ÉCOLE POLYTECH'NICE SOPHIA
SI5/M2 INFO
2022-2023

Julien Deantoni

Architecting IoT Systems, Beyond Functional Correctness

Two-Factor Identification Electronic Lock

⚠ May be refined in the next few weeks

Project delivery

This project concerns the delivery of the *Architecting IoT Systems, Beyond Functional Correctness* course. It will include the implementation of a two factor identification electronic lock for a house entrance (either a portal or a door), based on facial recognition and RFID badge. Of course, beyond functional correctness, you will be able to identify threats and mitigation of your solution. Note that there is no a priori extra-functional requirements for this project so that you will have to elicit them by yourself.
Deliveries are expected by email (to Julien Deantoni: firstname.lastname@univ-cotedazur.fr, with [IoT_BFC] as object prefix) followed by "team X project" where X is the name of your team (as used in the slack dedicated channel). The delivery is expected **before** the 23^{rd} of January 2023 at 10:00PM Paris Time. The delivery is expected as a PDF paper That follows classical scientific papers format.
The paper must contain :

- the name of the members of your team
- a link to the code of your DSL (typically a link to the git repository. Note that this git should clearly explain how to setup and use the project)
- an introduction section specifying the functional requirements and what are the extra functional requirements you elicited as being the most important; together with an explanation of why (to be further detailed in the "proposed solution" section).
- a critical description of existing solution ("state of the practice" section); with the pros and cons of each identified solution;
- a "proposed solution" section specifying:
  - the differentiating extra functional requirements you elicited ;
  - the main risks you identified and how you mitigated them. It can be done by rationalizing the choices you did in your architecture. note that the architecture should be specified and rationalized in terms of:
    - the application architecture;
    - the hardware architecture;
    - the placement specification;
  - a critical analysis of your own solution, specifying what you did right and what could be improved (and how)

- An “implementation and result” section highlighting how and why your solution is actually a good one (or not)
- A conclusion resuming the main pros and cons of your architecture, the responsibility of each member in the team with respect to the delivered project; as well as prospective on potential evolution.

Objectives: Two-factor identification electronic lock

We want to propose an electronic lock system where the nominal behavior is that the users are identified by both a facial recognition and by a RFID badge. When both identification methods are successful, the entrance is unlock. It should be possible to register faces and badges. All other aspects are left to your team. You can add features. You will add requirements. Note that targeting specific users and usages is a way to rationalize the other requirements. You have to respect the time to market (see the project delivery section)
Technical choices in terms of languages, libraries, frameworks or technologies are not imposed and you are free to choose the one(s) that seem(s) the most suitable to your team.
As available hardware, you’ll have access to:

- Raspberry PI 3
- Arduino boards with a shield with classical sensors/actuators (leds, buttons, temperature sensors, ...)
- raspberry hat for arduino sensor usage
- a pi camera module.
- a `SLC3711` USB RFID reader
- a USB ampere-meter (to be shared among teams)
- some radio frequency emitter/receivers
- a maximum of two laptops in your teams to be used as you feel is appropriate

You will not have access to an actual lock so the lock/unlock of the entrance will be shown by using two leds (typically a red and a green).
You can mock extra hardware that you main need by using simple Sensors/Actuators from Arduino.

Important note

Quality of your code is of course important in general but will not be taken into account for this project
Usability in term of graphical design and or physical ergonomics is of course important but will not be taken into account for this project
Told differently, the following extra functional properties will not be considered in this project: maintainability of your code, correct versioning of your code, genericity of your code, UI design and ergonomic aspects[1]

[1] I really like qualitative code but my feeling is that this is too much demanding for a 8 weeks project