OMMA

(**O**nline **M**echanical **M**easurement **A**nalysis)

TECHNICAL SPECIFICATION

Author: Ronald Rotter

Changelog

Date	Changed by	Changes performed
2023-12-21	Ronald Rotter	Initial version
2023-12-27	Ronald Rotter	added System Architecture
2024-01-08	Ronald Rotter	Added OMMA team members from Acoustic Engineering
2024-01-10	Ronald Rotter	Added Requirement Gathering Matrix
2024-01-18	Ronald Rotter	Updated Requirement Gathering Matrix
2024-01-24	Ronald Rotter	Updated Requirement Gathering Matrix Added data input formats
2024-01-25	Ronald Rotter	Added data input format detail definitions
2024-01-29	Ronald Rotter	Enhanced data input format
2024-01-30	Ronald Rotter	Enhanced data input format
2024-02-19	Ronald Rotter	Enhanced data input format changed example file links to non-embedded links

Table of Contents

1 Team

1.1 Project Management Office

- Christian Svatek (Sponsor)

1.2 Software Engineering

- Ronald Rotter (Project Manager, Database Architect, Software Architect)
- TBA (Backend Development, Frontend Development)

1.3 Software Users

1.3.1 Quality Engineering

- Florian Knar-Seufert (Head of Requirement input)
- Thomas Schilk (Requirement input)
- Thomas Doppelhammer (Requirement input)
- Anton-Karl Nedic (Requirement input)
- Walter Eschig (Requirement input)

1.3.2 Mechanical Engineering

- Markus Bogad (Requirement input)
- Ronald Sulz (Requirement input)

1.3.3 Acoustic Engineering

- Manuel Mefleh
- Andreas Hintennach

1.3.4 Process Engineering

- Florian Gruber (Requirement input)
- Bernhard Seiwald (Requirement input)

2 Source Code Control and Issue tracing

Will be performed by GitLab and can be found here:
https://ssiv-git.ssi.local/mma_group/omma_source

3 Introduction

3.1 Problem Description

The MMA tool (former OGPy tool) which is currently used to perform mechanical measurement analysis is very hard to maintain, has a lot of bugs and can only handle local data. The import and export capabilities are not very sophisticated.

For this reason, the whole application will be refactored. The new MMA tool can be accessed online via browser (OMMA – online mechanical measurement analysis). The raw measurement data as well as the modifications and statistic reports performed by the user will be stored in a central database.

3.2 System Architecture

An overview of the system architecture can be found here:
Sharepoint Link: system_architecture.pptx
Relative Location: assets/system_architecture.pptx

4 Requirements

4.1 Requirement Gathering Matrix

Dept.	Employee	Raw Data Import	Pre Proc.	Graph. Analysis	Stat. Analysis	Reports	Data Export
QE	Florian KS	X	X	X	X	X	X
	Thomas S	X	X				
	Thomas D	X	X				
	Karlo N	X	X	X			
	Walter E		X				
ME	Markus B			X			X
	Ronald S			X			X
AE	Manuel M	(X)		X	X	X	X
	Andreas H	(X)		X	X	X	X
	Martin S	(X)		X	X	X	X
	Patrick G	(X)			X	X	X
PE	Florian G	X	(X)	X	X	X	
	Bernhard S	X	(X)	X	X	X	

Table 1: Requirement Gathering Matrix

4.2 General Requirements

- A software with web-interface using the following technologies:
  - FastAPI
  - SQLModel
  - Jinja
  - HTMX
  - tailswindCSS / daisyUI
- User authorization shall be done via LDAP
- The software needs to be multilingual (e.g. Englisch and Chinese language)
- Users shall be able to save individual configuration (e.g. language, views, …)
- There shall be various user groups defined with different rights (e.g. read only, admin, data import)

4.3 Technical Requirements

4.3.1 Input Data Formats

Overview of the input data Formats:
Sharepoint Link: data_sources.xlsx
Relative Location: assets/data_sources.xlsx

The various input data formats are described in detail in the next chapters. The order of those chapters correlates to the priority order in the list above.

Example files of the different formats can be found in this folder (there are subfolders of the same name as the “Unique Name” in the list above:
https://soundsolutions.sharepoint.com/:f:/r/sites/VIE-PRO_LP_OMMA/General_Docs/01_technical_specification/assets/data_sources_details?csf=1&web=1&e=MXXQCC

4.3.1.1 auto_csv_Keyence_IM-8020_VIE

Example of auto_csv_Keyence_IM-8020_VIE:
Sharepoint Link: keyence1.csv
Relative Location: assets\data_sources_details\auto_csv_Keyence_IM-8020_VIE\examples\keyence1.csv

Each single file of this type can contain one or more production batches. One production batch can by separated in one or more single files of this type.

When a data source file is loaded, there shall always be the question of whether:

- the part numbering shall be used as provided by the data source file
- the part numbering shall be continued on the last one found in the database
- custom numbering of part number

The following use cases are possible:

- batch number does not exist in database
  → use part numbers as selected in dialog
- batch number does exist in database and dimensions (names) differ from existing datasets
  → add new dimension measurement values according to part numbers as selected in dialog
- batch number does exist in database and dimensions are the same as the existing ones in DB
  → User dialog: ask for reason of re-measurement and store the reason together with the measurement data in the database.

The example data below refers to the example file above.

The following data needs to be extracted from this particular data format:

- Data from filename
  - Expression: <YYYYMMDD><equipment_manufacturer>_<equipment_type>[_<batch>]
  - Example: 20230925Keyence_IM-8020_SSIV2023-QC0196
    - date: 25.09.2023
    - equipment_manufacturer: Keyence
    - equipment_type: IM-8020
    - batch (or something else): SSIV2023-QC0196
      (is optional and can be ignored by the import algorithm)

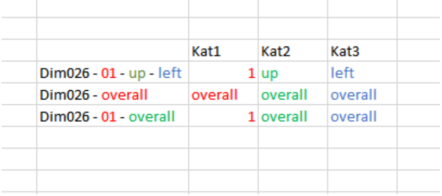
- Header Information
  It needs to be stated, that there is header information not only in the actual header of the file, but also in each row of the table. The header information can be found in each row but usually belongs to one batch. So each rows which belong to one batch contain the same information.
  - measurement_program_name: can be extracted starting from col1/row5. If data from Workfront can be accessed via QC number, this column can be just stored but does not need to be divided into its single information values.
  - date_time: can be extracted starting from col2/row5
  - batch: can be extracted starting from col3/row5
  - engineer: can be extracted starting from col6/row5
  - fai_ids[]: field which contains all dimension names to be measured. In the data source file this is named either as “DimXXX” name or “FAIXXX” name (e.g. “Dim004”, “FAI004”). If in the data source file there is only one or two digit for the numbering it shall be changed to 3 digits (e.g. “Dim1” → “Dim001”, “Dim01” → “Dim001”)
    - Can contain a single dimension like e.g. “Dim002”
    - Can contain a dimension which was measured on several positions e.g. “Dim026 – 01” and “Dim026 – 02” (and maybe more of them). There can be multiple grouping tokens like e.g. “Dim026 – 01 – left – up”
  - fai_nominals[]: field which contains all nominial dimension values, starting in col8/row2
  - fai_upper_limits[]: field which contains all upper limit dimension values, starting in col8/row3
  - fai_lower_limits[]: field which contains all lower limit dimension values, starting in col8/row4

- Measurement data information
  Part specific measurement values can be found starting from row5 in the following columns:
  - part_nr: col4
  - dim_result: col5
  - fai_values[]: starting in col8 field which contains all measured dimension values

- Validation to perform
  - Validate if all data values correspond to the defined data type, e.g.:
    - Validate if all dimension values are double values. If there is “,” change it to “.” as comma.
  - If there are empty values store “Null”.
  - Validate if there are nominal and limit values for all dimension values

- List of all data fields and their types
  - measurement_program_name: string
  - date_time: datetime
  - equipment_manufacturer: string
  - equipment_type: string
  - batch: string
  - engineer: string
  - fai_ids[]: [string]
  - fai_nominals[]: [double]
  - fai_upper_limits[]: [double]
  - fai_lower_limits[]: [double]
  - fai_values[]: [double]
  - part_nr: integer
  - dim_result: [„N.i.O.“, „Fehler“, „OK“]

- Exporting functions
    - When exporting to Minitab: If a specific Dimension (e.g. “Dim002”) has multiple grouping tokens (e.g. “Dim002 – 01 – left” and in another column there is the same Dimension with less grouping tokes (e.g. “Dim002 – overall”) then the missing parts shall be filled with the last token from the row (Kat1, Kat2, Kat3 are sorting columns):



	Kat1	Kat2	Kat3
Dim026 - 01 - up - left	1	up	left
Dim026 - overall	overall	overall	overall
Dim026 - 01 - overall	1	overall	overall

4.3.1.2 man_XLS_MIXED_ZJ

Example of man_XLS_MIXED_ZJ:
Sharepoint Link: mixed_1.xlsm
Relative Location: assets\data_sources_details\man_XLS_MIXED_ZJ\examples\mixed_1.xlsm

Each single file of this type can contains one production batch. If there are more than one tabs containing data (usually named e.g. “SS1-1”, “SS1-2”, …) it means that a tool with multiple cavities has been used for this production batch.

The example data below refers to the example file above.

The following data needs to be extracted from this particular data format:

- Header Information
  There are 9 header cell fields which contain the header information in the following way:
  - Drawing_nr (named Part Number): col4/row4
  - Component_name (named Part Description): col4/row5
  - Drawing_Revision (named Revision): col4/row6
  - Supplier: col8/row4
  - Inspector: col8/row5
  - Date (YYYY-MM-DD): col8/row6
  - Tool# (Cavity) / Build (TrialRun Number) / DOE Name (not relevant): col18/row4
  - Notes/Comments: col18/row5
  - CMM/OMM Program version: col18/row6

- FAI specific measurement values can be found in the following cells:
  - Classification (named SPC): col2/starting from row11 (e.g. M -> 100% measurements)
  - fai_ids[]: col3/starting from row11
  - Description: col4/starting from row11
  - Target cpk (not relevant for now): col5/starting from row11
  - Dimension Type (Category): col6/starting from row11 (e.g. Tolerance, GD&T)
  - fai_nominals[] (tolerance): col7/starting from row11
  - fai_upper_tol[] (tolerance): col8/starting from row11
  - fai_lower_tol[] (tolerance): col9/starting from row11
  - Measurement Equipment (named Inspection Method): col10/starting from row11
  - 100% Gauged (means no actual values, just pass/fail): col11/starting from row11

- Part specific measurement values can be found in the following cells:
  - Starting with col35 in row10 the part numbers can be found
  - Starting with col35 and row11 the actual measurement values can be found. Where each column represents a part.

- Validation to perform
  - Validate if all data values correspond to the defined data type, e.g.:
    - Validate if all dimension values are double values. If there is “,” change it to “.” as comma.
  - If there are empty values store “Null”.
  - Validate if there are nominal and limit values for all dimension values
  - If 100% Gauged value is “Yes” then there are no measurement values, otherwise measurement values have to be defined.

- List of all data fields and their types
    - TBD