Test Cases

TC.1 Server Start-up

TC1.1: Server startup fails when providing only the socket port number

Steps:
  1- Edit the run configurations in the main class
  2- Add only the correct port number "9000" in the project arguments.
  3- Run the project.

Expected result:
  - The system throws two exceptions and does not start the server

Actual result:
  - The system throws both exceptions and does not start the server as expected.

Output :

```
/Library/Java/JavaVirtualMachines/jdk-13.0.1.jdk/Contents/Home/bin/java "-javaagent:/Applications/IntelliJ IDEA.app/Contents/lib/idea_rt.jar=49833:/Applicati
Exception in thread "main" se.lnu.http.exceptions.NotADirectoryException: java.lang.ArrayIndexOutOfBoundsException: Index 1 out of bounds for length 1
    at se.lnu.http.view.ConsoleView.getDirectory(ConsoleView.java:77)
    at se.lnu.http.HTTPServerConsole.runConsole(HTTPServerConsole.java:36)
    at se.lnu.http.HTTPServerConsole.main(HTTPServerConsole.java:27)
Caused by: java.lang.ArrayIndexOutOfBoundsException: Index 1 out of bounds for length 1
    at se.lnu.http.view.ConsoleView.getDirectory(ConsoleView.java:66)
    ... 2 more
```

TC1.3: Server startup fails when providing only the resource container

Steps:
  1- Edit the run configurations in the main class
  2- Add only the correct resource container "bin/se/lnu/http/resources/inner/" in the project arguments.
  3- Run the project.

Expected result:
  - The system shows an error saying the port number is needed and the server does not start.

Actual result:
  - The system shows an error saying the port number is needed and the server does not start as expected.

Output:

```
/Library/Java/JavaVirtualMachines/jdk-13.0.1.jdk/Contents/Home/bin/java "-javaagent:/Applications/IntelliJ
Enter a valid port 1-65535 and a optional URL
Process finished with exit code 0
```

TC1.3: Start Server fails when providing taken socket port number

Steps:
1- Edit the run configurations in the main class
2- Add a taken socket in the project arguments.
3- Run the project.

Expected Result
- The system shows that the socket provided is taken

Actual result:
- The system shows that the socket was taken but does not which one exactly.

TC1.4: Successful Server startup with the correct port number and resource container

Steps:
1- Edit the run configurations in the main class
2- Add the correct port number and the correct resource container "9000 bin/se/lnu/http/resources/inner/" in the project arguments.
3- Run the project.

Expected result:
- The system gives a message saying the server started

Actual result:
- The system gives a message saying the server started

Output:

```
/Library/Java/JavaVirtualMachines/jdk-13.0.1.jdk/Contents/Home/bin/java "-javaagent:/Applications/IntelliJ IDEA.app
HTTP Server object constructed
HTTP Server started
Accept
Accept
ClientThread started nr: 1
Accept
ClientThread started nr: 2
ClientThread 1 served file : index.html
ClientThread 1 served file : works.png
ClientThread 2 served file : works2.png
ClientThread stopped nr: 1
ClientThread stopped nr: 2
```

TC2: Stop Server successfully

Precondition:
- A web server has been started

Steps:
- Write "stop" in the console.

Expected result:
- The server stops and the system shows that the server was stopped

Actual result:
- The server stops and the system shows that the server was stopped as expected

Output:

```
/Library/Java/JavaVirtualMachines/jdk-13.0.1.jdk/Contents/Home/bin/java "-javaag
HTTP Server object constructed
HTTP Server started
Accept
stop
HTTP Server Accept thread stopped
HTTP Server stopped
```

TC3: Server startup fails when shared resource is missing

Precondition:
- A web server has been started

Steps:
1- Copy the path to the resources folder in the project
2- remove the resources folder from the project
3- Edit the run configuration on the main class
4- Add the port number and a path that was copied in the first step to the project arguments.
5- Run the server.

Expected result:
- The system should show an error saying the resource cannot be accessed

Actual result:
- The system crashes and does not show an appropriate error.

Requirement 1: The web server should be responsive under high load.

TC4: The web server is responsive under high load.

Precondition:
- The server should be running on port 9000 and JMeter should be installed

Steps:
1- Locate the test file and open it using in JMeter
2- Run the tests and see the summary report.

Expected result:
- There should be no errors present

Actual result:
- No error was shown as expected.

Requirement 2: The web server must follow minimum requirements for HTTP 1.1

TC5: The web server must follow minimum requirements for HTTP 1.1

Precondition:
- The server should be running on port 9000 and JMeter should be installed

Steps:
1- Locate the test file and open it using in JMeter
2- Run the tests and see the assertion result.

Expected result:
- The http request is shown n times with a pass

Actual result:
- The http request is shown n times with a pass, as expected

Requirement 3: The web server must work on Linux, Mac, Windows*

TC6: The web server must work on Linux, Mac, Windows*.

TC6.1: Server works on Mac

Precondition:
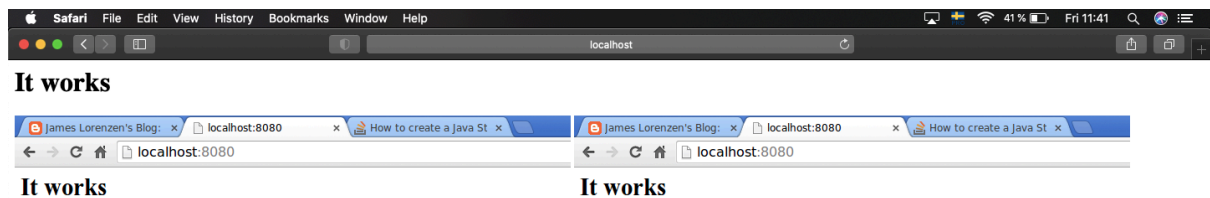- The jar file is on the server and Java is preinstalled.

Steps:
1- Run the server jar file
2- Access the URL using any browser

Expected result
- The server starts and a page with an image "it works" shows when accessing the URL.

Actual Result
- The server starts and the page expected loads successfully.

TC6.2: Server works on Windows 10

Precondition:
- The VM is setup properly with java installed and the jar present on the server

Steps:
1- Start the VM
2- Run the server jar file
3- Access the URL using any browser

Expected result
- The server starts and a page with an image "it works" shows when accessing the URL.

Actual Result
- The server starts and the page expected loads successfully.

TC6.3: Server works on Linux

Precondition:
-   The VM is setup properly with java installed and the jar present on the server

Steps:
1- Start the VM
2- Run the server jar file
3- Access the URL using any browser

Expected result
-   The server starts and a page with an image "it works" shows when accessing the URL.

Actual Result
-   The server starts and the page expected loads successfully

Requirement 4: The source code should be released under GPL-2.0.

TC7: The source code is released under GPL-2.0

Steps:
1- Check the file LICENCE in the server
2- See if this licence is released under GPL-2.

Expected result:
-   The source code is released under GPL-2.

Actual result:
-   The source code is released under GPL-2 as expected.

Requirement 5: The access log should be viewable from a text editor.

TC8: The access log should be viewable from a text editor.

Precondition:
-   The server must be run at least once before

Steps:
-   Locate the server folder and open the .log file in a text editor

Expected result:
-   The access log is viewable from the text editor

Actual result:
-   The access log is viewable from the text editor as expected