Test Plan

My Web Server

1. Purpose

The purpose with this document is to describe which requirements that are being tested and not, how important they are and how the project will test the requirements. The document will also provide potential risks in this test project that need a mitigations strategy and what deliverables the project has.

2. Requirements / Use cases to be tested in this iteration

Requirement		
Start server		
Requirement / Use Case ID	Prioritization	Type of requirement
UC1	High	Functional requirement

Requirement		
Stop server		
Requirement / Use Case ID	Prioritization	Type of requirement
UC2	High	Functional requirement

Requirement		
Request shared resource		
Requirement / Use Case ID	Prioritization	Type of requirement
UC3	High	Functional requirement

Requirement		
Simple to deploy on IOT-devices		
Requirement ID	Prioritization	Type of requirement
REQ4	Medium	Non-functional requirement

Requirement		
Compatibility with IOT-devices		
Requirement ID	Prioritization	Type of requirement
REQ5	High	Non-functional requirement

Requirement		
Web server should be able to handle 25 request per seconds		
Requirement ID	Prioritization	Type of requirement
REQ6	Medium	Non-functional requirement

Requirement		
Must follow minimum requirements for HTTP 1.1		
Requirement ID	Prioritization	Type of requirement
REQ7	Medium	Non-functional requirement

Requirement		
Source code released under GPL 2.0		
Requirement ID	Prioritization	Type of requirement
REQ8	Low	Non-functional requirement

Requirement		
Access log viewable from text-editor		
Requirement ID	Prioritization	Type of requirement
REQ9	Low	Non-functional requirement

Requirement		
Web server should be able to handle massive request load with less than 20% error responses and without crashing		
Requirement ID	Prioritization	Type of requirement
REQ10	Low	Non-functional requirement

3. Testing techniques

Type of testing
Exploratory testing
Test level
System-test
Description
Exploratory testing will be performed to get a better understanding of the system and what the functionality it has. Different inputs will be tested to see how the system responds. Also randomly clicking on mouse and keyboard will be performed to see if some sort of reaction is received.
Tester
Test lead

Type of testing Manual testing
Test level System-test
Requirements UC1 Start server UC2 Stop server UC3 Request shared resource
Description To test the functional requirements manual testing of the system will be performed to see if any defects from the scenarios in the use cases can be found.
Tester Test lead

Type of testing Automated testing
Test level Unit-test
Requirements UC1 Start server UC2 Stop server UC3 Request shared resource
Description The already existing automated tests in will be regression tested. In this iteration the quality of the existing automated test will not be examined, and the tests will be viewed as correctly done. Unless there are a failure the automated tests will not be read, and therefore they will not be traceable to a specific requirement. A specific test will however act as a starter for further investigations if a failure occurs in it, so the tests still fill a role in this test iteration.
Tester Test lead

Type of testing

Conformance testing

Test level

System-test

Requirements

REQ7 Must follow minimum requirements for HTTP 1.1
REQ8 Source code released under GPL 2.0
REQ9 Access log viewable from text-editor

Description

Conformance tests will be done to evaluate some of the non-functional requirements such as server follow minimum requirements for HTTP 1.1 and source code released under GPL-2.0.

Tester

Test lead

Type of testing

Compatibility testing

Test level

System-test

Requirements

REQ5 Compatibility with IOT-devices

Description

Conformance tests will be done to evaluate if the system is compatible on IOT-devices.

Tester

Test lead

Type of testing

Performance testing

Test level

System-test

Requirements

REQ6 Web server should be able to handle 50 request per seconds
REQ10 Web server should be able to handle massive request load with less than 20% error responses and without crashing

Description

Performance test to the non-functional requirements of web server should be able to handle 50 request per seconds and web server should be able to handle massive request load with less than 20% error responses and without crashing. This include both load tests and stress tests.

Tester

Test lead

4. What is not tested in this iteration

- Other versions of Windows than Windows 10 (that is the main desktop OS for the test lead), and all MacOS will not be tested since it seems very unlikely to be used in an IOT-device. If this is a needed requirement this can be tested in later iterations.
- Easy access for end users will not be tested in this iteration since there is no time and this is not prioritized at this stage to test.
- No software security testing will be done in this iteration except testing that no file is accessible outside the shared container. This is due to the limit of time and resources given.

5. Risk and mitigation strategy

Risk	Probability	Effects
Understanding IOT domain	High	Moderate
Mitigation strategy		
Since the test lead have no prior knowledge of developing software's for IOT devices there is a risk that this project will miss to test for things that is important for IOT-developers. To address this the test lead must read up on this domain before starting to perform test. For future iterations the test lead suggests to performing interviews with IOT-developers and more in-depth analysis to fully grasp the domain.		

Risk	Probability	Effects
Access to IOT-devices	Moderate	Serious
Mitigation strategy		
The test project lacks resources in different types of IOT-devices that the server can be tested on and only have an old Raspberry Pi as a resource. This can cause potential trouble to test the server on real types of devices and see the deploy ability and the performance "My Web Server" really have. To alert this alert the SDC Management of risk and suggest investments in a couple of new devices that can be used for testing a later iteration.		

Risk	Probability	Effects
Hardware crash	Low	Serious
Mitigation strategy		
If for example the Raspberry Pi 3 Model B that the test lead has as a resource crashes when performing stress tests etc or it for other reasons does not work this can be a serious problem for this test if there is no type of device that can be seen as an potential IOT-device as recourse for the test lead. To handle this, alert the SDC management of risk and ask for a ready investment approvement for a new Raspberry Pi 4.		

Risk	Probability	Effects
Test lead lack of knowledge in developing in Java will slow down the testing effort.	High	Serious
Mitigation strategy		
The test lead has little to none experience in developing in Java. Test lead will have to view tutorials on how to work with Java SDK, how to compile projects, how to create jar files and how to run them, before starting the testing process.		

Risk	Probability	Effects
Illness of developer	Moderate	Serious
Mitigation strategy		
Since this project only has one human resource it is very fragile. If the developer gets a serious illness and can’t work, the project deadlines can be missed, and the project will be postponed. To handle this the developer must strive to be ready many days in advance to have a safer approach. Also, alert SDC management of risk.		

6. Deliverables

The artefacts below should be handed to the SDC management after one week of testing, latest 2020-12-11.

- Test strategy document
- Test plan document
- Test cases document
- Test report document
- Load Test.jmx for Jmeter load test
- Stress Test.jmx for Jmeter stress test

7. After this iteration

The test report will show the outcome of the tests performed in this iteration and have a traceability to the requirements in this document. It will also contain a summary of the state of “My Web Server” and will have the test leads recommendation if SDC should proceed further with “My Web Server” or not. It is then up to the SDC management to make the decision to give it at go for a new test and development iteration or a no go.

To get a recommendation more than 90% manual test cases must pass and more than 90% of the automated test must pass.