Test Report

1.Revision History

Date	Version	Author	Description
2020-12-08	1.0	Test manager	Planning Test Report.
2020-12-08	1.1	Test manager	Adding images of JUnit Tests in **Metrics**.
2020-12-09	1.2	Test manager	Adding documentation for **Types of testing performed, Test Environment & Tool, Lessons Learned, Recommendation, Exit Criteria, Conclusion/ Sign Off, Definitions, Acronyms, and Abbreviations, Time Log.**
2020-12-09	1.3	Test manager	Adding documentation and reviewing, **Types of testing performed, Lessons Learned, Recommendation,**

			Exit Criteria, Conclusion/ Sign Off, Definitions, Acronyms, and Abbreviations, Time Log.

2.Purpose of the document

Purpose of these documentations have been to test the software made by the Small Software Development Company (SDC). We have been testing the software SDC for some time now and we have got a lot of results from the different testing. The results will be covered in this document by making some reflections, summaries and displaying the results with images.

3.Application Overview

The Test Strategy is made for a simple to deploy web server that is created by the small Software Company (SDC). The goal with the product is to easily deploy a java-web-server that can be deployed onto a lot of different devices. This to attract the attention of a wide range of Internet Of Things (IOT) developers who want an easily deployed java-web-server for their projects. By testing the different requirements given by the stakeholders we can assure that the software is working correctly and therefore build confidence in our software.

4.Testing Scope

In Scope

Functional Testing for the following modules are in Scope of Testing.

- Start Server
- Stop Server
- Request shared resource

The requirements that have been tested are, **The web server must follow minimum requirements for HTTP 1.1**, **The access log should be viewable from a text editor**. This has been done by Manual Test Cases and JUnit tests.

Out of Scope

Performance Testing has been done for this easy to deploy web server.
The requirement was: The web server should be responsive under high load.
That has been tested by a JUnit test for stressTest which makes the application be under pressure from a lot of requests.

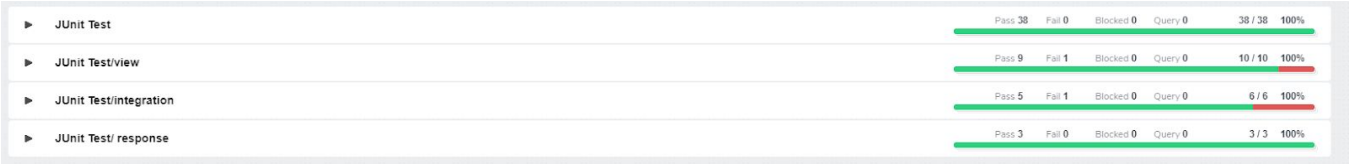
Items not tested

The Requirements that has partially been tested, **The web server must work on Linux, Mac, Windows*** and **The source code should be released under GPL-2.0.**
Due to some technical limitations. We had no access to the operating system Linux which made it hard for us to test that the web server works on that OS.

5.Metrics

Test Cases planned	Test cases executed	Test cases Pass	Test cases Failed
57	57	55	2

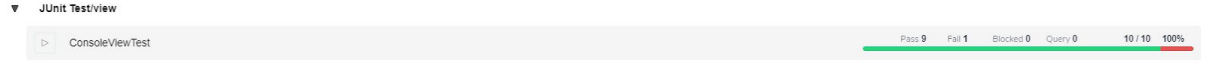
Overall JUnit Test Result



- Pass JUnit Tests **Coverage 96,49 %**
- Failed Junit Tests **Coverage 3,51 %**

JUnit Test Folder

JUnit Test / view Folder



JUnit Test / integration

JUnit Test / response Folder

6.Types of testing performed

- **Smoke Testning**

  This testing is done whenever a Build is received (deployed into the Test environment) for Testing to make sure the major functionality is working fine, Build can be accepted and Testing can start. Smoke Testing has been really important for us to make sure that the application has not been broken after the latest changes when adding extra functionalities into the existing of the build.

- **System Integration Testing**

  This is the Testing performed on the Application under test, to verify the entire application works as per the requirements.

  Critical Business scenarios were tested to make sure important functionality in the application works as intended without any errors.

- **Regression Testing**
  Regression testing was performed each time a new build is deployed for testing which contains defect fixes and new enhancements if any.

  Regression Testing is being done on the entire application and not just the new functionality and Defect fixes.

  This testing ensures that existing functionality works fine after defect fix and new enhancements are added to the existing application.

  Manual Test cases for new functionality are added to the existing manual test cases and executed.

7.Test Environment & Tools

Application URL	http://localhost:9000/
OS	Windows,OS
Browser	Google Chrome,Safari, Edge, IE, Opera.
Text Editor	Visual Studio Code & Intellij

8. Lessons Learned

We have got a deeper understanding for how JUnit tests are used and created as well as writing manual test cases from use cases that were given by the SDC. Planning how to test has been very hard but also a great learning experience, especially what to test, however the importance of a requirement and how time consuming it will be to get full coverage with tests, has led to the way we have prioritized the different requirements with our resources.

9. Recommendations

- Discord is a communication tool that has been used to keep communication with the Test development team, by hosting daily meetings. It is usually called “Skype for gamers” but it has some good screen sharing tools and good ways to send images and voice channels.

- Testpad has been used to store the outcome of all JUnit tests. For a simple but visually good looking test report.

- JUnit Tests are automatic tests that check inputs and expected output, these have come to be of good use by helping our testing team to understand how the application works. It is also a great way to detect bugs and isolate bugs.

10. Exit Criteria

- All test cases should be executed - **Yes**.
- JUnit Pass test coverage over 95% - **Yes**.
- Manual Test Cases should be easy to understand - **Yes**.
- All defects in Critical, Major, Medium severity should be verified and closed - **Yes**.

11. Conclusion/ Sign Off

As the Exit criteria were met and satisfied as mentioned in Section 10, this application is suggested to 'Go Live' by the Testing team.

12. Definitions, Acronyms, and Abbreviations

JUnit, is a java framework that is used to create automatic test cases.
TC, is an abbreviation for **Test Case**.
UC, is an abbreviation for **Use Case**.
SDC, is an abbreviation for **Software Development Company**.
OS, is an operating system.

13. Time Log

Role	Description	Estimated Time(H)	Actual Time(H)	Date
Test manager	Planning Test Report.	3	3	2020-12-08
Test manager	Adding images of JUnit Tests in Metrics.	3	3	2020-12-08
Test manager	Adding documentation for **Types of testing performed, Test Environment & Tool, Lessons Learned, Recommendation, Best Practice, Exit Criteria, Conclusion/ Sign Off, Definitions, Acronyms, and Abbreviations, Time Log.**	8	7	2020-12-09
Test manager	Adding documentation and reviewing, **Types of testing**	2	2	2020-12-09

	performed, Lessons Learned, Recommendation, Exit Criteria, Conclusion/ Sign Off, Definitions, Acronyms, and Abbreviations, Time Log.			