



# Manual Test Cases **My Web Server**

Version 1.0

Dated: 22<sup>nd</sup> December 2017

---

# Functional Testing

Test case 1.0 The server starts on Windows 10

<b>Test ID:</b> 1.0		<b>Tester:</b> Test-leader		
<b>Test Name:</b> Web server on Windows 10		<b>Requirement Covered:</b> 1		
<b>Test Type:</b> Functional Testing		<b>Execution Date:</b> 18/12/2017		
<b>Description:</b> Run the web server on Windows 10				
<b>Pre-Condition:</b> <ul style="list-style-type: none"><li>- Have java IDE</li><li>- Have the web server code</li></ul>				
<b>Post-Condition:</b> <ul style="list-style-type: none"><li>- Run the web server</li></ul>				
Step	Action	Expected Outcome	Actual Outcome	Pass/Fail
1	Set up port and path as arguments to the class HTTPServerConsole			
2	Compile the project	The web server should start		
3	Open a browser and request the URL = localhost:portnumber	The pictures “it works” appears in the browser		

Test case 1.1 The server starts on Windows 10 and a note written to the access log file

<b>Test ID:</b> 1.1		<b>Tester:</b> Test-leader		
<b>Test Name:</b> Web server on Windows 10 and write a note		<b>Requirement Covered:</b> 1		
<b>Test Type:</b> Functional Testing		<b>Execution Date:</b> 18/12/2017		
<b>Description:</b> Run the web server on Windows 10 and a note writes				
<b>Pre-Condition:</b> <ul style="list-style-type: none"><li>- Have java IDE</li><li>- Have the web server code</li></ul>				
<b>Post-Condition:</b> <ul style="list-style-type: none"><li>- Run the web server</li><li>- A note written to the access log file</li></ul>				
Step	Action	Expected Outcome	Actual Outcome	Pass/Fail
1	Set up port and path as arguments to the class HTTPServerConsole			
2	Compile the project	The web server should start		

3	Open a browser and request the URL = localhost:portnumber	The pictures “it works” appears in the browser and a note written to the access log file		
---	---	--	--	--

Test case 1.2 The web server could not be started due to socket was taken

<b>Test ID:</b> 1.2		<b>Tester:</b> Test-leader		
<b>Test Name:</b> Socket was taken		<b>Requirement Covered:</b> 1		
<b>Test Type:</b> Functional Testing		<b>Execution Date:</b> 18/12/2017		
<b>Description:</b> Couldn't run the server due to socket was taken				
<b>Pre-Condition:</b> <ul style="list-style-type: none"><li>- Have java IDE</li><li>- Have the web server code</li></ul>				
<b>Post-Condition:</b> <ul style="list-style-type: none"><li>- An error message: "Socket XX was taken" written in the console</li></ul>				
Step	Action	Expected Outcome	Actual Outcome	Pass/Fail
1	Set up port and path as arguments to the class HTTPServerConsole			
2	Compile the project	The web server should start		
3	Compile the project again	An error message: "Socket XX was taken" written in the console		

Test case 1.3 The web server could not be started due to restriction on shared resource

<b>Test ID:</b> 1.3		<b>Tester:</b> Test-leader		
<b>Test Name:</b> Restriction on shared resource		<b>Requirement Covered:</b> 1		
<b>Test Type:</b> Functional Testing		<b>Execution Date:</b> 18/12/2017		
<b>Description:</b> Couldn’t run the server due to restriction on shared resource				
<b>Pre-Condition:</b> <ul style="list-style-type: none"><li>- Have java IDE</li><li>- Have the web server code</li></ul>				
<b>Post-Condition:</b> An error message: “No access to folder XX” written in the console				
<b>Step</b>	<b>Action</b>	<b>Expected Outcome</b>	<b>Actual Outcome</b>	<b>Pass/Fail</b>
1	Change the permission to the resource container to none.			

2	Set up port and path as arguments to the class HTTPServerConsole			
3	Compile the project	The web server should not start and an error message: "No access to folder XX" written in the console		

Test case 1.4 Wrong with the access log file

<b>Test ID:</b> 1.4		<b>Tester:</b> Test-leader		
<b>Test Name:</b> Couldn't run Web server and a wrong with the access log file		<b>Requirement Covered:</b> 1		
<b>Test Type:</b> Functional Testing		<b>Execution Date:</b> 18/12/2017		
<b>Description:</b> Couldn't run the web server and the system presents an error message "Cannot write to server log file log.txt"				
<b>Pre-Condition:</b> <ul style="list-style-type: none"><li>- Have java IDE</li><li>- Have the web server code</li></ul>				
<b>Post-Condition:</b> <ul style="list-style-type: none"><li>- An error message "Cannot write to server log file log.txt" written in the console</li></ul>				
<b>Step</b>	<b>Action</b>	<b>Expected Outcome</b>	<b>Actual Outcome</b>	<b>Pass/Fail</b>
1	Set up port and path as arguments to the class HTTPServerConsole			
2	Compile the project	The web server shouldn't start and an error message "Cannot write to server log file log.txt" written in the console		

Test case 1.5 Stop the web server

<b>Test ID:</b> 1.5	<b>Tester:</b> Test-leader
<b>Test Name:</b> Stop the web server	<b>Requirement Covered:</b> 2
<b>Test Type:</b> Functional Testing	<b>Execution Date:</b> 18/12/2017
<b>Description:</b> Stop the web server successfully	
<b>Pre-Condition:</b> <ul style="list-style-type: none"><li>- The web server has been started</li></ul>	
<b>Post-Condition:</b> <ul style="list-style-type: none"><li>- The web server should stop</li></ul>	

Step	Action	Expected Outcome	Actual Outcome	Pass/Fail
1	Stop the server by clicking the stop-sign in the IDE.	The web server should stop		

Test case 1.6 Stop the web server and write a note to the access log file

<b>Test ID:</b> 1.6		<b>Tester:</b> Test-leader		
<b>Test Name:</b> Stop the web server		<b>Requirement Covered:</b> 2		
<b>Test Type:</b> Functional Testing		<b>Execution Date:</b> 18/12/2017		
<b>Description:</b> Stop the web server successfully and a note should be written				
<b>Pre-Condition:</b>				
- The web server has been started				
<b>Post-Condition:</b>				
- The web server should stop				
- A note written to the access log file				
<b>Step</b>	<b>Action</b>	<b>Expected Outcome</b>	<b>Actual Outcome</b>	<b>Pass/Fail</b>
1	Stop the server by clicking the stop-sign in the IDE.	The web server should stop, and a note should be written in the access log file		

Test case 1.7 The access log file is viewable

<b>Test ID:</b> 1.7		<b>Tester:</b> Test-leader		
<b>Test Name:</b> Access log file viewable		<b>Requirement Covered:</b> 5		
<b>Test Type:</b> Functional Testing		<b>Execution Date:</b> 18/12/2017		
<b>Description:</b> The access log file is viewable from a text editor				
<b>Pre-Condition:</b>				
<ul style="list-style-type: none"><li>- The web server has been started at least once</li><li>- Have a text editor</li></ul>				
<b>Post-Condition:</b>				
<b>Step</b>	<b>Action</b>	<b>Expected Outcome</b>	<b>Actual Outcome</b>	<b>Pass/Fail</b>
1	Open the folder “.metadata” in the server folder			
2	Open the file “.log” in the text editor	The file can be opened by a text editor, and the content is viewable.		

Test case 1.8 Successful access the recourse folder

<b>Test ID:</b> 1.8		<b>Tester:</b> Test-leader		
<b>Test Name:</b> Successful access the recourse folder		<b>Requirement Covered:</b> 3 - 4		
<b>Test Type:</b> Functional Testing		<b>Execution Date:</b> 19/12/2017		
<b>Description:</b> Browser Successfully access the recourse folder and show the content of the “inner” folder				
<b>Pre-Condition:</b> <ul style="list-style-type: none"><li>- Having files in the “inner” folder</li></ul>				
<b>Post-Condition:</b> <ul style="list-style-type: none"><li>- Show the content in the browser</li></ul>				
Step	Action	Expected Outcome	Actual Outcome	Pass/Fail
1	Set up port and path as arguments to the class HTTPServerConsole			
2	Open a browser and request the URL = localhost:portnumber	<ul style="list-style-type: none"><li>- Browser show the content of the “inner” folder</li></ul>		

Test case 1.9 Successful access the recourse folder and write a note to the access log

<b>Test ID:</b> 1.9		<b>Tester:</b> Test-leader		
<b>Test Name:</b> Successful access the recourse folder and write a note to access log		<b>Requirement Covered:</b> 3 - 4		
<b>Test Type:</b> Functional Testing		<b>Execution Date:</b> 19/12/2017		
<b>Description:</b> Browser Successfully access the recourse folder and show the content of the “inner” folder and a note should be written to the access log indicating this.				
<b>Pre-Condition:</b> <ul style="list-style-type: none"><li>- Having files in the “inner” folder</li></ul>				
<b>Post-Condition:</b> <ul style="list-style-type: none"><li>- Show the content in the browser</li><li>- Write a note to the access log file</li></ul>				
Step	Action	Expected Outcome	Actual Outcome	Pass/Fail
1	Set up port and path as arguments to the class HTTPServerConsole			
2	Open a browser and request the URL = localhost:portnumber	<ul style="list-style-type: none"><li>- Browser show the content of the “inner” folder</li><li>- A note written to the access log file</li></ul>		

Test case 1.10 404 Not found error

<b>Test ID:</b> 1.10		<b>Tester:</b> Test-leader		
<b>Test Name:</b> 404 Not found error		<b>Requirement Covered:</b> 3 - 4		
<b>Test Type:</b> Functional Testing		<b>Execution Date:</b> 19/12/2017		
<b>Description:</b> Resource folder not found, and the browser shows 404 Not found error.				
<b>Pre-Condition:</b>				
- The web server has been started				
<b>Post-Condition:</b>				
- The browser shows 404 Not found error				
<b>Step</b>	<b>Action</b>	<b>Expected Outcome</b>	<b>Actual Outcome</b>	<b>Pass/Fail</b>
1	Open a browser and request the URL = localhost:portnumber/XX, where XX is a file name isn't in the folder "inner"	- Browser show the error 404 Not found		

Test case 1.11 404 Not found error and write a note in the access log

<b>Test ID:</b> 1.11		<b>Tester:</b> Test-leader		
<b>Test Name:</b> 404 Not found error and write a note to access log		<b>Requirement Covered:</b> 3 - 4		
<b>Test Type:</b> Functional Testing		<b>Execution Date:</b> 19/12/2017		
<b>Description:</b> Resource folder not found, and the browser shows 404 Not found error and a note should be written in the access log				
<b>Pre-Condition:</b>				
- The web server has been started				
<b>Post-Condition:</b>				
- The browser shows 404 Not found error				
- A note written in the access log				
<b>Step</b>	<b>Action</b>	<b>Expected Outcome</b>	<b>Actual Outcome</b>	<b>Pass/Fail</b>
1	Open a browser and request the URL = localhost:portnumber/XX, where XX is a file name isn't in the folder "inner"	<ul style="list-style-type: none"><li>- Browser show the error 404 Not found</li><li>- A note written in the access log indicating this</li></ul>		

Test case 1.12 403 Forbidden error and write a note in the access log

<b>Test ID:</b> 1.12		<b>Tester:</b> Test-leader		
----------------------	--	----------------------------	--	--

<b>Test Name:</b> 403 Forbidden error and write a note to access log		<b>Requirement Covered:</b> 3 - 4		
<b>Test Type:</b> Functional Testing		<b>Execution Date:</b> 20/12/2017		
<b>Description:</b> Shared resource is outside the shared resource container which is “inner” folder, the browser shows 403 Not Forbidden error, and a note should be written in the access log				
<b>Pre-Condition:</b>				
- The web server has been started				
<b>Post-Condition:</b>				
- The browser shows 403 Forbidden error				
- A note written in the access log				
<b>Step</b>	<b>Action</b>	<b>Expected Outcome</b>	<b>Actual Outcome</b>	<b>Pass/Fail</b>
1	Open a browser and request the URL = localhost:portnumber/./secret.html, where the file secret.html is outside the folder “inner”.	<ul style="list-style-type: none"><li>- Browser show the error 403 Forbidden</li><li>- A note written in the access log indicating this</li></ul>		

Test case 1.13 400 Bad request error and write a note in the access log

<b>Test ID:</b> 1.13		<b>Tester:</b> Test-leader		
<b>Test Name:</b> 400 Bad request error and write a note to access log		<b>Requirement Covered:</b> 3 - 4		
<b>Test Type:</b> Functional Testing		<b>Execution Date:</b> 20/12/2017		
<b>Description:</b> Request is invalid or malformed, the browser shows 400 Bad request error, and a note should be written in the access log				
<b>Pre-Condition:</b>				
- The web server has been started				
<b>Post-Condition:</b>				
- The browser shows 400 Bad request error				
- A note written in the access log				
<b>Step</b>	<b>Action</b>	<b>Expected Outcome</b>	<b>Actual Outcome</b>	<b>Pass/Fail</b>
1	Open a browser and request the URL = localhost:portnumber./index.html	<ul style="list-style-type: none"><li>- Browser show the error 400 Bad request</li><li>- A note written in the access log indicating this</li></ul>		



Test case 1.14 An error when trying to process the request.

<b>Test ID:</b> 1.14		<b>Tester:</b> Test-leader		
<b>Test Name:</b> Error when trying to process the request		<b>Requirement Covered:</b> 3 - 4		
<b>Test Type:</b> Functional Testing		<b>Execution Date:</b> 20/12/2017		
<b>Description:</b> Server encounters an error when trying to process the request, and a note should be written in the access log				
<b>Pre-Condition:</b> <ul style="list-style-type: none"><li>- The web server has been started</li></ul>				
<b>Post-Condition:</b> <ul style="list-style-type: none"><li>- A note written in the access log that the system presents that the system has an internal error</li></ul>				
Step	Action	Expected Outcome	Actual Outcome	Pass/Fail
1	Open a browser and request the URL = localhost:portnumber/images	<ul style="list-style-type: none"><li>- System present an internal error</li><li>- A note written in the access log indicating this</li></ul>		

## Unit Testing

Test case 2.0 Run the existing JUnit test suites

<b>Test ID:</b> 2.0		<b>Tester:</b> Test-leader		
<b>Test Name:</b> JUnit tests		<b>Requirement Covered:</b> 7		
<b>Test Type:</b> Unit Testing		<b>Execution Date:</b> 21/12/2017		
<b>Description:</b> Run the existing unit suites				
<b>Pre-Condition:</b>				
- The web server has been started				
<b>Post-Condition:</b>				
<b>Step</b>	<b>Action</b>	<b>Expected Outcome</b>	<b>Actual Outcome</b>	<b>Pass/Fail</b>
1	Start Junit tests of the project	<ul style="list-style-type: none"><li>- All tests run</li><li>- All tests pass successfully</li></ul>		

## Integration Testing

Test case 3.0 Run the existing automated JUnit test suites

<b>Test ID:</b> 3.0			<b>Tester:</b> Test-leader	
<b>Test Name:</b> Integration Junit tests			<b>Requirement Covered:</b> 8	
<b>Test Type:</b> Integration Testing			<b>Execution Date:</b> 21/12/2017	
<b>Description:</b> Run the existing automated integration test suites				
<b>Pre-Condition:</b>				
- The web server has been started				
<b>Post-Condition:</b>				
<b>Step</b>	<b>Action</b>	<b>Expected Outcome</b>	<b>Actual Outcome</b>	<b>Pass/Fail</b>
1	Start the Junit tests of the integration folder in the project	<ul style="list-style-type: none"><li>- All tests run</li><li>- All tests pass successfully</li></ul>		

## Stress Testing

Test case 4.0 Stress test

Test ID: 4.0			Tester: Test-leader	
Test Name: Stress test			Requirement Covered: 6	
Test Type: Stress testing			Execution Date: 21/12/2017	
Description: Test the server under high load				
Pre-Condition:				
<ul style="list-style-type: none"><li>- The web server has been started</li><li>- Have JMeter</li></ul>				
Post-Condition:				
Step	Action	Expected Outcome	Actual Outcome	Pass/Fail
1	Test the server with 500 threads under 1s	The test pass without errors		

Test case 4.1 Stress test

<b>Test ID:</b> 4.1	<b>Tester:</b> Test-leader
<b>Test Name:</b> Stress test	<b>Requirement Covered:</b> 6
<b>Test Type:</b> Stress testing	<b>Execution Date:</b> 21/12/2017
<b>Description:</b> Test the server under high load	
<b>Pre-Condition:</b>	
<ul style="list-style-type: none"><li>- The web server has been started</li><li>- Have JMeter</li></ul>	
<b>Post-Condition:</b>	

Step	Action	Expected Outcome	Actual Outcome	Pass/Fail
1	Test the server with 1000 threads under 1s	The test pass without errors		

Test case 4.2 Stress test

<b>Test ID:</b> 4.2		<b>Tester:</b> Test-leader		
<b>Test Name:</b> Stress test		<b>Requirement Covered:</b> 6		
<b>Test Type:</b> Stress testing		<b>Execution Date:</b> 21/12/2017		
<b>Description:</b> Test the server under high load				
<b>Pre-Condition:</b>				
<ul style="list-style-type: none"><li>- The web server has been started</li><li>- Have JMeter</li></ul>				
<b>Post-Condition:</b>				
<b>Step</b>	<b>Action</b>	<b>Expected Outcome</b>	<b>Actual Outcome</b>	<b>Pass/Fail</b>
1	Test the server with 1000 threads under 2s	The test pass without errors		