Test Cases

Author: Xiaohe Zhu

Course: 2DV610 Software testing

1. Manual Test For Use Cases

1.1 UC1.1 The web server is started

TestID: UC1.1
Requirement covered: UC1, REQ3
Test description:
The web server starts while the administrator provides a socket port number and a shared resource container.
Precondition: The argument of the path and port number is clear.
The web server should not be started.
Test input:

1. Set up 9000 as socket port number and bin/se/lnu/http/resources/inner/ as a shared resource container path as an arguments to the class HTTPServerConsole
2. Compile the project.
3. Open Google Chrome and enter http://localhost:9000/ as URL

Expect output:

1. There should be two pictures showing it works in chrome.
2. The console should show “HTTP server started

1.2 UC1.2 The web server is started and writes a note in the access log

TestID: UC1.2
Requirement covered: UC1, REQ3, REQ5
Test description:
The web server starts while the administrator provides a socket port number and a shared resource container, an error message: “Socket XX was taken” should show, and a note is written in the access log file.
Precondition: The argument of the path and port number is clear.
The web server should not be started.
Test input:

1. Set up 9000 as socket port number and bin/se/lnu/http/resources/inner/ as a shared resource container path as an arguments to the class HTTPServerConsole
2. Compile the project.
3. Open Google Chrome and enter http://localhost:9000/ as URL

Expect output:

1. There should be two pictures showing it works in chrome.
2. The console should show “HTTP server started
3. The note should be written in the access log file.

1.3 UC1.3 The web server is not started since the already taken socket

TestID: UC1.3
Requirement covered: UC1, REQ3
Test description: The web server shouldn't be started due to the administrator providing a taken socket port number.
Precondition: The argument of the path and port number is clear.
The web server should not be started.
Test input:

1. Set up 80 as socket port number and bin/se/lnu/http/resources/inner/ as a shared resource container path as an arguments to the class HTTPServerConsole
2. Compile the project.
3. Open Google Chrome and enter http://localhost:80/ as URL

Expect output:

1. The web server shouldn't be started.
2. The console should show "Port is taken"

1.4 UC1.4 The web server is not started since restriction on the shared resource container

TestID:UC1.4
Requirement covered: UC1, REQ3
Test description: The web server shouldn't be started due to the restriction on the shared resource container and the system presents an error message: "No access to folder XX".
Precondition: The argument of the path and port number is clear.
The web server should not be started.
Test input:

1. Set up 9000 as socket port number and \var\www as a shared resource container path as an arguments to the class HTTPServerConsole
2. Compile the project.
3. Open Google Chrome and enter http://localhost:9000/ as URL

Expect output:

1. The web server shouldn't be started.
2. The system presents an error message: "directory does not exists"

1.5 UC1.5 The access log could not be written

TestID: UC1.5
Requirement covered: UC1, REQ3, REQ5
Test description: The web server cannot write to access log file and present the error message "Cannot write to server log file log.txt"
Precondition: The argument of the path and port number is clear.
The web server should not be started.

Deleting existing log.txt

Test input:

4. Set up 9000 as socket port number and bin/se/lnu/http/resources/inner/ as a shared resource container path as an arguments to the class HTTPServerConsole
5. Compile the project.
6. Open Google Chrome and enter http://localhost:9000/ as URL

Expect output:

1. The system presents an error message. “Cannot write to server log file log.txt”

1.6 UC2.1 Stopping Server

TestID: UC2.1
Requirement covered: UC2, REQ3
Test description: The system stops the web server and presents that the webserver has been stopped when a user wants to stop the server.
Precondition: The web server is started.
Test input:

1. Write stop in console.
2. Navigate to site

Expect output:.

1. The system stops the web server and presents that the webserver has been stopped

1.7 UC2.2 Stopping Server and a note in the access log was written

TestID: UC2.2
Requirement covered: UC2, REQ3,REQ5
Test description: The system stops the web server and presents that the webserver has been stopped when a user wants to stop the server and a note in the access log was written
Precondition: The web server is started.
Test input:

1. Write stop in console.
2. Navigate to site.

Expect output:

1. The system stops the web server and presents that the webserver has been stopped
2. A note in the access log was written

1.8 UC3.1 The web server can not find shared resource to the browser and present 404 Not Found

TestID: UC3.1
Requirement covered: UC3, REQ2, REQ3
Test description: The error message of 404 Not Found should be present when the resource cannot be found.
Precondition: The web server is started.

Test input:

1. Deleting the index.html file.
2. Navigate to site.

Expect output:

1. The system presents that the resource cannot be found and shows the 404 Not Found message.

1.9 UC3.2 The system delivers the shared resource and the HTTP 1.1 protocol is 200 OK

TestID: UC3.2
Requirement covered: UC3, REQ2, REQ3
Test description: The system delivers the shared resource and the HTTP 1.1 protocol is 200 OK
Precondition: The web server has not started yet.
The argument of the path and port number is clear.
Test input:

1. Do UC1.1
2. Right click google chrome and open the inspection
3. Open network, localhost header check the message.

Expect output:

The system delivers the shared resource and the HTTP 1.1 protocol is 200 OK

2. Requirement Test

2.1 REQ1.1 The web server should be responsive under high load.

TestID: REQ1.1
Requirement covered: REQ1, REQ3
Test description: The web server should be responsive under 500 threads for a second.
Precondition: The web server is started.
Test input:

1. Open Jmeter.
2. Adding thread groups for MyWebServer and setting the number of the threads to 50\00 and ramp up period of second as 1.
3. Adding HTTP request as sampler and setting server name and ip as localhost and the port number as 9000.
4. Adding a view result trees e as a listener.

Expect output:

1. The web server shouldn't crash and should load under high load.

2.2 REQ3.1The web server must work on Windows.

TestID: REQ3.1
Requirement covered: REQ3
Test description: The web server must work on Windows.
Precondition: The argument of the path and port number is clear.
The web server should not be started.
Test input:

1. Follow the step like UC1.1

Expect output:

1. There should be two pictures showing it works in chrome.
2. The console should show “HTTP server started

2.3 REQ4.1 The source code should be released under GPL-2.0.

TestID: REQ4.1
Requirement covered: REQ3
Test description: The source code should be released under GPL-2.0.
Precondition: None.
Test input:

1. Open the source code and there is a file named LICENSE.
2. Check file

Expect output:
The license should said that this application is under GPL-2.0.