**HỆ THỐNG QUẢN LÝ BÁN HÀNG CHO ONLINE SHOP**

**I. Mục tiêu:**

Xây dựng một hệ thống thương mại điện tử được thiết kế riêng cho một cửa hàng cá nhân, cho phép chủ cửa hàng quản lý và bán sản phẩm trực tuyến một cách hiệu quả. Hệ thống sẽ cung cấp các tính năng như quản lý danh mục sản phẩm, kiểm soát hang tồn kho, theo dõi các đơn hàng, kiểm soát được các khách hàng cũng như những người giao hàng. Ngoài ra, khách hàng có thể xem thông tin chi tiết về sản phẩm, từ hình ảnh đến giá cả của sản phẩm, và có thể đặt đơn dựa theo số dư còn lại trong tài khoản. Khách hàng cũng có thể kiểm soát số dư trong tài khoản của mình cũng như có thể xem được các giao dịch đã thực hiện. Ngoài ra, khách hàng cũng có thể hoàn trả lại sản phẩm nếu như sản phẩm bị lỗi hoặc không vừa ý. Cuối cùng là lớp người giao hàng có thể thực hiện chức năng giao hàng cũng như quản lý tài khoản của chính mình.

**II. Luồng hoạt động của ứng dụng:**

Khi truy cập vào ứng dụng, người dùng sẽ được lựa chọn đăng nhập với một trong ba vai trò: Admin, User, hoặc Supplier. Với vai trò Admin, người dùng có thể quản lý danh mục sản phẩm bằng cách thêm, chỉnh sửa hoặc xóa danh mục, sau đó gắn danh mục vào các sản phẩm vừa tạo. Admin cũng có thể quản lý sản phẩm, chỉnh sửa thông tin của các tài khoản người dùng đã đăng ký, và thêm mới cũng như quản lý các Supplier trên hệ thống. Một trong những nhiệm vụ quan trọng của Admin là phân công vai trò cho Supplier, đảm bảo các đơn hàng do User thực hiện được giao đúng nhà cung cấp. Ngoài ra, Admin có thể theo dõi toàn bộ giao dịch của hệ thống, bao gồm đơn hàng đã hoàn thành, đơn hàng bị hoàn trả và thống kê doanh thu.

Đối với Supplier, vai trò chính là thực hiện giao hàng sau khi được Admin phân công. Supplier có thể theo dõi danh sách các giao dịch mà họ đã thực hiện, đảm bảo rằng tất cả các đơn hàng được xử lý đúng thời gian. Supplier cũng có thể quản lý thông tin tài khoản cá nhân, bao gồm việc cập nhật các chi tiết cơ bản để đảm bảo tính chính xác trong quá trình hoạt động.

Cuối cùng, User có vai trò thực hiện các giao dịch mua hàng. Khi đăng ký tài khoản, mỗi User sẽ được liên kết với một tài khoản số dư (User Balance) để quản lý nguồn tiền. Người dùng có thể nạp tiền vào tài khoản để thực hiện giao dịch mua sắm. Trong UserDashboard, họ có thể xem các sản phẩm trên hệ thống cùng với thông tin chi tiết như số lượng, giá cả, và hình ảnh. Sau khi thêm sản phẩm vào giỏ hàng, nếu đủ số dư, User có thể hoàn tất việc mua hàng, tạo đơn hàng chờ Admin xử lý. Khi đơn hàng được giao bởi Supplier, User có quyền yêu cầu hoàn trả (Refund) và nhận lại số tiền đã chi tiêu. Ngoài ra, User có thể quản lý thông tin tài khoản cá nhân như cập nhật tên và mật khẩu để đảm bảo bảo mật.

**Tóm lại một chu kì hoạt động chính của ứng dụng sẽ như sau:**

Trước tiên đăng nhập bằng admin và tạo ra các Category cho Product, sau đó thêm các Product sử dụng chính các Category vừa tạo. Tiếp theo hãy thêm các Supplier mà trước đó đã đăng ký riêng với admin vào ứng dụng, sử dụng chức năng Add Supplier. Tiếp theo một khách hàng (User) đăng ký tài khoản trên hệ thống, sử dụng email và mật khẩu để đăng nhập. Khách hàng sau đó có thể nạp thêm số dư vào tài khoản. Tiếp theo khách hàng lựa chọn các mặt hàng để thêm vào giỏ hàng, nếu đủ số dư có thể tiến hành thanh toán và tạo một đơn hàng. Tiếp theo admin sẽ lựa chọn các Supplier để gán vào đơn hàng vừa tạo. Tiếp theo các Supplier sẽ nhận được thông báo rằng họ đã được gán cho một giao dịch và sẽ thực hiện chức năng Deliver để giao hàng. Quay lại lớp User nhận thấy đơn hàng đã được giao thành công, User có thể lựa chọn chức năng Refund để hoàn trả lại sản phẩm cũng như lấy lại số tiền của mình. Cuối cùng là lớp Admin sẽ tổng kết được toàn bộ giao dịch hay những đơn hàng đã được thực thi.

**III. Phân tích thiết kế lớp**

**1. User**

- Mục đích: Đại diện cho một khách hàng trong một hệ thống.

- Thuộc tính:

userId: Định danh duy nhất cho mỗi người dùng.

username, email, password: Thông tin cơ bản cần thiết để đăng nhập và tương tác với hệ thống.

phoneNumber: Số điện thoại của người dùng, dùng để liên hệ hoặc xác minh danh tính.

securityQuestion, securityQuestionAns: Câu hỏi bảo mật và câu trả lời, giúp xác thực tài khoản khi cần khôi phục thông tin.

address: Địa chỉ của người dùng, có thể được sử dụng trong việc giao hàng hoặc xác minh thông tin.

- Phương thức:

\* Sign in (đăng ký tài khoản)

\* Sign up (đăng nhập vào hệ thống)

\* Reset Password (thay đổi mật khẩu)

\* Add Balance (thêm số dư vào tài khoản)

\* ManageUser (quản lý các thông tin của tài khoản)

\* Purchase (xem và mua hàng)

\* PurchaseDetails (xem và quản lý các giao dịch)

- Quan hệ giữa các lớp:

Có quan hệ với lớp Transaction và UserBalance, Admin

**2. Category**

- Mục đích: Thêm các danh mục để quản lý các sản phẩm một cách tốt hơn

- Thuộc tính:

categoryID: Định danh duy nhất của danh mục, dùng để phân biệt các danh mục khác nhau trong hệ thống.

categoryName: Tên của danh mục, giúp người dùng dễ dàng nhận diện và tìm kiếm danh mục sản phẩm.

categoryDescription: Mô tả chi tiết về danh mục, cung cấp thông tin bổ sung về mục đích và các sản phẩm thuộc danh mục đó.

- Phương thức:

addCategory (thêm các category vào hệ thống)

- Quan hệ giữa các lớp:

Có quan hệ với lớp Product và Transaction, Admin

**3. Product**

- Mục đích: Đại diện cho các mặt hàng trong hệ thống

- Thuộc tính:

productID: Định danh duy nhất của sản phẩm, dùng để phân biệt các sản phẩm trong hệ thống.

productName: Tên của sản phẩm, giúp người dùng dễ dàng nhận diện và tìm kiếm sản phẩm.

productCategory: Danh mục mà sản phẩm thuộc về, giúp phân loại sản phẩm vào các nhóm liên quan.

quantity: Số lượng sản phẩm hiện có trong kho, giúp người dùng và quản trị viên biết được lượng hàng còn lại.

price: Giá bán của sản phẩm, cho biết chi phí mà khách hàng cần phải trả để mua sản phẩm.

productImageUrl: Đường dẫn tới hình ảnh của sản phẩm, giúp hiển thị hình ảnh của sản phẩm trong giao diện người dùng.

- Phương thức:

AddProduct (thêm các sản phẩm vào trong hệ thống)

AddCategory (thêm các Category cho Product)

- Quan hệ giữa các lớp:

Có quan hệ với lớp Category và Transaction, Admin

**4. UserBalance:**

- Mục đích: Quản lý số dư tài khoản cho mỗi User

- Thuộc tính:

userID: Định danh duy nhất của người dùng, liên kết với tài khoản của người dùng trong hệ thống.

balance: Số dư tài khoản của người dùng, đại diện cho số tiền hiện có mà người dùng có thể sử dụng để thực hiện giao dịch trong hệ thống.

- Phương thức:

addMoreBalance (nạp thêm tiền vào tài khoản UserBalance)

- Quan hệ giữa các lớp:

Có quan hệ với lớp User

**5. Supplier:**

- Mục đích: Đại diện cho một người cung cấp sản phẩm trong hệ thống

- Thuộc tính:

supplierID: Định danh duy nhất của nhà cung cấp, dùng để phân biệt các nhà cung cấp trong hệ thống.

supplierName: Tên của nhà cung cấp, giúp nhận diện nhà cung cấp trong hệ thống.

supplierEmail: Địa chỉ email của nhà cung cấp, dùng để liên lạc và gửi thông báo liên quan đến đơn hàng.

supplierPassword: Mật khẩu của nhà cung cấp, dùng để bảo mật tài khoản và xác thực khi đăng nhập vào hệ thống.

supplierPhone: Số điện thoại của nhà cung cấp, dùng để liên hệ trực tiếp khi cần thiết.

supplierAddress: Địa chỉ của nhà cung cấp, có thể được sử dụng cho mục đích giao hàng hoặc liên lạc.

- Phương thức:

Deliver (giao hàng cho một User)

ManageAccount (quản lý thông tin cho chính tài khoản của một Supplier)

ViewTransaction (xem chi tiết các giao dịch mà chính Supplier này đã thực hiện)

- Quan hệ giữa các lớp:

Có quan hệ với lớp Transaction, Admin

**6. Transaction:**

- Mục đích: Đại diện cho mỗi một giao dịch đã được thực hiện trong hệ thống.

- Thuộc tính:

purchaseID: Định danh duy nhất của đơn hàng, giúp phân biệt các đơn hàng trong hệ thống.

userID: Định danh duy nhất của người dùng, liên kết đơn hàng với tài khoản người dùng đã thực hiện mua hàng.

userName: Tên của người dùng đã thực hiện giao dịch mua hàng.

userPhone: Số điện thoại của người dùng, có thể dùng để liên hệ liên quan đến đơn hàng.

productID: Định danh duy nhất của sản phẩm được mua, dùng để xác định sản phẩm trong đơn hàng.

productName: Tên của sản phẩm được mua trong đơn hàng.

productPurchaseQuantity: Số lượng sản phẩm được mua trong đơn hàng.

productPrice: Giá của một đơn vị sản phẩm trong đơn hàng.

totalPrice: Tổng giá trị của đơn hàng, tính bằng cách nhân số lượng sản phẩm với giá của mỗi sản phẩm.

purchaseDate: Ngày thực hiện giao dịch mua hàng, giúp ghi lại thời gian của đơn hàng.

address: Địa chỉ giao hàng của người dùng, nơi sẽ nhận đơn hàng.

receivedDate: Ngày đơn hàng được nhận, xác định thời điểm giao hàng hoàn tất.

supplierName: Tên của nhà cung cấp sản phẩm trong đơn hàng, người chịu trách nhiệm cung cấp hàng hóa.

status: Trạng thái của đơn hàng (ví dụ: "đang xử lý", "đã giao", "hoàn trả"), cho biết tình trạng hiện tại của đơn hàng trong quy trình xử lý.

- Phương thức:

ViewTransaction (giúp xem và quản lý toàn bộ các giao dịch trong hệ thống)

- Quan hệ giữa các lớp:

Có quan hệ với lớp Admin, User, Supplier, Category, Product, UserBalance

**7. Admin**

- Mục đích: quản lý toàn bộ hệ thống, từ User, Supplier cho đến các Product, Category và Transaction

- Thuộc tính:

adminEmail: email của admin

adminPassword: password của admin

- Phương thức:

\* AddCategory (Thêm mới một Category)

\* AddProduct (thêm mới một Product)

\* AddCategoryToProduct (thêm một category vào một product)

\* DeleteCategory (xóa một category)

\* DeleteProduct (xóa một product)

\* AddSupplier (thêm mới một Supplier vào hệ thống)

\* ManageSupplier (thay đổi thông tin hoặc xóa Supplier ra khỏi hệ thống)

\* ManageUser (thay đổi thông tin hoặc xóa một User ra khỏi hệ thống)

\* SelectSupplier (lựa chọn Suppler cho một Transaction)

\* ViewTransaction (xem toàn bộ giao dịch đã được thực hiện trong hệ thống)

- Quan hệ giữa các lớp:

Có quan hệ với lớp User, Supplier, Category, Product, Transaction

**IV. UML và phân tích UML**

Ảnh có chứa văn bản, biểu đồ, Kế hoạch, Song song

Mô tả được tạo tự động

+) một product có một category

+) một category có thể có dung cho nhiều product

+) một User sẽ có một UserBalance

+) một UserBalance chỉ thuộc về một User

+) một supplier chỉ thuộc về một Admin

+) một Admin có thể quản lý nhiều Supplier

+) một User chỉ thuộc về một admin

+) một Admin sẽ quản lý được nhiều User

+) một category chỉ do một admin quản lý, một admin sẽ quản lý nhiều category

+) một product chỉ do một admin quản lý, một admin sẽ quản lý nhiều product

+) một Transaction chỉ do một admin quản lý, một admin sẽ quản lý nhiều Transaction

+) một User có thể tạo nhiều Transaction tương ứng, một Transaction tương ứng chỉ thuộc về một User

+) Một Supplier có thể thuộc về nhiều Transaction tương ứng, một Transaction tương ứng chỉ thuộc về một Supplier

+) Một product có thể xuất hiện trong nhiều Transaction khác nhau, mỗi Transaction chỉ có duy nhất một Product

**V. Xem xét các tính chất OOP đã có trong project**

+) Các thuộc tính trong mỗi class đều sử dụng private => nhằm mục đích bảo mật và tránh lộ thông tin.

+) Cùng một chức năng ví dụ như isInformationValid (xem thông tin nhập và đã hợp lệ hay chưa) nhưng đã được sửa lại cho mỗi class sao cho phù hợp hơn ví dụ để check thông tin đăng ký của User thì cần câu hỏi an ninh nhưng đối với thông tin đăng ký của Supplier thì không cần câu hỏi an ninh => tính chất đa hình đối với methods.