|  |  |  |
| --- | --- | --- |
| **Leaf Disk Analyzer** | | |
| Team Information | 1.Kyle Sargent  2.Colton Eddy  3.Erica Gitlin  4. Emily Box  5. Connor Jansen  6. Alex Wilson | |
| Document Type: Software Requirement Specification v1.0 | | *Last updated on*: 2/11/2018 |

**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 Overview 1

2. General Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Characteristics 2

2.4 General Constraints 2

2.5 Assumptions and Dependencies 2

3. Specific Requirements 3

3.1 Functional Requirements 3

FR.1 The software must allow the users to enter the date of the samples they want to upload. 3

FR.2 The software must allow the user to specify the trays number(s) they want to use. 3

FR.3 The software must allow the user to specify the picture number(s) they want to analyze. 3

FR.4 The software must analyze the user-specified photos to determine a leaf to mildew ratio. 4

FR.5 The software must export results from the analysis process to spreadsheet form. 4

3.2 Non-Functional Requirements 4

NFR.1 Each individual picture must be processed within 1 second 4

NFR.2 Growth rates for the mildew shall be determined with no less than a 90% accuracy rate 5

NFR.3 Up to eight photos may be uploaded simultaneously. 5

3.3 Design Constraints 5

3.3.1 Time 5

3.3.2 Cost 5

3.3.3 Operating System 5

3.4 External Interface Requirements 6

3.4.1 User Interfaces 6

EIR.1 The GUI must allow a user to create a new spreadsheet 6

EIR.2 The GUI must allow a user to open an existing spreadsheet 6

EIR.3 The GUI must have a drop down calendar for picking the date. 6

EIR.4 The GUI fields must support use of the “-“ operator to select a range of values. 6

EIR.5 The GUI fields must support use of the “,“ operator to select a list of values. 7

EIR.6 The GUI must disable all fields once the upload button is clicked 7

4. Change Management Process 7

4.1 Update Process 7

# 1. Introduction

## 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 multiple 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 quantifying 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 textboxes

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

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

## 1.4 Overview

The remainder of this document is organized into 3 more chapters and appendices. Chapter two will provide a general overview of the functions the product will perform, 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 appendices 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 Windows 10 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 spreadsheet 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 simultaneously 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

Dr. Kovacs and the people working in his lab will likely be the only people who ever use this software for its intended purpose, but it should still be easy to use. Using a GUI helps keep the learning curve down and enables less computer literate people to use the software. If the user understands how to open the program and navigate the file system, they should be able to use the software. Because the intended users wish to use the software for further academic research in a scientific discipline it needs to be very accurate.

## 2.4 General Constraints

The biggest constraint will be performance. Balancing the speed of analysis with the accuracy of the analysis. We expect that the program should take negligible or little time to run but should be very accurate to meet the stakeholders needs. There is no fiscal cost for this project but the time to implement it is limited.

## 2.5 Assumptions and Dependencies

It is assumed that the user will be using Windows operating system and that the minimum required hardware will be available. The minimum required hardware should be as low end as possible because we want this software to be able to run on a laptop or desktop not just on over-priced lab computers. We assume that we can accurately measure the amount of pathogen in a relatively short amount of time, e.g. less than two minutes for a set of eight 4K images. We assume that the users will want to analyze one or more sample at a time. All of this is dependent on using OpenCV. OpenCV is open source and publicly available but it could still cause problems since we are relying on a specific version’s implementation in our code. We assume that we will release the project with Python 3.4.1 and OpenCV 4.0.1.

# 3. Specific Requirements

## 3.1 Functional Requirements

### FR.1 The software must allow the users to enter the date of the samples they want to upload.

Source: Team Decision

Priority:

Introduction: The date is used for registering how many days past inoculation the samples are.

Inputs: None

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

Outputs: None

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

Related Requirements: FR4

### FR.2 The software must allow the user to specify the trays number(s) they want to use.

Source: Team Decision

Priority:

Introduction: The tray number(s) will determine which samples are pulled by the software for analysis.

Inputs: A number that specifies the number of trays the user will be pulling samples from.

Processing: The images will be selected by the user and readied for analysis by the Leaf Disk Analyzer.

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

Error Handling: Must make sure that the specified path is valid, if not an error is returned to the user.

Related Requirements: FR4

### FR.3 The software must allow the user to specify the picture number(s) they want to analyze.

Source:  Team Decision

Priority:

Introduction: The picture numbers will determine which samples are pulled by the software for analyzing.

Inputs: A single image number or collection of image numbers in the format of 1,2,3, or a range of numbers in the format of 1-9.

Processing: The images will be picked by the software and analyzed.

Outputs: None

Error Handling: Prompt the user to re enter the numbers in the specified format.

Related Requirements: FR4

### FR.4 The software must analyze the user-specified photos to determine a leaf to mildew ratio.

Source: Dr. Laszlo Kovacs – Customer Meeting

Priority:

Introduction: Once the images are specified, the software must use image processing tools to analyze the disk.

Inputs: The photos that the user wishes to analyze.

Processing: The software will analyze the photos that are specified using OpenCV Image Tools and the Canny Edge Detection Algorithm.

Outputs: The software will return 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 analysis is taking too long, then an error message will be displayed asking the user to retry the upload.

Related Requirements: FR1, FR2, FR3, NFR1, NFR2

### FR.5 The software must export results from the analysis process to spreadsheet form.

Source: Dr. Laszlo Kovacs – Customer Meeting

Priority:

Introduction: The system will write the results from analysis to a file.

Inputs: The results of the analysis of the photos specified.

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

Outputs: The software will output a file containing plant number, sample number, and percentage of mildew identified for each image in each tray with the .csv or .xls extension.

Error Handling: If writing to the output file is unsuccessful, then an error will be returned to the user.

Related Requirements: FR4, NFR2

## 3.2 Non-Functional Requirements

### NFR.1 Each individual picture must be processed within 1 second

Source: Team Decision

Priority:

Introduction: The image processor will be able to determine the percent of mildew on the grapevine leaves in each individual pictures. This will occur within no longer than 1 second depending on the size of the photo.

Inputs: digital images of types (.jpg or .png or .tif/.tiff) or a folder containing the trays taken from the sample

Processing: Each photo shall take no longer than 1 second(s)

Outputs: mildew percentage, tray name, sample name

Error Handling: If the image is unable to be processed, an error will be thrown to the user.

Related Requirements: FR4, FR5

### NFR.2 Growth rates for the mildew shall be determined with no less than a 90% accuracy rate

Source: Team Decision

Priority:

Introduction: After analyzing all the images, we can determine the rate of the mildew growth per image with no less than a 90% accuracy rate.

Inputs: Images (allowed formats: .jpg or .png or .tif/tiff) taken from the specified tray folder within the specified date folder.

Processing: This calculation should take no longer than the one second allowed per photo.

Outputs: A message specifying the percentage of growth per tray.

Error Handling: If the percentage of mildew growth cannot be determined, the user will be informed.

Related Requirements: FR4, FR5

### NFR.3 Up to eight photos may be uploaded simultaneously.

Source: Team Decision

Priority:

Introduction: Once the images are specified, the software must analyze all photos using a multithreading process.

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

Processing: The software will analyze the photos that are specified for OpenCV image tools and multithreading.

Outputs: The software will return a ratio of mildew to leaf disk for up to the max number of photos at the same time.

Error Handling: If the analysis is taking too long, then an error message will be displayed.

Related Requirements: FR4, FR5

## 3.3 Design Constraints

### 3.3.1 Time

The main constraint of the software is the time allotted 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. Additionally, milestones and demonstrations 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 The GUI must allow a user to create a new spreadsheet

Source: Team Decision

Priority:

Introduction: A button will be implemented to complete this function

Inputs: None

Processing: None

Outputs: A new spreadsheet to hold the results from the analysis process

Error Handling: If an error occurs while trying to create a new spreadsheet, an error is returned to the user.

### EIR.2 The GUI must allow a user to open an existing spreadsheet

Source: Team Decision

Priority:

Introduction: A button will be implemented to complete this function

Inputs: An existing spreadsheet saved on the user’s system.

Processing: None

Outputs: The spreadsheet name will be sent to the software

Error Handling: If an error occurs while trying to open an existing spreadsheet, an error is returned to the user.

### EIR.3 The GUI must have a drop down calendar for picking the date.

Source: Team Decision

Priority:

Introduction: A drop-down buttons will be implemented for use.

Inputs: None

Processing: None

Outputs: The date will be sent to the software for use in the spreadsheet as well as picture acquisition

Error Handling: If the date does not match a folder, then an error is sent to the user.

### EIR.4 The GUI fields must support use of the “-“ operator to select a range of values.

Source: Team Decision

Priority:

Introduction: The tray number(s) and picture number(s) input fields should support ranges of values like “1 – 3”. This range would include one, two, and three.

Inputs: Data entry from the user.

Processing: Text will be parsed, and the corresponding information will be used select images.

Outputs: None

Error Handling: If a user uses an unsupported operand an error will display prompting to enter the data and verify the formatting.

### EIR.5 The GUI fields must support use of the “,“ operator to select a list of values.

Source: Team Decision

Priority:

Introduction:  The tray number(s) and picture number(s) input fields should support a list of values separated by commas. For example, “1,2, 4” would return values one, two, and four.

Inputs: Data entry from the user.

Processing: Text will be parsed, and the corresponding information will be used select images.

Outputs: None

Error Handling: If a user uses an unsupported operand an error will display prompting to enter the data and verify the formatting.

### EIR.6 The GUI must disable all fields once the upload button is clicked

Source: Team Decision

Priority:

Introduction: Clicking the upload button will disable all fields and buttons from use.

Inputs: None

Processing: None

Outputs: A popup notifying the user that the analysis is taking place.

Error Handling: If a button is clicked while the popup is up, a message is shown to the user.

# 4. Change Management Process

### 4.1 Update Process

Should this document or any requirements need to be changed or updated, the Project Manager or Scrum Master must fill out the Change Request Form found here:

<https://github.com/KySarge23/LeafDiskAnalyzer/blob/master/documents/Change_Request_Form.doc>

Then, the 6 developers would need to discuss the ramifications of the change, detailed in the change request form, via Discord or in person. Every participant gets 1 minute for discussion. During their allotted time slot, each person will list their view of the pros and cons of the change and its effect on our current and projected tasks. After discussion has taken place, the team will come together and decide whether the change can be implemented. The maximum amount of time for these meetings will be 30 minutes. If the change is approved by the team, it then will go to Dr. Iqbal and Dr. Kovacs for final approval. Once an approval is returned by Dr. Iqbal and Dr. Kovacs, this document will be updated and sent back to the stakeholders to make sure these changes are indeed what was requested. Upon finalizing of the changes, the changes will be documented. However, if the change is not approved by the team, the team will draft a change they feel can be implemented based off the original request. This new change will then be sent to the stakeholders, Dr. Iqbal and Dr. Kovacs, for approval. Once this request is approved, documentation and implementation will begin.