

HACETTEPE UNIVERSITY

Department of Computer Engineering

BBM104 – Assignment 2 Report

Ufuk Ağaya - 2210356064

21.04.2023



Index

1.....	Problem
2.....	Benefits of Smart Home System
3.....	Benefits of OOP
4.....	Four Pillars of OOP
5.....	Definition of UML

The Problem

The problem we are on is time wasting with home accessories. Most of our time is wasted on making all thing with the accessories. On this project, we are aiming reducing this time wasting with some smart home accessories that makes all easier for us. For instance a smart camera provides more effective work compared to an ordinary camera. That means that smart accessories make our things easier so we wouldn't need to spend time on them. With the extra time the smart home system gives us, we do anything we want, managing the is your bussiness. Smart home accessories can provide an added layer of security to your home, such as smart locks, cameras, and motion sensors. Without these accessories, you may have limited options for monitoring and securing your home, which can increase the risk of burglary, theft, or other security issues. Another effect of smart home accessories is energy issue. Smart home accessories are designed to be energy-efficient, helping you to save money on your utility bills and reduce your carbon footprint. Without these accessories, you may be using more energy than necessary to heat, cool, and light your home. Overall, not having smart home accessories can limit your convenience, energy efficiency, security.

Benefits of Smart Home System

Smart home systems offer a range of benefits that can make your life more convenient, secure, and efficient. Here are some of the key advantages:

1. **Convenience:** Smart home systems allow you to control your home's devices and appliances with your smartphone, voice commands, or even automated routines. This means you can easily turn off lights, adjust the temperature, or lock the doors from anywhere in the world.
2. **Energy savings:** Smart home systems can help you save on energy bills by optimizing the use of appliances and lighting. For example, smart thermostats can learn your preferences and adjust the temperature accordingly, while smart lighting systems can turn off lights when no one is in the room.
3. **Improved security:** Smart home systems can provide enhanced security features, such as remote monitoring of your home through security cameras, motion detectors, and door/window sensors. You can receive alerts and notifications when there's an unauthorized entry or suspicious activity.
4. **Peace of mind:** With a smart home system, you can easily check on your home while you're away, ensuring that everything is in order. This can be especially reassuring if you have pets or elderly family members living in your home.
5. **Increased home value:** Installing a smart home system can increase the value of your home, making it more attractive to potential buyers. It can also be a selling point if you decide to put your home on the market.

Overall, a smart home system can offer a range of benefits that make your life easier, safer, and more efficient.

Benefits of OOP

Object-Oriented Programming is a programming paradigm that emphasizes the use of objects and classes to organize and structure code. Here are some of the benefits of OOP:

1. **Modularity:** OOP enables breaking down a program into modular components known as objects, each possessing unique properties and methods that can be independently accessed and modified.
2. **Reusability:** OOP promotes the reuse of code through the concept of inheritance. This means that you can create a new class based on an existing class and inherit all of its properties and methods.
3. **Encapsulation:** OOP provides a way to hide the complexity of an object's internal workings from the outside world. This is known as encapsulation, and it allows you to create objects that are self-contained and have well-defined interfaces. This makes the code more robust, reliable, and easier to test.
4. **Polymorphism:** OOP supports polymorphism, which means that objects can have multiple forms or behaviors. This allows you to write code that can work with different types of objects without having to know the details of each object's implementation. This makes the code more flexible and adaptable to changing requirements.

Overall, OOP provides a range of benefits that make it a popular programming paradigm. It can help you to write code that is modular, reusable, robust, flexible, and easy to understand.

Four Pillars of OOP

1. **Encapsulation:** Encapsulation provides information hiding, which means that the internal workings of an object are hidden from the outside world. This provides security and makes it easier to change the implementation details of an object without affecting the rest of the system.
2. **Abstraction:** Abstraction refers to the process of identifying the essential characteristics of an object and ignoring everything else. It allows you to create high-level abstractions that are easy to understand and manipulate. Abstraction helps to reduce complexity and increase modularity in a system.
3. **Inheritance:** Inheritance is creating a new class based on an existing class, inheriting its properties and methods. This allows you to reuse code and create classes that share common attributes and behaviors. Inheritance promotes code reuse and makes it easier to maintain and extend a system.
4. **Polymorphism:** Polymorphism is the ability of an object to take on different forms or behaviors depending on the context. It allows you to write code that can work with different types of objects without having to know the details of each object's implementation. Polymorphism provides flexibility and adaptability in a system.

Definition of UML

Unified Modeling Language is a visual modeling language used to represent and design software systems. There is some diagram types:

1. Class diagrams: These diagrams represent the structure of a system, showing the classes, attributes, methods, and relationships between them.
2. Use case diagrams: These diagrams represent the interactions between actors (users, systems, or external entities) and the system, showing the different use cases or scenarios that the system supports.
3. Sequence diagrams: These diagrams represent the dynamic behavior of a system, showing the interactions between objects and the order in which they occur.
4. State machine diagrams: These diagrams represent the behavior of a single object over time, showing the different states and transitions that the object can go through.

UML is a powerful tool for modeling and designing software systems, helping to improve communication, understanding, and collaboration among developers, stakeholders, and end-users.

