Báo cáo về việc sử dụng design concept and principle để đánh giá thiết kế và đưa ra giải pháp.

Phần mềm được thiết kế theo mô hình MVC, giúp giảm coupling và tăng cohesion (sự kết dính) giữa các thành phần của ứng dụng.

Hầu hết các lớp đều thực hiện được tính Single Responsibility Principle.

Một số đối tượng chỉ có 1 trong một phiên làm việc trong Phần mềm (Cart, AIMSDB, LOGGER) đã được thiết kế theo Singleton design pattern.

Các lớp DTO (OrderDTO) giúp tương tác với cơ sở dữ liệu, tách biệt rõ ràng giữa logic của cơ sở dữ liệu và logic của đối tượng Order.

Thiết kế có thể chưa bao quát được một số trường hợp lỗi (kiểm tra đối tượng null, kiểm tra trường dữ liệu người dùng nhập).

Các hàm getAllMedia(), getMediaById(), updateMediaFieldById() vi phạm nguyên tắc OCP.

> Bởi vì khi Sqlite thay đổi cách tổ chức dữ liệu, hoặc khi muốn lấy entity khác như book hay cd, hoặc là cần sửa đổi hoặc thêm mới các hàm get, update, delete thì lại cần phải sửa dổi code ở trong class Media.
>
> Cách sửa lại:
>
> - Cách 1: Định nghĩa một interface có các hàm get, update, delete,... để đại diện cho quy trình lấy dữ liệu từ cơ sở dữ liệu mà không cần biết chi tiết về
>
> cách nó được thực hiện. Sau đó implement nó ở trong các class con.
>
> - Cách 2: Chuyển Media thành abtract class sau đó để các hàm tương tác với Database là các abtract function để lớp con override lại

Các phương thức của lớp Utils khá độc lập, không liên kết theo tính nghiệp vụ, tuy nhiên không có cách giải quyết nào do lớp Utils cung cấp các hàm bổ trợ cho toàn bộ phần mềm.

Các lớp con DVD, CD, Book kế thừa từ lớp Media và ghi đè phương thức getAllMedia(), nhưng không thức hiện trả về đối tượng thỏa mãn. Cần sửa lại hoặc xóa bỏ các phương thức này ở lớp con.

Các phương thức sắp xếp theo Loại sản phẩm, Giá cả trong lớp điều khiển HomeController có thể thực hiện ngay trong lớp thực thể Media, để sau này có thể thực hiện sắp xếp tại các lớp điều khiển khác mà không cần lặp lại mã nguồn