Test Report

Result for manual test cases

Here follows a table with results of the Test cases along with a more detailed explanation of the failing tests.

Test Case	Use Case / Requirement	Result
1.1	UC1 / Req.3	OK
1.2	UC1 / Req.3	OK
1.3	UC1(4a)	NOT OK
1.4	UC1(4b)	NOT OK
1.5	UC1(4c)	NOT OK
2.1	UC2	OK
3.1	UC3 / Req.2	NOT OK
3.2	UC3(2a) / Req.2	OK
3.3	UC3(2b) / Req.2	NOT OK
3.4	UC3(2c) / Req.2	OK
4.1	N/A / Req.1	NOT OK
4.2	N/A / Req.1	OK
5.1	N/A / Req.5	NOT OK
6.1	N/A / Req.4	OK
Total: 14 st	coverage: ca 80% *	7/14

**Covers about 80% of the given requirements and use cases(linux, mac, internal server error are not tested)*

Test case 1.3

expected:

- The system should present the message:

  “Socket 9000 was taken”
- Server stops

actual:

- The system presents only the message:

  “Port is taken”

Test case 1.4

expected:

- The system should present the error message:

  “No access to folder /resources”

actual:

- There is no such error message, the server starts without issues

Test case 1.5

expected:

- The system should present the error message. “Cannot write to server log file log.txt”

actual:

- There is no such message, also the log.txt does not exist.

Test case 3.1

expected:

- The shared resource is shown in the browser and a success message is written to the access log
- In network section in your browser get the status code 200 OK

actual:

- There is no message written to the access log, due to that there is no existing log.

Test case 3.3

expected:

- The system presents that the resource is forbidden
- In your browser you will receive the message: “403 Forbidden”

actual:

- Instead the message given is “404 Not found”

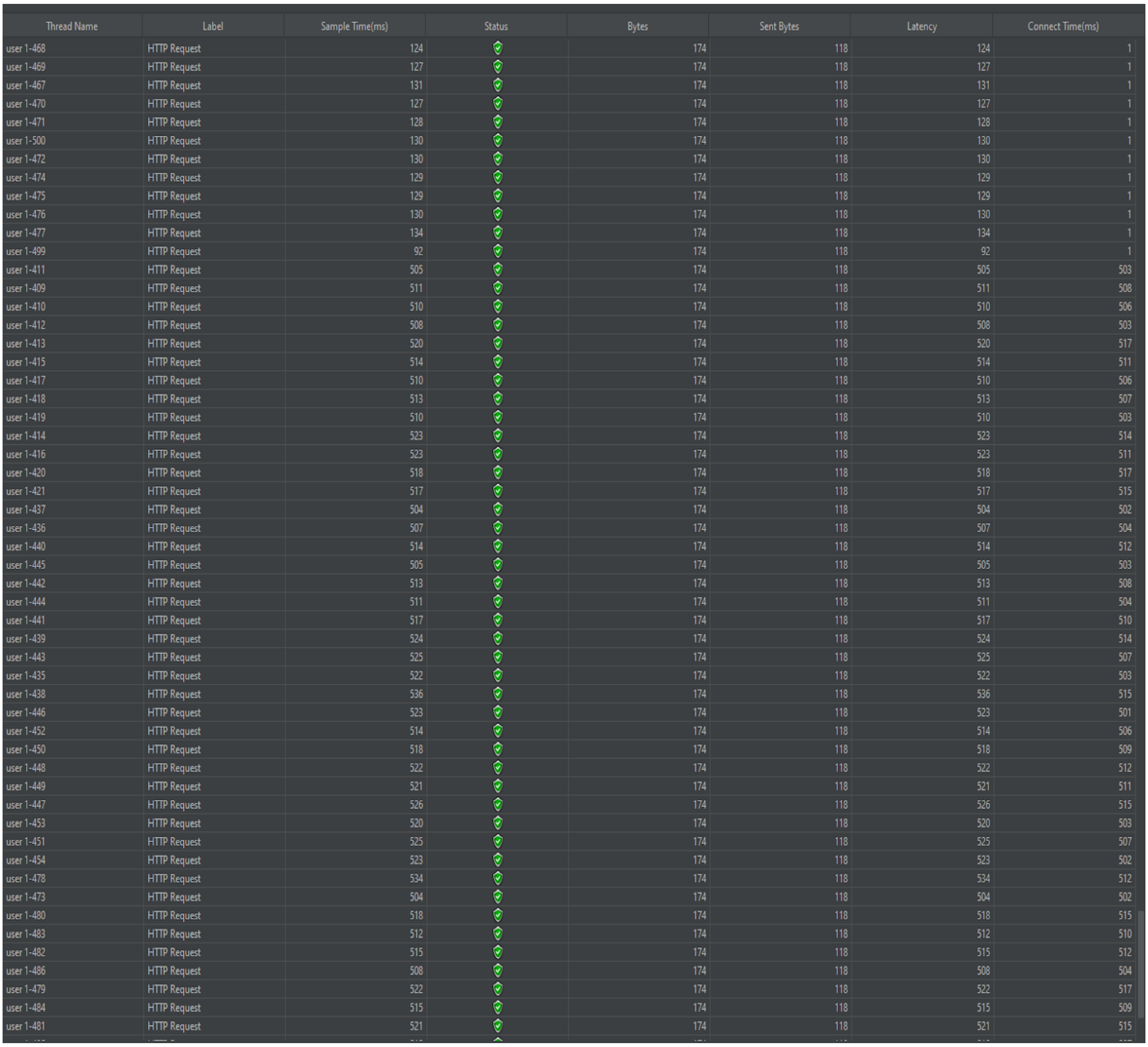
Test case 4.1

expected:

- The test should go through with no crashes or latency over 200 m/s

actual:

- There is latency over 200



Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
user 1-468	HTTP Request	124	✓	174	118	124	1
user 1-469	HTTP Request	127	✓	174	118	127	1
user 1-467	HTTP Request	131	✓	174	118	131	1
user 1-470	HTTP Request	127	✓	174	118	127	1
user 1-471	HTTP Request	128	✓	174	118	128	1
user 1-500	HTTP Request	130	✓	174	118	130	1
user 1-472	HTTP Request	130	✓	174	118	130	1
user 1-474	HTTP Request	129	✓	174	118	129	1
user 1-475	HTTP Request	129	✓	174	118	129	1
user 1-476	HTTP Request	130	✓	174	118	130	1
user 1-477	HTTP Request	134	✓	174	118	134	1
user 1-499	HTTP Request	92	✓	174	118	92	1
user 1-411	HTTP Request	505	✓	174	118	505	503
user 1-409	HTTP Request	511	✓	174	118	511	508
user 1-410	HTTP Request	510	✓	174	118	510	506
user 1-412	HTTP Request	508	✓	174	118	508	503
user 1-413	HTTP Request	520	✓	174	118	520	517
user 1-415	HTTP Request	514	✓	174	118	514	511
user 1-417	HTTP Request	510	✓	174	118	510	506
user 1-418	HTTP Request	513	✓	174	118	513	507
user 1-419	HTTP Request	510	✓	174	118	510	503
user 1-414	HTTP Request	523	✓	174	118	523	514
user 1-416	HTTP Request	523	✓	174	118	523	511
user 1-420	HTTP Request	518	✓	174	118	518	517
user 1-421	HTTP Request	517	✓	174	118	517	515
user 1-437	HTTP Request	504	✓	174	118	504	502
user 1-436	HTTP Request	507	✓	174	118	507	504
user 1-440	HTTP Request	514	✓	174	118	514	512
user 1-445	HTTP Request	505	✓	174	118	505	503
user 1-442	HTTP Request	513	✓	174	118	513	508
user 1-444	HTTP Request	511	✓	174	118	511	504
user 1-441	HTTP Request	517	✓	174	118	517	510
user 1-439	HTTP Request	524	✓	174	118	524	514
user 1-443	HTTP Request	525	✓	174	118	525	507
user 1-435	HTTP Request	522	✓	174	118	522	503
user 1-438	HTTP Request	536	✓	174	118	536	515
user 1-446	HTTP Request	523	✓	174	118	523	501
user 1-452	HTTP Request	514	✓	174	118	514	506
user 1-450	HTTP Request	518	✓	174	118	518	509
user 1-448	HTTP Request	522	✓	174	118	522	512
user 1-449	HTTP Request	521	✓	174	118	521	511
user 1-447	HTTP Request	526	✓	174	118	526	515
user 1-453	HTTP Request	520	✓	174	118	520	503
user 1-451	HTTP Request	525	✓	174	118	525	507
user 1-454	HTTP Request	523	✓	174	118	523	502
user 1-478	HTTP Request	534	✓	174	118	534	512
user 1-473	HTTP Request	504	✓	174	118	504	502
user 1-480	HTTP Request	518	✓	174	118	518	515
user 1-483	HTTP Request	512	✓	174	118	512	510
user 1-482	HTTP Request	515	✓	174	118	515	512
user 1-486	HTTP Request	508	✓	174	118	508	504
user 1-479	HTTP Request	522	✓	174	118	522	517
user 1-484	HTTP Request	515	✓	174	118	515	509
user 1-481	HTTP Request	521	✓	174	118	521	515

The heavy load test is dependent on the computer's hardware and can give different results. The testers have used two different computers with different specs to determine the result.

Test case 5.1

expected:

- The access log should have some content

actual:

- There is no access log

Result for automated tests

Tests	Coverage*(methods being tested)*	Results *(passing/all)*
AcceptThreadTest	100%	3/3
ClientFactory	0%*(no test sut)*	0/0
ClientSocketTest	75%	3/3
ClientThreadTest	100%	3/3
HeaderTest	100%	4/4
HTTPReaderTest	100%	4/4
HTTPRequestTest	0%	1/1
HTTPRequestParserTest	100%	5/5
HTTPServerTest	100%	4/4
HTTPServerConsoleTest	100%	3/3
PortTest	100%	3/3
ResponseFactoryTest	100%	5/5
ServerFactoryTest	100%	1/1
SharedFolderTest	100%	3/3
HTTPGetProtocoll	0%*(no test sut)*	0/0
SocketClientTest	100%	0/1
ContentTypeTest	100%	1/1
ErrorResponses	100%	1/1
HTMLFileResponseTest	100%	1/1
StressTest	100%	1/1
ConsoleViewTest	100%	9/10
	total: 84,5 %	total: 55/57

Traceability-Matrix

Requirement Identifiers	reqs tested	Req 1	Req 2	Req 3	Req 4	Req 5	UC 1	UC 2	UC 3
Test cases	14	2	4	2	1	1	5	1	4
1.1	2			X			X		
1.2	2			X			X		
1.3	1						X		
1.4	1						X		
1.5	1						X		
2.1	1							X	
3.1	2		X						X
3.2	2		X						X
3.3	2		X						X
3.4	2		X						X
4.1	1	X							
4.2	1	X							
5.1	1					X			
6.1	1				X				

Conclusion

The Tests have under 90% coverage in total, and only 50% of the manual test cases are passing, therefore the SDC management are not pleased with the result and the testers evaluate the web server to be unusable in its current state.

Timelog

Task	time spent
Writing Strategy	3h
Writing Test Plan	4h
Exploratory testing	4h
Writing Test Cases	20h
Executing Tests	5h
Writing Test Report	8h
Total:	44h