Project: Occupancy Control

This project aims to create a Single Page Application (SPA) using the knowledge and resources acquired during this class. The SPA to be built is a system that can control the number of people in a space, to enforce the rules defined by the Health Authorities to fight the COVID-19.

1. Relevant Dates

Publication of this document:	27/10/2020
Submission of the implementation:	12/12/2020
Oral presentation:	15/12/2020
Publication of results:	15/01/2020

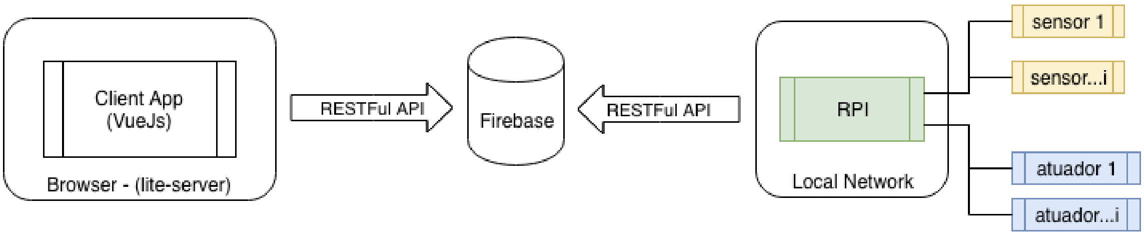
2. Scenario

The COVID-19 pandemic has forced people to change their behaviour and resulted in severe restrictions in the occupancy of spaces. The number of people inside a closed space and how people move need to be monitored and controlled.

The system to be implemented allows to **count each person** that enters and exits a space, the number of people disinfecting their hands and controls a red light **sign to stop** people from entering the space.

3. Architecture

The "Occupancy Control" app should adopt the SPA (Single Page Application) paradigm. For evaluation purposes, the SPA application can be executed locally using a (local) web server such as **http-server** and **lite-server**. The architecture of the system is illustrated next.



The architecture has 3 main components:

- Firebase service: this backend-as-a-service is where all information is persisted;
- RPI: a local device with a service that runs in a local network and has access to the internet (to be able to connect to Firebase). This device connects to sensors (e.g. IR cells, touch button on the disinfectant dispenser) and actuators (e.g. red light, pump of the disinfecting dispenser);
- The client app: the SPA application that allows to view all sensors and allows the user to define the maximum number of people allowed in the space.

Please note that the RPI device and the SPA application to not connect directly to each other.

4. Functional Requirements

The SPA application should implement the following requirements:

- A dashboard with counters for at least two zones. A zone is an aggregation of one entrance, one exit door and a disinfectant dispenser button;
- Show the current and allowed number of people in the space (it can be included in the dashboard);
- A settings screen that allows to:
  - set the maximum number of people allowed in the space;
  - set the periodicity for the reporting the counters (the same for all doors)
  - set the periodicity for the stop light (one periodicity for the
  - enable or disable a zone (entrance, exit and dispenser are all part of a zone)
- Show entrance and exit reports per hour or per day.

The application on the RPI device should implement the following requirements:

- To send periodically the number of people that entered and exited each door. The system should support at least two entrance doors, two exit doors and two disinfecting buttons (one per door) and two disinfecting pumps (controlled locally by the corresponding dispenser button);
- Get periodically the number of people in the space and the maximum number of people allowed in the space. The stop (red) light should be lit when the number of people is equal or above the maximum number of people.

For the sake of simplicity, the client SPA is not required to have any authentication.

5. Submission

The work submission is done using this class's space on the moodle platform. The submission should be a single zip file with the project report and three folders/directories:

- Firebase: the JSON file to import to the Firebase service;
- App: the client SPA source code;
- RPI: the source code for the RPI local device.
- Report file: file in PDF format (font size of 12 points with a maximum 10 pages) with an introduction to the implemented features (it should **not** include source code). It should include an "Auto evaluation" section.

6. Evaluation

The project can be implemented by groups of up to 3 elements.

The grade criteria are as follow:

- SPA client: 50%
- RPI application: 40%
- Report: 10%