Test Plan
My Web Server

Version 1.0

Dated: 22rd December 2017

1. Introduction

1.1 Purpose

This test plan for “My Web Server” supports the following objectives:

- Evaluate the current state of the software
- List the requirements that will be tested
- Define testing standards and procedures required to conduct the system testing
- Identify the resources and an estimated time for testing efforts
- Prepare and document the test cases
- Provide testing summary report and detect if any faults or errors exist

1.2 Background

My Web Server is an open source software. The Software Development Company (SDC) thinks of use this software on a wide range of Internet of Things (IoT) to show information from sensors. SDC wants to see if this software meets the requirements.

1.3 Scope

This test plan applies integration testing, unit testing, manual test cases, and stress testing. All these tests will be performed in this iteration.

1.4 Scenario

The small Software Development Company (SDC) has found a possible market niche for giving out a simple to deploy web-server. SDC aims to redistribute this server on a wide range of Internet of Things (IOT) to present information from sensors etc. SDC wants an easy to deploy java-web-server that can be deployed on many different devices and therefore that can attract attention of a wide range of IOT developers. IOT-developers want minimal configuration as well as easy integration and adaptation of the web-server. End-customers want easy access and absolute security. The SDC management has found an open source abandonware software called “My web server”. It is your job as SDC employee(s) to evaluate the current state of “My web server”. SDC needs to know if the abandoned software fulfills the requirements as stated in the requirement document. Your budget is one man-week times the number of students. SDC management want a strategy, plan, test-cases as well as a test-report by the 23 December.

2 Use-cases and supplementary specification

This section represents the supplementary specification and the use cases provided by SDC:s requirement document

2.1 Use cases

Actors
Administrator: Installs, Starts, stops the Web server, inspects usage.
Browser: Accesses shared resources.

2.1.1 UC1 Start Server

Primary Actor
Administrator

Postcondition:

- A web server has been started
- A note in the access log was written, that the server was started

Main scenario

1. Starts when an administrator wants to start the server.
2. System asks for socket port number and shared resource container
3. The administrator provides a socket port number and a shared resource container
4. System starts a web server on the given port and presents that the server was started and writes a note in the access log.

Alternate Scenarios

- 4a. The web server could not be started due to socket was taken
    1. System presents an error message: “Socket XX was taken” (XX is the socket number, Example “80”)
    2. Exit Use Case
- 4b. The web server could not be started due restriction on the shared resource container
    1. System presents an error message: “No access to folder XX” (XX is the shared resource container provided, Example “\var\www”)
    2. Exit Use Case
- 4c. The access log could not be written to
    1. System presents an error message. “Cannot write to server log file log.txt”
    2. Exit Use Case

2.1.2 UC2 Stop Server

Primary Actor
Administrator

Precondition:

- A web server has been started

Postcondition:

- A note in the access log was written, that the server was stopped

Main scenario

1. Starts when a user wants to stop the server.
2. System stops the web server and presents that the webserver has been stopped

2.1.3 UC3 Request shared resource

Primary Actor

Browser

Precondition:

- A web server has been started

Postcondition:

- A note in the access log was written, that access happened with request information and the result of the request.

Technical note

- Browser and System communicates using HTTP 1.1.
- Error messages are part of HTTP 1.1 protocol
    - 200 OK
    - 400 Bad request
    - 403 Forbidden
    - 404 Not Found

Main scenario

1. Starts when a Browser wants to access a shared resource
2. System delivers the shared resource to the browser and a success message is written to the access log.

Alternate Scenarios

- 2a: The shared resource cannot be found
    1. System presents that the resource cannot be found
    2. Exit Use Case
- 2b: The shared resource is outside the shared resource container
    1. System presents that the resource is forbidden
    2. Exit Use Case
- 2c: The resource request is invalid or malformed
    1. System presents that the request cannot be handled
    2. Exit Use Case
- 2d: The server encounters an error when trying to process the request
    1. System presents that it has an internal error
    2. Exit Use Case

2.2 Supplementary Specification

- Req 1. The web server should be responsive under high load
- Req 2. The web server must follow minimum requirements for HTTP 1.1
- Req 3. The web server must work on Linux, Mac, Windows
- Req 4. The source code should be released under GPL-2.0
- Req 5. The access log should be viewable from a text editor

3 Requirements to Be Tested

- Functional Testing
  - Req 1. UC1: Check a correct start of the server
  - Req 2. UC2: Check the termination of the server
  - Req 3. UC3: A normal access to the shared resources by the browser
  - Req 4. Supplementary specification, "Req 2. The web server must follow minimum requirements for HTTP 1.1"
  - Req 5. Supplementary specification, "Req 5. The access log should be viewable from a text editor"

- Stress Testing
  - Req 6. Supplementary specification, "Req 1. The web server should be responsive under high load"
- Unit Testing
  - Req 7. Confirm proper functioning of unit tests from the existing JUnit-Test suites
- Integration Testing
  - Req 8. Automated JUnit test suites confirm the system components integrity
- License Testing
  - Req 9. Supplementary specification, "Req 4. The source code should be released under GPL-2.0

4 Requirements will not be tested

Req 1. Will be tested only on Windows 10 because I don't have the access to these different OS

5 Project Milestones

Milestone Task	Effort	Start Date	End Date
Test Plan	10h	11/12/2017	13/12/2017
Test Design	20h	13/12/2017	18/12/2017
Test Execution	15h	18/12/2017	22/12/2017
Test Evaluation	5h	22/12/2017	23/12/2017

6 Responsibilities

The roles responsible for this project at SDC is only me as test-leader. I will be responsible for the whole documentation, generating the test plan, the test cases, and the test strategy. I will also execute the tests and list the results and collect them in one document