1 Use case 1: Start server

The first use case that will be tested is Use Case 1 which is starting the server. For this, we will need to have a jar file of our Web Server and load it using PowerShell. We use the command line in PowerShell to start the server.

Test ID: T1.1	**Actor:** Administrator
Test Name: Server start	**Requirement covered:** UC1
Test Strategy: Manual testing	**Date:** 2017-12-14

Scenario: Administrator starts server, system asks for socket number and shared resource, admin provides the necessary details and system starts web server on given port.

Pre-conditions: Have project as a jar file and have terminal open.

Post-conditions: Web server has started and there are logs in the access log

Notes: Use java -jar \Users\vikra\Desktop\2DV610\WebServer.jar 8080 "\Users\vikra\Desktop\SoftwareTesting\MyWebServer\tests\se\lnu\http\resources\inner" in terminal (port 8080)

Step	Test step	Test data	Expected result	Actual Result	Status (Pass/Fail)
1	Input the jar file along with the test date onto PowerShell	8080 "\Users\vikra\Desktop \SoftwareTesting\MyWebServer \tests\se\lnu\http\resources\inner"	Server should start with browser showing “it works” image and a message about system starting should be created in log file	The server started and image is on browser but no log file exists	Fail
2	Open a browser and open localhost:8080				

Use Case 1 Alternate Scenarios:

In this section we test a few alternative scenarios for use case 1.

Test ID: T1.2	**Actor:** Administrator
Test Name: Unavailable port	**Requirement covered:** UC1
Test Strategy: Manual Testing	**Date:** 2017-12-14

Scenario: Administrator starts server, system asks for socket number and shared resource, admin provides the necessary details, server does not load since socket number provided is already in use.

Pre-conditions: Have project as a jar file and have terminal open. Check what ports are taken using netstat -an

Post-conditions: Web server does not load and loading localhost on browser comes empty

Notes: Use java -jar \Users\vikra\Desktop\2DV610\WebServer.jar 455 "\Users\vikra\Desktop\SoftwareTesting\MyWebServer\tests\se\lnu\http\resources\inner" in terminal (port 445 which is used by svchost.exe)

Step	Test step	Test data	Expected result	Actual Result	Status (Pass/Fail)
1	Input the jar file along with the test date onto PowerShell	445 "\Users\vikra\Desktop \SoftwareTesting\MyWebServer \tests\se\lnu\http\resources\inner"	PowerShell comes with error “Port is taken” and browser comes up with “This site can’t be reached”	PowerShell comes with error and the site cannot be reached through browser	Pass
2	Load localhost:445 on browser				

Test ID: T1.3	**Actor:** Administrator
Test Name: Shared resource before socket param.	**Requirement covered:** UC1
Test Strategy: Manual Testing	**Date:** 2017-12-15

Scenario: Administrator starts server, system asks for socket number and shared resource, admin provides the necessary details (shared resource first), server does not load since argument order is wrong.

Pre-conditions: Have project as a jar file and have terminal open. Check what ports are taken using netstat -an

Post-conditions: Error shown up on PowerShell

Notes: Use java -jar \Users\vikra\Desktop\2DV610\WebServer.jar "\Users\vikra\Desktop\SoftwareTesting\MyWebServer\tests\se\lnu\http\resources\inner" 8080 in terminal (port 8080 is empty)

Step	Test step	Test data	Expected result	Actual Result	Status (Pass/Fail)
1	Input the jar file along with the test date onto PowerShell	"\Users\vikra\Desktop \SoftwareTesting\MyWebServer \tests\se\lnu\http\resources\inner" 8080	PowerShell comes with error “Enter a valid port between 1-65535”	PowerShell comes with error	Pass

Test ID: T1.4	**Actor:** Administrator
Test Name: Incorrect socket number	**Requirement covered:** UC1
Test Strategy: Manual Testing	**Date:** 2017-12-15

Scenario: Administrator starts server, system asks for socket number and shared resource, admin provides the necessary details, server does not load since socket number is not valid.

Pre-conditions: Have project as a jar file and have terminal open. Check what ports are taken using netstat -an

Post-conditions: Error shown up on PowerShell

Notes: Use java -jar \Users\vikra\Desktop\2DV610\WebServer.jar 0 "\Users\vikra\Desktop\SoftwareTesting\MyWebServer\tests\se\lnu\http\resources\inner" (socket number 0)

Step	Test step	Test data	Expected result	Actual Result	Status (Pass/Fail)
1	Input the jar file along with the test date onto PowerShell	0 "\Users\vikra\Desktop \SoftwareTesting\MyWebServer \tests\se\lnu\http\resources\inner"	PowerShell comes with error “Enter a valid port between 1-65535”	PowerShell comes with error	Pass

Test ID: T1.5	**Actor:** Administrator
Test Name: Incorrect shared resource	**Requirement covered:** UC1
Test Strategy: Manual Testing	**Date:** 2017-12-16

Scenario: Administrator starts server, system asks for socket number and shared resource, admin provides the necessary details, server loads but no image is shown since invalid shared resource.

Pre-conditions: Have project as a jar file and have terminal open. Check what ports are taken using netstat -an

Post-conditions: Error shown up on PowerShell

Notes: Use java -jar \Users\vikra\Desktop\2DV610\WebServer.jar 8080 "\Users\vikra\Desktop\”

Step	Test step	Test data	Expected result	Actual Result	Status (Pass/Fail)
1	Input the jar file along with the test date onto PowerShell	8080 "\Users\vikra\Desktop\”	Server should start with browser showing error “404 Not Found”	Server starts and browser shows error	Pass
2	Load localhost:8080 on browser				

Test ID: T1.6	**Actor:** Administrator
Test Name: Protected shared resource	**Requirement covered:** UC1
Test Strategy: Manual Testing	**Date:** 2017-12-16

Scenario: Administrator starts server, system asks for socket number and shared resource, admin provides the necessary details, server loads but no image is shown since invalid shared resource.

Pre-conditions: Have project as a jar file and have terminal open. Check what ports are taken using netstat -an

Post-conditions: Server should not run

Notes: Use java -jar \Users\vikra\Desktop\2DV610\WebServer.jar 8080 "\Users\vikra\Desktop\SoftwareTesting\MyWebServer\tests\se\lnu\http\resources\inner\Protected". Created a ”Protected” folder and revoke all permissions from the system

Step	Test step	Test data	Expected result	Actual Result	Status (Pass/Fail)
1	Input the jar file along with the test date onto PowerShell	8080 "\Users\vikra\Desktop\Software Testing\MyWebServer\tests\se\lnu\http\resources\inner\Protected"	Console should presents an error message: “No access to folder XX”	Server starts up	Fail
2	Load localhost:8080 on browser				

Test ID: T1.7	**Actor:** Administrator
Test Name: Writing to txt file	**Requirement covered:** UC1
Test Strategy: Manual Testing	**Date:** 2017-12-16

Scenario: Administrator starts server, system asks for socket number and shared resource, admin provides the necessary details, server loads and contents are written to txt file instead of terminal.

Pre-conditions: Have project as a jar file and have terminal open. Check what ports are taken using netstat -an

Post-conditions: Should not create any text file

Notes: Use java -jar \Users\vikra\Desktop\2DV610\WebServer.jar 8080 "\Users\vikra\Desktop\SoftwareTesting\MyWebServer\tests\se\lnu\http\resources\inner\" > "C:\Users\vikra\Desktop\SoftWareTesting\MyWebServer\tests\se\lnu\http\resources\log.txt" (> allows creation of new file

Step	Test step	Test data	Expected result	Actual Result	Status (Pass/Fail)
1	Input the jar file along with the test date onto PowerShell	8080 "\Users\vikra\Desktop\SoftwareTesting\MyWebServer\tests\se\lnu\http\resources\inner\" > "\Users\vikra\Desktop\SoftWareTesting\MyWebServer\tests\se\lnu\http\resources\log.txt"	Should present an error “Cannot write to server log file log.txt”	Log file is created	Fail
2	Check log.txt file in the given directory				

2 Use Case 2: Stop Server

In this section we manually test the stopping of the server by the admin through PowerShell. This is done by inputting stop into the command line and it shuts down the server.

Test ID: T2.1	**Actor:** Administrator
Test Name: Stopping server	**Requirement covered:** UC2
Test Strategy: Manual Testing	**Date:** 2017-12-16

Scenario: Administrator requests to shutdown server using stop in command line and server is shut down leaving a message on terminal.

Pre-conditions: Have server up and running.

Post-conditions: Server should not be running and inaccessible.

Notes: Make sure the server stops using the stop command

Step	Test step	Test data	Expected result	Actual Result	Status (Pass/Fail)
1	Load localhost:8080 to see if server is running	stop	PowerShell should print “HTTP Server accept thread stopped” and “HTTP Server stopped”. The server should be shut down. There should a note in the access log that the server was stopped.	PowerShell prints message and server stopped.	Fail
2	Type stop in PowerShell terminal				
3	Load localhost:8080 on browser to check if server has been shut down				

Use Case 2 Alternate scenarios:

This section contains an alternate scenario to stopping the server where the administrator types STOP (all capital). Another alternate scenario that would yield in the same result is if any one of the letters were capitalized.

Test ID: T2.2	**Actor:** Administrator
Test Name: Using capital STOP to stop the server	**Requirement covered:** UC2
Test Strategy: Manual Testing	**Date:** 2017-12-16

Scenario: Administrator requests to shutdown server using STOP in command line and server is shut down leaving a message on terminal.

Pre-conditions: Have server up and running.

Post-conditions: Server should be running

Notes: Server should not stop on wrong commands.

Step	Test step	Test data	Expected result	Actual Result	Status (Pass/Fail)
1	Load localhost:8080 to see if server is running	STOP	The server should not stop running	The server is still running	Pass
2	Type STOP in PowerShell terminal				
3	Load localhost:8080 on browser to check if server has been shut down				

3 Use Case 3: Request Shared Resource

This section covers test cases to match with Requirement 3, that the server must follow minimum requirements for HTTP 1.1.

Test ID: T3.1	**Actor:** Browser
Test Name: Response 200 OK	**Requirement covered:** UC3 and Requirement 2
Test Strategy: Manual Testing	**Date:** 2017-12-17

Scenario: Browser wants to access a shared resource, system delivers it to the browser and keeps a log in log file

Pre-conditions: Have server up and running and JMeter should be installed

Post-conditions:

Notes: Should follow HTTP 1.1 requirements, use GET index.html

Step	Test step	Test data	Expected result	Actual Result	Status (Pass/Fail)
1	Set up JMeter HTTP Request with server name localhost and port number 8080	Server name: localhost Port number: 8080 GET index.html	Response code 200 and response message OK should be returned and should be kept in the access log as well	Both response code and message were returned, there is no access log	Fail
2	Add result listeners and summary listener to the test				
3	Start the test and after completion, check the results and summary				

Test ID: T3.2	**Actor:** Browser
Test Name: Response 400 Bad Request	**Requirement covered:** UC3 and Requirement 2
Test Strategy: Function Testing	**Date:** 2017-12-17

Scenario: Browser wants to access a shared resource, system delivers it to the browser and keeps a log in log file

Pre-conditions: Have server up and running and JMeter should be installed

Post-conditions:

Notes: Should follow HTTP 1.1 requirements, use GET #index.html

Step	Test step	Test data	Expected result	Actual Result	Status (Pass/Fail)
1	Set up JMeter HTTP Request with server name localhost and port number 8080 and include #index.html in the path	Server name: localhost Port number: 8080 GET #index.html	Response code 400 and response message Bad Request should be returned and should be kept in the access log as well	Both response code and message were not returned but there is no access log	Fail
2	Add result listeners and summary listener to the test				
3	Start the test and after completion, check the results and summary				

Test ID: T3.3	**Actor:** Browser
Test Name: Response 403 Forbidden	**Requirement covered:** UC3 and Requirement 2
Test Strategy: Function Testing	**Date:** 2017-12-18

Scenario: Browser wants to access a shared resource, system delivers it to the browser and keeps a log in log file

Pre-conditions: Have server up and running and JMeter should be installed

Post-conditions:

Notes: Should follow HTTP 1.1 requirements, use GET ../secret.html

Step	Test step	Test data	Expected result	Actual Result	Status (Pass/Fail)
1	Set up JMeter HTTP Request with server name localhost and port number 8080 and include ../secret.html in the path	Server name: localhost Port number: 8080 GET ../secret.html	Response code 403 and response message Forbidden should be returned and should be kept in the access log as well	Both response code and message were returned and there is no access log	Fail
2	Add result listeners and summary listener to the test				
3	Start the test and after completion, check the results and summary				

Test ID: T3.4	**Actor:** Browser
Test Name: Response 404 Not Found	**Requirement covered:** UC3 and Requirement 2
Test Strategy: Function Testing	**Date:** 2017-12-18

Scenario: Browser wants to access a shared resource, system delivers it to the browser and keeps a log in log file

Pre-conditions: Have server up and running and JMeter should be installed

Post-conditions:

Notes: Should follow HTTP 1.1 requirements, use GET index.txt

Step	Test step	Test data	Expected result	Actual Result	Status (Pass/Fail)
1	Set up JMeter HTTP Request with server name localhost and port number 8080 and include index.txt in the path	Server name: localhost Port number: 8080 GET index.txt	Response code 404 and response message Not Found should be returned and should be kept in the access log as well	Both response code and message were returned, but there is no access log	Fail
2	Add result listeners and summary listener to the test				
3	Start the test and after completion, check the results and summary				

Test ID: T3.5	**Actor:** Browser
Test Name: Response 500 Internal Server Error	**Requirement covered:** UC3 and Requirement 2
Test Strategy: Function Testing	**Date:** 2017-12-19

Scenario: Browser wants to access a shared resource, system delivers it to the browser and keeps a log in log file

Pre-conditions: Have server up and running and JMeter should be installed

Post-conditions:

Notes: Should follow HTTP 1.1 requirements, use GET ../inner and images

Step	Test step	Test data	Expected result	Actual Result	Status (Pass/Fail)
1	Set up JMeter HTTP Request with server name localhost and port number 8080 and include ../inner in the path	Server name: localhost Port number: 8080 GET ../inner	Response code 500 and response message Internal Server Error should be returned and should be kept in the access log as well	Both response code and message were not returned and there is no access log	Fail
2	Add result listeners and summary listener to the test				
3	Start the test and after completion, check the results and summary				
4	Repeat steps 1-3 with images in the path	GET images	Response code and error should be printed as well as a log in the access log	No response code or message and no access log	Fail

4 Load and Performance Testing:

This section tests how well the server responds upon a heavy load of virtual users over a period and examines the responsiveness, stability and speed of the server.

Test ID: T4.1	**Actor:** Tester
Test Name: Response under high load	**Requirement covered:** Requirement 1
Test Strategy: Load and Performance Testing	**Date:** 2017-12-19

Scenario: Many clients try to access the webserver at the same time

Pre-conditions: Have server up and running

Post-conditions:

Notes: Testing with 1000 connections at 20 seconds, all response codes should be 200 and response time

Step	Test step	Test data	Expected result	Actual Result	Status (Pass/Fail)
1	Set number of threads in JMeter Thread Group to 1000, set Ramp-Up Period to 20	Number of Threads (users): 1000 Ramp-Up Period: 20	All response codes should be 200 and response time should not be higher than 2000 ms (2s)	All response codes were 200 and the max response time was 120 ms	Pass
2	Set up JMeter HTTP Request with server name localhost and port number 8080 and include index.html in the path	Server name: localhost Port number: 8080 GET index.html			
3	Add result listeners and summary listener to the test				

5 Security testing:

This section tests the secureness of the web server by checking if it accessible through https as well since the consensus is that https is more secure than http.

Test ID: T5.1	**Actor:** Tester
Test Name: Testing the server with https	**Requirement covered:** Extra
Test Strategy: Manual Testing	**Date:** 2017-12-20

Scenario: User tries to access the server using https instead of http

Pre-conditions: Have server up and running

Post-conditions:

Notes: Testing server with https instead of http to determine security

Step	Test step	Test data	Expected result	Actual Result	Status (Pass/Fail)
1	After the server is running, open a browser and access the given test data	https://localhost:8080/	The server should run on https	The server only runs on http	Fail