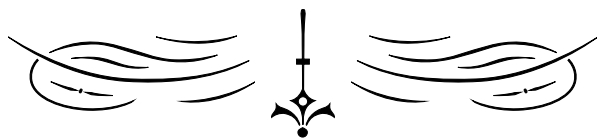


Testing Specification

MEDAWARE

by Haas, Clarissa
Patak, Rastislav
Sam, Elias
Wyrwas, Piotr

February, 2024



Contents

1	USERSTORY — 5.1.1 Intuitive frontend design	3
1.1	Test Case 1 - [VID] intuitive landing page and easy navigation	3
2	USERSTORY — 5.1.2 GHS hazard pictogram classification	4
2.1	Test Case 2 - [OCR] 180° rotated GHS pictograms	4
2.2	Test Case 3 - [OCR] 360° rotated GHS pictograms	4
2.3	Test Case 4 - Experimental Test Case [OCR] 90° rotated GHS pictograms	5
2.4	Test Case 5 - [OCR] UI only permits images	5
2.5	Test Case 6 - [OCR] UI denies other files than an image	5
2.6	Test Case 7 - [OCR] UI shows a loading spinner	6
2.7	Test Case 8 - [OCR] UI shows an image preview	6
3	USERSTORY — 5.1.3 OCR ingredient "detection"	7
3.1	Test Case 9 - [OCR] Catalyst finds 70% matching word	7
3.2	Test Case 10 - [OCR] Catalyst finds 100% matching word	7
3.3	Test Case 11 - [OCR] Catalyst don't find 20% matching word	8
3.4	Test Case 12 - [OCR] OCR reads the text of the image	8
3.5	Test Case 13 - [OCR] UI shows a loading spinner	8
3.6	Test Case 14 - [OCR] UI shows an image preview	9
4	USERSTORY — 5.1.4 Search Articles	10
4.1	Test Case 15 - [TAN] Search system is available	10
5	USERSTORY — 5.2.1 Content Management System availability	11
5.1	Test Case 16 - [TAN] Tangential is available	11
5.2	Test Case 17 - [TAN] Tangential user management	11
6	USERSTORY — 5.2.2 Topic Management	12
6.1	Test Case 18 - [TAN] Tangential is equipped with a "Topics" tab	12
6.2	Test Case 19 - [TAN] Behaviour of the "Topics" view	12
6.3	Test Case 20 - [TAN] The "Topics" view permits CRUD operations on topics	12
6.4	Test Case 21 - [TAN] Removing a topic triggers a fallback topic prompt	12
6.5	Test Case 22 - [TAN] Topic creation button is present	13
7	USERSTORY — 5.2.3 Dedicated Articles Panel	14
7.1	Test Case 23 - [TAN] Tangential is equipped with an "Articles" panel	14
7.2	Test Case 24 - [TAN] Behaviour of the "Articles" Panel	14
7.3	Test Case 25 - [TAN] Article display behavior	14
8	USERSTORY — 5.3.1 Article Editor Overview	15
8.1	Test Case 26 - [TAN] Tangential Dashboard Permits CRUD Operations	15
9	USERSTORY 5.3.2 — Creating Components	16
9.1	Test Case 27 - [TAN] Component Creation Button	16
9.2	Test Case 28 - [TAN] Component Creation Dialog	16
9.3	Test Case 29 - [TAN] Reactive Interface	16
10	USERSTORY 5.3.3 — Editing Components	17
10.1	Test Case 30 - [TAN] Controlling component parameters	17
10.2	Test Case 31 - [TAN] Selected Component Highlighting	17
10.3	Test Case 32 - [TAN] Component Properties Panel Visibility	17

1 USERSTORY — 5.1.1 Intuitive frontend design

1.1 Test Case 1 - [VID] intuitive landing page and easy navigation

TEST CASE	Intuitive frontend design and easy navigation
DESCRIPTION	The visitors' page is as minimal as possible, designed with accessibility and optimal usability in mind.
TEST REQUIREMENTS	A functional frontend that is available for testing. A test user unfamiliar with the system is selected.
INPUT	None
TEST PROCEDURE / STEPS	The test user opens the landing page. They attempt to locate key sections, such as the <i>homepage</i> , <i>about page</i> , <i>contact</i> , ... without external guidance. They provide feedback on ease of use and accessibility.
EXPECTED OUTCOME	The test user successfully navigates the site without confusion, as the key sections are easily accessible and identifiable. Then the design is confirmed as intuitive and user-friendly.

2 USERSTORY — 5.1.2 GHS hazard pictogram classification

2.1 Test Case 2 - [OCR] 180° rotated GHS pictograms

TEST CASE	180° rotated GHS pictograms
DESCRIPTION	It is possible to detect GHS hazard pictograms which are rotated by 180°.
TEST REQUIREMENTS	The ocr server has to be running; The endpoint /ocr/cnn at the ocr server has to work properly; The CNN has to be implemented and trained;
INPUT GENERAL:	An image of a GHS pictogram which is upside down (rotated by 180° or π rad).
CONCRETE:	<code>ghs_test_image_exclamation_mark_rotated_180_degree.jpeg</code>
TEST PROCEDURE / STEPS	An Administrator tests the /ocr/cnn endpoint with e.g. postman. Therefore the admin upload an image of a GHS pictogram which is rotated by 180° or π rad as form-data and send the post request. After the admin received the response he/she checks if the responded answer is correct. (For further reading about the format of api requests look into the ORC OpenAPI-specification.)
EXPECTED OUTCOME GENERAL:	The name of the uploaded GHS pictogram.
CONCRETE:	The name attribute at the response have to contain the label: Health Hazard/Hazardous to Ozone Layer.

2.2 Test Case 3 - [OCR] 360° rotated GHS pictograms

TEST CASE	360° rotated GHS pictograms
DESCRIPTION	It is possible to detect GHS hazard pictograms which are not rotated.
TEST REQUIREMENTS	The OCR server has to be running; The endpoint /ocr/cnn at the ocr server has to work properly; The CNN has to be implemented and trained;
INPUT GENERAL:	An image of a GHS pictogram which is in the "normal" position (not rotated).
CONCRETE:	<code>ghs_test_image_exclamation_mark.jpeg</code>
TEST PROCEDURE / STEPS	An Administrator tests the /ocr/cnn endpoint with e.g. postman. Therefore the admin upload an image of a GHS pictogram which is rotated by 360° or 2π rad as form-data and send the post request. After the admin received the response he/she checks if the responded answer is correct. (For further reading about the format of api requests look into the OCR OpenAPI-specification)
EXPECTED OUTCOME GENERAL:	The name of the uploaded GHS pictogram.
CONCRETE:	The name attribute at the response has to contain the label: Health Hazard/Hazardous to Ozone Layer.

2.3 Test Case 4 - Experimental Test Case [OCR] 90° rotated GHS pictograms

TEST CASE	Experimental Test Case 90° rotated GHS pictograms
DESCRIPTION	We want to know what happens if a GHS hazard pictograms which is rotated by 90° is processed. So this is an experimental test where very result is acceptable except of an exception.
TEST REQUIREMENTS	The OCR server has to be running; The endpoint /ocr/cnn at the OCR server has to work properly; The CNN has to be implemented and trained;
INPUT GENERAL:	An image of a GHS pictogram which is rotated by 90° (or $\frac{\pi}{2}$ rad).
CONCRETE:	<code>ghs_test_image_exclamation_mark_rotated_90_degree.jpeg</code>
TEST PROCEDURE / STEPS	An Administrator tests the /ocr/cnn endpoint with e.g. postman. Therefore the admin upload an image of a GHS pictogram which is rotated by 180° or π rad as form-data and send the post request. After the admin received the response he/she checks if the responded answer is correct. (For further reading about the format of api requests look into the ORC OpenAPI-specification.)
EXPECTED OUTCOME	To receive a response but it doesn't matter what stands in the name attribute.

2.4 Test Case 5 - [OCR] UI only permits images

TEST CASE	UI only permits images (.jpg, .jpeg, .png, .tiff).
DESCRIPTION	The UI only permits uploading image files.
TEST REQUIREMENTS	A finished or usable UI page for uploading GHS pictograms.
INPUT GENERAL:	A File with an image extension (.jpg, .jpeg, .png, .tiff).
CONCRETE:	<code>ghs_test_image_exclamation_mark_rotated_180_degree.jpeg</code>
TEST PROCEDURE / STEPS	A user uploads the file with an image extension.
EXPECTED OUTCOME	The upload works without getting a message that this datatype is not supported or another error-message and a preview of the image which was uploaded is shown.

2.5 Test Case 6 - [OCR] UI denies other files than an image

TEST CASE	UI denies other files than images.
DESCRIPTION	The UI denies uploading other files that are not an image. (e.g. .pdf, .docx, .txt, .md, .tex, ...)
TEST REQUIREMENTS	A finished or usable UI page for uploading GHS pictograms.
INPUT GENERAL:	A file with an other ending than an image. (e.g. a pdf file)
CONCRETE:	<code>pdf_test_file.pdf</code>
TEST PROCEDURE / STEPS	A user uploads the file with a other ending.
EXPECTED OUTCOME	The upload does not work. That means that an error message is displayed that somehow indicates that the upload is not working.

2.6 Test Case 7 - [OCR] UI shows a loading spinner

TEST CASE	UI shows a loading spinner
DESCRIPTION	After a valid file was uploaded and during the backend processes the data a loading spinner should be displayed.
TEST REQUIREMENTS	A finished or usable UI page for uploading GHS pictograms.
INPUT GENERAL:	A valid file (= an image).
CONCRETE:	<code>ghs_test_image_exclamation_mark.jpeg</code>
TEST PROCEDURE / STEPS	A user uploads the image and while the backend is processing the data a loader should be displayed.
EXPECTED OUTCOME	Somewhere on the page where you can upload the GHS pictograms a loading spinner is displayed during the backend is processing the data.

2.7 Test Case 8 - [OCR] UI shows an image preview

TEST CASE	UI shows an image preview
DESCRIPTION	A valid file is uploaded and a preview is shown.
TEST REQUIREMENTS	A finished or usable UI page for uploading GHS pictograms.
INPUT GENERAL:	A valid file (= an image).
CONCRETE:	<code>ghs_test_image_exclamation_mark.jpeg</code>
TEST PROCEDURE / STEPS	A user uploads the image.
EXPECTED OUTCOME	Somewhere on the page a preview of the image is displayed.

3 USERSTORY — 5.1.3 OCR ingredient "detection"

3.1 Test Case 9 - [OCR] Catalyst finds 70% matching word

TEST CASE	Catalyst finds 70% matching word.
DESCRIPTION	It is possible to recognize 70% of the correctly spelled (correctly detected) words in the search.
TEST REQUIREMENTS	Catalyst has to be running; The database has to be running; Catalyst is connected to the database; The specific method is implemented;
INPUT GENERAL:	A text which is to 30 % wrong spelled. (e.g. Hollo Warlt.)
CONCRETE:	Hollo Warlt
TEST PROCEDURE / STEPS	Pass the word to the method which processes the text from images and finds it in the DB.
EXPECTED OUTCOME GENERAL:	The information which is stored to the ingredient (e.g. to the ingredient "Hello World").
CONCRETE:	The information which is stored to the ingredient "Hello World."

3.2 Test Case 10 - [OCR] Catalyst finds 100% matching word

TEST CASE	Catalyst finds 100% matching word.
DESCRIPTION	It is possible to recognize 70% of the correctly spelled (correctly detected) words in the search. So it should also find words which are completely correctly spelled.
TEST REQUIREMENTS	Catalyst has to be running; The database has to be running; Catalyst is connected to the database; The specific method is implemented;
INPUT GENERAL:	A text which is an ingredient which is exactly written like in the database.
CONCRETE:	Hello World
TEST PROCEDURE / STEPS	Pass the word to the method which processes the text from images and finds it in the DB.
EXPECTED OUTCOME GENERAL:	The information which is stored to the ingredient.
CONCRETE:	The information which is stored to the ingredient "Hello World."

3.3 Test Case 11 - [OCR] Catalyst don't find 20% matching word

TEST CASE	Catalyst doesn't find 20% of the matching words.
DESCRIPTION	It is possible to recognize 70% of the correctly spelled (correctly detected) words in the search. So it doesn't have to be possible to find words that are only 20% correctly spelled.
TEST REQUIREMENTS	Catalyst has to be running; The database has to be running; Catalyst is connected to the database; The specific method is implemented;
INPUT GENERAL:	A text which is to 80 % wrong spelled. (e.g. Hoffa Varft.)
CONCRETE:	Hoffa Varft (the original word would be Hello World)
TEST PROCEDURE / STEPS	Pass the word to the method which processes the text from images and finds it in the DB.
EXPECTED OUTCOME	No information is found to the word in the DB. So no information are returned.

3.4 Test Case 12 - [OCR] OCR reads the text of the image

TEST CASE	OCR reads the text of the image.
DESCRIPTION	The OCR server reads the text from an image.
TEST REQUIREMENTS	The ocr server has to be running; The endpoint /ocr/ at the ocr server has to work properly;
INPUT GENERAL:	An image with chemical ingredients on it.
CONCRETE:	ocr_ingredient_test_file.jpeg
TEST PROCEDURE / STEPS	An administrator uploads the valid file to the OCR-Server which extracts the text on it.
EXPECTED OUTCOME	A similar text which is standing on the image. (In this case similar means that is understandable/readable for humans.)

3.5 Test Case 13 - [OCR] UI shows a loading spinner

TEST CASE	UI shows a loading spinner
DESCRIPTION	After a valid file was uploaded and during the backend processes the data a loading spinner should be displayed.
TEST REQUIREMENTS	A finished or usable UI page for uploading GHS pictograms.
INPUT GENERAL:	A valid file (= an image).
CONCRETE:	ghs_test_image_exclamation_mark.jpeg
TEST PROCEDURE / STEPS	A user uploads the image and while the backend is processing the data a loader should be displayed.
EXPECTED OUTCOME	Somewhere on the page where you can upload the "ingredients" a loading spinner is displayed during the backend is processing the data.

3.6 Test Case 14 - [OCR] UI shows an image preview

TEST CASE	UI shows an image preview
DESCRIPTION	A valid file is uploaded and a preview is shown.
TEST REQUIREMENTS	A finished or usable UI page for uploading GHS pictograms.
INPUT GENERAL:	A valid file (= an image).
CONCRETE:	<code>ghs_test_image_exclamation_mark.jpeg</code>
TEST PROCEDURE / STEPS	A user uploads the image.
EXPECTED OUTCOME	Somewhere on the page a preview of the image is displayed.

4 USERSTORY — 5.1.4 Search Articles

4.1 Test Case 15 - [TAN] Search system is available

TEST CASE	Is the search system available
DESCRIPTION	Testing if the search system is functional in the VIR - frontend
TEST REQUIREMENTS	Frontend and search system reachable. A test user is selected.
INPUT	Search term
TEST PROCEDURE / STEPS	The test user writes a term, e.g. “Sodium Hydroxide” into the searchbar and clicks on the “Search” button.
EXPECTED OUTCOME	The interface should then display all articles which match the search term literally — if the search term is a substring of the article content — or logically — if the article matches the search term according to a semantic analysis.

5 USERSTORY — 5.2.1 Content Management System availability

5.1 Test Case 16 - [TAN] Tangential is available

TEST CASE	Is the tangential service available
DESCRIPTION	Testing if tangential is functional and reachable
TEST REQUIREMENTS	Catalyst has to be running; The Tangential service has to be running; Tangential has to be separate from the frontend
INPUT	
TEST PROCEDURE / STEPS	A test user executes the following Unix shell command: <code>curl -o /dev/null -s -w "%{http_code}" <URL></code> Where URL is the web URL of the current deployment.
EXPECTED OUTCOME	Tangential accessible; The aforementioned command returns the status code 200

5.2 Test Case 17 - [TAN] Tangential user management

TEST CASE	Tangential allows for user management
DESCRIPTION	Testing if tangential allows to maintain and manage users
TEST REQUIREMENTS	Catalyst has to be running; The Tangential service has to be running and reachable;
INPUT	Edit Info on users
TEST PROCEDURE / STEPS	The Tangential interface is accessed by a test person through a URL. They enter the user management section through a tab on the left-hand side of the viewport, upon which they click the “Edit” button on one of the entries. Thereafter, the test user alters the value of the text fields representing the first and last names of the user, after which they press the “Save” button.
EXPECTED OUTCOME	Upon returning to the user list, the values of the altered fields have indeed changed.

6 USERSTORY — 5.2.2 Topic Management

6.1 Test Case 18 - [TAN] Tangential is equipped with a “Topics” tab

TEST CASE	The Tangential dashboard sidebar contains a “Topics” section.
DESCRIPTION	The Tangential system is equipped with a dedicated section for managing topics.
TEST REQUIREMENTS	The Tangential system (including Catalyst) is up and running.
INPUT	None
TEST PROCEDURE / STEPS	A beta-tester examines the left-hand side of the Tangential dashboard in search for a “Topics” tab.
EXPECTED OUTCOME	The tester determines that the dashboard contains a dedicated topic management section.

6.2 Test Case 19 - [TAN] Behaviour of the “Topics” view

TEST CASE	The “Topics” view lists all existing topics
DESCRIPTION	The “Topics” panel reveals a listing of all existing topics
TEST REQUIREMENTS	The Tangential system (including Catalyst) is up and running.
INPUT	None
TEST PROCEDURE / STEPS	A beta-tester enters the “Topics” view by clicking on the “Topics” panel on the left-hand side panel.
EXPECTED OUTCOME	They are presented with a list of all topics which were returned from the backend server.

6.3 Test Case 20 - [TAN] The “Topics” view permits CRUD operations on topics

TEST CASE	The “Topics” tab implements CRUD operations.
DESCRIPTION	The “Topics” panel allows the user to perform CRUD operations on the topics
TEST REQUIREMENTS	The Tangential system (including Catalyst) is up and running.
INPUT	None
TEST PROCEDURE / STEPS	A beta-tester examines the “Topics” dashboard.
EXPECTED OUTCOME	The tester determines that the “Topics” view allows for <i>creating</i> , <i>updating</i> and <i>deleting</i> topics through appropriately labelled buttons.

6.4 Test Case 21 - [TAN] Removing a topic triggers a fallback topic prompt

TEST CASE	Attempting to remove a Topic triggers a dialogue
DESCRIPTION	When the user attempts to delete a Topic, a dialogue prompts the user to pick a fallback topic.
TEST REQUIREMENTS	The Tangential system (including Catalyst) is up and running.
INPUT	None
TEST PROCEDURE / STEPS	A beta-tester attempts the removal of a topic.
EXPECTED OUTCOME	The tester determines that the action triggers the expected dialogue.

6.5 Test Case 22 - [TAN] Topic creation button is present

TEST CASE	A topic creation button (“Add”) is located in the top right corner.
DESCRIPTION	—”—
TEST REQUIREMENTS	The Tangential system (including Catalyst) is up and running.
INPUT	None
TEST PROCEDURE / STEPS	A beta-tester examines the “Topics” view.
EXPECTED OUTCOME	The tester determines that an “Add” button is located in the upper right-hand side corner.

7 USERSTORY — 5.2.3 Dedicated Articles Panel

7.1 Test Case 23 - [TAN] Tangential is equipped with an “Articles” panel

TEST CASE	The Tangential dashboard sidebar contains an “Article” section.
DESCRIPTION	The Tangential system is equipped with a dedicated section for managing articles.
TEST REQUIREMENTS	The Tangential system (including Catalyst) is up and running.
INPUT	None
TEST PROCEDURE / STEPS	A beta-tester examines the sidebar on the left-hand side of the Tangential dashboard.
EXPECTED OUTCOME	The tester determines that the sidebar contains an “Articles” tab.

7.2 Test Case 24 - [TAN] Behaviour of the “Articles” Panel

TEST CASE	The “Articles” panel reveals a listing of Articles.
DESCRIPTION	By default, the “Articles” panel displays a list of existing articles.
TEST REQUIREMENTS	The Tangential system (including Catalyst) is up and running.
INPUT	None
TEST PROCEDURE / STEPS	A beta-tester clicks on the “Articles” tab in the sidebar.
EXPECTED OUTCOME	The tester determines that they are presented with a list of articles upon clicking the “Articles” tab.

7.3 Test Case 25 - [TAN] Article display behavior

TEST CASE	The behavior of the Article listing can be changed.
DESCRIPTION	The article listing in the “Articles” tab can be switched between displaying all articles, or only the articles authored by the logged-in user.
TEST REQUIREMENTS	The Tangential system (including Catalyst) is up and running.
INPUT	None
TEST PROCEDURE / STEPS	A beta-tester switches the display type through a bistate mode selector in the Article listing view.
EXPECTED OUTCOME	The tester determines that the Article listing properly responds to changes in the display mode.

8 USERSTORY — 5.3.1 Article Editor Overview

8.1 Test Case 26 - [TAN] Tangential Dashboard Permits CRUD Operations

TEST CASE	CRUD functionality in the TAN dashboard
DESCRIPTION	The Tangential dashboard allows for performing CRUD operations on articles.
TEST REQUIREMENTS	The Tangential system (including Catalyst) is up and running.
INPUT	None
TEST PROCEDURE / STEPS	A beta-tester examines the available options for article entries in the Tangential dashboard.
EXPECTED OUTCOME	The tester concludes that the Tangential system allows for <i>creating</i> , <i>deleting</i> and <i>editing</i> articles through appropriately labelled buttons next to each entry.

9 USERSTORY 5.3.2 — Creating Components

9.1 Test Case 27 - [TAN] Component Creation Button

TEST CASE	Creating components is accessible through a button
DESCRIPTION	New components can be created with the help of a button in the WYSIWYG editor
TEST REQUIREMENTS	The Tangential system (including Catalyst) is up and running
INPUT	None
TEST PROCEDURE / STEPS	A beta tester examines the Tangential editor to establish the presence of the required functionality.
EXPECTED OUTCOME	The beta tester confirms that the editor does indeed allow for adding new components with a generic “Add” button

9.2 Test Case 28 - [TAN] Component Creation Dialog

TEST CASE	Creating component triggers a UI dialogue
DESCRIPTION	A UI dialogue pops up upon invoking the component creation process.
TEST REQUIREMENTS	The Tangential system (including Catalyst) is up and running
INPUT	None
TEST PROCEDURE / STEPS	A beta tester clicks on the “Add” button in the Tangential WYSIWYG UI.
EXPECTED OUTCOME	The beta tester confirms that the aforementioned action triggers a dialogue prompting the user for a component identifier.

9.3 Test Case 29 - [TAN] Reactive Interface

TEST CASE	Preview updates according on any change
DESCRIPTION	Creating a new component triggers the preview to re-render.
TEST REQUIREMENTS	The Tangential system (including Catalyst) is up and running
INPUT	None
TEST PROCEDURE / STEPS	A beta tester examines the Tangential editor; They create a new image component and check whether the change is reflected in the rendered preview.
EXPECTED OUTCOME	The beta tester confirms that the preview refreshed upon a change to the component structure.

10 USERSTORY 5.3.3 — Editing Components

10.1 Test Case 30 - [TAN] Controlling component parameters

TEST CASE	Component parameters can [individually] be altered
DESCRIPTION	The parameters of each component can be changed in the Tangential system
TEST REQUIREMENTS	The Tangential system (including Catalyst) is up and running
INPUT	None
TEST PROCEDURE / STEPS	An experienced beta tester is selected. They enter the WYSIWYG editor of a random article. Thereafter, they select a heading component from the left-hand side sidebar and switch to the “Properties” tab.
EXPECTED OUTCOME	The tester is presented with a list of properties for the selected component, mainly <code>ELEMENT_TYPE</code> and <code>TEXT</code>

10.2 Test Case 31 - [TAN] Selected Component Highlighting

TEST CASE	Highlighting selected components
DESCRIPTION	Selected article components are highlighted in the view
TEST REQUIREMENTS	The Tangential system (including Catalyst) is up and running
INPUT	None
TEST PROCEDURE / STEPS	A beta tester opens an article in the WYSIWYG editor and selects random components.
EXPECTED OUTCOME	The beta tester confirms that the view representation of the selected components is visually highlighted upon being selected.

10.3 Test Case 32 - [TAN] Component Properties Panel Visibility

TEST CASE	Conditional visibility of the component properties panel
DESCRIPTION	Component visibility panel is triggered by a component being selected
TEST REQUIREMENTS	The Tangential system (including Catalyst) is up and running
INPUT	None
TEST PROCEDURE / STEPS	A beta tester opens an article in the WYSIWYG editor and selects a random component whilst examining the left-hand side part of the view.
EXPECTED OUTCOME	The beta tester confirms that the property panel is only visible when a component is selected.