Linnéuniversitetet
Kalmar Växjö

My Web Server Application

Test plan

Authers: Anonymous
Supervisor: Anonymous
Semester: Autumn 2019
Course code: 2DV610

Contents

1 Objective:

The objective of this document is to provide plans to test the abandonware “My Web Server” for the test team. Furthermore, this document contains the following

- Requirements should be tested.
- Level of testing and how tests will be implemented
- Roles and responsibility of testers

2 Scope

- The web server should be tested on three different operating systems.
  - Mac
  - Windows
  - Linux
- The web server will be tested on local server.
- The web server will be tested under high load.

3 Out of Scope

- The web server will not be checked on
  - XP, Vista, 7, 8, Server 2008

4 Quality Objectives

4.1 The web server must work on Linux, Mac, Windows.
4.2 The web server must follow minimum requirements for HTTP 1.1
4.3 The web server should be responsive under high load.

5 Testing Goals

5.1 SDC goals:

- The project should be easy to deploy.
- The project has to run or deploy on many different devices.

5.2 Developer:

- The project is set up with minimal configuration.
- The web server should be easy to integrate through a certain API and documentation,

5.3 Customer:

- Easy access and convenience
- The application comes with absolute security.

6 Test levels and Test Types

6.1 Unit testing

Aim	Verify the functionality of a specific section of code within the Junit test suite to ensure that the specific function is working as expected.
Implementation technique	Execute each test within the Junit test suite.
Result	All planned tests have been executed, and the errors found have been documented.
Evaluation	The Junit tests must be executed while the server is active to simulate real scenarios. All tests can be found in source code.
Testers	Testers involved: Tester 1, Tester 2, Tester 3
Testing tools	Junit, Mockito

6.2 2. Integration test

Aim	Verify the interfaces between components (modules) against a software design.
Implementation technique	Execute the individual software modules as a group to verify the aggregation of the system.
Result	All planned tests have been executed, and the errors found have been documented.
Evaluation	If this test produces large unexpected traces, cut the test in smaller pieces to improve fault localization.
Testers	Testers involved: Tester 3, Tester 4

6.3 System test

Aim	Tests the completely integrated system to ensure that the system meets its requirements.
Implementation technique	Execute all the test cases and check the traceability
Result	All planned tests have been executed, and the errors found have been documented.
Evaluation	The system test must be performed only after Unit and Integration tests.
Testers	Testers involved: Tester 1, Tester 4

6.4 Functional testing

Aim	Verify the specific functionality of the code
Implementation technique	Execute each use case using valid input to obtain the expected results and on the other hand, using invalid input to obtain error or warning messages.
Result	All planned tests have been executed, and the errors found have been documented.
Evaluation	None
Testers	Testers involved: Tester 1, Tester 2, Tester 3, Tester 4
Test elements	Refer section 7 of this document

6.5 Non-functional testing

Aim	Verify the non-functional attributes of the system
Implementation technique	Testing the performance, security and load on the system.
Result	All planned tests have been executed and the errors found have been documented.
Evaluation	None
Testers	Testers involved: Tester 1, Tester 3

6.6 Acceptance testing

Aim	Verify the system's compliance with the requirements and access whether it is acceptable for delivery
Alternative	A smoke test is used as a build acceptance test prior to further testing
Result	All planned tests have been executed and the errors found have been documented.
Evaluation	This testing is performed by the customers themselves, often in their lab environment or their own hardware
Testers	Testers involved: Tester 1, Tester 2

6.7 Security testing

Aim	Evaluating the system vulnerability
Implementation technique	Testing performed using the automation tool ‘Ratproxy’
Result	All planned tests have been executed and the errors found have been documented.
Evaluation	This is test should be performed after the application is further developed as it is still in its early stage and it passes most of the tests
Testers	Testers involved: Tester 3, Tester 4

6.8 Load testing

Aim	Evaluating the responsiveness and stability of the system under a certain workload
Implementation technique	Testing performed using the automation tool ‘JMeter’
Result	All planned tests have been executed and the errors found have been documented.
Evaluation	The user load varies from system to system
Testers	Testers involved: Tester2, Tester 3, Tester 4
Test elements	Refer to the Test Cases report

7 Features to be tested

According to the given requirements and use cases, we conclude that the features to be tested are as follows

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

8 Prioritization

Test Case	Name	Priority
TUC1.1	Start Server	High
TUC2.1	Stop Server	High
TUC1.4	Access log start server	High
THL.1.1	Response under High Load	High
TUC3.1	HTTP 1.1 Status 200	High
TUC3.2	HTTP 1.1 Status 400	High
TUC3.3	HTTP 1.1 Status 403	High
TUC3.4	HTTP 1.1 Status 404	High
TUC3.5	HTTP 1.1 Status 405	Medium
TOS.1	Server on different operating systems	High
TA.1	Acceptance testing	High
TP.1	Performance	Medium
TSS.1.2	Wrong Socket	Medium
TSS2.2	Access log- Server Stopped	Low
TGPL2.0	Test GPL 2.0	Low

9 Test Environment

Web Server: My Web Server
Operating System: Windows 10 Pro, Windows 10 Home, Mac Mojave 10.14
Java Version: 1.7 and 1.8
Browser:

- Google Chrome
  - Version 78.0.3904.108 (Official Build) (64-bit)
- Safari
  - Version 5.1.7 for Windows
  - Version 12.0.2 for macOS
- Microsoft Edge
  - Version 44.17763.831.0
  -

10 Responsibilities:

Scrum Master:
One of four member who is responsible for flow of information among testers as well as for retrieving required information from project manager and documentation for whole testing iteration.

Testing Team:

Whole team is responsible for executing required manual and automated testing.

11 Schedule:

The schedule for this project is divided in 20 hours, 3-cycles over two weeks. Documents regarding tests will be updated every cycle in digital/physical meeting. Aim is to accomplish of all test plans, following test strategy.

Iteration	Aim
1	1. Test TSS.1.1 and TSS.2.2 Responsible: P1, P2, P3 of testing team.
2	1. Test TOS.1 & TA.1 Responsible: P1, P3 2. Access Log persistency & Response High Lode Responsible: P1, P2
3	1. Test Status Message & Performance Responsible: P3, P4. 2. Test Medium priority and low priority test cases Responsible: Whole TEAM.