Smart Home

Description

My task was to create a smart home simulation using design patterns.

Functional requirements

F1 ✓

F2 ✓

F3 ✓

F4 ✓

F5 ✓

F6 ✓

F7 ✓

F8 ✓

F9 ✓

F10✓

Used design patterns

Chain of Responsibility

All events have Handler.

Observer

Within my system, entities whose functionality is time-dependent are followed by "time" component.

Builder

This pattern is employed to establish the configuration of my residence.

Factory method

Utilized in tandem with the builder to craft items, devices and inhabitants.

An abstract factory

Employed alongside the builder to generate various sensor types, including internal and external variants(but I didn’t have time to make the external sensors).

Proxy

The SmartHome class encompasses a "start" method that invokes the "start" method of the Simulation class. Introducing the Proxy pattern in this context could involve using an intermediary proxy object to control or add functionality before the actual invocation of the "start" method in the Simulation class, providing an additional layer of abstraction and control in the process.

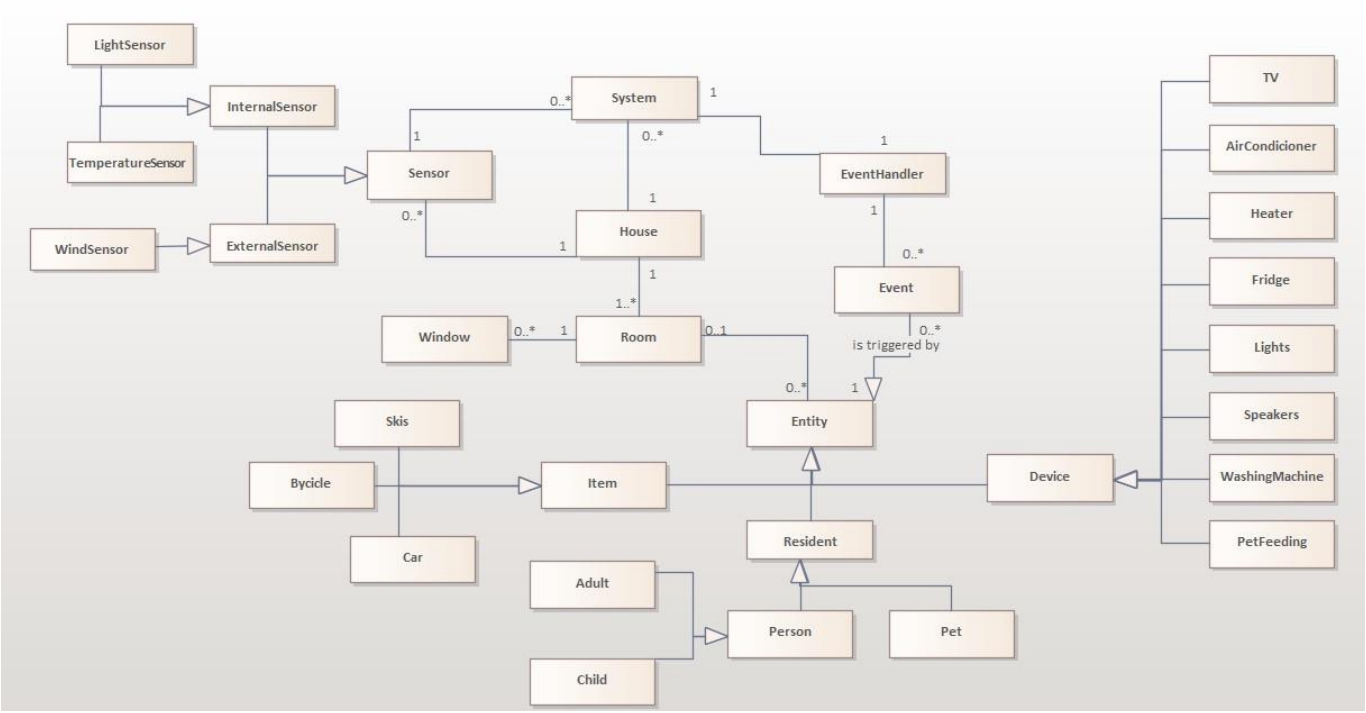
Singleton

Designed to house fundamental system state information and manage system controls, including the initiation and termination of the simulation and links to key system components.

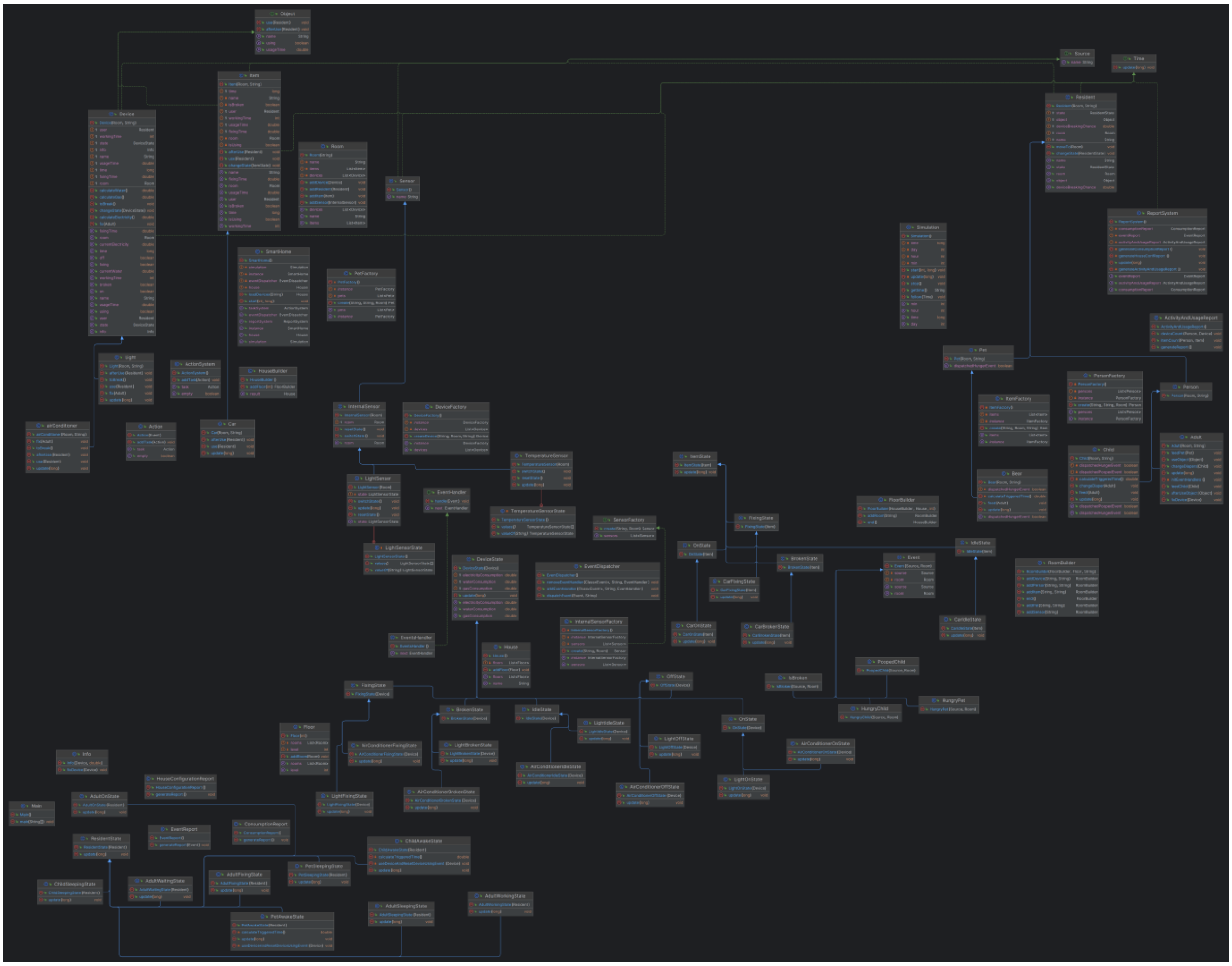
State machine

Every resident and device can exist in various states, each influencing their behavior differently.

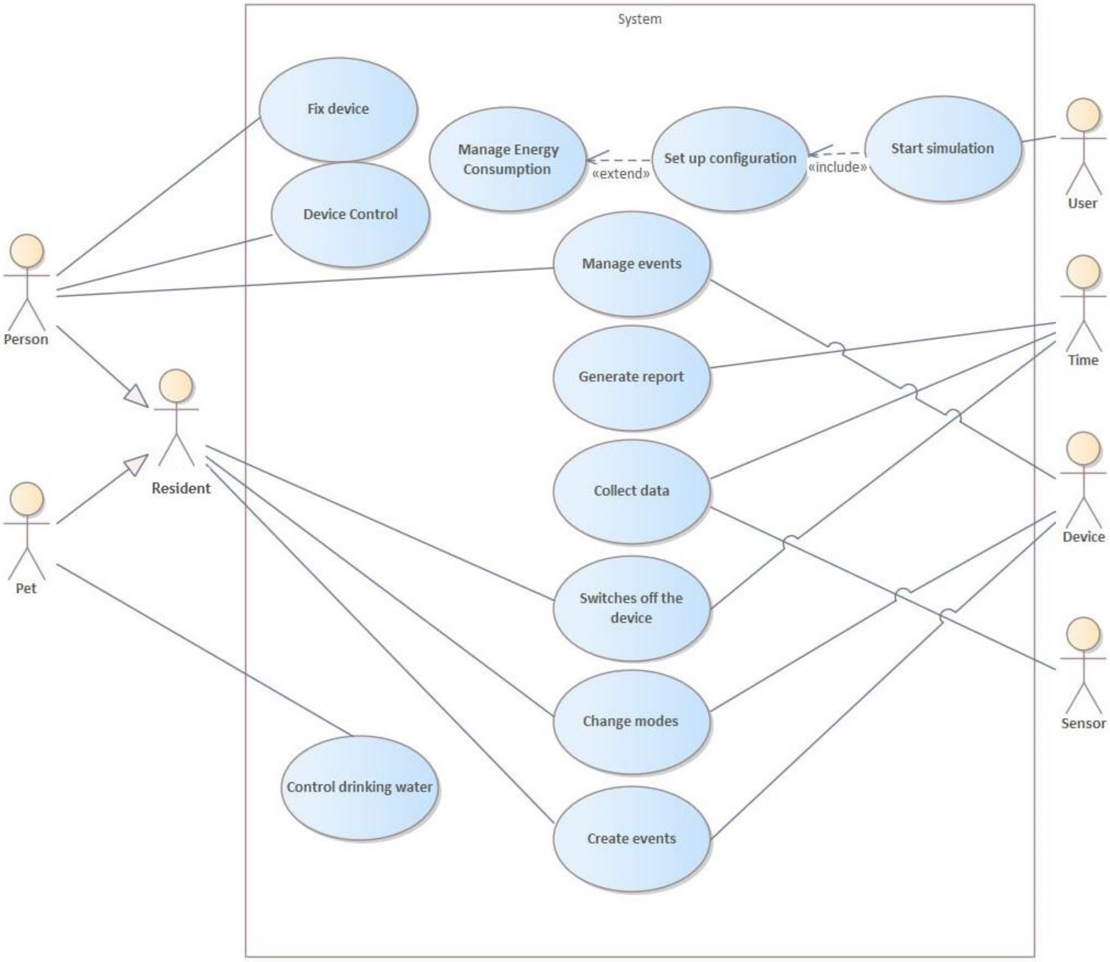
UML class diagram original



Use case diagram implemented



Use case diagram


System
Fix device
Manage Energy Consumption
«extend»
Set up configuration
«include»
Start simulation
User
Device Control
Manage events
Person
Generate report
Time
Resident
Collect data
Device
Pet
Switches off the device
Sensor
Change modes
Control drinking water
Create events