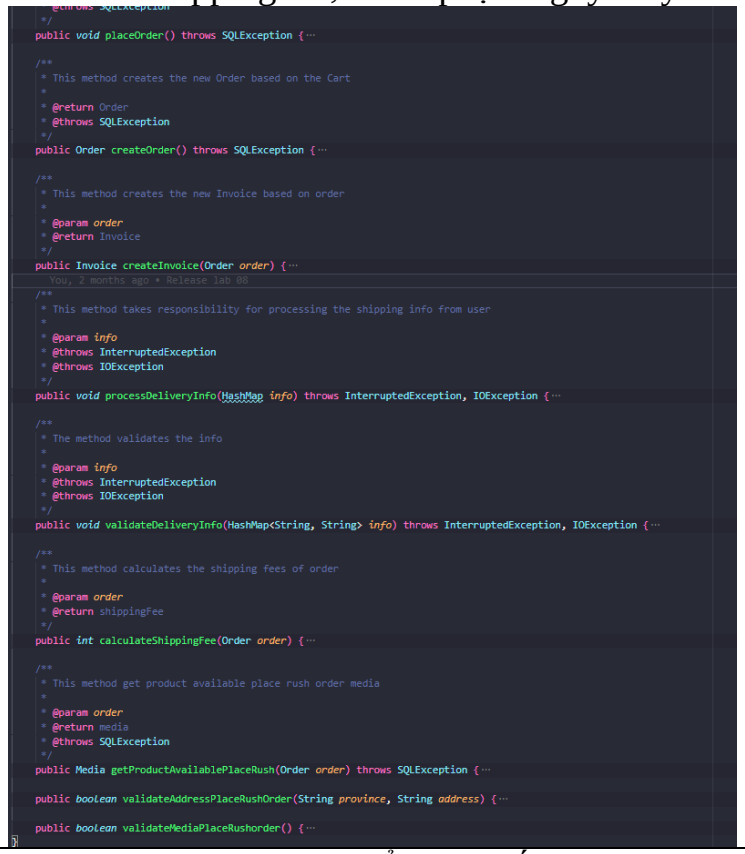
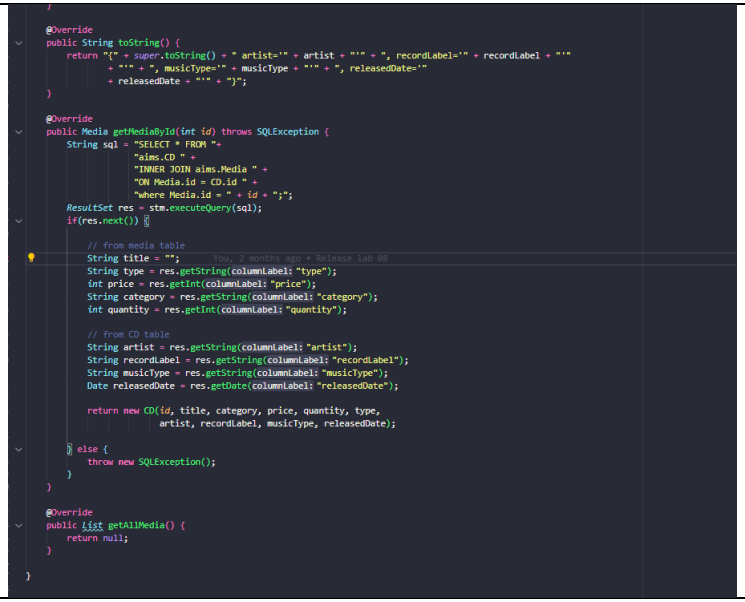
DESIGN PRINCIPLES

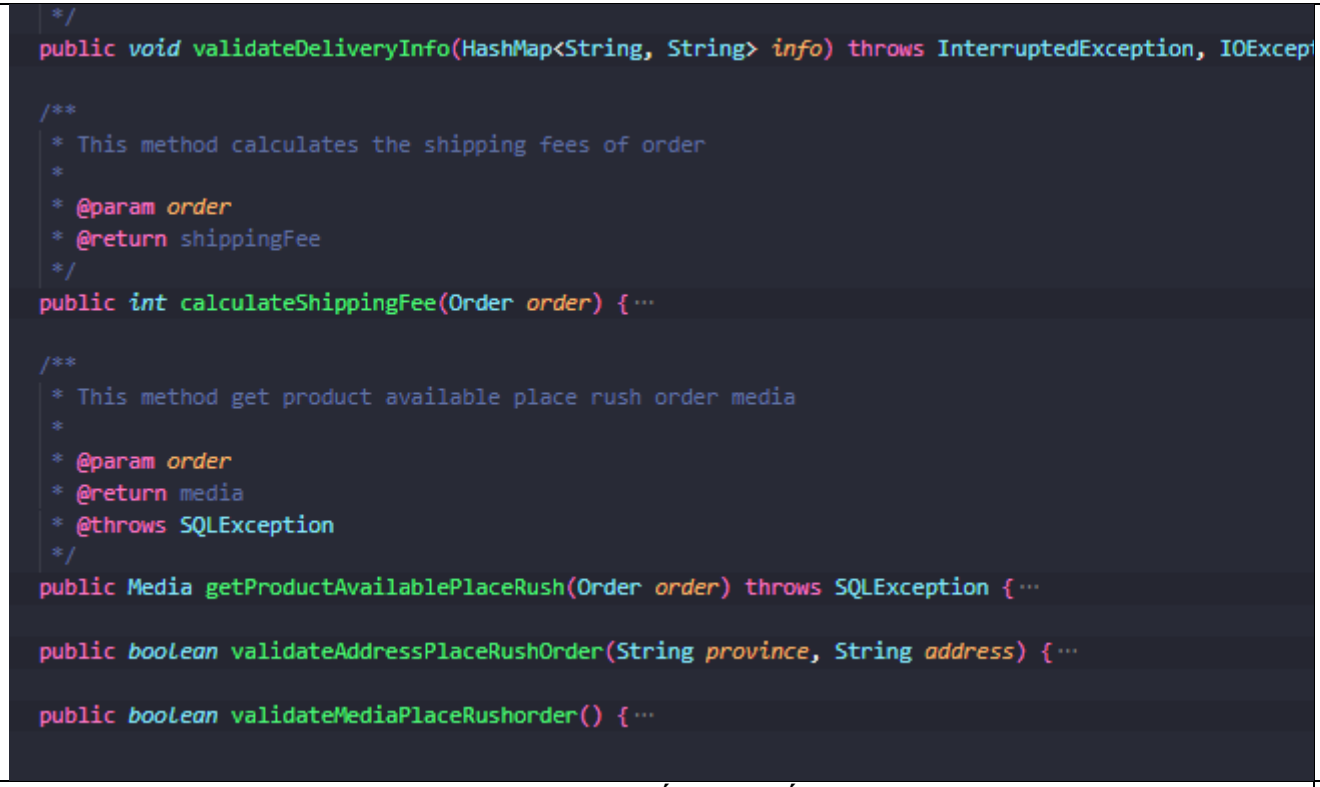
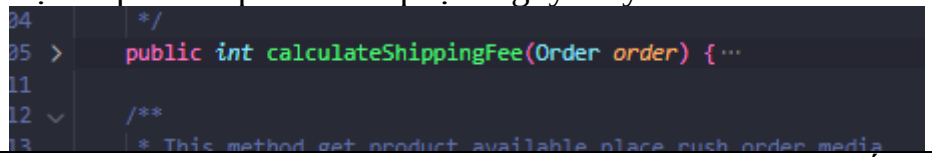
1. SINGLE RESPONSIBILITY PRINCIPLE

#	Related modules	Description	Improvement
1.1	Class PlaceOrderController	Class PlaceOrderController phải thực hiện cả việc createOrder, validateInfo và calculateShippingFee, => vi phạm nguyên lý S 	Tách các task này thành các lớp khác, trong lớp PlaceOrderController chỉ cần gọi đến các lớp này.
1.2	Các class entity	Có cả các phương thức để truy xuất vào database để lấy dữ liệu => vi phạm nguyên lý S	Tách phần truy xuất vào database thành một class riêng.

2. OPEN/CLOSED PRINCIPLE

#	Related modules	Description	Improvement
2.1	Class PlaceOrderController	Phương thức validateInfo có nhiều phương thức cấp thấp hơn là validateAddress, validateName, validatePhone, như vậy khi cần bổ sung thêm phương thức validate mới thì phải sửa vào trong lớp cũ => vi phạm nguyên lý O	Xây dựng một interface validate và các lớp thực thi validate cho từng thông tin, như vậy sau này muốn bổ sung validate cho một thông tin khác, chúng ta chỉ cần xây dựng thêm một lớp khác thực thi interface validate mà ko

			cần sửa vào những lớp khác.
2.2	Class PlaceOrderController	Phương thức calculateShippingFee được viết trực tiếp trong lớp PlaceOrderController, như vậy sau này khi thay đổi cách tính phí ship, sẽ phải sửa trực tiếp vào lớp cũ => vi phạm nguyên lý O 	Xây dựng một interface calculate và một lớp thực thi việc calculate shipping fee.
2.3	Class CartScreenHandler	Trong phương thức requestToPlaceOrder, sử dụng trực tiếp đến PlaceOrderController dẫn đến việc thêm các phương thức để thực thi cho chức năng PlaceRushOrder gặp khó khăn, phải thay đổi trong lớp cũ.	Tách thành một lớp khác thực hiện việc requestToPlaceOrder, khi có thêm yêu cầu RushOrder thì tạo

			lớp khác là extend lớp cũ.

3. LISKOV SUBSTITUTION PRINCIPLE

#	Related modules	Description	Improvement
3,1			

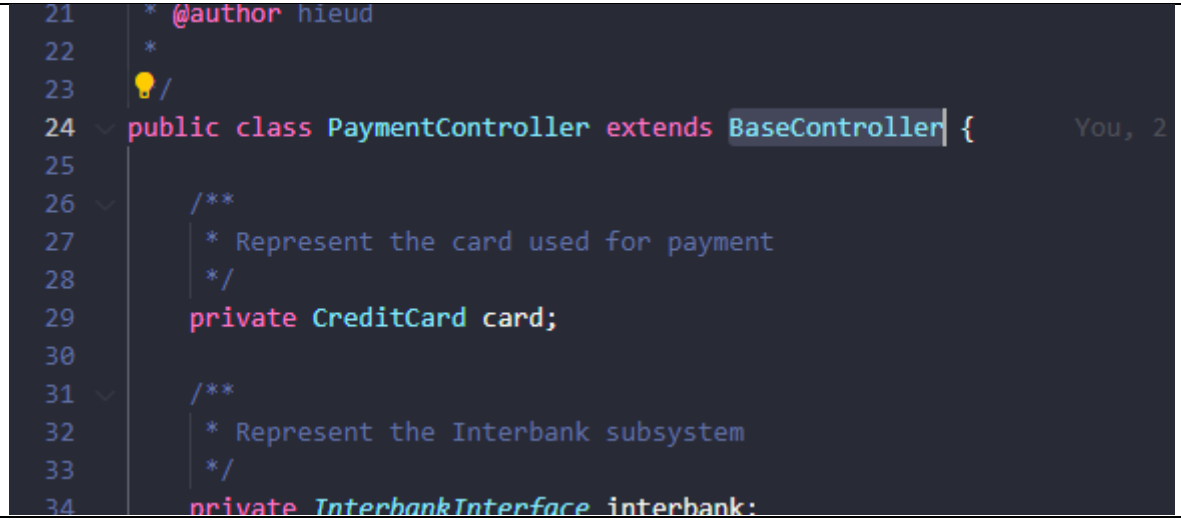
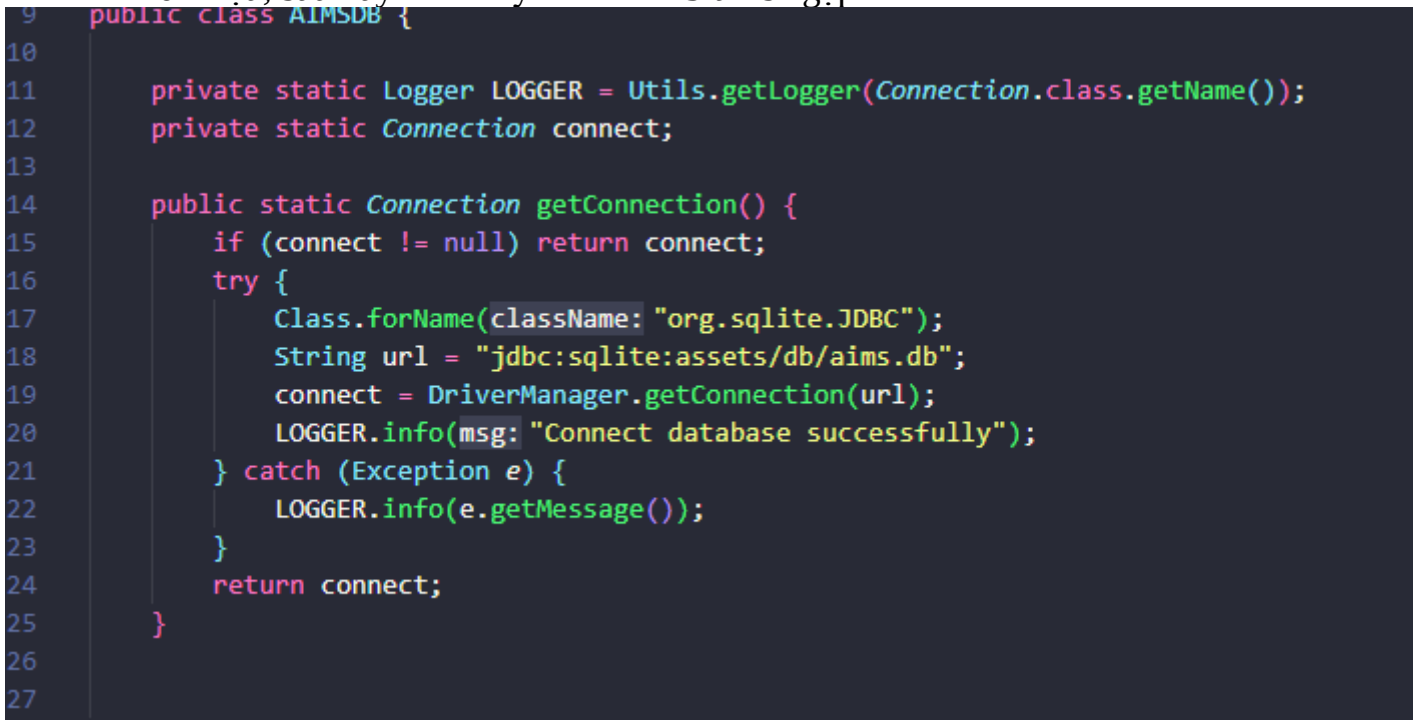
4. INTERFACE SEGREGATION PRINCIPLE

#	Related modules	Description	Improvement
4.1			

5. DEPENDENCY INVERSION PRINCIPLE

#	Related modules	Description	Improvement
5.1	Class CartScreenHandler	Tương tự như 2.3, trong phương thức requestToPlaceOrder đang fix cứng là gọi đến PlaceOrderController, nên khi muốn PlaceRushOrder khó khăn trong việc thay đổi	Giải pháp tương tự 2.3

			
5.2	Class PaymentController	Phụ thuộc trực vào class CreditCard, vì vậy khi thay đổi phương thức thanh toán cần phải sửa trực tiếp vào trong lớp này	

			
5.3	Class AIMSDB	Trong phương thức getConnection đang fix cứng là sử dụng sqlite, vì vậy đang bị phụ thuộc chi tiết vào cơ sở dữ liệu, sau này nếu thay đổi DBMS thì sẽ gặp khó khăn 	Xây dựng interface có các phương thức chung để truy xuất vào cơ sở dữ liệu, sau đó tùy vào việc sử dụng loại DBMS nào thì sẽ xây dựng các lớp để thực thi

			interface này.