

Deliverable D2 – Checklist

The Project website must contain, the following information. Feel free to insert them according to your website template. There are no specific requirements about how you structure the website, as long as the following information is present and clearly identifiable.

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 (to enable understanding of what is **included in the scope** of the project, what is **out of scope**).

Note: 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

Project Features

Make a list of all project features. Generally, you should have around 10-15 features in total. Revise slides 43-44 of the presentation “Aml Design process” for the definition.

Each feature must be identified by:

- A feature statement (something you’d print on a detailed datasheet, in a language intended to be shared with your customers/users)
- The **priority** of the features (priority 1 elements represent the core functionality of the project; priority 2-3 features may be suitable for future extensions). Typically, at the exam you will present a project consisting of priority 1 features, and some priority 2 ones.
- Features statements should be grouped into *functional areas*, to make them clearer.

System Architecture

List the main system components, what is their nature, and what kind of information they exchange with the environment, the user, and other components.

You should identify:

- Which **computational nodes** you have. Computational nodes are hardware components that must run some software developed by you. Computational nodes may be “central” (e.g., a server on a remote computer), “distributed” (e.g., a raspberry interacting with devices, or an Arduino managing some sensor), or “mobile” (e.g., smartphones or other “user” devices).
- Which sensors/actuators are used in the system (explain how where they are placed and how they will be interconnected)
- Which user interfaces need to be developed (on what devices, web vs. mobile, who are the target users).

Usually, the best way to give this information is to sketch a **diagram of the various system components**, with their main interconnections. In this diagram, the exact type and model of device is **not needed**. In the next sections (hardware, software and network architecture) the characteristic and the function of each element in the diagram will be explained and detailed. Be consistent with the names (consistency across the names in the sketch, those in the description, and those used in Deliverable D2).

In the next sections, also try to map the various system functionalities (from Functional Requirements in D2) to the actual HW components (what component implements what functions) and SW components (what applications / server / interface / ... implements what functions).

Hardware Architecture

Make a list of the required:

- Computational nodes: role in the architecture, devices managed or connected by the node.
- Devices (sensors/actuators): identify types, function, location, but not yet brand & model
- User interface devices: identify type, function, location

Give information about each of these components. Most of these should appear in the general system architecture sketch. Think of this section as a more detailed “tabular” representation of the hardware components appearing on the System architecture diagram, with the opportunity of writing 1-2 sentence explaining the role of each hw component.

Software Architecture

Make a list of the major software architectural modules. For example, web applications, mobile applications, gateways, computational nodes (e.g., computer vision, or signal processing, ...), etc.

For each software component, list:

- what functions it implements (mapped to a subset of functional requirements)
- where are they running (deployment), taken from the “computational nodes” section above
- how they interact (which APIs they offer, which APIs they call).

They may be existing (3rd party) components, or new software to be developed (in your group). You may also include external ‘cloud’ services.

Network Architecture

Detail how the different devices and computational nodes are connected to each other. There will be a mix of connection technologies (WiFi, 3G, Ethernet, Bluetooth, Z-Wave, ...): each device will be connected to one or more of these networks.

Also list any possible addressing requirements (e.g., the need of a “public” IP address, the need of two or more devices to be on the same WiFi, the need of using some specific network ports, etc...).

Selected components

Identify the actual products to populate the chosen architecture description.

This should be a “short” and “dry” section with the list of needed material. Please mark what material can be used from the LADISPE, and what material needs to be procured somehow (to be discussed in person).

This information, at this stage in the project, is very likely partial, incomplete, with details yet to be defined. It will be refined during the future project development. Try to write, especially, the information that is already available, and the information that still needs to be decided, so that it may be discussed in the supervised labs.

Hardware Components

The list (bill-of-material) of required devices. This section should be short and usually may be expressed with one table for OTS and one table for Ad-Hoc.

Off-the-Shelf (OTS)

List of existing OTS components. Include both computational nodes (e.g. Raspberries) and sensors/actuators and user devices. The list should “cover” all the requirements of the System Architecture.

If your system cannot be completely built (because of its size, for example), and you are planning to run a scaled-down demo, then list only the components needed for the demo.

AD-HOC

If you need to build (design and construct) some hardware devices (e.g., special sensors), then list them here, with a short description of their function and the technologies you will use for building them.

Software Components

The adopted libraries, frameworks and cloud services that you will use (refer to specific products and versions, they should match the components described above).

Open issues

Continuation of the open issues in D1.

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 ~~crossing~~ 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).