

Software Requirements Specification

Table of Contents

- 1. Introduction.....3
 - 1.1 Purpose.....3
 - 1.2 Scope.....3
 - 1.3 Definitions.....3
 - 1.4 References.....3
 - 1.5 Overview.....3
- 2. Overall Description.....5
 - 2.1 Product Perspective.....5
 - 2.1.1 User Interfaces.....5
 - 2.1.2 Hardware Interfaces.....5
 - 2.1.3 Software Interfaces.....5
 - 2.2 Product Functions.....5
 - 2.3 User Characteristics.....6
 - 2.4 Constrains.....6
 - 2.5 Assumptions and dependencies.....6
- 3. Specific Requirements.....7
 - 3.1 External Interfaces.....7
 - 3.1.1 User Interfaces.....7
 - 3.1.2 Hardware Interfaces.....8
 - 3.1.3 Software Interfaces.....8
 - 3.2 Functional Requirements.....8
 - 3.2.1 Functional Requirement 1.....8
 - 3.2.2 Functional Requirement 2.....8
 - 3.2.3 Functional Requirement 3.....9
 - 3.2.4 Functional Requirement 4.....9
 - 3.2.5 Functional Requirement 5.....9
 - 3.2.6 Functional Requirement 6.....9
 - 3.2.7 Functional Requirement 7.....9
 - 3.2.8 Functional Requirement 8.....9
 - 3.2.9 Functional Requirement 9.....9
 - 3.2.10 Functional Requirement 10.....9
 - 3.2.11 Functional Requirement 11.....9
 - 3.2.12 Functional Requirement 12.....10
 - 3.2.13 Functional Requirement 13.....10
 - 3.2.14 Functional Requirement 14.....10

- 3.2.15 Functional Requirement 15.....10
- 3.2.16 Functional Requirement 16.....10
- 3.3 Non-functional Requirements.....10
 - 3.3.1 Logical Database Requirements.....10
 - 3.3.1.1 Snippet Table.....10
 - 3.3.1.2 Tag Table.....10
 - 3.3.2 Portability.....11

1. Introduction

1.1 Purpose

The purpose of this document is to definitively set out the requirements for the “Code Snippet Manager” program. It will act as a plan for the development of the system. The intended audience for this document is the customer for their approval of the system and the development team so they can build a solution that will fulfil the customers requirements.

1.2 Scope

The “Code Snippet Manager” is a desktop application that will manage and organise code snippets for later use. The system will include editing tools for the aforementioned code snippets.

1.3 Definitions

Term	Definition
User	The individual who will make use of the system
Snippet	Small section of computer code
Tag	A keyword attached to a snippet to make identifying and searching for the snippet easier

1.4 References

[1] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998

[2] S Geagea et al, “Software Requirement Specification - Amazing Lunch Indicator” *University of Gothenburg Chalmers*, 2010. [Online] Available: http://www.cse.chalmers.se/~feldt/courses/regeng/examples/srs_example_2010_group2.pdf [Accessed: 13 March 2018].

1.5 Overview

The remaining two sections of this document provide an “Overall Description” of the system and provide a series of “Specific Requirements” that the system must

fulfil. The “Overall Description” section is intended primary for the describing the system to the user/customer whilst the “Specific Requirements” are intended primary for the developers of the system.

2. Overall Description

2.1 Product Perspective

The system is individual, self-contained and does not fit into any larger system.

2.1.1 User Interfaces

The user interface will provide a facility for the user to create snippet entries. Each snippet will contain a small section of reusable code. Various pieces of metadata will be stored about each snippet including a name, short description, set of keywords/tags, and a programming language. The user interface will allow the user to modify this metadata. The user interface will organise the code snippets and present them to the user. It will also allow the user to create, store, delete, view, edit and search these snippets.

2.1.2 Hardware Interfaces

The system will need to make use of the keyboard, mouse and display. The system also requires data to be stored permanently on the user's backing storage. This will be achieved through the use of a database package.

2.1.3 Software Interfaces

Due to the systems reliance on the user's backing storage, a database package will be employed. A database package will provide more organised and structured data management which in turn will provide faster and more efficient data retrieval. The database management system that will be employed is H2 Database Engine version 1.4.196.

2.2 Product Functions

The system will allow users to organise a collection of code snippets, allowing them to create, store, delete, view, edit and search for small sections of code. Each snippet will be associated with a name, short description, set of keywords/tags, and a programming language. This metadata and the snippet itself will be modifiable after its creation. Text editing tools will be included for modifying the text, finding and replacing text and highlighting text. This editing tool should- also provide syntax highlighting. The snippet and associated metadata will be stored permanently in a database.

2.3 User Characteristics

The intended audience of the system is considered to be software developers. They are considered to have a high level of education and experience in software development and programming and considered to have considerable technical expertise.

2.4 Constrains

The intended audience of the system is considered to be software developers who typically have fairly high performance computer systems, hence the system is not constrained by memory.

2.5 Assumptions and dependencies

The user of the software is assumed to have a high performance system, so little consideration will be given to the memory consumption of the system. Therefore if the user does not possess a high performance system then they may encounter issues using the system.

3. Specific Requirements

3.1 External Interfaces

3.1.1 User Interfaces

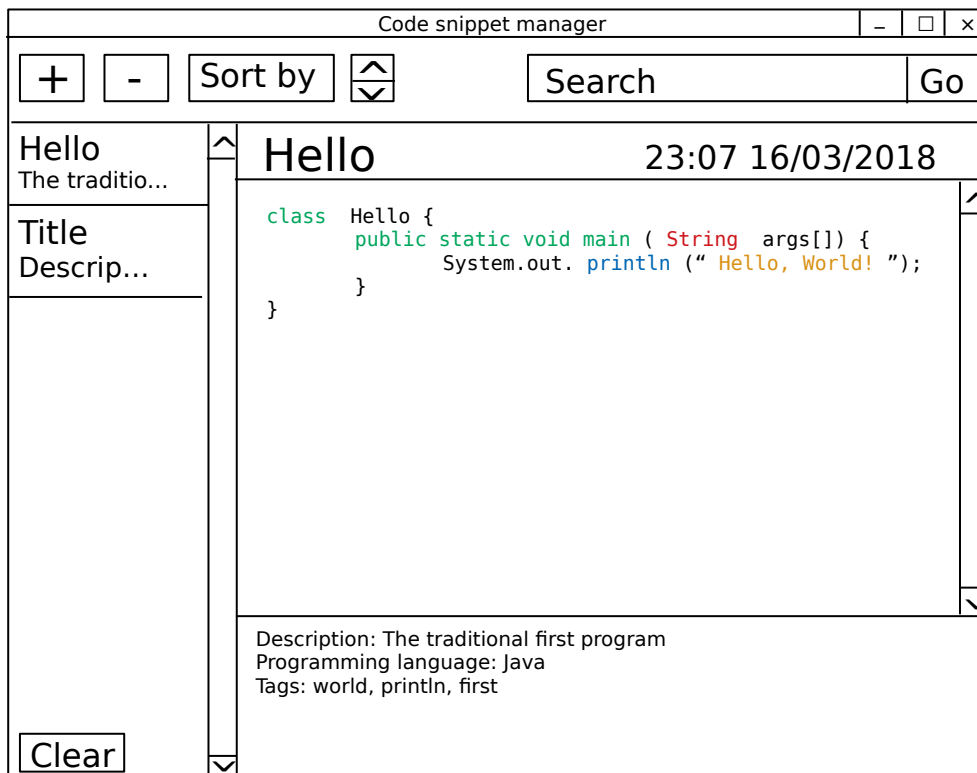


Figure 1

The user interface will consist of three panes, as shown in Figure 1 (above). The first pane, down the left side of the window will be populated with the title and description of each code snippet. The second pane will take up most of the right side of the window and will show the code snippet, above this the title of the snippet will be shown, along with the date the snippet was created. Underneath this pane, the third pane will contain show all the meta data related to the snippet including its description, programming language and any associated tags. Across the top of the window there will be a search bar which will allow the user to filter the snippets shown in the left pane. There will be a clear button underneath the left pane that will allow the user to clear the search and return

all snippets to the left pane. This button will be disabled when no search has been made. There will also be a drop down list to select how this pane should be sorted, and controls to add or delete code snippets.

3.1.2 Hardware Interfaces

The system will need to make use of the keyboard, mouse and display of the computer system that it is run on, however these resources will be managed by the underlying operating system and standard libraries. The system also needs to store data permanently on the backing storage of the user’s computer system, a software database package will be employed to manage this resource.

3.1.3 Software Interfaces

The system will make use of the H2 database management system package to manage the data that the system stores on the user’s computer system. This database will be embedded within the system. Figure 2 (below) shows an Enhanced Entity Relationship (EER) diagram for the database.

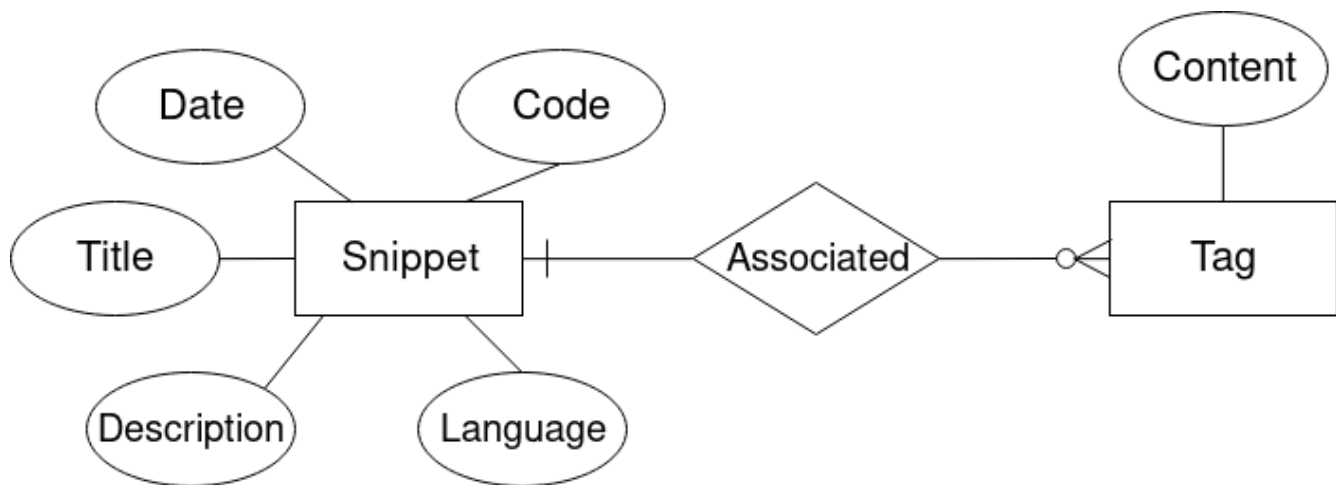


Figure 2

3.2 Functional Requirements

3.2.1 Functional Requirement 1

The system shall allow users to create code snippets by clicking the new snippet button.

3.2.2 Functional Requirement 2

Each snippet will have associated with it a title, description, programming language, date created and a set of tags.

3.2.3 Functional Requirement 3

The system will automatically set the date created field, but all other metadata fields and the code snippet itself must be set by the user.

3.2.4 Functional Requirement 4

The data must be stored permanently in a database.

3.2.5 Functional Requirement 5

All the code snippets must be displayed in the left most pane of the user interfaces, unless a search term has been entered.

3.2.6 Functional Requirement 6

The left most pane will be sorted by name, date created or programming language in ascending or descending order.

3.2.7 Functional Requirement 7

A drop-down list will be provided so the user can select which sorting option they want to use. Two buttons will be provided so the user can choose between ascending or descending order.

3.2.8 Functional Requirement 8

The user must be able to search for code snippet by title, description, programming language or tag. The results of the search will be displayed in the left most pane. The user will be able to clear the search by clicking a button.

3.2.9 Functional Requirement 9

The user will be able to select a snippet by clicking it in the left most pane.

3.2.10 Functional Requirement 10

Once selected, a snippet can be deleted by clicking the delete button.

3.2.11 Functional Requirement 11

Once selected, the title of the snippet along with the date it was created will be displayed above the upper right pane. The user will be able to edit the title, and any changes will be saved automatically, however the system will not allow the user to leave the title blank. The date of creation field will not be mutable.

3.2.12 Functional Requirement 12

Once selected, the code of the snippet will be displayed in the upper right pane.

3.2.13 Functional Requirement 13

Code in the upper right pane will be editable, and changes will be saved automatically.

3.2.14 Functional Requirement 14

The code in the upper right pane will be highlighted according to the syntax of the programming language.

3.2.15 Functional Requirement 15

Syntax specifications for each supported programming language will be supplied in text files. Users will be able to create and use their own specifications.

3.2.16 Functional Requirement 16

The description, programming language and any tags will be displayed in the lower right pane. These details will be editable by the user, and any changes will be saved automatically.

3.3 Non-functional Requirements

3.3.1 Logical Database Requirements

3.3.1.1 Snippet Table

Attribute	Data Type
Title	VARCHAR(40)
Code	CLOB
Date Created	TIMESTAMP

Language	VARCHAR(40)
Description	CLOB

3.3.1.2 Tag Table

Attribute	Data Type
Content	VARCHAR(30)

3.3.2 Portability

The system must be compatible with Windows, Linux and Macintosh OS X as these are the most popular development platforms.