Documentation for the IoT Project

Name: Börzsei Marcell László

Neptun-Code: BQHFLX

Description:

The project creates a simulation for a real life IoT device that can control different smart devices such as a smart light, a thermostat, or a security camera. It is also has automation rules, like it turns on the light if it detects motion.

Class SmartLight:

Functions:

* \_\_init\_\_: Initializes the starting values: deviceId, status and brightness.
* turn\_on: It changes the object’s status to True.
* turn\_off: It changes the object’s status to False.
* get\_status: Getter function for status attribute.
* get\_device\_id: Getter function for deviceId attribute.
* get\_brightness: Getter function for brightness attribute.
* toggle: It turns on the light if it’s off, otherwise it turns it on.
* set\_brightness: Setter function for brightness attribute.
* gradual\_dimming: It changes the brightness slowly not instantly.
* simulate\_random\_changes: It simulates random changes to show how the light works.

Class Thermostat:

Functions:

* \_\_init\_\_: Initializes the starting values: deviceId, status, temperature, minTempreature and maxTemperature.
* turn\_on: It changes the object’s status to True.
* turn\_off: It changes the object’s status to False.
* get\_status: Getter function for status attribute.
* get\_device\_id: Getter function for deviceId attribute.
* get\_temperature: Getter function for temperature attribute.
* get\_min\_temperature: Getter function for maxTemperature attribute.
* get\_max\_temperature: Getter function for minTemperature attribute.
* toggle: It turns on the thermostat if it’s off, otherwise it turns it on.
* temp\_range: It changes the range of the temperature scale.
* set\_temperature: Setter function for temperature attribute.
* simulate\_random\_changes: It simulates random changes to show how the light works.

Class SecurityCamera:

Functions:

* \_\_init\_\_: Initializes the starting values: deviceId, status and lightReference (SmartLight object).
* turn\_on: It changes the object’s status to True.
* turn\_off: It changes the object’s status to False.
* turnOnSecurityStatus: It changes the securityStatus to True.
* turnOffSecurityStatus: It changes the securityStatus to False.
* get\_status: Getter function for status attribute.
* get\_device\_id: Getter function for deviceId attribute.
* get\_security\_status: Getter function for securityStatus attribute.
* toggle: It turns on the light if it’s off, otherwise it turns it on.
* simulate\_random\_motion: It simultates a random motion at a 50% chance and if the camera detects random motion, it turns the light on and sets its brightness to 100.

Class AutomationSystem:

Functions:

* \_\_init\_\_: Initializes the starting values: devices list.
* add\_device: it gives a new device to the devices list attribute.
* getDevices: Getter function for devices attribute.
* run\_life\_simulation: Runs real life simulations.

main.py:

It creates a window, an instance for the monitoring dashboard and starts the mainloop.

MonitoringDashboard:

This file creates the GUI for the IoT simulator it contains in itself:

* Functions for writing out the changes of values (brightness, temperature), and changes of statuses of the devices.
* It creates buttons, so that we can manage, change and set the value of the devices’ attributes, and so that we can detect random motion.
* It creates two scrollbars for changing the brightness and temperature dynamically.
* It has a function (that implements the automation rule of turning the light on if there is motion detected), that checks random motion at every then second and if it could detect motion, it turns the light on and sets the brightness to 100, and it runs on a different thread from the GUI’s thread.
* It creates a board that every change that has been made to the devices’ values.
* Creates one of each device (SmartLight, Thermostat, SecurityCamera), and gives them to an AutomationSystem.