

# Deliverable **D2**

## Deadline: 2016-04-28 (extended)

**Commented [FC1]:** The contents of this document must be uploaded on the project website, in a proper location (according to the website structure)

### 1 Purpose and scope

Define what is the goal of the system, what problems tries to solve. Be clear and explicit about the boundaries of the system (what is IN, what is OUT).

This information is similar to the Vision, but here it should be more concise, formal and precise. The intended readers, in this case, are system designers, or other engineers or managers, not the end users.

Length: max 2-3 paragraphs

### 2 Definitions

#### 2.1 Glossary

Define the terms, the conventions, the concepts that you will use in your system specification. In this section you define the *meaning* of the words that you will use later in the requirements, in order to wipe out any ambiguity.

For example, “the interface” is a very general term, but in the glossary you may define “Interface = web application used to access the schedule from a PC or smartphone”.

#### 2.2 Actors

In particular, you must define the ACTORS of the system, i.e., all the types of users that will directly interact with the system. Be specific (i.e., teacher giving a class, or student seeking a study room, not generically ‘teacher’ or ‘student’).

### 3 System Requirements

#### 3.1 Functional Requirements

You may find useful to group the requirements according to “Functional Areas”, i.e., related groups of functionalities that are related to a portion of the system features.

For example:

Functional Area	Description
1	User login, registration and logout
2	Public web pages visible by any user (even not registered users)
3	Notifications sent by the system to user devices
4	Interface for setting user preferences
5	... etc etc ...

**Commented [FC2]:** Define a short identifier for each area. E.g., a number (1, 2, 3, 4, ...) or a short string (LOG, WEB, NOTIF, Prefs, ...)

You may then group and list functional requisites according to their functional area, by using a numeric code X.Y (X=area, Y=requisite within the area).

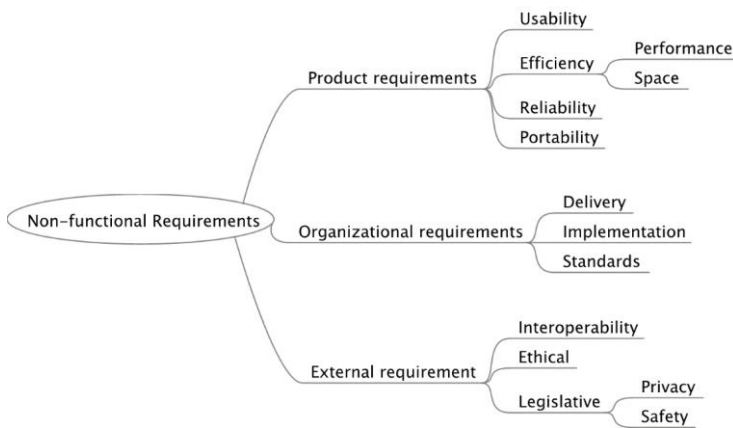
The template for each requisite should contain the following information:

- FR X.Y: Title
  - Description (verifiable)
  - Priority: 1--5

### 3.2 Non-Functional Requirements

- NFR Z: Title
  - Description
  - Area (select one item from the picture below)

The most important areas to consider in our prototypes/projects are: **Portability** (list all the compatible devices), **Interoperability** (which external systems are involved), **Efficiency** (mention the expected response times). Also, include any Legislative requirements, if you have any (e.g., for medical interfacing devices).



## 4 Open issues

Continuation of the open issues in D1.

**Commented [FC3]:** X: Use the same number or string as the Functional Area

**Commented [FC4]:** Y: Progressive number of the requisite inside the functional area

**Commented [FC5]:** Title: Short definition (e.g., "Edit preferences")

**Commented [FC6]:** Description (2-6 lines) of the **functionality** that must be achieved. Write this description in a way that allows and facilitates the **verification** of the actual and correct implementation.

**Commented [FC7]:** Priority: when such a requirement must be implemented. 1=in the first version, 2...5=in later interactions of the development. Priority is a combination of **Necessity** (it's nothing special, but needs to be there for the system to work) and **Relevance** (it's the key aspect specific to our project). If you want, you may also list separately the Necessity and Relevance (in a scale 1..5) to help yourself in defining the priority.

**Commented [FC8]:** Number your NFR.

**Commented [FC9]:** Try to give quantitative and verifiable descriptions

**Commented [FC10]:** Update the list of Open Issues, with respect to what was presented in D1. You may mark some (older) issues as "solved" (e.g., by ~~erasing~~ them), and some as "new". Try also to outline which are the **major** issues.

This should be a list (not a prose), to be able to check and modify it easily.

The open issues are the problems whose solution is not clear at the moment (they are not a "things to do" list, since most of the things to do are well-known, you only need time and effort).