Test Cases

- My web server -
Software Development Company™

Table of contents

Overview

No.	Requirement	Use Case	Test Case	Name	Type
1	Req. 1	-	TJU.1	Integration	Automated
2	Req. 2	-	TJU.2	Response	Automated
3	-	-	TJU.3	View	Automated
4	-	-	TJU.4	Grouped	Automated
5	Req. 3	UC1	TSS.1.1	Start Server	Manual
6	Req. 3	UC1	TSS.1.2	Wrong Socket	Manual
7	Req. 3	UC1	TSS.1.3	Taken Socket	Manual
8	Req. 5	UC1	TSS.1.4	Access Log Written	Manual
9	Req. 3	UC2	TSS.2.1	Stop Server	Manual
10	Req. 5	UC2	TSS.2.2	Server is stopped - write to access log	Manual
11	Req. 2	UC3	TS.1	HTTP 1.1 Status 200	Manual
12	Req. 2	UC3	TS.2	HTTP 1.1 Status 400	Manual
13	Req. 2	UC3	TS.3	HTTP 1.1 Status 403	Manual
14	Req. 2	UC3	TS.4	HTTP 1.1 Status 404	Manual
15	Req. 2	UC3	TS.5	HTTP 1.1 Status 405	Manual
16	Req. 1	-	TL.1	High Load (1 000 users)	Manual
17	Req. 1	-	TL.2	High Load (50 000 users)	Manual
18	Req. 4	-	TGPL2.0	Test GPL-2.0	Manual
19	Req. 3	-	TOS.1.1	Test Operating System	Manual
20	-	UC1	TV.001	Vulnerability check	Automated
21	-	-	TB1.1	Test Browser	Manual
22	-	-	TP.1	Performance	Manual
23	-	-	TA.1	Acceptance	Aut./Man.

JUnit Tests

Integration Test

TJU.1	Integration
Requirement	Req 1. The web server should be responsive under high load
Use Case	--
Scenario	Confirming the proper functioning of the already existing JUnit-Tests.
Precondition	Server is up and running
Test Steps	1. Run the integration package as JUnit Test
Expected result	All tests pass. A string shows how many times the server is created successfully, fails or throws an exception.
Comments	Automated test package tests\se\lnu\http\integration;

Response Test

TJU.2	Response
Requirement	Req 2. The web server must follow minimum requirements for HTTP 1.1
Use Case	--
Scenario	Confirming the proper functioning of the already existing JUnit-Tests.
Precondition	Server is up and running
Test Steps	1. Run the response package as JUnit Test
Expected result	All tests pass. Response statuses should pop up as expected.
Comments	Automated test package tests\se\lnu\http\response;

View Test

TJU.3	View
Requirement	--
Use Case	--
Scenario	Confirming the proper functioning of the already existing JUnit-Tests. Test if the console shows what is expected from the server.
Precondition	Server is up and running
Test Steps	1. Run the view package as JUnit Test
Expected result	All tests pass. Console shows what is expected from the server.
Comments	Communicate error to the View department to fix it. Rerun the test after that.

Grouped Test

TJU.4	Grouped
Requirement	--
Use Case	--
Scenario	A group of automated tests confirming the proper functioning of the already existing JUnit-Tests.
Precondition	Server is up and running
Test Steps	1. Run the se.lnu.http package as JUnit Test
Expected result	All tests pass.
Comments	All resources that are found will be tested, to see whether the communication between client and server is working as expected. Automated test package tests\se\lnu\http;

UC1 Start Server

Starting Server

TSS.1.1	Start Server
Requirement	Req 3. The web server must work on Linux, Mac, Windows*.
Use Case	UC1 Start Server
Scenario	The Server should always start on all OS after the input of the port socket number and the shared resource container.
Precondition	Server is off.
Test Steps	1. Administrator opens Terminal 2. Navigate to the directory where the .jar file is located 3. Type java –jar MyWebServer.jar PortSocket SharedResource 4. Open a browser 5. Enter "localhost:1091" 6. Press Enter
Expected result	"HTTP Server started" is shown in the terminal window. "It works" image is shown on the web page.
Alternate Scenario	6a. The web server could not be started due to socket was taken a. System presents an error message: "Socket XX was taken" b. Exit Test Case 6b. The web server could not be started due restriction on the shared resource container a. System presents an error message: "No access to folder XX" b. Exit Test Case
Comments	PortSocket: 1091 SharedResource: \D:\lnu\java_courses\MyWebServer-master\tests\se\lnu\http\resources\inner

Wrong Socket

TSS.1.2	Wrong Socket
Requirement	Req 3. The web server must work on Linux, Mac, Windows*.
Use Case	UC1 Start Server
Scenario	The Server should show an error message if a wrong port socket is entered.
Precondition	Server is off.
Test Steps	1. Administrator opens Terminal 2. Navigate to the directory where the .jar file is located 3. Type java –jar MyWebServer.jar PortSocket SharedResource
Expected result	“Enter a valid port 1-65535 and a optional URL” is shown in the terminal window.
Alternate Scenario	3a. The web server starts successfully a. System presents “HTTP Server started” b. Exit Test Case
Comments	PortSocket: 0 SharedResource: \D:\lnu\java_courses\MyWebServer-master\tests\se\lnu\http\resources\inner

Taken Socket

TSS.1.3	Taken Socket
Requirement	Req 3. The web server must work on Linux, Mac, Windows*.
Use Case	UC1 Start Server
Scenario	The Server should show an error message if a taken port socket is entered.
Precondition	Server is off.
Test Steps	1. Administrator opens IDE 2. Enter Arguments: PortSocket SharedResource 3. Run HTTPServerConsole 4. Run HTTPServerConsole again
Expected result	“Port is taken” is shown in the console window.
Alternate Scenario	4a. The web server starts successfully a. System presents “HTTP Server started” b. Exit Test Case
Comments	PortSocket: 1091 SharedResource: \D:\lnu\java_courses\MyWebServer-master\tests\se\lnu\http\resources\inner

Access Log written (when request is sent)

TSS.1.4	Access log written
Requirement	Req 5. The access log should be viewable from a text editor.
Use Case	UC1 Start Server
Scenario	When a request is sent to the server, a note about the sent request is written in the access log.
Precondition	Server is up and running.
Test Steps	1. Send a request to the server 2. A note is written in the access log 3. The access log is viewable in a text-editor
Expected result	Log is created. The access log is viewable in a text-editor.
Alternate Scenario	3a. Log is not created. a. No access log is created b. Exit Test Case 3b. Log is created, but nothing is written to it. a. The log file is created, but stays empty. b. Exit Test Case
Comments	The Log file was not created and no error message popped up either. The requests are only written in the console.

UC2 Stop Server

Stopping Server

TSS.2.1	Stop Server
Requirement	Req 3. The web server must work on Linux, Mac, Windows*.
Use Case	UC2 Stop Server
Scenario	System stops the web server and presents that the webserver has been stopped.
Precondition	Server is off.
Test Steps	1. Administrator opens Terminal 2. Navigate to the directory where the .jar file is located 3. Type java –jar MyWebServer.jar PortSocket SharedResource 4. Open a browser 5. Enter “localhost:1091” 6. Input “stop” in terminal 7. Press Enter
Expected result	“HTTP Server stopped” is shown in the terminal window. Web page will not load.
Alternate Scenario	7a. The web server does not stop a. System is still up and running b. Exit Test Case
Comments	PortSocket: 1091 SharedResource: \D:\lnu\java_courses\MyWebServer-master\tests\se\lnu\http\resources\inner

When Server is stopped – Access log is written

TSS.2.2	Server is stopped –> Access log is written
Requirement	Req 5. The web server must work on Linux, Mac, Windows*.
Use Case	UC2 Stop Server
Scenario	When the server has stopped a note in the access log is written.
Precondition	Server is up and running.
Test Steps	1. Input “stop” in terminal 2. Press Enter 3. Open log.txt
Expected result	“HTTP Server stopped” is shown in the last line of the log file.
Alternate Scenario	3a. The web server does not stop a. System is still up and running b. Exit Test Case 3b. The request is not written in the file a. The log file is created, but the stop request is not on the last line b. Exit Test Case
Comments	No log file was created so there is also no “HTTP Server stopped” on the last line.

UC3 Request shared resource (Req. 2)

TS.1	HTTP 1.1 Status 200
Requirement	Req 2. The web server must follow minimum requirements for HTTP 1.1
Use Case	UC 3 Request Shared Resource
Scenario	The web server should be able to respond with HTML and the status code 200
Precondition	Server running on port 1091
Test Steps	1. HTTP Request Get: /index.html 2. Execute Test
Expected result	Server responds with status code 200
Alternate Scenario	2a: The server responds with a different status code a. System presents a different status code b. Exit Test Case
Comments	Success response tested: 200 - OK Tested with JMeter

TS.2	HTTP 1.1 Status 400
Requirement	Req 2. The web server must follow minimum requirements for HTTP 1.1
Use Case	UC 3 Request Shared Resource
Scenario	The web server should be able to respond with the status code 400
Precondition	Server running on port 1091
Test Steps	1. HTTP Request Get: /index.html 2. Enter “fml” in Body Data 3. Execute Test
Expected result	Server responds with status code 400
Alternate Scenario	2a: The server responds with a different status code a. System presents a different status code b. Exit Test Case
Comments	Failure response tested: 400 - Bad request. Tested with JMeter

TS.3	HTTP 1.1 Status 403
Requirement	Req 2. The web server must follow minimum requirements for HTTP 1.1
Use Case	UC 3 Request Shared Resource
Scenario	The web server should be able to respond with the status code 403
Precondition	Server running on port 1091
Test Steps	1. HTTP Request Get: /../secret.html 2. Execute Test
Expected result	Server responds with status code 403
Alternate Scenario	2a: The server responds with a different status code a. System presents a different status code b. Exit Test Case
Comments	Failure response tested: 403 - Forbidden. Tested with JMeter

TS.4	HTTP 1.1 Status 404
Requirement	Req 2. The web server must follow minimum requirements for HTTP 1.1
Use Case	UC 3 Request Shared Resource
Scenario	The web server should be able to respond with the status code 404
Precondition	Server running on port 1091
Test Steps	1. HTTP Request Get: /sunshine.fml 2. Execute Test
Expected result	Server responds with status code 404
Alternate Scenario	2a: The server responds with a different status code a. System presents a different status code b. Exit Test Case
Comments	Failure response tested: 404 - Not Found Tested with JMeter

TS.5	HTTP 1.1 Status 405
Requirement	Req 2. The web server must follow minimum requirements for HTTP 1.1
Use Case	UC 3 Request Shared Resource
Scenario	The web server should be able to respond with the status code 405
Precondition	Server running on port 1091
Test Steps	1. HTTP Request POST: /indx.html 2. Execute Test
Expected result	Server responds with status code 405
Alternate Scenario	2a: The server responds with a different status code a. System presents a different status code b. Exit Test Case
Comments	Failure response tested: 405 - Method Not Supported Tested with JMeter

Req. 1 Responsive Server under high load (JMeter)

TL.1	High Load
Requirement	Req 1. The web server should be responsive under high load
Use Case	--
Scenario	The web server should be able to respond under high load
Precondition	Server running on port 1091
Test Steps	1. Create 1000 Thread Group Users 2. Set Loop Count to 1 3. Execute Test
Expected result	The Server is still responsive
Alternate Scenario	3.a: The Server is not responsive under high load a. Long response time b. Error message is shown
Comments	Tested with JMeter

TL.2	High Load
Requirement	Req 1. The web server should be responsive under high load
Use Case	--
Scenario	The web server should be able to respond under high load
Precondition	Server running on port 1091
Test Steps	1. Create 100 Thread Group Users 2. Set Loop Count to 500 3. Execute Test
Expected result	The Server is still responsive
Alternate Scenario	3.a: The Server is not responsive under high load a. Long response time b. Error message is shown
Comments	Tested with JMeter

Req. 3 Operating System

TOS.1.1	Test Operating System
Requirement	Req 3. The web server must work on Linux, Mac, Windows*.
Use Case	--
Scenario	Test the server in three different environments, using Linux, Mac and Windows operating systems.
Precondition	Availability of computers with the three operative systems said before.
Test Steps	1. Install and run the server in Windows 2. Install and run the server in Linux 3. Install and run the server in Mac
Expected result	The server should work in all the operative systems
Comments	Communicate to the responsible department to fix the problem in the next iteration.

Req. 4 (GPL-2.0. license)

TGPL2.0	Test GPL-2.0
Requirement	Req 4. The source code should be released under GPL-2.0.
Use Case	--
Scenario	The server should be under a GPL 2.0 open source license.
Precondition	--
Test Steps	1. Check the documentation from the readme file. 2. Check if the license of the server is GPL.
Expected result	The server is under GPL license.
Comments	GPL 2.0 information link: https://www.gnu.org/licenses/old-licenses/gpl-2.0.html MIT information link: https://opensource.org/licenses/MIT

Additional Tests

TV.001	Vulnerability check
Requirement	--
Use Case	UC1 Start Server
Scenario	The web server should pass a minimum vulnerability check during server start.
Precondition	Server should be online.
Test Steps	1. Start server 2. Select a URL with the port 1091 3. Execute test
Expected result	No vulnerabilities are found.
Comments	An attacker may be able to bypass a cross-site scripting filter by encoding their malicious payload in an alternate character set if the server web pages would contain dynamically-generated content from users. (such as logging in with a username and password). Tested with Vega.

TB.1.1.	Test browser
Requirement	--
Use Case	--
Scenario	The server should be accessible in different browsers
Precondition	Server is on
Test Steps	● Open local host: 1091 using Chrome Version 63.0.3239.84 (Official Build) (64-bit) ● Open local host: 1091 using Firefox Quantum Version 57.0.2 (64-bit) ● Open local host: 1091 using Microsoft Edge Version 40.15063.674.0 (64-bit) ● Open local host: 1091 using Microsoft EdgeHTML Version 15.15063 (64-bit)
Expected result	“It works” image is shown on the web page in all the browsers.
Comments	--

TP.1	Performance
Requirement	--
Use Case	--
Scenario	The server should be able to respond well during a performance test
Precondition	Server is on
Test Steps	1. Start server 2. Select a URL with the port 1091 3. Run SmartMeter.io test 4. Wait for results
Expected result	The Server is still responsive
Alternate Scenario	4.a: The Server is not responsive a. Long response time b. Error message is shown
Comments	Tested with SmartMeter.io Communicate to the responsible department to retest and reevaluate this in the next iteration.

TA.1	Acceptance
Requirement	--
Use Case	--
Scenario	The server is tested and ready to deliver.
Precondition	None.
Test Steps	1. Run JUnit tests 2. Run Manual tests
Expected result	All tests succeed.
Comments	The system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.