**System Test Plan**

Author: Yannis Plaschko

(TINF20C, SWE I Praxisprojekt 2021/2022)

Project: Websockets in a LwIP HTTP Server

Customer: Rentschler & Holder

Rotebühlplatz 41

70178 Stuttgart

Supplier: Team 4: Laura Reeken, inf20051@lehre.dhbw-stuttgart.de

   Benjamin Esenwein, inf20074@lehre.dhbw-stuttgart.de

   Yannis Plaschko, inf20093@lehre.dhbw-stuttgart.de

   Maximilian Meier, inf20084@lehre.dhbw-stuttgart.de

   Lucas Kaczynski, inf20147@lehre.dhbw-stuttgart.de

   Isabel Schwalm, inf20085@lehre.dhbw-stuttgart.de

   Rotebühlplatz 41

   70178 Stuttgart

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comment** |
| 0.1 | 29.04.2022 | Yannis Plaschko | created |
| 0.2 | 04.05.2022 | Yannis Plaschko | Add further tests |
| 1.0 | 05.05.2022 | Yannis Plaschko | Document format fix |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Contents**

1. Scope 3

2. Abbreviations Fehler! Textmarke nicht definiert.

3. Product Names and Attributes 3

4. Features 4

5. Test Preparation Strategy 5

6. Test Execution Strategy 5

7. Test Equipment 5

8. Test Schedule and Budget 5

9. Test Planning 5

10. References / Standards 5

11. Appendix: Testcases 6

**11.1. Testsuite <TS-001 WebSocket with its Testclient> 6**

**11.1.1. <TC-001-001> (Choose the correct network adapter) 6**

**11.1.2. <TC-001-002> (Choose the wrong network adapter) 6**

**11.1.3. <TC-001-003> (Connect to the WebSocket) 7**

**11.1.4. <TC-001-004> (Message the Server)**

**11.1.5. <TC-001-005> (Disconnect from WebSocket) 8**

**11.2. Testsuite <TS-002 API> 9**

**11.2.1. <TC-002-001> (Call API Root) 9**

**11.2.2. <TC-002-002> (Call API /something) 9**

**11.2.3. <TC-002-003> (Call API /identification directly) 10**

**11.2.4. <TC-002-004> (Call API /identification from root) 10**

# Scope

The STP (System Test Plan) specifies the test strategy and test planning. It references tests to be performed to verify the accordance of the demanded features given by the SRS (System Requirements Specification) to the implemented features. The document derived from the STP is the STR (System Test Report) where additionally the results are given.

# Abbreviations

* TC– Testcase
* TS – Testsuite
* GUI – Graphical User Interface
* API – Application Programming Interface

# Product Names and Attributes

The following test objects must be verified:

|  |  |  |  |
| --- | --- | --- | --- |
| **Ref.-Id.** | **Product Number** | **Product Name** | **Product Description** |
| 1 | Build v1.0 | Websockets in a LwIP HTTP Server | An implementation of Websockets and an API in LwIP |

# Features

The following requirements must be verified, as long as they are not classified as “not to be tested”. This table shows the test coverage between functionality and test suites or test cases.

|  |  |  |  |
| --- | --- | --- | --- |
| **Req. - ID** | **Functionality** | **Priority** | **Testsuite ID** |
| LF10: Choose the correct network adapter | Checks if lwIP starts correctly after choosing the correct network adapter | A | TS-001 |
| LF20: Choose the wrong network adapter | Checks if lwIP fails noisy after choosing a wrong network adapter | B | TS-001 |
| LF30: Connect to the Websocket | Checks if the Testclient is able to establish a connection to the LwIP server | A | TS-001 |
| LF40: Message the Server | Checks whether the LwIP server echoes incoming messages | A | TS-001 |
| LF50: Disconnect from Websocket | Checks if the Testclient is able to destroy a connection to the LwIP server | A | TS-001 |
| LF60: Call API root | Checks if calling the Root Endpoint returns a sample website | A | TS-002 |
| LF70: Call API /something | Checks if calling an undefined Endpoint, results in an error Message | B | TS-002 |
| LF80: Call API /identification directly | Sends Information about the Server | A | TS-002 |
| LF90: Call API /identification from root | Sends Information about the Server | B | TS-002 |

# Test Preparation Strategy

The creation of tests will be based on the different use-cases. Two use-cases can be identified, the WebSocket with its Testclient and the API.

The WebSocket with its Testclient represents the first main application case. The Testclient should be able to visually demonstrate the WebSocket functionality.

The API is the second main application case. It should be able to deliver Information about the device on which it is running. These responses must be checked.

# Test Execution Strategy

Since it is an extension of an already existing software, a complete test is not necessary.

1. WebSocket with its Testclient
2. Application Programming Interface

# Test Equipment

The following equipment must be available for testing:

* A computer with Windows 10
* A functioning and correctly configured Installation of LwIP (For tests after TC-001-001 / 002)

# Test Schedule and Budget

The testing of the application begins as soon as the application is completed. This makes it possible to make the necessary corrections quickly. The WebSocket Support can be tested independently from the API which makes an asynchronous testing possible.

No budget is needed for the tests, as they are all performed by hand.

# Test Planning

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Testsuite** | **Test objective** | **Testplan Creator** | **Testplan Reviewer** | **Tester** |
| TS-001 | WebSocket with its Testclient | Yannis Plaschko | Benjamin Esenwein | Yannis Plaschko |
| TS-002 | Application Programming Interface | Yannis Plaschko | Benjamin Esenwein | Yannis Plaschko |

# References / Standards

[[1] SRS TINF20C Websockets with LwIP](https://github.com/TINF20C/Team_4_Websockets-lwIP/wiki/SRS:-System-Requirements-Specification)

# Appendix: Testcases

## Testsuite <TS-001 WebSocket with its Testclient>

### <TC-001-001> (Choose the correct network adapter)

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase ID:** | | TC-001-001 | |
| **Testcase Name:** | | Choose the correct network adapter | |
| **Req.-ID:** | | LF10 | |
| **Description:** | | This test case verifies that lwIP starts correctly after choosing the correct network adapter. | |
| **Test Steps** | | | |
| **Step** | **Action** | | **Expected result** |
| 1 | Run lwIP\_Test.exe | | Application starts without problems. |
| 2 | Insert the correct number of your network adapter | | LwIP accepts the Input and runs correctly |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Testdata:** | | TD-001-001 | |
| **Dataset** | **Input File** | | **Output File** |
| 1 | lwIP\_Test.exe | | X |
|  | | | |

### <TC-001-002> (Choose the wrong network adapter)

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase ID:** | | TC-001-002 | |
| **Testcase Name:** | | Choose the wrong network adapter | |
| **Req.-ID:** | | LF20 | |
| **Description:** | | This test case verifies that lwIP fails noisy after choosing a wrong network adapter. | |
| **Test Steps** | | | |
| **Step** | **Action** | | **Expected result** |
| 1 | Run lwIP\_Test.exe | | Application starts without problems. |
| 2 | Insert a wrong number which does not belong to a network adapter | | A Pop-Up with an error message should appear and the lwIP window will print an error |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Testdata:** | | TD-001-002 | |
| **Dataset** | **Input File** | | **Output File** |
| 1 | lwIP\_Test.exe | | X |
|  | | | |

### <TC-001-003> (Connect to the WebSocket)

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase ID:** | | TC-001-003 | |
| **Testcase Name:** | | Connect to the WebSocket | |
| **Req.-ID:** | | LF30 | |
| **Description:** | | This test case verifies that the Testclient is able to connect to the WebSocket. | |
| **Test Steps** | | | |
| **Step** | **Action** | | **Expected result** |
| 1 | Open the Testclient | | Application starts without problems. |
| 2 | Select “Connect” Menu point | | A Pop-Up with an input window should appear |
| 3 | Input the IP of the WebSocket and click on “connect” | | The GUI should show “Connection Established” and display a message field |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Testdata:** | | TD-001-003 | |
| **Dataset** | **Input File** | | **Output File** |
| 1 | X | | X |
|  | | | |

### <TC-001-004> (Message the Server)

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase ID:** | | TC-001-004 | |
| **Testcase Name:** | | Message the Server | |
| **Req.-ID:** | | LF30, LF40 | |
| **Description:** | | This test case verifies that a message sent to the server will be echoed back by it. | |
| **Test Steps** | | | |
| **Step** | **Action** | | **Expected result** |
| 1 | Connect the Testclient to a running WebSocket | | Application starts without problems. |
| 2 | Type the Message from Dataset 1 into the text field and press “Send” | | The WebSocket echoes the Message back to the Testclient (The Message should be visible *twice*) |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Testdata:** | | TD-001-004 | |
| **Dataset** | **Input File** | | **Output File** |
| 1 | TINF20C\_testMessage\_Team4 | | X |
|  | | | |

### <TC-001-005> (Disconnect from WebSocket)

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase ID:** | | TC-001-005 | |
| **Testcase Name:** | | Disconnect from WebSocket | |
| **Req.-ID:** | | LF30, LF40, LF50 | |
| **Description:** | | This test case verifies that the Testclient can destroy an existing connection. | |
| **Test Steps** | | | |
| **Step** | **Action** | | **Expected result** |
| 1 | Connect the Testclient to a running WebSocket | | Application starts without problems. |
| 2 | Click on the “Disconnect” button | | The Testclient disconnects from the WebSocket and shows the message “Successfully disconnected” |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Testdata:** | | TD-001-005 | |
| **Dataset** | **Input File** | | **Output File** |
| 1 | X | | X |
|  | | | |

## Testsuite <TS-002 API>

### <TC-002-001> (Call API Root)

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase ID:** | | TC-002-001 | |
| **Testcase Name:** | | Call API Root | |
| **Req.-ID:** | | LF60 | |
| **Description:** | | This test case verifies that calling the Root Endpoint returns a sample website. | |
| **Test Steps** | | | |
| **Step** | **Action** | | **Expected result** |
| 1 | Run the LwIP HttpServer | | Application starts without problems. |
| 2 | Open the Browser and type the IP of the Server in the Search bar and hit enter | | A Site should become visible with the Headline “LwIP Test Application”, a Text block and a link to “/identification” |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Testdata:** | | TD-002-001 | |
| **Dataset** | **Input File** | | **Output File** |
| 1 | X | | X |
|  | | | |

### <TC-002-002> (Call API /something)

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase ID:** | | TC-002-002 | |
| **Testcase Name:** | | Call API /something | |
| **Req.-ID:** | | LF70 | |
| **Description:** | | This test case verifies that calling a not existing endpoint results in the display of an error message. | |
| **Test Steps** | | | |
| **Step** | **Action** | | **Expected result** |
| 1 | Run the LwIP HttpServer | | Application starts without problems. |
| 2 | Open the Browser and type the IP of the Server in the Search bar plus “/something” and hit enter | | The Browser should display an error Message depending on the used Browser. |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Testdata:** | | TD-002-002 | |
| **Dataset** | **Input File** | | **Output File** |
| 1 | X | | X |
|  | | | |

### <TC-002-003> (Call API /identification directly)

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase ID:** | | TC-002-003 | |
| **Testcase Name:** | | Call API /identification directly | |
| **Req.-ID:** | | LF80 | |
| **Description:** | | This test case verifies whether a direct call displays information about the Server. | |
| **Test Steps** | | | |
| **Step** | **Action** | | **Expected result** |
| 1 | Run the LwIP HttpServer | | Application starts without problems. |
| 2 | Open the Browser and type the IP of the Server in the Search bar plus “/identification” and hit enter | | The Browser should display the information about the server in JSON. |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Testdata:** | | TD-002-003 | |
| **Dataset** | **Input File** | | **Output File** |
| 1 | X | | X |
|  | | | |

### <TC-002-004> (Call API /identification from root)

|  |  |  |  |
| --- | --- | --- | --- |
| **Testcase ID:** | | TC-002-004 | |
| **Testcase Name:** | | Call API /identification from root | |
| **Req.-ID:** | | LF60, LF90 | |
| **Description:** | | This test case verifies whether a click on the link redirects to displays information about the Server. | |
| **Test Steps** | | | |
| **Step** | **Action** | | **Expected result** |
| 1 | Run the LwIP HttpServer | | Application starts without problems. |
| 2 | Open the Browser and type the IP of the Server in the Search and hit enter | | A Site should become visible with the Headline “LwIP Test Application”, a Text block and a link to “/identification” |
| 3 | Click on the Link | | The Browser should display the information about the server in JSON. |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Testdata:** | | TD-002-004 | |
| **Dataset** | **Input File** | | **Output File** |
| 1 | X | | X |
|  | | | |