



TED UNIVERSITY

CMPE 491 – Senior Project I Analysis Report

Team Members

Tolga AKYOL

Yiğit AYDIN

Supervisor

Dr. Evren Coşkun

Jury Members

Prof. Dr. Tansel Dökeroğlu, Dr. Deniz Cantürk

Document Version: v1.0, Date: 28/11/2025

SmartLeaf

1. Introduction

SmartLeaf is an image-based plant analysis system developed to support agricultural users in identifying plant health conditions from leaf photographs. The system applies computer vision and deep learning techniques to detect visible diseases, estimate overall leaf health, and provide indications related to potential nutrient deficiencies. The purpose of this analysis report is to examine the system's requirements, constraints, actors, interactions, and internal structure in detail. The document formalizes the specification of the system and clarifies how SmartLeaf will operate before the design and implementation phases.

The analysis phase focuses on understanding the problem domain, refining the system requirements defined in earlier documents, describing the system boundaries, and identifying all actors and possible interactions. This report also introduces the use case model, class model, and behavioral models that illustrate how the system will function. These models form the basis for later stages such as system design, architecture development, and implementation.

2. Current System

There is currently no automated system used by the stakeholders for plant health analysis. Farmers and agricultural workers primarily rely on manual inspection to identify diseases or deficiencies on plant leaves. This process depends heavily on personal experience, visual judgment, and access to expert support. Manual inspection can be slow, inconsistent, and prone to error, especially in large-scale agricultural environments.

Existing digital solutions available to the public are limited in scope, often supporting only a small set of plant species or disease types. These tools typically lack features such as health scoring, nutrient-deficiency indication, and structured reporting. In many cases, the applications are commercial and not accessible to small or independent farmers. As a result, users face challenges in obtaining timely and accurate evaluations of plant conditions.

SmartLeaf is designed to replace this manual, inconsistent approach with an automated, standardized analysis system that works through a simple web interface. Because the system is being developed from scratch and no legacy or previous software exists, this document focuses entirely on defining the proposed system and its behavior.

3. Proposed System

3.1 Overview

SmartLeaf is a web-based application designed to analyze plant leaf images using computer vision and deep learning. The goal of the system is to support farmers and agricultural users by detecting visible diseases, estimating overall leaf health, and providing possible nutrient deficiency indicators. The system receives an image uploaded by a user, applies pre-processing and segmentation, classifies the leaf condition, and presents the results through a structured interface.

The system will be designed as a modular architecture, separating the front-end user interface, back-end processing components, and machine learning models. This modularity allows future extensions such as growth tracking, severity estimation, or multi-crop support.

3.2 Functional Requirements

Image Input

The system shall allow users to upload images of plant leaves through a web interface. Supported formats will include JPEG and PNG.

Pre-processing

The system shall perform pre-processing steps such as resizing, noise reduction, normalization, and leaf segmentation to prepare the image for analysis.

Disease Classification

The system shall classify the leaf image into predefined disease categories (e.g., healthy, early blight, late blight) and provide a confidence value.

Health Scoring

The system shall compute a health score between 0 and 100 using color and texture-based metrics extracted from the input image.

Nutrient Deficiency Indicator

The system shall identify potential nutrient deficiencies (e.g., nitrogen, potassium, phosphorus deficiency) based on color features observed on the leaf surface.

Result Output

The system shall display a structured results page showing disease prediction, confidence score, health score, color analysis, and nutrient deficiency indication.

Feedback and Error Handling

The system shall inform users when the uploaded image is invalid, unclear, or not analyzable.

Report Generation

The system shall allow users to download the analysis result in PDF format.

Data Handling

The system shall delete uploaded images after analysis unless explicitly saved by the user. No sensitive data will be stored without permission.

Future Extensions

The modular structure of the system shall allow additional features such as growth monitoring and multi-plant support to be integrated later.

3.3 Nonfunctional Requirements

Performance

Image analysis operations shall complete within approximately five seconds for standard image sizes under one megabyte.

Accuracy

The classification model shall achieve at least 80% accuracy on validation datasets, and repeated analyses of identical images shall produce consistent results.

Usability

The interface shall be simple to navigate and require no more than three steps from image upload to result display. It shall support major browsers and mobile devices.

Scalability

The system shall support multiple concurrent users, with the back-end capable of scaling independently of the front-end.

Security

All communication between user and server shall use HTTPS. No personal information shall be stored without explicit consent.

Maintainability

The codebase shall follow modular design principles and be documented to support future extensions.

Reliability

The system shall remain operational with minimal downtime and include tests validating critical operations such as image upload and inference.

Portability

The application shall function on both desktop and mobile devices, and model files shall be deployable on lightweight environments such as TensorFlow Lite.

3.4 Pseudo Requirements

Pseudo requirements describe environmental or project-specific constraints that are not part of the core system behavior.

- The system must be accessible through a standard web browser.
- Development tools will include Python (for model training), JavaScript/React (for front-end), and a cloud hosting platform.
- Only publicly available datasets (e.g., PlantVillage) will be used during development.
- All machine learning models must be trained using resources available to the student team.
- The system must comply with ethical and departmental requirements described in earlier documents.
- The project must fit within the timeline of the CMPE491 course.

3.5 System Models

3.5.1 Scenarios

Scenario 1: Basic Leaf Analysis

1. User opens the SmartLeaf website.
2. User selects a plant leaf image to upload.
3. System pre-processes the image and performs analysis.
4. System displays disease result, confidence, health score, and deficiency indications.
5. User optionally downloads PDF report.

Scenario 2: Invalid Image

1. User uploads an unclear or non-leaf image.
2. System detects insufficient analysis quality.
3. System displays an error message requesting a clearer image.

Scenario 3: Low Confidence Result

1. System detects low classification confidence.
2. System notifies user that analysis may be unreliable.
3. User is encouraged to upload another image.

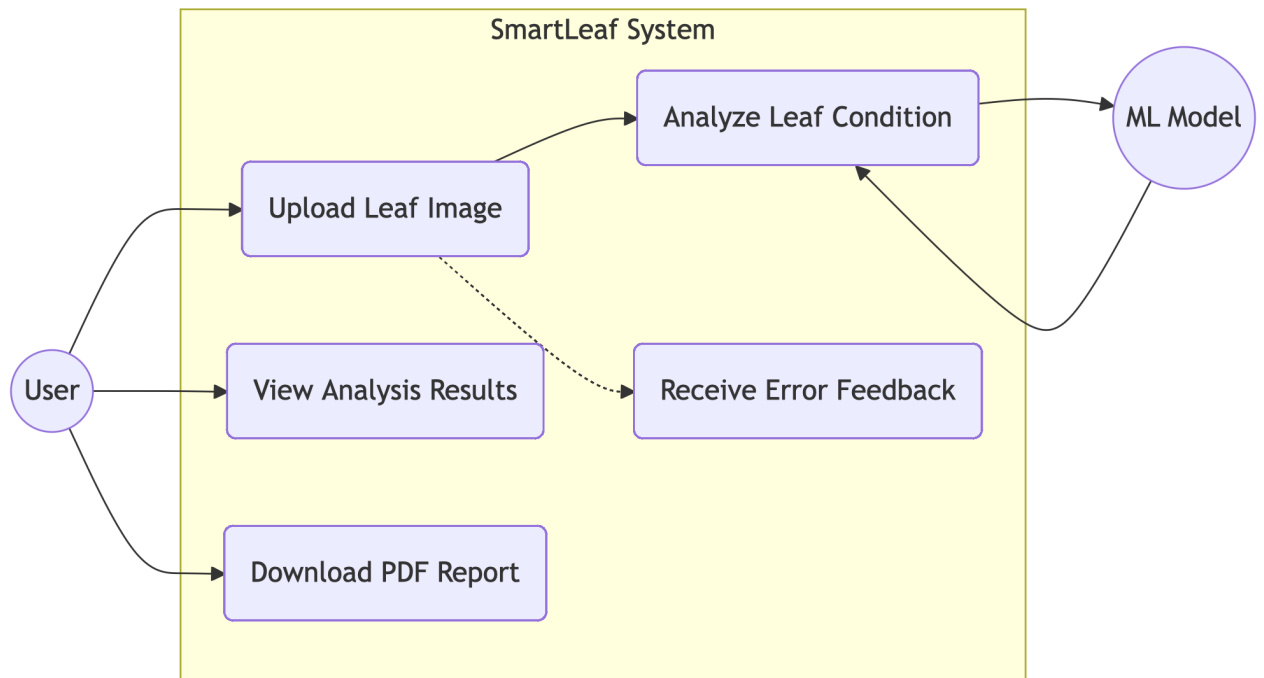
3.5.2 Use Case Model

Actors

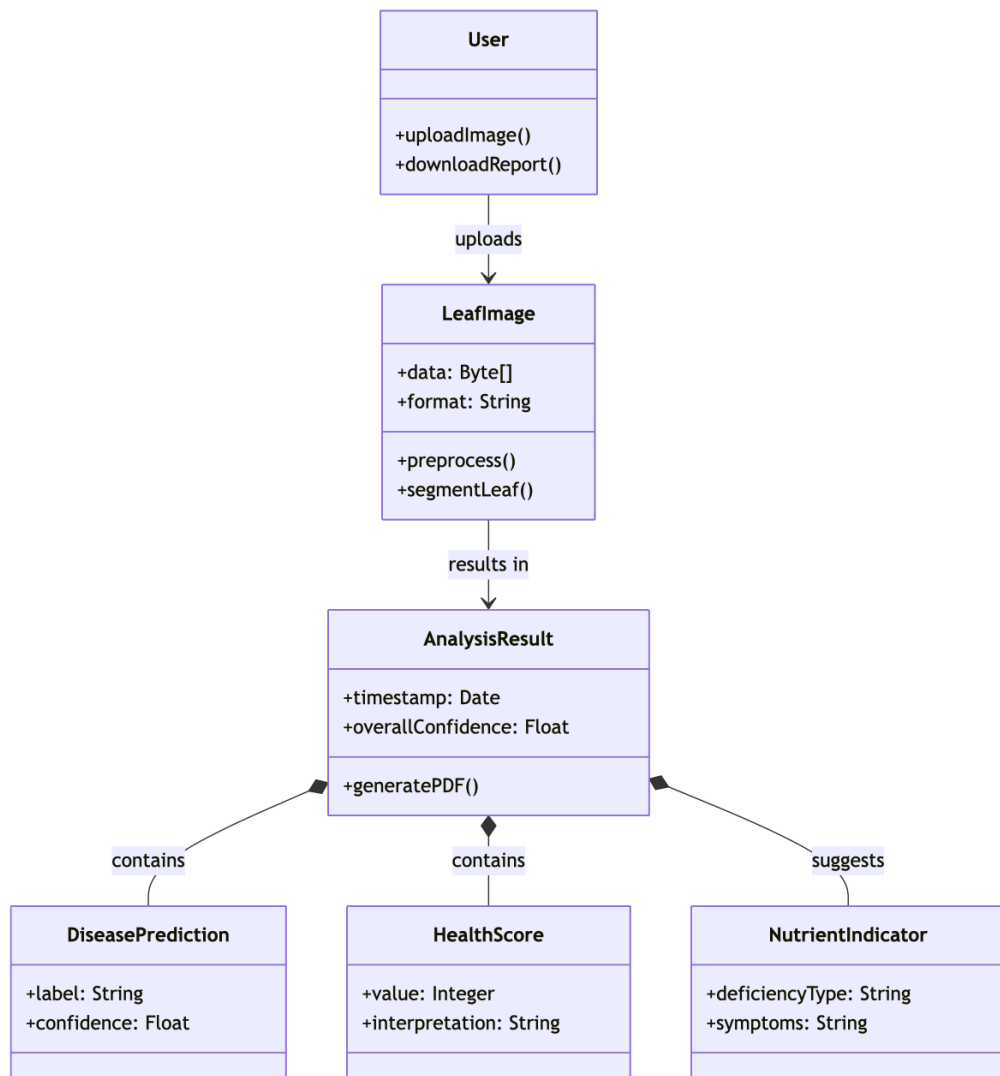
- User: A farmer, agronomist, or researcher
- System: SmartLeaf application
- ML Model: Classification and scoring component

Main Use Cases

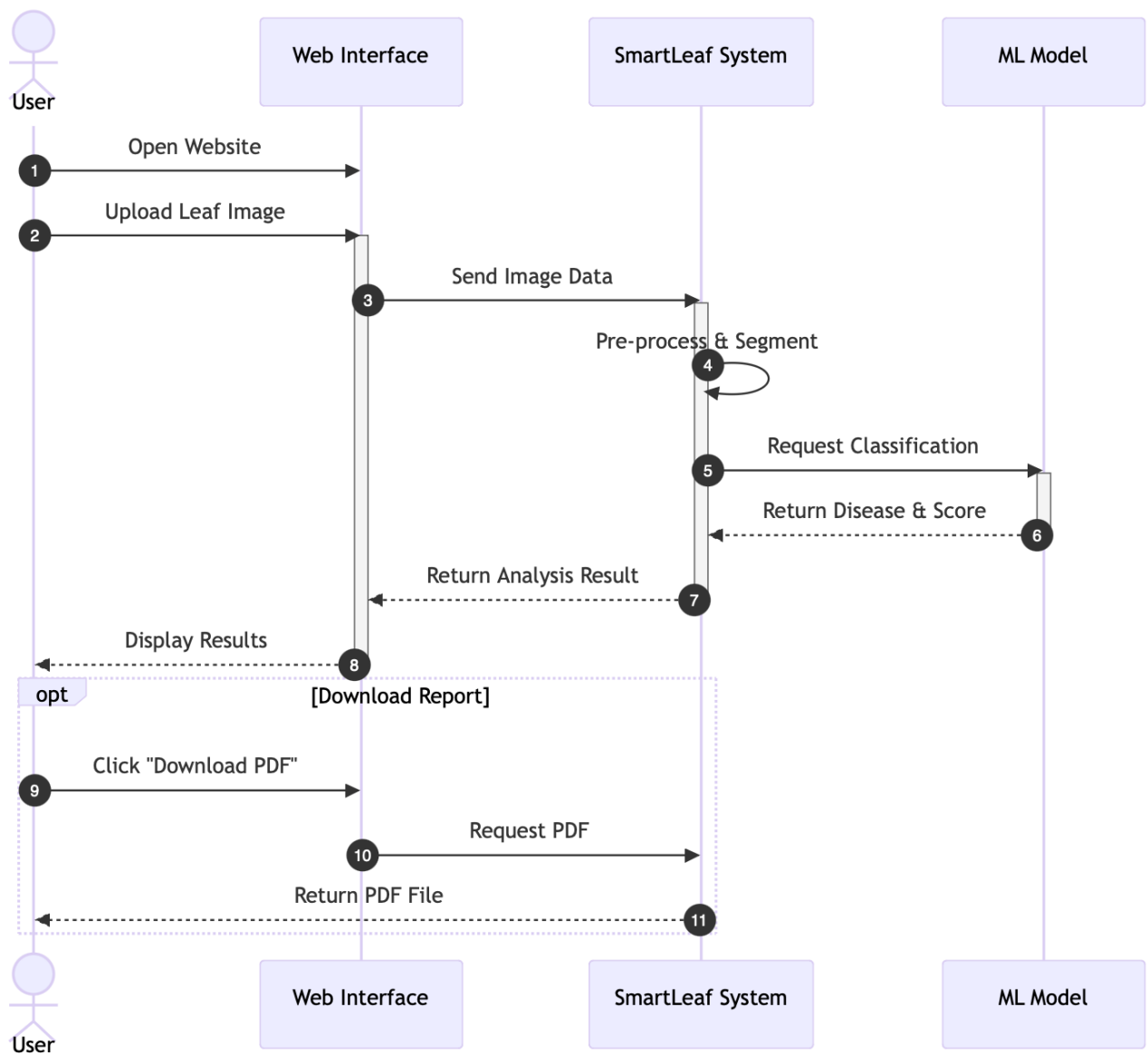
- Upload Image
- Analyze Leaf
- View Results
- Download Report
- Receive Error Feedback



3.5.3 Object and Class Model



3.5.4 Dynamic Model (Sequence Diagram)



3.5.5 User Interface – Navigational Paths & Mockups

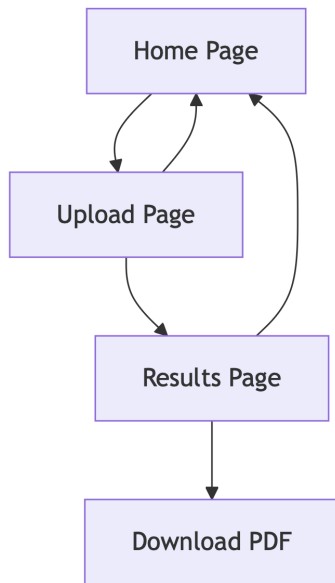


Figure 1: Home Page

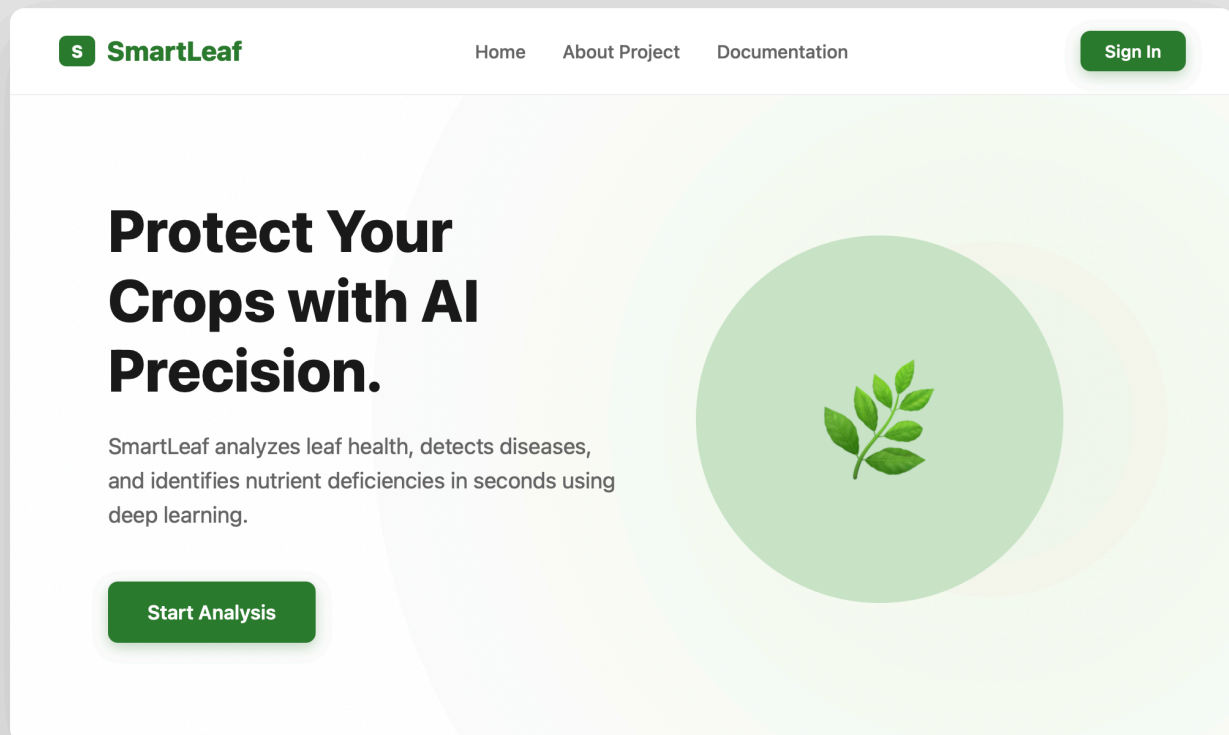


Figure 2: Upload Interface

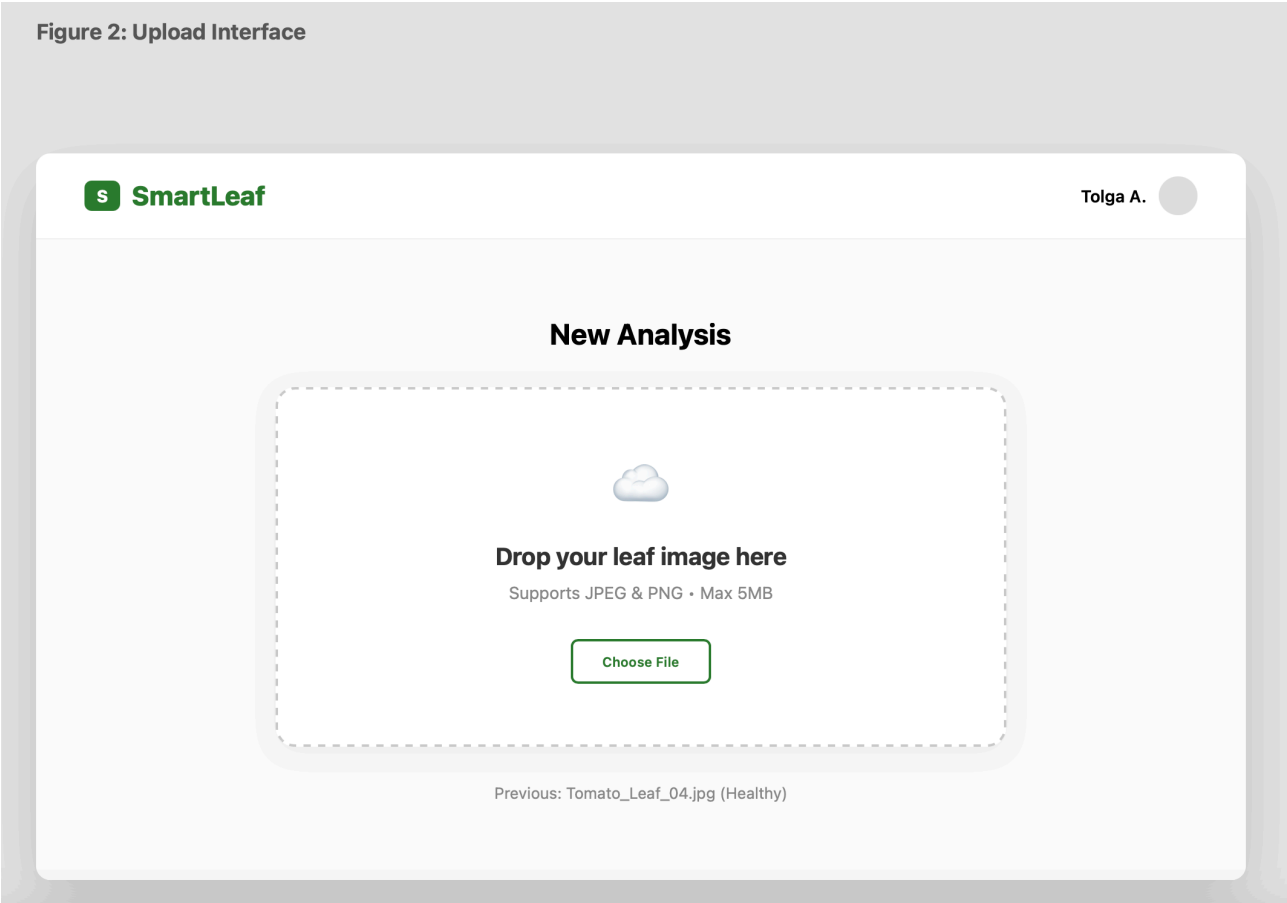
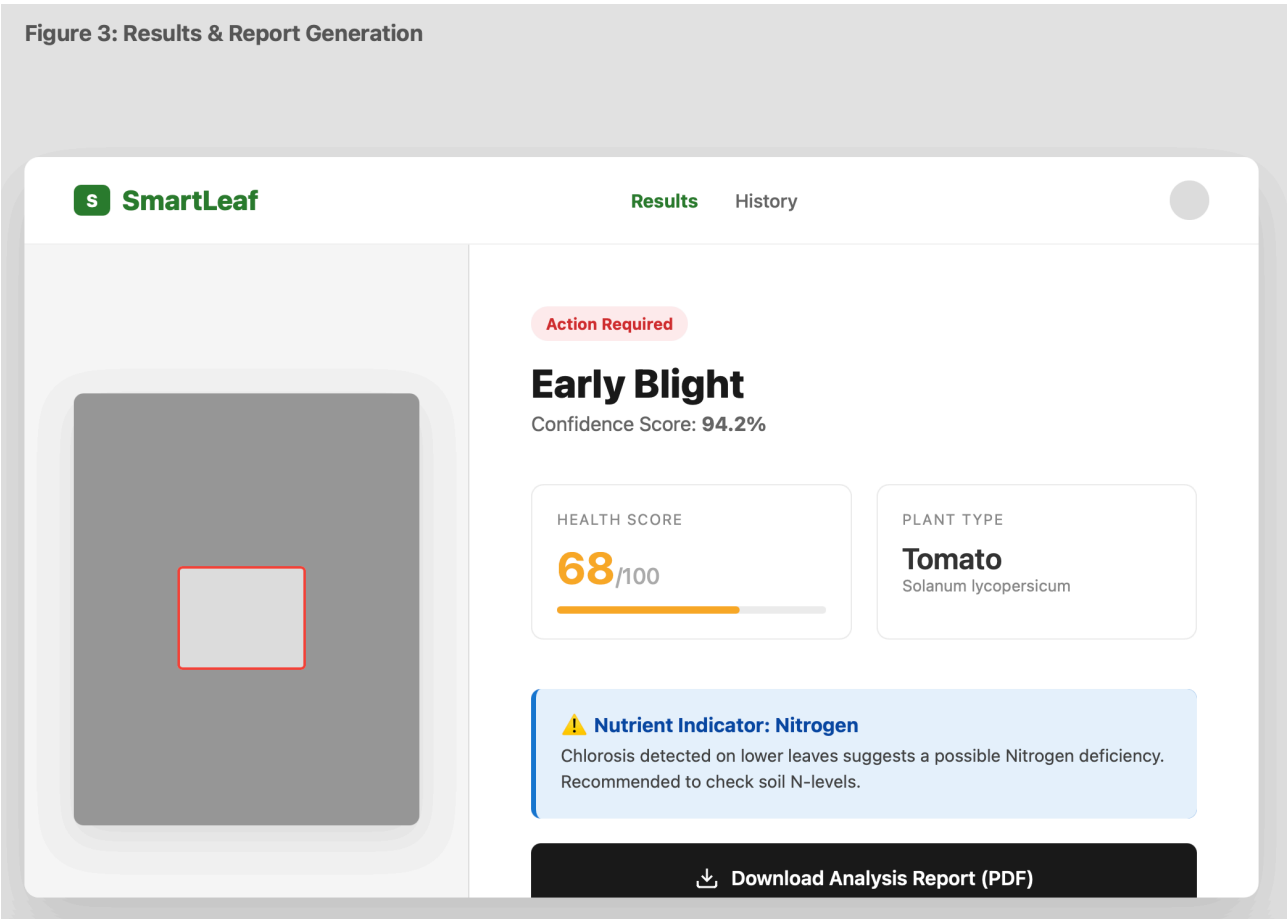


Figure 3: Results & Report Generation



4. Glossary

Artificial Intelligence (AI)

A field of computer science that enables machines to perform tasks that typically require human intelligence, such as classification or pattern recognition.

Computer Vision

A subfield of AI that focuses on extracting meaningful information from images or videos.

Deep Learning

A machine learning approach that uses neural networks with many layers to learn representations from data.

Leaf Segmentation

The process of isolating the plant leaf from the background of the image so that analysis can be performed more accurately.

Disease Classification

The task of categorizing a leaf image into a specific disease class or determining whether it is healthy.

Health Score

A numerical value (0–100) used to indicate the general health condition of a plant leaf based on color and texture properties.

Nutrient Deficiency Indicator

A component of the system that detects visual patterns related to insufficient nutrients such as nitrogen or potassium.

Confidence Score

A numerical value representing how certain the model is about its prediction.

Use Case

A description of how a user interacts with the system to achieve a particular goal.

Actor

Any user or external component that interacts with the system.

Scenario

A step-by-step description of how the system behaves in a particular situation.

Sequence Diagram

A model showing the interaction between system components over time.

Class Model

A representation of the classes, objects, and relationships used within an object-oriented system.

Preprocessing

Initial transformations applied to the input image (e.g., resizing, normalization) before analysis.

Dataset

A collection of images used to train, validate, or test machine learning models.

PDF Report

A downloadable file containing the analysis results generated by the system.

5. References

[1] SmartLeaf Project Proposal, 2025.

(Internal course document submitted to CMPE491.)

[2] SmartLeaf Project Specifications Report, 2025.

(Internal course document submitted to CMPE491.)

[3] SmartLeaf Project Website. Available at: <https://smart-leaf-nine.vercel.app/>

[4] PlantVillage Dataset. Pennsylvania State University. Available at: <https://plantvillage.psu.edu/>

[5] B. Bruegge and A. Dutoit, Object-Oriented Software Engineering: Using UML, Patterns, and Java, 2nd Edition, Prentice Hall, 2004.

[6] SmartLeaf UI Mockup Prototype (HTML Export)

SmartLeaf Project Team, Interactive UI Wireframe Mockups (Home, Upload, Results Screens),

Custom HTML Prototype, Generated using AI-assisted design (Gemini), November 2025.