

 University of Zurich <small>UZH</small>	Software Requirements Specification for HTML Generator Project	Authors: S. Plüss, E. Esati, B. Solenthaler and G.R. Prinz Doc.No.: HTMLGEN-REQ-001 Date: 2017-10-15 Number of Pages: 13
--	---	--

Contents

1	Introduction	2
1.1	Purpose	2
1.2	Scope	2
1.3	Abbreviations	2
1.4	Glossary	2
1.5	References	3
1.6	Overview	3
2	Overall Description	3
2.1	Product Perspective	3
2.2	Product Functions	3
2.3	User Characteristics	4
2.4	Constraints	4
2.5	Assumptions and Dependencies	4
3	Specific Requirements	4
3.1	External Interfaces	5
3.2	Functional Requirements	5
3.3	Non-Functional Requirements	9
3.4	Performance Requirements	10
3.5	Maintainability	11
3.6	Design Constraints	11

Revision History

Date	Version	Description	Author(s)
2017-10-05	0.1	Requirements added	Benjamin Solenthaler
2017-10-05	0.2	Requirements added	Elfat Esati
2017-10-05	0.3	Revision	Benjamin Solenthaler
2017-10-05	0.4	Revision	Elfat Esati
2017-10-06	0.5	Part 'Overall Description' added	Gian Raphael Prinz
2017-10-06	0.6	Chapter 'Introduction' added	Severin Plüss
2017-10-08	0.7	Revision	Gian Raphael Prinz
2017-10-10	1.0	Revision	Severin Plüss
2017-10-13	1.1	Revision	Gian Raphael Prinz
2017-10-15	1.2	Revision	Severin Plüss

1 Introduction

1.1 Purpose

This document delineates the Software Requirements Specification (SRS) of a HTML Generator. It is described within the project description of a software construction course held at University of Zurich. The document not only describes requirements but also design constraints, system interfaces and the performance. It strongly follows the structure defined by the IEEE 830-1998 standard [1] for specifying software requirements.

1.2 Scope

The HTML Generator is a library that software developers can use to create documents in markdown languages, especially HTML. It relieves programmers from the burden to create reports manually. This SRS document focuses on the requirements of the HTML Generator. Therefore, external systems are not described within this document.

1.3 Abbreviations

Term	Explanation
SRS	Software Requirements Specification
HTML	Hypertext Markup Language
API	Application Programming Interface
URL	Uniform Resource Locator

1.4 Glossary

Term	Explanation
HTML Generator	Name of the project, so throughout this document it will be used referring to the project itself.
User	Is the person who is going use the HTML Generator which in this case is the developer himself.
Stakeholder	Person with interest or concern in the software
Library	Collection of modules that can be accessed by external programs
API	Set of tools allowing the communication between different software components.
Tag	Tags are markup language code that defines the structure of an HTML page, such placements of text to be shown, image position and hyperlinks.
Indentation	Indentation helps for a better layout, such as text alignment, margin and so on.

Links	They are hyperlinks. When we click on it, it jumps to another document or to the document itself on a different position.
Markdown	Markup language for annotating documents.
HTML	Markup language on which websites are based. HTML 5 is the most recent version of HTML language. With new elements, attributes, and behavior.
Minification	Process of removing all unnecessary characters from source code without changing its functionality.

1.5 References

[1] IEEE Computer Society, Software Engineering Technology Committee, Committee, Institute of Electrical, and Electronics Engineers, IEEE Recommended Practice for Software Requirements Specifications, IEEE Std. Institute of Electrical and Electronics Engineers, 1998. Previous years documents from ETH has been taken as a reference too.

1.6 Overview

This document consists of four sections. The second chapter introduces the HTML Generator. Functions and limitations are also addressed for the first time in this chapter. It is followed by the third chapter specifying specific requirements (functional as well as non-functional) and describing different system interfaces. The last chapter contains supporting information for this software requirements specification.

2 Overall Description

2.1 Product Perspective

The HTML generator is an Eiffel library for generating reports. It contains several subsystems to satisfy all requirements. But there are not only interfaces between the subcomponents but also between the library and the user. This user interface will be realized in form of an API that allows other Eiffel programs to take advantage of the library's functionalities.

2.2 Product Functions

This section gives a general overview on the functionalities provided by the HTML generator. A detailed description can be found in the third section (Specific Requirements).

The HTML generator

- allows to create static reports
- creates reports that are based on markup languages, especially HTML and Markdown
- allows to include existing markup snippets in a report
- allows to create multipage documents
- supports a wide range of tags
- recognizes invalid inputs
- is based on the newest markup language specifications
- has an API

- does not support import functionalities
- is limited to pure markup code (no JavaScript or CSS supported)
- has no graphical user interface

2.3 User Characteristics

The HTML generator is used by people with a strong background in software construction and experience in programming with Eiffel. The library is not designed for people without advanced informatic skills.

2.4 Constraints

The SRS document is based on a fictional project description and does not represent real conditions. Thus, it is partly based on assumptions rather than requirements elicitation. The project also must be finished until the end of December 2017. This deadline is only determined by the course structure. Real factors (available money, human resources, stakeholder needs, etc.) didn't matter when defining the time span available for working on the project. The platform on which the library is running is limited to Eiffel platforms. Hardware constraints are not assumed.

2.5 Assumptions and Dependencies

It is assumed that the user is familiar with the Eiffel programming language, HTML and integrating and interacting with an API. It is also assumed that the user has a modern computer able to run the software that supports viewing the generated reports. The library depends on the Eiffel programming language and its functionalities. Changes in Eiffel can affect the behavior of the report generator.

3 Specific Requirements

Under this section both functional and non-functional requirements will be exposed. To provide clear and simple explanation, the requirements follow the structure described below.

Requirement ID	ID that allows identifying each requirement uniquely.
Title	Describes the requirement concise.
Description	Defines the requirement in detail.
Priority	Shows the order in which requirements should be implemented. Priorities are classified in 3 groups (highest to lowest) 1, 2, and 3. Requirements of priority 1 are mandatory for the first Implementation; Requirements of priority 2 are mandatory for the final Implementation. A priority greater or equal than 3 represents optional features.
Risk	Specifies the risk of not implementing the requirement. It tells how critical the requirement is to the system as a whole. The following risk levels are defined over the impact of not being implemented correctly. <ul style="list-style-type: none"> ▪ Critical (C) It will break the main functionality of the system. The system cannot be used if this requirement is not implemented. ▪ High (H) It will impact the main functionality of the system. Some function of the system could be inaccessible, but the system can be generally used.

	<ul style="list-style-type: none"> ▪ Medium (M) It will impact some system features, but not the main functionality. The system can still be used with some limitation. ▪ Low (L) The system can be used without limitation, but with some workarounds.
References	The IDs of requirement that are relevant in this context are listed here.

3.1 External Interfaces

All the functions or classes (concepts) available for the users or developers should be available to external programs too in the form of an open API. It will entirely depend on the user or developer who will be using the library to generate documents and what information they retrieve from the reports.

3.2 Functional Requirements

3.2.1 General

Requirement ID	R3.2.5.001
Title	Inexplicit usage of functions
Description	The functions (classes) within the software library should be used within the program body, without defining them explicitly.
Priority	1
Risk	C
References	

Requirement ID	R3.2.5.002
Title	System
Description	The library should be usable from within Eiffel projects.
Priority	1
Risk	H
References	

3.2.2 Supported Tags

Requirement ID	R3.2.1.001
Title	Generating Links
Description	The library should be able to generate HTML links to be placed in the document. Linking to other parts of the report. The user can specify the link display text.
Priority	1
Risk	C
References	

Requirement ID	R3.2.1.002
Title	Linking Documents
Description	The User should be able to reference a Document "B" in another Document "A" to create a link from A to B.
Priority	2
Risk	H
References	

Requirement ID	R3.2.1.003
Title	Generating external http(s) Links
Description	A Document should reference external pages referenced by a URL.
Priority	2
Risk	M
References	

Requirement ID	R3.2.1.004
Title	Referencing images by URL
Description	An image should be generated by using a remote URL (http / ssl).
Priority	2
Risk	H
References	

Requirement ID	R3.2.1.005
Title	Generating images by local file
Description	An image should be generated by using a local file reference.
Priority	2
Risk	H
References	

Requirement ID	R3.2.1.006
Title	Creating a bullet list
Description	The user can create a bullet list of entries inside the document.
Priority	2
Risk	H
References	

Requirement ID	R3.2.1.007
Title	Including Tables
Description	A document should contain data displayed in a two-dimensional table.
Priority	2
Risk	H
References	

Requirement ID	R3.2.1.008
Title	Including unformatted Text
Description	A document should be able to include unformatted text.
Priority	1
Risk	C
References	

Requirement ID	R3.2.1.009
Title	Headings
Description	It is possible to define titles that are marking sections.
Priority	2
Risk	H
References	R3.2.1.008

Requirement ID	R3.2.1.010
Title	Text Styling
Description	It is possible to change the standard style of unformatted text inline. This means changing the color, weight, or font size.
Priority	2
Risk	H
References	R3.2.1.008, R3.2.1.009

Requirement ID	R3.2.1.011
Title	Including existing HTML code
Description	A document should be able to include existing.
Priority	2
Risk	H
References	

Requirement ID	R3.2.1.012
Title	Reading from file
Description	The system should be able to read content from a file for further use of its content.
Priority	2
Risk	H
References	

Requirement ID	R3.2.1.013
Title	Including elements with other tags
Description	A document should be able to contain elements with tags not specified in this document, but supplied by the user while using the library
Priority	2
Risk	H
References	

3.2.3 Document Structure

Requirement ID	R3.2.2.001
Title	Document Title
Description	A document should be able to have a title.
Priority	3
Risk	L
References	

Requirement ID	R3.2.2.002
Title	Indentation
Description	Standard indentation should be used to make the text readable, neat and pretty to the user (formatter will keep spaces and tabs between content tags such as div and span as it's considered to be valid content.).
Priority	2
Risk	M
References	

3.2.4 Output

Requirement ID	R3.2.3.001
Title	Output to HTML (Clear Text)
Description	A document should be presented to the user in clear text that's adhering to the HTML5-Standard.
Priority	2
Risk	H
References	

Requirement ID	R3.2.3.002
Title	HTML Output to local file
Description	A document should be saved to a local file.
Priority	2
Risk	H
References	

Requirement ID	R3.2.3.003
Title	Output to Markdown
Description	A document should be able to have an output to the Markdown language.
Priority	3
Risk	L
References	

Requirement ID	R3.2.3.004
Title	Minified HTML output
Description	A document should be able to be made rid of unnecessary characters to use less space.
Priority	3
Risk	L
References	

Requirement ID	R3.2.3.005
Title	Creating Documents
Description	The user should be able to generate a document.
Priority	1
Risk	C
References	

Requirement ID	R3.2.3.006
Title	Correctly indented HTML output
Description	A document should have an indented output where children are indented relative to its parent by a globally defined space.
Priority	2
Risk	H
References	

3.2.5 Error Handling

Requirement ID	R3.2.4.001
Title	Handling Wrong Tags
Description	The library should store a complete set of tags required to correct wrong tags (when you put several pairs of tags into a document, you must balance each pair of tags, making sure they do not overlap).
Priority	1
Risk	M
References	

Requirement ID	R3.2.4.002
Title	Verifying HTML
Description	A document should contain data displayed in a two-dimensional table.
Priority	3
Risk	M
References	

3.3 Non-Functional Requirements

3.3.1 Usability

Requirement ID	R3.3.1.001
Title	Usability
Description	A developer with programming experience and little exposure to management tools should be able to use all the functions provided in 3 days.
Priority	3
Risk	C
References	

3.3.2 Availability

Requirement ID	R3.3.2.001
Title	Availability
Description	The library should compile in Eiffel Studio and be available for Eiffel Studio.
Priority	1
Risk	C
References	R3.6.0.002

3.3.3 Reliability

Requirement ID	R3.3.3.001
Title	Reliability
Description	The system will always work correctly and uncorrupted considering a failure free system.
Priority	1
Risk	C
References	

3.3.4 Scalability

Requirement ID	R3.3.4.001
Title	Scalability
Description	The HTML Generator library is not restricted for extension.
Priority	3
Risk	C
References	

3.3.5 Robustness

Requirement ID	R3.3.5.001
Title	Robustness
Description	The HTML library is simple and transparent. With little effort a developer can visualize the potential situations it might encounter.
Priority	3
Risk	C
References	

Requirement ID	R3.3.5.002
Title	Dependency on external libraries
Description	The library should not depend on external libraries.
Priority	3
Risk	C
References	

3.4 Performance Requirements

Requirement ID	R3.4.0.001
Title	Response Time
Description	The maximum time to generate the HTML file should be 1 minute. Assuming the HTML generator does not generate dynamic websites and there is no database connection.
Priority	2
Risk	L
References	

Requirement ID	R3.4.0.002
Title	Building Time
Description	The HTML library is a small library and should not take more than 1 minute of building time.
Priority	2
Risk	L
References	

3.5 Maintainability

Requirement ID	R3.5.0.001
Title	Frequent Updates
Description	The library has to be built in a way that allows for updates by the developers without breaking existing functionalities and it has to ensure backwards compatibility
Priority	1
Risk	C
References	

Requirement ID	R3.5.0.002
Title	Coding standards
Description	The code should be logical in every aspect so that high quality code is produced. Standard coding conventions ought to be followed.
Priority	1
Risk	C
References	

Requirement ID	R3.5.0.003
Title	Naming Conventions
Description	We ought to ensure the code is readable and understandable in a long term. To do that standard naming conventions must be followed.
Priority	1
Risk	M
References	

Requirement ID	R3.5.0.004
Title	Change of team members
Description	The library has to be built in a way that allows new members of the development team to easily get accustomed to the project and be able to contribute without a long learning phase
Priority	1
Risk	C
References	

3.6 Design Constraints

The system ought to generate HTML sites. To be more specific it should only produce static HTML sites. The main limitation of the library is that it doesn't allow producing dynamic HTML sites. In addition to that, we are not allowed to use JavaScript or even basic CSS styles for the production of the HTML document. We want to emphasize that this project is an experiment SRS and not all real stakeholders have participated, we assume there are stakeholders with whom we have cooperated.

Requirement ID	R3.6.0.001
Title	Dynamic HTML Sites
Description	The HTML Generator library should only produce static HTML sites. It is not intended for generating dynamic HTML sites.
Priority	
Risk	L

References	
------------	--

Requirement ID	R3.6.0.002
Title	Programming Language
Description	All coding will be done in Eiffel programming language.
Priority	
Risk	L
References	R3.3.1.001

Requirement ID	R3.6.0.003
Title	
Description	The HTML Generator library is a module for the developer or the user. It is a base for building further projects, not a standalone application.
Priority	
Risk	L
References	