|  |  |  |
| --- | --- | --- |
| **Leaf Disk Analyzer** | | |
| Team Information | 1.  2.  3.  4.  5.  6. | |
| 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

*This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents.*

## 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

*This subsection of the SRS puts the product into perspective with other related products or projects. (See the “IEEE Guide for Software Requirements Specifications”).*

## 2.2 Product Functions

This subsection of the SRS should provide an outline/summary of the functions that the software will perform. You should consult with your project charter and formal customer meeting notes.

## 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

The next few sentences are for your understanding purpose. Remove these sentences/instructions and start from section 3.1.

This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 2, but this section will give the detailed requirements that are used to guide the project’s software design, implementation, and testing.

Each requirement in this section should be:

* Uniquely identifiable (usually via numbering like 1,2.3,4.1.1 etc.)
* Correct
* Complete
* Consistent
* Unambiguous
* Traceable
* Prioritized
* Verifiable (i.e. testable)

Attention should be paid to the careful organization of the requirements presented in this section so that they may easily be accessed and understood. Furthermore, this SRS is not the software design document, therefore one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.

## 3.1 Functional Requirements

### FR.1 <Functional Requirement or Feature #1>

Source:

Priority:

Introduction:

Inputs:

Processing:

Outputs:

Error Handling:

…

## 3.2 Non-Functional Requirements

Non-functional requirements may exist for the above functional requirements or other generic attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second; System downtime may not exceed 1 minute per day, etc). Categories for your reference - Performance, Reliability, Availability, Security, Maintainability, Portability.

### NFR.1 <Non-Functional Requirement #1>

Source:

Priority:

Introduction:

Inputs:

Processing:

Outputs:

Error Handling:

…

## 3.3 Design Constraints

Specify design constrains imposed by other standards, company policies, hardware limitation, etc. that will impact this software project.

## 3.4 External Interface Requirements

### 3.4.1 User Interfaces

### EIR.1 < External Interface Requirement #1>

Source:

Priority:

Introduction:

Inputs:

Processing:

Outputs:

Error Handling:

…

### 3.4.2 Hardware Interfaces

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

…

### 3.4.3 Software Interfaces

…

### 3.4.4 Communications Interfaces

…

## 3.5 Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.

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

Source:

Priority:

Introduction:

Inputs:

Processing:

Outputs:

Error Handling:

…

## 3.6 Other Requirements

Catchall section for any additional requirements.

# 4. Change Management Process

Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.

# 

# 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.*