**System Requirements  
Specification**

***(Pflichtenheft)***

*(TINF21C, SWE I Praxisprojekt 2022/23)*

|  |  |
| --- | --- |
| *project:* | Modelling Wizard Improvements |
| *customer:* | Rentschler & Holder  Rotebühlplatz 41  70178 Stuttgart |
| *supplier:* | Team 4: Robin Ziegler, Dana Frey, Max Trumpp, Fabian Kreuzer, Nils Hoffmann, Michael Grote  Rotebühlplatz 41  70178 Stuttgart |

|  |  |  |
| --- | --- | --- |
| Version | Date | Comment |
| 1.0 | 07.10.2022 | created |
| 1.1 | 14.10.2022 | Added ‘Introduction’ and ‘Use Cases’ |

Table of contents

1. Introduction

Our project deals with the already existing program called „Modelling Wizard“. The program can create a device model in a graphical user interface (GUI). It is possible to add device interfaces (such as physical ports) or file attachments to the created device model. If the user already owns descriptive files, those files can be used to create a device model as well. The current version of the Modelling Wizard is only capable of handling AMLX, AML, EDZ, IODD and GSD imports and AMLX exports.

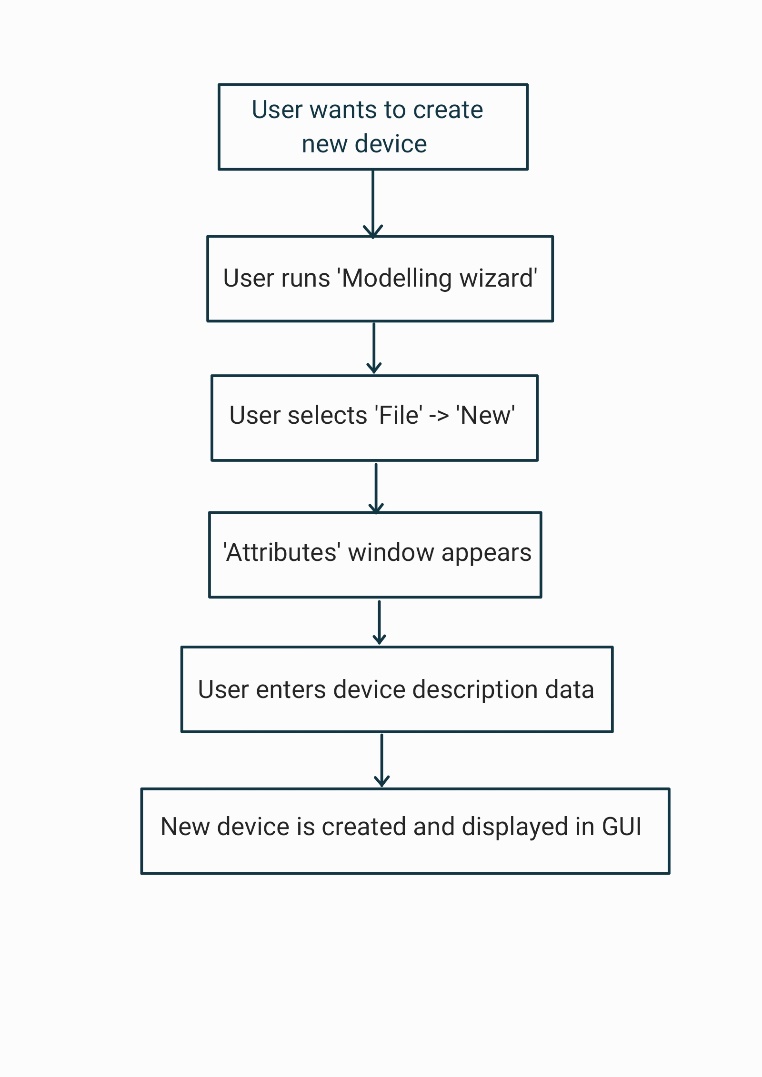
The goal of our project is to create a better version of the Modelling Wizard, which should also support a greater variety of import and export file types.

2. Use Cases

Most of the following Use Cases have been taken over by the previous project team. Reason is that our project builds on the previous project and many Use Cases haven’t changed during that time. Every underlined subject represents a new Use Case which has been added in terms of our project.

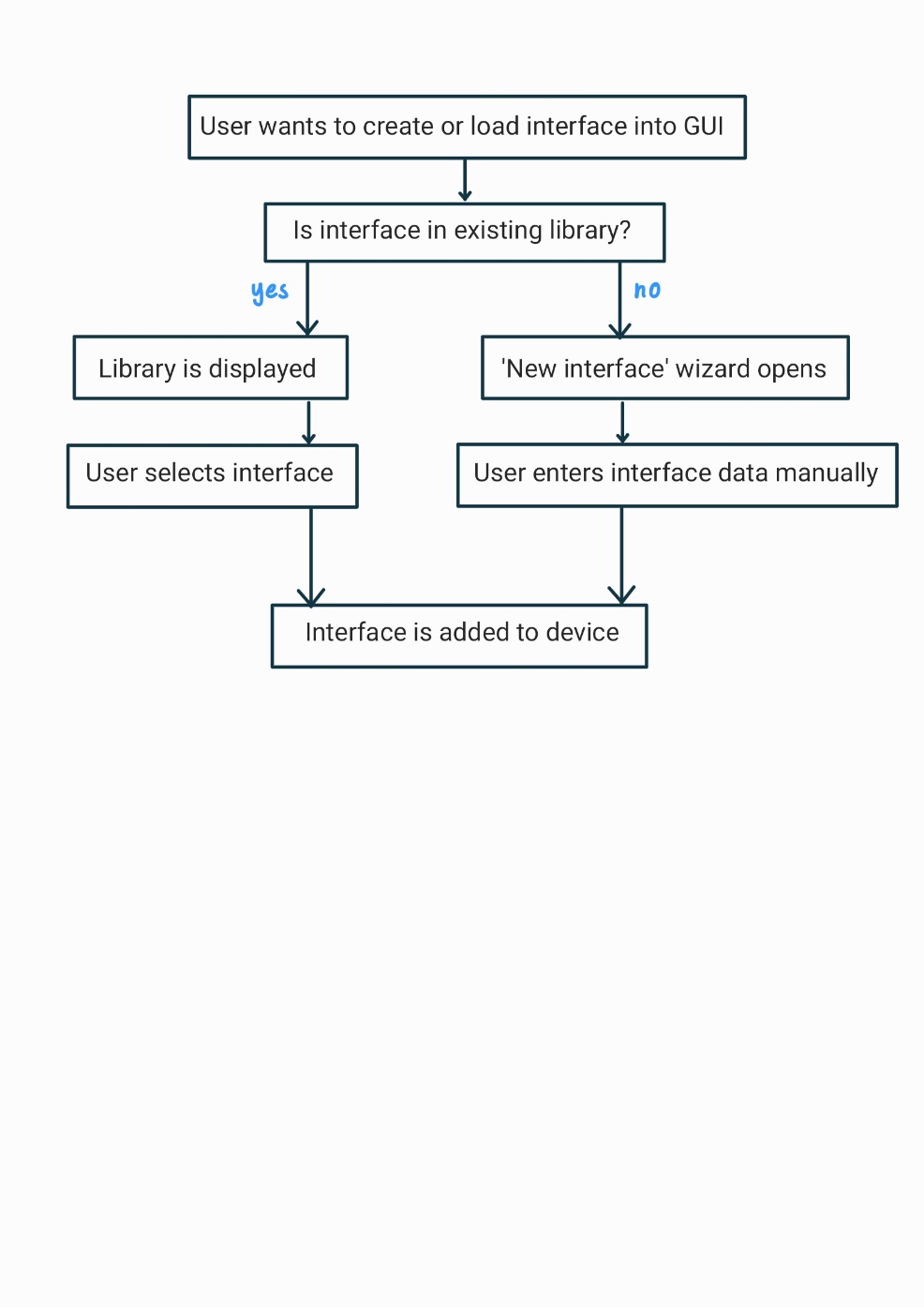
2.1 <UC.001> Create new device

|  |  |
| --- | --- |
| Use Case's Objective: | User wants to create a device by inserting the data manually into the user interface of the application |
| System Boundary: | The application itself |
| Precondition: | The user needs to have the minimal required data for the device on hand. The program needs to be installed on the user’s system and opened. |
| Postcondition on success: | The entered data is displayed completely and correctly |
| Involved Users: | Every end-user of the application |
| Triggering Event: | When the user opens the application and uses the 'New' function to create a new device |



2.2 <UC.002> Create interface or load interface from library

|  |  |
| --- | --- |
| Use Case's Objective: | Creating a device interface by inserting the data manually into the user interface. Or to add an interface from one of the existing libraries |
| System Boundary: | The application itself |
| Precondition: | The user needs to have the minimal required data for the device or interface to be added. |
| Postcondition on success: | The user has submitted the specific data completely and correctly |
| Involved Users: | Every end-user of the application |
| Triggering Event: | When the user wants to add or create a device interface |

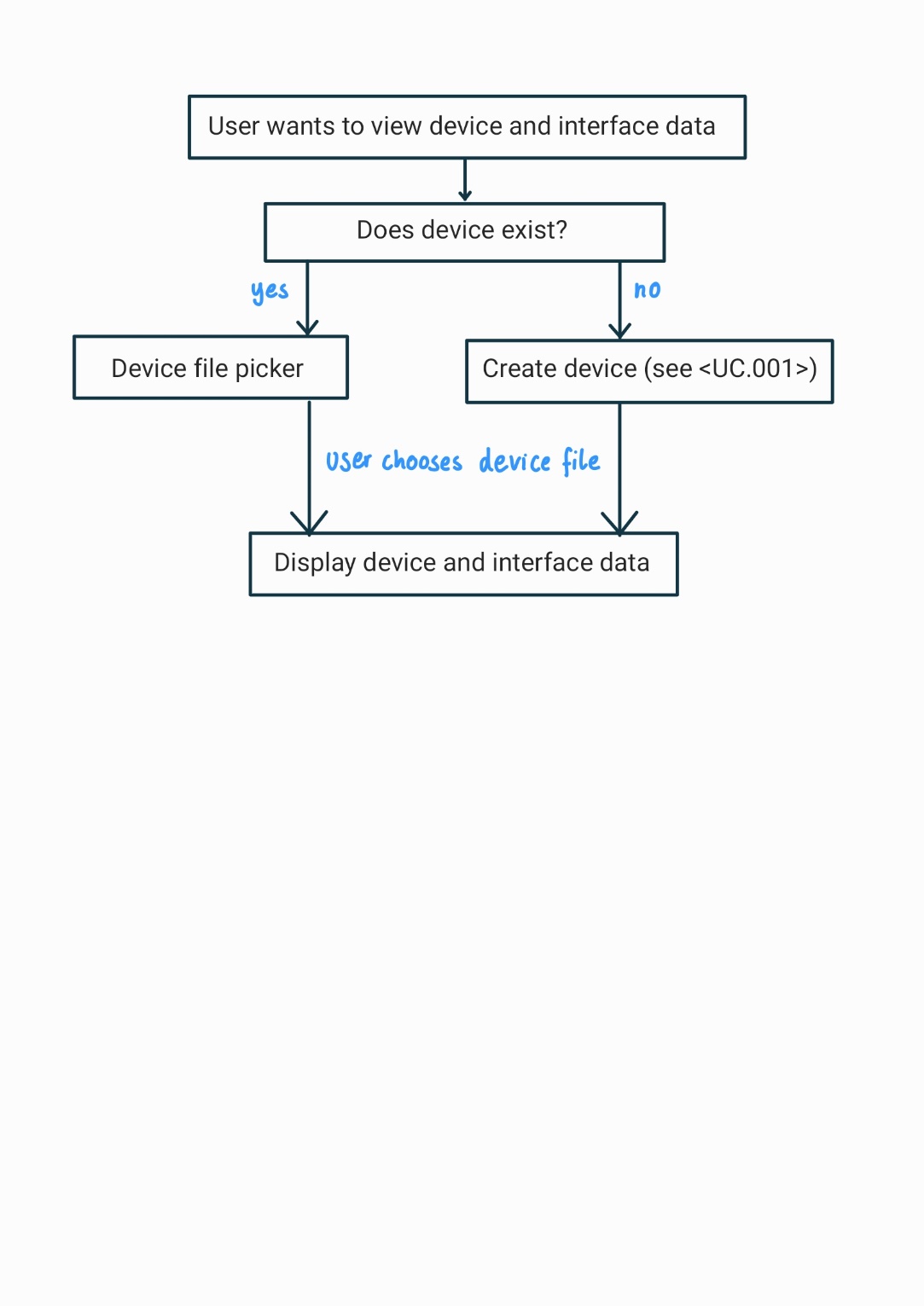


Supported input formats are:

CAEX 2.15, CAEX 3.0, EDS, AASX, EDZ, IODD, GSD, BMECat

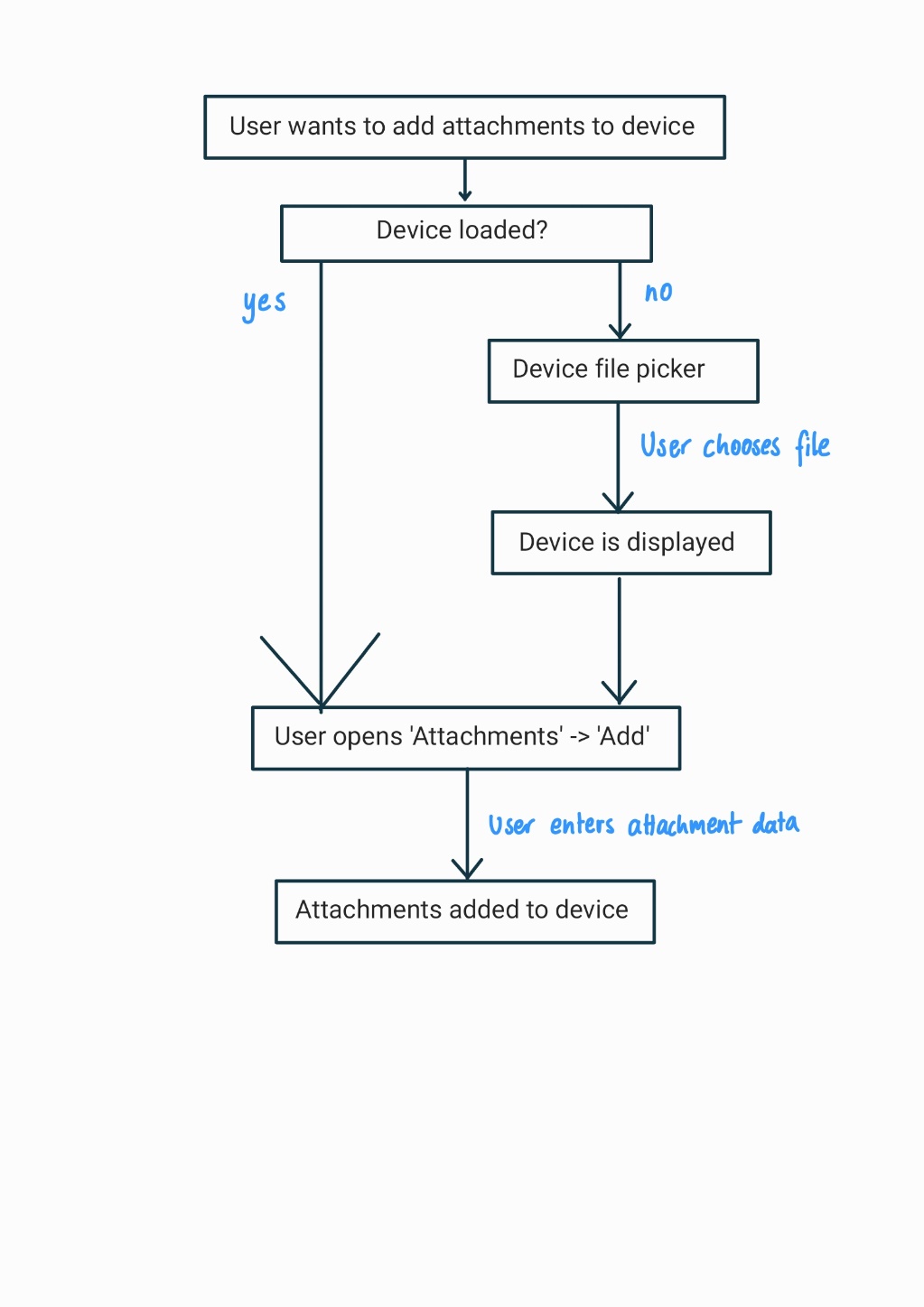
2.3 <UC.003> View device data and device interface data

|  |  |
| --- | --- |
| Use Case's Objective: | After at least one device was successfully added, the device data should be visible and editable on the user interface |
| System Boundary: | The application itself |
| Precondition: | The user added or loaded a device |
| Postcondition on success: | The user added or loaded at least one device successfully |
| Involved Users: | Every end-user of the application |
| Triggering Event: | When the user wants to view device data and device interface data |



2.4 <UC.004> Add attachments to device

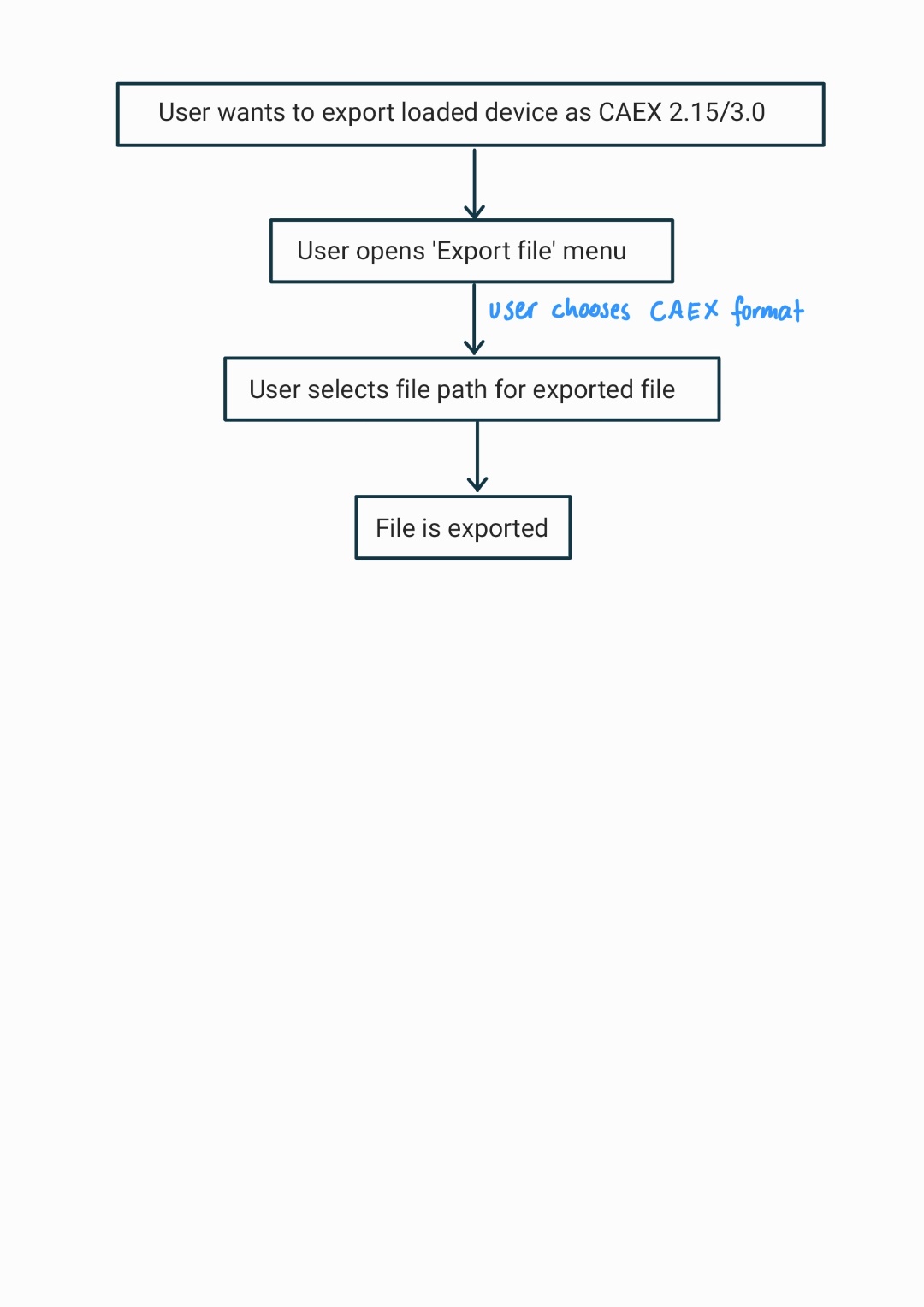
|  |  |
| --- | --- |
| Use Case's Objective: | It is possible to add an attachment to the object, such as a manufacturer's icon |
| System Boundary: | The application itself |
| Precondition: | The user added or loaded a device |
| Postcondition on success: | The user added or loaded at least one device successfully |
| Involved Users: | Every end-user of the application |
| Triggering Event: | When the user has the need to edit device data and add attachments such as icons. |



2.5 <UC.005> Outputs

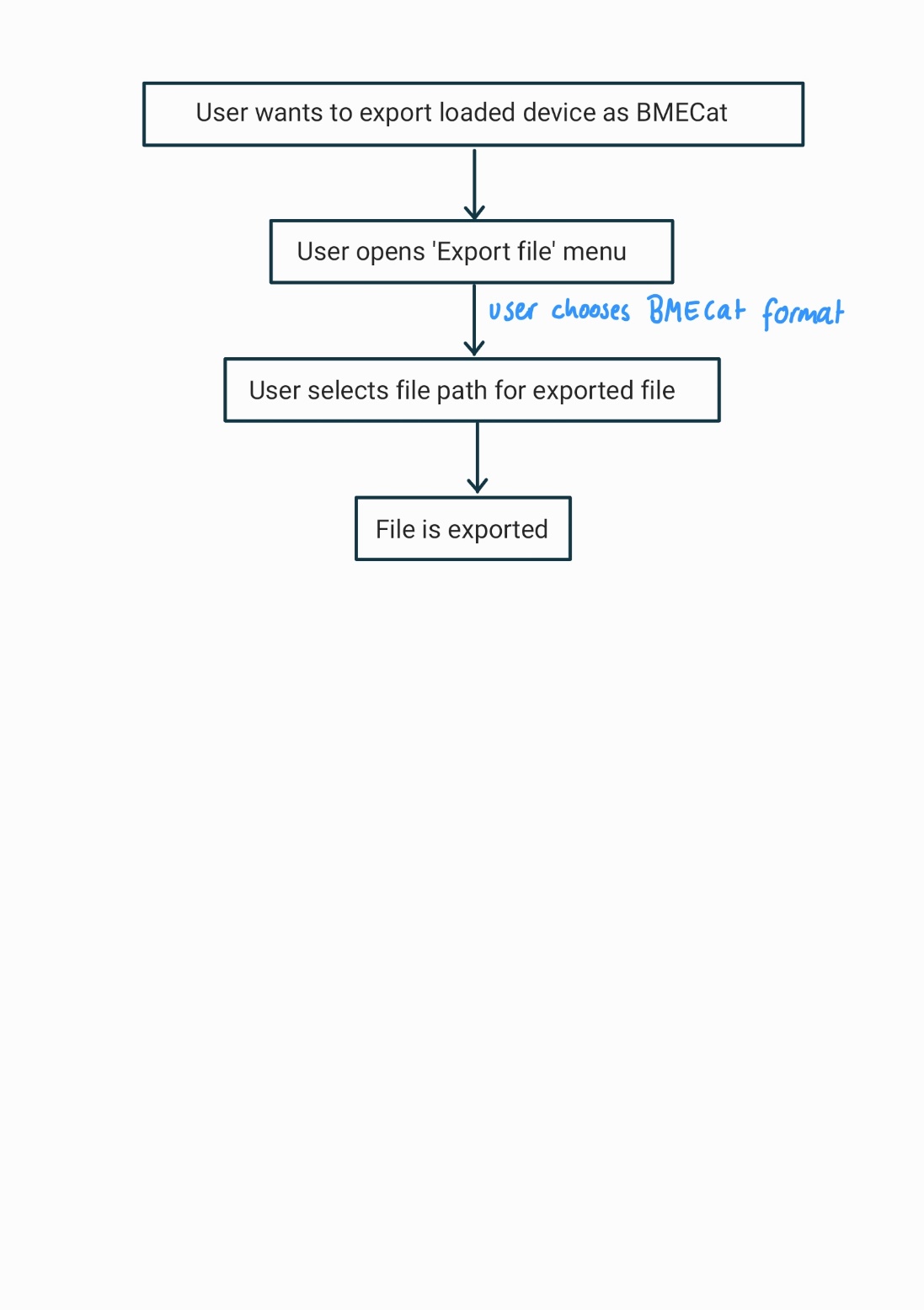
2.5.1 <UC.0051> Format output as CAEX version 2.15/3.0

|  |  |
| --- | --- |
| Use Case's Objective: | Make the export to CAEX formats possible for devices |
| System Boundary: | The application itself |
| Precondition: | The user added or loaded a device |
| Postcondition on success: | The user added or loaded at least one device successfully |
| Involved Users: | Every end-user of the application |
| Triggering Event: | When the user wants to save a device in the CAEX format |



2.5.2 <UC.0052> Format output as BMECat

|  |  |
| --- | --- |
| Use Case's Objective: | Make the export to the BMECat format possible for devices |
| System Boundary: | The application itself |
| Precondition: | The user added or loaded a device |
| Postcondition on success: | The user added or loaded at least one device successfully |
| Involved Users: | Every end-user of the application |
| Triggering Event: | When the user wants to save a device in the BMECat format |



3. Features

3.1 <LF10> Import

The application should be able to import a file by the absolute path to the file. This import supports files of the file types: CAEX 2.15, CAEX 3.0, EDS, AASX, EDZ, IODD, GSD and BMECat.

3.2 <LF20> File validation

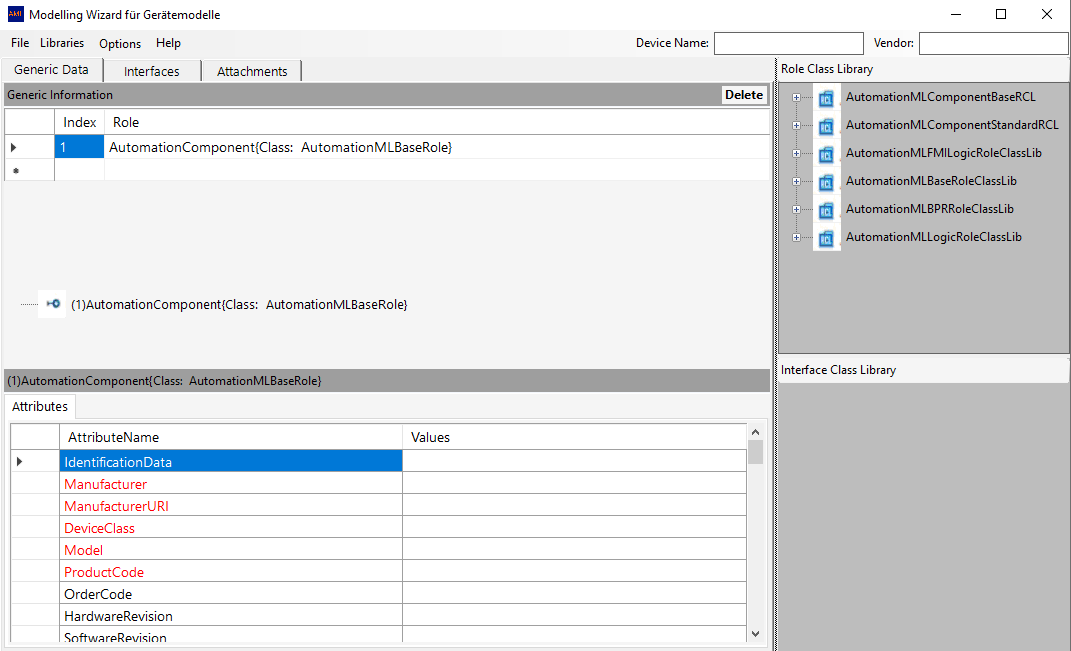
The system shall be able to detect wrongly formatted imported files and throw an error to the user.

3.3 <LF30> Error handling

The system shall be able to handle errors (unexpected shut down, wrongly formatted files, ...) and throw an error to the user.

3.4 <LF40> GUI

The system should display a graphical user interface after startup of the standalone application. The user will interact with this GUI for every other functionality of the application. For this, the application must be converted from a plugin to a full application on its own.



3.5 <LF50> Display device in a readable way

When a device is loaded or created the attributes of the element should be displayed directly and easily readable for the user.

3.6 <LF60> Edit device

When the attributes of a loaded device are displayed to the user, the user should be able to edit every attribute he wants to change.

3.7 <LF70> Create device

When the application is started, the user should be able to create a new and empty device model.

3.8 <LF80> Export device

When the user has edited a device, he should be able to save the device to a file.

4. Product data

5. Bug fixes

*Yet to be determined.*

6. References

[1] <https://github.com/H4CK3R-01/TINF20C_ModellingWizard_Devices/wiki/1.-Software-Requirements--Specification#UC3>

7. Glossary

|  |  |
| --- | --- |
| AML | Automation Markup Language is an open standard data format for storing and exchanging plant planning data |
| AML DD | AML Device Description |
| AMLX | AML Package |
| CAEX | Computer-Aided Engineering Exchange |
| CLI | Command Line Interface |
| GUI | Graphical User Interface |
| IODD | Input/Output Device Description |