Summary

[1 Measurement strategy phase 2](#_Toc149114586)

[1.1 PURPOSE 2](#_Toc149114587)

[1.2 SCOPE 2](#_Toc149114588)

[1.3 LEVEL OF DECOMPOSITION 2](#_Toc149114589)

[1.4 LEVEL OF GRANULARITY 2](#_Toc149114590)

[1.5 FUNCTIONAL USERS 2](#_Toc149114591)

[2 Mapping Phase 3](#_Toc149114592)

[2.1 Identity FUNCTIONAL PROCESSES 3](#_Toc149114593)

[2.2 Identify OBJECTS OF INTEREST and DATA GROUPS 3](#_Toc149114594)

[2.3 Identity DATA ATTRIBUTES 4](#_Toc149114595)

[3 Measurement phase 4](#_Toc149114596)

[3.1 Loading of clinical data 4](#_Toc149114597)

[3.2 List model projects 5](#_Toc149114598)

[3.3 Open project 5](#_Toc149114599)

[3.4 Univariate and bivariate analysis 6](#_Toc149114600)

[3.5 Comparison of categorical features 6](#_Toc149114601)

[3.6 Bivariate analysis of continuous features 6](#_Toc149114602)

[3.7 Multivariate analysis 7](#_Toc149114603)

[3.8 SVC model training 7](#_Toc149114604)

[3.9 Model test 7](#_Toc149114605)

[3.10 Patient classification 8](#_Toc149114606)

[4 Measurement Summary 8](#_Toc149114607)

# Measurement strategy phase

## PURPOSE

The purpose of this measurement is to measure a functional size of the **U-CURE** application that can be used as a basis to estimate the required effort to build the software. U-CURE is based on a microservice architecture.

## SCOPE

The scope of the measurement is all of the FUR that are related to the **U-CURE** application, i.e. which are derived from the requirements in U-CURE-UC document.

## LEVEL OF DECOMPOSITION

The level of decomposition of this scope is that of a whole application. All the functionality described in the FUR that is in scope for this measurement resides in the same layer.

## LEVEL OF GRANULARITY

The requirements are at the standard level of granularity, meaning that the functional users are individual humans (Machine learning engineer, Doctor, Etc.) and not groups of these. The functional users that provide input data detect single occurrences of events that the U-CARE application must respond to.

By measuring at the standard level of granularity it is possible to use this measurement not only for the purpose of this measurement but also for benchmarking purposes, since most benchmark data is available at the standard level of granularity.

## FUNCTIONAL USERS

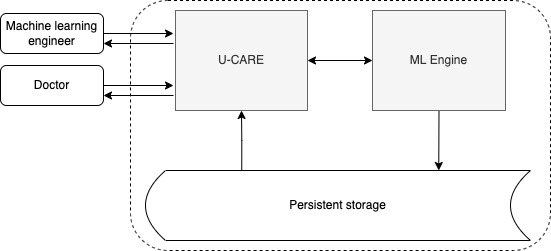
The functional users are the senders and/or intended recipients of data in the FUR of the U-CARE application.

Therefore, the functional users are:

* Machine learning engineer.
* Doctor.

There is exchange of data between the U-CARE application and other services, components or external applications.

The boundary of the U-CARE application is a conceptual interface between this piece of software and its functional users. It is indicated by the dashed line in the following picture:



The arrows represent the exchange of data between functional users, external components/services, and the U-CARE application.

# Mapping Phase

## Identity FUNCTIONAL PROCESSES

The following functional processes are identified in the FUR:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N. | Triggering event | Functional user | The data moved by the Triggering Entry | Functional process | UCs |
| 1 | Loading of clinical data | Machine learning engineer | Dataset details | Loading of clinical data | UC.AI.001 |
| 2 | List model projects | Machine learning engineer | All Projects | List model projects | UC.AI.002 |
| 3 | Open project | Machine learning engineer | Project ID | Open project | UC.AI.003 |
| 4 | Univariate and bivariate analysis | Machine learning engineer | Variables | Univariate and bivariate analysis | UC.AI.004 |
| 5 | Comparison of categorical features | Machine learning engineer | Categorical features | Comparison of categorical features | UC.AI.005 |
| 6 | Bivariate analysis of continuous features | Machine learning engineer | Continuous features | Bivariate analysis of continuous features | UC.AI.006 |
| 7 | Multivariate analysis | Machine learning engineer | Features  Analysis | Features  Analysis | UC.AI.007 |
| 8 | SVC model training | Machine learning engineer | Model parameters | SVC model training | UC.AI.008 |
| 9 | Model test | Machine learning engineer | Model parameters | Model test | UC.AI.009 |
| 10 | Patient classification | Doctor | Patient ID | Patient classification | UC.AI.0010 |

## Identify OBJECTS OF INTEREST and DATA GROUPS

The following objects of interest and data groups are identified:

|  |  |  |  |
| --- | --- | --- | --- |
| N. | Functional process | Object of interest | Data group(s) |
| 1 | Loading of clinical data | Project  EDA  Variable  Message | Dataset details  Project details  EDA  EDA details  Variable details  Transformation details  Messages |
| 2 | List model projects | Project  Message | All Projects  Project details  Messages |
| 3 | Open project | Project  Message | Project ID  Project details  Messages |
| 4 | Univariate and bivariate analysis | Variable  Analysis  T-test | Variables  Analysis request  Analysis details  t-test request  t-test details |
| 5 | Comparison of categorical features | Feature | Categorical features  Comparison request  Comparison results |
| 6 | Bivariate analysis of continuous features | Feature  Analysis | Continuous features  Analysis |
| 7 | Multivariate analysis | Feature  Analysis | Continuous features  Continuous feature analysis request  Analysis results |
| 8 | SVC model training | Model | Model parameters  Training results  Model status  Training details |
| 9 | Model test | Model | Model parameters  Test results |
| 10 | Patient classification | Patient  Model | Patient ID  SVC model parameters  Patient data  Patient classification |

## Identity DATA ATTRIBUTES

The following data characteristics are found in the object of interests specified in the preceding section:

|  |  |
| --- | --- |
| Object of interest | Data attributes |
| Patient | Patient ID, surname, name |
| Project | Project ID, title, description, dataset, created on |
| EDA | ID, Graph data, date, variable |
| Variable | Variable name, type, isFeature |
| Message | Message |
| Analysis | ID, Analysis details, date, variable1, variable2 |
| T-Test | ID, T-test data, variable |
| Feature | Feature ID, Variable ID, Feature name, type |
| Model | Model ID, Project ID, Model name, status, created on, trained on, tested on, test dataset |

# Measurement phase

The table below shows the functional processes identified in section 2.1, including all their movements of data groups (each of which describes an object of interest identified in section 2.2).

## Loading of clinical data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | FU | DG | OOI | DM | CFP |
| 1. The Machine learning engineer clicks the suboption “Load dataset”. |  |  |  |  |  |
| 2. The system displays a form containing the dataset fields. |  |  |  |  |  |
| 3. The Machine learning engineer fills in the dataset name and other details. | Machine learning engineer | Dataset details | Project | E | 1 |
| 4. The Machine learning engineer presses “Save”. |  |  |  |  |  |
| 5. The system saves the new project data. |  | Dataset details | Project | W | 1 |
| 6. The system requests the ML Engine to process the dataset. |  | Dataset details | Project | E | 1 |
| 7. The ML engine replies with the EDA. | ML engine | EDA details | EDA | E | 1 |
| 8. The system displays the EDA details (categorical, continuous, etc.). |  | EDA details | EDA | X | 1 |
| 9. The Machine learning engineer fills in the variable of his (or her) interest. | Machine learning engineer | Variable details | Variable | E | 1 |
| 10. The Machine learning engineer presses ‘Transform’. |  |  |  |  |  |
| 11. The system requests the ML Engine to process the transformation. |  | Transformation details | Variable | X | 1 |
| 12. The ML engine replies with the transformation. | ML engine | Transformation details | Variable | E | 1 |
| 13. The system displays the transformation details. |  | Transformation details | Variable | X | 1 |
| Exceptions: |  |  |  |  |  |
| 14. The system displays an error message reporting the dataset inconsistency. |  | Messages | Message | X | 1 |
| **TOTAL** | | | | | **10** |

## List model projects

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | FU | DG | OOI | DM | CFP |
| 1. Machine learning engineer: The Machine learning engineer presses the “List all projects” link to list all projects. | Machine learning engineer | All projects | Project | E | 1 |
| 2. System: The system searches for all projects. |  | Project details | Project | R | 1 |
| 3. System: The system displays the project list. |  | Project details | Project | X | 1 |
| 4. Machine learning engineer: The Machine learning engineer navigates the results. |  |  |  |  |  |
| Exceptions: |  |  |  |  |  |
| 5. System: The system displays an error message stating that no data have been retrieved. |  | Messages | Message | X | 1 |
| TOTAL | | | | | **4** |

## Open project

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | FU | DG | OOI | DM | CFP |
| 1. The Machine learning engineer clicks “Open” relating to the project of his (or) interest. | Machine learning engineer | Project ID | Project | E | 1 |
| 2. The system searches the project details. |  | Project details | Project | R | 1 |
| 3. The system displays the project details. |  | Project details | Project | X | 1 |
| 4. The Machine learning engineer views the project details. |  |  |  |  |  |
| Exceptions |  |  |  |  |  |
| 5. The system displays an error message. |  | Messages | Message | X | 1 |
| TOTAL | | | | | **4** |

## Univariate and bivariate analysis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | FU | DG | OOI | DM | CFP |
| 1. The Machine learning engineer clicks “Analyze” relating to the variables to analyze. | Machine learning engineer | Variables | Variable | E | 1 |
| 2. The system requests the ML Engine to process the analysis. |  | Analysis request | Variable | X | 1 |
| 3. The ML engine replies with the analysis details. | ML Engine | Analysis details | Analysis | E | 1 |
| 4. The system displays the analysis details. |  | Analysis details | Analysis | X | 1 |
| 5. The Machine learning engineer fills in the variables of his (or her) interest. | Machine learning engineer | Variables | Variable | E | 1 |
| 6. The Machine learning engineer presses ‘T-test’. |  |  |  |  |  |
| 7. The system requests the ML Engine component to process the t-test. |  | T-test request | Variable | X | 1 |
| 8. The ML engine replies with the t-test details. | ML engine | T-test details | T-test | E | 1 |
| 9. The system displays the t-test details. |  | T-test details | T-test | X | 1 |
| TOTAL | | | | | **8** |

## Comparison of categorical features

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | FU | DG | OOI | DM | CFP |
| 1. The Machine learning engineer clicks “Compare categorical features” relating the categorical variables. | Machine learning engineer | Categorical features | Feature | E | 1 |
| 2. The system requests the ML Engine to process the features comparison. |  | Comparison request | Feature | X | 1 |
| 3. The ML engine replies with the features comparison. | ML engine | Comparison results | Feature | E | 1 |
| 4. The system displays the features comparison details. |  | Comparison results | Feature | X | 1 |
| TOTAL | | | | | **4** |

## Bivariate analysis of continuous features

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | FU | DG | OOI | DM | CFP |
| 1. The Machine learning engineer clicks “Continuous feature analysis” relating the continuous features of his interest. | Machine learning engineer | Continuous features | Feature | E | 1 |
| 2. The system sends the features to analyze. |  | Continuous feature analysis request | Feature | X | 1 |
| 3. The ML engine replies with the analysis. | ML Engine | Analysis results | Analysis | E | 1 |
| 4. The system displays the analysis. |  | Analysis results | Analysis | X | 1 |
| TOTAL | | | | | **4** |

## Multivariate analysis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | FU | DG | OOI | DM | CFP |
| 1. The Machine learning engineer presses “Multivariate analysis”. |  |  |  |  |  |
| 2. The system displays a form to select the features. |  |  |  |  |  |
| 3. The Machine learning engineer selects the features of his (or her) interest. | Machine learning engineer | Features | Feature | E | 1 |
| 4. The Machine learning engineer presses “Execute”. |  |  |  |  |  |
| 5. The system requests the ML Engine to process the Multivariate analysis of the selected features. |  | Features | Feature | X | 1 |
| 6. The ML engine replies with the multivariate analysis. | ML engine | Analysis | Analysis | E | 1 |
| 7. The system displays the analysis. |  | Analysis | Analysis | X | 1 |
| TOTAL | | | | | **4** |

## SVC model training

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | FU | DG | OOI | DM | CFP |
| 1. The Machine learning engineer clicks “Train” link. |  |  |  |  |  |
| 2. The system shows a form containing the following fields: training parameters, dataset. |  |  |  |  |  |
| 3. The Machine learning engineer fills in the model parameters. | Machine learning engineer | Model parameters | Model | E | 1 |
| 4. The Machine learning engineer presses “Execute”. |  |  |  |  |  |
| 5. The system requests the ML Engine to train the model. |  | Model parameters | Model | X | 1 |
| 6. The ML engine replies with the training results. | ML Engine | Training results | Model | E | 1 |
| 7. The system updates the project setting the new model status. |  | Model status | Model | W | 1 |
| 8. The system displays the training details (precision, recall and F1-score). |  | Training details | Model | X | 1 |
| TOTAL | | | | | **5** |

## Model test

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | FU | DG | OOI | DM | CFP |
| 1. The Machine learning engineer clicks the “Test” link. |  |  |  |  |  |
| 2. The system displays a form containing the training parameters and dataset fields. |  |  |  |  |  |
| 3. The Machine learning engineer fills in the parameters for the model. | Machine learning engineer | Model parameters | Model | E | 1 |
| 4. The Machine learning engineer presses “Execute”. |  |  |  |  |  |
| 5. The system requests the ML Engine to test the SVC model. |  | Model parameters | Model | X | 1 |
| 6. The ML engine replies with the test results for the model. | ML engine | Test results | Model | E | 1 |
| 7. The system displays the test details (precision, recall and F1-score) for the model. |  | Test results | Model | X | 1 |
| TOTAL | | | | | **4** |

## Patient classification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | FU | DG | OOI | DM | CFP |
| 1. The doctor clicks “Diagnosis” relating to the patient of his (or her) interest. | Doctor | Patient ID | Patient | E | 1 |
| 2. The system searches for the SVC model parameters. |  | SVC model parameters | Model | R | 1 |
| 3. The system requests the ML Engine to classify the patient. |  | Patient data | Patient | X | 1 |
|  |  | SVC model parameters | Model | X | 1 |
| 4. The ML engine replies with the patient classification. | ML Engine | Patient classification | Patient | E | 1 |
| 5. The system displays the patient classification. |  | Patient classification | Patient | X | 1 |
| TOTAL | | | | | **6** |

Legend:

N/A = Not applicable

AC = Already counted

# Measurement Summary

|  |  |  |
| --- | --- | --- |
| N. | Functional Process | CFP |
| 1 | Loading of clinical data | 3 |
| 2 | List model projects | 10 |
| 3 | Open project | 4 |
| 4 | Univariate and bivariate analysis | 8 |
| 5 | Comparison of categorical features | 4 |
| 6 | Bivariate analysis of continuous features | 4 |
| 7 | Multivariate analysis | 4 |
| 8 | SVC model training | 5 |
| 9 | Model test | 4 |
| 10 | Patient classification | 6 |
| TOTAL | | **52** |