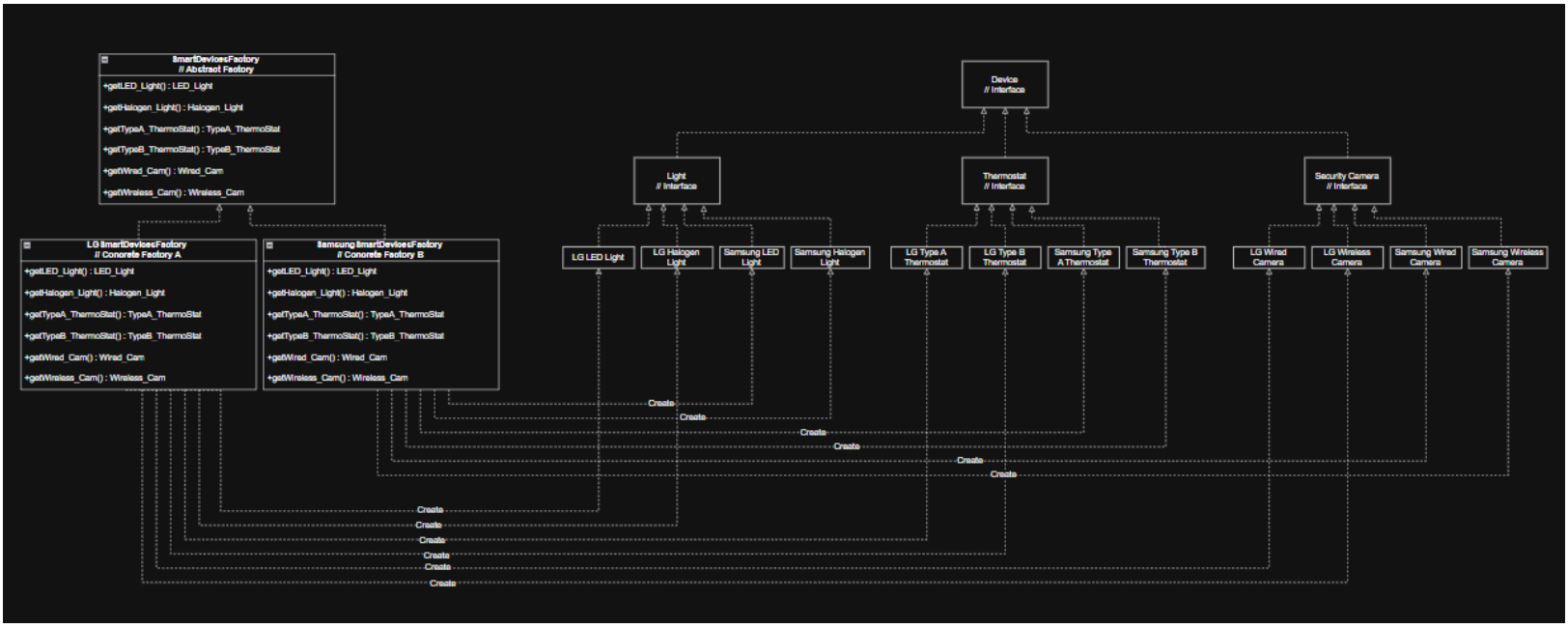
1. Creational Design Patterns Used

a) Abstract Factory Design Pattern

- Used for creating objects with several types & families such as light, thermostat & security cameras.
- Mapped UML Diagram:

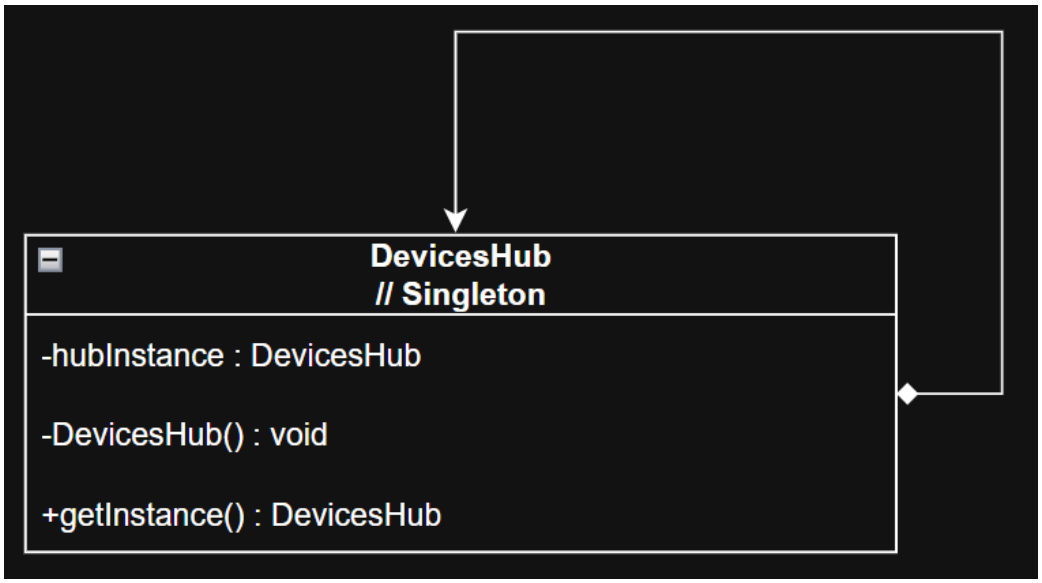


b) Factory Method Design Pattern

- Used for creating object with single type and family such as door lock & motion sensor
- Mapped UML Diagram:

c) Singleton

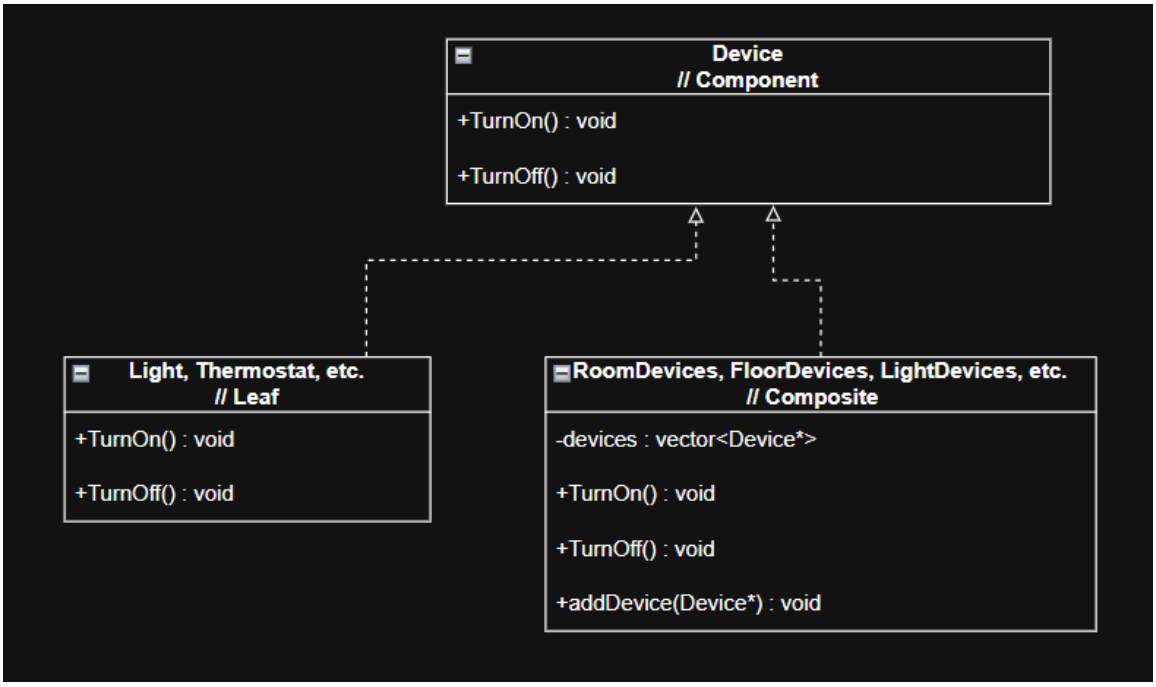
- Used for creating a single centralized controller or hub responsible for managing all devices within the system
- Mapped UML Diagram:



2. Structural Design Patterns Used

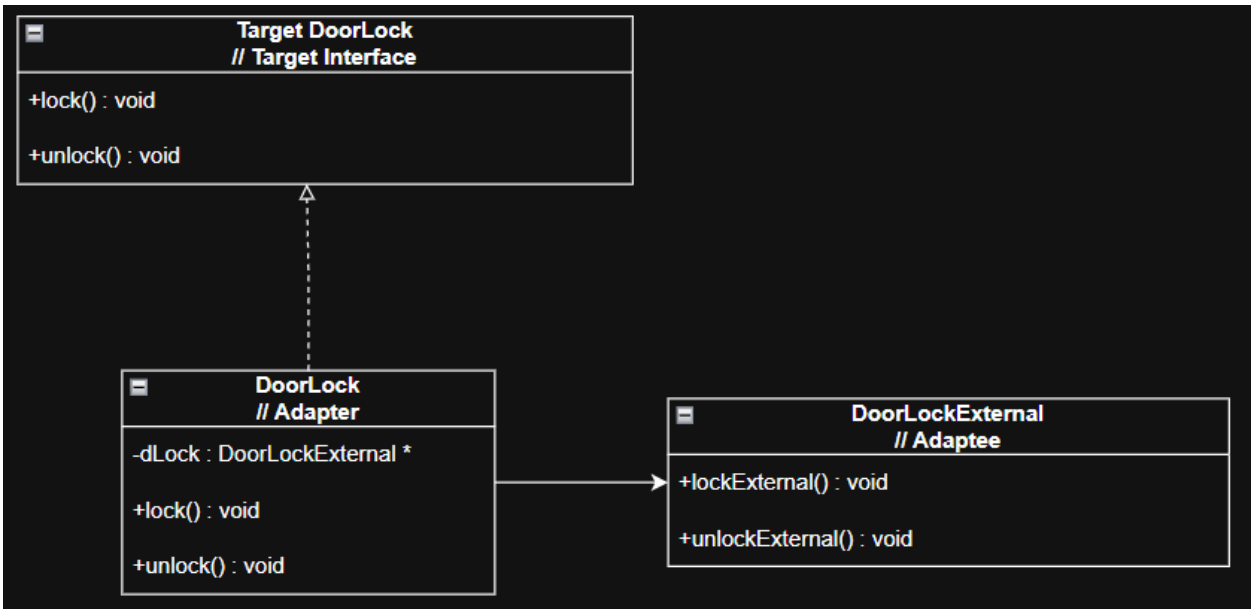
a) Composite Design Pattern

- Used for grouping devices into logical groups while maintaining that operations can be done on both individual devices & groups.
- Mapped UML Diagram:



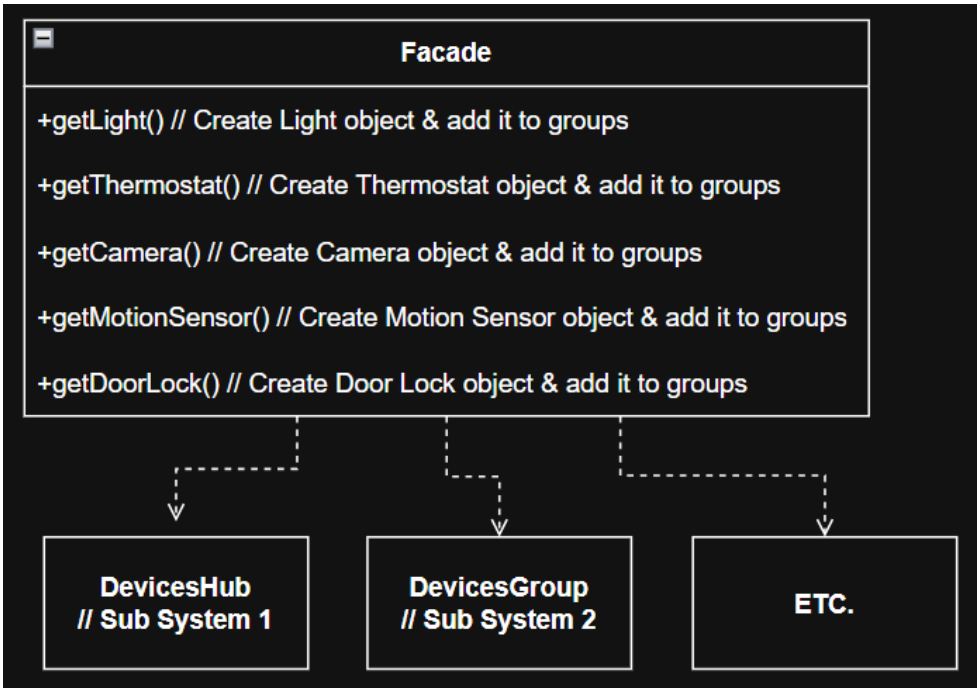
b) Adapter Design Pattern

- Used for integrating devices that may have different interfaces or come from external/legacy sources such as DoorLockExternal class
- Mapped UML Diagram:



c) Facade Design Pattern

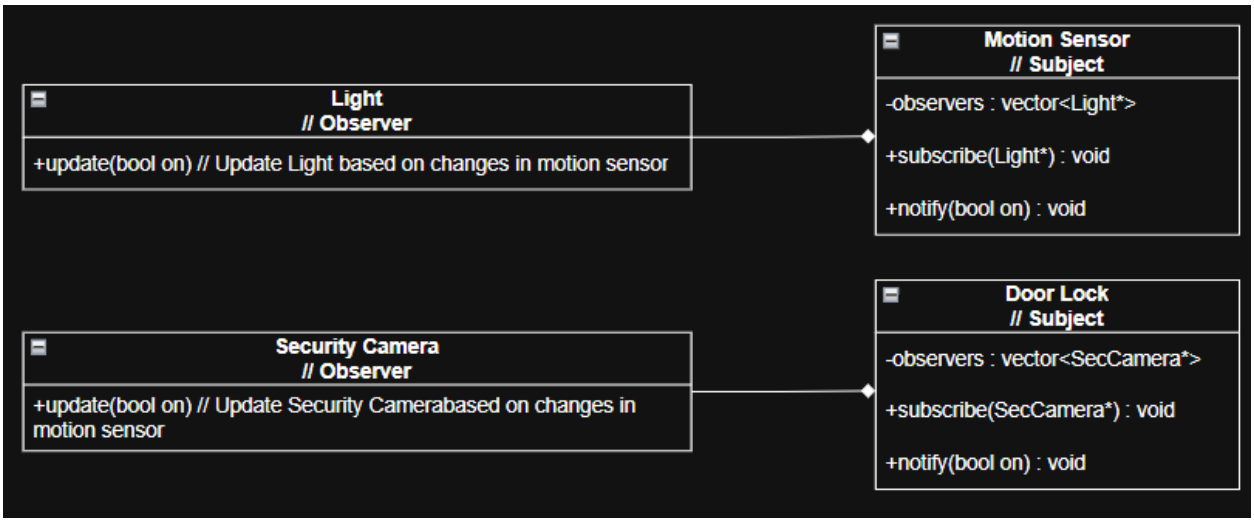
- Used for Providing a simplified interface that clients can use to interact without needing to understand the underlying complexities
- Mapped UML Design:



3. Behavioral Design Patterns Used

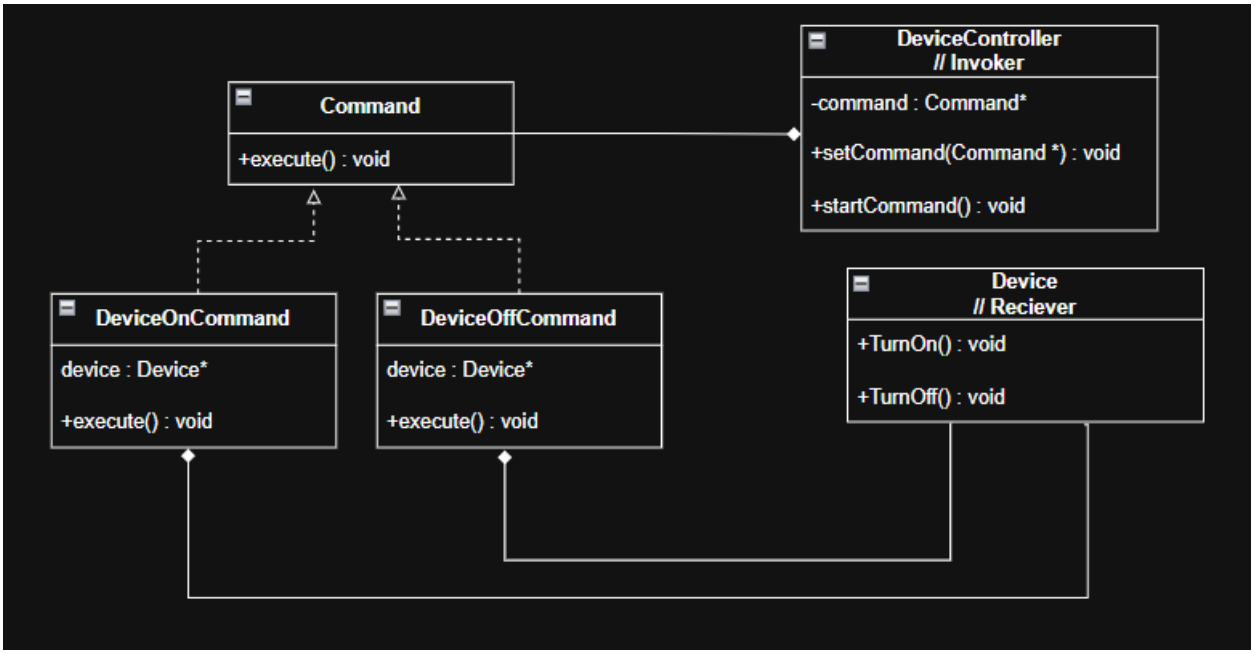
a) Observer Design Pattern

- Used to Design a system where devices can communicate changes or updates to interested components automatically is relations between classes (light + motion sensor) & (door lock & security camera)
- Mapped UML Design:



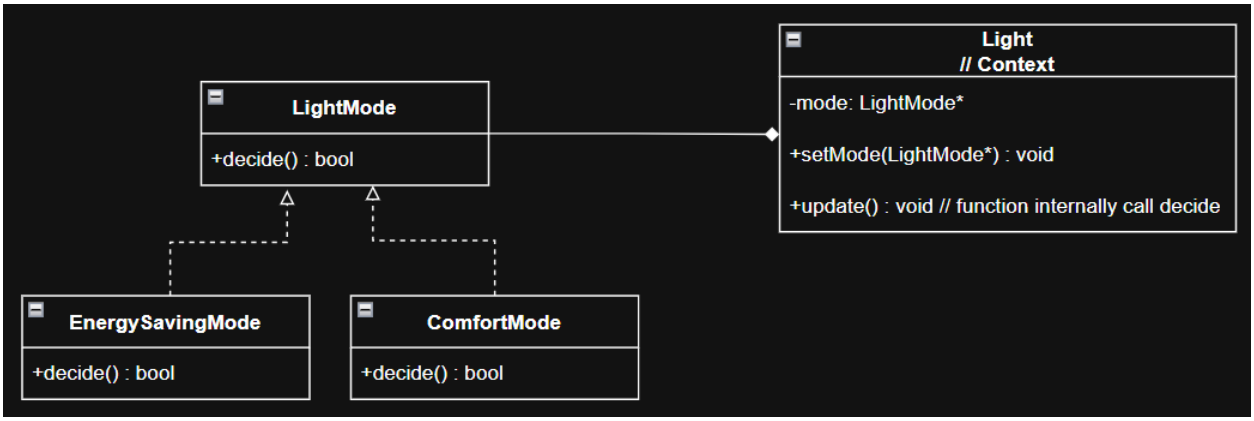
b) Command Design Pattern

- Used to Implement a way to represent user or automated requests as objects that can be queued, executed, or undone
- Mapped UML Design:



c) Strategy Design Pattern

- Allow different modes of automation logic to be applied, which can be changed or switched at runtime for classes like Light
- Mapped UML Design:



d) State Design Pattern

- Handle the internal states of devices effectively
- Mapped UML Design: