Software Requirements Specification

Website Analysis

Aaron Berman

Philip Conner

Joseph Demarco

Mimi Ho

Carlo Waje

Table of Contents

[**1. Introduction**](#_ev5zw0cdqj7n) **3**

[1.1 Purpose](#_r1zkmqfsk2gh) 3

[1.2 Scope](#_hsshuvd7fn79) 3

[1.3 Definitions, acronyms, and abbreviations](#_gx8z5feol0wi) 4

[1.4 References](#_izz559dpcljy) 4

[1.5 Overview](#_ubzbl1z90us8) 4

[**2. Overall description**](#_ao59a5l1oo1m) **4**

[2.1 Product perspective](#_9lkukr9f737y) 4

[2.2 Product functions](#_q7bw1b6k93e8) 5

[2.3 User characteristics](#_nt3alp7u4ysd) 5

[2.4 Constraints](#_o6l5d49oxw1e) 5

[2.5 Assumptions and dependencies](#_k08pimfr4lqf) 6

[**3. Specific requirements**](#_chhomlrpg4zv) **6**

# **1.** **Introduction**

This section provides the purpose, scope and overview of everything in this SRS document. It also includes a list of definitions, acronyms, and abbreviations used throughout this document.

## 1.1 Purpose

a) Delineate the purpose of the SRS;

b) Specify the intended audience for the SRS.

The purpose of this document is to provide a detailed description of the requirements for the *Website Analysis* software. This document is primarily intended to the professor for his approval and as a reference for development team when constructing the *Website Analysis* software.

## 1.2 Scope

This subsection should

a) Identify the software product(s) to be produced by name (Website Analysis);

b) Explain what the software product(s) will, and, if necessary, will not do;

c) Describe the application of the software being specified, including relevant benefits, objectives, and goals;

d) Be consistent with similar statements in higher-level specifications

The *Website Analysis* is a software that will analyze the static contents on each page within a local copy of a website. The software will analyze each page within the site and the site as a whole. Static contents to be analyze include images, javascript, cascading style sheets, hyperlinks, and data files (archives, video, audio).

The software will be operated from a command-line interface, and the only parameters required is the path to the local copy of the website and one or more URLs that must be translated to the local directory structure as part of analysis (when encountered).

## 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

a) Provide a complete list of all documents referenced elsewhere in the SRS;

b) Identify each document by title, report number (if applicable), date, and publishing organization;

c) Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.

## 1.5 Overview

This subsection should

a) Describe what the rest of the SRS contains;

b) Explain how the SRS is organized.

# 2. Overall description

This section of the SRS should describe the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3 of the SRS and makes them easier to understand.

## 2.1 Product perspective

This subsection of the SRS should put the product into perspective with other related products. If the product is independent and totally self-contained, it should be so stated here. If the SRS defines a product that is a component of a larger system, as frequently occurs, then this subsection should relate the requirements of that larger system to functionality of the software and should identify interfaces between that system and the software. A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful.

## 2.2 Product functions

This subsection of the SRS should provide a summary of the major functions that the software will perform. For example, an SRS for an accounting program may use this part to address customer account maintenance, customer statement, and invoice preparation without mentioning the vast amount of detail that each of those functions requires. Sometimes the function summary that is necessary for this part can be taken directly from the section of the higher-level specification (if one exists) that allocates particular functions to the software product. Note that for the sake of clarity

a) The functions should be organized in a way that makes the list of functions understandable to the customer or to anyone else reading the document for the first time.

b) Textual or graphical methods can be used to show the different functions and their relationships. Such a diagram is not intended to show a design of a product, but simply shows the logical relationships among variables.

Link extraction, link analysis, image analysis, javascript and css analysis

The system should be accessible through a Java API that would allow extraction and analysis of links, images, scripts, and style sheets. The result of the analysis will generate three different files that contain the number and file size of each static content.

## 2.3 User characteristics

This subsection of the SRS should describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise. It should not be used to state specific requirements, but rather should provide the reasons why certain specific requirements are later specified in Section 3 of the SRS.

Who is gonna be using this system?

## 2.4 Constraints

This subsection of the SRS should provide a general description of any other items that will limit the developer’s options. These include

a) Regulatory policies;

b) Hardware limitations (e.g., signal timing requirements);

c) Interfaces to other applications;

d)Parallel operation;

e) Audit functions;

f) Control functions;

g) Higher-order language requirements;

h) Signal handshake protocols (e.g., XON-XOFF, ACK-NACK);

i) Reliability requirements;

j) Criticality of the application;

k) Safety and security considerations.

The system must run on Windows, Linux, and OS/X systems equipped with an appropriate Java JRE.

All analysis is to be performed offline and must not make accesses to live webpages.

The number of pages contained in the site is also a constraint to the system. Websites that contain less than 1 page or more than 1000 pages will produce an error message.

Site boundary is also another constraint.

## 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 may 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**