

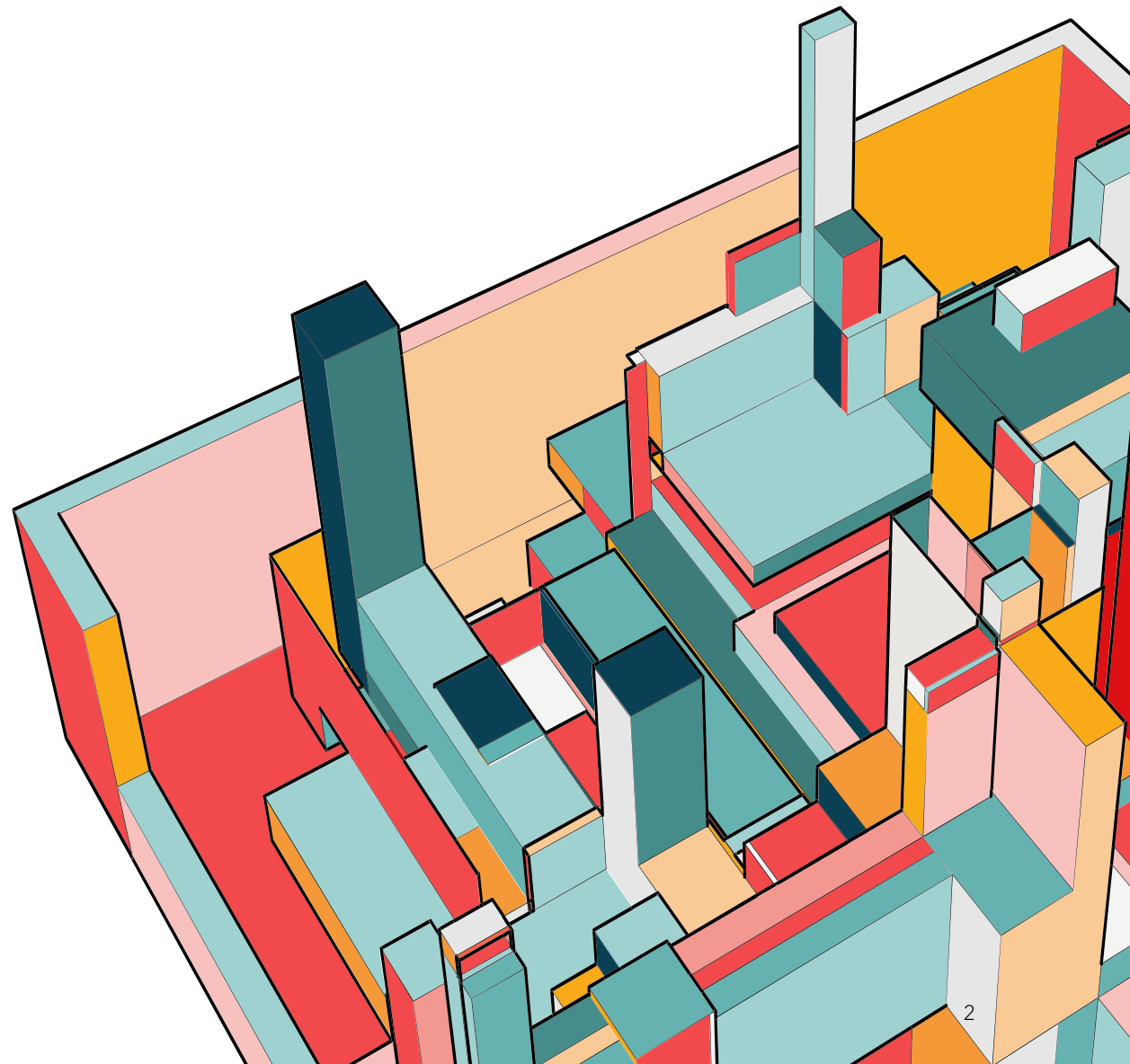


MVC

model view controller

WHAT IS THE CONCEPT OF MVC?

แนวคิดของ MVC นั้นจะใช้หลักการของ OOP ซึ่งแบ่งการทำงานหลักๆให้เป็นรูปแบบของ object โดยที่ MVC นั้นกำหนดชื่อ object มาให้เรียบร้อยแล้วตามชื่อเลยก็คือ model view controller ซึ่งมีกติกาคือ การทำงานของทั้ง 3 object นี้จะแยกการทำงานอย่างชัดเจนห้ามท้าวทำงานกันเด็ดขาด



Model

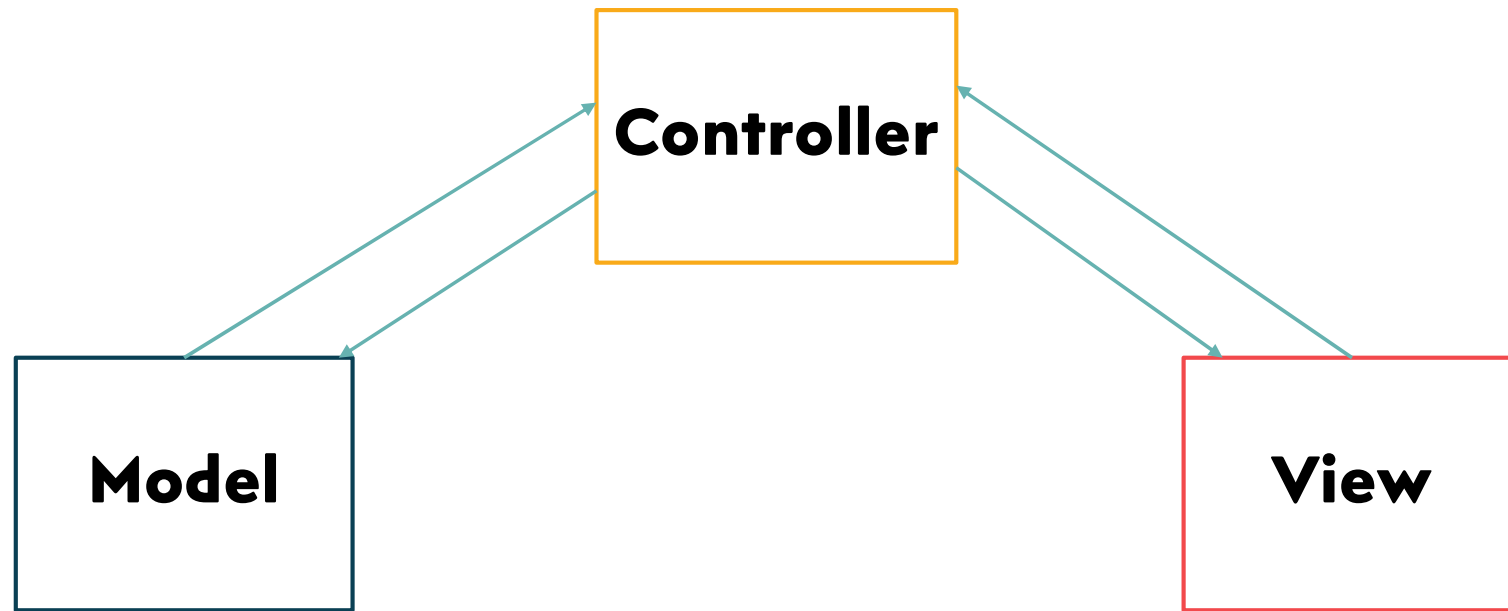
การทำงานของ model จะจัดการส่วนที่ข้อมูลทั้งหมดจะคอยเตรียมข้อมูลที่เหมาะสมไว้ และ model นั้นจะทำงานเมื่อ controller ร้องขอเท่านั้น

View

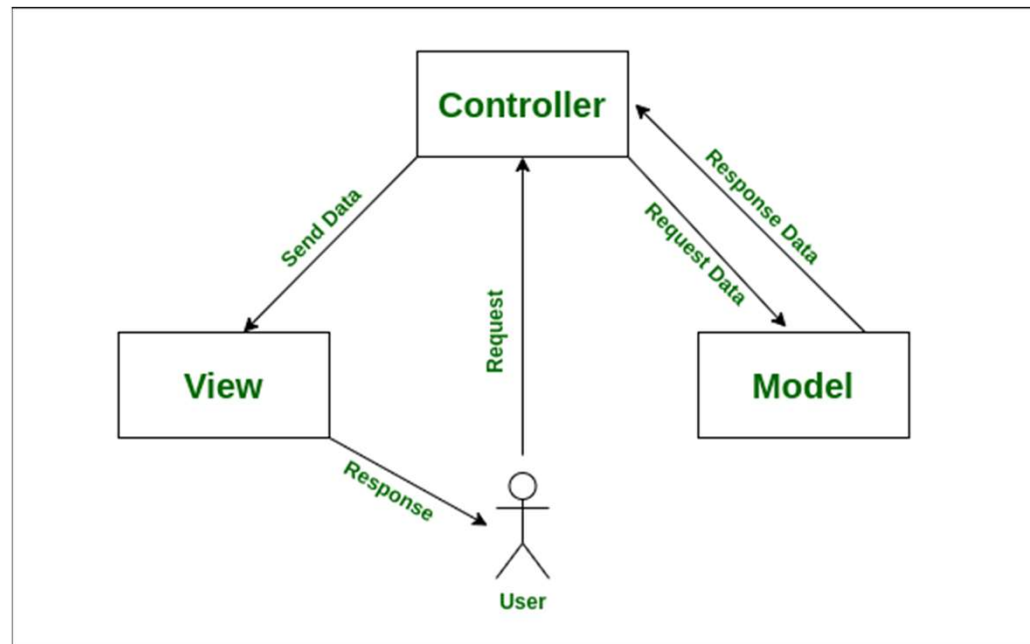
view นั้นจะจัดการส่วนของหน้าต่างทั้งหมด หรือส่วนติดต่อกับผู้ใช้โดยตรง (user interface) โดย view นั้นจะรับคำสั่งการทำงานจาก controller และเป็นตัวกลางให้ผู้ใช้ติดต่อกับ controller อีกด้วย

Controller

controller เปรียบเสมือนกับมันสมองและศูนย์กลางการทำงานทั้งหมด จะเห็นว่าทุกส่วนนั้นจะติดต่อกับ controller ทั้งหมดรอคอยคำสั่งจาก controller นอกจากนี้ controller จะจัดการทำงานในส่วนที่เป็น logic ทั้งหมดในระบบ

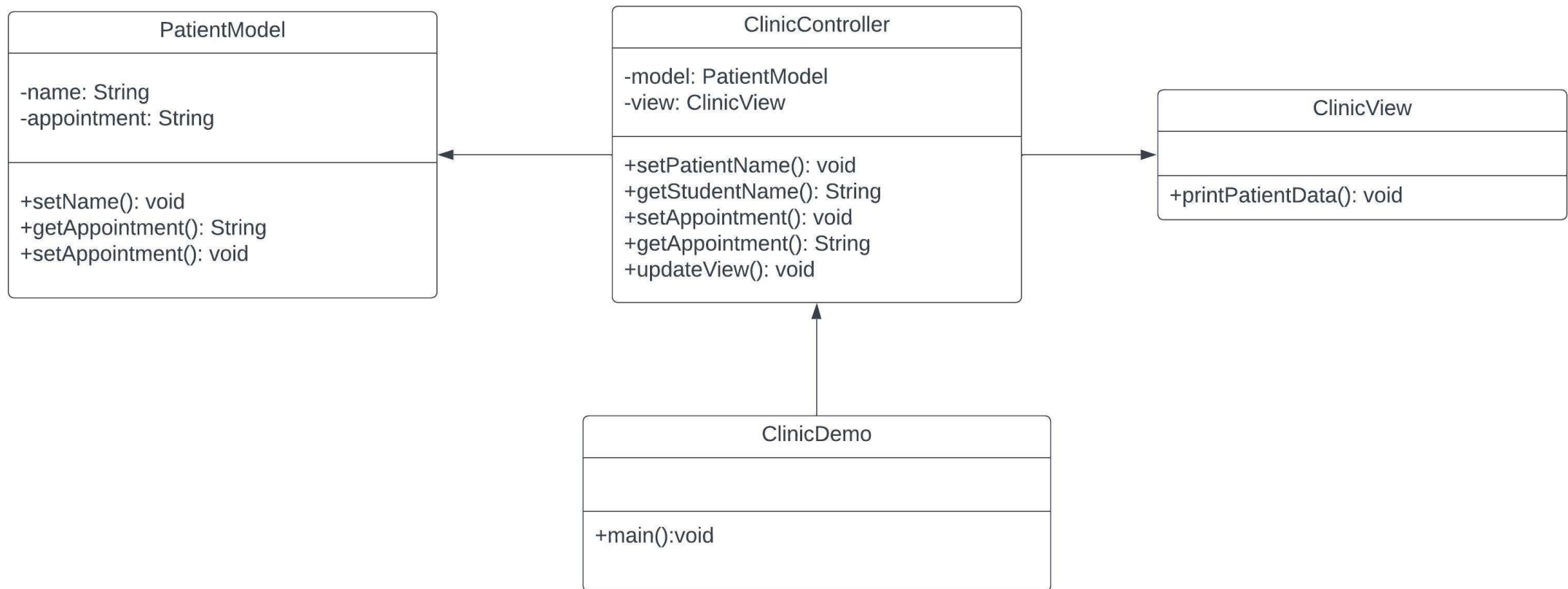


UML DIAGRAM MVC DESIGN PATTERN



ที่มา : <https://www.geeksforgeeks.org/mvc-design-pattern/>

Clinic Class Diagram



Slide 6

as1

Patient appointment

arinchai sangjun, 10/9/2023

```
public class PatientModel
{
    private String name;
    private String appointment;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getAppointment() {
        return appointment;
    }
    public void setAppointment(String appointment) {
        this.appointment = appointment;
    }
}
```

```
public class ClinicView
{
    public void printPatientData(String NamePatient,String Appointment){
        System.out.println("Clinic queue:");
        System.out.println("Name: " + NamePatient);
        System.out.println("Appointment Date: " + Appointment);
    }
}
```



```
public class ClinicController {  
    private PatientModel model;  
    private ClinicView view;  
  
    public ClinicController(PatientModel model, ClinicView view){  
        this.model = model;  
        this.view = view;  
    }  
  
    public void setPatientName(String name){  
        model.setName(name);  
    }  
  
    public String getStudentName(){  
        return model.getName();  
    }  
  
    public void setAppointment(String appointment){  
        model.setAppointment(appointment);  
    }  
  
    public String getAppointment(){  
        return model.getAppointment();  
    }  
  
    public void updateView(){  
        view.printPatientData(model.getName(), model.getAppointment());  
    }  
}
```

```

public class ClinicDemo {
    public static void main(String[] args) {

        PatientModel model = new PatientModel();

        //Create a view : to write student details on console
        ClinicView view = new ClinicView();

        ClinicController controller = new ClinicController(model, view);

        controller.setPatientName("Zom");
        controller.setAppointment("13/Oct/2023");
        controller.updateView();

        //update model data
        controller.setPatientName("Atom");
        controller.setAppointment("14/Nov/2023");
        controller.updateView();
    }
}

```

Options

Clinic queue:
 Name: Zom
 Appointment Date: 13/Oct/2023
 Clinic queue:
 Name: Atom
 Appointment Date: 14/Nov/2023

MVC เหมาะกับงานประเภทไหน

MVC เหมาะกับงานที่มีความซับซ้อนและข้อมูลที่ต้องการการจัดการและการแสดงผลอย่างชัดเจน และมีความสำคัญในการแยกแยะระหว่างส่วนต่าง ๆ ของระบบ เช่น:

- เว็บแอปพลิเคชัน: MVC เป็นโครงสร้างที่นิยมใช้ในการพัฒนาเว็บแอปพลิเคชัน เพราะมีการแยกแยะระหว่าง การจัดการข้อมูล (model) แสดงผล (view) และการควบคุมกระบวนการ (controller) อย่างชัดเจน

CATEGORY

MVC เป็นแนวทางการออกแบบซอฟต์แวร์ที่ไม่ขึ้นกับหมวดหมู่หรือแพลตฟอร์มเฉพาะ เป็นเครื่องมือหรือแนวคิดที่สามารถใช้ในการพัฒนาซอฟต์แวร์ต่าง ๆ อย่างยืดหยุ่น มีการแยกแยะความรับผิดชอบในการพัฒนาแอปพลิเคชันและช่วยจัดโค้ดให้มีระเบียบ และง่ายต่อการบำรุงรักษาในระยะยาว.

