



# Project - Smart Home Automation System in C++

## Project Overview:

This project involves designing and implementing a modular Smart Home Automation System in C++. The system will simulate managing various smart devices found in a modern home environment, such as lights, thermostats, and security cameras. Your goal is to build a flexible and extensible application that allows easy creation and management of these devices, supports grouping devices by rooms or zones, and enables smooth interaction between components for automated control and user commands.

You will work on different aspects of the system, focusing on solving specific problems related to object creation, component organization, and communication between devices. The project encourages careful architectural design to create a maintainable and scalable solution.

---

## Detailed Project Requirements

### 1. Device Creation and Management

- Implement a mechanism to create different types and families of smart devices dynamically.
- Ensure that the system can support multiple brands or variants of the same device types without major changes to the core system.
- Provide a centralized controller or hub responsible for managing all devices within the system.

### 2. Device Organization

- Group devices are logically based on their location or function, such as by room, floor, or specific zones within the home.
- Allow operations to be performed on both individual devices and groups of devices using the same interface.
- Integrate devices that may have different interfaces or come from external/legacy sources into your system seamlessly.
- Provide a simplified interface that clients can use to interact with the entire system without needing to understand the underlying complexities.

### 3. Interaction and Automation

- Design a system where devices can communicate changes or updates to interested components automatically.
- Implement a way to represent user or automated requests as objects that can be queued, executed, or undone.
- Allow different modes of automation logic to be applied, which can be changed or switched at runtime based on user preference or environment.
- Handle the internal states of devices effectively, ensuring devices can transition between states smoothly based on commands or external triggers.

---

## Examples

### Examples of Devices to Create and Manage

#### 1. Light

- Supports operations: turn ON, turn OFF, dim to a specific brightness level.
- Variants: LEDLight, HalogenLight, different brands or models.

#### 2. Thermostat

- Controls home temperature.
- Supports setting target temperature, switching modes (heating, cooling, off).
- Variants: SmartThermostatA, SmartThermostatB.

#### 3. Security Camera

- Supports operations: start recording, stop recording, enable night vision.
- Variants: Wired, wireless, different manufacturers.

#### 4. Door Lock

- Supports locking and unlocking operations.
- Can provide status updates (locked/unlocked).

#### 5. Motion Sensor

- Detects movement in a room or zone.
  - Sends alerts or triggers other devices.
- 

### Examples of Device Organization

- **By Room:**

- LivingRoomGroup: Contains multiple devices like LivingRoomLight1, LivingRoomLight2, LivingRoomThermostat, LivingRoomCamera.
- BedroomGroup: Contains BedroomLight, BedroomThermostat, BedroomMotionSensor.

- **By Floor:**

- FirstFloorGroup: Groups rooms on the first floor such as LivingRoomGroup, KitchenGroup.
- SecondFloorGroup: Groups bedrooms and bathrooms.

- **By Function:**

- LightingGroup: Includes all lights from different rooms.
- SecurityGroup: Includes cameras, motion sensors, and door locks.

**Operations should be possible to apply to individual devices or whole groups, e.g., turning off all lights in the living room, or activating all security devices on the second floor.**

---

## Examples of Interaction and Automation

- **Motion Detection Triggers Light**  
When a motion sensor detects movement in the hallway at night, it automatically turns ON the hallway light.
  - **Temperature Adjustment**  
When the thermostat detects that the temperature is below the target, it sends a signal to the heating system to increase heat.
  - **Security Alert**  
If the door lock is unlocked unexpectedly while the security system is armed, the security camera starts recording and sends an alert notification.
  - **Command Execution and Undo**  
A user command to turn OFF all bedroom lights can be sent as an object, executed immediately or queued, and undone if needed.
  - **Automation Modes**  
The system can switch between "Energy Saving Mode" and "Comfort Mode":
    - In Energy Saving Mode, lights dim or turn off automatically when rooms are unoccupied.
    - In Comfort Mode, lights maintain preset brightness and temperature preferences.
  - **Device State Transitions**  
A light might have states like OFF, ON, and DIMMED, and smoothly transition between these based on commands or automation logic.
- 

## Deliverables

1. Source Code zipped
2. Running Executable
3. Document with your design (The used design pattern and its mapped UML Diagram)
4. Video Illustrate that document

Any missing requirement of the deliverables, your project will be considered rejected

**Thank You**  
**Edges For Training Team**