System Test Plan

TINF21C, Software Engineering I
Practical project 2022/23

Project

AAS-Webclient

Customer

Markus Rentschler, Christian Holder

Rotebühlplatz 41, 70178 Stuttgart

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)

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
1.0	03.05.2023	Anja Niedermeier	Add test cases and finalize document
1.1	10.05.2023	Anja Niedermeier	Final adjustments

Content

1 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 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 features that are to be tested, the testing strategy and approach, the required testing equipment and the different test suites and test cases

1.1 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).

1.2 Definitions

STP System Test Plan
TS Test Suite
TC Test Case

1.3 Product names and attributes

The following test object will be examined:

Ref.-Id	Product Number	Product Name	Product Description
1	Build v1.0 (commit d5094d5)	AAS-Webclient	User friendly web-tool for displaying AAS-Servers and its contents while providing filter and search functionalities

2 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	Test suite ID
LF10: Import Server	Import a server by its URL and display its content	A	TC-SERVER-IMPORT-001, TC-SERVER-SWITCH-002, TC-SERVER-LOAD-003
LF20: Server Validation	Detect false server-URLs and throw errors	A	TC-SERVER-IMPORT-001
LF30: Error Handling	Handle and report errors like no entries found, unexpected errors, etc.	B	TC-SERVER-IMPORT-001, TC-SERVER-SWITCH-002, TC-SERVER-LOAD-003, TC-ASSET-SEARCH-002
LF40: Display content in a clear way	Display all contents in a clear and readable way	A	TC-ASSET-BROWSE-001,

			TC-SERVER-IMPORT
LF50: Display digital twin	Display more detailed information about individual assets/digital twins	A	TC-ASSET-BROWSE-001
LF60: Sort by year	Sort the displayed assets by manufacturer year	A	TC-ASSET-SORT-004
LF70: Filter for manufacturer	Filter the displayed assets by manufacturer	A	TC-ASSET-FILTER-003
LF80: Search for digital twin	Search for asset/digital twin by name	A	TC-ASSET-SEARCH-002

3 Overall strategy and approach

3.1 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.

3.2 System testing entrance criteria

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

3.3 Test Preparation

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

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 switched out as needed. The system should be able to detect false server-URLs and throw an error, whenever a server cannot be reached. It should be ensured, that all assets within the server are actually loaded.

Asset exploration involves all steps that involve browsing through the assets that appear as soon as the server has connected successfully. 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 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.

3.4 Testing Types

3.4.1 Usability Testing

User interface, cosmetic presentation and content will be tested for accuracy and general usability. The goal of usability testing is to ensure consistent and appropriate access and navigation to the different functionalities of the application.

3.4.2 Functional Testing

The goal of functional testing is to ensure that all requirements and functions that were specified in the SRS are working accordingly to the expectations. This refers to functional requirements, business goals or conditions as well as other requests that were coming into scope throughout the course of this project (like resolutions of issues, change requests or feedback)

3.5 Suspension Criteria and Resumption Requirements

This section will specify the criteria that will be used to suspend all or a portion of the testing activities on the items associated with this test plan.

3.5.1 Suspension Criteria

Testing will be suspended if an incident is found that does not allow further testing of the application. If testing is stopped and changes are made to the frontend or backend, they will be paused until the test manager decides that the resumption requirements are met

3.5.2 Resumption Requirements

Resumption of testing will be possible when the functionality that caused the suspension of testing has been fixed and retested successfully. In this case the whole system test plan is re-executed to ensure that the bug-fixing did not cause any other problems.

4 Test Equipment

The following prerequisites are required for testing:

- A computer with a browser supporting the HTML5 standards
- A working internet connection

5 Test data

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 only input that is to be made by the user is therefore a valid AAS-Server URL. The following table TD-001 will provide some valid and invalid URLs that will be used for testing.

TEST DATA:	TD-001	
DATASET	**URL**	**EXPECTED RESULT**
1	https://ccae4836-001e-48c2-a4f9-235554f9400b.ma.bw-cloud-instance.org/	PASS
2	http://aas.murrelektronik.com:4001/aas/	PASS
3	https://www.dhbw-stuttgart.de/	FAIL

Furthermore, for the tests concerning the asset exploration, the Server-URL and results that can be seen in TD-002 and TD-003 are used.

TEST DATA:	TD-002	
DATASET	**URL**	**EXPECTED RESULT**
1	https://ccae4836-001e-48c2-a4f9-235554f9400b.ma.bw-cloud-instance.org/	[REGISTRY, Norgren_B84G_4GK_AP3_RME, Norgren_ISOLine_PRA_802032_M_100, ARGO-HYTOS Filter Element EXAPOR MAX3, ARGO_HYTOS_Return_Filter_ES075, Parker_D1FPE50MB9NB0_ISDE8HU, Parker_HMI-2203250342054_HGDK8HU, Parker_PV046R2L1T1NMMC_JU64L8HZ, AAS_Type_CD55B20_50, AAS_Type_JSY205220, AAS_Demo_4WRPEH6, AAS_Demo_CytroPac, AAS_R412026837, AAS_R481712899, AAS_Type_SPAU-P10R-T-G18M-L-PNLK-PNVBA-M8D,

		AAS_Type_VUVS-L25-M52-AD-G14-F8-1C1, AAS_Type_DSBC-63-125-PPVA-N3, ExampleMotor]

TEST DATA:		TD-003	
URL		https://ccae4836-001e-48c2-a4f9-235554f9400b.ma.bw-cloud-instance.org/	
DATASET	**EQUIVALENCE CLASS**	**SEARCH STRINGS**	**EXPECTED RESULT**
1	String at beginning of asset	"aas"	[AAS_Type_CD55B20_50, AAS_Type_JSY205220, AAS_Demo_4WRPEH6, AAS_Demo_CytroPac, AAS_R412026837, AAS_R481712899, AAS_Type_SPAU-P10R-T-G18M-L-PNLK-PNVBA-M8D, AAS_Type_VUVS-L25-M52-AD-G14-F8-1C1, AAS_Type_DSBC-63-125-PPVA-N3]
2	Substring in the middle of asset	"filter"	[ARGO-HYTOS Filter Element EXAPOR MAX3, ARGO_HYTOS_Return_Filter_ES075]
3	Numbers	"7"	[ARGO_HYTOS_Return_Filter_ES075, AAS_R412026837, AAS_R481712899]
4	Special Characters	"-"	[ARGO-HYTOS Filter Element EXAPOR MAX3, Parker_HMI-2203250342054_HGDK8HU, AAS_Type_SPAU-P10R-T-G18M-L-PNLK-PNVBA-M8D, AAS_Type_VUVS-L25-M52-AD-G14-F8-1C1, AAS_Type_DSBC-63-125-PPVA-N3]
5	Exact name of asset	"ARGO-HYTOS Filter Element EXAPOR MAX3"	[ARGO-HYTOS Filter Element EXAPOR MAX3]
6	Search string with no results	"abcd"	[] + Error warning
7	Empty string	""	*All assets* (compare TD-002, dataset 1, expected result)

6 Test specifications

6.1 Naming conventions

Testcases and test suites will be named after the following structure:

Test suite = TS-**TSFUNC**
Test case = TC-**TSFUNC**-**TCFUNC**-**SEQNR**
TSFUNC = Abbreviation for the related test suite

TCFUNC = Abbreviation for the related requirement
SEQNR = sequential numbering of testcase within test suite (001)

6.2 Test suite <TS-SERVER>

6.2.1 Test case <TC-SERVER-IMPORT-001>

TEST CASE		
ID:	TC-SERVER-IMPORT-001	
NAME:	Import server by URL	
REQ.-ID:	LF10, LF20, LF30, LF40	
DESCRIPTION:	The test case verifies the correct functionality of the import of an AAS-server by its URL. It verifies the display of an error message in case the given URL is incorrect. The test case uses the test data from table TD-001. The test set up consists of a computer, with an active internet connection and a working browser.	
TEST STEPS		
STEP	**ACTION**	**EXPECTED RESULT**
1	Click on the button “Server Menu” in the upper right corner	Dropdown server menu opens, providing an input field for a server URL as well as some predefined AAS-server URLs
2	Click on one of the predefined Server URLs	The selected URL is displayed as the current server and its assets are loaded on the left side of the screen (names and preview images)
3	Click into the text input field for the Server URL and type in the URLs specified in TD-001	URLs that are marked with as “PASS” are loaded as the current server and their assets are loaded on the left side of the screen. URLs that are marked as “FAIL” produce an error message.

6.2.2 Test case <TC-SERVER-SWITCH-002>

TEST CASE		
ID:	TC-SERVER-SWITCH-002	
NAME:	Switch out servers	
REQ.-ID:	LF10, LF30	
DESCRIPTION:	The test case verifies the correct functionality of switching out a currently used server with another one. It verifies the display of an error message in case the given URL is incorrect. The test case uses the predefined server-URLs that are already provided within the application itself as test data. The test set up consists of a computer, with an active internet connection and a working browser.	
TEST STEPS		
STEP	**ACTION**	**EXPECTED RESULT**
1	Click on the button “Server Menu” in the upper right corner	Dropdown server menu opens, providing an input field for a server URL as well as some predefined AAS-server URLs
2	Click on one of the predefined Server-URLs	The selected URL is displayed as the current server and its assets are loaded on the left side of the screen (names and preview images)
3	Click on a different predefined Server-URL	The current server information in the top right corner switches to the newly selected server. The assets of the old server disappear from the

		left side of the window and only the assets of the new URL are displayed.

6.2.3 Test case <TC-SERVER-LOAD-003>

TEST CASE		
ID:	TC-SERVER-LOAD-003	
NAME:	Display all assets of server	
REQ.-ID:	LF10, LF30	
DESCRIPTION:	The test case verifies the correct functionality of displaying all assets which are contained in the currently used server. The test case uses the server-URL specified in TD-002 The test set up consists of a computer, with an active internet connection and a working browser.	
TEST STEPS		
STEP	**ACTION**	**EXPECTED RESULT**
1	Click on the button “Server Menu” in the upper right corner	Dropdown server menu opens, providing an input field for a server URL as well as some predefined AAS-server URLs
2	Click on the text input field for the server URL and type in the server-URL specified in TD-002. Then click on the “Add Server” button	The selected URL is displayed as the current server and its assets are loaded on the left side of the screen. (names and preview images) The exact same assets that are specified as expected results in TD-002 are displayed.

6.3 Test suite <TS-ASSET>

6.3.1 Test case <TC-ASSET-BROWSE-001>

TEST CASE		
ID:	TC-ASSET-BROWSE-001	
NAME:	Browse through different assets	
REQ.-ID:	LF40, LF50	
DESCRIPTION:	The test case verifies the correct functionality of selecting one of the displayed assets to see more information about it. The test set up consists of a computer, with an active internet connection and a working browser. Additionally, the application must already have a working connection to an AAS-Server, meaning that its different assets are already loaded on to the left side of the screen.	
TEST STEPS		
STEP	**ACTION**	**EXPECTED RESULT**
1	Move cursor to the left side of the screen onto the asset view and start scrolling up and down	The asset view scrolls up and down and reveals all assets that are available.
2	Click on one of the assets	The space on the right side fills up with a more detailed view of the selected asset, providing an image (if available), its name, as well as information about all the sub models, which can be expanded and collapsed.
3	Click on a different asset	The information about the old asset is replaced by the detailed view about the new asset. All information is replaced.

6.3.2 Test case <TC-ASSET-SEARCH-002>

TEST CASE		
ID:	TC-ASSET-SEARCH-002	
NAME:	Search for asset by name	
REQ.-ID:	LF80, LF30	
DESCRIPTION:	The test case verifies the correct functionality of searching for specific assets by their name or by a substring of their name. For testing, the inputs and expected results from the TD-003 table are used. As not all possible search inputs can be tested, those are categorized by several equivalence classes that are supposed to cover and represent all major groups of input possibilities. The test set up consists of a computer, with an active internet connection and a working browser. Additionally, the application must already have a working connection to an AAS-Server, meaning that its different assets are already loaded on to the left side of the screen. In this case the server URL provided in TD-003 must be used to verify all the results.	
TEST STEPS		
STEP	**ACTION**	**EXPECTED RESULT**
1	Click on the search text input field within the filter bar above the assets and start typing in a name.	Below the search field an autocomplete field appears with all assets that start with the same characters as the input.
2	Select one of the autocomplete suggestions	All assets except the chosen one disappear.
3	Click on the cross next to the search input field	The search field is empty and all assets are displayed again.
4	Click on the search text input field again	/
5	Type in the different search strings that are provided in TD-003 and click on the search icon	The assets that are specified in the expected results section of TD-003 for each input string are displayed. In case of an error, next to the search field, a error message appears.

6.3.3 Test case <TC-ASSET-FILTER-003>

TEST CASE		
ID:	TC-ASSET-FILTER-003	
NAME:	Filter for asset by manufacturer name	
REQ.-ID:	LF70	
DESCRIPTION:	The test case verifies the correct functionality of filtering for specific assets by the name of their manufacturer. The test set up consists of a computer, with an active internet connection and a working browser. Additionally, the application must already have a working connection to an AAS-Server, meaning that its different assets are already loaded on to the left side of the screen.	
TEST STEPS		
STEP	**ACTION**	**EXPECTED RESULT**
1	Click on the manufacturer button within the filter bar above the assets	A drop-down menu appears with all the manufacturers that can be found in the assets of the connected server.
2	Select one of the drop-down options	All assets are displayed that are from the selected manufacturer.

3	Click on the cross next to the search input field	All assets are displayed again.

6.3.4 Test case <TC-ASSET-SORT-004>

TEST CASE		
ID:	TC-ASSET-SORT-004	
NAME:	Sort assets by their manufacturing year	
REQ.-ID:	LF60	
DESCRIPTION:	The test case verifies the correct functionality of sorting the assets by the year of their manufacturing. The test set up consists of a computer, with an active internet connection and a working browser. Additionally, the application must already have a working connection to an AAS-Server, meaning that its different assets are already loaded on to the left side of the screen.	
TEST STEPS		
STEP	**ACTION**	**EXPECTED RESULT**
1	Click on the “Year” - button within the filter bar above the assets	A drop-down menu appears with the options “oldest first” and “newest first”
2	Select the option “oldest first”	All assets are sorted from oldest to newest asset
3	Select the option “newest first”	All assets are sorted from newest to oldest now