Embedded-World

T-EMB-800

Embedded-World

Software Architecture Specifications

Table of contents

[Introduction 3](#_Toc500948735)

[1. Project context 3](#_Toc500948736)

[2. Global architecture 3](#_Toc500948737)

[3. Component description 3](#_Toc500948738)

[a. Component 1 3](#_Toc500948739)

[b. Component 2 3](#_Toc500948740)

[c. Component 3 3](#_Toc500948741)

[4. Traceability matrix 3](#_Toc500948742)

# Introduction

The aim of this software architecture specification (SAS) is to present the technical elements necessary for the Embedded-World project.

# Project context

The project consists of create failure reporter software for Airbus Plane.

# Global architecture

In this software, we have 4 components. The first component is parser. He initializes 2 structures named header and failure. Header contains 3 information’s : id of plane, type of plane and numbers of failures. The second structure contains 5 information’s concerning detected failure. This parser set a different values received by user interface. The second component is a parser. This parser take information set by first parser and verify if received data is available. The 3th component is user interface. He is input of a software. And the last component is writer. He is used for create failure file. This file contains every detected failure.

# Component description

## First Parser

He take a input error and initializes 2 structures named header and failure. Header contains 3 information’s : id of plane, type of plane and numbers of failures. The second structure contains 5 information’s concerning detected failure. This parser set a different values received by user interface.

## Validation Parser

This parser take information set by first parser and verify if received data is available. He verify a information with xslx document provided.

## User interface

He is input of software. He is simple interface create with C# and .NET. He contains 2 method for simule different error: automatic and manual.

## Writing

This component create and write error in a file named by a convention.

# Traceability matrix

This matrix makes the correspondence between components, classes, functions and requirements developed in the request for proposal.

All REQ\_DESIGN is ok excepted 030.

|  |  |  |  |
| --- | --- | --- | --- |
| Id requirement | ***Requirement description*** | ***Component*** | ***Function / action*** |
| REQ\_FUNC\_010 | aised errors must be encoded with the following format ... | writing | faillist\_writing\_header  faillist\_writing\_failure |
| REQ\_FUNC\_020 |  | Parser 1 and 2 | Faillist\_parse\_id\_plane  Faillist\_validate\_id\_plane |
| REQ\_FUNC\_030 |  | Parser 1 and 2 | Faillist\_parse\_type\_plane  Faillist\_validate\_type\_plane |
| REQ\_FUNC\_040 |  | Parser 1 and 2 | Faillist\_parse\_nb\_failures  Faillist\_validate\_nb\_failures |
| REQ\_FUNC\_050 | Ok | Parser 1 and 2 | Faillist\_parse\_date  Faillist\_validate\_date |
| REQ\_FUNC\_060 | ok | Parser 1 and 2 | Faillist\_parse\_criticity  Faillist\_validate\_criticity |
| REQ\_FUNC\_070 | ok | Parser 1 and 2 | Faillist\_parse\_id\_failure  Faillist\_validate\_id\_failure |
| REQ\_FUNC\_080 |  | Parser 1 and 2 | Faillist\_parse\_comment\_failure\_size  Faillist\_validate\_ comment\_failure\_size |
| REQ\_FUNC\_090 |  | Parser 1 and 2 | Faillist\_parse\_comment\_failure  Faillist\_validate\_ comment\_failure |
| REQ\_FUNC\_100 |  | Parser 1 and 2 | Faillist\_parse\_id\_component  Faillist\_validate\_id\_component |
| REQ\_FUNC\_110 |  | writing | faillist\_writing\_header  faillist\_writing\_failure |
| REQ\_FUNC\_120 |  | writing | faillist\_writing\_header  faillist\_writing\_failure |
| REQ\_FUNC\_130 |  | Parser 1 and 2 |  |
| REQ\_FUNC\_140 |  | Parser 1 and 2 |  |
| REQ\_FUNC\_150 |  | Parser 1 and 2 |  |
| REQ\_FUNC\_160 |  | Parser 1 and 2 |  |
| REQ\_FUNC\_170 |  | Parser 1 and 2 |  |
| REQ\_FUNC\_180 |  | Parser 1 and 2 |  |
| REQ\_FUNC\_190 |  | Parser 1 and 2 |  |
| REQ\_FUNC\_200 |  | Parser 1 and 2 |  |
| REQ\_TEST\_010 |  |  |  |
| REQ\_TEST\_020 |  |  |  |
| REQ\_TEST\_030 |  |  |  |
| REQ\_TEST\_040 |  |  |  |
| REQ\_TEST\_050 |  |  |  |
| REQ\_TEST\_060 |  |  |  |
| REQ\_TEST\_070 |  |  |  |
| REQ\_TEST\_080 |  |  |  |
| REQ\_TEST\_090 |  |  |  |
| REQ\_TEST\_100 |  |  |  |
| REQ\_TEST\_110 |  |  |  |
| REQ\_TEST\_120 |  |  |  |
| REQ\_TEST\_130 |  |  |  |
| REQ\_TEST\_140 |  |  |  |
| REQ\_TEST\_150 |  |  |  |
| REQ\_DATA\_010 |  | Parser 1 | Failist\_parse\_id\_plane |
| REQ\_DATA\_020 |  | Parser 2 | Failist\_validate\_id\_plane |
| REQ\_DATA\_030 |  | Parser 1 |  |
| REQ\_DATA\_040 |  | Parser 2 |  |
| REQ\_DATA\_050 |  | Parser 1 |  |
| REQ\_DATA\_060 |  | Parser 2 |  |
| REQ\_DATA\_070 |  | Parser 1 |  |
| REQ\_DATA\_080 |  | Parser 2 |  |
| REQ\_DATA\_090 |  | Parser 2 |  |
| REQ\_DATA\_100 |  | Parser 1 |  |
| REQ\_DATA\_110 |  | Parser 2 |  |
| REQ\_DATA\_120 |  | Parser 1 |  |
| REQ\_DATA\_130 |  | Parser 2 |  |
| REQ\_DATA\_140 |  | Parser 1 |  |
| REQ\_DATA\_150 |  | Parser 2 |  |
| REQ\_DATA\_160 |  | Parser 1 |  |
| REQ\_DATA\_170 |  | Parser 2 |  |
| REQ\_DATA\_180 |  | Parser 1 | Define MAX\_COMMENT\_FILE\_SIZE |
| REQ\_DATA\_190 |  | Parser 1 |  |
| REQ\_DATA\_200 |  | Parser 2 |  |