# **Software Requirements Specification**
for
Sorting Algorithms Visualization

Prepared by:

Nilusche Liyanaarachchi

FH Aachen University of Applied Sciences, Germany
Software Engineering

Contents

Chapter 1

Introduction

1.1 Disclaimer

This document is prepared or accomplished by Nilusche Liyanaarachchi (Student Assistant) in his own personal capacity.

This fictional product exists for educational grounds and is not intended for commercial usage.

1.2 Purpose

The purpose of this document is to present a detailed description of the requirements and features of the Visualization Applet
The software is used to learn about Asynchronous JavaScript and to refresh sorting algorithms.

1.3 Intended Audience

This document does not have a particular audience in mind. Ideally this document should address fellow software engineers that have experience in reading/writing Software Requirements Specifications (SRS).
The Software is intended to explain Sorting Algorithms to Students using visualizaton.

1.4 Product Scope

The purpose of this clone is to provide the essential functionalities of an Visualization Applet and above all to ease learning about Sorting Algorithms for all kinds of users.

1.5 Definitions, Acronyms, and Abbreviations

Term/ Acronym / Abbreviation	Expansion / Description
Sorting Algorithm	SA
GUI	Graphical User Interface

1.6 References

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

Chapter 2

Overall Description

2.1 Product Perspective

2.1.1 System Interface

Since this product will only have a web-interface, appropriate Web-Browsers like Firefox, Chrome, Edge should be used in their latest stable version.

2.1.2 User Interface

The application GUI provides menus, buttons, containers, grids allowing for easy control by a mouse

2.1.3 Operating Environment

- Operating System: Windows
- JavaScript ES6 capable Browser

2.2 Product functions

- Generate random arrays
- Sort random arrays using different sorting algorithms
- Adjusting sorting speed

Product function are refined in Section 3. **Functional Requirements**

2.3 Constraints

The application is developed without any frameworks and only using HTML, CSS and Vanilla JavaScript ES6

Chapter 3

System Features and Requirements

3.1 Functional Requirements

These specific functional requirements will be presented as User story and Use case to guide the developer during their implementation.

3.1.1 Generate new Array

Title: Generate new Array	**Priority: high**
As a general user I want to generate new random arrays so that i can test the sorting algorithms on them	

Actors: User
Preconditions:

1. None

Trigger: Clicking the menu "Generate new Array"
Procedure:

1. User on the Button "Generate new Array"
2. System generates random array
3. System visualizes random array

Postconditions:

1. A new random Array is visualized on the site

Exceptions:

1. None

3.1.2 Presort Arrays to ascending or descending

Title: Presort Arrays to ascending or descending	**Priority: high**
As a general user I want to presort my array to descending or ascending ones so that i can test stability of sorting algorithms	

Actors: User
Preconditions:

1. None

Trigger: Clicking the menu "Set Values to ascending" or "Set Values to descending"
Procedure:

1. User on the Button "Set Values to ascending" or "Set Values to descending"
2. System generates a sorted array
3. System visualizes random array

Postconditions:

1. A new sorted Array is visualized on the site

Exceptions:

1. None

3.1.3 Pick different Sorting Algorithms

Title: Pick different Sorting Algorithms	**Priority: high**
As a general user I want to pick different kinds of sorting algorithms so that i can see how they are visualized	

Actors: User
Preconditions:

1. None

Trigger: Clicking any Sorting Algorithm Button
Procedure:

1. User selects Sorting Algorithm
2. System starts selected sorting algorithm
3. System visualizes every significant step of sorting algorithm using different colors

Postconditions:

1. The array is sorting or is sorted after clicking the Button

Exceptions:

1. None

3.1.4 Choose Sorting Speed

Title: Choose Sorting Speed	**Priority: medium**
As a general user I want to pick different kinds of sorting speeds so that i can slow down or increase the visualization speed	

Actors: User
Preconditions:

1. Sorting Algorithms needs to be already in the sorting stage

Trigger: Adjusting the speed slider
Procedure:

1. User adjust the the speed slider
2. System calculates speed
3. System applies speed/intervals to the sorting visualization

Postconditions:

1. The sorting process is either slowed down of sped up

Exceptions:

1. None

3.2 Non Functional Requirements

3.2.1 Performance Requirements

1. The service needs to be highly available
2. Acceptable latency to generate the timeline is 200ms

3.2.2 Software Quality Attributes

- Availability: The Sorting Algorithms need to be immediately sorted after clicking the button
- Correctness: The Applet should ensure a algorithmics correctness
- Usability: The interface should be easy to learn without a tutorial and allow users to accomplish their goals without errors.

3.3 Extended Requirements

1. Graph Algorithms Visualization
2. Replay feature

3.4 External Interface Requirements

3.4.1 Hardware Interfaces

- A browser which supports HTML5 and JavaScript ES6

Appendices

Appendix A

Appendix: Sample Screenshots of Product as is

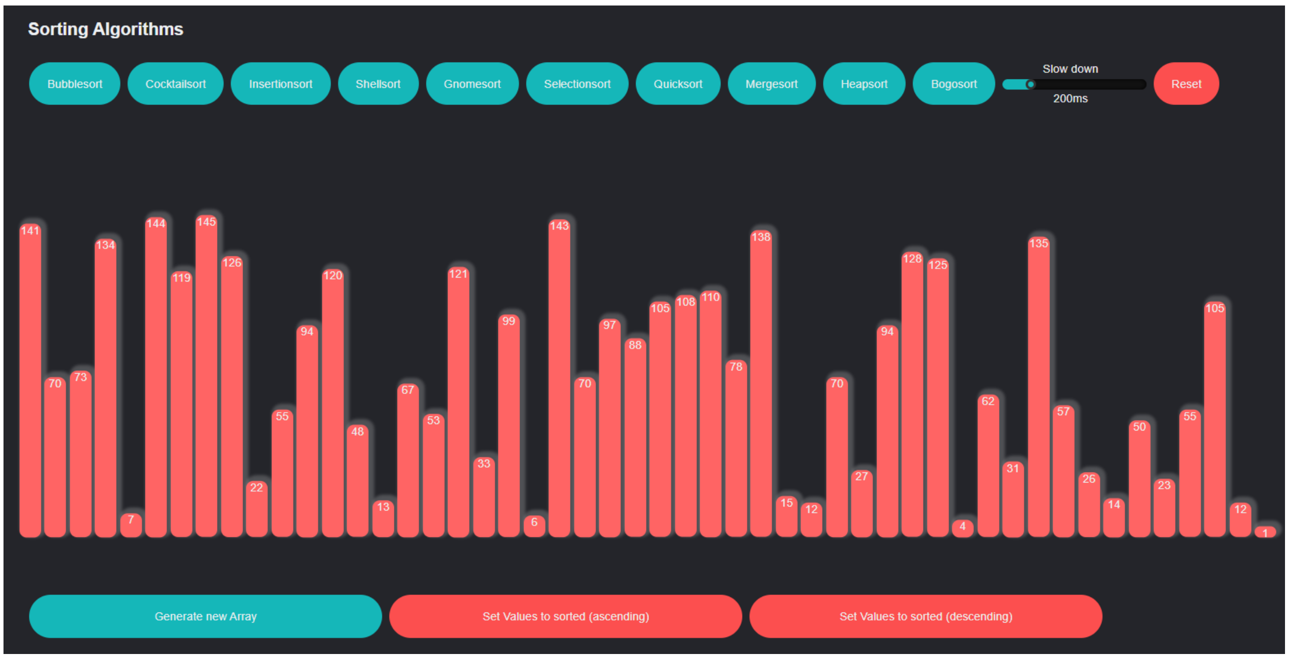

Figure A.1: Homepage View

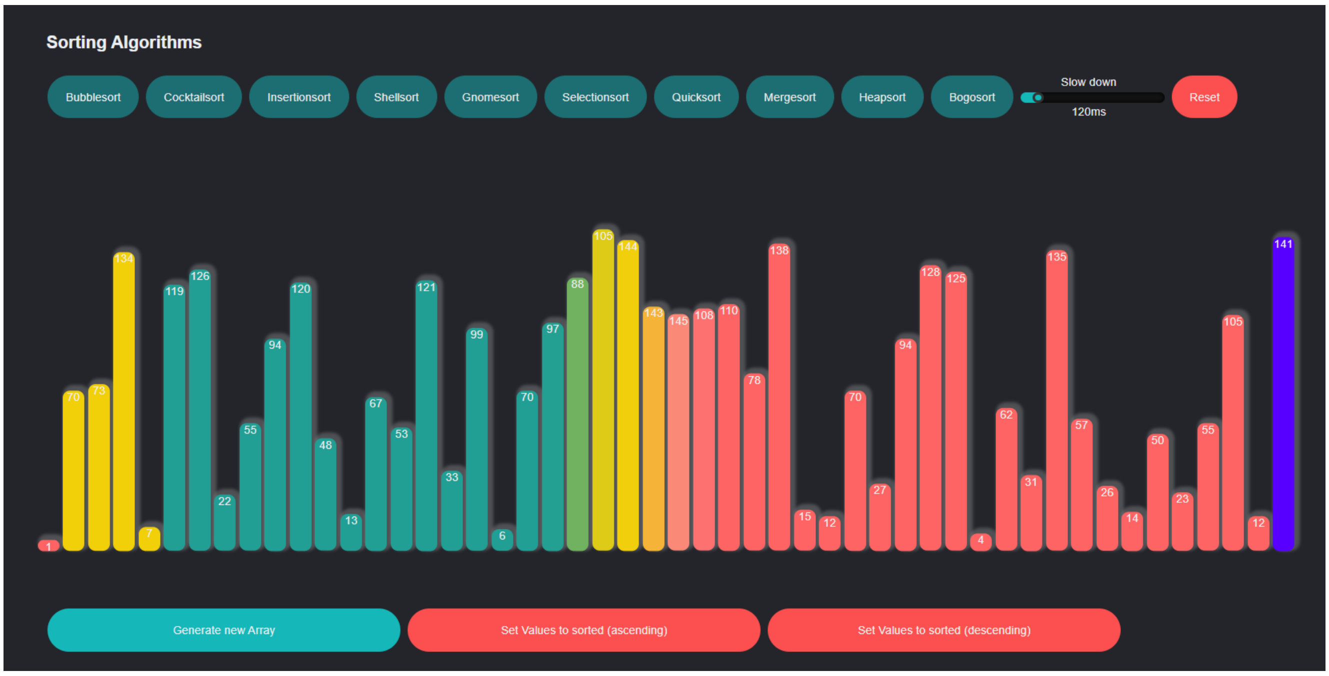

Figure A.2: Sorting View