Dublin Coding School - QA course exam

Instruction manual:

1. After opening this file, download it to your computer and save it to your Google Drive:
   a. Go to File -> Download -> Microsoft word (or other format)
   b. Open your personal Google Drive, press the "+ New" button in the upper left corner -> File upload -> Choose this file downloaded to your computer
   c. Start working!
2. Task structure:
   a. Context and useful information written in black
   b. Real points for which answers are expected are marked in blue
3. Where to register a defect?
   a. 1.variants:
      i. create a new ticket for your JIRA project (can also be a final work project)
      ii. insert a link to the created defect in this document
   b. 2.variants:
      i. Create a table here even in the document and fill it
4. Where to write Test Case?
   a. 1.variants:
      i. Create a new task in the JIRA project
      ii. Attach the QAlity test case
      iii. Place a link to the created defect in this document, if one is discovered
   b. 2.variants:
      i. Create a table here even in the document and fill it
5. **Share this document after execution:**
   a. Press the "Share" button in the upper right corner
   b. Specify e-mail
   c. Specify "Editor" rights

1. DEFECT REGISTRATION

Comment on the task, registers the described defect with all necessary fields and descriptions of what could happen further, taking into account the life cycle of defects and the given situation.

Product: Mobile app for photo editing

Situation: Day Three of the Agile Sprint. 7.0 is scheduled for release in three weeks
Our team wants to release convenient search sections in the release so that users can search for the stickers, pictures and frames they need.

Content of the test task:

US-1234 [Search] Implement search functionality
Description:

As a user, I want to be able to search for stickers, pictures and frames to edit pictures more easily.

Tasks for backend developer:

1. Create a data category and send it with the search request to narrow down the search. For example, if a user searches for something in the sticker finder and uses the word "Sun", then the search should only be on data marked as "Stickers", not all data.
2. The searcUsearch, then the search results should return both the sun, the world, and sunglasses.
3. Results should be returned starting with the highest match.

Tasks for frontend developer:

1. Pass the user input to the backend along with the data category to search for (stickers, frames or pictures)
2. The results should be returned as an endlessly scrollable list.

AC:

- Users can search for stickers, pictures and frames in separate search engines. Users can search Stickers, Photos and Frames separately in their sections.
- If the user searches for something in the "Stickers" section, the results returned contain only stickers.
- If the user searches for something in the "Frames" section, the results returned contain only frames.
- If the user searches for something in the "Images" section, the returned results contain only images.

Comments:

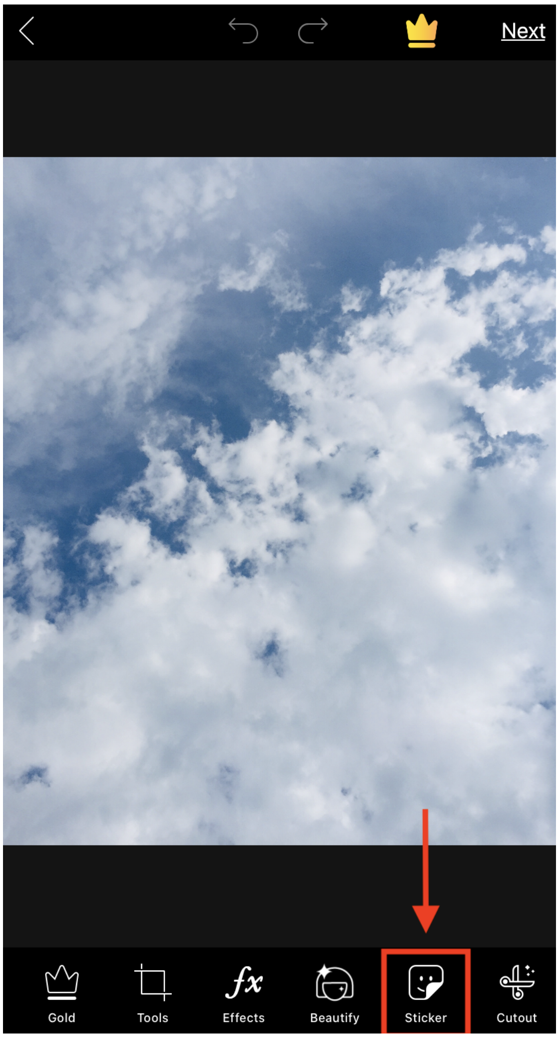
Backend Developer Comment:
Developed within BE-2715 and deployed on QA test environment with version 23810.2367.12. Ready for testing.

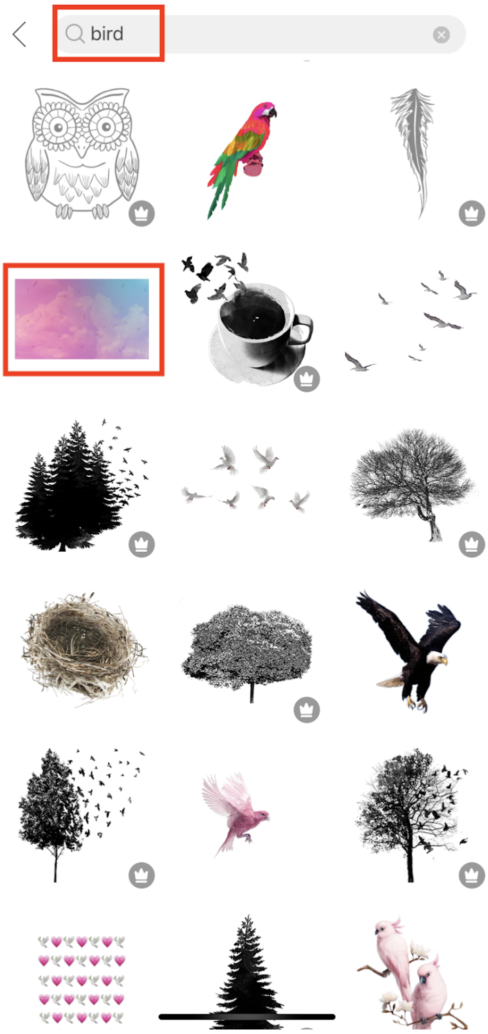
Frontend Developer Comment:
Developed within FE-3792 and deployed on QA test environment with version 114632.2367.123321. Ready for testing.

QA Manual Engineer Comment:
Tested in accordance with AC and found a medium-severity functional bug. The bug is registered in Jira and ready for fixing by the backend developer.

PLACE FOR YOUR COMMENT:
While testing the new functionality, you noticed that when you search for something in the "Stickers" section, sometimes the searcher returns not only stickers, but also pictures (see screenshots below). Post a comment here after your testing, finding a bug.

Screenshots with the defect:





TASK:

1. Write a comment in the description of the testing task under developer comments (highlighted in blue), as you would if you completed testing and found one bug.
   DONE
2. What is your procedure for investigating the defect? (what else will you do to isolate the impact area and get additional information)
   → Read the requirements carefully and apply common sense. Also, it is important to consider user needs, comfort, and security. We need to understand WHAT is happening, WHERE it is happening, and if it can happen anywhere else, and rule out the possibility that these are configuration or other errors.
   →To isolate the impact area and get additional information, I would check if the configurations are correct, for example, if some functionality should be tested with specific settings. Also, I would check if the environment is stable and if the changes have really been on the test environment. It is also important to know if there are any blocked access issues.
3. Register a defect. →
   https://rigacoding.atlassian.net/browse/ML-362?atlOrigin=eyJpIjoiNzExYWNhODY2MDU0NGNmN2I2MGY2MjRlYTA3NzZjMDMiLCJwIjoiaiJ9
4. Read the text CAREFULLY!! All the necessary details are described there. OK:)

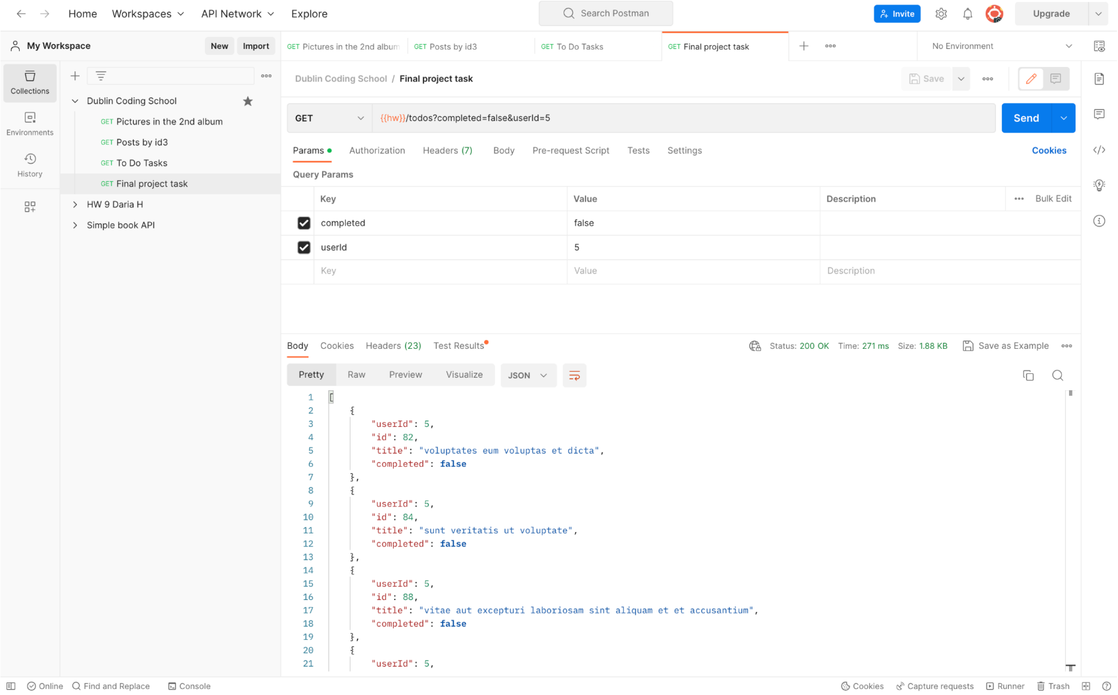
5. Short descriptions of the further life cycle of the defect, taking into account the situation described at the beginning of the task. (Who will you pass the defect on to, which will happen within the sprint, and which will happen within the release)
   → The created defect is assigned to the Dev team manager who checks the validity of it. If there's no missing info, the defect is passed on to the developer who has to fix it. Once the defect is fixed, it is deployed back into the QA test environment and assigned to the QA team member. The retesting happens either at the end of the sprint or during the release stage as there might be other tasks of a higher priority.

2. Postman request

Request: Find all the fifth user's uncompleted quest(s). (using the jsonplaceholder we already used in the lessons).

TASK:

1. Create the above request using the Postman tool. DONE



Dublin Coding School / Final project task

GET {{hw}}/todos?completed=false&userId=5

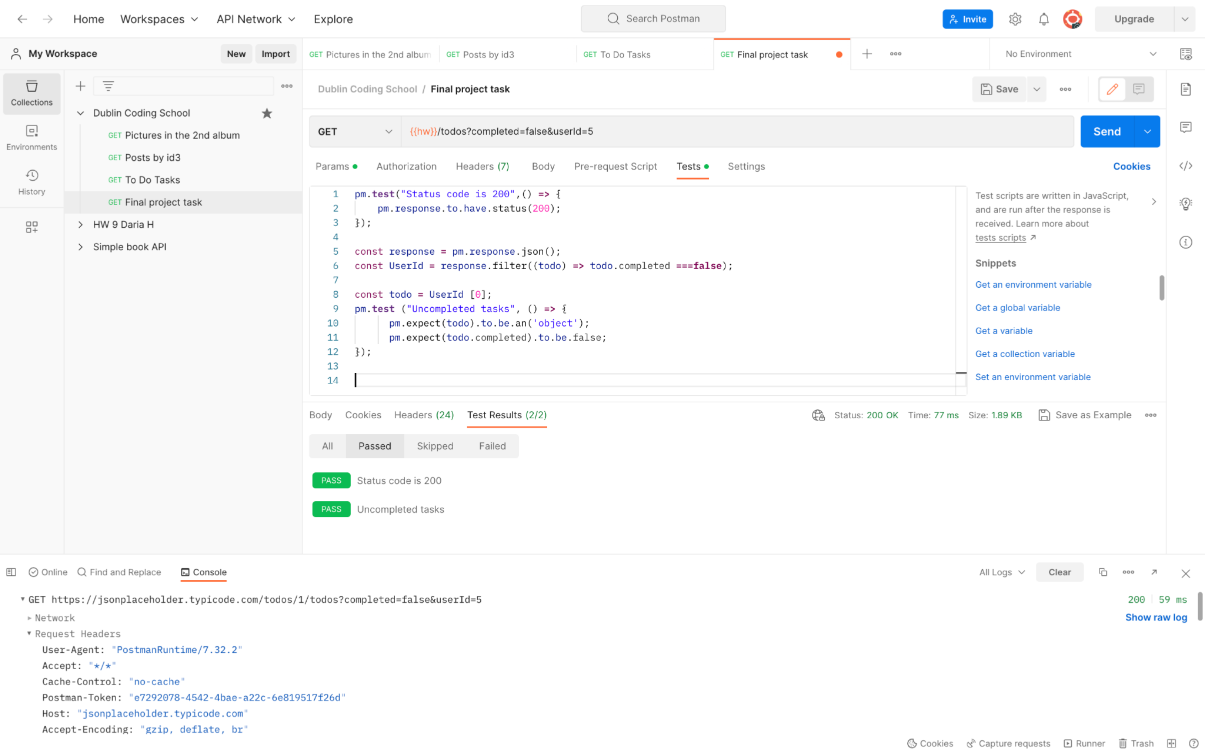
Query Params

	Key	Value	Description
☑	completed	false	
☑	userId	5	
	Key	Value	Description

Status: 200 OK Time: 271 ms Size: 1.88 KB

```
[
    {
        "userId": 5,
        "id": 82,
        "title": "voluptates eum voluptas et dicta",
        "completed": false
    },
    {
        "userId": 5,
        "id": 84,
        "title": "sunt veritatis ut voluptate",
        "completed": false
    },
    {
        "userId": 5,
        "id": 88,
        "title": "vitae aut excepturi laboriosam sint aliquam et et accusantium",
        "completed": false
    },
    {
        "userId": 5,
```

2. Create Test Case.
   a. Remember that a created Test case does not include the actual result and pass/fail status!!
   b. The Test Case does not need to be placed in the Test Cycle and executed, only the Test Case itself must be created.



GET {{hw}}/todos?completed=false&userId=5

```
pm.test("Status code is 200",() => {
    pm.response.to.have.status(200);
});

const response = pm.response.json();
const UserId = response.filter((todo) => todo.completed ===false);

const todo = UserId [0];
pm.test ("Uncompleted tasks", () => {
        pm.expect(todo).to.be.an('object');
        pm.expect(todo.completed).to.be.false;
});

```

PASS Status code is 200

PASS Uncompleted tasks

```
▾ GET https://jsonplaceholder.typicode.com/todos/1/todos?completed=false&userId=5        200 | 59 ms
  ▸ Network
  ▾ Request Headers
     User-Agent: "PostmanRuntime/7.32.2"
     Accept: "*/*"
     Cache-Control: "no-cache"
     Postman-Token: "e7292078-4542-4bae-a22c-6e819517f26d"
     Host: "jsonplaceholder.typicode.com"
     Accept-Encoding: "gzip, deflate, br"
```

3. Test this User story

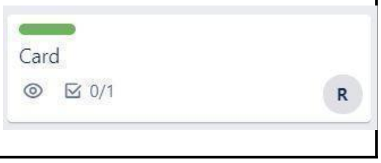
To do this, you need to log in to Trello and create/use your own board.

US-1234 [UI][Labels] Implement labels design and filtering

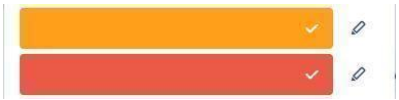
Description: as a user, I would like to be able to add labels to cards to make them easier to find and filter effectively.

Tasks for the frontend developer:

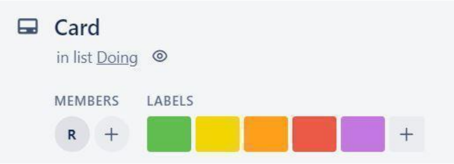
- The Label section must be available in the User's menu (User's menu -> More…)
- There should be 6 main marks available in the section with the following colors
- In the desktop version, when hovering the mouse over one of the labels, a "shadow" should also appear on the left side
- Colors for marks and shadows:
    - Green: background colour #61bd4f & shadow #519839
    - Yellow: background colour #f2d600 & shadow #d9b51c
    - Orange: background colour #ff9f1a & shadow #cd8313
    - Red: background colour #eb5a46 & shadow #b04632
    - Purple: background colour #c377e0 & shadow #89609e
    - Dark blue: background colour #0079bf & shadow #055a8c
- The user can add extra labels. Available colors and shades:
    - Light Green: background colour #51e898 & shadow #4bbf6b
    - Light Blue: background colour #00c2e0 & shadow #0098b7
    - Pink: background colour #ff78cb & shadow #c9558f



    - Dark: background colour #344563 & shadow #091e42
    - No colour: background colour #b3bac5 & shadow #97a0af
- Design examples for label placement on cards:
    - When a tag is attached to a card, it appears as a line with rounded corners in the appropriate color. One line can fit 5 tags at a time. The sixth mark is already moving to a new line.
    - If a tag is attached to a map, a white mark appears on it when editing tags:



    - When you open the card, the added marks are reflected as follows:

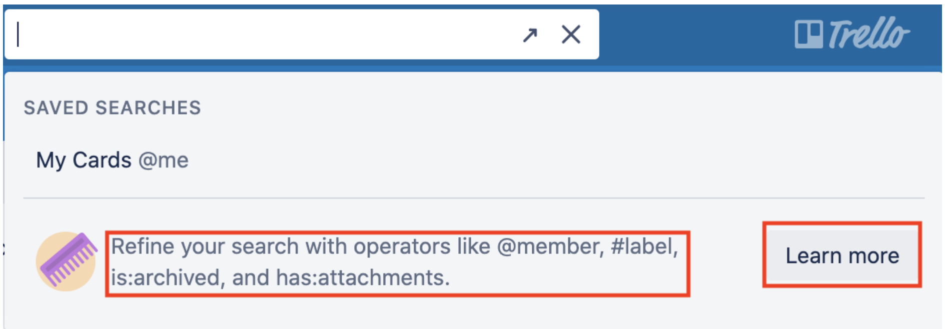


Tasks for the backend developer:

- Develop filtering capability using tags (user uses search field for filtering)
- It should be possible to filter by tag color and name
- The search results should reflect the card preview and location on the board and column.

Instructions for testing:

- Check the color match
- Verify that the development matches the design examples
- Pay special attention to the colorless label, is it really attached to a card, because it will not be shown on the card on the board.
- Check label filtering. The search syntax must be described when clicking in the search field:



Refine your search with operators like @member, #label, is:archived, and has:attachments.

Comments:

Backend developer comment**:**
Ready for testing on QA test environment with version 48720.4628205.674921

Frontend developer comment**:**
Ready for testing on QA test environment with version 46382.521

Comment from Product Owner:
QA please also add the tags HTML element*data-color* value, we will probably use it as well.

A place for your test scenarios:
Space for your final comment with visuals:

TS-01
Feature: Adding labels with the main marks to cards
Scenario: User should be able to add labels to cards

Given I am on the homepage
And I created a board
When I add a card
And clicked on the User’s menu button
And chose ‘Labels’
Then checkbox menu with 6 labels of different colors is displayed
And user is able to select and add the labels successfully

TS-02
Feature: Adding extra labels to cards
Scenario: User should be able to add extra labels to cards

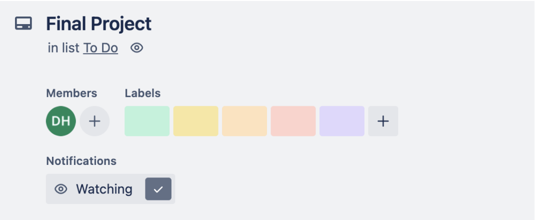
Given I created a card
And have a row of selected labels on it
When I clicked on the empty label
And clicked ‘Create a new label’ button
Then color-picker with colors box appeared
And the user is able to select color and add an extra label successfully.

Final comment:

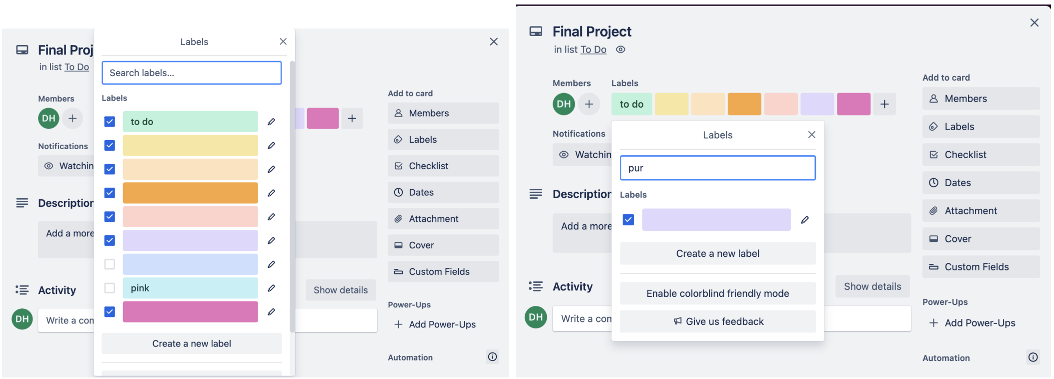
- The color match corresponds with the requirement.

Green: background colour #61bd4f & shadow #519839
Yellow: background colour #f2d600 & shadow #d9b51c
Orange: background colour #ff9f1a & shadow #cd8313
Red: background colour #eb5a46 & shadow #b04632
Purple: background colour #c377e0 & shadow #89609e
Dark blue: background colour #0079bf & shadow #055a8c
Light Green: background colour #51e898 & shadow #4bbf6b
Light Blue: background colour #00c2e0 & shadow #0098b7
Pink: background colour #ff78cb & shadow #c9558f
Dark: background colour #344563 & shadow #091e42
No colour: background colour #b3bac5 & shadow #97a0af

- The development matches the design examples and the colorless label is attached to a card.



- The label filtering works as expected. The search syntax is described when clicking in the search field.



TASK

Test this US

Add your testing scenarios in the comments section (marked in blue) - do not write detailed Test cases!!

Add your comment with the test results in the section (marked in blue) according to the test results

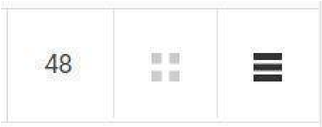
You follow **FOR EVERYTHING** task flow, as we practiced in the lessons! IF there are defects, then the corresponding actions and comments (it is not necessary to register the defect yourself, but it must be mentioned). If there are other comments/suggestions, we also mention them.

4. Execute the Test Case given below!

Do not forget about the mandatory details that must be specified for each execution!

Test scenario: Check the default and minimum values in the finder field Pre-condition: Open the 1a.lv page Note: All steps are mandatory on PC, only steps marked on mobile phones should be checked					
No.	Test Steps	Test Data	Expected Result	**Actual/Comment**	**Status**
TC 01	Make sure the search field is filled by default		The search field is filled with gray text by default.	Make sure the search field was filled with gray test by default	
	Repeat the previous step on your mobile device			The previous step was repeated on mobile device Android version 13	

				successfully.	
TC 02	Click the cursor in the finder input field (desktop only)		If something has been searched before, the search history appears. If not, nothing is displayed.	The search history appeared when something has been searched before. Nothing was displayed with no previous search history.	
	Enter one letter in the search engine, indicate which letter was tested with.	Any single letter	No search results. An info message is displayed - "Enter at least 2 symbols in the search!"	No search results. An info message was displayed - "Enter at least 2 symbols in the search!"	
	Repeat the previous step on your mobile device				
Test scenario: Verify search result filtering and display Pre-condition: Open 1a.lv homepage Note: All steps obligatory on desktop, on mobile only where specified					
TC 03	Enter two letters in the search engine. (Indicate the selected combination of letters Actual result)	Any two letters	Suggestions appear in the form of a dropdown.	Suggestions appeared in the form of a dropdown.	
	Press the search symbol (magnifying glass)		Clicking on the magnifying glass shows: ● Search results for the phrase"<your_typed_phrase>" (<number>) ● Number of items found ● Number of products in categories in the side menu ● The displayed results display 48 units per page by default The displayed results do contain the two letter combination you are looking for	Clicking on the magnifying glass showed: ● Search results for the phrase"<your_typed_phrase>" (<number>) ● Number of items found ● Number of products in categories in the side menu ● The displayed results display 48 units per page by default The displayed results contained the two-letter combination you	

				were looking for	
	Repeat the previous two steps on your mobile device			The previous two steps were repeated on mobile device Android version 13 successfully.	
TC 04	(Desktop only) Change the number of displayed results to 24 in the searched results		The order and visual appearance are unchanged, but the displayed results are reduced to 24 visible items.	The order and visual appearance were unchanged, but the displayed results were reduced to 24 visible items.	
	(Desktop only) Change the number of displayed results to 72 in the searched results		The order and visual appearance remain unchanged, but the displayed results increase to 72 visible items.	The order and visual appearance remained unchanged, but the displayed results increased to 72 visible items.	
TC 05	Change the result layout from squares to list format and back. 		If the arrangement according to squares is chosen, then the goods stand next to each other. If the list format is selected, the items are placed one below the other. Reloading the page does not change the appearance.	The arrangement according to squares was chosen, and the goods were positioned next to each other. The list format was selected, and the items were placed one below the other. Reloading the page did not change the appearance.	
	Repeat previous on mobile device			The previous was repeated on mobile device Android version 13 successfully.	