Contents

Revision History

Date	Version	Description	Author(s)
15/10/2017	1.0	First Draft of Specification Requirement Sheet	Andrea Meier Anthony Kalbermatten Cheng-Hsin WUU Deniz Sarici
20/11/2017	1.2	Small adaptions	Andrea Meier
04/12/2017	1.3	Final Version	Andrea Meier Anthony Kalbermatten Cheng-Hsin WUU Deniz Sarici

1 Introduction

1.1 Purpose

This document represents the Software Requirements Specification (SRS) for the HTML Generator. It is designed and written for the stake holders, such as the teaching assistants, professors and the developers involved in the software project. Its purpose is to describe the scope of the system, both functional and non-functional requirements for software, design constraints and system interfaces.

1.2 Scope

The product described in this document is a HTML generator for creating a website with multiple linked pages with images, bulleted lists, tables, and so on. It will also be able to include existing HTML snippets in a document. For instance, an image map already stored in a file or in a string object.

Due to the elevated number of proposed functions, some of the specifications in Section 3 are missing. For the purpose of this (academic) document, we consider it to be more important to present a general view of a complex system. As opposed to restraining the SRS to a smaller number of limited functions.

After the reading of this document the reader should understand the concept behind the proposed software.

The most important goal of this software is to relieve the programmer from typing HTML code in a messy format, especially by eliminating the time for debugging wrong links or finding missing closing tags and identifying hard-to-read indentation.

1.3 Definitions, Acronyms and Abbreviations

API
Application Programming Interface. Allows other programs to access a set of functions and tools. An API does not necessarily require in-depth knowledge of programming.

Eiffel
An object-oriented programming language

EiffelStudio
IDE (Integrated development environment) for the programming language Eiffel.

Github
Web-based version control repository

HTML
Hypertext Markup Language, Standard markup language for creating web pages

Markdown
Simplified markup language that uses plain text for formatting.

Static web page
A web page that is delivered to the user as stored on the server. Thus all users get access to the same information.

Use case
List of actions required to accomplish a specified goal or task. Describes the interactions between actors and the system.

User
Any person interacting with the system. Following users are distinguished:

- Programming user: person creating a web page using the HTML/ Markdown generator
- Web user: person viewing a generated web page with a browser

1.4 References

Eiffel Software's Dual Licensing Business Model,
https://www.eiffel.com/eiffelstudio/licensing/
GNU License, https://www.eiffel.com/eiffelstudio/licensing/gpl/
HTML Reference, https://www.w3schools.com/tags/ref_byfunc.asp
HTML Text Formatting, https://www.w3schools.com/html/html_formatting.asp
Software Installation for EiffelStudio,
https://www.eiffel.org/doc/eiffelstudio/Software%20Installation%20for%20EiffelStudio
W3C Markup Validation Service, https://validator.w3.org/
Web Browser Image Format Support,
https://en.wikipedia.org/wiki/Comparison_of_web_browsers#Image_format_support

1.5 Overview

The rest of this SRS analyzes the detailed requirements of the HTML generator. The document is organized as follows:

Section 1 provides an overview of the software described in this document. The reading of this section is recommended since it helps to understand the basics and the purpose of this document. Some important definitions are also given.

Section 2 explores the product specification further, delineating the perspective of this product, the functions, and other general information.

Section 3 is a technical description of the functionality of this software. It is used to define without ambiguity the exact behavior of the desired function. It should be used as the information and specification base for the implementation.

The document is structured according to the IEEE 830-1998 standard [IEEE-830].

2 Overall Description

2.1 Product perspective

The HTML generator can not work on it's own. It is a library included in EiffelStudio. However, all functionalities are based in the HTML generator library. The interface is provided through EiffelStudio and user activities are handled over EiffelStudio. The HTML generator library is needed to generate the HTML file.

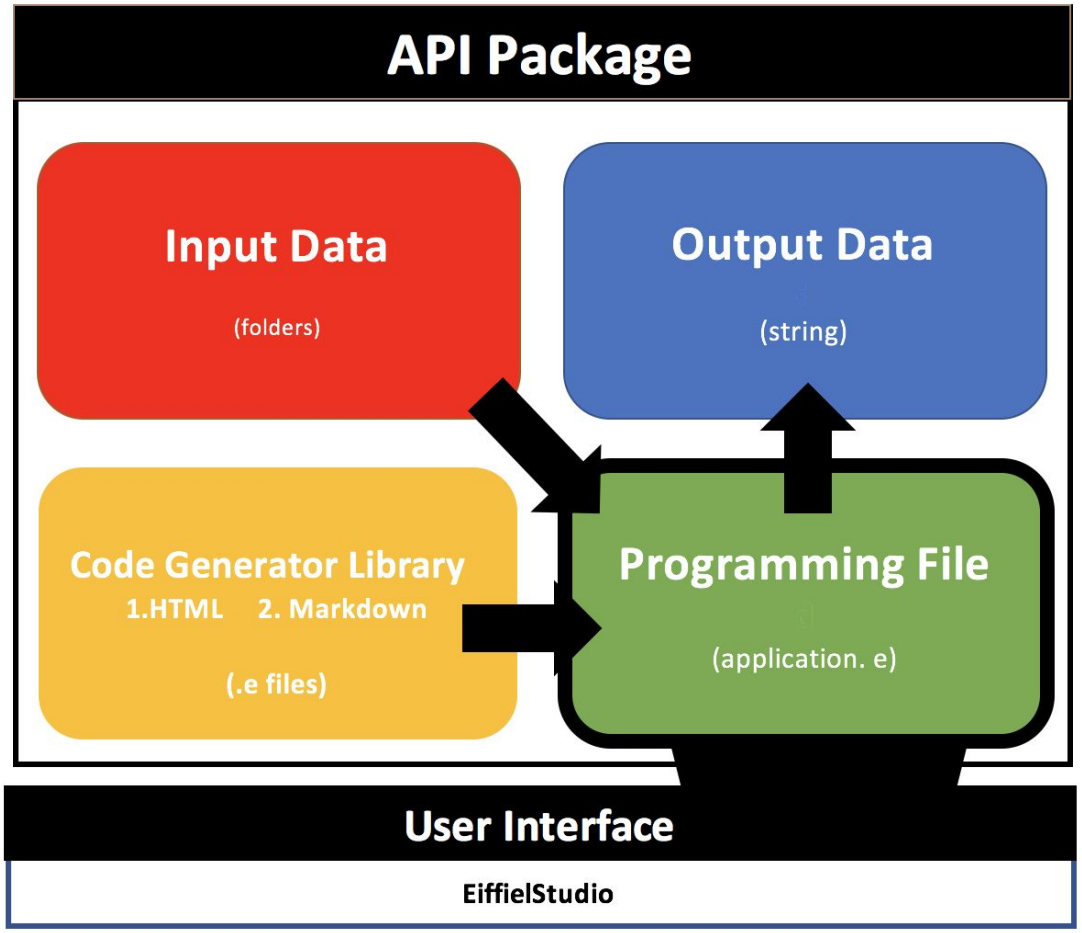

Figure 1. Product Overview

2.2 Product function

In this section we present all the functions the HTML generator library will provide. General actions include:

- create a paragraph
- define a title
- create a text with or without style
- insert a snippet
- insert an image
- create links (external and internal links)
- create list
- create a table

For more detailed information and secondary functions consult section 3.1.

2.3 User characteristics

The users of our systems are single users. The users are intended to handle EiffelStudio and know how to open the application.e file, which includes the classes and functions of the HTML code generator. He should be able to use the HTML code generator library without any specific HTML knowledge. However, the user is expected to apply the HTML code generator's library vocabulary. In addition to that the user should have computer experience that allow him, for example, to manage files or to verify an URL.

2.4 Constraints

As a student work SRS, this document might lack the correct terms used for describing information technology circumstances. Furthermore, we will only talk about possible solutions for Markup to be included without implementing any specific code.

2.5 Assumptions and dependencies

- The supported version of EiffelStudio is installed.
- The user creates the HTML code solely through the generator using EiffelStudio.
- The HTML meets the HTML5 standard. Except for the anchors.
- The snippet does not contain any header and as a result should be directly insertable in the body without causing any hierarchy violations.
- The user will not use HTML snippets except in the snippets function.
- The generated HTML page is not optimized for mobile browser.

3 Specific Requirements

This section contains all software requirements, both functional and non-functional. The functional requirements are grouped according to use cases. A requirement has the following properties:

Property	Description
Requirement ID	Uniquely identifies requirement within the SRS document
Title	Defines the functional group the requirement belongs to. Gives the requirement a symbolic name.
Description	The definition of the requirement
Priority	Defines the order in which requirements should be implemented. Priorities are designated (highest to lowest) "1", "2", and "3" … Requirements of priority 1 must be implemented in the first productive

	system release. The requirements of priority 2 and lower are subject of a special release agreement, which is out of scope of this document.
Risk	Specifies the risk of not implementing the requirement. It shows how the particular requirement is critical to the system. These are the following risk levels and their associated impact to the system if the requirement is not implemented or implemented incorrectly: • Critical (C) – will break the main functionality of the system. The system can not be used if this requirement is not implemented. • High (H) – will impact the main functionality of the system. Some functions of the system could be inaccessible, but the system can be generally used. • Low (L) – the system can be used without limitation, but with some workarounds needed.

3.1 Classes

In order to provide a clear and simple explanation an object oriented approach will be taken.. The purpose of the following classes is only to specify in a more formal approach the precise elements of the software and not to analyze the implementation details. For this reason the presented classes will only have attributes. We will describe the most important attributes of each class before continuing with the functional requirements.

3.1.1 Text

This class is the base class for all text inputs. The programming user can decide if the text should be bold, italic or underlined or any combination of them three. If the programming user makes no further specifications the text stays normal (not bold, italic or underlined). Objects of this class can be used in many other classes as input.

3.1.2 Paragraph

This class will help programming user to create paragraph with no length restriction.

3.1.2 Title

This class will help the programming user to create titles in 6 different sizes.

3.1.3 List

These classes will help the programming user to create either ordered or unordered lists.

3.1.5 Image

This class will help the programming user to display an image.

3.1.6 Snippet Insertion

This class allows programming user to include an existing HTML code inside the new HTML code. The programming user has the responsibility to remove the head and body tag from the inserted HTML code.

3.1.7 Table

This class allows programming user to create a table. The programming user needs to provide the number of columns. With the add function he can add a row with the content of the cells of this row. Every content of a cell is an other object by itself.

3.1.8 Link

These classes allow the programming user to have a link to another segment in the same HTML file or an external link to an web page or to another HTML file.

3.2 Functionality

This section describes the main functional requirements of the HTML Generator. The requirements are structured by functionality area and correspond in general to the user case.

3.2.1 General HTML

Property	Description
Requirement ID	R 3.2.1.01
Title	HTML Structure
Description	User can create HTML by putting different blocks in the file. Blocks refer to text, links, image, table, lists and snippets, and are treated as an object containing essential elements inside. (R 3.1)
Priority	1
Risk	C

Property	Description
Requirement ID	R 3.2.1.02

Title	Text
Description	Creates a text object which can have the following attributes in any combination (including no attribute). ● Bold text ● Underlined text ● Italic text This will be implemented as an object on its own as text is the base for many other functions.
Priority	1
Risk	H

3.2.1 Title

Property	Description
Requirement ID	R 3.2.1.01
Title	Title
Description	User can create a title with 6 different sizes.
Priority	2
Risk	H

3.2.3 Paragraph

Property	Description
Requirement ID	R 3.2.3.01
Title	Paragraph
Description	Create a paragraph, which allows the user to group multiple elements into one block. There cannot be a paragraph inside a paragraph.
Priority	2
Risk	H

3.2.4 Images

Property	Description

Requirement ID	R 3.2.4.01
Title	Image
Description	The programming user is able to add an image from an image file to the HTML File. The programming user has to ensure that the image is placed in the “img” folder (this folder has the same directory of the all HTML files). The programming user has to provide the image itself as well as a description of the image. The programming user is only allowed to place one image per line. The image formats supported rely on the browser of viewing web page, and reference can be found: https://en.wikipedia.org/wiki/Comparison_of_web_browsers#Image_format_support
Priority	1
Risk	C

3.2.5 Links

Property	Description
Requirement ID	R 3.2.5.01
Title	External Link
Description	The programming user will be able to create a link to another file or an external webpage. The programming user needs to provide the text, the link will be hidden behind and the path of the file or the url of the website the web user will be sent to.
Priority	1
Risk	C

Property	Description
Requirement ID	R 3.2.5.02
Title	Anchor
Description	The programming user can put an anchor wherever he wants to and create internal links with the same id as the anchor.
Priority	1

Risk	C

Property	Description
Requirement ID	R 3.2.5.03
Title	Anchor Link
Description	The programming user will be able to connect a link to an anchor within the same HTML page.
Priority	1
Risk	C

3.2.6 Lists

Property	Description
Requirement ID	R 3.2.6.01
Title	Unordered List
Description	The programming user will be able to create a bullet pointed list. The content of the rows can be added one by one by the add_row function.
Priority	1
Risk	C

Property	Description
Requirement ID	R 3.2.6.02
Title	Ordered List
Description	The programming user will be able to create a list with numerated elements.The content of the rows can be added one by one by the add_row function. The ordered List will always start with 1.
Priority	1
Risk	C

3.2.7 Tables

Property	Description
Requirement ID	R 3.2.7.01
Title	Tables
Description	The programming user will be able to create a table. The programming user needs to provide the number of columns. Afterwards he can add rows one by one. To create a row the programming user creates a list with all the elements, he wants as cells in this row. The list needs to have the same length as there are columns.
Priority	1
Risk	C

3.2.8 Snippets Insertion

Property	Description
Requirement ID	R 3.2.8.01
Title	Snippets Insertion
Description	The programming user will be able to include an existing HTML code inside the new HTML code. The programming user has the responsibility to remove the head and body tag from the inserted HTML code. Proper indentation is up to the user.
Priority	1
Risk	C

3.2.9 Markdown

Property	Description
Requirement ID	R 3.2.9.01
Title	Markdown

Description	Markdown could easily be implemented as there is a factory and already existing over classes.
Priority	3
Risk	H

3.3 Usability

Property	Description
Requirement ID	R 3.2.03
Title	Language
Description	All error messages are provided in English.
Priority	1
References	

Property	Description
Requirement ID	R 3.2.04
Title	Usability standards
Description	As we use the prevailing user interface of EiffelStudio, we will not discuss usability standards in this SRS document.
Priority	
References	

3.3 Extensibility

Property	Description
Requirement ID	R 3.3.01
Title	Extensibility

Description	Markdown could easily be implemented as there is a factory and already existing over classes
Priority	1
References	

3.4 Performance

Property	Description
Requirement ID	R 3.4.01
Title	Maximum Duration of Generation
Description	The HTML file for a single page will be generated in less than fifteen seconds.
Priority	2
References	

Property	Description
Requirement ID	R 3.4.02
Title	Storage Size of Generator Library
Description	The generator library files will not exceed the size of 0.2 Megabyte.
Priority	
References	

3.5 Maintainability

3.5.1 HTML Standard

Property	Description
Requirement ID	R 3.5.1.01

Title	HTML Standard
Description	All output shall be validatable by https://validator.w3.org Consequently, the knowledge of HTML and Eiffel is enough to maintain the software.
Priority	1
References	

3.5.2 Class libraries

Property	Description
Requirement ID	R 3.5.2.01
Title	Class libraries
Description	The class libraries shall be as general as possible. It should be easy to extend the software by additional markup languages.
Priority	1
References	

Property	Description
Requirement ID	R 3.5.2.02
Title	Bug Fixing
Description	“Critical bugs” are defined as errors that make the HTML file not interpretable or when a function does not work as specified in 3.1. “Non-critical bugs” are defined as errors that are only noticed by the programming user (e.g. bad indentation, compiler still displays website correctly). The time period from finding a critical bug until it is fixed should on average take no longer than one week.
Priority	1
References	R 3.2.

3.6 Design Constraints

3.6.1 Programming languages

Property	Description
Requirement ID	R 3.6.1.01
Title	Programming Language
Description	The programming language being used is Eiffel.

3.6.2 Platform constraints

Property	Description
Requirement ID	R 3.6.2.01
Title	Operating systems
Description	The platform constraints shall be limited only by the constraints imposed by EiffelStudio. EiffelStudio supports Windows (7, 8, 8.1 and 10), Mac OS X (10.12 or above), any Linux system with glibc 2.7 and GTK+ 2.4. Other supported operating systems are listed on the EiffelStudio installation guide.
Priority	1
Source	https://www.eiffel.org/doc/eiffelstudio/Software%20Installation%20for%20EiffelStudio
References	

Property	Description
Requirement ID	R 3.6.2.02
Title	Hardware requirements
Description	EiffelStudio requires 4 GB of RAM or higher and 1 GB space on the hard disk.

Priority	1
Source	https://www.eiffel.org/doc/eiffelstudio/Software%20Installation%20for%20EiffelStudio
References	

3.7 On-line User Documentation and Help System Requirements

Property	Description
Requirement ID	R 3.7.01
Title	Documentation
Description	The developers shall provide an online user documentation. The online documentation provides insight into what the different classes and functions do and how to get access to them. The documentation shall contain a table of contents and an index.
Priority	1
Risk	C
References	

3.8 Purchased Components

Property	Description
Requirement ID	R 3.8.01
Title	Interface
Description	As our HTML Generator will be Open Source and without commercial benefit involved, the Eiffel Software version is free for use according to Eiffel Software's Dual Licensing model for EiffelStudio. Only choosing Open Source Licensing for Eiffel Software is needed.
Priority	3

Source	
References	

3.9 Interfaces

3.9.1 User Interfaces

Property	Description
Requirement ID	R 3.9.01
Title	Interface
Description	The Interface provided for the user to create the HTML file will be EiffelStudio. Therefore, the only systems supported will be home computers or laptops.
Priority	1
Source	
References	2.1

3.9.2 Software Interfaces

Requirement ID	R 3.9.2.01
Title	Interface
Description	For our project, this aspect is out of scope but could be implemented with some additional work.
Priority	3
Source	
References	

3.9.3 Communications Interfaces

Requirement ID	R 3.9.3.01
Title	Interface
Description	For our project, this aspect is out of scope but could be implemented with some additional work.
Priority	3
Source	

3.10 Licensing Requirements

Property	Description
Requirement ID	R 3.10.01
Title	Licensing
Description	The HTML generator library will be available under the general public license. The license will be for free and everyone with the intention to use our program has to accept its terms. Everyone that copies, distributes or modifies the HTML generator library must distribute the software under the GNU general public license. If a modification of our library gets distributed, its full source code must be provided.
Source	https://www.eiffel.com/eiffelstudio/licensing/gpl/

3.11 Legal, Copyright, and Other Notices

Property	Description
Requirement ID	R 3.11.01
Title	Legal
Description	We do not own EiffelStudio and consider our usage of it as “fair use”. We will not be held responsible for any misuse of the HTML generator.