Ankara Yıldırım Beyazıt University
Department of Computer Engineering

CENG 201 – Object Oriented Programming
Course Project

G##: Smart Home Automation

Design Report

Bilgenur Erkan,
Buse Köroğlu,
Enes Eldeş,
Enes Mert Aydın,
İdris Aydın

Instructor: Muhammed Abdullah Bülbül
Teaching Assistant: Elif Şanlıalp, Yusuf Şevki Günaydın
Date: 12/12/2024

Table of Contents

1. Introduction

This report presents a detailed analysis of a design prepared for a remote home automation system project. Classes such as *User*, *Home*, *Device* in the draft represent different components of the system and the relationships between them. This remote home automation system project is a technology that allows various devices at home to be controlled remotely. Thanks to these systems, users will be able to manage many different features such as lighting, heating, security systems, even when they are not at home.

One of the most important features of the project is the security system that is installed to ensure safety in the house. This system can detect important events and can directly trigger the house alarm. In addition, by following the movements of the people living in the house, user can intervene quickly in the event of a possible emergency.

Another concept of the project *Settable* devices are to specify that the particular device will have special attributes and behaviors.

In this section, the basics of the project are briefly explained. More detailed information will be given in the following sections.

2. Class Responsibility Collaboration (CRC) Cards

The class responsibility collaboration cards were written in detail to show the relations and behaviors of each class. These cards are important to give a detailed overview of the project

2.1. Base CRC Cards

The classes in the figure 1 form the basis of the project. *Home* class is the system itself. It manages all the *Users* and *Rooms*.

Home	
Responsibility	**Collaboration**
- Manage the home automation system, including users and rooms.	- Room, User
- Save and load the state of the home to a file.	

Room	
Responsibility	**Collaboration**
- Manage room-specific properties.	
- Manage devices, check for risky situations.	- Device

Device	
Responsibility	**Collaboration**
- Open and close the device.	
- Display device information.	
- Calculate thread level and electricity consupmtion.	

User	
Responsibility	**Collaboration**
- Store and verify user login credentials.	
- Enable interaction with the smart home system.	- Home
- Manage rooms and control smart devices.	- Room

Admin	
Responsibility	**Collaboration**
- Add and remove new users to the system.	- Home
- Add and remove devices in specific rooms.	- Device, Room

Child	
Responsibility	**Collaboration**
-Simulate child-specific behavior, such as reacting (cry).	

Figure 1. Base CRC Cards

2.2. SecurityDevice CRC Cards

Security system of the project is shown in the figure 2. Here, *SecurityAlarm* is a separate class and listens the signals from the *SecurityDevice* classes which they can trigger SecurityAlarm.

SecurityDevice	
Responsibility	**Collaboration**
- Detect and evaluate threats in a room.	- Room
- Trigger alarm.	- SecurityAlarm

SecurityAlarm	
Responsibility	**Collaboration**
- Analyze threat levels based on inputs from the security devices.	- SecurityDevice
- Trigger alarms when a critical condition is detected.	

SecurityCamera	
Responsibility	**Collaboration**
- Monitor the room and detect motions.	- Room
- Alert the system in case of suspicious activity.	- SecurityAlarm

Detector	
Responsibility	**Collaboration**
- Detect and evaluate threats in a room.	- Room
- Trigger alarm.	- SecurityAlarm

SmokeDetector	
Responsibility	**Collaboration**
- Check thread level of the devices in a room.	- Room
- Trigger alarm.	- SecurityAlarm

MotionDetector	
Responsibility	**Collaboration**
- Detect the motion in a room.	- Room
- Trigger alarm.	- SecurityAlarm

Figure 2. SecurityDevice CRC Cards

2.3. SettableDevice CRC Cards

SettableDevice class represents all the devices that can be properties to set within a home. It allows to set not only an attribute, but a timer and also a work mode for that particular machine. (For example: Set Oven as Grill mode at 250C degrees for 55 minutes)

Although SettableDevice had more subclasses, some of them were eliminated to build a more *user-friendly* and *realistic* automation system.

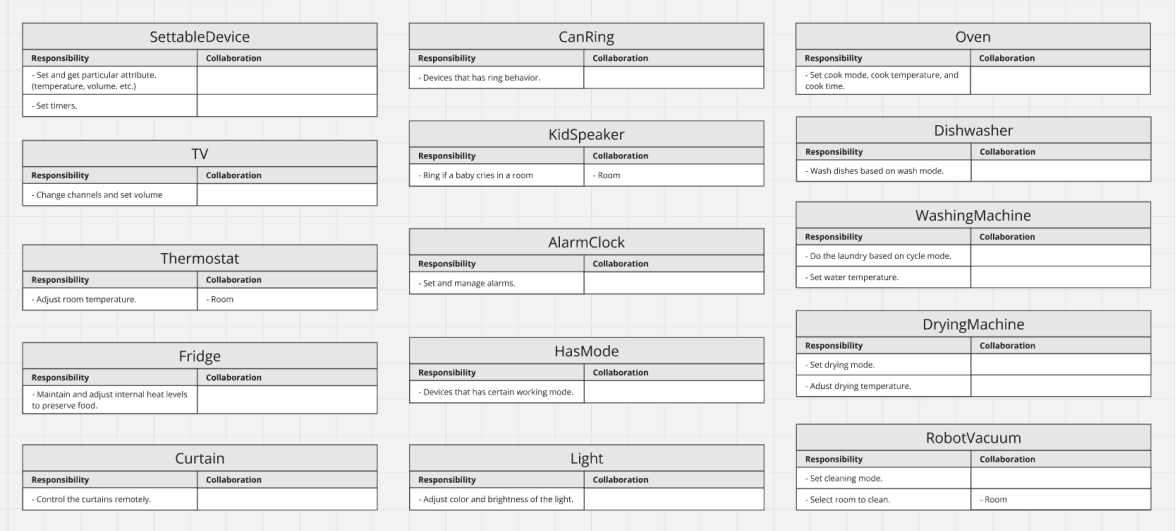

Figure 3. SettableDevice CRC Cards

3. Class Diagram

The ensure that a project is created in accordance with Object Oriented Programming Concepts, class diagram is created after a fierce brainstorming process as a whole team. The result of this team work can be seen in the figures 4, 5, and 6 part by part. Also check figure 7 to see the complete class diagram. (Red classes are abstract, blues are enumerations)

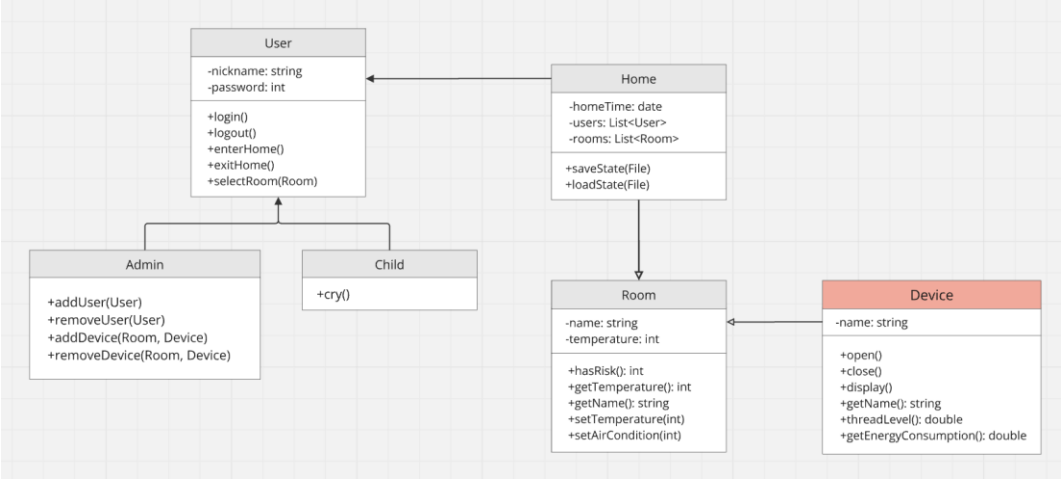

Figure 4. Base class diagram

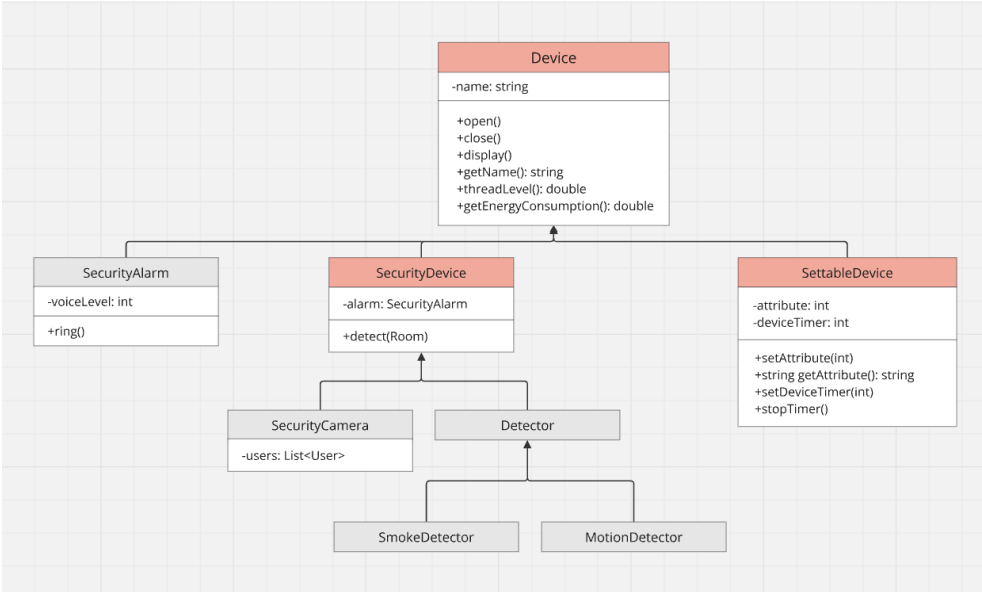

Figure 5. Device hierarchy

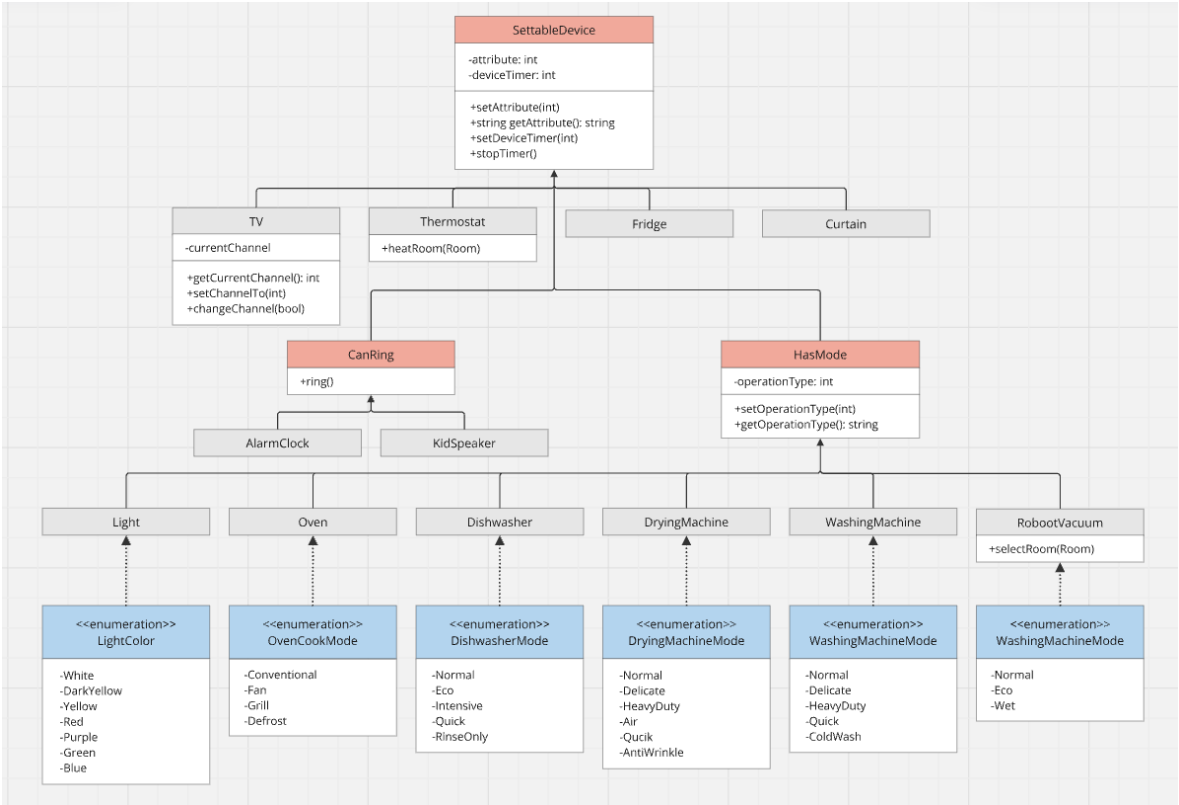

Figure 6. SettableDevice hierarchy

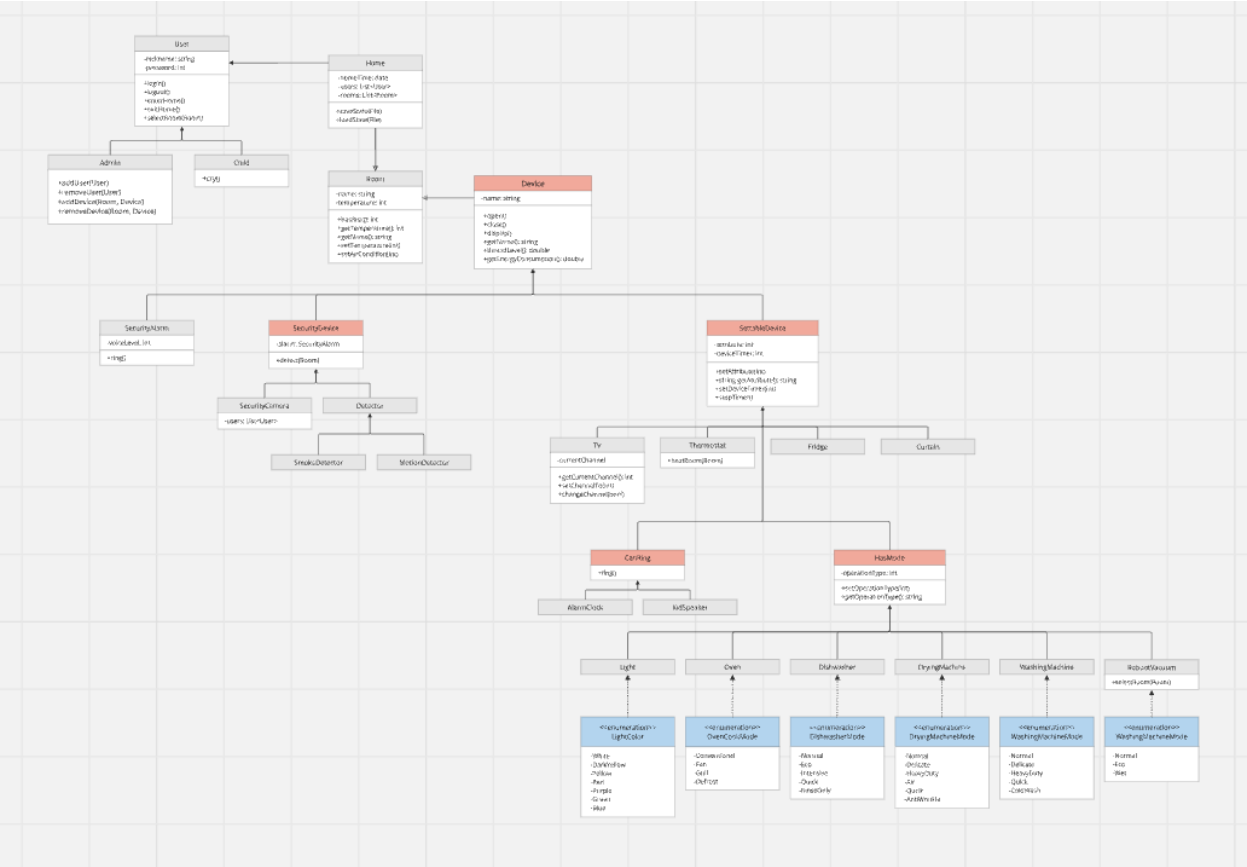

Figure 7. Project complete class diagram (Texts may not be seen)

4. Conclusion

In summary, the smart home automation system provides a centralized, user-friendly platform for managing smart devices, with a focus on enhanced security, flexibility, and convenience. Leveraging object-oriented programming principles, the system is designed to be both scalable and adaptable to meet evolving user needs.

With this report, class diagram of the smart home project is designed. Class responsibilities, collaborations, special methods and fields are clarified. All necessary preliminary preparations have been made to proceed to the construction phase of the project.

Each team member played a vital role in the project:

- Class diagram of the project is created together with all team members through both face-to-face and online meetings.
- Bilge and Buse prepared the CRC cards together.
- İdris prepared the introduction section.
- Enes Mert Aydın prepared the conclusion section.
- Enes Eldeş gave the final look to the class diagram, led the preparation of the report.