|  |  |  |
| --- | --- | --- |
| **9Leaf Disk Analyzer** | | |
| Team Information | 1. Alex Wilson  2. Colton Eddy  3. Connor Jansen  4. Emily Box  5. Erica Gitlin  6. Kyle Sargent | |
| Document Type: Software Requirement Specification v1.0 | | *Last updated on*: mm-dd-yyyy |

**Document Approval**

This Software Requirements Specification has been accepted and approved by the following stakeholders:

|  |  |  |
| --- | --- | --- |
| **Printed Name** | **Title** | **Signature and Date** |
|  |  |  |
|  |  |  |
| R. IQBAL | Customer/Course Instructor |  |

**Table of Contents**

1. Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 References 1

1.5 Overview 1

# 1. Introduction

The introduction to the Software Requirement Specification (SRS) document should provide an overview of the complete SRS document. While writing this document please remember that this document should contain all of the information needed by a software engineer to adequately design and implement the software product described by the requirements listed in this document.

Note-1: You must replace anything in italics with your own team project related information.

Note-2: If a section/sub-section is not applicable to your project then do not remove the section/sub-section – just write Not Applicable.

## 1.1 Purpose

The purpose of this document is to give a detailed description of the Leaf Disk Analyzer. It will provide the purpose, features, and functions of the system. It will also provide all functional and non-functional requirements as well as constraints in which the system must follow. The document is intended for both the stakeholders and the developers of this project.

## 1.2 Scope

The Leaf Disk Analyzer is a GUI-based computer application that allows users to upload images of grapevine leaves and analyze how much mildew has grown on the leaves. The user will have the option of uploading multipule files, the software will then process those images using image analysis, quantify the mildew that resides on the leaf, and give results specifying the total surface area of the leaf that contains mildew. The user then has the option to transfer the data to a spreadsheet to keep a record of all information collected.

The Leaf Disk Analyzer can be extremely useful in reducing the loss of grape production worldwide. By quantifiying the amount of mildew on the plant, we will be able to map the location of the plant genes and determine how resistant the plant is to the mildew, thus providing an inexpensive, non-destructive way of identifying this bacteria*.*

## 1.3 Definitions, Acronyms, and Abbreviations

**GUI:** Graphical User Interface, software that allows a user to interact with the program through graphical icons, such as buttons and tect boxes

**OpenCV:** The library of python functions used to analyze the images

**Python:** The programming language used on this project.

## 1.4 References

*This subsection should:*

*(1) Provide a complete list of all documents referenced elsewhere in the SRS, or in a separate, specified document.*

*(2) Identify each document by title, report number - if applicable - date, and publishing organization.*

*(3) Specify the sources from which the references can be obtained.*

## 1.5 Overview

The remainder of this document is organized into 3 more chapters and appendicies. Chapter two will provide a general overview of the functions the product will preform, perspective on how the product relates to other similar products, and the characteristics of the intended user. Furthermore, this chapter will cover any general constraints, assumptions, and dependencies.

Chapter three will cover specific functional and non-functional requirements, as well as design constraints, any external hardware requirements and any external database requirements.

The fourth chapter deals with the process we will use when dealing with any change in requirements or scope, who can make these changes and how they will be approved.

The appencices will provide any additional information that may be helpful.

# 2. General Description

This section of the SRS will describe the general factors that affect the product and its requirements. This section must not state specific requirements; it only makes those requirements easier to understand in their respective sections.

## 2.1 Product Perspective

The product being developed will be a cross-platform desktop application. We will be using the OpenCV Python library to develop the image analysis software. This software will be integrated with a GUI-based user interface as well as a spreedsheet software to export results.

## 2.2 Product Functions

With this product, the user will be able to upload several files that they wish to analyze. The software will then process all images simultaniously and calculate how much of the leaf consists of mildew. Once the software is finished processing, the system will display a percentage of the total surface area of pathogen detected on each file. The user will then have the option to export all results to a spreadsheet that will be organized by file name.

## 2.3 User Characteristics

This subsection of the SRS should describe those general characteristics of the eventual users of the product that will affect the specific requirements.

## 2.4 General Constraints

*This subsection of the SRS should provide a general description of any other items that will limit the developer’s options for designing the system.*

## 2.5 Assumptions and Dependencies

This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption might be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

# 3. Specific Requirements

## 3.1 Functional Requirements

### FR.1 The user must be able to upload photos so the software can analyze them.

Source: Dr. Laszlo Kovacs

Priority: 10

Introduction: Upon loading the software, the user will be prompted to upload 8 photos to be analyzed.

Inputs: A set of 8 photos that the user wishes to analyze

Processing: The images will be selected by the user and readied for analyzing by the software.

Outputs: The system will notify the user that the upload is complete and is beginning the analyzing process.

Error Handling: Must make sure that what is chosen to be uploaded is a valid path to a photo and is actually a path to a photo.

### FR.2 Users must be able to specify the amount of days after the disk was injected with the pathogen.

Source: Dr. Laszlo Kovacs

Priority: 10

Introduction: The different days will specify what pictures may be uploaded.

Inputs: Photos that reside in that photo set.

Processing: The number of days past injection will be input by the user.

Outputs: System will notify the user that they may only look in the specific folder with the corresponding number of days past injection.

Error Handling: If the user tries to use photos from a different set, and error will be thrown to the user asking them to pick a photo in the respective set.

### FR.3 Uploaded photos must be analyzed and determine a mildew to leaf ratio

Source: Dr. Laszlo Kovacs

Priority: 10

Introduction: Once the images are uploaded, the software must determine a ratio of mildew to leaf for the current disk.

Inputs: A set of 8 photos that the user wishes to analyze

Processing: The software will analyze the photos that are uploaded using OpenCV Image Tools

Outputs: The software will return the a ratio of mildew to leaf disk ratio back to the user in percentage form.

Error Handling: If the photo is of a non-suitable quality, then an error will be displayed asking for an image of higher quality. If the analyzing is taking too long, then an error message will be displayed asking the user to retry the upload.

### FR.4 Results from the analyzing process shall be exported to spreadsheet form.

Source: Kyle Sargent, Emily Box, Erica Gitlin, Alex Wilson, Connor Jansen, Dr. Laszlo Kovacs

Priority: 10

Introduction: After analyzing the photos and detemining a ratio of mildew to leaf, the software must export the data collected to spreadsheet form.

Inputs: Results from the analyzing of the photos uploaded

Processing: Each result will be written to the output file by the system.

Outputs: The software willoutput a file containing plant number, sample number, and percentage of mold identified with the .csv or xls extension.

Error Handling: If writing to a file doesn’t work for the photo currently being worked on by the system, then an error will be returned to the user.

## 

## 3.2 Non-Functional Requirements

The image processesor will be able to determine the percent of mold on the wine leaves in each individual pictures within <set time limit> depending on the size of the photo. The excel document shall be written within<set time limit> depending on how many picure are added to the system. The system shall be able to accept photo between <smallest image size> and <largest image size> all within the target time. The excel document should be able to record between and <largest amount of photos> within the target time. The accuracy of the estimate of mold should be at least 95% accurate to the amount of mold depicted in the picture.

## 3.2 Non-Functional Requirements

### NFR.1 Each individual picture will be processed within 3 seconds

Source: Dr. Laszlo Kovacs/Dr. Razib Iqbal

Priority: 1

Introduction: The image processesor will be able to determine the percent of mildew on the wine leaves in each individual pictures within no longer than 3 seconds depending on the size of the photo with 95% of the imported photos.

Inputs: digital images of types (.jpg or .png) or a folder containing the trays taked from the sample

Processing: Each photo shall take no longer than 3 seconds

Outputs: mildew percentage, tray name, sample name

Error Handling: If the image is too large or too small to process an error message will appear to the user. Also if the image is of an unsupported type an error message will appeat to the user.

### NFR.2 The excell sheet shall add each result within 1 second of the image being processed

Source: Dr. Laszlo Kovacs

Priority: 4

Introduction: The excel document shall add each result to the excell spread sheet no longer than 1 second per image. The total time completion on the spread sheet will depend on the number of trays added per sample.

Inputs: sample name, tray number, and percent mildew per tray will be entered into the excel document.

Processing: The total time for the document will depend on the amount of images inputed into the software.

Outputs: An excell sheet containing sample name, tray number, and percent mildew on plant.

Error Handling: If the excell document fails to be made a error message is sent to the user.

### NFR.3 Mildew coverage will be classified after the eight photos are analyzed.

Source: Dr. Laszlo Kovacs

Priority: 1

Introduction: Once all images have been uploaded and analyzed, classification of the mildew coverage can begin. Mildew coverage will belong to two categories: heavily or lightly covered on the leaf disk. We can further detail what percentage of the leaf disks had heavy coverage versus light coverage of mildew as well. Heavy coverage will be considered if more than 50% of the leaf disk is covered. Whereas light coverage will be considered if less than 50% of the leaf disk is covered.

Inputs: digital images of types (.jpg or .png) or a folder containing the trays taked from the sample

Processing: This calculation should take no longer than the three seconds allowed per photo

Outputs: A string containg the information of its classification “heavy coverage” or “light coverage”

Error Handling: If the classification cannot be determined the the string will contain “unable to classify”

### NFR.4 Growth areas for the mildew shall be determined once the eight photos are analyzed.

Source: Dr. Laszlo Kovacs

Priority: 1

Introduction: After analyzing all the images, we can track where the mildew grows on the disks. Thus, we can utilize this data to quantify how many disks grew mildew on the edges, center, between the edge and center, left, right, top-left, top-right, bottom-left, and bottom-right sections of the leaf disk.

Inputs: digital images of types (.jpg or .png) or a folder containing the trays taked from the sample

Processing: This calculation should take no longer than the three seconds allowed per photo

Outputs: A message specifying area containing amount of growth per area of tray

Error Handling: If the specific areas cannot be determined the the string will contain “unable map growth area”

## 3.3 Design Constraints

### 3.3.1 Time

The main constraint of the software is the time alotted for development. Due to the nature of the project, the software must be in a working state that fulfills all functional and non-functional requirements by the 15th of May in 2019. Additionaly, milestones and demonstartions must be completed by assigned dates to be given or revised by Dr. Razib Iqbal, the instructor.

### 3.3.2 Cost

The software project is a learning experience for the developers, and so has no fiscal budget.

### 3.3.3 Operating System

Dr. Laszlo Kovacs demonstrated example images on the machine he will be running our software on, which ran a windows operating system. The system must be capable of running at full potential on a windows device.

## 3.4 External Interface Requirements

### 3.4.1 User Interfaces

### EIR.1 GUI?

Source:

Priority:

Introduction:

Inputs:

Processing:

Outputs:

Error Handling:

…

### 3.4.2 Hardware Interfaces

### EIR.2 < External Interface Requirement #2>

N/A

### 3.4.3 Software Interfaces

GUI?

### 3.4.4 Communications Interfaces

N/A

## 3.5 Logical Database Requirements

### LDR.1 < Logical Database Requirement #1>

Source: User input of infected leaf disks

Priority: High

Introduction: The application will take in images of grapevine leaves infected with a pathogen.

Inputs: .tif, .jpeg, .png images

Processing: Pathogen detection with OpenCV

Outputs: .csv or .xls.

Error Handling: Throw error visable to users.

## 3.6 Other Requirements

Catchall section for any additional requirements.

# 4. Change Management Process

### 4.1 Update Process

Should this document, the SRS, need to be changed or updated, the 6 developers would need to discuss the ramifications of the change, before electing one of them to update the .docx file. The updated file will then be sent to Dr. Iqbal and Dr. Kovacs for approval.

# Appendices

Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*