Home automation system report

A report explaining the classes, their attributes, methods, and relationships in the provided home automation system code:

Classes:

1. Device

The Device class is the base class for all devices in the home automation system. It defines the common properties and methods shared by all devices.

Attributes:

- id : Unique identifier for the device

- status : Current status of the device (ON or OFF)

- type : Type of the device (e.g., Light, Thermostat, Camera)

- location : Location of the device

Methods:

- toggle\_status() : Toggles the status of the device between ON and OFF

- update\_settings(settings) : Updates the device settings based on the provided settings dictionary (implemented in subclasses)

1. Light

The Light class inherits from the Device class and represents a light device in the home automation system.

Attributes:

- brightness : Current brightness level of the light (0-100)

Methods:

- adjust\_brightness(level) : Adjusts the brightness of the light to the specified level

- update\_settings(settings) : Updates the light settings based on the provided settings dictionary (checks and sets the brightness level)

1. Thermostat

The Thermostat class inherits from the Device class and represents a thermostat device in the home automation system.

Attributes:

- temperature : Current temperature set on the thermostat

Methods:

- set\_temperature(temp) : Sets the temperature on the thermostat

- update\_settings(settings) : Updates the thermostat settings based on the provided settings dictionary (checks and sets the temperature)

1. Camera

The Camera class inherits from the Device class and represents a camera device in the home automation system.

Attributes:

- angle : Current angle of the camera

Methods:

- adjust\_angle(new\_angle) : Adjusts the angle of the camera to the specified angle

- update\_settings(settings) : Updates the camera settings based on the provided settings dictionary (checks and sets the angle)

1. User

The User class represents a user in the home automation system.

Attributes:

- user\_id : Unique identifier for the user

- name : Name of the user

- access\_level : Access level of the user

Methods:

- authenticate(credentials) : Authenticates the user based on the provided credentials (not implemented in the code snippet)

- send\_command(command, controller) : Sends a command to the controller to be processed

1. Scheduler

The Scheduler class manages scheduled tasks in the home automation system.

Attributes:

- scheduled\_tasks : List of scheduled tasks

Methods:

- add\_task(task) : Adds a new task to the list of scheduled tasks

- remove\_task(task\_id) : Removes a task from the list of scheduled tasks based on the task ID

- `execute\_tasks()`: Executes the scheduled tasks (not implemented in the code snippet)

1. Controller

The Controller class acts as the central point of interaction in the home automation system.

Attributes:

- devices : List of devices managed by the controller

- users : List of users managed by the controller

Methods:

- add\_device(device) : Adds a new device to the list of managed devices

- remove\_device(device\_id) : Removes a device from the list of managed devices based on the device ID

- process\_command(user, command) : Processes a command sent by a user (not implemented in the code snippet)

1. Scheduled task

The scheduled task class represents a scheduled task in the home automation system.

Attributes:

- event\_time : Time when the task is scheduled to execute

- device : Device associated with the task

- action : Action to be performed on the device (e.g., 'Set Temperature', 'Set Brightness', 'Set Angle')

- settings : Settings for the action (e.g., temperature, brightness, angle)

Relationships

The classes in the home automation system have the following relationships:

- The Device class is the base class for Light, Thermostat, and Camera classes. These classes inherit from Device and add specific functionality for each type of device.

- The User class interacts with the Controller class to send commands and perform actions in the system.

- The Scheduler class manages scheduled tasks, which are represented by the scheduled task class.

- The Controller class manages the devices and users in the system. It processes commands from users and controls the devices accordingly.

Testing

The test script uses the unittest framework to define test cases for the home automation system. The TestHomeAutomationSystem class sets up test objects for each class and defines methods to test specific functionalities:

- test\_device\_toggle\_status : Verifies that the toggle\_status method correctly changes the device status.

- test\_light\_adjust\_brightness : Tests the adjust\_brightness method of the Light class.

- test\_thermostat\_set\_temperature : Checks the set\_temperature method of the Thermostat class.

- test\_camera\_adjust\_angle : Verifies the adjust\_angle method of the Camera class.

- test\_user\_authentication : Tests the authenticate method of the User class.

- test\_scheduler\_add\_remove\_task : Checks the add\_task and remove\_task methods of the Scheduler class.

- test\_controller\_add\_remove\_device : Tests the add\_device and remove\_device methods of the Controller class.

- test\_controller\_process\_command : Checks the process\_command method of the Controller class.

Conclusion

The included code and tests show how to create a simple Python home automation system. Because of the object-oriented approach's modularity and extensibility, adding new device types or changing current ones is made simpler. The test suite makes sure that each component is functioning correctly, which contributes to the integrity of the system.