UNIVERSITÉ CÔTE D'AZUR
SI5/M2 INFO
2023-2024

Julien Deantoni, Nicolas Ferry

Architecting IoT Systems, Beyond Functional Correctness

Automated Car Entrance System

⚠ 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 entails the development of a comprehensive system designed to automate the management of vehicular entrances. Teams will be responsible for delineating the specific operational environment for the system. In addition to ensuring functional correctness, participants are expected to discern potential threats (to any extra-functional concerns - eg. performances, scalability, energy) and proactively devise mitigation strategies for their solutions. It is important to note that no a priori extra-functional requirements have been specified for this project, necessitating teams to independently elicit and address them.
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 21^{nd} of January 2024 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 system (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 context of use of the system, its 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:
    1. the application architecture;
    2. the hardware architecture;
    3. the deployment 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: Automated Car Entrance System

We aim to present a solution for automated vehicular access control (and possibly guidance), characterized by a seamless process that eliminates the need for users to explicitly request entry. The system should operate autonomously, allowing access as necessary. The functionality should encompass the ability to register new vehicles while also providing the option to restrict access for specific vehicles. Your team is granted the flexibility to introduce additional features and specifications. It is imperative to incorporate any essential requirements and consider targeting specific user demographics and use cases to optimize overall system efficacy. Furthermore, adherence to the specified time-to-market constraints, as outlined in the project delivery section, is of utmost importance.
Obviously, your solution should go beyond classical access control solutions typically available nowadays for collective or private car entrance.
It is mandatory to validate the operational environment and the solution envisioned with one of the school representatives.
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 and 4
- Arduino boards with a shield with classical sensors/actuators (leds, buttons, temperature sensors, bluetooth, RFID, NFC, infra red, ...)
- raspberry hat for arduino sensor usage
- a pi camera module.
- 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

Regrettably, access to an authentic vehicle or a physical garage for testing purposes is unavailable. As a result, the demonstration of entrance management will be executed through alternative methodologies, such as mocking, simulating, or presenting printed images to a camera.
To replicate any additional required hardware, you may employ basic sensors and actuators from Arduino as part of the mock setup.

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