Contents

Revision History

Date	Version	Description	Author(s)
29.09.2017	0.1	Writing Purpose and Scope	Yulia Brun, Aleksandra Markovic, Andjela Markovic, Florian Imami
03.10.2017	0.2	Writing Introduction and Overall description	Yulia Brun, Aleksandra Markovic, Andjela Markovic, Florian Imami
08.10.2017	1.0	Writing Specific requirements, the first draft is ready	Yulia Brun, Aleksandra Markovic, Andjela Markovic, Florian Imami
11.10.2017	2.0	Some changes in Introduction, Overall description, Functionality and Usability. The second draft is ready	Yulia Brun, Aleksandra Markovic, Andjela Markovic, Florian Imami
12.10.2017	2.1	Adding Index	Yulia Brun, Aleksandra Markovic, Andjela Markovic, Florian Imami
13.10.2017	2.2	Some minor changes	Yulia Brun, Aleksandra Markovic, Andjela Markovic, Florian Imami
18.10.2017	3.0	Changed after the feedback from the TA	Yulia Brun, Aleksandra Markovic, Andjela Markovic,Nimra Ahmed
7.11.2017	4.0	Changed after the creation of the design document	Yulia Brun, Aleksandra Markovic, Andjela Markovic, Nimra Ahmed
25.11.2017	4.1	Minor changes after the implementation	Yulia Brun, Aleksandra Markovic, Andjela Markovic, Nimra Ahmed

1 Introduction

1.1 Purpose

The purpose of this document is to describe a detailed Software Requirement Specification (SRS) for a markup language generator specified for the HTML format. It is meant to outline concepts of the project. More specific, it defines the functional and non-functional requirements of the markup language generator and its design constraints. This document is addressed to the target audience of the project, such as software users, software developers, project managers, software documenters, software testers and domain experts.

1.2 Scope

The Markup Language (ML) Generator is a library, implemented in the Eiffel programming language, that creates a markup language document and handles appropriate functions which support the creation of valid markup language documents. The ML Generator will be specified for the HTML document format.

The ML Generator is meant to support the creation of markup language documents with linked pages, enable and facilitate inserting objects and give the possibility to include existing markup language snippets in a document. The ML Generator is extendable for other markup language elements and various output formats and can be reused for other higher-level applications.

The ML Generator is not purposed to do anything not mentioned in this scope, it will manage the creation of markup language documents but not also, for instance, parsing of input or the transition of markup language documents to a web server.

The project's goal is to provide an organised generation of static websites, which will be easy to read, extend, change or debug. Accordingly, software developers will benefit from it by reusability of the ML Generator which can be used as a sub-component for other more complex applications.

1.3 Definitions, Acronyms and Abbreviations

The following table explains the terms, acronyms and abbreviations used in this SRS document.

Term/ Acronym/ Abbreviation	Definition
SRS	Software Requirements Specification
ML Generator	Markup Language Generator
URL	Uniform Resource Locator - a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it.
Markup language	A system for annotating a document in a way that is syntactically distinguishable from the text.
HTML	Hypertext Markup Language - the standard markup language for creating web pages and web applications.
API	Application programming interface - a set of subroutine definitions, protocols, and tools for building application software.

Software user	A person that uses the software product and interacts with it.
Software developer	A person that develops a software computer system in a computer language, including the process of research, design and programming.
Project manager	A person that faces project planning, people and task management, conduction and implementation of an (engineering) project.
Software documentor	A person concerned with the documentation of a project, in our case, the team members of the project.
Software tester	A person that has the task of testing the software system and providing information according to its results and functionality.
Domain expert	A person with special knowledge or skills in the software domain area.
KLOC	Thousand Lines of Code
CPU	Central Processing Unit - basic unit of a computer that carries out instructions of a computer program by performing arithmetic, logical, control and input/output operations specified by the instructions.
String object	A sequence of characters.
Linked List	In computer science, a linked list is a linear collection of data elements, in which linear order is not given by their physical placement in memory. Instead, each element points to the next. It is a data structure consisting of a group of nodes which together represent a sequence.
Internal/External links	An internal link is a type of hyperlink on a webpage to another page or resource, such as an image or document, on the same website or domain. Hyperlinks are considered either "external" or "internal" depending on their target or destination. Generally, a link to a page outside the same domain or website is considered external, whereas one that points at another section of the same webpage or to another page of the same website or domain is considered internal.[1]
Data Storage System/DSS	Data Storage System – technology consisting of computer components and recording media that are used to retain digital data.[2]

1.4 References

The following table defines the list of all documents referenced in the SRS.

[1] https://en.wikipedia.org/wiki/Internal_link

[2] https://en.wikipedia.org/wiki/Computer_data_storage

Document title	Publishing organization	Link	Date
Project Description	Department of Informatics UZH		September 2017
SRS examples 1&2&3	Department of Informatics UZH		September 2017
IEEE Computer Society (1998)	The Institute of Electrical and Electronics Engineers (IEEE), New York		October 1998
IEEE Recommended Practice for Software Requirements Specifications.	The Institute of Electrical and Electronics Engineers (IEEE), New York		June 1998
ISO/IEC 9126-1:2001	International Organization for Standardization (ISO)		June 2001
MIT licence	MIT	https://opensource.org/licenses/MIT	State: 10 October 2017
Wikipedia articles	Wikipedia, The Free Encyclopedia	https://en.wikipedia.org	State: 10 October 2017

1.5 Overview

The rest of the SRS document describes the ML Generator requirements thorough and is organised thusly:

Section 2 describes the general factors that have an effect on the product and its requirements. This section includes the product perspective and functions, restrictions to be observed as other general information.

Section 3 defines functional and non-functional requirements of the ML Generator to a level of detail and precisely explains its behaviour. The primary function of this section is to provide a specified base of requirements to serve software designers for implementing the project and software testers due to verification of the project outcome.

Section 4 contains index.

2 Overall Description

2.1 Product perspective

The ML Generator is not a standalone application. In order to work, it should be supported by the API. It can be also integrated in more complex systems or used together with a third-party application which provides the ML Generator with an input. An example for such an application is a software application which calculates statistics. The ML Generator shall receive the statistic data from this application and create a string in the required markup language which would represent this input. The ML Generator returns a string as output and is able to save the generated document as a file.
We distinguish the API from a third-party application in the way that the API is essential for normal functioning of the ML Generator.

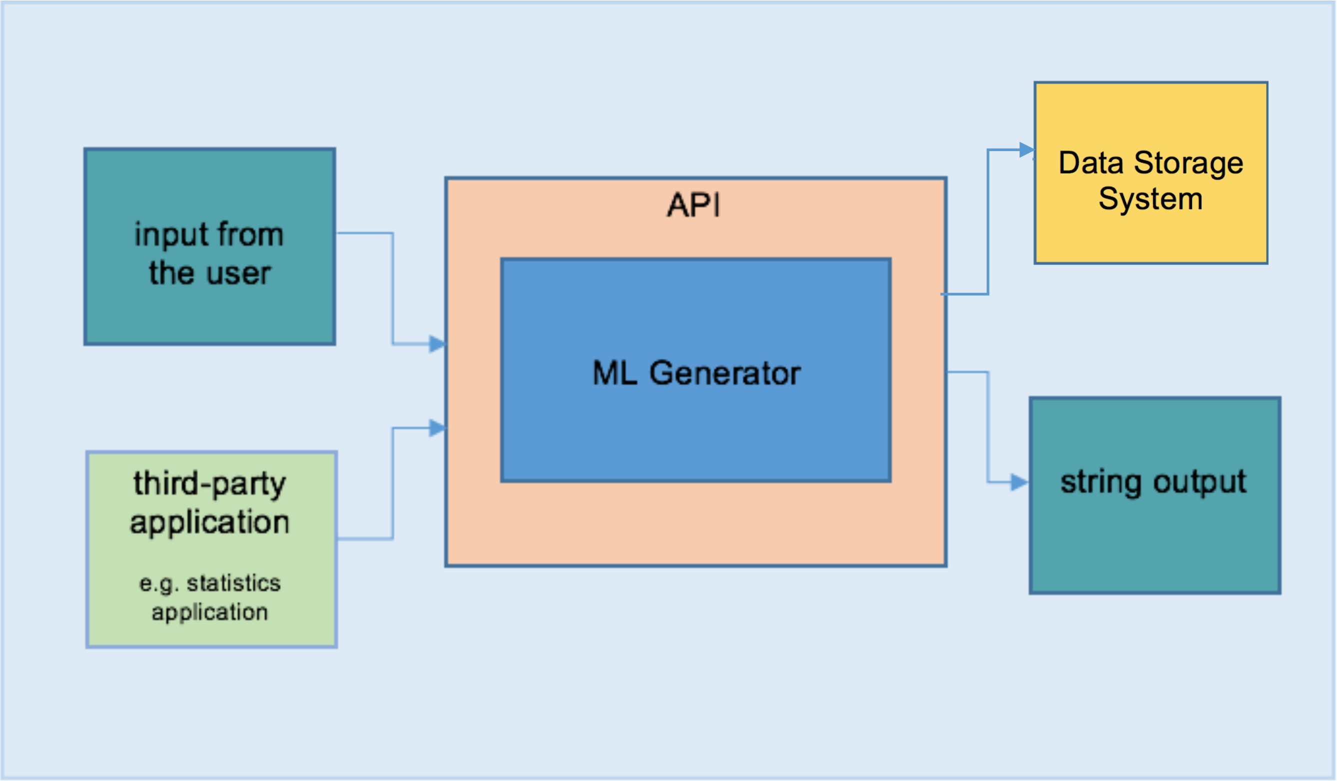

Figure 1: ML Generator's perspective

2.2 Project Functions

In this section we provide a general description of the functions, the ML Generator should fulfil. The purpose of this section is to present only the general functions. They will be explained more detailed in section 3.
The main function of the ML Generator is to create a valid markup language document, as mentioned earlier. This includes the following:

- Reading input elements from the API
- Handling URL generation
- Supporting multiple external and internal linked pages
- Supporting ordered and unordered lists
- Supporting tables
- Supporting page title and headings
- Including images and diagrams
- Including existing ML snippets already stored in a string object
- Maintaining correct ML tag naming
- Supporting paragraphs and correct indentation
- Supporting text semantics
- Displaying the generated ML document in the API
- Saving the output document in the Data Storage System

The ML generator has the following features:

- Supporting extension of the library to more ML elements
- Supporting extension of the library to function for other markup languages
- Supporting the integration of the application into more complex systems

The ML Generator does not support the following functions:

- Parsing the Input
- Modifying the input data
- Importing the ML document
- Distributing the ML document
- Transforming a document into another document format

2.3 User characteristics

The ML Generator is meant to be used by programmers, that have the intention to generate ML documents, with correct tag naming and linking. In order to use the ML Generator, users should have a basic programming knowledge in the Eiffel programming language and be able to perform simple programming tasks, like using classes from libraries.
The programmers will benefit from the direct use of the ML Generator and from the extendibility and reusability of it.

2.4 Constraints

This document does not represent a real-life SRS, but a study project. The document gives only requirement templates for creating a software project. Other constraints are based on the lack of communication during the writing of the different parts of this document and possible different SRS standards that were used producing aforesaid document.

2.5 Assumptions and dependencies

2.5.1 Data Storage System

The ML-Generator is capable of saving the generated documents on the Data Storage System, by request of the user. There are no specific requirements for a Data Storage System. We assume that it has enough free storage space to store generated documents.

2.5.2 Library working condition

In order to work, our library should be supported by API or integrated in a more complex program.
If the ML Generator is used together with a third-party application which provides it with an input, we assume that the input is in the suitable format.

3 Specific Requirements

In this section the specific software requirements (functional and non-functional) will be discussed in detail.
A requirement will be described using the following properties[3]:

Requirement ID	Defines the unique symbolic representation name for a requirement in accordance with the specific number of the requirement, and the number of the functional group the requirement belongs to.
Title	Presents a descriptive name for the requirement.
Description	Presents the definition of the requirement.

[3] The requirement properties are based on *Example2.*

Priority	Defines the order in which the requirement should be implemented. Priorities are designated (highest to lowest) 1, 2 and 3 … A requirement of the priority 1 must be implemented first. A requirement of priority 2 is mandatory for the final implementation. A requirement of priority 3 or greater represents optional features.
Source	In a real-time SRS it refers to the source the requirement originates from.
Risk	Specifies the risk of not implementing the requirement. It shows how critical the requirement is to the system. The following risk levels are defined over the impact of not being implemented correctly. • 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. • 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	Lists the related requirements.

3.1 Functionality

This section describes the functional requirements of the ML Generator. The section will be structured by features and capabilities of the ML Generator.

3.1.1 Features

3.1.1.1 Input features

Requirement ID	R 3.1.1.1 001
Title	Input\Reading input data
Description	The ML Generator shall support reading the input data from the API, when the user of the library provides the input.
Priority	1
Source	
Risk	H
References	

Requirement ID	R 3.1.1.1 002
Title	Input\Choosing the document format
Description	The ML Generator shall support giving the user the possibility to choose the document format he wants to generate. This happens over the API.
Priority	2
Source	
Risk	H
References	

Requirement ID	R 3.1.1.1 003
Title	Input\Input data formalization
Description	The ML Generator shall support reading the input data as function input elements.
Priority	1

Source	
Risk	H
References	R 3.1.1.1 001

Requirement ID	R 3.1.1.1 004
Title	Input\Interpreting the functionality of input elements
Description	The ML Generator shall support interpreting the input as arguments for the functions.
Priority	1
Source	
Risk	H
References	R 3.1.1.1 003

Requirement ID	R 3.1.1.1 005
Title	Input\Input overflow
Description	If a provided input consists of more than 500 characters (letters, digits, whitespace and punctuation marks), the API shall display an error message about an input overflow and abort the operation of adding that input element to the document.
Priority	2
Source	
Risk	M
References	

Requirement ID	R 3.1.1.1 006
Title	Input\Empty or wrong input
Description	If the input for a function contains no value or a wrong input value, the API shall display an appropriate error message.
Priority	2
Source	
Risk	C
References	R 3.1.1.1 003

Requirement ID	R 3.1.1.1 007
Title	Input\Document format
Description	If the user intends to use a document format which is not supported by the library or no document format is provided, an error message is raised in the API.
Priority	2
Source	
Risk	H
References	R 3.1.1.1 002

Requirement ID	R 3.1.1.1 008
Title	Input\Storing the input
Description	During the runtime the ML Generator shall store all input elements.
Priority	1
Source	
Risk	H
References	

3.1.1.2 Formatting features

Requirement ID	R 3.1.1.2 001
Title	Formatting features\Document
Description	The ML Generator shall support creating a document (that represents a page) with a name and page title in supported markup language formats. The user shall be given the possibility to chose the page title and name. Page title and name shall be provided as strings.
Priority	1
Source	
Risk	H
References	

Requirement ID	R 3.1.1.2 002
Group	Formatting features\Overwriting
Description	If the user commands to create an object under the same name it already exists, the characteristics of this object get overwritten. This does not work for tables or lists. The user will have to change the input in the call.
Priority	1
Source	
Risk	C
References	R 3.1.1.2 001

Requirement ID	R 3.1.1.2 003
Title	Formatting features\Headings
Description	The ML Generator shall support creating headings of different sizes in supported markup language formats. These headings contain only a string. The user choses the size of the heading by providing a number in form of a integer which can’t be larger than 6.
Priority	1
Source	
Risk	M
References	

Requirement ID	R 3.1.1.2 004
Group	Formatting features\Text
Description	The ML Generator shall support creating a text object, in form of a string, which can be either normal, italic, bold or underlined.
Priority	1
Source	
Risk	C
References	

Requirement ID	R 3.1.1.2 005
Title	Formatting features\URL generation
Description	The ML Generator shall be able to generate an URL.
Priority	1
Source	
Risk	C
References	

Requirement ID	R 3.1.1.2 006
Title	Formatting features\Internal linked pages
Description	The ML Generator shall support the installation of multiple internal links into the output. The user provides the to be linked document as the input as well as the desired link name written as a string.
Priority	1
Source	
Risk	C
References	R 3.1.1.2 005

Requirement ID	R 3.1.1.2 007
Title	Formatting features\External linked pages
Description	The ML Generator shall support the installation of multiple external links into the output which represents the document that is being generated. The user provides the links as strings and the word which should contain the link.
Priority	1
Source	
Risk	C
References	R 3.1.1.2 005

Requirement ID	R 3.1.1.2 008
Title	Formatting features\Unordered and ordered lists
Description	The ML Generator shall support the creation of unordered and ordered lists in supported markup languages. Lists shall be able to contain all objects including itself.
Priority	1
Source	
Risk	M
References	

Requirement ID	R 3.1.1.2 009
Title	Formatting features\Tables
Description	The ML Generator shall support the creation of tables in by the library supported markup languages. Tables shall be able to contain all objects including itself.
Priority	1
Source	
Risk	M
References	

Requirement ID	R 3.1.1.2 010
Title	Formatting features\Images
Description	The ML Generator shall support the insertion of an image (also diagrams as images), in by the library supported markup languages, in the string output that represents the generated document. The generator shall also support creating a description for an image that is displayed if the image is not available. The user shall provide the description and image link as a string.
Priority	1
Source	
Risk	M
References	

Requirement ID	R 3.1.1.2 011
Title	Formatting features\Existing ML snippets
Description	The ML Generator shall be able to insert already existing snippets into the output string object. They shall be input as strings.
Priority	1
Source	
Risk	M
References	

Requirement ID	R 3.1.1.2 012
Title	Formatting features\Paragraphs
Description	The ML Generator shall support the creation of paragraphs in the library supported markup languages. Paragraphs shall only contain text. The ML Generator shall also support adding text to the paragraph in different formatting (normal, bold, italic, underline).
Priority	1
Source	
Risk	C
References	

3.1.1.3 Output features

Requirement ID	R 3.1.1.3 001
Title	Output\Displaying the output document
Description	The user shall be able to output the generated ML document. The document shall be displayed as a string in the API.
Priority	1
Source	
Risk	C
References	

Requirement ID	R 3.1.1.3 002
Title	Output\Saving the output document
Description	The user shall be able to save the generated ML document as a file.
Priority	1
Source	
Risk	C
References	

Requirement ID	R 3.1.1.3 003
Title	Output\Saving the output document\File name
Description	The ML Generator shall provide the user with the possibility to name the output file he wants to save. If the user doesn't provide a file name a default file name shall be used. If the user provides a file name that is already used for another file, it shall be replaced.
Priority	1
Source	
Risk	H
References	R 3.1.1.3 002

3.1.2 Maintenance functionality

The ML Generator is an extendible software and therefore open to updates. It shall be possible to update the software manually, which shall be done by experienced programmers. However, providing updates and patches for the library is out of scope for this project, so updates and their installation won't be considered further.

3.1.2.1 Displaying

Requirement ID	R 3.1.2.1 001
Title	Maintenance\Displaying
Description	The ML Generator shall provide the user with the ability to display the so far generated document, any time, as a string object. The output shall be displayed in the API.
Priority	1
Source	
Risk	L
References	

3.2 Usability

This section includes all requirements that affect usability.

3.2.1 Application Programming Interface

Requirement ID	R 3.2.1.001
Title	Usability\API
Description	The library conforms to the API standards pertinently to the Eiffel Studio programming environment. Input and output are provided there due to testing of the library.
Priority	1
Source	
Risk	H
References	R 3.1.1.1.001

3.2.2 Training

Requirement ID	R 3.2.2.001
Title	Usability\User Training
Description	The proficient software user should be able to use the library efficient, in the role: Software developer - after 2 days of training Project Manager - after 1 day of training Software documenter - after 1 day of training Software tester - after a ½ day of training
Priority	1
Source	
References	

Requirement ID	R 3.2.2.002
Title	Usability\Documentation

Description	The software documentation shall supply the software user with adequate information to apprehend the abilities and limitations of the library and directly use the library’s basic features. The documentation provides descriptions of all implemented functionalities of the software.
Priority	1
Source	
References	

3.2.3 *Task Times*

Requirement ID	R 3.2.3.001
Title	Usability\Task times
Description	The following tasks must be done within the specified time by the software user under the condition that the user has completed the appropriate training. Providing text inputs to the ML Generator software user 10min Implementing objects to the ML Generator software user 5min Displaying the output of the ML Generator software user/tester 1min
Priority	1
Source	
References	

3.2.4 *Language*

Requirement ID	R 3.2.4.001
Title	Usability\Language
Description	All software documentation and error messages must be in English.
Priority	1
Source	
References	

Requirement ID	R 3.2.3.002
Title	Usability\Localisation
Description	Due to local usage of the library, it is designed in a way to be extendable easily and changeable for integration in other software systems or applications. The integration of the library in other software systems is out of scope of this document and shall be defined elsewhere.
Priority	1
Source	
References	

3.3 Reliability

3.3.1 *Availability*

Requirement ID	R 3.3.1.001
Title	Reliability\Availability
Description	The ML Generator is available for use if the software system is running. Distribution and further implementation of the library is possible in all time zones.
Priority	1

Source	
References	

Requirement ID	R 3.3.1.002
Title	Reliability\MTTR
Description	The Mean Time to Repair (MTTR) shall not succeed 8 hours.
Priority	1
Source	
References	

3.3.2 *Accuracy*

Requirement ID	R 3.3.2.001
Title	Reliability\Accuracy
Description	The output requires 70% precision by the HTML standard for supported functions.
Priority	1
Source	
References	

3.3.3 *Defect Rate*

Requirement ID	R 3.3.3.001
Title	Reliability\Maximum Bugs
Description	The maximal number of bugs in the library shall not overrun 20 bugs in thousand lines of code (20/KLOC).
Priority	1
Source	
References	

Requirement ID	R 3.3.3.002
Title	Reliability\Bugs
Description	Accruing bugs can be categorized in the following three groups of bugs: ➢ *Critical bugs* are defined as errors which lead to complete inability of particular or several functions the library shall perform. A critical bug prevents the program from further testing of the product and has high priority of solving. Risk: H ➢ *Significant bugs* are defined as errors with medium severity, they impact smaller parts of the library which lead to an outcome that does not result as wished/expected, they do not completely disable functionalities of the library. Risk: M ➢ *Minor bugs* are defined as syntax errors that are of no relevance to the library's functionalities, they have no effect on the library's general performance. Risk: L
Priority	1
Source	
References	

3.4 Performance

3.4.1 *Number of simultaneous users*

Requirement ID	R 3.4.1 001
Title	Performance\Number of simultaneous users
Description	There shall be only one user of the software at a time.
Priority	1
Source	
Risk	C
References	

3.4.2 *Response time*

Requirement ID	R 3.4.2 001
Title	Performance\Response time
Description	Under the condition that the hardware fulfils the requirement R 3.9.2 001 and the input consists of one image, one table and a half page of text, the application shall have the following average response time shall be 30 seconds. The maximum response time shall not exceed 1 minute.
Priority	1
Source	
Risk	H
References	R 3.9.2 001

3.4.3 *Data Storage System*

Requirement ID	R 3.4.3 001
Title	Performance\Data Storage System
Description	The DSS shall have enough free storage space to save the output document.
Priority	1
Source	
Risk	C
References	R 3.1.1.3 002; R 3.1.1.3 003; R 3.1.1.3 004; 3.1.2.2 001; R 3.9.2 002

3.5 Maintainability

3.5.1 *Change of developers*

In case the developer team or company, that uses our library, should change after the 1rst release of the software, the new developers shall be able to begin working on the software in less than 3 days, if instructed by the old team. We assume that also the new developers have a basic programming knowledge, like the previous ones.

3.6 Design Constraints

There is no particular design constraint imposed, except that the output gets printed in the API. The user shall be able to write his input in the API, which shall give him the right output after the program generated a string. Therefore, the ML Generator will not contain built-in input. The library's API shall support creating a string, that represents a valid ML document.

3.7 On-Line User Documentation and Help System Requirements

In the programming code of the library, the function of every class will be explained. The explanations are included as comments in the code and shall describe how to use the functions properly. This method shall make the functions simple to understand, hence the programmer that uses the ML Generator can look out for them in the code.

3.8 Purchased Components

The interaction between the user and the library shall take place over the API. Output and input will be provided there. The architecture of the library's API will be created by the project team, and hence no component of it will be purchased.

3.9 Interfaces

In this section we describe interfaces which are relevant for the ML Generator.

3.9.1 User Interfaces

There is no special user interface in this application. The user interacts with it by modifying the program code. The application shall automatically display the output in the API.

3.9.2 Hardware Interfaces

These following requirements outline the minimal system configuration that the hardware must fulfil, in order to run the ML Generator.

Requirement ID	R 3.9.2 001
Title	Interfaces\Hardware interface
Description	The hardware interface shall consist of CPU, display and keyboard. The minimum requirements that the hardware shall fulfil: Processor: 1.3 GHz Intel Core i5 Memory: 4 GB 1600 MHz DDR3
Priority	1
Source	
Risk	C
References	Performance 3.4

Requirement ID	R 3.9.2 002
Title	Interfaces\Hardware interface\Data Storage System
Description	The hardware requirement of DSS is out-of-scope. We assume that the DSS's performance enables saving of the output document.
Priority	1
Source	
Risk	M
References	Performance 3.4

3.9.3 Software Interfaces

Requirement ID	R 3.9.3 001
Title	Interfaces\Software interface
Description	The third-party application shall provide the software with an input. Further requirements are out-of-scope of this document.
Priority	1
Source	
Risk	H

References	

3.9.4 *Communications Interfaces*

There is no communications interface for now. Future versions of the application may have communications interfaces for networks. Specific requirements for this communications interfaces are out-of-scope of this document.

3.10 Licensing Requirements

There are no licensing requirements for this software application.

3.11 Legal, Copyright, and Other Notices

3.12 Applicable Standards

Requirement ID	R 3.12 001
Title	Standards
Description	ISO 9126:2001 should be applied to the application. It must be explicitly named in the documentation.
Priority	1
Source	
Risk	
References	

4 Supporting Information

4.1 Index

[4] The copyright notice is based on the MIT license. Link: https://opensource.org/licenses/MIT