

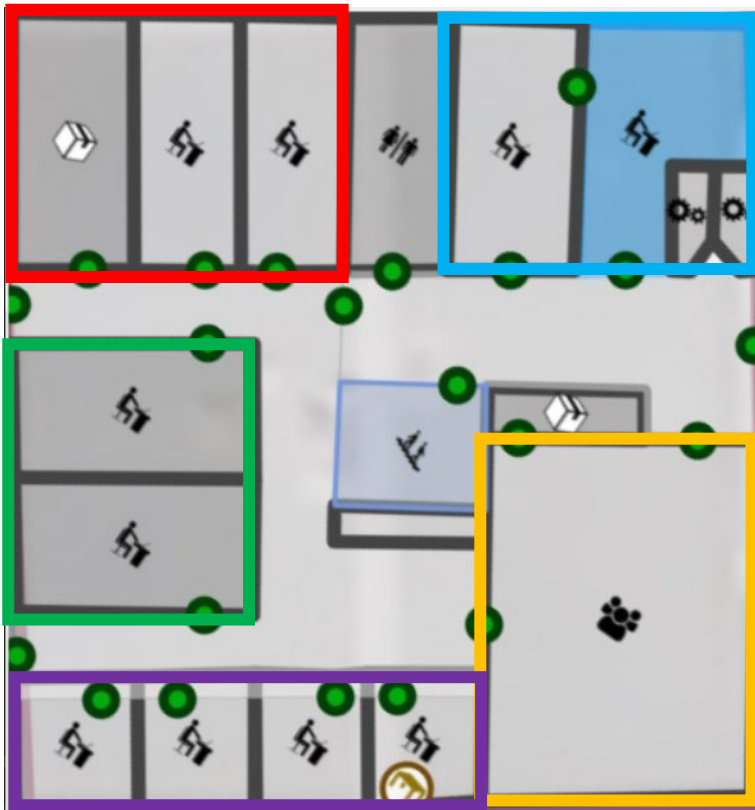
# Rapport choix développeurs

## Pourquoi API Feet ?

L'application API Feet, qui tient son nom d'un célèbre film d'animation, vient en aide à l'administration de l'UPSSITECH, une école d'ingénieur située sur le campus de l'université Paul Sabatier à Toulouse.

En effet celle-ci leur permet de connaître, à l'aide de l'utilisation d'API REST et de capteur Arduino, différentes données tels que la température, la pression et d'autres données au sein des salles de l'administration, pour permettre de mieux réguler la consommation électrique, le chauffage en hiver notamment et ainsi faire des économies d'énergie, ou bien pouvoir connaître les allers et venues des personnes pour, par exemple, être capable de pouvoir débloquent d'autres salles.

Pour cela nous avons divisé l'administration en 5 agrégats, on appellera agrégat un regroupement de plusieurs salles.



Ces 5 agrégats sont les suivants :

- « **Le nord** » qui représente les salles de l'administration le plus en haut sur le plan
- « **La terre du milieu** » qui représente les salles qui sont au milieu du plan
- « **Le sud** » qui représente les salles les plus en bas
- « **Clément adhère** » qui représente la salle de conférence
- « **BDE** » qui représente le bureau des élèves et le foyer

Pour les besoins de la simulation, nous faisons appel à une API OpenWeather, pour nous donner la météo sur les 4 premiers agrégats. Les données météorologiques sont définies comme suit :

- « Le nord » correspond à la ville de Nuuk, capitale du Groenland,
- « La terre du milieu » correspond à la ville de Moscou, capitale de Russie,
- « Le sud » correspond à la ville de Casablanca au Maroc,
- « Clément adhère » correspond à la ville de Antananarivo, capitale de Madagascar.

Pour l'agrégat du BDE il s'agit des valeurs renvoyées par les capteurs de la carte Arduino. Celles-ci sont récupérées grâce à un thread pour permettre au programme principal qui affiche l'application de continuer de s'exécuter et mettre à jour les informations.

Lorsque nous affichons un autre agrégat le thread se met en pause et redémarre lorsque nous remettons l'agrégat BDE.

Il est important de lire la notice d'utilisation fournie en parallèle de ce rapport pour savoir comment démarre et comment s'arrête la communication entre l'Arduino et l'application.

## Partie Arduino

Pour Arduino nous disposons d'un capteur de température/humidité, et de deux boutons poussoirs, nous avons donc écrit des programmes Arduino gérant ces entrées, puis écrit un script python permettant de récupérer les informations communiquées par la carte sur le port série afin de les afficher dûment.

### Arduino

Pour commencer, sur le programme Arduino, nous avons deux grandes dépendances à séparer, le traitement du capteur et le traitement des deux boutons.

- Pour le capteur, nous avons fait en sorte de récupérer les valeurs envoyées toutes les 500ms. Nous affichons ensuite ces informations sur le port série.

Pour notre application, le but est de s'imaginer le nombre total de personnes rentrées dans le BDE sur une heure, afin d'avoir une idée de la surcharge de la salle, pour éventuellement débloquent d'autres salles aux moments pleins (par exemple en période de révision ou lors du repas de midi), ou encore réduire le chauffage pour faire des économies d'énergie. Pour cela nous établissons une moyenne sur l'heure passé de la température et du nombre de personnes présentes dans le BDE.

- Pour simuler le nombre de personnes présentes dans la salle on utilise des boutons poussoirs, nous récupérons la valeur du port correspondant, et nous en servons pour décrémenter ou incrémenter un compteur de personnes dans la pièce.

NB : Une limite de présence est fixée, on considère que l'on ne peut pas dépasser 50 personnes à la fois dans le BDE.

Nous affichons aussi un compteur du nombre total de personnes rentrées dans la salle, qui s'incrémente pendant 20 secondes, avant de se réinitialiser à 0, il s'agit du nombre total d'élèves rentrés dans le BDE pendant l'heure.

### Python

Pour la partie Python, en utilisant la librairie serial nous lisons la valeur du port série actif. Le baud rate quant à lui est fixé à 9600.

Ce qui est récupéré par ce script est une chaîne de caractères comprenant toutes les informations, au début il était prévu de ne récupérer uniquement que les valeurs de chaque composant et de les renvoyer, mais on récupère finalement la chaîne de caractère envoyée par l'Arduino, pour l'afficher sur notre application.

## Interface de l'application

L'application possède une interface entièrement faite sous python et dont l'utilisation est décrite dans le manuel d'utilisation.

L'utilisation d'une carte, de l'afficher et de pouvoir visualiser, sur celle-ci, les agrégats sélectionnés est un point important pour ce projet, car il faut être en mesure de pouvoir se positionner et de permettre à l'utilisateur de voir à quoi correspondent les valeurs obtenues à travers les APIs.

Une petite surprise vous attend aussi dans le menu aide de l'application. 😊

*Les développeurs*