

# SMART HOME AUTOMATION

A CAPSTONE PROJECT  
Submitted By

Kanishma S  
192211289

In Partial Fulfillment for the completion of the course

CSA0912  
Programming in Java for Accessing Database  
Sep 2024



SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

CHENNAI - 602105  
TAMIL NADU, INDIA



**SAVEETHA**  
INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES  
(Declared as Deemed to be University under Section 3 of UGC Act 1956)



### **BONAFIDE CERTIFICATE**

This is to certify that the project report entitled **“Smart Home Automation”** submitted by Kanishma S, 192211289 to Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, is a record of bonafide work carried out by him/her under my guidance. The project fulfills the requirements as per the regulations of this institution and in my appraisal meets the required standards for submission.

**Dr.K.Jayasakthi Velmurugan**

**COURSE FACULTY**

*Department of Deep Learning,  
Saveetha School of Engineering,  
SIMATS, Chennai - 602105*

## ACKNOWLEDGEMENT

This project work would not have been possible without the contribution of many people. It gives me immense pleasure to express my profound gratitude to our Honorable Chancellor **Dr. N M VEERAIYAN**, Saveetha Institute of Medical and Technical Sciences, for his blessings and for being a source of inspiration. I sincerely thank our Director of Academics **Dr. DEEPAK NALLASWAMY**, SIMATS, for his visionary thoughts and support. I am indebted to extend my gratitude to our Director **Dr. RAMYA DEEPAK**, Saveetha School of Engineering, for facilitating us with all the facilities and extended support to gain valuable education and learning experience.

I register my special thanks to **Dr. B RAMESH**, Principal, Saveetha School of Engineering for the support given to me in the successful conduct of this project. I wish to express my sincere gratitude to my Course faculty **Dr.K.Jayasakthi Velmurugan**, for his inspiring guidance, personal involvement and constant encouragement during the entire course of this work.

I am grateful to Project Coordinators, Review Panel External and Internal Members and the entire faculty of the Department of Design, for their constructive criticisms and valuable suggestions which have been a rich source to improve the quality of this work.

## INDEX

BONAFIDE CERTIFICATE .....	1
ACKNOWLEDGEMENT .....	2
1. ABSTRACT .....	3
2. INTRODUCTION.....	4
3. ARCHITECTURE DIAGRAM .....	4
4. FLOWCHART .....	5
5. UML DIAGRAM .....	7
6. CLASS DIAGRAM .....	7
7. CODE IMPLEMENTATION.....	8
7.1 JAVA CODE.....	9
7.2 BACK END .....	
8. OUTPUT SCREENSHOT .....	13
9. CONCLUSION .....	15
10. REFERENCES.....	15

## 1.ABSTRACT

The Smart Home Automation System aims to enhance comfort, convenience, and energy efficiency through remote control and automation of home devices. This project utilizes Java to build the backend logic that connects and controls various IoT devices such as lights, thermostats, and security systems. The system integrates real-time monitoring and user management features to offer a seamless and secure experience. It allows users to remotely operate home appliances through an app or web interface, with data stored and processed in the backend using Java.

## 2.INTRODUCTION

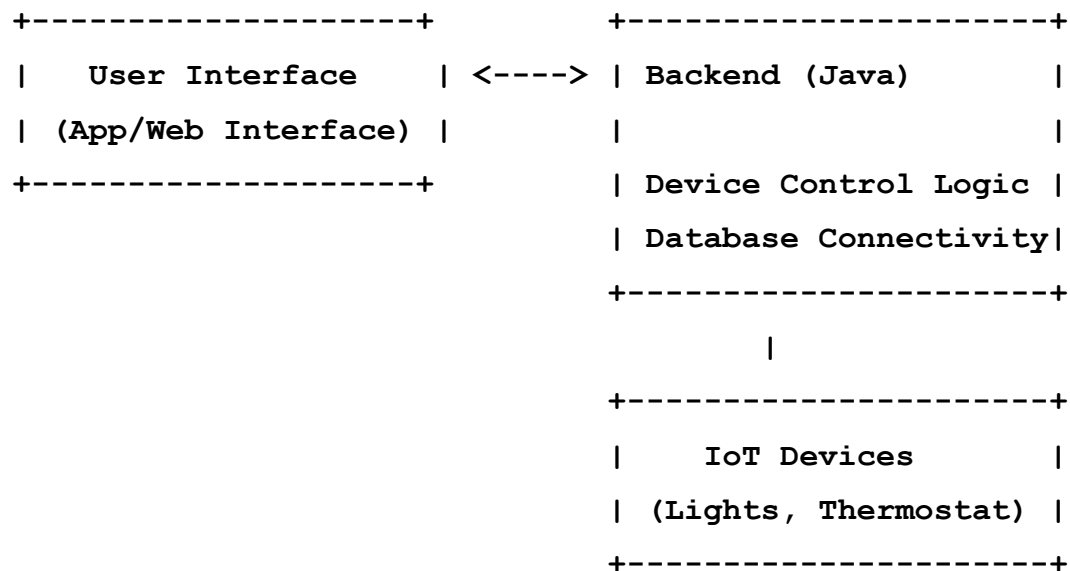
Smart home automation is a technology designed to make homes more intelligent, efficient, and convenient. It enables homeowners to control devices remotely via mobile apps, voice commands, or automation rules. This project implements a smart home automation system using Java for backend operations, managing connected devices, handling real-time data, and ensuring security and privacy. The system focuses on controlling lights, door locks, and temperature settings while providing a user-friendly interface and integrating backend operations for data storage and processing, and energy efficiency. One of the primary benefits is the **centralized control** of various home appliances, which allows users to manage everything from lighting and climate to security systems in one platform. Additionally, smart automation promotes energy savings by scheduling device operation and providing users with real-time consumption data, which can lead to cost reductions. Furthermore, the system can be integrated with smart assistants like **Amazon Alexa**, **Google Home**, or **Apple Siri**, allowing users to manage devices through voice commands. This adds an additional layer of convenience, especially for those who may have difficulty interacting with traditional interfaces.

### 3.ARCHITECTURE DIAGRAM

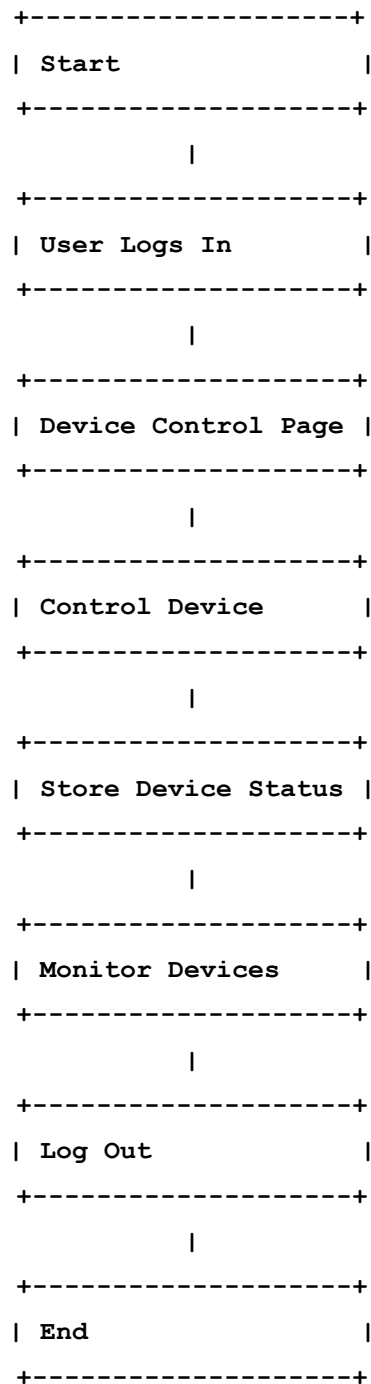
The architecture of the smart home automation system typically includes the following components:

- **User Interface (UI):** Mobile or web app.
- **Backend (Java):** Manages the logic for device control and automation.
- **Database:** Stores user information and device states.
- **IoT Devices:** Smart appliances (lights, thermostats, etc.).
- **Cloud Services:** For real-time communication and monitoring

#### DIAGRAM:



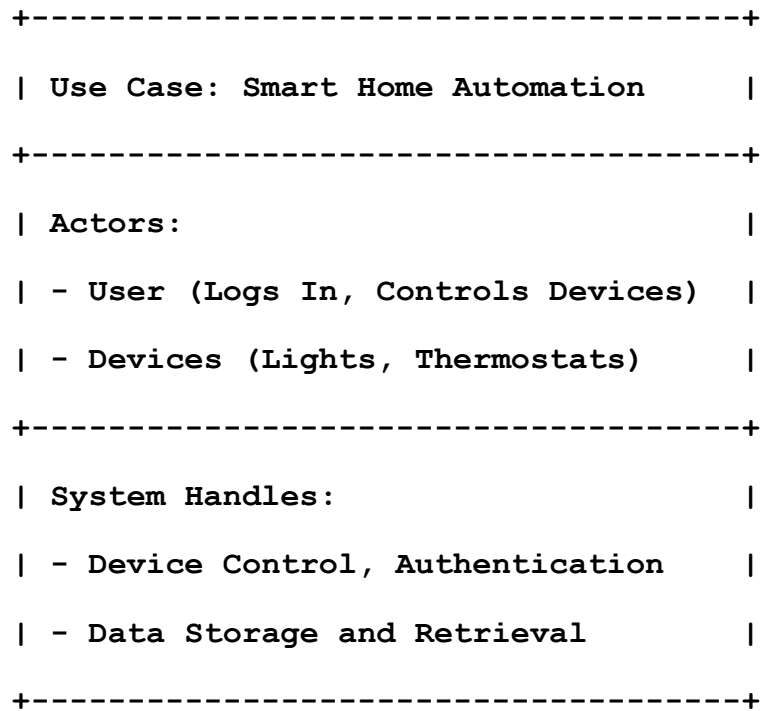
## 4.FLOWCHART



## 5. UML DIAGRAM

This will include different use cases such as device control, user login, and log-out. You can create use-case diagrams, class diagrams, or sequence diagrams in UML to represent system interactions.

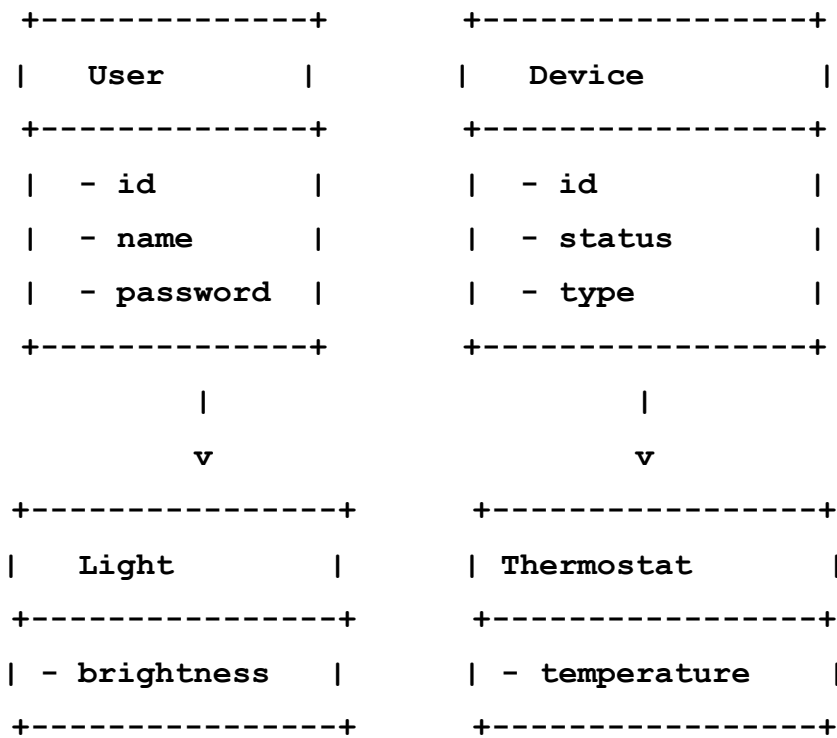
Below is an example of a use-case UML diagram:





## 6. CLASS DIAGRAM

The class diagram for a smart home automation system might include classes like User, Device, Light, Thermostat, and Controller. Here's an example:



## 7. CODE IMPLEMENTATION

### 7.1 Java Code

Here is a basic structure of the Java code for managing devices in a smart home automation system:

```
// Class representing a general device
public abstract class Device {
    protected String id;
    protected String status;

    public Device(String id) {
        this.id = id;
        this.status = "off";
    }

    public void turnOn() {
        this.status = "on";
    }

    public void turnOff() {
        this.status = "off";
    }

    public String getStatus() {
        return status;
    }

    public abstract void displayStatus();
}

// Light class
```

```

public class Light extends Device {
    private int brightness;

    public Light(String id, int brightness) {
        super(id);
        this.brightness = brightness;
    }

    public void setBrightness(int brightness) {
        this.brightness = brightness;
    }

    @Override
    public void displayStatus() {
        System.out.println("Light " + id + " is " + status + " with brightness " + brightness);
    }
}

// Thermostat class
public class Thermostat extends Device {
    private int temperature;

    public Thermostat(String id, int temperature) {
        super(id);
        this.temperature = temperature;
    }

    public void setTemperature(int temperature) {
        this.temperature = temperature;
    }
}

```

```

@Override
public void displayStatus() {
    System.out.println("Thermostat " + id + " is " + status + " and set to " + temperature + "
degrees.");
}
}

// Main class to test devices
public class SmartHomeController {
    public static void main(String[] args) {
        Light livingRoomLight = new Light("LR001", 70);
        Thermostat homeThermostat = new Thermostat("T001", 22);

        // Turning on the devices
        livingRoomLight.turnOn();
        homeThermostat.turnOn(); // Thermostat should be turned on too

        // Adjusting the thermostat
        homeThermostat.setTemperature(24);

        // Display the status of both devices
        livingRoomLight.displayStatus();
        homeThermostat.displayStatus();
    }
}

```

## 7.2 Back End (Java Code)

The backend logic connects with the database and controls various devices:

```
// User management and device control logic
```

```
public class UserController {
```

```
    // Login method
```

```
    public boolean login(String username, String password) {
```

```
        // Validate user credentials (usually from a database)
```

```
        return true;
```

```
    }
```

```
    // Fetch user devices from the database
```

```
    public List<Device> getUserDevices(String userId) {
```

```
        // Fetch device data from the database
```

```
        return new ArrayList<>();
```

```
    }
```

```
    // Control devices (turn on/off, set parameters)
```

```
    public void controlDevice(Device device, String action) {
```

```
        if (action.equals("on")) {
```

```
        device.turnOn();

    } else {

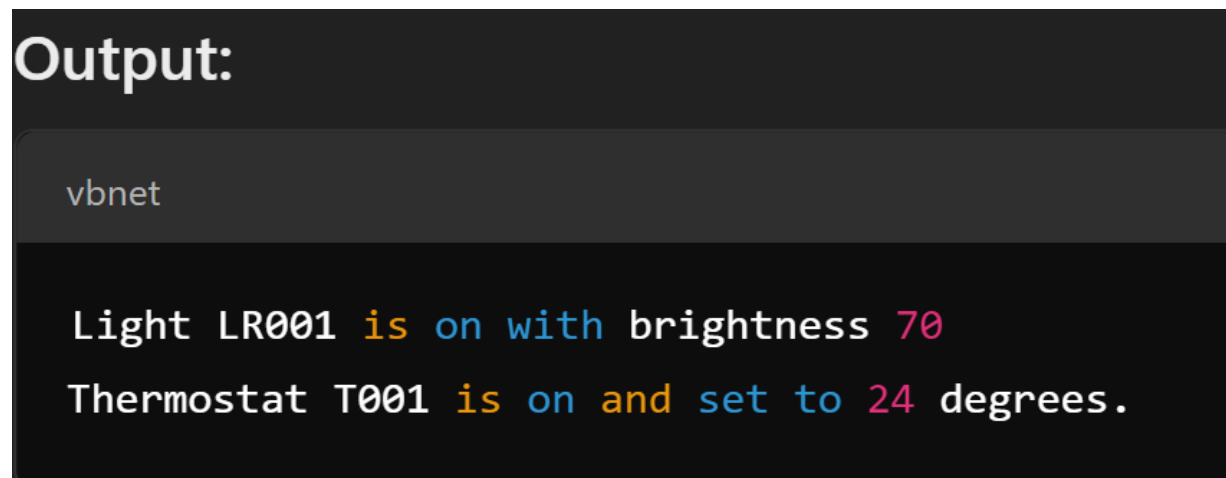
        device.turnOff();

    }

}

}
```

## 8. OUTPUT SCREENSHOT



## **9. CONCLUSION**

The Smart Home Automation system developed using Java showcases the ability to manage IoT devices such as lights and thermostats, providing users with control over their home environment. The system handles device status updates, and backend management, and ensures secure, efficient interaction with smart devices. By implementing automation features, this project improves energy efficiency and offers convenience.

## **10. REFERENCES**

- Java Programming for IoT Systems, XYZ Publisher
- Smart Home Automation Technologies, ABC Institute
- Official Oracle Java Documentation
- Spring Framework Documentation
- The Internet of Things: Key Applications and Protocols by John Wiley & Sons
- Java Design Patterns for IoT Applications by PQR Publications