

## DESIGN CONCEPTS AIMS

### 1. Coupling

#### 1.1. Content Coupling

Related modules	Description	Improvement
<b>Order.java</b>	Trong tương lai, có thể xuất hiện một module mới mà nó gọi đến phương thức <code>getDeliveryInfo()</code> để quản lý <code>deliveryInfo</code> thì module ấy có thể thay đổi trực tiếp vào đối tượng đó bằng phương thức <code>put</code> của <code>HashMap</code>	Xóa bỏ phương thức <code>getDeliveryInfo()</code> <pre>public HashMap getDeliveryInfo() {     return deliveryInfo; }</pre> và thay thế bằng phương thức <code>setDeliveryInfoField()</code> <pre>public String setDeliveryInfoField(String key, String value) {     return this.deliveryInfo.put(key, value); }</pre>

#### 1.2. Common Coupling

Related modules	Description	Improvement
<b>Utils.java</b>	<pre>public static DateFormat DATE_FORMATER = new SimpleDateFormat(pattern: "yyyy/MM/dd HH:mm:ss");</pre> Biến static trên được dùng ở nhiều operation => có thể thay đổi dễ dàng	Đặt tên biến thành final

#### 1.3. Control Coupling

Project không tồn tại loại coupling này.

#### 1.4. Stamp Coupling

Related modules	Description	Improvement
<b>Cart.java</b>	<pre>public CartMedia checkMediaInCart(Media media){     for (CartMedia cartMedia : lstCartMedia) {         if (cartMedia.getMedia().getId() == media.getId()) return cartMedia;     }     return null; }</pre> Chỉ sử dụng id media nhưng truyền cả media vào	Chỉ cần truyền media id
<b>PlaceOrderController.java</b>	<pre>/**  * The method validates the info  */ @param info @throws InterruptedException @throws IOException public void validateDeliveryInfo(HashMap&lt;String, String&gt; info) throws InterruptedException, IOException {     if (validateAddress(info.get(key: "address"))         &amp;&amp; validateName(info.get(key: "name"))         &amp;&amp; validatePhoneNumber(info.get(key: "phone"))) {         throw new InvalidDeliveryInfoException(message: "Invalid delivery info");     } }</pre> Tương tự, truyền thừa các tham số không cần thiết	Chỉ cần truyền vào 3 trường

#### 1.5. Data Coupling


Related modules	Description	Improvement
<b>PaymentController.java</b>	<pre>/**  * Pay order, and then return the result with a message.  *  * @param amount the amount to pay  * @param contents the transaction contents  * @param cardNumber the card number  * @param cardholderName the card holder name  * @param expirationDate the expiration date in the format "MM/YY"  * @param securityCode the security code of the credit card  * @return String (success/fail) may represent the payment result with a message  */ public Map&lt;String, String&gt; payOrder(int amount, String contents, String cardNumber, String cardholderName,</pre> Bản thiết kế tốt	none

## 2. Cohesion

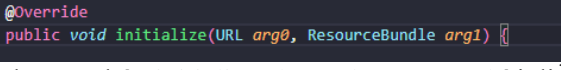
### 2.1. Coincidental Cohesion

Related modules	Description	Improvement
<b>ApplicationProgrammingInterface.java</b>	 trường DATE_FOMATER được khai báo, nhưng không được sử dụng	Xóa

### 2.2. Logical Cohesion

Related modules	Description	Improvement
<b>PlaceOrderController.java</b>	 Đều validate dữ liệu nhưng mà khác cách xử lý	Tách chúng ra các class riêng và chỉ có tác dụng validate dữ liệu.

### 2.3. Temporal Cohesion

Related modules	Description	Improvement
<b>HomeScreenHandler.java</b>	 Phương thức initialize chỉ sử dụng trong thời điểm khởi tạo đối tượng	NONE

### 2.4. Procedural Cohesion

Related modules	Description	Improvement
<b>PlaceOrderController.java</b>	 3 phương thức validate chỉ gắn kết với nhau về mặt thứ tự	

### 2.5. Communicational Cohesion

Related modules	Description	Improvement
<b>InterbankSubsystemController.java</b>	Cả module cùng trả về kiểu dữ liệu là PaymentTransaction	

### 2.6. Sequential Cohesion

Related modules	Description	Improvement

**ApplicationProgramming  
Interface.java**

```
/**
 * @param url
 * @param requestMethod
 * @param token
 * @return HttpURLConnection
 * @throws IOException
 */
private static HttpURLConnection generateConnection(String url, String requestMeth
final var conn = (HttpURLConnection) extracted(url).openConnection();
conn.setDoInput(true);
conn.setDoOutput(true);
conn.setRequestMethod(requestMethod);
conn.setRequestProperty("Content-Type", "application/json");
if (requestMethod.equals("GET") && token != null) {
    conn.setRequestProperty("Authorization", "Bearer " + token);
}
return conn;
}
```

Đầu ra của phương thức này là đầu ra của các phương thức khác trong module

**2.7. Functional Cohesion**

Related modules	Description	Improve ment
<b>ApplicationProgrammingI nterface.java</b>	<pre>/**  * @param url  * @param requestMethod  * @param token  * @return HttpURLConnection  * @throws IOException  */ private static HttpURLConnection generateConnection(String url, String requestMethod, final var conn = (HttpURLConnection) extracted(url).openConnection(); conn.setDoInput(true); conn.setDoOutput(true); conn.setRequestMethod(requestMethod); conn.setRequestProperty("Content-Type", "application/json"); if (requestMethod.equals("GET") &amp;&amp; token != null) {     conn.setRequestProperty("Authorization", "Bearer " + token); } return conn; }</pre> <p>Đầu ra của phương thức này chính là đầu vào của các phương thức khác, và nó là cần thiết cho chức năng của module</p>	