®TuDu

Use-Case-Realization Specification: Create task

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 2

1.1 Purpose 2

1.2 Scope 2

1.3 Definitions, Acronyms, and Abbreviations 2

1.4 References 2

1.5 Overview 2

2. Flow of Events—Design 2

3. Derived Requirements 2

Use-Case-Realization Specification: create Task

# Introduction

This Use-Case Realization Specification will specify the process implementation for the use-case “add Task to a list”. To do this, this document will go over purpose, scope, definitions, acronyms, abbreviations, references, and overview of this Use-Case Realization Specification.

## Purpose

Adding a task to a list is a critical part for maintaing a todo-list. By defining this process, this Use-Case Realization Specification helps avoiding misunderstandings.

## Scope

As this is an essential use case, this Use-Case Realization Specification will affect almost all of the systems components. Those include the front end, the back end, the authenticator service as well as the database itself.

## Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| ToDo/TuDu-item | A single item on a to do list. |
| DB | Database |
| ID | Identifier, a code that is assigned to an object |
| msg | message |

## References

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Title** | **Report number** | **Date** | **Publishing organization** | **Source** |
| seqDia-01-addItem.drawio.png | n/a | 29/Oct/22 | TuDu | ../diagrams/export |

## Overview

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 “Create task”.

# Flow of Events—Design

**2.1 Sequence Diagram**

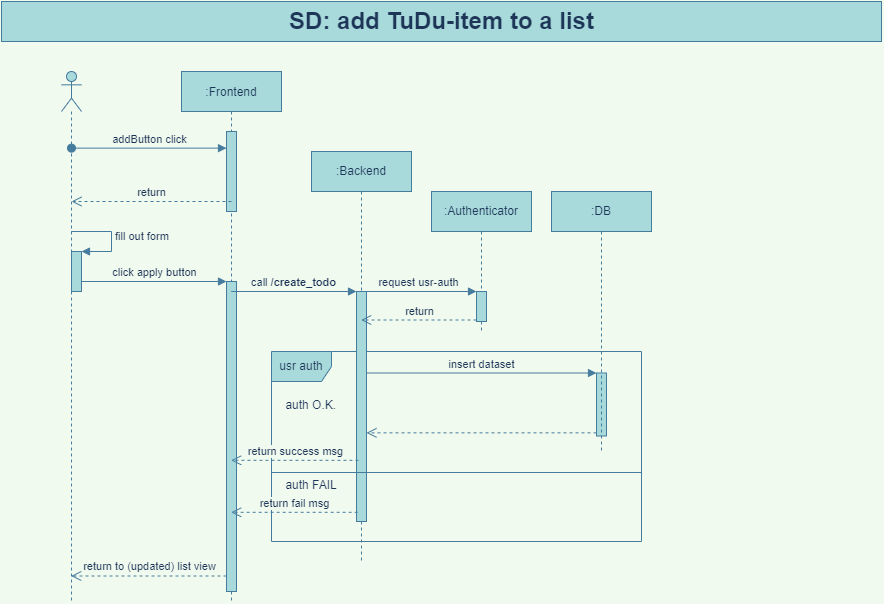


Figure 1: Sequence Diagram: Create a todo-item

**2.2 Description of flow of events**

The User presses the “add” button on the front end. The front end then prompts the user to fill out a form with the required information for a new task. When the user clicks apply the front end will call the back end with a “create\_todo” request.

The back end checks whether the user is a valid user via a “usr-auth” request to the authenticator service. If the user is valid, the user ID will be returned to the back end. When the authentication was successful, the back end will proceed to insert the parameters of the task to the corresponding database. If the DB insert was successful, a “successful” message will be returned to the back end and after that to the front end.

If the authentication or the query was not successful, the back end will return a fail message to the front end. The front end will close the “add” window on recieving a message. Either way, the message will be displayed to the user and the task list will be synced.