Customer Requirements Specification

TINF21C, Software Engineering I
Praxisproject 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	02.10.2022	Martin Rittmann	Created, first draft
0.2	23.10.2022	Martin Rittmann	Added Product Usage
0.3	02.11.2022	Martin Rittmann	Added non functional requirements
1.0	04.11.2022	Martin Rittmann	Final draft of the CRS
2.0	20.11.2022	Martin Rittmann	Feedback applied (remove add server from business process and add new figures)
2.1	10.05.2022	Martin Rittmann	Small changes, fixed typos, formulations

CONTENTS

Goal

The goal is to create a React based webpage for Asset Administration Shells (AAS) to provide a clearer presentation of digital twins based on AAS-Servers. The currently existing pages with similar functions provide a more technical view of digital twins and are therefore not user friendly. This webpage shall provide an easier way to browse and find digital twins with adding own AAS-Servers and search/filter functionalities. The frontend shall be designed in an intuitive way with reduced information, which could distract the user. The more specific and possibly confusing content shall only be delivered on request.

Product Environment

The Asset Administration Shell (AAS) is the Standard for modelling digital twins in Industry 4.0. Digital twins are models of so-called assets. Those can be tools, systems and products, etc. The twins shall be as similar to the existing asset as possible, even if it hasn't even been built before. It should simulate a real asset in every (technical) detail. You can virtually build machines with digital twins or test them in virtual environments before building the machine to test its functionality, usability, etc.

Product Usage

The following business processes, use cases and features shall be supported by the system.

Business Processes

<BP.001>: Find digital twin

Triggering Event:	The user wants to find a digital twin in an AAS-Server
Result:	The user can search, filter or browse through the digital twins shown on the webpage and receive further information at will.
Involved Roles:	User

Use Cases

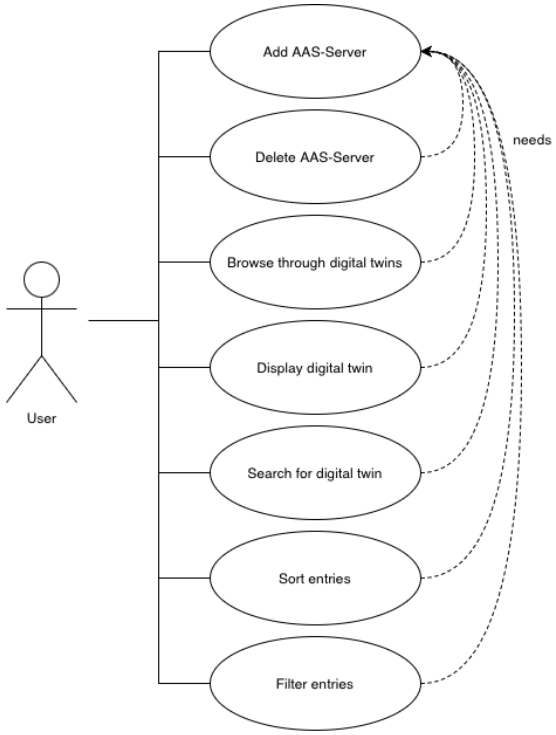

Figure 1: Use case diagram

<UC.001>: Add AAS-Server

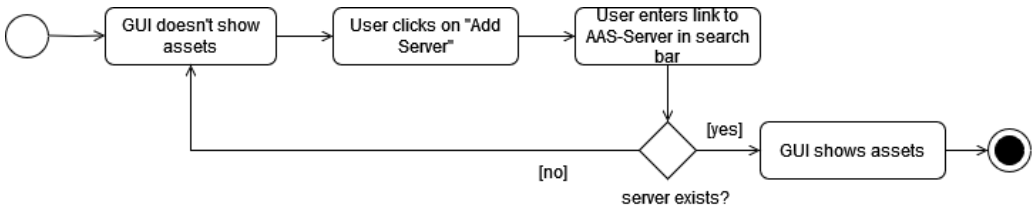

Figure 2: Add server activity diagram

Related Business Process:	<BP.001>: Find digital twin
Use Cases Objective:	User wants to add an AAS-Server by inserting the server's URL manually into the webpage.
System Boundary:	The application itself.
Precondition:	The user needs to have the URL to the AAS-Server. The webpage must be open and connected to the Internet. There must not be a currently active

	AAS-Server. If so, delete it before adding a new one.
Postcondition on success:	The server's content is displayed on the webpage.
Involved Users:	Every end-user of the application.
Triggering Event:	When the user presses "new AAS-Server" on the webpage, enters the URL and presses "Save".

<UC.002>: Remove AAS-Server

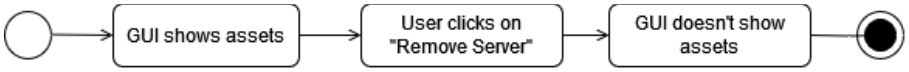

Figure 3: Remove server activity diagram

Related Business Process:	<BP.001>: Find digital twin
Use Cases Objective:	Removing a prior added AAS-Server and its content from the webpage.
System Boundary:	The application itself.
Precondition:	An AAS-Server has been added before.
Postcondition on success:	No content of the removed server is displayed anymore.
Involved Users:	Every end-user of the application.
Triggering Event:	When the user presses "remove AAS-Server".

<UC.003>: Browse through digital twins

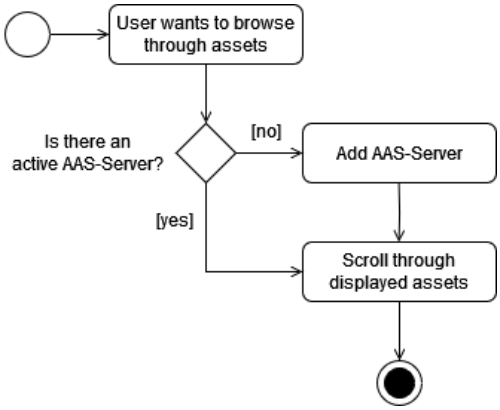

Figure 4: Browse digital twins activity diagram

Related Business Process:	<BP.001>: Find digital twin
Use Cases Objective:	The digital twins are displayed alphabetically sorted on the webpage. The user can scroll through them.
System Boundary:	The application itself
Precondition:	The user added a working AAS-Server.
Postcondition on success:	The user can browse through all content of the server and find digital twins coincidentally.
Involved Users:	Every end-user of the application.
Triggering Event:	When the user wants to look through all content of the server without the need to find a specific digital twin.

<UC.004>: Display digital twin

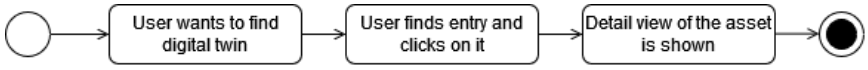

Figure 5: Display digital twin activity diagram

Related Business Process:	<BP.001>: Find digital twin
Use Cases Objective:	User can see the information of a specific digital twin
System Boundary:	The application itself
Precondition:	The user added a working AAS-Server.
Postcondition on success:	The user can see the information regarding a specific digital twin
Involved Users:	Every end-user of the application.
Triggering Event:	The user expands an entry of a digital twin on the webpage

<UC.005>: Search for digital twin

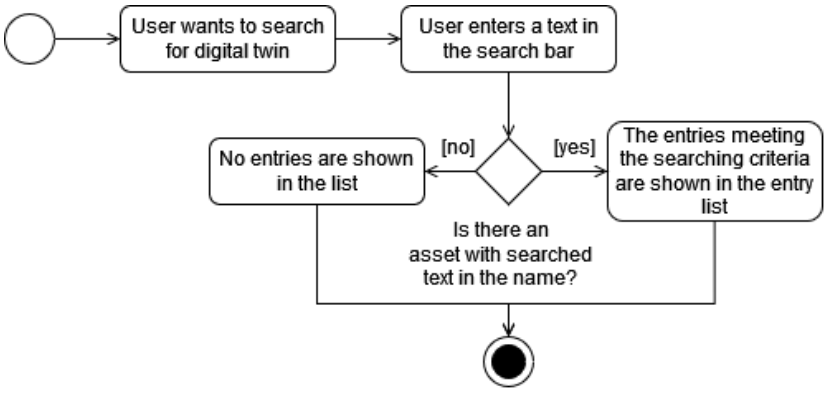

Figure 6: Search for digital twin activity diagram

Related Business Process:	<BP.001>: Find digital twin
Use Cases Objective:	Search for a specific digital twin by serial number or by free text search.
System Boundary:	The application itself

Precondition:	The user added a working AAS-Server.
Postcondition on success:	A reduced number of digital twins is displayed on the webpage in the same way it was displayed before the search.
Involved Users:	Every end-user of the application.
Triggering Event:	The user enters a text in the search bar and presses “search” or Enter and wants to find a specific digital twin.

<UC.006>: Filter entries

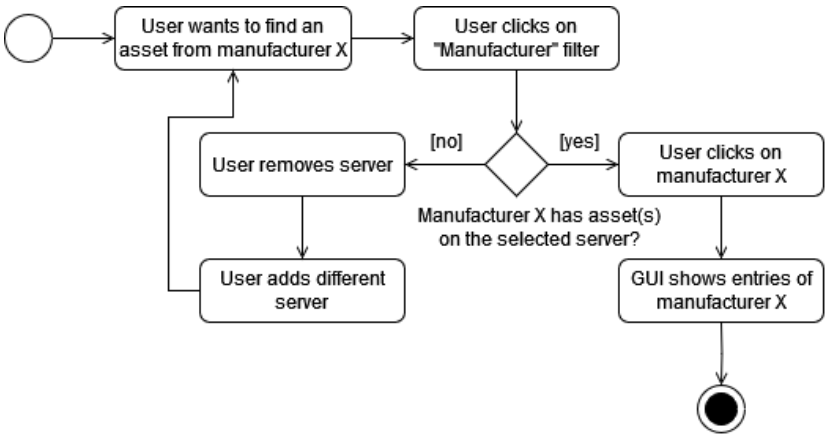

Figure 7: Filter entries activity diagram

Related Business Process:	<BP.001>: Find digital twin
Use Cases Objective:	Filter all content on the webpage to only show assets of one manufacturer name.
System Boundary:	The application itself.
Precondition:	The user added a working AAS-Server.
Postcondition on success:	The entries are specified and displayed on the webpage.
Involved Users:	Every end-user of the application.
Triggering Event:	The user enters wished filtering criteria in designated area.

<UC.007>: Sort entries

Figure 8: Sort entries activity diagram

Related Business Process:	*<BP.001>: Find digital twin*
Use Cases Objective:	*Filter all content on the webpage to show the newest/oldest assets at the top of the list based on manufacturing year*
System Boundary:	*The application itself.*
Precondition:	*The user added a working AAS-Server.*
Postcondition on success:	*The entries are sorted and displayed on the webpage.*
Involved Users:	*Every end-user of the application.*
Triggering Event:	*The user enters wished sorting criteria in designated area.*

Features

/LF10/ Import server

The webpage should be able to import a server by its URL and display its content.

/LF20/ Server validation

The system shall be able to detect false server-URLs when adding a new server and throw an error to the user.

/LF30/ Error handling

The system shall be able to handle errors (no entries found, unexpected errors, false server-URL, ...) and throw an error to the user.

/LF40/ Intuitive GUI

The webpage shall display a graphical user interface (GUI) to the user. This GUI must display every function provided to the user in a simple and intuitive way. It will be the only way to interact with the application.

/LF50/ Display content in a clear way

The entries including the digital twins are shown in a clear and readable way as a list one entry under the other while more information can be seen when clicking on the asset.

Product Data

/LD10/ Data

The data displayed in the webpage is delivered through an AAS-Server with REST-Calls. There is no functionality to export data or import other data than an AAS-Server by its URL.

Other Product Characteristics

This section describes the already known non-functional requirements for the product.

/NF10/ Browser

The webpage shall work in every Browser supporting the HTML 5 standard.

/NF20/ Efficiency

The webpage shall add servers and apply filters in the fastest way possible as well as the user being able to find desired results with the lowest possible amount of steps.

/NF30/ Usability

A user searching a specific product shall find that product as fast as possible, intuitively know how the webpage works and there shall be enough online documentation for new users. No further training or experience is required.