|  |  |  |  |
| --- | --- | --- | --- |
| *Version* | *Date* | *Author* | *Comment* |
| 0.1 | 27.04.2023 | Anja Niedermeier | Created, set up basic structure |
| 0.3 | 28.04.2023 | Anja Niedermeier | Add Introduction and Test Strategy |
|  |  |  |  |
|  |  |  |  |

*Supplier*

Project Leader: Samara Dominik (inf21001@lehre.dhbw-stuttgart.de)

Product Manager: Martin Rittmann (inf21157@lehre.dhbw-stuttgart.de)

System Architect: Marcel Hintze (inf21056@lehre.dhbw-stuttgart.de)

Test-Manager: Anja Niedermeier (inf21097@lehre.dhbw-stuttgart.de)

Developer: Severin Helms (inf21047@lehre.dhbw-stuttgart.de)

Technical Documentation: Tom Engelmann (inf21010@lehre.dhbw-stuttgart.de)

Rotebühlplatz 41, 70178 Stuttgart

*Project*

AAS-Webclient

**System Test Plan**

TINF21C, Software Engineering I Practical project 2022/23

Business Case

*Customer*

Markus Rentschler, Christian Holder

Rotebühlplatz 41, 70178 Stuttgart

Content

[1 Introduction 3](#_Toc133570422)

[1.1 Scope 3](#_Toc133570423)

[1.2 Definitions 3](#_Toc133570424)

[1.3 Product names and attributes 3](#_Toc133570425)

[2 Features 3](#_Toc133570426)

[3 Overall strategy and approach 3](#_Toc133570427)

[3.1 Testing strategy 3](#_Toc133570428)

[3.2 System testing entrance criteria 3](#_Toc133570429)

[3.3 Application cases 3](#_Toc133570430)

[3.4 Testing Types 4](#_Toc133570431)

[3.4.1 Usability Testing 4](#_Toc133570432)

[3.4.2 Functional Testing 4](#_Toc133570433)

[3.5 Test data 4](#_Toc133570434)

[4 Test equipment 4](#_Toc133570435)

[5 Test schedule 4](#_Toc133570436)

[6 Test planning 4](#_Toc133570437)

[7 Test cases 4](#_Toc133570438)

# Introduction

This document specifies the testing approach and procedures for the software application “TINF21C\_Team1\_AAS-Server-Webclient”. Its purpose is to verify the functionality and reliability of the project before it is finalized. The AAS-Webclient is a web-based tool that allows users to add AAS-Servers and then view their contents, as well as filter and search for specific assets, all while providing a human-oriented graphic interface. The tests which are outlined in this document will be conducted using manual/statisch/dynamisch testing methods. Based on this test plan, the System Test Report (STR) will be created that will provide all the results of the performed tests. This test plan provides an overview of the system test, including the testing approach, test suites and test cases

## Scope

This STP will verify whether the mentioned software performs well on all the requirements and functionalities that have been described by the SRS (System Requirement Specification).

## Definitions

**TS** Test Suite  
**TC** Test Case

## Product names and attributes

The following test object will be examined:

|  |  |  |  |
| --- | --- | --- | --- |
| Ref.-Id | Product Number | Product Name | Product Description |
| 1 | Build v0.9 (commit 9a21d27) | AAS-Webclient | User friendly web-tool for displaying AAS-Servers and its contents while providing filter and search functionalities |

# Features

The following requirements must be verified through the testing process. The table below shows the mapping between the requirements from the SRS and the according Test suites or Test cases.

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement Id | Functionality | Priority | Testsuite Id |
|  |  |  |  |

# Overall strategy and approach

## Testing strategy

The test cases will be based on the requirements specified in the SRS. For more information about those, refer to chapter 2 Features. Each requirement must be covered by at least one test case. Where it makes sense, equivalence classes will be used to improve the efficiency and coverage of the conducted tests. The limits of possible inputs will also be tested.

## System testing entrance criteria

In order to start system testing, the software has to be signed off as ready by the development team.

## Application cases

Three main application cases can be identified: server management, asset exploration and asset display.

Server management is about being able to add AAS-Servers to the application which are then used to fetch data. Those servers should also be able to be removed again or switched out as needed. The system should be able to detect false server-URLs and throw an error, whenever a server cannot be reached.

Asset exploration involves all steps that involve browsing through the assets that appear as soon as the server has connected successfully. This includes first, reliability that all assets within the server are actually displayed. The user should then be able to scroll through the list of assets and see their names as well as a preview image. If needed, the user should also be able to search for assets by their name, filter for specific manufacturers and sort the assets by year.

The last application case is asset display. The user should be able to select a certain asset by clicking on it and then see a detailed view with all information that is available about it. This should also include all the sub-models.

## Testing Types

### Usability Testing

Part of the goal of the project included improving usability of the AAS-Webclient compared to already existing solutions. This is why also the usability should be tested. The intuitiveness and cosmetic representation will be inspected to evaluate whether user friendliness is given to the expected extent.

### Functional Testing

This ensures that all requirements and functions that were specified are working accordingly to the expectations.

## Test data

As all the data for the application is provided by AAS-Servers through REST-calls, this will be also the fact for the test conduction. The servers that will be used are:

* <https://v3.admin-shell-io.com/>
* <http://aas.murrelektronik.com:4001/aas/>
* https://admin-shell-io.com/5001/

# Test equipment

# Test schedule

# Test planning

# Test cases