classification and retrieval must be completed within 5 seconds.
- The system must maintain a throughput of 1,000 concurrent users, scalable to 10,000 users.

## 5.2 Safety Requirements

- System failures should not compromise stored data or ongoing operations.
- Provide fallback mechanisms in case of backend service unavailability.

## 5.3 Security Requirements

- User data is encrypted using AES-256 standards.
- Secure login via OAuth2 with multi-factor authentication..

## 5.4 User Documentation

1. Provide a comprehensive user manual, covering: App navigation.
2. Steps for uploading images and interpreting results. FAQs and troubleshooting.

# 6. Assumptions and Dependencies

**Assumptions:**

- Users have access to a device with a functioning camera and internet connection.
- Cloud services (GCP) are available for hosting and computation.

**Dependencies:**

- Availability of an annotated dataset for training the CNN model.
- Vectorized knowledge base for RAG integration.

**7. References**

- Tensor-Flow Documentation - <https://www.tensorflow.org>
- Google Cloud Platform Documentation - <https://cloud.google.com>
- React Native Official Guide - <https://reactnative.dev>

Ref. No.	Document Title	Date of Release/ Publication	Document Source
[1]	A novel framework for potato leaf disease detection using an efficient deep learning model	19 Apr 2022	<a href="https://doi.org/10.1080/10807039.2022.2064814">https://doi.org/10.1080/10807039.2022.2064814</a>
[2]	Early Detection of Potato Disease Using an Enhanced Convolutional Neural Network-Long Short-Term Memory Deep Learning Model	20 June 2024	<a href="https://link.springer.com/article/10.1007/s11540-024-09760-x">https://link.springer.com/article/10.1007/s11540-024-09760-x</a>
[3]	Potato Leaf Disease Detection using Deep Learning	11 November 2022	<a href="https://www.researchgate.net/publication/365344910">https://www.researchgate.net/publication/365344910</a>
[4]	Artificial Intelligence in Potato Leaf Disease Classification: A Deep Learning Approach	6 January 2021	<a href="https://www.researchgate.net/publication/347877211">https://www.researchgate.net/publication/347877211</a>
	Potato Leaf Disease Classification Using Optimized Machine Learning Models and Feature Selection Techniques	25 June 2024	<a href="https://link.springer.com/article/10.1007/s11540-024-09763-8">https://link.springer.com/article/10.1007/s11540-024-09763-8</a>
[5]	Potato Plant Leaves Disease Detection and Classification using Machine Learning Methodologies	8 June 2021	<a href="https://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012121">https://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012121</a>
[6]	CLASSIFICATION OF POTATO LEAF DISEASES USING CONVOLUTIONAL NEURAL NETWORK	2 September 2023	<a href="https://www.researchgate.net/publication/374861487_CLASSIFICATION_OF_POTATO_LEAF_DISEASES_USING_CONVOLUTIONAL_NEURAL_NETWORK">https://www.researchgate.net/publication/374861487_CLASSIFICATION_OF_POTATO_LEAF_DISEASES_USING_CONVOLUTIONAL_NEURAL_NETWORK</a>
[7]	Potato Leaf Disease Detection and Classification using CNN	5 December 2022	<a href="https://www.proquest.com/openview/14291bc74fe8758a972f2c1c131fb595/1?pq-origsite=gscholar&amp;cbl=2035897">https://www.proquest.com/openview/14291bc74fe8758a972f2c1c131fb595/1?pq-origsite=gscholar&amp;cbl=2035897</a>

## 8. Appendices

### **Appendix A: Glossary**

*CNN: Convolutional Neural Network, used for image classification.*

*RAG: Retrieval-Augmented Generation, integrates retrieval and summarization for knowledge-based responses.*

*LLM: Large Language Model, used for generating disease summaries.*

*GCP: Google Cloud Platform, hosting and backend services.*

### **Appendix B: Workflow Diagram**

*Step 1: User uploads an image.*

*Step 2: Backend validates and processes the image.*

*Step 3: CNN model classifies the disease.*

*Step 4: RAG system fetches disease details.*

*Step 5: Results are displayed on the user interface.*