Grow Wise
Software Requirement Specification

Table of Contents

1. Introduction

1.1. Purpose

This document aims to provide end-user interactions with the Grow Wise application, as well as all functional and non-functional requirements. Once finalized, this document will state what must be accomplished for the application to be considered complete.

1.2. Scope

The Grow Wise application will be a web application that will be able to identify diseases/infections that affect some plants through their leaves, as well as their cure. It will also suggest crops to seed based on soil tests and environmental data input by the user. Any device with an active internet connection and a web browser can be used to access the Grow Wise web application. Users can directly use any provided services without registering themselves on the system.

1.3. Product Perspective

1.3.1. System Interface

The web application will run on the latest versions of Chrome, Firefox, Safari, and Edge browsers on Windows, Linux, and Mac. It should be designed and developed to ensure proper rendering and functionality across these browsers. Image captured by any smartphone or digital camera having a minimum of 2 megapixels of resolution. A machine learning model will identify the diseases from the extracted features of the processed image and predict the disease. It will also recommend crops to sow based on user-input soil test data. The database stores comprehensive disease information, including disease names, symptoms, and associated treatment recommendations, as well as providing sowing information about crops.

1.3.2. User interface

The user interface of the web app is designed to be user-friendly and intuitive, allowing users to easily interact with the system. The GUI provides menus, toolbars, buttons, panes, containers, and grids, allowing for easy control.

1.3.3. Hardware Interface

The web application will be compatible with a wide range of devices that have an internet connection and a web browser. If the device has a camera, it can take a picture of the cotton plant leaf instantly instead of uploading a pre-captured image.

1.3.4. Software Interface

The web app allows the user to select an image from files in PNG, jpg, and jpeg formats. The app leverages image processing libraries to analyze the uploaded images of plant leaves, identify diseases, and suggest the required treatment for the plants. The user can enter the soil test and environmental data as input for the crop recommendation system.

1.4. User Characteristics

The end users of this web app, including farmers, agricultural researchers, and agronomists, are individuals involved in plant cultivation. They are capable of using devices such as mobile phones or other devices with internet connectivity and a web browser to access the web app. They can capture clear digital images of plant leaves and upload them, or they can enter data using a keyboard into the web app without requiring advanced computer skills.

1.5. Limitations

The constraints of the web application depend on the nature of the picture given by the client. The exactness of the illness distinguishing proof, treatment proposal, and yield suggestion highlights is vigorously dependent on the nature of the client input information. On the off chance that the client transfers a low-goal picture, the web application cannot distinguish the infection appropriately. The exactness of the harvest proposal framework will likewise be impacted by unexpected changes in climatic circumstances.

1.6. Assumptions and Dependencies

- A sufficient dataset and collaboration with domain experts are necessary to train the ML model to meet standard accuracy.
- Users are expected to have a stable internet connection.
- Users need to make sure that their device and browser are compatible with the web app.
- Users also need to input current soil test data accurately.
- Users also need to take pictures of the affected plant as clearly as possible.

1.7. Acronyms and Abbreviations

- **SRS:** Software Requirements Specification
- **GUI:** Graphical User Interface
- **ML:** Machine learning
- **API:** Application program interface
- **UI:** User interface
- **PNG:** Portable network graphics
- **JPG/ JPEG:** Joint photographic expert group

2. Requirements

2.1. External Interface

2.1.1. User Interface (UI)

The UI allows users to interact with the web app. It is intuitive and user-friendly. Users can upload images, view disease identification results, and access treatment recommendations. The user can input their current soil test data and view the recommended crop.

2.1.2. Hardware Interface

The system must be able to accept images of the leaves of cotton plants from a variety of input devices, such as smartphones, tablets, etc.

2.1.3. Software Interface

2.1.3.1. Image Processing Libraries

The system will use image processing libraries to analyze the images of plant leaves and extract pertinent features to identify diseases.

2.1.3.2. Database Interface

The system will need to be integrated with a database to store and retrieve information about plants, their growth patterns, characteristics, diseases, and treatment recommendations.

2.2. Functions

2.2.1. Functional Requirements

Title: Image upload and processing

Description: The image upload and processing allow the user to upload an image of a cotton plant leaf through the web interface. The system processes the uploaded image using image processing algorithms to enhance image quality, remove noise, and extract relevant features for disease identification.

Title: Suggesting the treatment

Description: This function allows the user to view the name of the identified diseases and their respective treatments.

Title: Searching for plant, disease, and symptoms information

Description: Access to information about plants, including details such as growth conditions, preferred soil types, and environmental requirements. Also, information about diseases and methods for minimizing the likelihood of infection must be made available to users.

Title: Input crop recommendation data

Description: This function allows users to enter soil test and environmental data.

Title: View suggested crop

Description: This function allows the user to view the recommended crop.

2.3. Usability Requirements

- The system's UI should be as minimalistic as possible so that users can easily learn to interact with it.
- The predicted disease or suggested cure should be unambiguous.
- The system needs to have greater fault tolerance.

2.4. Performance Requirement

- **Response time:** The system's loading time should be minimal, and it should swiftly identify diseases and propose treatments upon the upload of an image. Furthermore, the system should respond expeditiously to the input of recommendation data.
- **Accuracy:** The system should be capable of accurately diagnosing disease and proposing treatments that have proven efficacy in combating the ailment. Moreover, the system should possess the capability to recommend crops accurately, contingent upon soil test results and environmental data.
- **Availability:** The system should have an uptime of about 99%.

2.5. Logical Database Requirement

- Data stored about defective cotton leaves must be properly annotated
- Database need to store various plant details, such as growth conditions, preferred soil types, and environmental requirements.
- There should be information about pests, how they cause damage, and what the early symptoms are on cotton leaves.
- Data about cures must also be labeled appropriately for diseases.

- User interactions and feedback also need to be stored.
- The database should store system configuration and settings that govern the behavior and functionality of the application.
- The database also needs to add information about data backup methods in case of data loss.

2.6. Design Constraints

2.6.1. Platform Compatibility

- The web app should be compatible with popular web browsers like Chrome, Firefox, Edge, and Safari.

2.6.2. Device Compatibility

- The web app should be accessible and functional on various devices, including desktop computers, laptops, tablets, and smartphones.
- The user interface should adapt and provide a consistent user experience across different screen sizes and resolutions.

2.6.3. Image Format Support

- The system should support the upload and processing of image files in commonly used formats such as PNG, JPG, and JPEG.

2.6.4. Performance and Response Time

- The system should be designed to deliver responsive performance, providing timely disease identification, treatment recommendations, and crop recommendations.
- Consider optimization techniques such as efficient algorithms, caching mechanisms, and server-side optimizations to minimize response times.

2.7. Software System Attributes

- **Accuracy:** The system should correctly identify plant diseases, suggest effective treatments, and recommend appropriate crops
- **Speed:** The system should be able to identify diseases quickly.
- **Ease of use:** The system should be easy to use.
- **Reliability:** The system should be reliable and should not crash or freeze.
- **Security:** The system should be secure and protect user data.
- **Scalability:** The system should be scalable and able to handle a large number of users and requests.
- **Maintainability:** The system should be maintainable and easy to update and fix.
- **Usability:** The system should be usable and provide a good user experience.
- **Accessibility:** The system should be accessible to users with disabilities. This means that the system should use features such as screen readers and high-contrast text to make it accessible to users who are blind or have low vision.