Software Design Document

(sdc-validation)

[Goals 2](#_Toc514332131)

[Prerequisites 2](#_Toc514332132)

[Dependencies 2](#_Toc514332133)

[Depend on other components? 2](#_Toc514332134)

[Services used 2](#_Toc514332135)

[New external NPM modules 2](#_Toc514332136)

[Other components 2](#_Toc514332137)

[Module details 2](#_Toc514332138)

[User Interface 2](#_Toc514332139)

[Detailed Design 2](#_Toc514332140)

[Component HTML usage 2](#_Toc514332141)

[Component structure 3](#_Toc514332142)

[Classes explained 3](#_Toc514332143)

[ValidationComponent (validation.component.ts) 3](#_Toc514332144)

[ValidatableComponent (validatable.component.ts) 3](#_Toc514332145)

[IValidatableComponent (validatable.interface.ts) 4](#_Toc514332146)

[ValidationGroupComponent (validation-group.component.ts) 4](#_Toc514332147)

[ValidatorComponent (base.validator.component.ts) 4](#_Toc514332148)

[IValidator (validator.interface.ts) 4](#_Toc514332149)

[Types of validators: 4](#_Toc514332150)

[Component Lifecycle 5](#_Toc514332151)

[ngAfterContentInit() in class ValidationComponent 5](#_Toc514332152)

[Test coverage 5](#_Toc514332153)

[Open Issues 5](#_Toc514332154)

# Goals

The goal of the component is to validate client side components value and to show error message in case of validation error.

The component should be easy to use, and can print the error messages in different places of the screen.

The component should validate while user enter text (or change selected item in dropdown).

Component can validate multiple components (group) at once.

# Prerequisites

N/A

# Dependencies

## Depend on other components?

Childs components that are part of validation components: like regex-validation and required validator.

## Services used

N/A

## New external NPM modules

N/A

## Other components

Validation child components

## Module details

Validation.module imported by form-elements.module and exported in form-elements.module.

# User Interface



# Detailed Design

## Component HTML usage

<sdc-input #**email** label="Please enter valid email address" [maxLength]="50" required="true"></sdc-input>

<sdc-validation [validateElement]="**email**">

<sdc-required-validator message="Field is required!"></sdc-required-validator>

<sdc-regex-validator message="This is not a valid email!" [pattern]="emailPattern"></sdc-regex-validator>

</sdc-validation>

## Component structure

**ValidationGroupComponent**

**ValidationComponent**

**ValidatableComponent**

SdcInput

SdcDropDown

**ValidatorComponent**

RegexValidatorComponent

RequiredValidatorComponent

CustomValidatorComponent

# Classes explained

## ValidationComponent (validation.component.ts)

### Description:

Main validation component, wrapper for **ValidatorComponent**.

Has method validate that iterate over all validators and execute validate method in each one.

### Interfaces:

@Input() public validateElement: ValidatableComponent;

Reference to other angular component to validate, the other component should extends **ValidatableComponent** that implements **IValidatableComponent**

@Input() public disabled: boolean;

Set disable / enable validation

## ValidatableComponent (validatable.component.ts)

### Description:

Abstract class, that each component that wants to support validation should extend (for example input component). There are 2 important methods:

1. Abstract method getValue() <- the component (exp. sdc-input) should return the value.
2. protected valueChanged = (value: any) <- the component (exp. sdc-input) should call the method when the value changed.

### Example of implementation in sdc-input component:

public getValue(): any {

return this.value;

}

onKeyPress(value: string) {

this.valueChanged(this.value);

}

## IValidatableComponent (validatable.interface.ts)

### Description:

Interface with one method getValue(), **ValidatableComponent** implements this as abstract method.

## ValidationGroupComponent (validation-group.component.ts)

This is a wrapper component for group of elements, has public method validate(): Boolean that execute validate() method in each **ValidationComponent** in the group.

## ValidatorComponent (base.validator.component.ts)

### Description:

Base class for all validator components, each validator should extends this class.

The class is responsible for showing the error message including the icon.

## IValidator (validator.interface.ts)

### Description:

Validator interface, contain only one method that each validator should implement:

validate(value: any): void;

## Types of validators:

Each validator should extends **ValidatorComponent** implements **IValidator**

Base validator receives message and disabled inputs.

### RequiredValidatorComponent

Validate if user insert value or not

### RegexValidatorComponent

Validate the value against the regex pattern

### CustomValidatorComponent

Validate the value in callback function that return Boolean.

# Component Lifecycle

## ngAfterContentInit() in class ValidationComponent

Get reference to all validators components, and subscribe to value changed event in each **ValidatableComponent.**

# Test coverage

N/A

# Open Issues

N/A