**CODE REVIEW REPORT**

**Project: Circle-K Grocery Store App**

**Date: 15/4/2023**

**Reviewer: Nguyễn Hữu Tín, Lê Huỳnh Phát**

**INTRODUCE**

Mục đích của bài reivew này nhằm cung cấp những thông tin cơ bản về quá trình thực hiện và xây dựng Ứng dụng Quản lí cửa hàng tiện lợi Circle-K. Ứng dụng được xây dựng dựa trên mô hình 3 lớp: BUS, DAO và DTO.

**FINDINGS:**

1. **Quy chuẩn code**

* Sử dụng chữ cái **Viết Hoa**đúng là chìa khóa để tuân theo quy ước đặt tên:
* **Lowercase**: Là tên mà tất cả các chữ cái trong một từ được viết mà không Viết Hoa (ví dụ: while, if, mypackage).
* **Uppercase**: Là tên trong đó tất cả các chữ cái trong một từ được viết bằng chữ HOA. Khi có nhiều hơn hai từ trong tên, hãy sử dụng dấu gạch dưới để phân tách chúng (ví dụ: MAX\_HOURS, FIRST\_DAY\_OF\_WEEK).
* **CamelCase**: (còn được gọi là Upper CamelCase) là tên mà mỗi từ mới bắt đầu bằng chữ in hoa (ví dụ: CamelCase, CustomerAccount, PlayingCard).
* **Mixed case**: (còn được gọi là Lower CamelCase) giống như CamelCase ngoại trừ chữ cái đầu tiên của tên được viết bằng chữ thường (ví dụ: hasChildren, customerFirstName, customerLastName).Danh
* sách dưới đây cho bạn thấy các **Quy ước đặt tên tiêu chuẩn** cho từng loại định danh:
* **Quy ước đặt tên Packages**
* Tên **packages** nên được **viết thường**. Với các dự án nhỏ chỉ có một vài package, bạn chỉ cần đặt cho chúng những cái tên đơn giản (nhưng có ý nghĩa!)\

* Nhưng trong các công ty phần mềm và các dự án lớn nơi các package có thể được nhập vào các package khác, các tên thường sẽ được chia nhỏ.  
    
  Thông thường, điều này sẽ bắt đầu với domain công ty trước khi được chia thành các lớp hoặc tính năng:
* **Quy ước đặt tên Class**
* Áp dụng **CamelCase**. Sử dụng **danh từ** để đặt tên cho Class

class Customer class Account

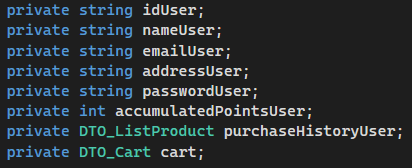
* **Quy ước đặt tên Interfaces**
* **Tên Interfaces trong project** phải có áp dụng CamelCase. Chúng có xu hướng có một tên có thể mô tả một hoạt động mà một class có thể làm:
* **Quy ước đặt tên Phương thức (Method)**
* **Tên Method trong project** nên được áp dụng quy tắc **Mixed case**. Và nên sử dụng các động từ để mô tả những gì phương thức sẽ làm:

void calculateTax() string getSurname()

**Quy tắc đặt tên biến (Variables)**

* **Tên biến** tuân theo quy tắc **Mixed case**. Các tên biến sẽ đại diện cho những gì giá trị của biến đại diện:

**Quy chuẩn đặt tên biến**

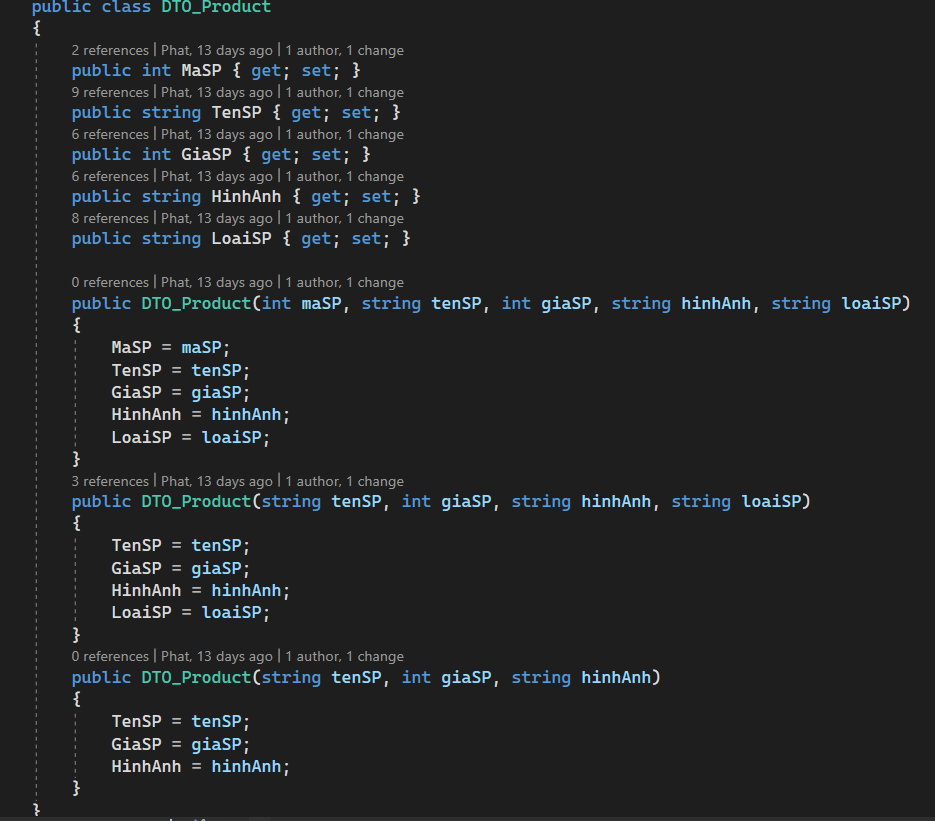
* ****
* Tên biến đặt bằng việc kết hợp từ các từ tiếng anh, viết liền vd: idUser, emailUser, …
* Chữ cái đầu viết hoa, bắt đầu từ thứ hai, các từ còn lại có chữ cái đầu viết hoa.

1. **Lớp DTO**

Trong lớp DTO, DTO là thư viện được xây dựng nhằm chứa các lớp được sử dụng để xây dựng các đối tượng. Trong ứng dụng này, lớp dto sẽ bao gồm các lớp sau: DTO\_Cart, DTO\_Product, DTO\_ListProduct, DTO\_Voucher, DTO\_ListVoucher.

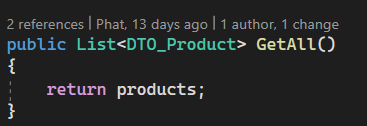
1. DTO\_Product

* Lớp này để xây dựng DTO\_Product, được sử dụng để lưu trữ dữ liệu về các sản phẩm trong cửa hàng.
* Lớp này bao gồm các thuộc tính: nameProduct (string), priceProduct(int), typeProduct(string),..



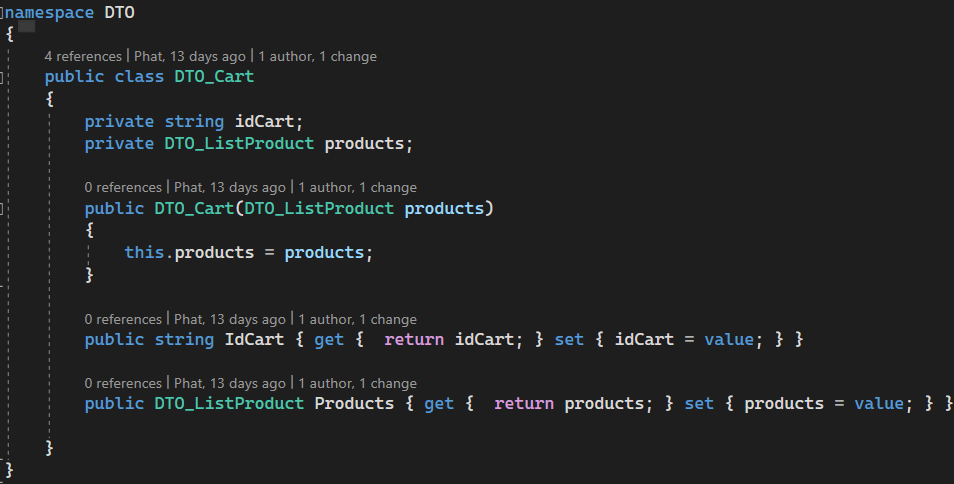
1. DTO\_ListProduct

* Lớp này được xây dựng để lưu trữ các danh sách sản phẩm được phân loại theo loại sản phẩm
* Lớp này bao gồm các thuộc tính: List<product> product
* Lớp này gồm phương thức:
  + List<product> getALL(): được dùng để lấy tất cả các sản phẩm lưu trong cơ sở dữ liệu



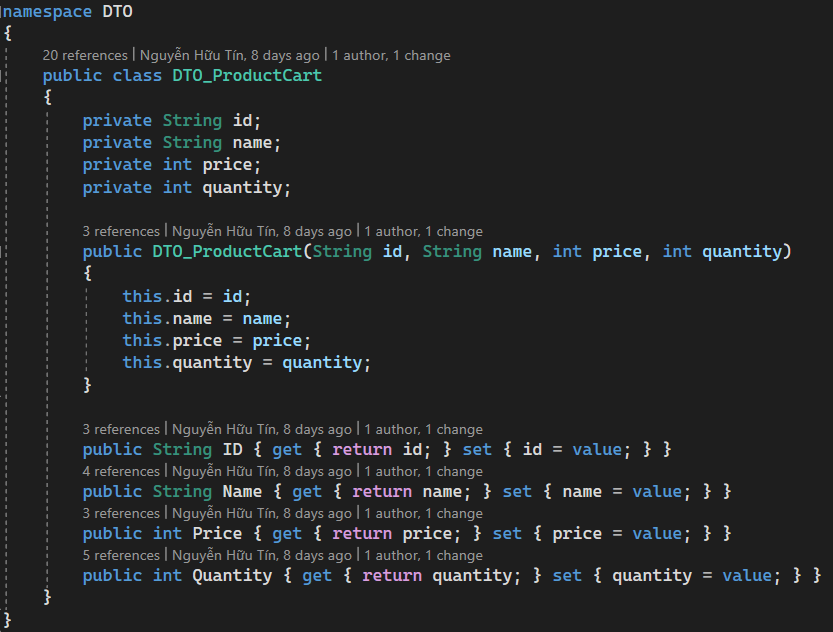
1. DTO\_Cart

* Lớp này được xây dựng để lưu trữ các sản phẩm được người dùng chọn với mục đích lưu trữ hoặc mua hàng.
* Lớp này bao gồm các thuộc tính: idCart(string), products (DTO\_ListProduct)



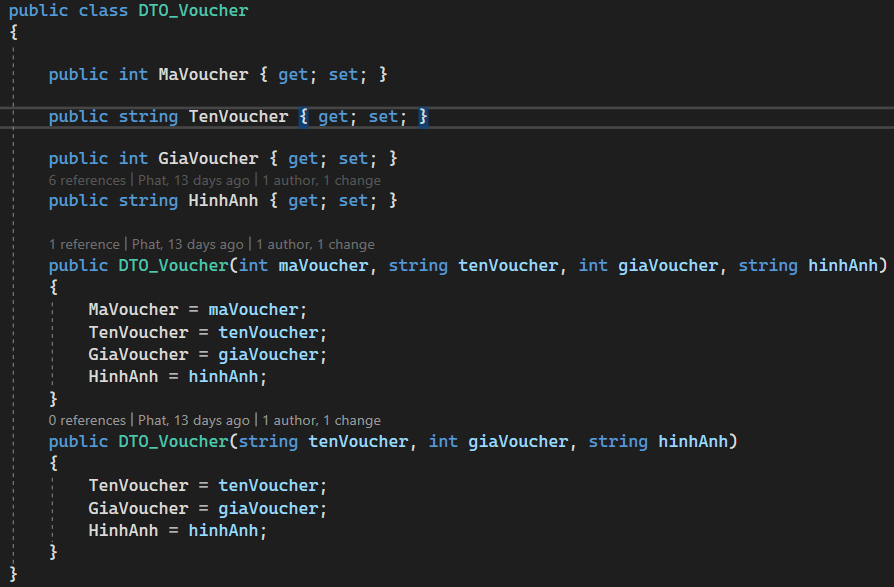
1. DTO\_ProductCart

* Lớp này được xây dựng để lưu trữ các sản phẩm được người dùng chọn với mục đích lưu trữ vào trong giỏ hàng của người dùng.
* Lớp này bao gồm các thuộc tính: id(string), name(string), price(int), quatity(int)



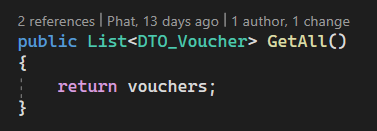
1. DTO\_Voucher

* Lớp này được xây dựng để lưu trữ các voucher được cung cấp, người dùng có thể sử dụng điểm tích lũy để đổi lấy voucher.
* Lớp này bao gồm các thuộc tính: maVoucher(string), tenVoucher(string), giaVoucher(int), hinhAnh(string).



1. DTO\_ListVoucher

* Lớp này được xây dựng để lưu trữ các voucher được cung cấp, người dùng có thể sử dụng điểm tích lũy để đổi lấy voucher.
* Lớp này bao gồm các thuộc tính: List<Voucher> voucher chứa danh sách voucher.
* Lớp này có phương thức getAll() để lấy về danh sách tất cả các voucher.



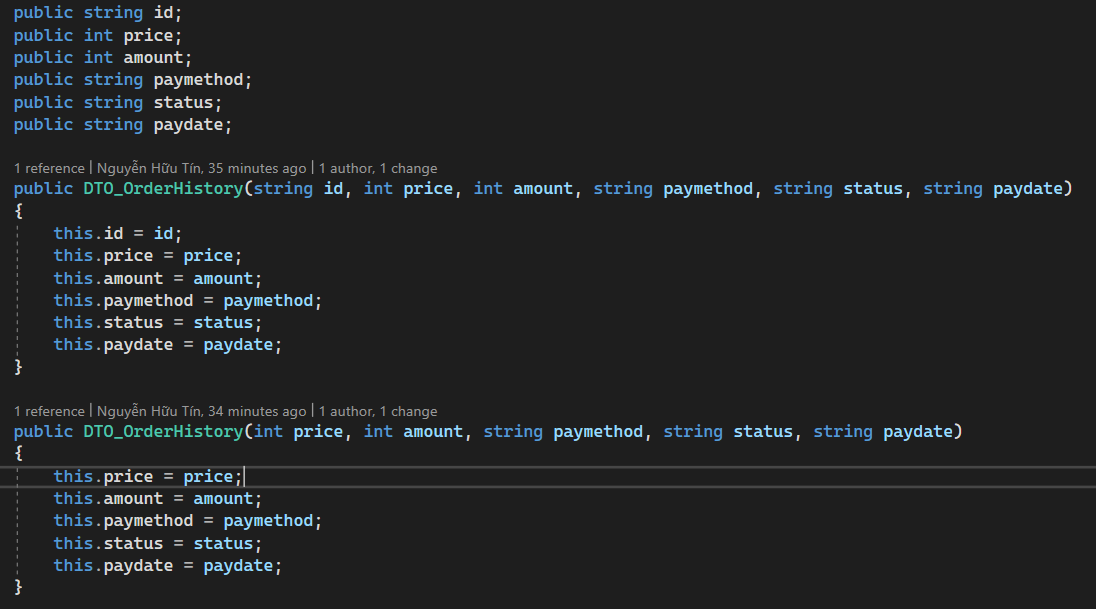
1. DTO\_MyVoucher

* Lớp này được xây dựng để lưu trữ các voucher mà người dùng dùng điểm để đổi.
* Lớp này bao gồm các thuộc tính: maMyVoucher(string), tenMyVoucher(string), giaMyVoucher(int), hinhAnh(string), quatity(int).



1. DTO\_OrderHistory

* Lớp này được xây dựng để lưu trữ các lịch sử sau khi người dùng thanh toán
* Lớp này chứa các thuộc tính: id (string), price (int), amount(int), paymethod (string), status (string), paydate (string).



1. **Lớp DAO**

Trong lớp DAO, DAO là thư viện được xây dựng nhằm truy xuất cơ sở dữ liệu, liên kết giữa cơ sở dữ liệu với giao diện người dùng, hiển thị sản phẩm, giỏ hàng, lịch sử mua hàng từ trong cơ sở dữ liệu lên giao diện người dùng.

* Lớp này bao gồm các lớp: DAO\_Cart, DAO\_Product, DAO\_ListProduct, DAO\_Voucher, DAO\_ListVoucher, DAO\_User

1. DAO\_Product

* Lớp này được xây dựng để truy xuất cơ sở dữ liệu, lấy các dữ liệu liên quan đến sản phẩm được lưu lại của cửa hàng.
* Lớp này gồm các phương thức:

1. DAO\_ListProduct

* Lớp này được xây dựng để truy xuất cơ sở dữ liệu, lấy dữ liệu liên quan đến danh sách các sản phẩm được phân loại theo loại sản phầm hoặc theo một từ hay cụm từ theo tên của sản phẩm.
* Lớp này bao gồm các phương thức:

1. DAO\_Cart
2. **Lớp BUS**

Trong lớp BUS, BUS là thư viện được xây dựng để gọi lại các phương thức của lớp DAO, qua đó gián tiếp hiển thị lên giao diện người dùng, lớp này sẽ được lớp giao diện gọi lại để lấy các thông tin, dữ liệu hiển thị lên giai diện.

* Lớp này bao gồm các lớp: BUS\_Product, BUS\_ListProduct, BUS\_Cart,..

1. BUS\_Product
2. **Lớp giao diện người dùng**

Trong lớp giao diện: Chứa các trang giao diện chính của ứng dụng: trang đăng nhập, đăng kí, trang chủ và trang admin, bên cạnh đó còn có các control được xây dựng để phục vụ cho việc hiển thị của trang chủ.

1. Trang đăng nhập
2. Trang đăng ký
3. Trang chủ
4. Trang admin

**CONCLUSION:**