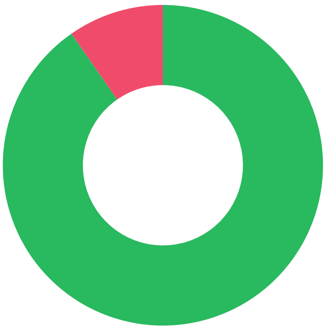
Test run 2025/07/21

Test run to test users, auth, test case and test steps functionalities

Completion chart



Completion stats

- ● Untested (0)
- ● Passed (28)
- ● Failed (3)
- ● Blocked (0)
- ● Retest (0)
- ● Skipped (0)
- ● Invalid (0)

Completion rate

100%

Started by	Start time	Estimated	Time spent
Simakin Viacheslav	2025-07-21 15:54:25	00:02:28.408	00:40:40.424

Environment	Milestone
-	-

DEMO-1 / Authorization

Authorization

Status	Time spent	Assignee
Passed	00:00:34.614	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:00:34.614	Simakin Viacheslav	-

Finish time

2025-07-21 15:55:04

Steps

Step	1
Action	Send POST request to "https://vyacheslav-simakin/login" with username and password as x-www-form-urlencoded data.
Input data	username: mail@mail.com password: 12345678 Curl request: "curl --location 'http://127.0.0.1:8000/login' \\ --header 'Content-Type: application/x-www-form-urlencoded' \\ --data-urlencode 'username=mail@com.com' \\ --data-urlencode 'password=test_pass'"
Expected result	"200 OK" response from server, new access bearer token is present in response.

DEMO-43 / Authorization

Verify that the app doesn't accept expired tokens

Status	**Time spent**	**Assignee**
Passed	00:02:43.462	Simakin Viacheslav

Results

Result 1

Status	**Time spent**	**User**	**Defects**
Passed	00:02:43.462	Simakin Viacheslav	-

Finish time

2025-07-21 15:57:49

Steps

Step	1
Action	Send POST request to "/login" with username and password as x-www-form-urlencoded data.
Input data	username: mail@mail.com password: 12345678 Curl request: "curl --location 'http://127.0.0.1:8000/login' \ --header 'Content-Type: application/x-www-form-urlencoded' \ --data-urlencode 'username=mail@com.com' \ --data-urlencode 'password=test_pass'"
Expected result	"200 OK" response from server, new access bearer token is present in response.

Step	2
Action	Send GET request to "/me"
Input data	Curl request: "curl --location 'http://127.0.0.1:8000/me' \ --header 'Authorization: Bearer \<TOKEN>'\"
Expected result	"200 OK" response from server, user_id, first_name, last_name and email fields are present in response.

Step	3
Action	Wait for token to expire

Step	4
Action	Send GET request to "/me"
Input data	Curl request: "curl --location 'http://127.0.0.1:8000/me' \ --header 'Authorization: Bearer \<TOKEN>'\"
Expected result	"401 Not authenticated" response from server.

DEMO-46 / Authorization

Verify that it is impossible to login with wrong password

Status	Time spent	Assignee
Passed	00:00:40.002	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:00:40.002	Simakin Viacheslav	-

Finish time

2025-07-21 15:58:29

Steps

Step	1
Action	Send POST request to "https://vyacheslav-simakin/login" with username and password of a user, that is in the system, but with wrong password, as x-www-form-urlencoded data.
Input data	username: mail@mail.com password: 123456781 Curl request: "curl --location 'http://127.0.0.1:8000/login' \\ --header 'Content-Type: application/x-www-form-urlencoded' \\ --data-urlencode 'username=mail@com.com' \\ --data-urlencode 'password=123456781'"
Expected result	"401 Unauthorized" response from server, "Incorrect username or password" is present in response.

Verify that it is not possible to login with a user that is not in the system

Status	Time spent	Assignee
Passed	00:00:51.542	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:00:51.542	Simakin Viacheslav	-

Finish time

2025-07-21 15:59:22

Steps

Step	1
Action	Send POST request to "https://vyacheslav-simakin/login" with username and password of a user, that is not in the system, as x-www-form-urlencoded data.
Input data	username: bad@mail.com password: 12345678 Curl request: "curl --location 'http://127.0.0.1:8000/login' \\ --header 'Content-Type: application/x-www-form-urlencoded' \\ --data-urlencode 'username=badmail@com.com' \\ --data-urlencode 'password=test_pass'"
Expected result	"401 Unauthorized" response from server, "Incorrect username or password" is present in response.

DEMO-47 / Authorization

Verify that it is impossible to log in without required fields

Status	Time spent	Assignee
Passed	00:00:48.848	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:00:48.848	Simakin Viacheslav	-

Finish time

2025-07-21 16:00:12

Steps

Step	1
Action	Send POST request to "https://vyacheslav-simakin/login" with username of a user, that is in the system, as x-www-form-urlencoded data.
Input data	username: mail@mail.com Curl request: "curl --location 'http://127.0.0.1:8000/login' \\ --header 'Content-Type: application/x-www-form-urlencoded' \\ --data-urlencode 'username=mail@com.com' \\ --data-urlencode 'password=123456781'"
Expected result	"422" response from server, error about missing password is present in response.

Step	2
Action	Send POST request to "https://vyacheslav-simakin/login" with password of a user, that is in the system, as x-www-form-urlencoded data.
Input data	password: 123456781 Curl request: "curl --location 'http://127.0.0.1:8000/login' \\ --header 'Content-Type: application/x-www-form-urlencoded' \\ --data-urlencode 'username=mail@com.com' \\ --data-urlencode 'password=123456781'"
Expected result	"422" response from server, error about missing username is present in response.

Step	3
Action	Send POST request to "https://vyacheslav-simakin/login" without username and password.
Input data	Curl request: "curl --location 'http://127.0.0.1:8000/login' "
Expected result	"422" response from server, error about missing username and password is present in response.

DEMO-12 / Users

Check user creation

Status	Time spent	Assignee
Passed	00:00:27.185	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:00:27.185	Simakin Viacheslav	-

Finish time

2025-07-21 16:00:40

Steps

Step	1
Action	Send POST request to "http://127.0.0.1:8000/users/" with first_name, last_name, email and password fields.
Input data	first_name: "Fname", last_name: "Lname", email: "newmail30@mail.coa", password: "new_pass Curl request: "curl --location 'http://127.0.0.1:8000/users/' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data-raw '{ "first_name": "Fname", "last_name": "Lname", "email": "newmail30@mail.coa", "password": "new_pass" }'
Expected result	"200 OK" response from server, "User created." response body.

DEMO-32 / Users

Check that trailing and leading spaces in email are removed

Status	Time spent	Assignee
Passed	00:00:34.083	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:00:34.083	Simakin Viacheslav	-

Finish time

2025-07-21 16:01:15

Steps

Step	1
Action	Send POST request to "http://127.0.0.1:8000/users/" with first_name, last_name, email with trailing and leading spaces and password fields.
Input data	first_name: "Fname", last_name: "Lname", email: " newmail32@mail.coa ", password: "new_pass" Curl request: "curl --location 'http://127.0.0.1:8000/users/' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data-raw '{ "first_name": "Fname", "last_name": "Lname", "email": " newmail32@mail.coa ", "password": "new_pass" }'
Expected result	"200 OK" response from server, "User created." response body.

DEMO-13 / Users

Check user deletion

Status	Time spent	Assignee
Passed	00:01:16.660	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:01:16.660	Simakin Viacheslav	-

Finish time

2025-07-21 16:02:33

Steps

Step	1
Action	Delete a user by sending DELETE request to 'http://127.0.0.1:8000/users/{id}"
Input data	Curl request: "curl --location --request DELETE http://127.0.0.1:8000/users/9'\ --header 'Authorization: Bearer \<TOKEN>'"
Expected result	"200 OK" response from server, "User deleted." response body received.

DEMO-14 / Users

Check user updating

Status	Time spent	Assignee
Passed	00:00:35.241	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:00:35.241	Simakin Viacheslav	-

Finish time

2025-07-21 16:03:09

Steps

Step	1
Action	Send PATCH request to "http://127.0.0.1:8000/users/{id}"
Input data	first_name: "Updated fname", last_name: "Updated lname" Curl request: "curl --location --request PATCH http://127.0.0.1:8000/users/4' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data '{ "first_name": "Updated fname", "last_name": "Updated lname" }'"
Expected result	"200 OK" response from server.

DEMO-15 / Users

Check for user list retrieving.

Status	Time spent	Assignee
Passed	00:00:46.536	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:00:46.536	Simakin Viacheslav	-

Finish time

2025-07-21 16:03:57

Steps

Step	1
Action	Send GET request to 'http://127.0.0.1:8000/users/'
Input data	Curl request: "curl --location 'http://127.0.0.1:8000/users/' \ --header 'Authorization: Bearer \<TOKEN>' "
Expected result	"200 OK" response from server with a body that contains list of users with first_name, last_name and user_id for each user received.

DEMO-16 / Users

Check for user retrieving by id

Status	Time spent	Assignee
Passed	00:01:00.646	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:01:00.646	Simakin Viacheslav	-

Finish time

2025-07-21 16:04:59

Steps

Step	1
Action	Send GET request to 'http://127.0.0.1:8000/users/{id} \ --header 'Authorization: Bearer \<TOKEN>'"
Input data	Curl request: "curl --location 'http://127.0.0.1:8000/users/1'"
Expected result	"200 OK" response from server with first_name, last_name and user_id received in the response body.

DEMO-33 / Users

Check for user retrieving by id, that doesn't exist in system.

Status	Time spent	Assignee
Passed	00:00:20.688	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:00:20.688	Simakin Viacheslav	-

Finish time

2025-07-21 16:05:20

Steps

Step	1
Action	Send GET request to "http://127.0.0.1:8000/users/{id}"
Input data	Curl request: "curl --location 'http://127.0.0.1:8000/users/0'\ --header 'Authorization: Bearer \<TOKEN>'"
Expected result	"404 Not Found" response from server.

DEMO-17 / Users

Check for current user info

Status	Time spent	Assignee
Passed	00:00:19.579	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:00:19.579	Simakin Viacheslav	-

Finish time

2025-07-21 16:05:41

Steps

Step	1
Action	Send GET request "http://127.0.0.1:8000/me"
Input data	Curl request: "curl --location 'http://127.0.0.1:8000/me'\ --header 'Authorization: Bearer \<TOKEN>'"
Expected result	"200 OK" response from server, user_id, first_name, last_name and email fields are present in response.

DEMO-29 / Users

Check user creation with email of user that is already in system

Status	Time spent	Assignee
Passed	00:00:46.552	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:00:46.552	Simakin Viacheslav	-

Finish time

2025-07-21 16:06:28

Steps

Step	1
Action	Send POST request to "http://127.0.0.1:8000/users/" with first_name, last_name, email and password fields.
Input data	first_name: "Fname", last_name: "Lname", email: "newmail30@mail.coa", password: "new_pass Curl request: "curl --location 'http://127.0.0.1:8000/users/' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data-raw '{ "first_name": "Fname", "last_name": "Lname", "email": "newmail30@mail.coa", "password": "new_pass" }'
Expected result	"200 OK" response from server, "User created." in response body.

Step	2
Action	Send POST the same request with the same data one more time.
Expected result	"400 Bad request" response from server, "User already exists" in response body.

DEMO-30 / Users

Check user creation with wrong email

Status	Time spent	Assignee
Passed	00:00:34.501	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:00:34.501	Simakin Viacheslav	-

Finish time

2025-07-21 16:07:03

Steps

Step	1
Action	Send POST request to "http://127.0.0.1:8000/users/" with first_name, last_name, password and wrong email fields.
Input data	Wrong email such as " @mail.com" "123" "name@gmail" "mail@mail.", "name@gmail..com", "name..@gmail.com" Curl request: "curl --location 'http://127.0.0.1:8000/users/' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data-raw '{ "first_name": "Fname", "last_name": "Lname", "email": "@mail.coa", "password": "new_pass" }'
Expected result	"400 Bad request" response from server, "value is not a valid email address" is present in the response body.

DEMO-31 / Users

Check user creation without required fields

Status	Time spent	Assignee
Passed	00:01:06.701	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:01:06.701	Simakin Viacheslav	-

Finish time

2025-07-21 16:08:11

Steps

Step	1
Action	Send POST request to "http://127.0.0.1:8000/users/" without first_name, with last_name, email and password fields.
Input data	last_name: "Lname", email: "newmail30@mail.coa", password: "new_pass Curl request: "curl --location 'http://127.0.0.1:8000/users/' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data-raw '{ "last_name": "Lname", "email": "newmail30@mail.coa", "password": "new_pass" }'
Expected result	"400 Bad request" response from server, "msg": "Field required" is in the response body.
Step	2
Action	Send POST request to "http://127.0.0.1:8000/users/" without email, with first_name and password fields.
Input data	first_name: "Fname", password: "new_pass Curl request: "curl --location 'http://127.0.0.1:8000/users/' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data-raw '{ "first_name": "Fname", "password": "new_pass" }'
Expected result	"400 Bad request" response from server, "msg": "Field required" is in the response body.
Step	3
Action	Send POST request to "http://127.0.0.1:8000/users/" without password, with first_name and email fields.

Input data	first_name: "Fname", last_name: "Lname", email: "newmail30@mail.coa", Curl request: "curl --location 'http://127.0.0.1:8000/users/' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data-raw '{ "first_name": "Fname", "last_name": "Lname", "email": "newmail30@mail.coa", }'
Expected result	"400 Bad request" response from server, "msg": "Field required" is in the response body.

DEMO-18 / Test cases

Create test case

Status	Time spent	Assignee
Passed	00:00:35.636	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:00:35.636	Simakin Viacheslav	-

Finish time
2025-07-21 16:08:47

Steps

Step	1
Action	Send POST request to "http://127.0.0.1:8000/test-cases/".
Input data	name:"test case creation", description: "new tc", pre-conditions: "do this and do that", project_id: "e0387296-5af9-4fd0-8f5a-8c376d5d9314 Curl request: "curl --location 'http://127.0.0.1:8000/test-cases/' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data '{"name":"test case creation", "description": "new tc", "pre-conditions": "do this and do that", "project_id": "e0387296-5af9-4fd0-8f5a-8c376d5d9314"}' "
Expected result	"200 OK" response from server

Step	2
Action	Check that test case exists in database.
Expected result	Test case exists.

Step	3
Action	Check that step, associated with test case is created in db.
Expected result	Step exists.

DEMO-21 / Test cases

Update test case

Status	Time spent	Assignee
Passed	00:01:10.059	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:01:10.059	Simakin Viacheslav	-

Finish time

2025-07-21 16:09:59

Steps

Step	1
Action	Send PATCH request to "/test-cases/{id}"
Input data	Curl request: curl --location --request PATCH http://127.0.0.1:8000/test-cases/{id}' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data '{"name":"t edition", "description": "new changed tc", "pre-conditions": "first, cra'\''te user, ", "priority":"LOW"} '
Expected result	"200 OK" response from server

Delete test case

Status	Time spent	Assignee
Passed	00:00:57.207	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:00:57.207	Simakin Viacheslav	-

Finish time

2025-07-21 16:10:57

Steps

Step	1
Action	Send DELETE request to "/test-cases/{id}".
Input data	Curl request: "curl --location --request DELETE http://127.0.0.1:8000/test-cases/0aefbd7d-7975-4c18-892c-17f87c121f7c'\ --header 'Authorization: Bearer \<TOKEN>'"
Expected result	"200 OK" response from server with "Test case deleted." received in the response body.

Get test case with details

Status	Time spent	Assignee
Passed	00:00:56.635	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:00:56.635	Simakin Viacheslav	-

Finish time

2025-07-21 16:11:54

Steps

Step	1
Action	Send GET request to "/test-cases/{id}"
Input data	Curl request: "curl --location 'http://127.0.0.1:8000/test-cases/{id}' \ --header 'Authorization: Bearer \<TOKEN>'\"
Expected result	"200 OK" response from server with name, description, pre_conditions, priority, steps, test_case_id, created_by and project_id received in the response body.

Get list of test cases

Status	Time spent	Assignee
Failed	00:01:27	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Failed	00:01:27	Simakin Viacheslav	D-1 (Open)

Finish time

2025-07-21 16:13:53

Actual result

Missing test_case_id field

Steps

Step	1
Action	Send GET request to "/test-cases/"
Input data	Curl request: "curl --location 'http://127.0.0.1:8000/test-cases'\ --header 'Authorization: Bearer \<TOKEN>'"
Expected result	"200 OK" response from server with list of test cases with their name, test_case_id and priority in the response body.

Create new step in test case

Status	Time spent	Assignee
Passed	00:02:00.222	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:02:00.222	Simakin Viacheslav	-

Finish time

2025-07-21 16:18:22

Steps

Step	1
Action	Send POST request to "/test-cases/{test-case-id}/steps/" with test case id of already existing test case and step id of first step, associated with test case.
Input data	description: "new tc2", expected_result: "do this and do that", prev_step_id: "a7abaf5c-b569-42c4-bc54-d134fe42562c" Curl request: "curl --location 'http://127.0.0.1:8000/test-cases/{test-case-id}/steps/' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data '{ "description": "new tc2", "expected_result": "do this and do that", "prev_step_id": "a7abaf5c-b569-42c4-bc54-d134fe42562c"}'"
Expected result	"200 OK" response from server

DEMO-25 / Test cases

Delete step from test case

Status	Time spent	Assignee
Passed	00:01:04.711	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:01:04.711	Simakin Viacheslav	-

Finish time

2025-07-21 16:19:28

Steps

Step	1
Action	Send DELETE request to "/test-cases/{test-case-id}/steps/{step_id}"
Input data	Curl request: "curl --location --request DELETE 'http://127.0.0.1:8000/{step-id}'\ --header 'Authorization: Bearer \<TOKEN>'\ "
Expected result	"200 OK" response from server received.

Update step in test case

Status	Time spent	Assignee
Passed	00:01:05.552	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:01:05.552	Simakin Viacheslav	-

Finish time

2025-07-21 16:20:34

Steps

Step	1
Action	Send PATCH request to "/test-cases/steps/{id}"
Input data	description: "updated step", expected_result: "do this and do that" Curl request: "curl --location --request PATCH http://127.0.0.1:8000/test-cases/steps/{step-id}' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data '{ "description": "updated step", "expected_result": "do this and do that" }' "
Expected result	"200 OK" response from server received.

DEMO-27 / Test cases

Create new step with step displacement

Status	Time spent	Assignee
Passed	00:01:48.454	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:01:48.454	Simakin Viacheslav	-

Finish time

2025-07-21 16:22:23

Steps

Step	1
Action	Send POST request to "/test-cases/{test-case-id}/steps/" with test case id of already existing test case and step id of first step, associated with test case.
Input data	description: "step number 4 eventually", expected_result: "do this and do that", prev_step_id: "a7abaf5c-b569-42c4-bc54-d134fe42562c" Curl request: "curl --location 'http://127.0.0.1:8000/test-cases/{test-case-id}/steps/' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data '{ "description": "new tc2", "expected_result": "do this and do that", "prev_step_id": "a7abaf5c-b569-42c4-bc54-d134fe42562c"}'"
Expected result	"200 OK" response from server

Step	2
Action	Retrieve test case and verify, that steps are ordered correctly.
Input data	Curl request: "curl --location 'http://127.0.0.1:8000/test-cases/{id}'\ --header 'Authorization: Bearer \<TOKEN>'"
Expected result	• Step, created at first is second in the list;

Step	3
Action	Send POST request to "/test-cases/{test-case-id}/steps/" with test case id of already existing test case and step id of first step as prev_step_id and step_id of step, created in the previous step as next_step_id, associated with test case.

Input data	description: "step number 3 eventually", expected_result: "do this and do that", prev_step_id: "a7abaf5c-b569-42c4-bc54-d134fe42562c" next_step_id: "a7abaf5c-b569-42c4-bc54-d134fe42562c" Curl request: "curl --location 'http://127.0.0.1:8000/test-cases/{test-case-id}/steps/' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data '{ "description": "new tc2", "expected_result": "do this and do that", "prev_step_id": "a7abaf5c-b569-42c4-bc54-d134fe42562c" "prev_step_id": "a7abaf5c-b569-42c4-bc54-d134fe42562c"}'"
Expected result	"200 OK" response from server
Step	4
Action	Retrieve test case and verify, that steps are ordered correctly.
Input data	Curl request: "curl --location 'http://127.0.0.1:8000/test-cases/{id}'\ --header 'Authorization: Bearer \<TOKEN>'\"
Expected result	• Step, created at first is third in the list; • Step, created second is second in the list;
Step	5
Action	Send POST request to "/test-cases/{test-case-id}/steps/" with test case id of already existing test case and step id of first step, associated with test case.
Input data	description: "step number 1 eventually", expected_result: "do this and do that", next_step_id: "a7abaf5c-b569-42c4-bc54-d134fe42562c" Curl request: "curl --location 'http://127.0.0.1:8000/test-cases/{test-case-id}/steps/' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data '{ "description": "new tc2", "expected_result": "do this and do that", "next_step_id": "a7abaf5c-b569-42c4-bc54-d134fe42562c"}'"
Expected result	"200 OK" response from server
Step	6
Action	Retrieve test case and verify, that steps are ordered correctly.
Input data	Curl request: "curl --location 'http://127.0.0.1:8000/test-cases/{id}'\ --header 'Authorization: Bearer \<TOKEN>'"
Expected result	• Step, created at first is forth in the list; • Step, created second is third in the list; • Step, created last is first in the list;

DEMO-28 / Test cases

Delete step with step displacement

Status	Time spent	Assignee
Passed	00:03:10.937	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:03:10.937	Simakin Viacheslav	-

Finish time

2025-07-21 16:25:36

Steps

Step	1
Action	Send DELETE request to "/test-cases/{test-case-id}/steps/{step_id}" with step_id equal to second step from test case.
Input data	Curl request: "curl --location --request DELETE http://127.0.0.1:8000/test-cases/steps/{id}'\ --header 'Authorization: Bearer \<TOKEN>'"
Expected result	"200 OK" response from server received.

Step	2
Action	Retrieve test case and verify, that steps are ordered correctly, and second step was deleted.
Input data	Curl request: "curl --location 'http://127.0.0.1:8000/test-cases/{id}'\ --header 'Authorization: Bearer \<TOKEN>'"
Expected result	Three steps, that initially were number 1, 3 and 4 left in the test case.

Step	3
Action	Send DELETE request to "/test-cases/{test-case-id}/steps/{step_id}" with step_id equal to first step from test case.
Input data	Curl request: "curl --location --request DELETE http://127.0.0.1:8000/test-cases/steps/{id}'\ --header 'Authorization: Bearer \<TOKEN>'"
Expected result	"200 OK" response from server received.

Step	4
Action	Retrieve test case and verify, that steps are ordered correctly, and first step was deleted.
Input data	Curl request: "curl --location 'http://127.0.0.1:8000/test-cases/{id}'\ --header 'Authorization: Bearer \<TOKEN>'"
Expected result	Two steps, that initially were number 3 and 4 left in the test case.

Step	5
Action	Send DELETE request to "/test-cases/{test-case-id}/steps/{step_id}" with step_id equal to last step from test case.
Input data	Curl request: "curl --location --request DELETE 'http://127.0.0.1:8000/test-cases/steps/{id}'\ --header 'Authorization: Bearer \<TOKEN>'"
Expected result	"200 OK" response from server received.

Step	6
Action	Retrieve test case and verify, that steps are ordered correctly, and last step was deleted.
Input data	Curl request: "curl --location 'http://127.0.0.1:8000/test-cases/{id}'\ --header 'Authorization: Bearer \<TOKEN>'"
Expected result	Only one step, that initially was number 3 left in the test case.

Delete last step from test case

Status	Time spent	Assignee
Passed	00:02:35.469	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:02:35.469	Simakin Viacheslav	-

Finish time

2025-07-21 16:28:12

Steps

Step	1
Action	Send DELETE request to "/test-cases/steps/{step_id}"
Input data	Curl request: "curl --location --request DELETE http://127.0.0.1:8000/test-cases/steps/{step-id}'\ --header 'Authorization: Bearer \<TOKEN>'\"
Expected result	"400 Bad Request" response from server with "Can't delete step. Test case must contain at least one step." received in the response body.

DEMO-36 / Test cases

Create new step with displacement and wrong pointer at the start or end

Status	Time spent	Assignee
Failed	00:02:18	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Failed	00:02:18	Simakin Viacheslav	D-2 (Open)

Finish time

2025-07-21 16:31:16

Actual result

Received 200 Ok response from server. It is possible to create new step with id from different test case

Steps

Step	1
Action	Send POST request to "/test-cases/{test-case-id}/steps/" with test case id of already existing test case #1 and step id of first step, associated with test case #2.
Input data	description: "bad step", expected_result: "do this and do that", **prev_step_id:** "a7abaf5c-b569-42c4-bc54-d134fe42562c" Curl request: "curl --location 'http://127.0.0.1:8000/test-cases/{test-case-id}/steps/' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data '{ "description": "new tc2", "expected_result": "do this and do that", "prev_step_id": "a7abaf5c-b569-42c4-bc54-d134fe42562c"}'"
Expected result	"400 Bad request." response from server with "Can't create step. Steps must correspond to test cases." received in the response body.

Step	2
Action	Send POST request to "/test-cases/{test-case-id}/steps/" with test case id of already existing test case #1 and step id of first step, associated with test case #2.
Input data	description: "bad step", expected_result: "do this and do that", **next_step_id:** "a7abaf5c-b569-42c4-bc54-d134fe42562c" Curl request: "curl --location 'http://127.0.0.1:8000/test-cases/{test-case-id}/steps/' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data '{ "description": "new tc2", "expected_result": "do this and do that", "next_step_id": "a7abaf5c-b569-42c4-bc54-d134fe42562c"}'"
Expected result	"400 Bad request." response from server with "Can't create step. Steps must correspond to test cases." received in the response body.

Create new step with displacement and wrong pointers in the middle

Status	Time spent	Assignee
Failed	00:01:59	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Failed	00:01:59	Simakin Viacheslav	-

Finish time

2025-07-21 16:34:50

Actual result

Received 200 Ok response from server. It is possible to create new step with id from different test case (for both next and prev step id)

Steps

Step	1
Action	Send POST request to "/test-cases/{test-case-id}/steps/" with test case id of already existing test case #1 and step id of first step as both next_step and prev_step, leaving second step without a link.
Input data	description: "bad step", expected_result: "do this and do that", **prev_step_id:** "a7abaf5c-b569-42c4-bc54-d134fe42562c" **next_step_id:** "a7abaf5c-b569-42c4-bc54-d134fe42562c" Curl request: "curl --location 'http://127.0.0.1:8000/test-cases/{test-case-id}/steps/' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data '{ "description": "new tc2", "expected_result": "do this and do that", "prev_step_id": "a7abaf5c-b569-42c4-bc54-d134fe42562c", "next_step_id": "a7abaf5c-b569-42c4-bc54-d134fe42562c"}}'"
Expected result	"400 Bad request." response from server with "Can't create step. Steps must correspond to test cases." received in the response body.

DEMO-39 / Test cases

Create a test case without required fields

Status	Time spent	Assignee
Passed	00:02:44.971	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:02:44.971	Simakin Viacheslav	-

Finish time

2025-07-21 16:37:39

Steps

Step	1
Action	Send POST request without name to "/test-cases/"
Input data	description: "new tc", pre-conditions: "do this and do that", project_id: "e0387296-5af9-4fd0-8f5a-8c376d5d9314 Curl request: "curl --location 'http://127.0.0.1:8000/test-cases/' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data '{"name":"test case creation", "description": "new tc", "pre-conditions": "do this and do that", "project_id": "e0387296-5af9-4fd0-8f5a-8c376d5d9314"}' "
Expected result	"400 Bad Request" response from server with error about missing field in the response body.

Step	2
Action	Send POST request without description to "/test-cases/"
Input data	name: "new tc", pre-conditions: "do this and do that", project_id: "e0387296-5af9-4fd0-8f5a-8c376d5d9314 Curl request: "curl --location 'http://127.0.0.1:8000/test-cases/' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data '{"name":"test case creation", "pre-conditions": "do this and do that", "project_id": "e0387296-5af9-4fd0-8f5a-8c376d5d9314"}' "
Expected result	"400 Bad Request" response from server with error about missing field in the response body.

Step	3
Action	Send POST request without project_id to "/test-cases/"

Input data	name:"test case creation", description: "new tc", pre-conditions: "do this and do that" Curl request: "curl --location 'http://127.0.0.1:8000/test-cases/' \ --header 'Authorization: Bearer \<TOKEN>'\ --header 'Content-Type: application/json' \ --data '{"name":"test case creation", "description": "new tc", "pre-conditions": "do this and do that"}' "
Expected result	"400 Bad Request" response from server with error about missing field in the response body.

Update test case with wrong Priority field

Status	Time spent	Assignee
Passed	00:03:19.731	Simakin Viacheslav

Results

Result 1

Status	Time spent	User	Defects
Passed	00:03:19.731	Simakin Viacheslav	-

Finish time

2025-07-21 16:40:59

Steps

Step	1
Action	Send PATCH request to "/test-cases/{id}"
Input data	Curl request: "curl --location --request PATCH http://127.0.0.1:8000/test-cases/{id}' \ --header 'Content-Type: application/json' \ --header 'Authorization: Bearer \<TOKEN>'\ --data '{"priority":"bad"}' "
Expected result	"400 Bad Request" response from server, with "Input should be 'CRITICAL', 'HIGH', 'MEDIUM' or 'LOW'" in the response body.