

**Hacettepe University**  
**Project Assignment 2: Smart Home System**  
**Süleyman Yolcu – 2210765016**



Author: Süleyman YOLCU

Course: BBM104

Instructor: Gökem AKYILDIZ

Date: 21.04.2023

[b2210765016@cs.hacettepe.edu.tr](mailto:b2210765016@cs.hacettepe.edu.tr)

# INDEX

## Problem and Solution-----2

Implementing the devices-----2

Implementing the time-----2

Implementing the methods-----2

## Benefits of this System-----2

## Benefits of OOP-----2

## UML Diagram-----3

Main-----4

SmartHome-----4

SmartDevices-----4

SmartPlug-----4

SmartLamp-----4

SmartColorLamp-----4

SmartCamera-----4

Commands-----4

FileInput-----4

# 1 Problem and Solution

The project includes the following smart home accessories: Smart Plug, Smart Camera, and Smart Lamp (with white ambience and color-white ambience variations). The work at hand involves controlling these tools and the passage of time in accordance with commands. The project can be divided into several parts, including implementation of the devices and their bodies, implementation of time passage, and implementation of the necessary methods for handy use.

## 1.1 Implementing the Devices

During the implementation of smart devices, an imaginary UML diagram was drawn and the inheritance was decided. Then, common attributes were determined for some devices with unique methods and attributes. The first classes were created, and the devices were stored in a LinkedHashMap because the order of object creation was important in the project.

## 1.2 Implementing the Time

The abstract time concept was created by holding some time information and controlling it through search operations. The decision was made to hold the time variable in the LocalDateTime type.

## 1.3 Implementing the Methods

The most time-consuming part of the project was implementing the methods for every possible scenario to avoid hard coding, which required a lot of time and effort. Ensuring the suitable commands was difficult, and almost every command needed special attention to handle possible errors. The ChangeName command was particularly challenging. The smart devices were stored in a LinkedHashMap, which held device names as the key (String) and objects as values. Since map keys are immutable, it was difficult to find a solution. The solution was found by changing the shape of the map, performing necessary operations, and then returning to the original shape.

# 2 Benefits of this System

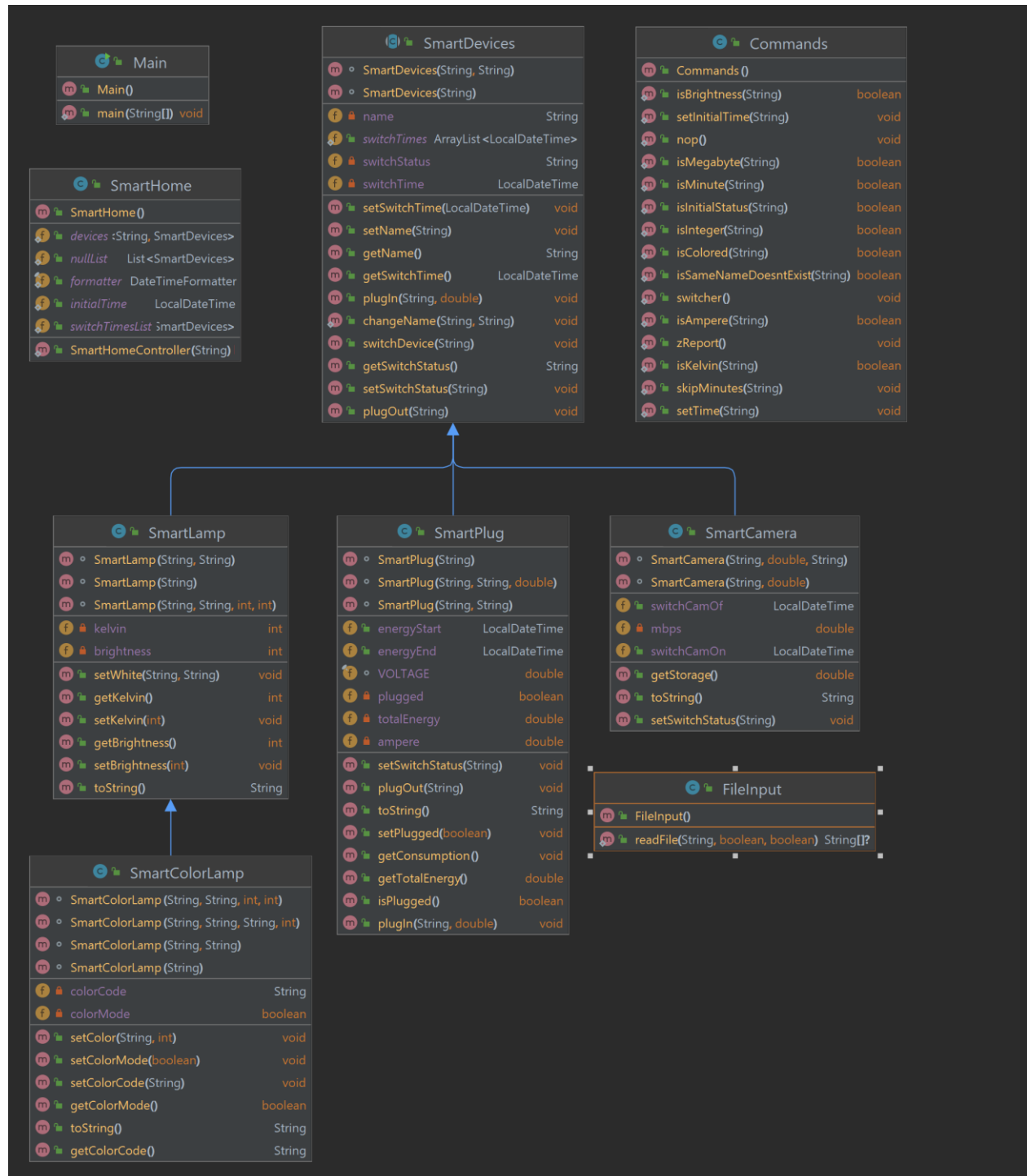
A lot of benefits are provided by this system. The switching on or off of the lamps can be decided automatically. The colors and brightness of the lamps can be arranged according to desire. The energy consumption of devices can be calculated using Smart plug. If Smart Cameras are used to record footage, the switching on or off can be done automatically, and the storage it used can be calculated.

# 3 Benefits of OOP

A huge software system can be divided into smaller, easier-to-manage modules, each with its own data and behavior, thanks to OOP. Because modifications to one module will have less of an effect on the other modules, this makes the system easier to manage and upgrade. Encapsulation, or the keeping of an object's data and functionality in a single unit. This helps to prevent unauthorized access to or change of the object's data and makes the object simpler to comprehend and use. Because OOP supports inheritance, you may make new objects that take on the characteristics and behaviors of

older ones. You can reuse code from existing objects rather than building new ones, which can save time and effort.

## 4 UML Diagram / Explanation



#### **4.1 Main**

The SmartHome is created and run with SmartHomeController by the Main class. Input output is also handled by it.

#### **4.2 SmartHome**

The devices and other necessary data are held by the SmartHome class. The SmartHomeController method to control the Home is also present in it.

#### **4.3 SmartDevices**

The necessary constructors along with some crucial methods related to Time are present in this class. Getters and setters for private fields are also available. Also, it is an AbstractClass. However, it does not have any abstract method because not all devices require the same method and overriding them would be unnecessary. It has two empty-bodied methods (plugIn, plugOut) which will be defined in SmartPlug.

#### **4.4 SmartPlug**

Three type constructors are present in the SmartPlug class. It also holds the information related to energy consumption. Unique methods like plugIn and plugOut are present.

#### **4.5 SmartLamp**

This class is a superclass. It has unique fields like kelvin and brightness. It also has methods like setWhite, setKelvin, and SetBrightntess which will be inherited by SmartColorLamp.

#### **4.6 SmartColorLamp**

The color code is a unique variable in this class which is modifiable using the setColor and setColorMode methods. It is a subclass of SmartLamp.

#### **4.7 SmartCamera**

The Storage it uses is calculated by the SmartCamera class's switchCamOn and switchCamOf fields.

#### **4.8 Commands**

This is the class that holds all important, necessary methods to avoid hardcoding. The diagram shows the variety of methods.

#### **4.9 FileInput**

This class is used solely for getting inputs.

Important: The toString method for all sub-classes of SmartDevices has been overridden in order to display the attributes.