Design an Energy Management System for a Smart Home Application using Object-Oriented Programming Techniques

Suhaili Beeran Kutty
Faculty of Electrical Engineering
Universiti Teknologi MARA Johor Branch, Pasir Gudang Campus
Johor, Malaysia
suhaili88@uitm.edu.my

Ariefah Nadya Azhari
Faculty of Electrical Engineering
Universiti Teknologi MARA Johor Branch, Pasir Gudang Campus
Johor, Malaysia
2023277524@student.uitm.edu.my

Noor Hazlini Borhan
Faculty of Computer Science and Information Technology
Universiti Malaysia Sarawak
Sarawak, Malaysia
bnhazlini@unimas.my

Zahari Abu Bakar
Faculty of Electrical Engineering
Universiti Teknologi MARA Johor Branch, Pasir Gudang Campus
Johor, Malaysia
zahar311@uitm.edu.my

Siti Hazurah Indera Putera
Faculty of Electrical Engineering
Universiti Teknologi MARA Johor Branch, Pasir Gudang Campus
Johor, Malaysia
hazurah@uitm.edu.my

2025 IEEE 8th International Conference on Electrical, Electronics, and System Engineering (ICEESE) | 979-8-3503-9246-3/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICEESE66784.2025.11212715

***Abstract*—The smart home concept has become increasingly popular due to its ability to enable remote and automated control of household systems. However, designing such systems presents challenges, particularly in ensuring operational functionality and energy efficiency. This paper presents the implementation of a smart energy management system using object-oriented programming (OOP) techniques. The system features a base class for generic devices and a derived class for smart devices, with an integrated energy metering unit that calculates total energy usage and manages interactions among devices. Polymorphism is employed within the smart device class to handle task-specific behaviors dynamically. The energy meter issues warnings when consumption reaches predefined limits. This implementation demonstrates key OOP principles, including inheritance, encapsulation, and polymorphism, in a real-world smart home context. Experimental results under controlled conditions validate the system's effectiveness and scalability for home automation applications, enabling efficient and reliable energy monitoring.**

Keywords*—energy management system, OOP, smart home, UML.

I. Introduction

A smart home is a house that is equipped with appliances and technology that can be remotely controlled and monitored from anywhere via smartphone, personal computers or electronic devices such as a smart watch that provides a voice assistant. Currently, the smart home concept has grown rapidly and the system has become complex [1]. The challenges are not only to the user but also to the developer of the system. According to [2], the developers must guarantee that the devices selected by the homeowner integrate efficiently within the ecosystem of the smart home.

The energy management system (EMS) is one of the most important components in the smart home ecosystem, where one of the purposes is to reduce the total cost of the operation [3]. The maximum wattage that a standard home outlet can support depends on the voltage and current rating of the circuit. In regions like Asia, including Malaysia, the standard voltage is 230 volts on a 13-amp circuit, which means the maximum wattage is approximately 3,000 watts. It is advisable to use no more than 80% of this maximum capacity to avoid overloading the circuit. Exceeding this limit could risk damaging devices or cause issues such as tripping the circuit breaker or even starting a fire.

This situation will undoubtedly necessitate managing homes with much greater efficiency in electricity consumption, especially considering the rising costs of electricity, along with major concerns about safety and environmental sustainability[4]. A well-defined energy management system for smart homes would enable residents to monitor their expenses and optimize energy usage while balancing affordability, comfort, and data privacy. Designing such a system will address the conflicting needs for advanced energy-saving features at low utility costs, ultimately leading to a highly energy-efficient system. It is a complex project to design and implement.

Developing a smart home electrical appliance monitoring system is a complex project, and using Object-Oriented Programming (OOP) with tools like Unified Modeling Language (UML) is the right way to design and structure it. The reason why UML is used in designing software is that it has a dominant role in contemporary system design ideas [5]. OOP uses a modular design where an object is created, and all of the objects come with attributes and methods [6]. For example, the solution to this problem is through the organization of the system in well-defined classes like appliances, sensors, energy meters, and controllers in the system. Each of which is defined by its attributes, such as energy consumption, appliance status, and sensor readings, and also includes methods that take care of real-time monitoring, automated controls, and energy savings suggestions. OOP also offers code reusability, scalability, and maintainability [7].

II. Related Works

As information and communication technologies advance, their integration into EMS becomes essential for achieving efficient and reliable power usage in residential settings. Smart homes are known to combine software, electrical systems, and user interfaces to automate and optimize energy consumption [8]. These systems aim to reduce electricity costs, enhance user comfort, and support sustainable energy practices [9], [10]. In particular, researchers have proposed various energy scheduling and monitoring strategies to minimize waste and improve performance.

In designing a smart home application, [5] effectively demonstrates how object-oriented modeling combined with UML diagrams can enhance the development of smart home automation systems. The approach ensures modularity, clarity, and easier translation into code, making it a solid foundation for further development, testing, and deployment in real-world smart home applications. In [11], they explicitly use object-oriented programming principles, especially in the modeling of the electrical thermal storage (ETS) system. The ETS unit is modeled using an object-oriented language tailored for physical systems. Each subsystem of the ETS (e.g., heating rod, bricks, control blocks) is modeled as an independent component and connected modularly. In recent developments, Internet of Things (IoT) technologies have also been employed to enhance smart home energy systems. The work in [12] successfully demonstrates how model-driven architecture with UML can be a powerful tool to handle device-level interoperability in industrial IoT. By using UML class diagrams and profiles, the approach simplifies the development process, eliminates dependency on network intermediaries, and provides a scalable, extensible, and platform-independent solution for heterogeneous IoT systems. UML tools provide the order of analysis diagram, it helps to display typical operations and the sequence of user operations to complete a function in the smart home EMS design [13]. Overall, the related works establishes a foundation for designing an object-oriented, modular, and simulation-based EMS that reflects the current innovations in smart home technologies.

III. Methodology

This effective energy management system's implementation involves a methodical, systematic approach using the concepts of object-oriented programming. The design and visualization of the entire system have been depicted through UML diagrams showing various components related to its key elements. These diagrams serve as the blueprints for defining system architecture including specification of class structures, interactions and functionality.

A. UML Use Case

The UML Use Case Diagram shows various interactions between the main actor, homeowner and energy management system of the smart homes. Homeowner as shown in Fig. 1 can perform many actions for example turning ON/OFF devices, monitoring energy usage, device status displaying, managing devices, and energy limiting. These use cases specify the functionalities that the system offers. In addition, include and exclude relationships are specified to show dependency among use cases. Energy usage monitoring and alert on high usage are included into limit energy management, so that users can receive alerts beyond threshold according to power consumption.

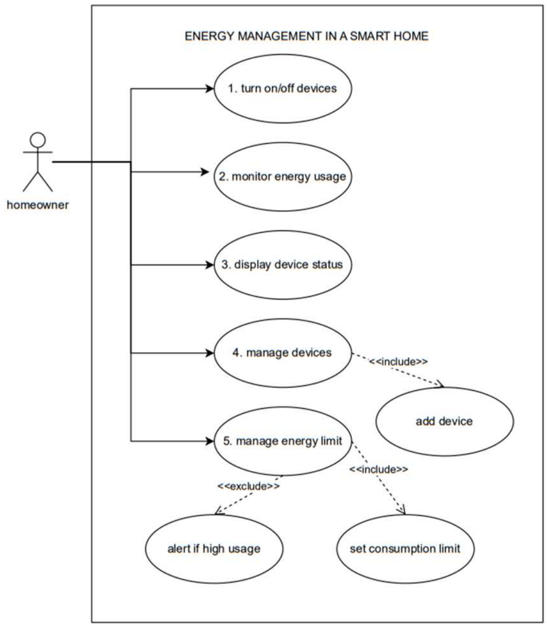

Fig. 1. UML Case Study

B. UML Class Diagram

The UML Class Diagram represents in detail the arrangement played among system actors to show the relationship of all the components. For example, Fig.2 shows UML class diagram for this paper. The class Device has properties such as name (for identification of devices), status (to indicate whether it is ON or OFF), and powerConsumption (which records the amount of power used). It shows that class SmartDevice has the above properties in addition to being applicable to further functionalities such as remote control. On the other hand, the EnergyMeter class tracks overall consumption whereby it defines methods for the update of energy usage and displays this information. The Controller class contains several devices along with the energy meter and defines methods, for instance, monitorUsage(), controlDevices(), and showDeviceStatus(). Hence, it connects with devices and energy meters wherein control on power consumption becomes centralized.

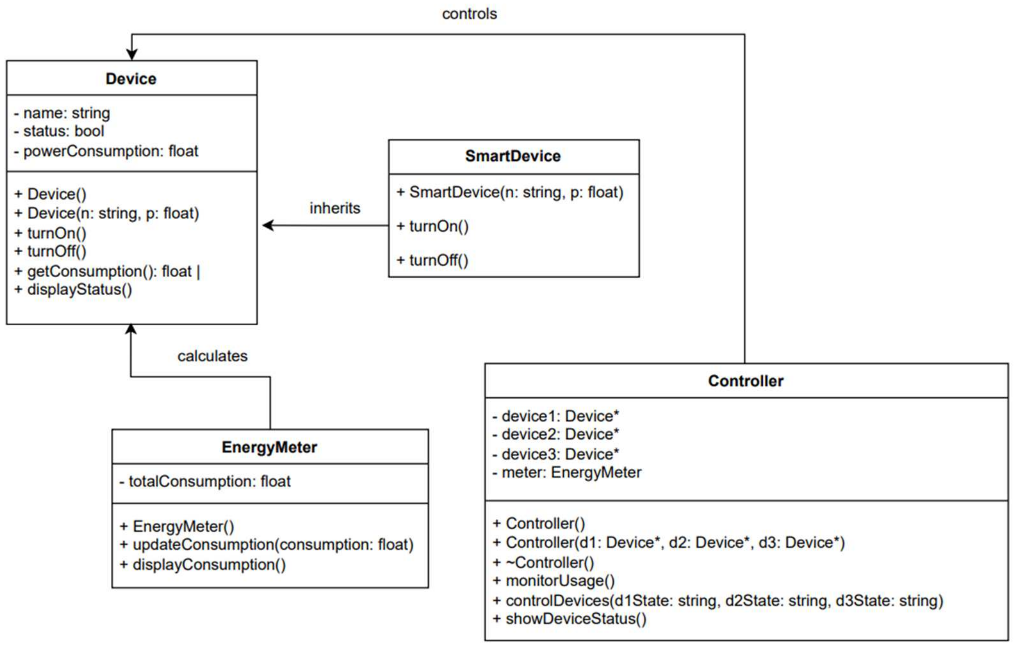

Fig. 2. UML Class Diagram

C. *UML Sequence Diagram*

The UML Sequence Diagram reflects the interaction between the components during a certain period over time regarding interaction with the system. Initially, the homeowner is interfacing with the controller to switch either ON/OFF of the devices. The controller will update the devices according to the actions and then communicate with the EnergyMeter for updating as well as displaying power usage. If power usage exceeds the predefined limit, the system will activate a warning on high energy and prompt the owner to act. This process will monitor and control energy usage in a smart home environment efficiently. UML sequence diagram of this project is shown in Fig. 3.

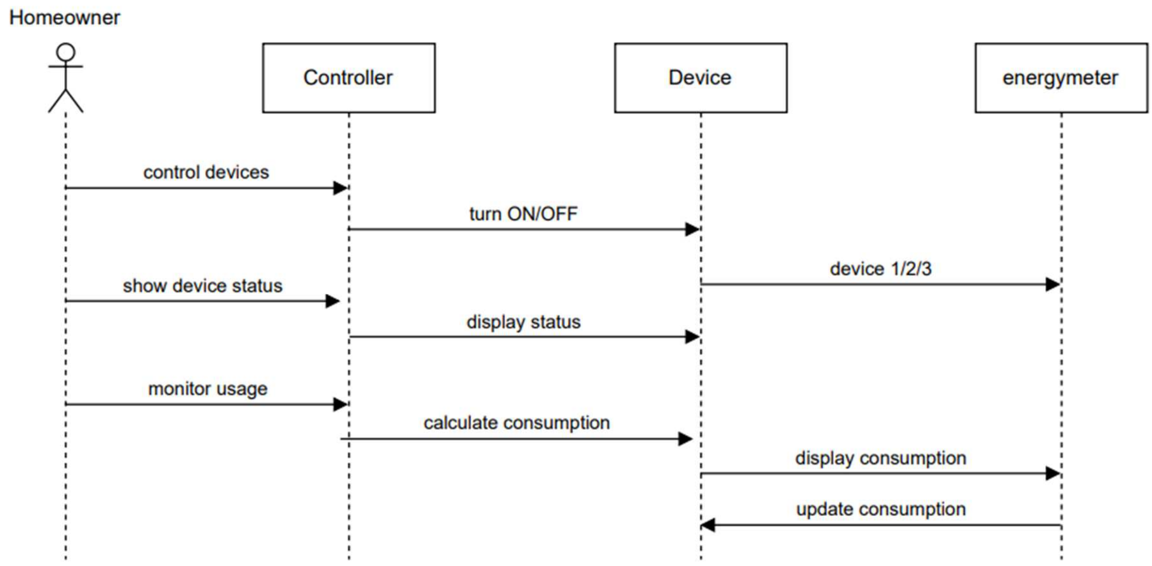

Fig. 3. UML Sequence Diagram

D. UML State Diagram

The UML State Diagram shows the different states in system behavior. Fig. 4 is shows the UML State Diagram of this project. The state of the controller is initialized without devices. Once devices are connected, they change states depending on what the homeowner commands. The energy consumption updates when the device switches to ON and evaluates if the consumption exceeds the limit. If the energy usage is still within limit, the system stays on its normal usage state. Otherwise, if consumption exceeds its threshold, it goes to a high energy usage warning state advising the homeowner to switch off some devices which are not in use. This approach is to ensure the efficient power management and user awareness of their energy consumption.

Fig. 4. UML State Diagram

IV. Results and Discussion

A. Object-Oriented Programming (OOP) Techniques Applied

The smart energy management system inside the house is done using basic object-oriented programming principles. The code is a combination of encapsulation, inheritance, polymorphism, and abstraction for the structuring of interactions between devices, energy monitoring, and control mechanisms. Table 1 gives a breakdown of the application of the OOP principles with the classes involved and their descriptions.

TABLE I. OOP PRINCIPLES IN THE CODE DEVELOPMENT

OOP technique	Class and Object	Description
Object & Classes (Encapsulation)	`Device`, `SmartDevice`, `EnergyMeter`, `Controller`	The data members are not directly accessible from outside the class
Inheritance	class SmartDevice : public Device	SmartDevice inherits the attributes and methods of an existing class Device
Polymorphism	**Base Class Pointers Point to Derived Class Objects**	It allows a base class pointer or reference to call derived class methods at runtime, depending on the actual object type, not the pointer type.
Abstraction	Device Class	The Device class provides an abstract interface to interact with any type of device.

B. Program Execution Flow

1) Default Controller (no devices)

Executed code:

```
Controller defaultController;
  defaultController.showDeviceStatus();
  defaultController.monitorUsage();
```

A Controller object (defaultController) is created without any devices (device1, device2, device3 are nullptr). showDeviceStatus() is called. Since theres no devices exist, nothing is displayed. Here is also where monitorUsage() is called: where total power consumption is 0.00W as there are no devices. There is also no warning message appears.

Output:

```
Total Energy Consumption: 0.00W
```

2) Creating Devices and a Controller

Executed code:

```
SmartDevice* fan = new SmartDevice("Fan", 75.0);
Device* light = new Device("Light", 40.0);
Device* kettle = new Device("Kettle", 1500.0);
Controller homeController(fan, light, kettle);
```

Three device objects are dynamically allocated:

- fan: SmartDevice, 75W
- light: Device, 40W
- kettle: Device, 1500W

A Controller (homeController) is created, managing these devices.

3) Controlling Devices

Executed code:

```
homeController.controlDevices("off", "on", "on");
```

The controlDevices() sets:

- fan: OFF
- light: ON
- kettle: ON

Output:

```
Fan is OFF. Power: 0.00W
Light is ON. Power: 40.00W
Kettle is ON. Power: 1500.00W
Total Energy Consumption: 1540.00W
```

4) Monitoring Energy Consumption

Executed code:

```
homeController.monitorUsage();
```

The total power consumption is calculated:

- Light: 40W
- Kettle: 1500W
- Fan: 0W

Total: 1540W

Since 1540W is more than threshold (1000W), a warning message is displayed.

Output:

```
WARNING: High Energy Usage! Consider turning off some devices.
```

V. Conclusion

To sum up, this project is intended to show and demonstrate the very need for OOP in designing efficient energy management for smart home ecosystem. This system can effectively control, monitor and optimize power consumption in different devices by organizing them into different classes. The availability of subclassing and polymorphism rules has a wide range of applications-generalizing various types of devices into a single system. In addition, by embedding an energy monitoring feature, this application tracks energy usage and alerts as soon as the amount of energy consumed becomes unsafe. This model resembles real-life automation and monitoring techniques that will be applied to actual situations to get energy-efficient applications and concepts into reality. This project can be termed as the very basis for almost all further development in the direction of smarter and sustainable energy management systems.

Acknowledgment

The authors would like to thank Universiti Teknologi MARA (UiTM) for the support and opportunity to publish this work.

References

[1] M. P. Woźniak *et al.*, "Connecting the dots: How users understand and diagnose smart home ecosystems," *International Journal of Human Computer Studies*, vol. 203, Sep. 2025, doi: 10.1016/j.ijhcs.2025.103559.

[2] D. Muñoz-Heredia, Á. J. Varela-Vaca, D. Borrego, and M. T. Gómez-López, "Optimising secure and sustainable smart home configurations," *Internet of Things (The Netherlands)*, vol. 32, Jul. 2025, doi: 10.1016/j.iot.2025.101637.

[3] M. Ebeed, S. hassan, S. Kamel, L. Nasrat, A. W. Mohamed, and A. R. Youssef, "Smart building energy management with renewables and storage systems using a modified weighted mean of vectors algorithm," *Sci Rep*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-024-79782-5.

[4] S. S. Ahmad, F. Almasalha, M. H. Qutqut, and M. Hijjawi, "Centralized smart energy monitoring system for legacy home appliances," *Energy Informatics*, vol. 7, no. 1, Dec. 2024, doi: 10.1186/s42162-024-00334-2.

[5] S. Popov and V. Bureva, "Object-Oriented Modelling of Home Automation Management System," in *2022 22nd International Symposium on Electrical Apparatus and Technologies, SIELA 2022 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/SIELA54794.2022.9845692.

[6] N.-K. P.E., O. F.U., D. A.U., I. J.S., and O. N.H., "Detailed Study of the Object-Oriented Programming (OOP) Features in Python," *British Journal of Computer, Networking and Information Technology*, vol. 6, no. 1, pp. 83–93, Nov. 2023, doi: 10.52589/bjcnit-facsojao.

[7] Agu S. C. and Elugwu F., "Object Oriented Programming Approach A Panacea for Effective Software Development," *African Journal of Advanced Sciences & Technology Research*, vol. 6, no. 1, pp. 1–14, Sep. 2022, [Online]. Available: www.afropolitanjournals.com

[8] G. Akçay, B. G. Celik, and A. Eroğlu, "Optimizing Energy Efficiency in IoT-Based Smart Home Systems," in *S. Koçer & Ö. Dündar (Eds.), Intelligent Systems and Optimization in Engineering*, ISRES Publishing, 2024, ch. 5. [Online]. Available: https://www.researchgate.net/publication/387516833

[9] N. P. Manchev, B. D. Karapenev, and K. K. Angelov, "Design and Development of Smart Home Electrical Management and Switching Panel," in *2021 30th International Scientific Conference Electronics, ET 2021 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Sep. 2021. doi: 10.1109/ET52713.2021.9580002.

[10] S. Shikhli, A. Shikhli, A. Jarndal, I. Alsyouf, and A. Cheaitou, "Towards Sustainability in Buildings: a Case Study on the Impacts of Smart Home Automation Systems," in *2022 Advances in Science and Engineering Technology International Conferences, ASET 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ASET53988.2022.9735126.

[11] J. A. Dominguez, L. Rueda, N. Henao, K. Agbossou, and J. Campillo, "Distributed Co-simulation for Smart Homes Energy Management in the Presence of Electrical Thermal Storage," in *IECON Proceedings (Industrial Electronics Conference)*, IEEE Computer Society, 2022. doi: 10.1109/IECON49645.2022.9969092.

[12] A. Amjad, F. Azam, and M. W. Anwar, "Device Interoperability for Industrial IoT using Model-Driven Architecture," in *2022 24th International Multitopic Conference, INMIC 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/INMIC56986.2022.9972976.

[13] Xiaoyun Zhao, Chunlai Zhou, and Wenhui Huang, "Smart Home Power Management System Design Based on Human-Computer Interaction Model," in *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology : ICCSNT 2013 : Oct 12-13, 2013, Dalian, China*, IEEE, 2013, pp. 1247–1250.