

- RentBikeController: Vi phạm nguyên lý S
 - o Method không liên quan đến nhau -> có thể chia thành các module BarcodeController, InvoiceController.
 - o Các method validate không nên đặt trong lớp này -> Đặt sang lớp Utils chuyên về validate
- EntityClass: Vi phạm nguyên lý S
 - o Không nên có các phương thức lấy dữ liệu từ database trong lớp entity -> Thêm các lớp DAO để lấy dữ liệu từ database
- InterBankSubsystem: Vi phạm nguyên lý O
 - o Khi thanh toán, phương thức payDeposit và refund đều được truyền CreditCard, muốn thêm phương thức thanh toán khác sẽ phải sửa mã nguồn -> Thay bằng một lớp Abstract.

```

public Map<String, String> payDeposit(int amount, String contents, String cardNumber, String ca
    String expirationDate, String securityCode) {
    Map<String, String> result = new Hashtable<String, String>();
    result.put("RESULT", "PAYMENT FAILED!");
    try {
        this.card = new CreditCard(cardNumber, cardHolderName, Integer.parseInt(securityCode),
            getExpirationDate(expirationDate));

        this.interbank = new InterbankSubsystem();
        PaymentTransaction transaction = interbank.payDeposit(card, amount, contents);

        result.put("RESULT", "PAYMENT SUCCESSFUL!");
        result.put("MESSAGE", "You have succesffully paid the order!");
    } catch (PaymentException | UnrecognizedException ex) {
        result.put("MESSAGE", ex.getMessage());
    }
    return result;
}
}

```