TuDu SE

®TuDu
Use-Case-Realization Specification: Create task

Version <1.0>

Revision History

Date	Version	Description	Author

TuDu	Version: <1.0>
Use-Case-Realization Specification: Sort List View	Issue Date: <26/Oct/22>
<document identifier>	

Table of Contents

Use-Case-Realization Specification: Sort List View

1.Introduction

This Use-Case Realization Specification will specify the process implementation for the use-case "Sort List View". To do this, this document will go over purpose, scope, definitions, acronyms, abbreviations, references, and overview of this Use-Case Realization Specification.

1.1Purpose

By sorting the list view, the user is able to get the desired overview over their tasks while prioritizing the more important and displaying them on top of the list.

1.2Scope

This Use-Case Realization Specification affects the front end as well as the front end logic on how to compute and save certain information.

1.3Definitions, Acronyms, and Abbreviations

Term	Definition
ToDo/TuDu-item	A single item on a to do list.
JS	javaScript
Obj	Object, saved in memory
msg	message

1.4References

Title	Report number	Date	Publishing organization	Source
seqDia-02-sortListView.drawio.png	n/a	29/Oct/22	TuDu	../diagrams/export

1.5Overview

The most important part is the Flow of Design Section. It contains a diagram and a description of the workflow which implements the use-case "Sort List View".

2. Flow of Events—Design
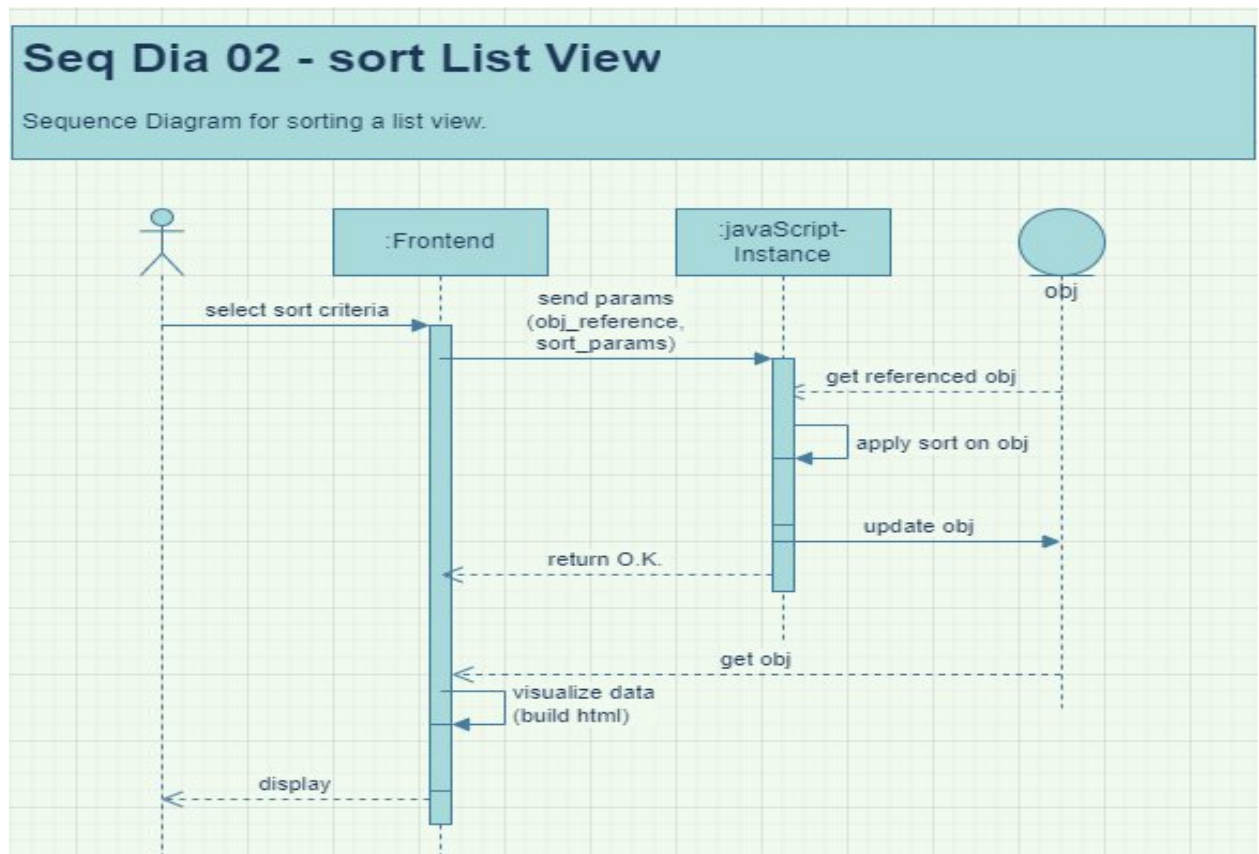
2.1 Sequence Diagram

Figure 1: Sequence Diagram: Sort List View

2.2 Description of flow of events

The User presses the a button on the front end in order to sort their todo-items. In this button, the necessary information for sorting is already contained in the button press (drop-down selection menu). The front end will call a JS function to which it sends the parameters.

The JS instance will get the required objects which were given as parameters in the function call. Based on the sort parameters which were given in the function call as well, JS will then perform a sort by the selected parameter and update the abject afterwards. A simple success information will be given to the front end.

The front end will update the referenced object and display this information to the user.

The new list will be sorted by the selected parameter.