

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN MÔN HỌC**

**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI:**

**XÂY DỰNG WEBSITE BÁN ĐỒNG HỒ**

**GVHD: TS. NGUYỄN THÀNH SƠN**

**Sinh viên thực hiện:**

**NGUYỄN THÀNH LONG 18110313**

**TRẦN NGỌC MINH THIỆN 18110371**

**NGUYỄN TRUNG TÍN 18110381**

**HUỲNH THỊ THÚY VY 18110400**

**Tp. Hồ Chí Minh, tháng 12 năm 2020**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

----\*\*\*----

Tp. Hồ Chí Minh, ngày 17 tháng 12 năm 2020

# **NHIỆM VỤ ĐỀ TÀI MÔN HỌC**

Họ và tên sinh viên: MSSV:

Nguyễn Thành Long 18110313

Trần Ngọc Minh Thiện 18110371

Nguyễn Trung Tín 18110381

Huỳnh Thị Thúy Vy 18110400

Khoa: Công nghệ Thông tin Lớp: Thứ 2, tiết 1 - 4

Giảng viên hướng dẫn: TS.Nguyễn Thành Sơn ĐT: 0329 619 123

1. Tên đề tài:

2. Số liệu ban đầu:

3. Nội dung thực hiện đề tài:

4. Sản phẩm:

TRƯỞNG KHOA GIÁO VIÊN HƯỚNG DẪN

(Ký thay Trưởng khoa)

**Lê Văn Vinh TS. Nguyễn Thành Sơn**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

----\*\*\*----

Tp. Hồ Chí Minh, ngày 17 tháng 11 năm 2019

# **PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Họ và tên Sinh viên: MSSV:

Ngành:

Tên đề tài:

Họ và tên Giáo viên hướng dẫn:

**NHẬN XÉT**

1. Về nội dung đề tài và khối lượng thực hiện:

2. Ưu điểm:

3. Khuyết điểm:

4. Đề nghị cho bảo vệ hay không?

5. Đánh giá loại:

6. Điểm: (Bằng chữ: )

Tp. Hồ Chí Minh, ngày tháng năm 20

Giáo viên hướng dẫn

*(Ký và ghi rõ họ tên)*

**TS. Nguyễn Thành Sơn**

LỜI CAM ĐOAN

Tôi xin cam đoan:

1. Những nội dung trong báo cáo này là do tôi thực hiện dưới sự hướng dẫn trực tiếp của thầy Nguyễn Thành Sơn.

2. Mọi tham khảo dùng trong báo cáo đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.

3. Mọi sao chép không hợp lệ, vi phạm quy chế đào tạo, hay gian trá, tôi xin chịu hoàn toàn trách nhiệm.

Hồ Chí Minh, ngày 21 tháng 12 năm 2020

Nguyễn Thành Long Trần Ngọc Minh Thiện

Nguyễn Trung Tín Huỳnh Thị Thúy Vy

MỤC LỤC

[**NHIỆM VỤ ĐỀ TÀI MÔN HỌC** 1](#_Toc59406588)

[PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN 2](#_Toc59406589)

[LỜI CAM ĐOAN 4](#_Toc59406590)

[DANH SÁCH CÁC TỪ VIẾT TẮT 11](#_Toc59406591)

[DANH SÁCH CÁC HÌNH ẢNH, BIỂU ĐỒ 12](#_Toc59406592)

[DANH SÁCH BẢNG BIỂU 13](#_Toc59406593)

[**CHƯƠNG 1. TỔNG QUAN CHUNG** 14](#_Toc59406594)

[1.1. Mục tiêu 14](#_Toc59406595)

[1.2. Phạm vi 14](#_Toc59406596)

[1.3. Mô tả đề tài 14](#_Toc59406597)

[1.4. Thông tin phần mềm 15](#_Toc59406598)

[1.5. Chức năng 15](#_Toc59406599)

[1.5.1 Quản lý Tài khoản 15](#_Toc59406611)

[1.5.2 Quản lý Đồng hồ 15](#_Toc59406612)

[1.5.3 Quản lý Đơn hàng 15](#_Toc59406613)

[1.5.4 Tìm kiếm theo yêu cầu người dùng 15](#_Toc59406614)

[1.5.5 Lọc và sắp xếp theo tiêu chí 16](#_Toc59406615)

[1.5.6 Thống kê 16](#_Toc59406616)

[1.5.7 Quản lý Giỏ hàng 16](#_Toc59406617)

[1.6. Kế hoạch thực hiện 16](#_Toc59406618)

[**CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ** 18](#_Toc59406619)

[3.1 Lược đồ ERD 18](#_Toc59406620)

[3.2 Lược đồ quan hệ 18](#_Toc59406621)

[3.3 Thiết kế Cơ sở dữ liệu 19](#_Toc59406622)

[3.4 Sơ đồ Diagram 31](#_Toc59406623)

[**CHƯƠNG 3. XÂY DỰNG CƠ SỞ DỮ LIỆU** 32](#_Toc59406624)

[3.1. Danh sách các Stored Procedure 32](#_Toc59406625)

[3.2. Danh sách các Function 33](#_Toc59406626)

[3.3. Danh sách các Trigger 33](#_Toc59406627)

[3.4. Danh sách các View 34](#_Toc59406628)

[**CHƯƠNG 4. CÀI ĐẶT SQL** 35](#_Toc59406629)

[4.1. Store Procedure 35](#_Toc59406630)

[4.1.1. ChangePassword 35](#_Toc59406631)

[4.1.2. sp\_AddCartDetail 35](#_Toc59406632)

[4.1.3. sp\_ChangeInfo 36](#_Toc59406633)

[4.1.4. sp\_CheckLogin 36](#_Toc59406634)

[4.1.5. sp\_CheckWatch 37](#_Toc59406635)

[4.1.6. sp\_ConfirmBill 37](#_Toc59406636)

[4.1.7. sp\_ConfirmForCancel 37](#_Toc59406637)

[4.1.8. sp\_ConfirmForComplete 37](#_Toc59406638)

[4.1.9. sp\_ConfirmForShip 37](#_Toc59406639)

[4.1.10. sp\_ConfirmOrder 37](#_Toc59406640)

[4.1.11. sp\_DeleteCategory 38](#_Toc59406641)

[4.1.12. sp\_DeleteFirm 38](#_Toc59406642)

[4.1.13. sp\_DeleteSex 38](#_Toc59406643)

[4.1.14. sp\_GuiMaXacNhan 38](#_Toc59406644)

[4.1.15. sp\_InsertCategories 38](#_Toc59406645)

[4.1.16. sp\_InserFirm 39](#_Toc59406646)

[4.1.17. sp\_InsertAccount 39](#_Toc59406647)

[4.1.18. sp\_InsertSex 41](#_Toc59406648)

[4.1.19. sp\_InsertDiscount 41](#_Toc59406649)

[4.1.20. sp\_InsertNewOrder 41](#_Toc59406650)

[4.1.21. sp\_InsertOrderDetails 42](#_Toc59406651)

[4.1.22. sp\_InsertTypeDiscount 42](#_Toc59406652)

[4.1.23. sp\_InsertWatch 42](#_Toc59406653)

[4.1.24. sp\_InsertWatches\_Categories 42](#_Toc59406654)

[4.1.25. sp\_LockAccount 42](#_Toc59406655)

[4.1.26. sp\_LockWatch 43](#_Toc59406656)

[4.1.27. sp\_Logout 43](#_Toc59406657)

[4.1.28. sp\_RateWatch 43](#_Toc59406658)

[4.1.29. sp\_SearchWatch 43](#_Toc59406659)

[4.1.30. sp\_ThongKeDoanhThu 44](#_Toc59406660)

[4.1.31. sp\_UpdateFirm 44](#_Toc59406661)

[4.1.32. sp\_UpdatePhone 44](#_Toc59406662)

[4.1.33. sp\_UpdateSex 45](#_Toc59406663)

[4.1.34. sp\_InsertCartDetails 45](#_Toc59406664)

[4.1.35. sp\_DropCartDetails 46](#_Toc59406665)

[4.2. Function 48](#_Toc59406666)

[4.2.1. fn\_CheckLockUser 48](#_Toc59406667)

[4.2.2. fn\_LoadBill 48](#_Toc59406668)

[4.2.3. fn\_LoadBillforAccept 48](#_Toc59406669)

[4.2.4. fn\_LoadBillforShipper 48](#_Toc59406670)

[4.2.5. fn\_LoadBillforCancel 48](#_Toc59406671)

[4.2.6. fn\_LoadFirm 48](#_Toc59406672)

[4.2.7. fn\_LoadSex 49](#_Toc59406673)

[4.2.8. fn\_LoadWatchBill 49](#_Toc59406674)

[4.2.9. fn\_PhanTrang 49](#_Toc59406675)

[4.2.10. fn\_SearchWatch 49](#_Toc59406676)

[4.2.11. fn\_UserSeller 49](#_Toc59406677)

[4.2.12. fn\_WatchMaxDeal 50](#_Toc59406678)

[4.2.13. fn\_GetPrice 50](#_Toc59406679)

[4.2.14. fn\_CheckBan 50](#_Toc59406680)

[4.2.15. fn\_CheckLogin 50](#_Toc59406681)

[4.2.16. fn\_GetMaxBillID 51](#_Toc59406682)

[4.2.17. fn\_GetPrice 51](#_Toc59406683)

[4.2.18. fn\_GetPriceTotal 51](#_Toc59406684)

[4.3. Trigger 51](#_Toc59406685)

[4.3.1. tg\_CheckAddUser 51](#_Toc59406686)

[4.3.2. tg\_AddOrder 52](#_Toc59406687)

[4.3.3. tg\_CancelOrder 52](#_Toc59406688)

[4.3.4. tg\_InsertOrder 52](#_Toc59406689)

[4.3.5. tg\_InsertWatch 52](#_Toc59406690)

[4.3.6. tg\_RateWatch 52](#_Toc59406691)

[4.3.7. tg\_InsertAccount 53](#_Toc59406692)

[4.4. View 53](#_Toc59406693)

[4.4.1. view\_Account 53](#_Toc59406694)

[4.4.2. view\_Cart 53](#_Toc59406695)

[4.4.3. view\_CartDetails 54](#_Toc59406696)

[4.4.4. view\_Categories 54](#_Toc59406697)

[4.4.5. view\_District 54](#_Toc59406698)

[4.4.6. view\_OrderCancel 54](#_Toc59406699)

[4.4.7. view\_OrderDetail 54](#_Toc59406700)

[4.4.8. view\_Orders 55](#_Toc59406701)

[4.4.9. view\_OrdertoComplete 55](#_Toc59406702)

[4.4.10. view\_OrdertoConfirm 55](#_Toc59406703)

[4.4.11. view\_OrdertoShip 55](#_Toc59406704)

[4.4.12. view\_Province 55](#_Toc59406705)

[4.4.13. view\_Sex 55](#_Toc59406706)

[4.4.14. view\_UserSeller 55](#_Toc59406707)

[4.4.15. view\_Watches 56](#_Toc59406708)

[4.5. Phân quyền 57](#_Toc59406709)

[4.5.1. Phân quyền Admin 57](#_Toc59406718)

[4.5.2. Phân quyền Seller 59](#_Toc59406719)

[**CHƯƠNG 5. CÀI ĐẶT CHƯƠNG TRÌNH** 61](#_Toc59406720)

[5.1. Giao diện trang chủ mua hàng 61](#_Toc59406721)

[5.2. Giao diện trang bán hàng 62](#_Toc59406722)

[5.3. Tổ chức các lớp trong chương trình 65](#_Toc59406723)

[5.4. Kết nối CSDL từ xa 67](#_Toc59406724)

[**CHƯƠNG 6. TỔNG KẾT** 72](#_Toc59406725)

[6.1. Đóng góp đề tài 72](#_Toc59406726)

[6.2. Các tính năng bổ sung, đặc biệt trong đề tài 72](#_Toc59406727)

[6.2.1. Gửi mail 72](#_Toc59406728)

[6.2.2. Sử dụng Cursor 73](#_Toc59406729)

[6.2.3. Phân trang cho bảng 73](#_Toc59406730)

[6.2.4. Thêm login kết nối server 73](#_Toc59406731)

[6.2.5. Xóa login kết nối server 74](#_Toc59406732)

[6.3. Các hạn chế 75](#_Toc59406733)

[6.4. Hướng phát triển 75](#_Toc59406734)

[**TÀI LIỆU THAM KHẢO** 76](#_Toc59406735)

[**PHỤ LỤC** 77](#_Toc59406736)

[Hướng dẫn sử dụng 77](#_Toc59406737)

DANH SÁCH CÁC TỪ VIẾT TẮT

DBMS: Database Management System

CSDL: Cơ sở dữ liệu

DANH SÁCH CÁC HÌNH ẢNH, BIỂU ĐỒ

[Hình 1. Lược đồ ERD 13](#_Toc59406245)

[Hình 2. Diagram 26](file:///D:\VY%20HUỲNH\3rd%20Year\Database%20Management%20System\FINAL%20PROJECT\Nhom2_TrangWebBanDongHo.docx#_Toc59406246)

[Hình 3. Giao diện trang mua hàng 56](file:///D:\VY%20HUỲNH\3rd%20Year\Database%20Management%20System\FINAL%20PROJECT\Nhom2_TrangWebBanDongHo.docx#_Toc59406247)

[Hình 4. Giao diện trang mua hàng 56](file:///D:\VY%20HUỲNH\3rd%20Year\Database%20Management%20System\FINAL%20PROJECT\Nhom2_TrangWebBanDongHo.docx#_Toc59406248)

[Hình 5. Đơn hàng 57](file:///D:\VY%20HUỲNH\3rd%20Year\Database%20Management%20System\FINAL%20PROJECT\Nhom2_TrangWebBanDongHo.docx#_Toc59406249)

[Hình 6. Giao diện trang đăng ký 57](#_Toc59406250)

[Hình 7. Giao diện trang đăng nhập 58](file:///D:\VY%20HUỲNH\3rd%20Year\Database%20Management%20System\FINAL%20PROJECT\Nhom2_TrangWebBanDongHo.docx#_Toc59406251)

[Hình 8. Giao diện thống kê 58](#_Toc59406252)

[Hình 9. Giao diện thống kê 59](#_Toc59406253)

[Hình 10. Trang Sản phẩm 59](#_Toc59406254)

[Hình 11. Tổ chức các project trong solution 60](file:///D:\VY%20HUỲNH\3rd%20Year\Database%20Management%20System\FINAL%20PROJECT\Nhom2_TrangWebBanDongHo.docx#_Toc59406255)

[Hình 12. Models 61](file:///D:\VY%20HUỲNH\3rd%20Year\Database%20Management%20System\FINAL%20PROJECT\Nhom2_TrangWebBanDongHo.docx#_Toc59406256)

[Hình 13. Chuỗi ConnectionStrings 61](file:///D:\VY%20HUỲNH\3rd%20Year\Database%20Management%20System\FINAL%20PROJECT\Nhom2_TrangWebBanDongHo.docx#_Toc59406257)

DANH SÁCH BẢNG BIỂU

[Table 1. Kế hoạch thực hiện 11](#_Toc59405890)

[Table 2. Danh sách Stored Procedure 28](#_Toc59405891)

[Table 3. Danh sách Function 28](#_Toc59405892)

[Table 4. Danh sách Trigger 28](#_Toc59405893)

[Table 5. Danh sách View 29](#_Toc59405894)

CHƯƠNG 1. TỔNG QUAN CHUNG

* 1. Mục tiêu

Với sự phát triển không ngừng của công nghệ và kỹ thuật trong cuộc sống hiện nay, điện thoại, máy tính đang dần trở thành công cụ làm việc, giải trí không thể thiếu đối với con người. Có thể thấy, những năm gần đây, việc cập nhật các thông tin cũng như thực hiện các giao dịch qua mạng Internet ngày càng trở nên phổ biến và dễ dàng. Và một loại hình kinh doanh, được khai thác từ khía cạnh trên, hiện nay khá phổ biến đó là buôn bán qua mạng. Hình thức kinh doanh này mang lại rất nhiều lợi ích cho cả người tiêu dùng cũng như nhà cung cấp, giúp rút ngắn thời gian, công sức, chi phí đi lại của người tiêu dùng khi phải mua hàng trực tiếp tại store, nơi bán, đồng thời giúp doanh nghiệp dễ dàng theo dõi doanh thu, tiết kiệm thời gian, nhân lực trong khâu quản lý sản phẩm tại cửa hàng, phát triển, mở rộng thị trường, được nhiều người biết đến nhưng vẫn thuận tiện trong việc chăm sóc khách hàng, giảm chi phí quảng cáo và chi phí phát sinh trong quá trình vận hành. Bên cạnh đó, đảm bảo được dữ liệu luôn chính xác, đúng đắn, hạn chế sự cố ngoài ý muốn.

Với sự mở rộng cả về quy mô và chất lượng của các dịch vụ mua bán, trao đổi hàng hóa, cũng như nhu cầu của con người theo cách nhanh chóng và thuận tiện, các trang website bán hàng ngày càng đa dạng và đáp ứng được thị hiếu khách hàng.

Nắm bắt được tình hình trên, cùng với mục đích tìm hiểu, tiếp cận và góp phần đẩy mạnh sự phát triển của nền thương mại điện tử ở nước ta, nhóm em đã chọn thực hiện đề tài “XÂY DỰNG WEBSITE ĐỒNG HỒ”.

* 1. Phạm vi

Trang web áp dụng cho các cửa hàng, cá nhân buôn bán đồng hồ, đồng thời, áp dụng cho người dùng có nhu cầu mua sản phẩm thông qua mô hình mua bán qua mạng.

Các chức năng của trang web được xây dựng tập trung vào việc trao đổi buôn bán, lưu giữ thông tin và lịch sử mua hàng của người dùng, theo dõi và quản lý các đơn hàng, cung cấp các thông tin cần thiết của cửa hàng bán đồng hồ, cùng thông tin khách hàng đăng ký thực hiện mua đồng hồ.

* 1. Mô tả đề tài

Mỗi người dùng (cả mua và bán) đăng ký tài khoản ở trang web được phân biệt với nhau bởi Username và Email. Ngoài ra, cần cung cấp các thông tin cá nhân khác để hoàn thiện đăng ký bao gồm họ tên, địa chỉ, số điện thoại, giới tính, ảnh đại diện. Mỗi account khi tham gia mua hàng sẽ được tích lũy điểm Point theo quy định, có thể dùng để mua hàng. Mỗi account sẽ được phân loại tương ứng, phân chia theo quyền để truy cập trang web.

Mỗi Account có thể tạo ra các Shop để buôn bán, thông tin bao gồm ID, Tên shop, User làm chủ Shop, điểm Point để lưu số điểm Khách hàng đã sử dụng mua hàng. Mỗi shop sẽ được quản lý bởi một hoặc nhiều Seller, và ngược lại, một Seller cũng sẽ quản lý được nhiều Shop, Seller phải khai báo thông tin CMND và tài khoản ngân hàng để xác thực theo quy định.

Shop buôn bán Đồng hồ, mỗi Đồng hồ phải có những thông tin cơ bản bao gồm: hình ảnh đồng hồ, giới tính sử dụng, hãng, thể loại, giá, số lượng, thông tin chi tiết, shop buôn bán, đánh giá.

Mỗi tài khoản sẽ ứng với một giỏ hàng tương ứng, thông tin sản phẩm khi người dùng bỏ vào giỏ hàng sẽ được lưu lại, để đưa vào Đặt hàng khi người dùng xác nhận.

Thông tin Đặt hàng người dùng cần nhập vào, bao gồm: địa chỉ giao hàng, code giảm giá (nếu có), phương thức thanh toán. Sau khi đơn hàng được duyệt, sẽ tiến hành giao hàng, cập nhật thông tin giao hàng ở Status. Thông tin đơn hàng và sản phẩm tương ứng được lưu trong Chi tiết đơn hàng cho người dùng dễ quản lý và theo dõi.

Ngoài ra, mỗi tài khoản đều được ghi nhận lịch sử đăng nhập.

* 1. Thông tin phần mềm
* Tên phần mềm: Bee Watch Web
* Ngôn ngữ: C#
* Môi trường cài đặt: Visual Studio.
* Sử dụng SQL Server.
  1. Chức năng

6. 5. 1. Quản lý Tài khoản

Thêm, sửa, cập nhật thông tin tài khoản người dùng.

* + 1. Quản lý Đồng hồ

Thêm, sửa, cập nhật thông tin Chủng loại, Phái, Hãng, Đồng hồ có trong cửa hàng.

* + 1. Quản lý Đơn hàng

Theo dõi, thêm, sửa, cập nhật Đơn hàng người mua.

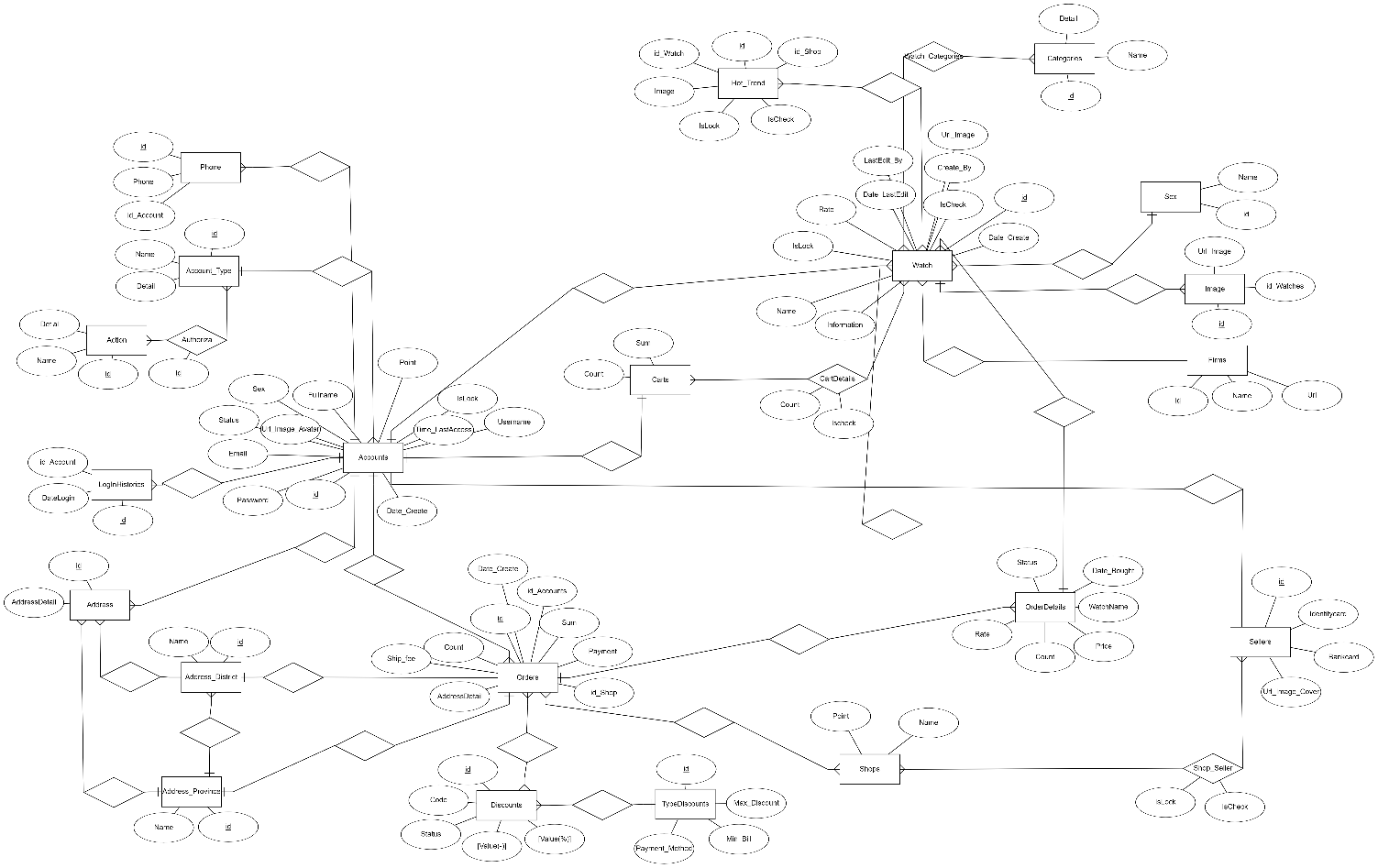
* + 1. Tìm kiếm theo yêu cầu người dùng
* Tìm kiếm người dùng, shop còn hoạt động.
* Tìm kiếm đồng hồ theo tên, tên gần đúng, theo phái, theo hãng, theo chủng loại, theo shop.
  + 1. Lọc và sắp xếp theo tiêu chí
* Lọc đồng hồ theo hãng, theo chủng loại, theo giá.
  + 1. Thống kê
* Thống kê doanh thu sản phẩm: theo tuần, theo tháng, theo năm.
* Thống kê số lượng người dùng: người dùng bị khóa tài khoản, người dùng đăng ký tài khoản.
  + 1. Quản lý Giỏ hàng
* Thêm, xóa, sửa số lượng sản phẩm trong giỏ hàng.
* Xác nhận đặt hàng trong giỏ để tạo đơn hàng mới.
  1. Kế hoạch thực hiện

Table 1. Kế hoạch thực hiện

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TUẦN** | **CÔNG VIỆC THỰC HIỆN** | **NGÀY BD** | **NGÀY KT** | **KẾT QUẢ** |
| 1 | Viết mô tả về đề tài (nêu rõ các thành phần dữ liệu, chức năng và giao diện cần có để project chạy tốt) | 5/10/2020 | 12/10/2020 | Hoàn thành |
| 2 | Thiết kế CSDL và các ràng buộc (dùng ERD) | 12/20/2020 | 19/10/2020 | Hoàn thành |
| 3 | Cài đặt CSDL và các ràng buộc, trigger.  Nhập dữ liệu | 19/10/2020 | 26/10/2020 | Hoàn thành |
| 4 | Thiết kế các view (trình bày bằng mã giả) | 26/10/2020 | 2/11/2020 | Hoàn thành |
| 5 | Cài đặt các view | 2/11/2020 | 9/11/2020 |  |
| 6 | Thiết kế giao diện, xác định các users và quyền sử dụng CSDL của họ | 9/11/2020 | 16/11/2020 | Hoàn thành |
| 7 | Cài đặt giao diện và kết nối chương trình với CSDL, tạo user, phần quyền | 16/11/2020 | 23/11/2020 | Hoàn thành |
| 8 | Thiết kế các hàm, thủ tục thực hiện các chức năng của đề tài | 23/11/2020 | 30/11/2020 | Hoàn thành |
| 9 |  | 30/11/2020 | 07/12/2020 |  |
| 10 | Cài đặt các hàm, thủ tục thực hiện các chức năng của đề tài | 07/12/2020 | 14/12/2020 | Hoàn thành |
| 11 | Viết báo cáo | 14/12/2020 | 21/12/2020 | Hoàn thành |

CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ

1. Lược đồ ERD



Hình 1. Lược đồ ERD

1. Lược đồ quan hệ

**Account\_Type** (id, Name, Detail)

**Account** (id, Username, Fullname, Password, id\_Account\_Type, Url\_Image\_Avatar, Email, Sex, Point, Date\_Create, Time\_LastAccess, IsLock, Status)

**Action** (id, Name, Detail)

**Address** (id, id\_Account, id\_Province, id\_District, AddressDetail)

**Address\_ Province** (id, Name)

**Address\_District** (id, Name, id\_Province)

**Authoriza** (id, id\_Account\_Type, id\_Action)

**Carts** (id, Count, Sum)

**CartDetails** (id\_Watch, id\_Cart, Count, IsCheck)

**Categories** (id, Name, Detail, Status)

**Discounts** (id, Code, id\_Type, [Value(%)], [Value(-)], DateFrom, DateTo, Status)

**Firms** (id, Name, URL, Status)

**Hot\_Trend** (id, id\_Watch, id\_Shop, Image, IsCheck, IsLock)

**Image** (id, id\_Watches, Url\_Image)

**LogInHistories** (id, id\_Account, DateLogin)

**Orders** (id, id\_Shop, id\_Accounts, Address\_Province, Address\_District, AddressDetail, Count, Sum, Ship\_fee, id\_Discount, Payment, Date\_Create, Date\_Check, User\_Check, Status)

**OrderDetails** (id\_Order, id\_Watches, WatchName, Price, Count, Rate, Date\_Bought, Status)

**Phone** (id, id\_Account, Phone)

**Sellers** (id, Identitycard, Bankcard, Url\_Image\_Cover)

**Sex** (id, Name, Status)

**Shops** (id, Name, id\_Master, Address, Point, Status)

**Shop\_Seller** (id\_Shop, id\_Seller, IsCheck, IsLock)

**Type\_Discounts** (id, Min\_Bill, Max\_Discount, Payment\_Method)

**Watches** (id, Url\_Image, id\_Sex, id\_Firms, Name, Price, Count, Information, id\_Shop, Rate, Date\_Create, Create\_By, Date\_LastEdit, LastEdit\_By, IsLock, IsCheck, IsExist)

**Watches\_Categories** (id\_Watch, id\_Category, Detail)

1. Thiết kế Cơ sở dữ liệu

* Bảng Account\_Type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Account\_Type** | | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | id | Int | PRIMARY KEY | Khóa chính phân biệt các loại tài khoản. |
| 2 | Name | nvarchar(50) | NOT NULL | Tên loại tài khoản. |
| 3 | Detail | nvarchar(100) |  | Chi tiết chức năng tài khoản. |

* Bảng Account

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Account** | | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | Id | Int | PRIMARY KEY | Khóa chính phân biệt các tài khoản. |
| 2 | Username | nvarchar(50) | NOT NULL,  UNIQUE | Tên tài khoản |
| 3 | Fullname | nvarchar(50) | NOT NULL | Họ và tên người dùng |
| 4 | Password | nvarchar(50) | NOT NULL | Mật khẩu tài khoản |
| 5 | id\_Account\_Type | int | FOREIGN KEY | Id loại tài khoản |
| 6 | Url\_Image\_Avatar | nvarchar(MAX) |  | Đường link ảnh đại diện |
| 7 | Email | nvarchar(50) | NOT NULL,  UNIQUE | Email đăng ký tài khoản |
| 8 | Sex | nchar(10) |  | Giới tính |
| 9 | Point | float |  | Điểm tích lũy |
| 10 | Date\_Create | date |  | Ngày tạo tài khoản |
| 11 | Time\_LastAccess | datetime |  | Thời gian truy cập cuối |
| 12 | IsLock | bit |  | 0 – Không bị khóa  1 – Bị khóa |
| 13 | Status | bit |  | Tình trạng tài khoản:  0 – Offline  1 – Đang online |

* Bảng Action

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Action** | | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | id | int | PRIMARY KEY | Khóa chính phân biệt các loại hành động |
| 2 | Name | nvarchar(50) | NOT NULL | Tên hành động |
| 3 | Detail | nvarchar(MAX) | NOT NULL | Chi tiết hành động lên hệ thống. |

* Bảng Authoriza

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Authoriza** | | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | id | int | PRIMARY KEY | Khóa chính phân biệt các quyền |
| 2 | id\_Account\_Type | int | FOREIGN KEY | Id loại tài khoản. |
| 3 | id\_Action | int | FOREIGN KEY | Id hành động. |

* Bảng Address\_ Province

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address\_Province** | | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | id | nchar(10) | PRIMARY KEY | Khóa chính phân biệt các tỉnh, thành phố |
| 2 | Name | nvarchar(MAX) | NOT NULL | Tên tỉnh, thành phố |

* Bảng Address\_District

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address\_ District** | | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | id | nchar(10) | PRIMARY KEY | Khóa chính phân biệt các huyện, quận |
| 2 | Name | nvarchar(MAX) | NOT NULL | Tên huyện, quận |
| 3 | id\_Province | nchar(10) | FOREIGN KEY | Id tỉnh, thành phố thuộc về. |

* Bảng Address

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Address** | | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | id | int | PRIMARY KEY | Khóa chính phân biệt các địa chỉ |
| 2 | id\_Account | int | FOREIGN KEY | Id account của địa chỉ |
| 3 | id\_Province | nchar(10) | FOREIGN KEY | Id tỉnh, thành phố |
| 4 | id\_District | nchar(10) | FOREIGN KEY | Id huyện, quận |
| 5 | AddressDetail | nvarchar(MAX) | NOT NULL | Địa chỉ cụ thể: số nhà, xã, phường, thị trấn,… |

* Bảng Carts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Carts** | | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | id | int | PRIMARY KEY | Khóa chính phân biệt các giỏ hàng, mỗi id account tương ứng với 1 id giỏ hàng |
| 2 | Count | int |  | Số lượng sản phẩm có trong giỏ |
| 3 | Sum | float |  | Tổng tiền sản phẩm có trong giỏ |

* Bảng CartDetails

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CartDetails** | | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | id\_Watch | int | PRIMARY KEY | Id đồng hồ có trong giỏ |
| 2 | id\_Cart | int | PRIMARY KEY | Id giỏ hàng |
| 3 | Count | int | NOT NULL | Số lượng đồng hồ người dùng mua tương ứng |
| 4 | IsCheck | bit |  | 1 – Người dùng xác nhận mua đồng hồ  0 – Người dùng chưa xác nhận |

* Bảng Categories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Categories** | | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | id | int | PRIMARY KEY | Id chủng loại của đồng hồ |
| 2 | Name | nvarchar(50) | NOT NULL | Tên chủng loại |
| 3 | Detail | nvarchar(MAX) | NOT NULL | Chi tiết chủng loại đồng hồ |
| 4 | Status | bit |  | 1 – Hợp lệ, còn được sử dụng  0 – Không hợp lệ, không còn được sử dụng |

* Bảng Discounts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Discounts** | | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | id | int | PRIMARY KEY | Khóa chính phân biệt các Mã giảm giá |
| 2 | Code | nchar(10) | NOT NULL | Mã giảm giá |
| 3 | id\_Type | int | FOREIGN KEY | Id loại mã giảm giá |
| 4 | [Value(%)] | float | NOT NULL | Giảm giá tính theo phần trăm tiền mua sắm |
| 5 | [Value(-)] | float | NOT NULL | Giảm giá tính theo trừ tiền từ tiền mua sắm |
| 6 | DateFrom | datetime | NOT NULL | Ngày bắt đầu giảm giá |
| 7 | DateTo | datetime | NOT NULL | Ngày kết thúc giảm giá |
| 8 | Status | bit | NOT NULL | 1 – Có thể sử dụng  0 – Hết hạn, không thể sử dụng |

* Bảng Firms

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Firms** | | | | |
| 1 | id | int | PRIMARY KEY | Khóa chính phân biệt các hãng đồng hồ |
| 2 | Name | nvarchar(50) |  | Tên của hãng đồng hồ |
| 3 | Url | nvarchar(MAX) |  | Đường link ảnh logo hãng |
| 4 | Status | bit |  | 1 – Hợp lệ, còn được sử dụng  0 – Không hợp lệ, không còn được sử dụng |

* Bảng Hot\_Trend

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hot\_Trend** | | | | |
| 1 | id | int | PRIMARY KEY | Khóa chính phân biệt các loại đồng hồ thịnh hành |
| 2 | id\_Watch | int | FOREIGN KEY | Khóa ngoại trỏ đến đồng hồ theo Id của đồng hồ |
| 3 | id\_Shop | int | FOREIGN KEY | Khóa ngoại trỏ đến Shop theo Id của Shop |
| 4 | Image | nvarchar(MAX) |  | Link dẫn đến ảnh của Trend |
| 5 | IsCheck | bit |  | 1 – Được duyệt đưa lên trang web  0 – Không được duyệt |
| 6 | IsLock | bit |  | 0 – Không bị khóa  1 – Bị khóa |

* Bảng Image

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Image** | | | | |
| 1 | id | int | PRIMARY KEY | Khóa chính phân biệt các Link dẫn đến các ảnh |
| 2 | id\_Watches | int | FOREIGN KEY | Khóa ngoại trỏ đến đồng hồ theo Id |
| 3 | Url\_Image | nvarchar(MAX) |  | Link dẫn đến ảnh đồng hồ |

* Bảng LogInHistories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **LogInHistories** | | | | |
| 1 | id | int | PRIMARY KEY | Khóa chính phân biệt các lần đăng nhập |
| 2 | id\_Account | int | FOREIGN KEY | Khóa ngoại trỏ đến các tài khoản theo Id |
| 3 | DateLogin | datetime |  | Thời gian đăng nhập |

* Bảng OrdersDetails

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OrderDetails** | | | | |
| 1 | id\_Order | int | PRIMARY KEY | Id Order |
| 2 | id\_Watches | int | PRIMARY KEY | Id Watch |
| 3 | WatchName | nvarchar(50) | NOT NULL | Tên đồng hồ |
| 4 | Price | float | NOT NULL | Giá đồng hồ |
| 5 | Count | int |  | Số lượng đồng hồ |
| 6 | Rate | int |  | Đánh giá |
| 7 | Date\_Bought | datetime |  | Ngày mua đồng hồ |
| 8 | Status | int |  | 1. – Đợi xác nhận 2. – Đợi shipper lấy hàng 3. – Đang giao hàng 4. – Đã giao 5. – Đã hủy |

* Bảng Orders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Orders** | | | | |
| 1 | id | int | PRIMARY KEY | Khóa chính phân biệt các đơn hàng |
| 2 | id\_Shop | int | FOREIGN KEY | Id Shop |
| 3 | id\_Accounts | int | FOREIGN KEY | Id Tài khoản |
| 4 | Address\_Province | nchar(10) |  | Tỉnh |
| 5 | Address\_District | nchar(10) |  | Huyện |
| 6 | AddressDetail | nvarchar(MAX) |  | Địa chỉ cụ thể |
| 7 | Count | int |  | Số lượng hàng |
| 8 | Sum | float |  | Tổng tiền đơn hàng |
| 9 | Ship\_fee | float |  | Phí vận chuyện |
| 10 | id\_Discount | int |  | Id tài khoản |
| 11 | Payment | nvarchar(20) |  | Số tài khoản |
| 12 | Date\_Create | datetime |  | Ngày tạo hóa đơn |
| 13 | Date\_Check | datetime |  | Ngày kiểm tra |
| 14 | User\_Check | nvarchar(50) |  | Người kiểm tra |
| 15 | Status | int |  | 1. – Đợi xác nhận 2. – Đợi shipper lấy hàng 3. – Đang giao hàng 4. – Đã giao 5. – Đã hủy |

* Bảng Phone

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phone** | | | | |
| **1** | id\_Account | int | FOREIGN KEY | Id tài khoản |
| 2 | Phone | nchar(10) | NOT NULL | Số điện thoại |
| 3 | id | int | PRIMARY KEY | Khóa chính để phân biệt các số điện thoại |

* Bảng Sellers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sellers** | | | | |
| **1** | id | int | PRIMARY KEY | Khóa chính phân biệt các người bán |
| 2 | Identitycard | char(9) |  | Chứng minh nhân dân |
| 3 | Bankcard | char(20) |  | Tài khoản ngân hàng |
| 4 | Url\_Image\_Cover | nvarchar(MAX) |  | Link ảnh đại diện |

* Bảng Sex

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sex** | | | | |
| **1** | id | int | PRIMARY KEY | Khóa chính phân biệt |
| 2 | Name | nvarchar(20) | NOT NULL | Loại giới tính |
| 3 | Status | bit |  | 0 – Không tồn tại  1 – Tồn tại |

* Bảng Shops

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Shops** | | | | |
| **1** | id | int | PRIMARY KEY | Khóa chính phân biệt các Shop |
| 2 | Name | nvarchar(MAX) |  | Tên Shop |
| 3 | id\_Master | int | FOREIGN KEY | Id chủ cửa hàng |
| 4 | Address | nvarchar(50) |  | Địa chỉ |
| 5 | Point | float |  | Điểm |
| 6 | Status | bit |  | 1 – Còn hoạt động  0 – Ngừng hoạt động |

* Bảng Shop\_Seller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Shop\_Seller** | | | | |
| **1** | id\_Shop | int | PRIMARY KEY | Id Shop |
| 2 | id\_Seller | int | PRIMARY KEY | Id Người bán |
| 3 | IsCheck | bit |  | 1 – Duyệt hợp lệ  0 – Chưa duyệt |
| 4 | IsLock | bit |  | 1 – Bị khóa  0 – Còn hoạt động |

* Bảng Type\_Discounts

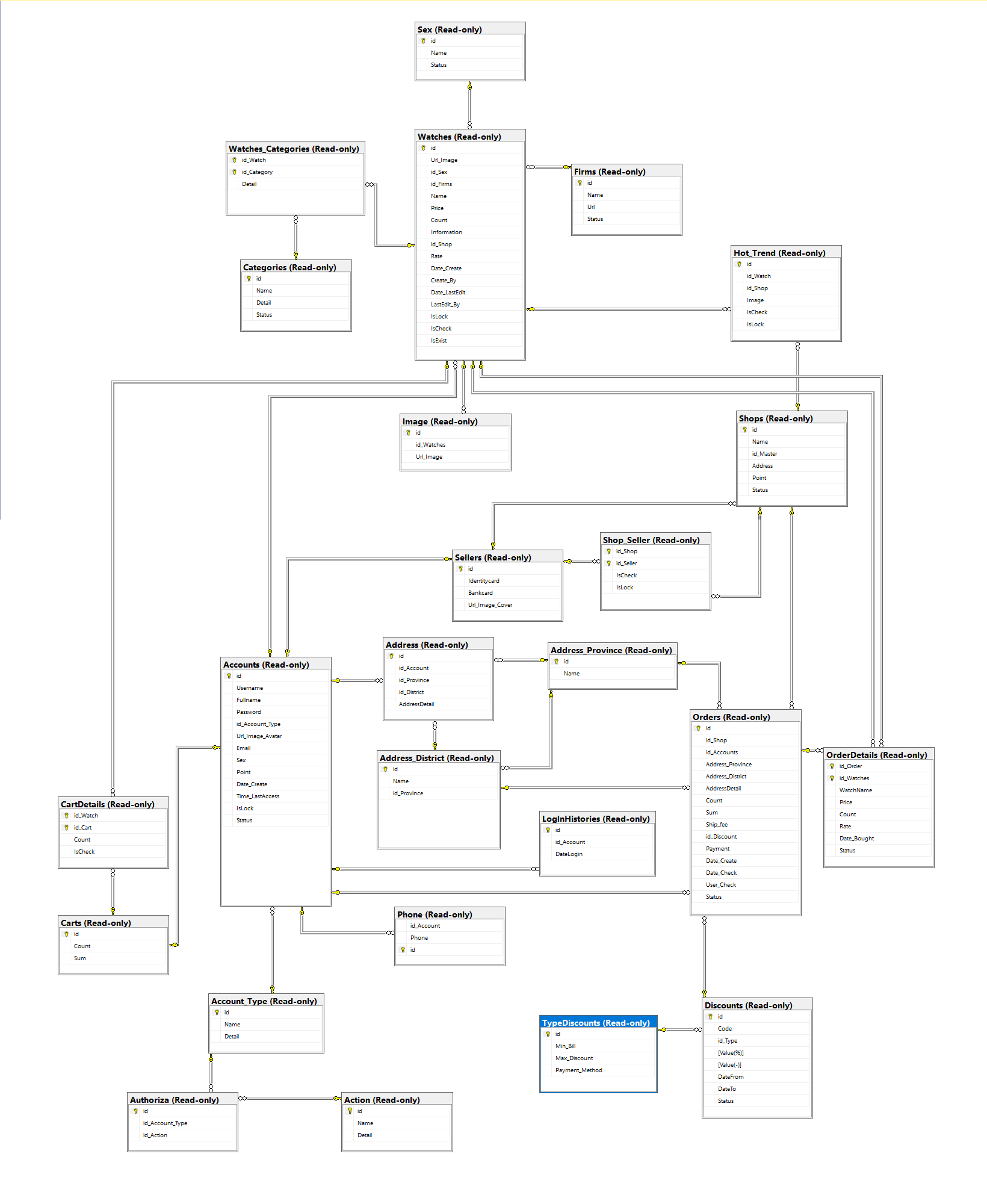
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TypeDiscounts** | | | | |
| **1** | id | int | PRIMARY KEY | Khóa chính phân biệt các loại giảm giá |
| 2 | Min\_Bill | float |  | Giá trị tối thiểu của hóa đơn |
| 3 | Max\_Discount | float |  | Số tiền giảm tối đa |
| 4 | Payment\_Method | nvarchar(50) |  | Phương thức thanh toán |

* Bảng Watches

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Watches** | | | | |
| **1** | id | int | PRIMARY KEY | Id phân biệt các đồng hồ |
| 2 | Url\_Image | nvarchar(MAX) |  | Link ảnh các đồng hồ |
| 3 | id\_Sex | int | FOREIGN KEY | Id giới tính |
| 4 | id\_Firms | int | FOREIGN KEY | Id Hãng sản xuất |
| 5 | Name | nvarchar(50) | NOT NULL | Tên đồng hồ |
| 6 | Price | float | NOT NULL | Giá |
| 7 | Count | int | NOT NULL | Số lượng |
| 8 | Information | nvarchar(MAX) | NOT NULL | Thông tin chi tiết |
| 9 | id\_Shop | int | FOREIGN KEY | Id cửa hàng |
| 10 | Rate | float |  | Đánh giá |
| 11 | Date\_Create | datetime |  | Ngày nhập hàng |
| 12 | Create\_By | nvarchar(50) |  | Nhập bởi |
| 13 | Date\_LastEdit | datetime |  | Lần chỉnh sửa cuối cùng |
| 14 | LastEdit\_By | nvarchar(50) |  | Người chỉnh sửa cuối cùng |
| 15 | IsLock | bit |  | 1 – Bị khóa  0 – Không bị khóa |
| 16 | IsCheck | bit |  | 1 – Được duyệt lên web  0 – Chưa được duyệt |
| 17 | IsExist | bit |  | 1 – Còn tồn tại  0 – Không còn tồn tại |

* Bảng Watches\_Categories

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Watches\_Categories** | | | | |
| **1** | id\_Watch | int | PRIMARY KEY | Id đồng hồ |
| 2 | id\_Category | int | PRIMARY KEY | Id loại đồng hồ |
| 3 | Detail | nvarchar(50) |  | Thông tin chi tiết |

* 1. Sơ đồ Diagram

Hình 2. Diagram

CHƯƠNG 3. XÂY DỰNG CƠ SỞ DỮ LIỆU

1. Danh sách các Stored Procedure

|  |  |  |
| --- | --- | --- |
| STT | TÊN | MÔ TẢ |
| 1 | ChangePassword | Thay đổi mật khẩu tài khoản |
| 2 | sp\_AddCartDetail | Thêm thông tin chi tiết Giỏ hàng |
| 3 | sp\_ChangeInfo | Thay đổi thông tin người dùng |
| 4 | sp\_CheckLogin | Kiểm tra đăng nhập |
| 5 | sp\_CheckWatch | Duyệt sản phẩm lên trang web |
| 6 | sp\_ConfirmBill | Xác nhận đơn hàng |
| 7 | sp\_ConfirmForCancel | Xác nhận hủy đơn hàng |
| 8 | sp\_ConfirmForComplete | Xác nhận hoàn tất thanh toán |
| 9 | sp\_ConfirmForShip | Xác nhận vận chuyển |
| 10 | sp\_ConfirmOrder | Xác nhận đặt hàng |
| 11 | sp\_DeleteCategory | Xóa loại đồng hồ |
| 12 | sp\_DeleteFirm | Xóa hãng đồng hồ |
| 13 | sp\_DeleteSex | Xóa phái của đồng hồ |
| 14 | sp\_GuiMaXacNhan | Gửi mã xác nhận đơn hàng |
| 15 | sp\_InsertCategories | Thêm loại đồng hồ |
| 16 | sp\_InserFirm | Thêm hãng sản xuất |
| 17 | sp\_InsertAccount | Thêm mới tài khoản |
| 18 | sp\_InsertSex | Thêm giới tính |
| 19 | sp\_InsertDiscount | Thêm mã giảm giá |
| 20 | sp\_InsertNewOrder | Thêm đơn hàng mới |
| 21 | sp\_InsertOrderDetails | Thêm chi tiết đơn hàng mới |
| 22 | sp\_InsertTypeDiscount | Thêm loại giảm giá |
| 23 | sp\_InsertWatch | Thêm đồng hồ mới |
| 24 | sp\_InsertWatches\_Categories | Thêm chủng loại và đồng hồ tương ứng |
| 25 | sp\_LockAccount | Khóa tài khoản |
| 26 | sp\_LockWatch | Khóa đồng hồ |
| 27 | sp\_Logout | Đăng xuất |
| 28 | sp\_RateWatch | Đánh giá sản phẩm |
| 29 | sp\_SearchWatch | Tìm sản phẩm |
| 30 | sp\_ThongKeDoanhThu | Thống kê doanh thu |
| 31 | sp\_UpdateFirm | Chỉnh sửa hãng sản xuất |
| 32 | sp\_UpdatePhone | Chỉnh sửa số điện thoại |
| 33 | sp\_UpdateSex | Cập nhật thông tin Phái |
| 34 | sp\_UpdateWatch | Chỉnh sửa đồng hồ |

Table 2. Danh sách Stored Procedure

1. Danh sách các Function

|  |  |  |
| --- | --- | --- |
| STT | TÊN | MÔ TẢ |
| 1 | fn\_CheckLockUser | Kiểm tra User có bị Khóa hay không (có hành vi sai quy định sẽ bị khóa tài khoản) |
| 2 | fn\_LoadBill | Lấy thông tin hóa đơn từ Tài khoản |
| 3 | fn\_LoadBillforAccept | Lấy thông tin Bill chờ xác nhận |
| 4 | fn\_LoadBillforShipper | Lấy thông tin Bill chờ Shipper lấy hàng |
| 5 | fn\_LoadBillforCancel | Lấy thông tin Bill đã hủy |
| 6 | fn\_LoadFirm | Lấy thông tin Đồng hồ theo Hãng |
| 7 | fn\_LoadSex | Lấy thông tin Đồng hồ theo giới tính |
| 8 | fn\_LoadWatchBill | Lấy thông tin đơn hàng Chờ xác nhận |
| 9 | fn\_PhanTrang | Phân trang |
| 10 | fn\_SearchWatch | Tìm kiếm Đồng hồ |
| 11 | fn\_UserSeller | Lấy thông tin Tài khoản là người bán và không bị khóa |
| 12 | fn\_WatchMaxDeal | Lấy thông tin sản phẩm bán chạy |

Table 3. Danh sách Function

1. Danh sách các Trigger

|  |  |  |
| --- | --- | --- |
| STT | TÊN | MÔ TẢ |
| 1 | tg\_CheckAddUser | Kiểm tra thông tin người dùng nhập vào không được trống |
| 2 | tg\_AddOrder | Xóa bớt số lượng hàng tồn khi thêm đơn hàng |
| 3 | tg\_CancelOrder | Cộng thêm số lượng hàng tồn khi hủy đơn hàng |
| 4 | tg\_InsertOrder | Thêm hóa đơn mới |
| 5 | tg\_InsertWatch | Thêm sản phẩm đồng hồ mới |
| 6 | tg\_RateWatch | Đánh giá sản phẩm thông qua hóa đơn chi tiết |
| 7 | tg\_InsertAccount | Thêm tài khoản người dùng mới |

Table 4. Danh sách Trigger

1. Danh sách các View

|  |  |  |
| --- | --- | --- |
| STT | TÊN | MÔ TẢ |
| 1 | view\_Account | Khung nhìn tài khoản |
| 2 | view\_Cart | Khung nhìn giỏ hàng |
| 3 | view\_CartDetails | Khung nhìn chi tiết giỏ hàng |
| 4 | view\_Categories | Khung nhìn chủng loại đồng hồ |
| 5 | view\_District | Khung nhìn danh sách huyện |
| 6 | view\_OrderCancel | Khung nhìn hóa đơn đã hủy |
| 7 | view\_OrderDetail | Khung nhìn chi tiết hóa đơn |
| 8 | view\_Orders | Khung nhìn hóa đơn |
| 9 | view\_OrdertoComplete | Khung nhìn hóa đơn hoàn thành |
| 10 | view\_OrdertoConfirm | Khung nhìn hóa đơn chờ xác nhận |
| 11 | view\_OrdertoShip | Khung nhìn hóa đơn chờ Shipper lấy hàng |
| 12 | view\_Province | Khung nhìn danh sách Tỉnh |
| 13 | view\_Sex | Khung nhìn đồng hồ theo giới tính |
| 14 | view\_UserSeller | Khung nhìn người dùng là người bán hàng |
| 15 | view\_Watches | Khung nhìn đồng hồ |

Table 5. Danh sách View

CHƯƠNG 4. CÀI ĐẶT SQL

1. Store Procedure

4.1.1. ChangePassword

- Thay đổi mật khẩu tài khoản: được thực thi mật khẩu nhập vào không được rỗng, mật khẩu mới không được trùng với mật khẩu cũ.

|  |
| --- |
| CREATE PROC [dbo].[ChangePassword] @id INT, @username nvarchar(50),  @old\_pass nvarchar(50), @new\_pass nvarchar(50), @repassword nvarchar(50)  AS  BEGIN  DECLARE @old nvarchar(50) = (SELECT Password  FROM Accounts  WHERE id = @id AND Username = @username);  IF(@old\_pass != '' AND @new\_pass != '' AND @repassword != '')  BEGIN  IF(@old\_pass = @old AND @new\_pass != @old\_pass)  IF(@new\_pass = @repassword)  BEGIN  UPDATE Accounts SET Password = @new\_pass  WHERE id = @id AND Username = @username  END  ELSE  ROLLBACK  END  END  GO |

4.1.2. sp\_AddCartDetail

- Lưu thông tin sản phẩm trong giỏ hàng: kiểm tra nếu giỏ hàng đã có sản phẩm đó thì tăng số lượng Count, ngược lại, nếu chưa có thì thêm sản phẩm mới.

|  |
| --- |
| CREATE PROC [dbo].[sp\_AddCartDetail] @idcart int, @idwatch int, @count int  AS  BEGIN  IF EXISTS(SELECT \* FROM CartDetails WHERE id\_Cart = @idcart AND id\_Watch = @idwatch)  BEGIN  UPDATE CartDetails SET Count = Count + @count  END  ELSE INSERT INTO CartDetails(id\_Cart, id\_Watch, Count, IsCheck) VALUES(@idcart, @idwatch, @count, 0)  END  GO  /\*\*\*\*\*\* Object: StoredProcedure [dbo].[sp\_ChangeInfo] Script Date: 19-Dec-20 14:40:27 \*\*\*\*\*\*/  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO |

4.1.3. sp\_ChangeInfo

- Thay đổi thông tin của tài khoản: tài khoản chỉ được thay đổi thông tin khi nhập đúng username và password.

|  |
| --- |
| CREATE PROCEDURE [dbo].[sp\_ChangeInfo] @username nvarchar(50), @password nvarchar(50), @newpass nvarchar(50), @avt nvarchar(MAX),  @email nvarchar(50), @fullname nvarchar(50)  AS  BEGIN  DECLARE @IsRightPass int = 0;  SELECT @IsRightPass = COUNT(\*) FROM Accounts  WHERE Username = @username AND Password = @password  IF(@IsRightPass = 1)  BEGIN  IF(@newpass = NULL OR @newpass = '')  BEGIN  UPDATE Accounts  SET Url\_Image\_Avatar = @avt, Email = @email, Fullname = @fullname  WHERE Username = @username  END  ELSE  UPDATE Accounts  SET Password = @newpass, Url\_Image\_Avatar = @avt, Email = @email, Fullname = @fullname  WHERE Username = @username  END  END  GO |

4.1.4. sp\_CheckLogin

|  |
| --- |
| CREATE PROCEDURE [dbo].[sp\_CheckLogin] @id int, @username nvarchar(50), @password nvarchar(50)  AS  BEGIN  IF EXISTS (SELECT id FROM Accounts WHERE id = @id AND Username = @username AND Password = @password)  BEGIN  UPDATE Accounts  SET Status = 1  WHERE Username = @username AND Password = @password  INSERT INTO dbo.LogInHistories(id\_Account, DateLogIn) VALUES(@id, GETDATE())  END  ELSE ROLLBACK  END  GO  /\*\*\*\*\*\* Object: StoredProcedure [dbo].[sp\_CheckWatch] Script Date: 19-Dec-20 14:40:27 \*\*\*\*\*\*/  SET ANSI\_NULLS ON  GO  SET QUOTED\_IDENTIFIER ON  GO |

4.1.5. sp\_CheckWatch

|  |
| --- |
| CREATE PROC [dbo].[sp\_CheckWatch] @id int, @name NVARCHAR(50)  AS  IF EXISTS (SELECT \* FROM Watches WHERE id = @id and Name = @name)  BEGIN TRANSACTION  UPDATE Watches  SET IsCheck = 'True'  WHERE id = @id AND Name = @name  COMMIT TRANSACTION  GO |

4.1.6. sp\_ConfirmBill

|  |
| --- |
| CREATE PROC [dbo].[sp\_ConfirmBill] @idorder int  AS  BEGIN TRANSACTION  UPDATE Orders SET Status = 1, Date\_Check = GETDATE() WHERE id = @idorder  COMMIT TRANSACTION  GO |

4.1.7. sp\_ConfirmForCancel

|  |
| --- |
| CREATE PROC [dbo].[sp\_ConfirmForCancel] @idorder int  AS  BEGIN TRANSACTION  UPDATE Orders SET Status = 4, Date\_Check = GETDATE() WHERE id = @idorder  COMMIT TRANSACTION  GO |

4.1.8. sp\_ConfirmForComplete

|  |
| --- |
| CREATE PROC [dbo].[sp\_ConfirmForComplete] @idorder int  AS  BEGIN TRANSACTION  UPDATE Orders SET Status = 3, Date\_Check = GETDATE() WHERE id = @idorder  COMMIT TRANSACTION  GO |

4.1.9. sp\_ConfirmForShip

|  |
| --- |
| CREATE PROC [dbo].[sp\_ConfirmForShip] @idorder int  AS  BEGIN TRANSACTION  UPDATE Orders SET Status = 2, Date\_Check = GETDATE() WHERE id = @idorder  COMMIT TRANSACTION  GO |

4.1.10. sp\_ConfirmOrder

|  |
| --- |
| CREATE PROC [dbo].[sp\_ConfirmOrder] @idBill int  AS  BEGIN TRANSACTION  UPDATE Orders SET Status = 1 WHERE id = @idBill  COMMIT TRANSACTION  GO |

4.1.11. sp\_DeleteCategory

|  |
| --- |
| CREATE PROC [dbo].[sp\_DeleteCategory] @id int, @name NVARCHAR(50)  AS  IF NOT EXISTS (SELECT \* FROM Watches\_Categories WHERE id\_Category = @id)  BEGIN  DELETE Categories WHERE id = @id AND Name = @name  END  ELSE  ROLLBACK  GO |

4.1.12. sp\_DeleteFirm

|  |
| --- |
| CREATE PROC [dbo].[sp\_DeleteFirm] @id int, @name NVARCHAR(50)  AS  IF NOT EXISTS (SELECT \* FROM Watches WHERE id\_Firms = @id)  BEGIN  DELETE Firms WHERE id = @id AND Name = @name  END  ELSE  ROLLBACK  GO |

### 4.1.13. sp\_DeleteSex

|  |
| --- |
| CREATE PROC [dbo].[sp\_DeleteSex] @id int, @name NVARCHAR(50)  AS  IF NOT EXISTS (SELECT \* FROM Watches WHERE id\_Sex = @id)  BEGIN  DELETE Sex WHERE id = @id AND Name = @name  END  ELSE  ROLLBACK  GO |

4.1.14. sp\_GuiMaXacNhan

|  |
| --- |
| CREATE PROC [dbo].[sp\_GuiMaXacNhan] @email VARCHAR(50), @code VARCHAR(6) OUTPUT  AS  BEGIN  DECLARE @main NVARCHAR(200)  SET @code = (CONVERT(NUMERIC(6,0),RAND() \* 899999) + 100000)  SET @main= 'Enter the code below to confirm your password reset' + CHAR(13) + @code  EXECUTE msdb.dbo.sp\_send\_dbmail  @profile\_name = 'BeeWatch Supporter',  @recipients = @Email,  @subject = 'Reset password',  @body = @main;  END  GO |

4.1.15. sp\_InsertCategories

|  |
| --- |
| CREATE PROC [dbo].[sp\_InserCategories] @name NVARCHAR(50), @detail nvarchar(MAX)  AS  BEGIN  INSERT INTO Categories(Name, Detail) VALUES (@name, @detail);  END  GO |

4.1.16. sp\_InserFirm

|  |
| --- |
| CREATE PROC [dbo].[sp\_InserFirm] @newName NVARCHAR(50), @url nvarchar(MAX)  AS  BEGIN  INSERT INTO Firms(Name, Url) VALUES (@newName, @url);  END  GO |

### 4.1.17. sp\_InsertAccount

|  |
| --- |
| CREATE PROC [dbo].[sp\_InsertAccount] @username nvarchar(50), @password nvarchar(50), @avt nvarchar(MAX),  @email nvarchar(50), @fullname nvarchar(50), @sex nvarchar(10),  @address\_province nchar(10), @address\_district nchar(10), @address\_detail nvarchar(max)  AS  SET NOCOUNT ON  BEGIN  IF EXISTS (SELECT id FROM Accounts WHERE Email = @email)  BEGIN  RAISERROR('Email', 16, 1);  ROLLBACK  END    IF EXISTS (SELECT id FROM Accounts WHERE Username = @username)  BEGIN  RAISERROR('Email', 16, 1);  ROLLBACK  END  ELSE  DECLARE @id int  INSERT INTO Accounts(Username, Password, Url\_Image\_Avatar, Email, Fullname, Sex, id\_Account\_Type, IsLock, Status, Point)  VALUES(@username, @password, @avt, @email, @fullname, @sex, 2, 0, 0, 0)  SET @id = (SELECT id FROM Accounts WHERE Username = @username)  INSERT INTO Address(id\_Account, id\_Province, id\_District, AddressDetail)  VALUES (@id, @address\_province, @address\_district, @address\_detail)  RETURN @id  END  GO |

Code sử dụng Procedure:

|  |
| --- |
| [Route("api/Accounts/Register")]  [HttpPost]          [ResponseType(typeof(view\_Account))]          public HttpResponseMessage Register(Accounts accounts)          {              int id = 0;              try              {                  var usernameParam = new SqlParameter("username", SqlDbType.NVarChar, 50);                  usernameParam.Value = accounts.Username;                  var passwordParam = new SqlParameter("password", SqlDbType.NVarChar, 50);                  passwordParam.Value = accounts.Password;                  var avtParam = new SqlParameter("avt", SqlDbType.NVarChar, -1);                  avtParam.Value = (accounts.Url\_Image\_Avatar == null ? "" : accounts.Url\_Image\_Avatar);                  var emailParam = new SqlParameter("email", SqlDbType.NVarChar, 50);                  emailParam.Value = accounts.Email;                  var fullnameParam = new SqlParameter("fullname", SqlDbType.NVarChar, 50);                  fullnameParam.Value = accounts.Fullname;                  var sexParam = new SqlParameter("sex", SqlDbType.NVarChar, 10);                  sexParam.Value = (accounts.Sex == null ? "" : accounts.Sex);                  var address\_provinceParam = new SqlParameter("address\_province", SqlDbType.NChar, 10);                  address\_provinceParam.Value = accounts.Address.FirstOrDefault().id\_Province;                  var address\_districtParam = new SqlParameter("address\_district", SqlDbType.NChar, 10);                  address\_districtParam.Value = accounts.Address.FirstOrDefault().id\_District;                  var address\_detailParam = new SqlParameter("address\_detail", SqlDbType.NVarChar, -1);                  address\_detailParam.Value = (accounts.Address.FirstOrDefault().AddressDetail == null ? "" : accounts.Address.FirstOrDefault().AddressDetail);                  id = db.Database.ExecuteSqlCommand("EXEC [dbo].[sp\_InsertAccount] @username, @password, @avt, @email, @fullname, @sex, @address\_province, @address\_district, @address\_detail", usernameParam, passwordParam, avtParam, emailParam, fullnameParam, sexParam, address\_provinceParam, address\_districtParam, address\_detailParam);              }              catch(Exception e)              {                  return Request.CreateResponse(HttpStatusCode.BadRequest,e.Message);              }              view\_Account result = db.view\_Account.Where(x => x.id == id).FirstOrDefault();                if (result == null)              {                  return Request.CreateResponse(HttpStatusCode.NotFound);              }              return Request.CreateResponse(HttpStatusCode.OK,result);          } |

4.1.18. sp\_InsertSex

|  |
| --- |
| CREATE PROC [dbo].[sp\_InsertSex] @new NVARCHAR(50)  AS  BEGIN  INSERT INTO Sex(Name) VALUES (@new);  END  GO |

### **4.1.19. sp\_InsertDiscount**

|  |
| --- |
| CREATE PROC [dbo].[sp\_InsertDiscount] @code nvarchar(10), @id\_Type int, @value\_percent float, @value\_sub float,  @datefrom date, @dateto date  AS  BEGIN  INSERT INTO Discounts(Code, id\_Type, [Value(%)], [Value(-)], DateFrom, DateTo)  VALUES(@code, @id\_Type, @value\_percent, @value\_sub, @datefrom, @dateto)  END  GO |

4.1.20. sp\_InsertNewOrder

|  |
| --- |
| CREATE PROC [dbo].[sp\_InsertNewOrder] @id\_shop int, @id\_account int, @address\_province int, @address\_district int, @address\_detail nvarchar(max),  @count int, @sum float, @ship\_fee float, @id\_discount int, @payment nvarchar(50)  AS  SET NOCOUNT ON  BEGIN  INSERT INTO Orders(id\_Accounts, id\_Shop, Address\_Province, Address\_District, AddressDetail, Count, Sum, Ship\_fee, id\_Discount, Payment, Date\_Create, Status)  VALUES(@id\_account, @id\_shop, @address\_province, @address\_district, @address\_detail,  @count, @sum, @ship\_fee, @id\_discount, @payment, GETDATE(), 0)  END  GO |

4.1.21. sp\_InsertOrderDetails

|  |
| --- |
| CREATE PROC [dbo].[sp\_InsertOrderDetails] @idCart int, @idWatch int, @count int  AS  BEGIN  IF EXISTS (SELECT \*  FROM CartDetails INNER JOIN Carts ON CartDetails.id\_Cart = Carts.id  WHERE id\_Cart = @idCart AND id\_Watch = @idWatch)  BEGIN  UPDATE CartDetails SET Count = Count + 1  WHERE id\_Cart = @idCart AND id\_Watch = @idWatch  END  ELSE INSERT INTO CartDetails(id\_Cart, id\_Watch, Count) VALUES(@idCart, @idWatch, 1)  END  GO |

4.1.22. sp\_InsertTypeDiscount

|  |
| --- |
| CREATE PROC [dbo].[sp\_InsertTypeDiscount] @min\_bill float, @max\_discount float, @payment\_methods nvarchar(20)  AS  BEGIN  INSERT INTO TypeDiscounts(Min\_Bill, Max\_Discount, Payment\_Method)  VALUES(@min\_bill, @max\_discount, @payment\_methods)  END  GO |

4.1.23. sp\_InsertWatch

|  |
| --- |
| CREATE PROC [dbo].[sp\_InsertWatch] @name nvarchar(50), @id\_Sex int, @id\_Firms int, @image nvarchar(MAX),  @price float, @count int, @information nvarchar(MAX), @useradd nvarchar(50)  AS  BEGIN  INSERT INTO Watches(Name, id\_Sex, id\_Firms, Url\_Image, Price, Count, Information, IsLock, IsCheck, Create\_By)  VALUES(@name, @id\_Sex, @id\_Firms, @image, @price, @count, @information, 0, 0, @useradd)  END  GO |

4.1.24. sp\_InsertWatches\_Categories

|  |
| --- |
| CREATE PROC [dbo].[sp\_InserWatches\_Categories] @idwatch int, @idcat int  AS  BEGIN  INSERT INTO Watches\_Categories(id\_Category, id\_Watch) VALUES (@idcat, @idwatch);  END  GO |

### **4.1.25. sp\_LockAccount**

|  |
| --- |
| CREATE PROC [dbo].[sp\_LockAccount] @id int, @name NVARCHAR(50)  AS  IF EXISTS (SELECT \* FROM Accounts WHERE id = @id and Username = @name)  BEGIN TRANSACTION  UPDATE Accounts  SET IsLock = 'True'  WHERE id = @id AND Username = @name  COMMIT TRANSACTION  GO |

4.1.26. sp\_LockWatch

|  |
| --- |
| CREATE PROC [dbo].[sp\_LockWatch] @id int, @name NVARCHAR(50)  AS  IF EXISTS (SELECT \* FROM Watches WHERE id = @id and Name = @name)  BEGIN TRANSACTION  UPDATE Watches  SET IsLock = 'True'  WHERE id = @id AND Name = @name  COMMIT TRANSACTION  GO |

4.1.27. sp\_Logout

|  |
| --- |
| CREATE PROCEDURE [dbo].[sp\_Logout] @username nvarchar(50), @password nvarchar(50)  AS  BEGIN  UPDATE Accounts  SET Status = 0  WHERE Username = @username AND Password = @password  END  GO |

4.1.28. sp\_RateWatch

|  |
| --- |
| CREATE PROC [dbo].[sp\_RateWatch] @idWatch int  AS  SET NOCOUNT ON  BEGIN  DECLARE @Count INT, @Sum Float  SET @Count = (SELECT COUNT(idAccount)  FROM view\_OrderDetail  WHERE id\_Watches = @idWatch)  SET @Sum = (SELECT SUM(Rate)  FROM view\_OrderDetail  WHERE id\_Watches = @idWatch)  UPDATE Watches  SET Rate = @Sum/@Count  WHERE id = @idWatch  END  GO |

4.1.29. sp\_SearchWatch

|  |
| --- |
| CREATE PROC [dbo].[sp\_SearchWatch] @name NVARCHAR(100)  AS  SET NOCOUNT ON  BEGIN  DECLARE @id\_firm int, @id\_sex int  SET @id\_firm = (SELECT id\_Firms  FROM Watches  WHERE Name = @name AND IsLock = 0 AND IsExist = 1)  SET @id\_sex = (SELECT id\_Sex  FROM Watches  WHERE Name = @name AND IsLock = 0 AND IsExist = 1)  SELECT id, id\_Firms, id\_Sex, id\_Shop, Url\_Image, Price, Information, Rate  FROM Watches  WHERE Name LIKE '%'+@name+'%' AND IsLock = 0 AND IsExist = 1  SELECT id, id\_Firms, id\_Sex, id\_Shop, Url\_Image, Price, Information, Rate  FROM Watches  WHERE id\_Firms = @id\_firm AND IsLock = 0 AND IsExist = 1  SELECT id, id\_Firms, id\_Sex, id\_Shop, Url\_Image, Price, Information, Rate  FROM Watches  WHERE id\_Sex = @id\_sex AND IsLock = 0 AND IsExist = 1  END  GO |

4.1.30. sp\_ThongKeDoanhThu

|  |
| --- |
| CREATE PROCEDURE [dbo].[sp\_ThongKeDoanhThu] @idShop int, @DateFrom date, @DateTo date  AS  BEGIN  SELECT id\_Shop, Orders.id, id\_Accounts, Username, Orders.Date\_Create, Sum, Payment  FROM Orders, Accounts, Discounts  WHERE id\_Shop = @idShop  AND Orders.id\_Accounts = Accounts.id  AND Orders.id\_Discount = Discounts.id  AND Orders.Date\_Create > @DateFrom AND Orders.Date\_Create < @DateTo  AND Orders.Status = 3 ---3 la da giao hang  END  GO |

4.1.31. sp\_UpdateFirm

|  |
| --- |
| CREATE PROC [dbo].[sp\_UpdateFirm] @id INT, @name NVARCHAR(50), @url nvarchar(MAX)  AS  BEGIN TRANSACTION  UPDATE Firms  SET name = @name, Url = @url  WHERE id = @id  COMMIT TRANSACTION  GO |

4.1.32. sp\_UpdatePhone

|  |
| --- |
| CREATE PROCEDURE [dbo].[sp\_UpdatePhone] @idAcc int, @idPhone int, @phone nchar(10)  AS  BEGIN  UPDATE Phone  SET Phone = @phone  WHERE id = @idPhone AND id\_Account = @idAcc  END  GO |

4.1.33. sp\_UpdateSex

|  |
| --- |
| CREATE PROC [dbo].[sp\_UpdateSex] @id INT, @name NVARCHAR(50)  AS  BEGIN TRANSACTION  UPDATE Sex  SET name = @name  WHERE id = @id  COMMIT TRANSACTION  GO |

* + 1. sp\_InsertCartDetails

|  |
| --- |
| CREATE PROC sp\_InsertCartDetails @idCart int, @idWatch int  AS  BEGIN  IF EXISTS (SELECT \*  FROM CartDetails INNER JOIN Carts ON CartDetails.id\_Cart = Carts.id  WHERE id\_Cart = @idCart AND id\_Watch = @idWatch)  BEGIN  UPDATE CartDetails SET Count = Count + 1  WHERE id\_Cart = @idCart AND id\_Watch = @idWatch  UPDATE Carts SET Count = Count + 1  WHERE id = @idCart  END  ELSE INSERT INTO CartDetails(id\_Cart, id\_Watch, Count) VALUES(@idCart, @idWatch, 1)  END  GO |

Code sử dụng procedure

|  |
| --- |
| //Cộng số lượng của Item trong Cart Detail lên 1 (thao tác '+' trên Giao diện Giỏ Hàng)  // Input: id\_Cart [FromBody]  // id\_CartDetail [FromBody]  // Output: Ok(1) Update Bản ghi  // Ok(2) Thêm Bản ghi  // BadRequest Lỗi SQL  // NotFound Không tìm thấy  [Route("api/CartDetails/AddCountItem/{id\_Cart:int}/{id\_CartDetail:int}")]  [HttpPost]  [ResponseType(typeof(int))]  public IHttpActionResult AddCountItem([FromUri] int id\_Cart, [FromUri] int id\_CartDetail)  {  if (id\_Cart <= 0 || id\_CartDetail <= 0)  {  return BadRequest();  }  if (!ModelState.IsValid)  {  return BadRequest(ModelState);  }  var idCartParam = new SqlParameter("idCart", SqlDbType.Int);  idCartParam.Value = id\_Cart;  var idWatchParam = new SqlParameter("idWatch", SqlDbType.Int);  idWatchParam.Value = id\_CartDetail;  int result;  try  {  result = db.Database.ExecuteSqlCommand("EXEC [dbo].[sp\_InsertCartDetails] @idCart, @idWatch",  idCartParam, id\_CartDetail);  }  catch (Exception e)  {  return BadRequest(message: e.Message);  }  switch (result)  {  case 1:  return Ok(1);  case 2:  return Ok(2);  case 0:  return NotFound();  }  return BadRequest();  } |

### **4.1.35. sp\_DropCartDetails**

|  |
| --- |
| CREATE PROC sp\_DropCartDetails @idCart int, @idWatch int  AS  BEGIN  IF EXISTS (SELECT \*  FROM CartDetails INNER JOIN Carts ON CartDetails.id\_Cart = Carts.id  WHERE id\_Cart = @idCart AND id\_Watch = @idWatch AND CartDetails.Count > 1)  BEGIN  UPDATE CartDetails SET Count = Count - 1  WHERE id\_Cart = @idCart AND id\_Watch = @idWatch  UPDATE Carts SET Count = Count - 1  WHERE id = @idCart  END  ELSE IF EXISTS (SELECT \*  FROM CartDetails INNER JOIN Carts ON CartDetails.id\_Cart = Carts.id  WHERE id\_Cart = @idCart AND id\_Watch = @idWatch AND CartDetails.Count = 1)  BEGIN  DELETE FROM CartDetails  WHERE id\_Cart = @idCart AND id\_Watch = @idWatch  UPDATE Carts SET Count = Count - 1  WHERE id = @idCart  END  END  GO |

Code sử dụng procedure

|  |
| --- |
| //Trừ số lượng của Item trong Cart Detail xuống 1 (thao tác '-' trên Giao diện Giỏ Hàng)  // Input: id\_Cart [FromBody]  // id\_CartDetail [FromBody]  // Output: Ok(1) Trừ Bản ghi  // Ok(2) Xóa Bản ghi  // BadRequest Lỗi SQL  // NotFound Không tìm thấy  [Route("api/CartDetails/SubCountItem/{id\_Cart:int}/{id\_CartDetail:int}")]  [HttpPost]  [ResponseType(typeof(int))]  public IHttpActionResult SubCountItem([FromUri] int id\_Cart, [FromUri] int id\_CartDetail)  {  if (id\_Cart <= 0 || id\_CartDetail <= 0)  {  return BadRequest();  }  if (!ModelState.IsValid)  {  return BadRequest(ModelState);  }  var idCartParam = new SqlParameter("idCart", SqlDbType.Int);  idCartParam.Value = id\_Cart;  var idWatchParam = new SqlParameter("idWatch", SqlDbType.Int);  idWatchParam.Value = id\_CartDetail;  int result;  try  {  result = db.Database.ExecuteSqlCommand("EXEC [dbo].[sp\_DropCartDetails] @idCart, @idWatch",  idCartParam, id\_CartDetail);  }  catch (Exception e)  {  return BadRequest(message: e.Message);  }  switch (result)  {  case 1:  return Ok(1);  case 2:  return Ok(2);  case 0:  return NotFound();  }  return BadRequest();  } |

1. Function

4.2.1. fn\_CheckLockUser

|  |
| --- |
| CREATE FUNCTION fn\_CheckLockUser (@lock bit)  RETURNS TABLE  AS  RETURN(SELECT \* FROM Accounts WHERE IsLock = @lock)  GO |

4.2.2. fn\_LoadBill

|  |
| --- |
| CREATE FUNCTION fn\_LoadBill(@idAccount int, @username nvarchar(50))  RETURNS TABLE  AS  RETURN (SELECT Orders.id AS ID\_Bill, Orders.Date\_Create, Orders.Payment, Orders.Ship\_fee, Orders.Sum, Accounts.id AS ID\_ACC, Accounts.Username  FROM Orders INNER JOIN Accounts ON Orders.id\_Accounts = Accounts.id  WHERE Accounts.id = @idAccount AND Username = @username AND IsLock = 0)  GO |

4.2.3. fn\_LoadBillforAccept

|  |
| --- |
| CREATE FUNCTION fn\_LoadBillforAccept()  RETURNS TABLE  AS  RETURN (SELECT \* FROM Orders WHERE Status = 0)  GO |

4.2.4. fn\_LoadBillforShipper

|  |
| --- |
| CREATE FUNCTION fn\_LoadBillforShipper()  RETURNS TABLE  AS  RETURN (SELECT \* FROM Orders WHERE Status = 1)  GO |

4.2.5. fn\_LoadBillforCancel

|  |
| --- |
| CREATE FUNCTION fn\_LoadBillforCancel()  RETURNS TABLE  AS  RETURN (SELECT \* FROM Orders WHERE Status = 4)  GO |

4.2.6. fn\_LoadFirm

|  |
| --- |
| ---Lay thong tin dong ho tu Hang  CREATE FUNCTION fn\_LoadFirm(@id int, @name nvarchar(50))  RETURNS TABLE  AS RETURN  (SELECT Watches.Name as WatchName, Watches.Price, Watches.Rate, Watches.Url\_Image, Watches.Information,  Watches.id\_Firms, Firms.Name as FirmName  FROM Watches INNER JOIN Firms ON Watches.id\_Firms = Firms.id  WHERE Firms.id = @id AND Firms.Name = @name  AND Watches.IsLock = 0 AND Watches.IsExist = 1)  GO |

4.2.7. fn\_LoadSex

|  |
| --- |
| ---Lay thong tin dong ho tu Gioi tinh  CREATE FUNCTION fn\_LoadSex(@id int, @name nvarchar(50))  RETURNS TABLE  AS RETURN  (SELECT Watches.Name as WatchName, Watches.Price, Watches.Rate, Watches.Url\_Image, Watches.Information,  Watches.id\_Sex, Sex.Name as SexName  FROM Watches INNER JOIN Sex ON Watches.id\_Sex = Sex.id  WHERE Sex.id = @id AND Sex.Name = @name  AND Watches.IsLock = 0 AND Watches.IsExist = 1)  GO |

4.2.8. fn\_LoadWatchBill

|  |
| --- |
| CREATE FUNCTION fn\_LoadWatchBill()  RETURNS TABLE  AS  RETURN (SELECT Orders.id AS ID\_Bill, OrderDetails.id\_Watches, Watches.Name AS WatchName,  OrderDetails.Price, OrderDetails.Count, Orders.id\_Accounts, Orders.Address\_District,  Orders.Address\_Province, Orders.AddressDetail  FROM Orders, OrderDetails, Accounts, Watches  WHERE Orders.id\_Accounts = Accounts.id  AND Orders.id = OrderDetails.id\_Order  AND OrderDetails.id\_Watches = Watches.id  AND Orders.Status = 1)  GO |

4.2.9. fn\_PhanTrang

|  |
| --- |
| CREATE FUNCTION fn\_PhanTrang(@num\_offset int, @num\_next int)  RETURNS TABLE  AS  RETURN  SELECT \* FROM Watches  ORDER BY id OFFSET (@num\_offset \* @num\_next - @num\_next) ROWS FETCH NEXT @num\_next ROWS ONLY;  GO |

4.2.10. fn\_SearchWatch

|  |
| --- |
| ---Tim ten dong ho khi nhap gan dung  CREATE FUNCTION fn\_SearchWatch(@Name NVARCHAR(100))  RETURNS TABLE  AS RETURN SELECT \* FROM Watches WHERE Name LIKE '%'+@Name+'%'  GO |

4.2.11. fn\_UserSeller

|  |
| --- |
| CREATE FUNCTION fn\_UserSeller()  RETURNS TABLE  AS  RETURN (SELECT Accounts.id, Accounts.Username, Accounts.Email, Accounts.Url\_Image\_Avatar,  Sellers.Identitycard, Sellers.Bankcard  FROM Accounts INNER JOIN Sellers ON Accounts.id = Sellers.id  WHERE id\_Account\_Type = 2 AND Status = 0)  GO |

4.2.12. fn\_WatchMaxDeal

|  |
| --- |
| CREATE FUNCTION fn\_WatchMaxDeal()  RETURNS TABLE  AS  RETURN (  SELECT Watches.id, Watches.Name, Watches.Url\_Image, Watches.Price, SUM(CartDetails.Count) AS Tong  FROM Watches INNER JOIN CartDetails ON Watches.id = CartDetails.id\_Watch  GROUP BY Watches.id, Watches.Name, Watches.Url\_Image, Watches.Price  );  GO |

4.2.13. fn\_GetPrice

|  |
| --- |
| CREATE FUNCTION fn\_GetPrice(@id int, @name nvarchar(50))  RETURNS FLOAT  AS  BEGIN  RETURN (SELECT Price FROM Watches WHERE id = @id AND Name = @name  AND Watches.IsLock = 0 AND Watches.IsExist = 1)  END  GO |

4.2.14. fn\_CheckBan

|  |
| --- |
| CREATE FUNCTION fn\_CheckBan (@username nvarchar(50))  RETURNS BIT  AS  BEGIN  DECLARE @check BIT  DECLARE @count INT = (SELECT COUNT(\*)  FROM Accounts  WHERE Username = @username AND IsLock = 1)  IF(@count > 0)  SET @check = 1  ELSE SET @check = 0  RETURN @check  END  GO |

4.2.15. fn\_CheckLogin

|  |
| --- |
| CREATE FUNCTION fn\_CheckLogin (@username NVARCHAR(50), @password NVARCHAR(50))  RETURNS INT  AS  BEGIN  DECLARE @id INT  IF EXISTS (SELECT id FROM Accounts WHERE Username = @username AND Password = @password)  BEGIN  SET @id = (SELECT id FROM Accounts WHERE Username = @username AND Password = @password)  END  ELSE  BEGIN  SET @id = 0  END  RETURN @id  END  GO |

4.2.16. fn\_GetMaxBillID

|  |
| --- |
| CREATE FUNCTION fn\_GetMaxBillID()  RETURNS INT  AS  BEGIN  RETURN (SELECT MAX(id) FROM Orders)  END  GO |

4.2.17. fn\_GetPrice

|  |
| --- |
| CREATE FUNCTION fn\_GetPrice(@id int, @name nvarchar(50))  RETURNS FLOAT  AS  BEGIN  RETURN (SELECT Price FROM Watches WHERE id = @id AND Name = @name  AND Watches.IsLock = 0 AND Watches.IsExist = 1)  END  GO |

4.2.18. fn\_GetPriceTotal

|  |
| --- |
| CREATE FUNCTION fn\_GetPriceTotal(@sum float, @discount float)  RETURNS FLOAT  AS  BEGIN  DECLARE @ret FLOAT;  IF(@discount < 0)  SET @ret = (@sum - @discount)  ELSE SET @ret = (@sum \* @discount)  RETURN @ret  END  GO |

1. Trigger

4.3.1. tg\_CheckAddUser

|  |
| --- |
| CREATE TRIGGER [dbo].[tg\_CheckAddUser] ON [dbo].[Accounts]  FOR INSERT, UPDATE  AS  BEGIN  DECLARE @username NVARCHAR(50), @email CHAR(9), @fullname nvarchar(50),@sex nchar(10), @address nvarchar(100), @phone char(10)  SELECT @username = Inserted.Username, @email = Inserted.Email, @fullname = Inserted.Fullname,  @sex = Inserted.Sex  FROM Inserted  IF (LEN(@username)<=0 OR LEN(@email)<=0 OR LEN(@fullname)<=0 OR LEN(@sex)<=0)  BEGIN  RAISERROR('Vui long nhap thong tin', 16, 1);  ROLLBACK;  END  END;  GO |

4.3.2. tg\_AddOrder

|  |
| --- |
| CREATE TRIGGER [dbo].[tg\_AddOrder] on [dbo].[OrderDetails] AFTER INSERT  AS  BEGIN TRANSACTION  UPDATE Watches  SET Count = Watches.Count - ( SELECT Count  FROM inserted  WHERE id = Watches.id)  FROM Watches JOIN inserted ON Watches.id = inserted.id\_Watches  COMMIT TRANSACTION  GO |

4.3.3. tg\_CancelOrder

|  |
| --- |
| CREATE TRIGGER [dbo].[tg\_CancelOrder] on [dbo].[Orders] AFTER UPDATE  AS  BEGIN TRANSACTION  UPDATE Watches  SET Count = Watches.Count + (SELECT Count  FROM deleted  WHERE id = Watches.id)  FROM Watches JOIN deleted ON Watches.id = deleted.id  COMMIT TRANSACTION  GO |

### 4.3.4. tg\_InsertOrder

|  |
| --- |
| CREATE TRIGGER [dbo].[tg\_InsertOrder] ON [dbo].[Orders] AFTER INSERT  AS  BEGIN TRANSACTION  UPDATE Orders  SET Date\_Create = GETDATE()  FROM Orders INNER JOIN inserted ON Orders.id = inserted.id  COMMIT TRANSACTION  GO |

4.3.5. tg\_InsertWatch

|  |
| --- |
| CREATE TRIGGER [dbo].[tg\_InsertWatch] ON [dbo].[Watches] AFTER INSERT  AS  BEGIN TRAN  UPDATE Watches  SET Date\_Create = GETDATE()  FROM Watches INNER JOIN inserted ON Watches.id = inserted.id  COMMIT TRAN  GO |

### 4.3.6. tg\_RateWatch

|  |
| --- |
| CREATE TRIGGER [dbo].[tg\_RateWatch] ON [dbo].[OrderDetails] AFTER UPDATE  AS  BEGIN TRANSACTION  DECLARE @count INT, @sum FLOAT;  DECLARE @id\_updated INT;  SET @id\_updated = (SELECT id\_Order  FROM deleted)    SET @count = (SELECT COUNT(id\_Order)  FROM OrderDetails INNER JOIN Orders ON OrderDetails.id\_Order = Orders.id  WHERE OrderDetails.id\_Watches = @id\_updated  AND Orders.Status = 3 AND Rate != NULL)  SET @sum = (SELECT SUM(Rate)  FROM OrderDetails  WHERE OrderDetails.id\_Watches = @id\_updated)  UPDATE Watches SET Rate = @sum / @count  FROM Watches JOIN deleted ON Watches.id = deleted.id\_Watches  COMMIT TRANSACTION  GO |

4.3.7. tg\_InsertAccount

|  |
| --- |
| CREATE TRIGGER [dbo].[tg\_InsertAccount] ON [dbo].[Accounts] AFTER INSERT  AS  BEGIN TRANSACTION  UPDATE Accounts  SET Date\_Create = GETDATE()  FROM Accounts INNER JOIN inserted ON Accounts.id = inserted.id  COMMIT TRANSACTION  GO |

1. View

4.4.1. view\_Account

|  |
| --- |
| CREATE VIEW [dbo].[view\_Account]  AS  SELECT DISTINCT Accounts.id\_Account\_Type, Account\_Type.Name AS AccountTypeName, Accounts.Username , Accounts.Fullname, Accounts.Date\_Create,  Accounts.Email, Accounts.Password, Accounts.Sex, Accounts.Url\_Image\_Avatar,  Accounts.IsLock, Accounts.Point, Accounts.Time\_LastAccess,  Address\_Province.Name AS ProvinceName, Address.id\_Province, Address\_District.Name AS DistrictName, Address.id\_District,  Address.AddressDetail  FROM (Accounts INNER JOIN Address ON Accounts.id = Address.id\_Account), Address\_District, Address\_Province, Account\_Type  WHERE Address.id\_District = Address\_District.id AND Address.id\_Province = Address\_Province.id  AND Account\_Type.id = Accounts.id\_Account\_Type  GO |

4.4.2. view\_Cart

|  |
| --- |
| CREATE TABLE [dbo].[Carts](  [id] [int] IDENTITY(1,1) NOT NULL,  [Count] [int] NULL,  [Sum] [float] NULL,  CONSTRAINT [PK\_\_Cart\_\_3213E83FFB256D87] PRIMARY KEY CLUSTERED  (  [id] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY]  GO |

4.4.3. view\_CartDetails

|  |
| --- |
| CREATE VIEW [dbo].[view\_CartDetails]  AS  SELECT \*  FROM CartDetails  GO |

4.4.4. view\_Categories

|  |
| --- |
| CREATE VIEW view\_Categories  AS  SELECT id, Name  FROM Categories  WHERE Status = 1  GO |

4.4.5. view\_District

|  |
| --- |
| CREATE VIEW view\_District  AS  SELECT \* FROM Address\_District  GO |

4.4.6. view\_OrderCancel

|  |
| --- |
| CREATE VIEW [dbo].[view\_OrderCancel]  AS  SELECT \*  FROM Orders  WHERE Status = 3  GO |

4.4.7. view\_OrderDetail

|  |
| --- |
| CREATE VIEW view\_OrderDetail  AS  SELECT o.id as idOrder, o.id\_Accounts as idAccount, o.id\_Shop as idShop, o.Date\_Create,  s.Name as ShopName, a.Username as UserName, od.id\_Watches, w.Name as WatchName,  od.Count, od.Price, od.Status, od.Rate  FROM Orders o, OrderDetails od, Shops s, Accounts a, Watches w  WHERE o.id = od.id\_Order AND w.id = od.id\_Watches AND s.id = o.id\_Shop  AND o.id\_Accounts = a.id  GO |

### 4.4.8. view\_Orders

|  |
| --- |
| CREATE VIEW view\_Orders  AS  SELECT \*  FROM Orders  GO |

4.4.9. view\_OrdertoComplete

|  |
| --- |
| CREATE VIEW [dbo].[view\_OrdertoComplete]  AS  SELECT \*  FROM Orders  WHERE Status = 2  GO |

4.4.10. view\_OrdertoConfirm

|  |
| --- |
| CREATE VIEW view\_OrdertoConfirm  AS  SELECT \*  FROM Orders  WHERE Status = 0  GO |

4.4.11. view\_OrdertoShip

|  |
| --- |
| CREATE VIEW [dbo].[view\_OrdertoShip]  AS  SELECT \*  FROM Orders  WHERE Status = 1  GO |

4.4.12. view\_Province

|  |
| --- |
| CREATE VIEW view\_Province  AS  SELECT \* FROM Address\_Province  GO |

4.4.13. view\_Sex

|  |
| --- |
| CREATE VIEW view\_Sex  AS  SELECT id, Name  FROM Sex  WHERE Status = 1  GO |

4.4.14. view\_UserSeller

|  |
| --- |
| CREATE VIEW view\_UserSeller  AS  SELECT Accounts.id, Accounts.Username, Accounts.Email, Accounts.Url\_Image\_Avatar,  Sellers.Identitycard, Sellers.Bankcard  FROM Accounts INNER JOIN Sellers ON Accounts.id = Sellers.id  WHERE id\_Account\_Type = 1 AND Status = 0  GO |

4.4.15. view\_Watches

|  |
| --- |
| CREATE VIEW [dbo].[view\_Watches]  AS  SELECT TOP 10000 Watches.id, Watches.Name AS WatchName, Watches.Count, Watches.Url\_Image, Watches.Price, Watches.Rate,  Watches.Information, Watches.id\_Shop, Watches.Date\_Create,  Sex.Name AS NameofSex, Firms.Name AS NameofFirm,  (SELECT SUM(OrderDetails.Count) AS LuotMua  FROM OrderDetails  WHERE Status = 3) AS LuotMua  FROM Watches FULL OUTER JOIN OrderDetails ON Watches.id = OrderDetails.id\_Watches, Sex, Firms  WHERE Watches.id\_Firms = Firms.id AND Watches.id\_Sex = Sex.id    GROUP BY Watches.id, Watches.Name, Watches.Count, Watches.Url\_Image, Watches.Price, Watches.Rate,  Watches.Information, Watches.id\_Shop, Watches.Date\_Create,  Sex.Name, Firms.Name  ORDER BY LuotMua  GO |

1. Phân quyền

4. 5. 1. Phân quyền Admin

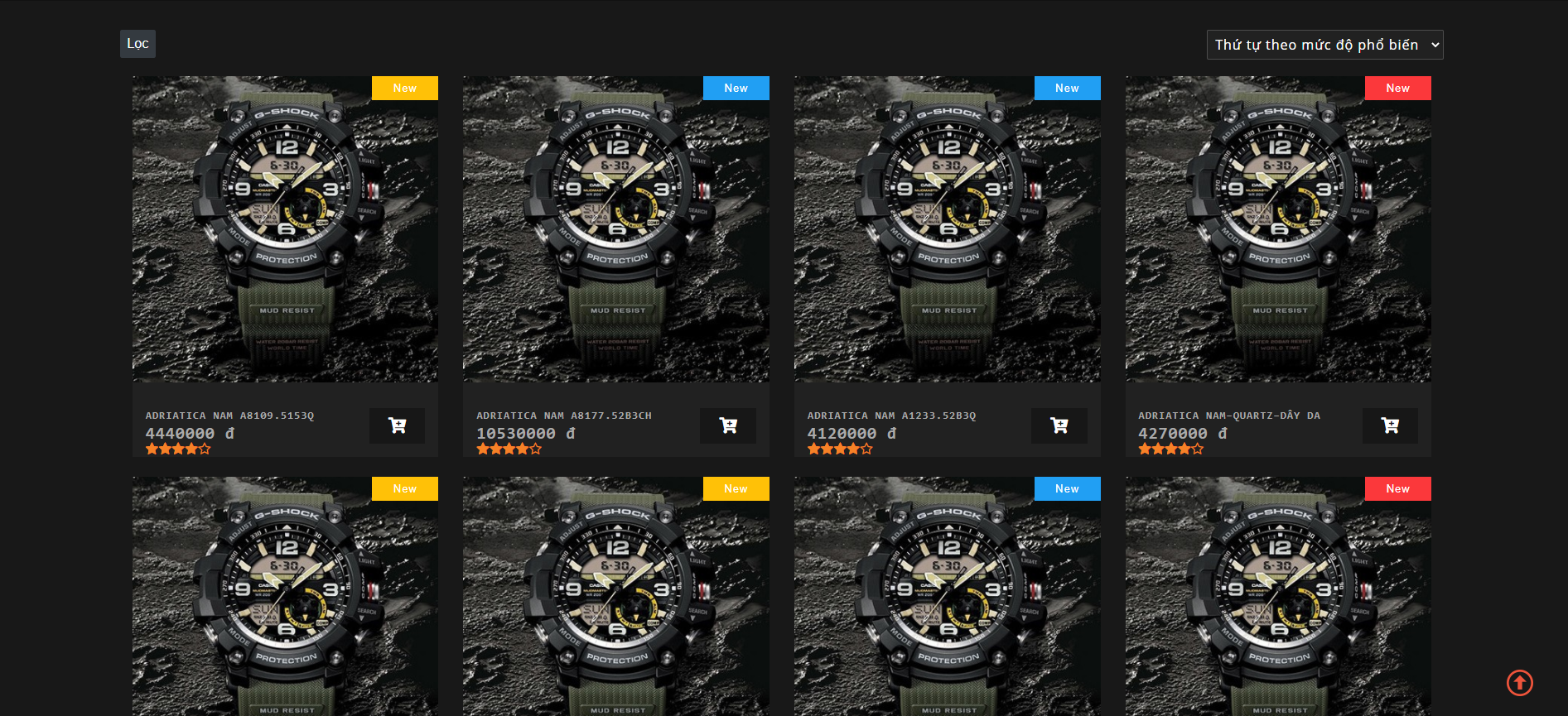
* Các thao tác được phép truy cập:
  + Xem, sửa, xóa các bảng trong CSDL
  + Xem các view trong CSDL
  + Thực thi các thủ tục và hàm
* Code thực thi

|  |
| --- |
| ---Tạo nhóm quyền  CREATE ROLE Admin  ---Tạo log in  CREATE LOGIN vyhuynh18110400 WITH PASSWORD = '123';  ---Tạo user  CREATE USER vyhuynh18110400 FOR LOGIN vyhuynh18110400;  GO  --- Phân quyền user  sp\_addRoleMember 'Admin', 'vyhuynh18110400'  GO  ---Cấp quyền  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Accounts TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Account\_Type TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Action TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Address TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Address\_District TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Address\_Province TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Authoriza TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Carts TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.CartDetails TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Categories TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Discounts TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Firms TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Hot\_Trend TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Image TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.LogInHistories TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.OrderDetails TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Orders TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Phone TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Sellers TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Sex TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Shop\_Seller TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Shops TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.TypeDiscounts TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Watches TO Admin WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE, DELETE ON dbo.Watches\_Categories TO Admin WITH GRANT OPTION  GRANT SELECT ON dbo.view\_Categories TO Admin WITH GRANT OPTION  GRANT SELECT ON dbo.view\_OrderDetail TO Admin WITH GRANT OPTION  GRANT SELECT ON dbo.view\_UserSeller TO Admin WITH GRANT OPTION  GRANT SELECT ON dbo.view\_Sex TO Admin WITH GRANT OPTION  GRANT SELECT ON dbo.view\_Account TO Admin WITH GRANT OPTION  GRANT SELECT ON dbo.view\_Cart TO Admin WITH GRANT OPTION  GRANT SELECT ON dbo.view\_CartDetails TO Admin WITH GRANT OPTION  GRANT SELECT ON dbo.view\_OrderCancel TO Admin WITH GRANT OPTION  GRANT SELECT ON dbo.view\_Orders TO Admin WITH GRANT OPTION  GRANT SELECT ON dbo.view\_OrdertoComplete TO Admin WITH GRANT OPTION  GRANT SELECT ON dbo.view\_OrdertoConfirm TO Admin WITH GRANT OPTION  GRANT SELECT ON dbo.view\_OrdertoShip TO Admin WITH GRANT OPTION  GRANT SELECT ON dbo.view\_Watches TO Admin WITH GRANT OPTION  GRANT SELECT ON dbo.view\_Province TO Admin WITH GRANT OPTION  GRANT SELECT ON dbo.view\_District TO Admin WITH GRANT OPTION  GRANT SELECT ON dbo.view\_AccountType TO Admin WITH GRANT OPTION  GRANT SELECT ON dbo.view\_Action TO Admin WITH GRANT OPTION  GRANT SELECT ON dbo.view\_Authoriza TO Admin WITH GRANT OPTION  ---Thực thi các Procedure  GRANT EXECUTE ON dbo.ChangePassword TO Admin WITH GRANT OPTION  GRANT EXECUTE ON dbo.sp\_AddCartDetail TO Admin WITH GRANT OPTION  GRANT EXECUTE ON dbo.sp\_ChangeInfo TO Admin WITH GRANT OPTION  GRANT EXECUTE ON dbo.sp\_CheckLogin TO Admin WITH GRANT OPTION  GRANT EXECUTE ON dbo.sp\_CheckWatch TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteCategory] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteFirm] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteSex] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_GuiMaXacNhan] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InserCategories] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InserFirm] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertAccount] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertSex] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertDiscount] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertNewOrder] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertOrderDetails] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertTypeDiscount] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertWatch] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InserWatches\_Categories] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_LockAccount] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_LockWatch] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_Logout] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_RateWatch] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_SearchWatch] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_Send] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_ThongKeDoanhThu] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateAccount] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateFirm] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdatePhone] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateSex] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateWatch] TO Admin WITH GRANT OPTION  ---Thực thi Function  GRANT SELECT ON [dbo].[fn\_CheckLockUser] TO Admin WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_LoadBill] TO Admin WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_LoadBillforAccept] TO Admin WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_LoadBillforCancel] TO Admin WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_LoadBillforShipper] TO Admin WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_LoadFirm] TO Admin WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_LoadSex] TO Admin WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_LoadWatchBill] TO Admin WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_SearchWatch] TO Admin WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_UserSeller] TO Admin WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_WatchMaxDeal] TO Admin WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_PhanTrang] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[fn\_CheckBan] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[fn\_GetMaxBillID] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[fn\_GetPrice] TO Admin WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[fn\_GetPriceTotal] TO Admin WITH GRANT OPTION |

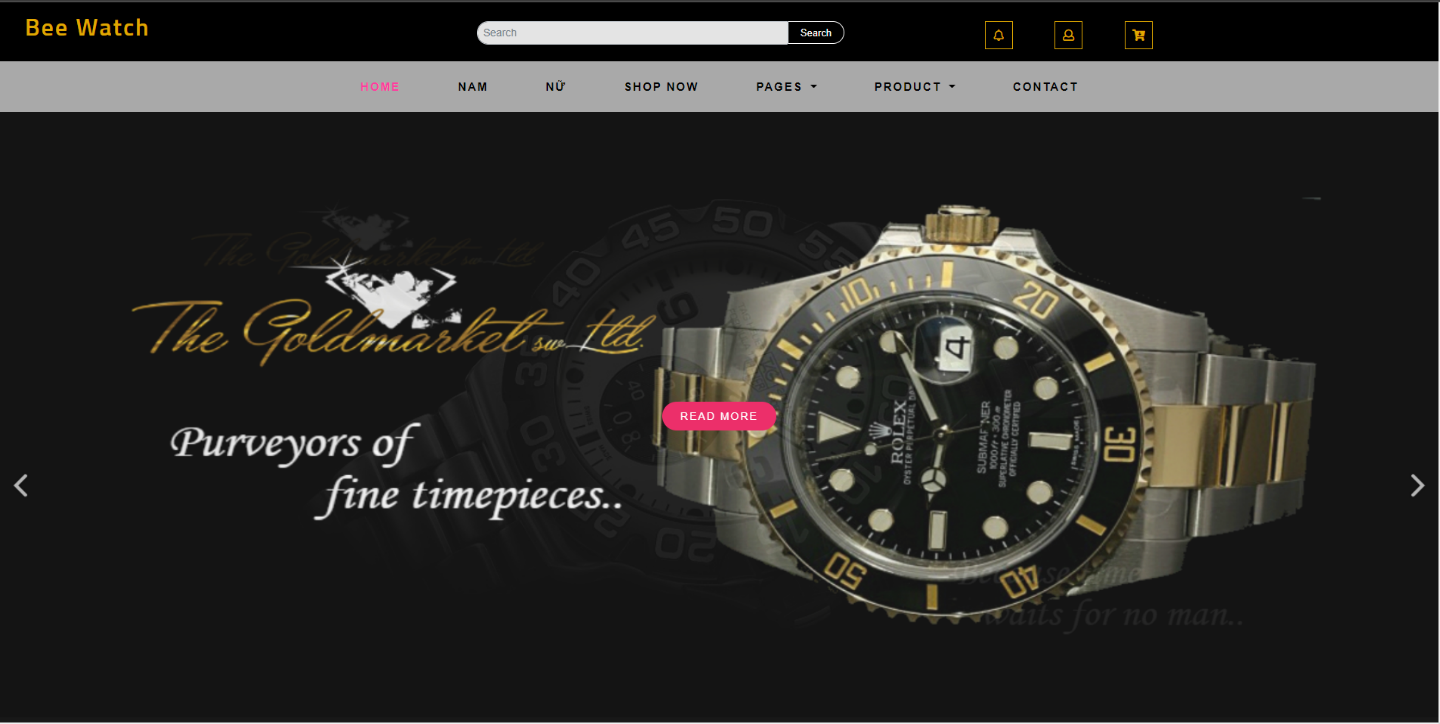
* + 1. Phân quyền Seller
* Các thao tác được phép truy cập:
  + Xem, sửa một số bảng trong CSDL
  + Thực thi một số thủ tục và hàm
* Code thực thi

|  |
| --- |
| ---Tạo nhóm quyền  CREATE ROLE Seller  ---Tạo log in  CREATE LOGIN seller1 WITH PASSWORD = '123';  ---Tạo user  CREATE USER seller1 FOR LOGIN seller1;  GO  --- Phân quyền user  sp\_addRoleMember 'Seller', 'seller1'  GO  ---Cấp quyền  GRANT SELECT, UPDATE ON dbo.Address TO Seller WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE ON dbo.Categories TO Seller WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE ON dbo.Discounts TO Seller WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE ON dbo.Firms TO Seller WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE ON dbo.Image TO Seller WITH GRANT OPTION  GRANT SELECT, UPDATE ON dbo.OrderDetails TO Seller WITH GRANT OPTION  GRANT SELECT, UPDATE ON dbo.Orders TO Seller WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE ON dbo.Phone TO Seller WITH GRANT OPTION  GRANT UPDATE ON dbo.Sellers TO Seller WITH GRANT OPTION  GRANT INSERT, UPDATE, DELETE ON dbo.Sex TO Seller WITH GRANT OPTION  GRANT INSERT, UPDATE ON dbo.Shop\_Seller TO Seller WITH GRANT OPTION  GRANT INSERT, UPDATE ON dbo.Shops TO Seller WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE ON dbo.TypeDiscounts TO Seller WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE ON dbo.Watches TO Seller WITH GRANT OPTION  GRANT SELECT, INSERT, UPDATE ON dbo.Watches\_Categories TO Seller WITH GRANT OPTION  ---Thực thi các Procedure  GRANT EXECUTE ON dbo.ChangePassword TO Seller WITH GRANT OPTION  GRANT EXECUTE ON dbo.sp\_AddCartDetail TO Seller WITH GRANT OPTION  GRANT EXECUTE ON dbo.sp\_ChangeInfo TO Seller WITH GRANT OPTION  GRANT EXECUTE ON dbo.sp\_CheckLogin TO Seller WITH GRANT OPTION  GRANT EXECUTE ON dbo.sp\_CheckWatch TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteCategory] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteFirm] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_DeleteSex] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_GuiMaXacNhan] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InserCategories] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InserFirm] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertAccount] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertSex] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertDiscount] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertNewOrder] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertOrderDetails] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertTypeDiscount] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InsertWatch] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_InserWatches\_Categories] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_LockWatch] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_Logout] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_RateWatch] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_SearchWatch] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_Send] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_ThongKeDoanhThu] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateAccount] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateFirm] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdatePhone] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateSex] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[sp\_UpdateWatch] TO Seller WITH GRANT OPTION  ---Thực thi Function  GRANT SELECT ON [dbo].[fn\_CheckLockUser] TO Seller WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_LoadBill] TO Seller WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_LoadBillforAccept] TO Seller WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_LoadBillforCancel] TO Seller WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_LoadBillforShipper] TO Seller WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_LoadFirm] TO Seller WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_LoadSex] TO Seller WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_LoadWatchBill] TO Seller WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_SearchWatch] TO Seller WITH GRANT OPTION  GRANT SELECT ON [dbo].[fn\_WatchMaxDeal] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[fn\_GetMaxBillID] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[fn\_GetPrice] TO Seller WITH GRANT OPTION  GRANT EXECUTE ON [dbo].[fn\_GetPriceTotal] TO Seller WITH GRANT OPTION |

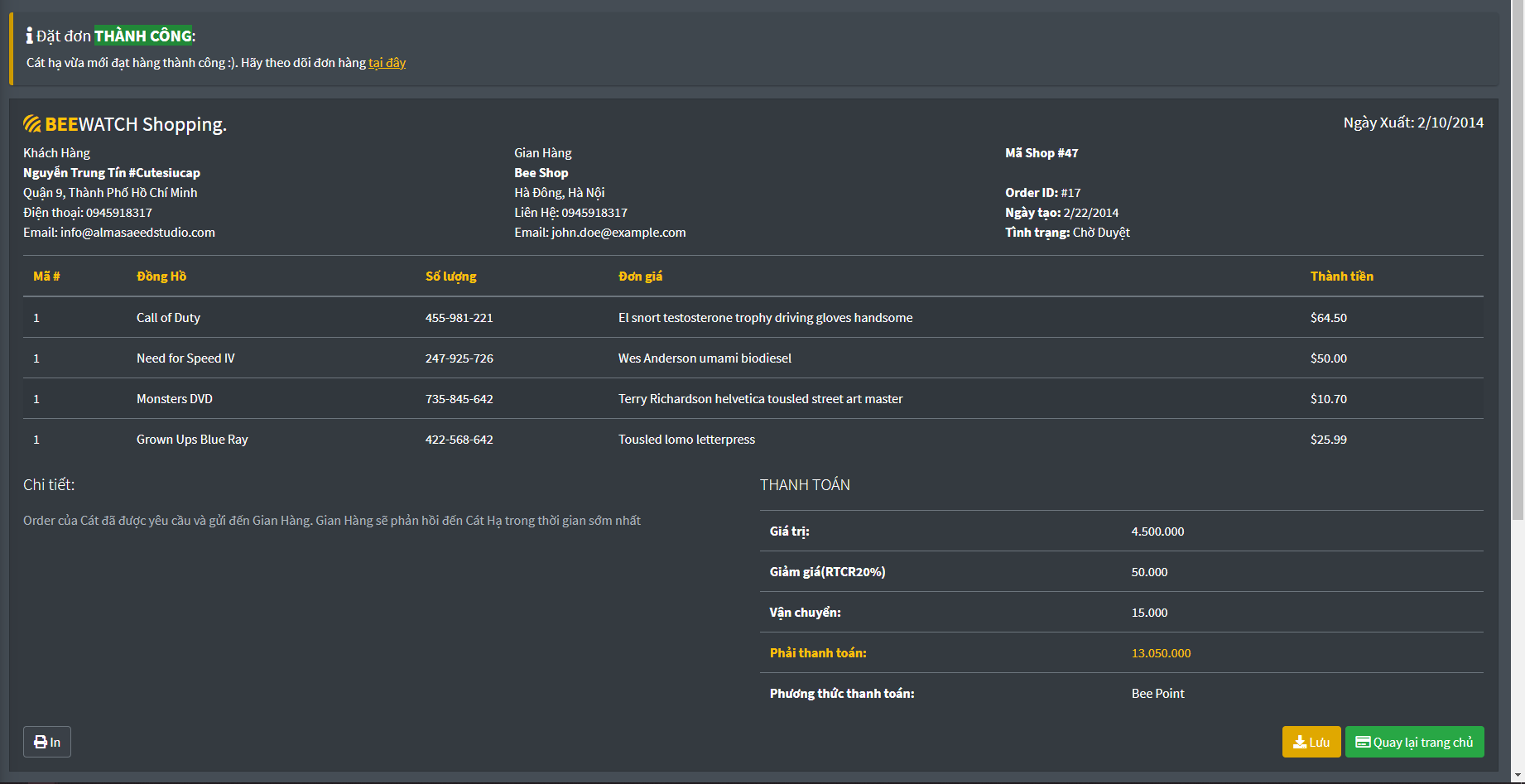
CHƯƠNG 5. CÀI ĐẶT CHƯƠNG TRÌNH

1. Giao diện trang chủ mua hàng

Hình 3. Giao diện trang mua hàng

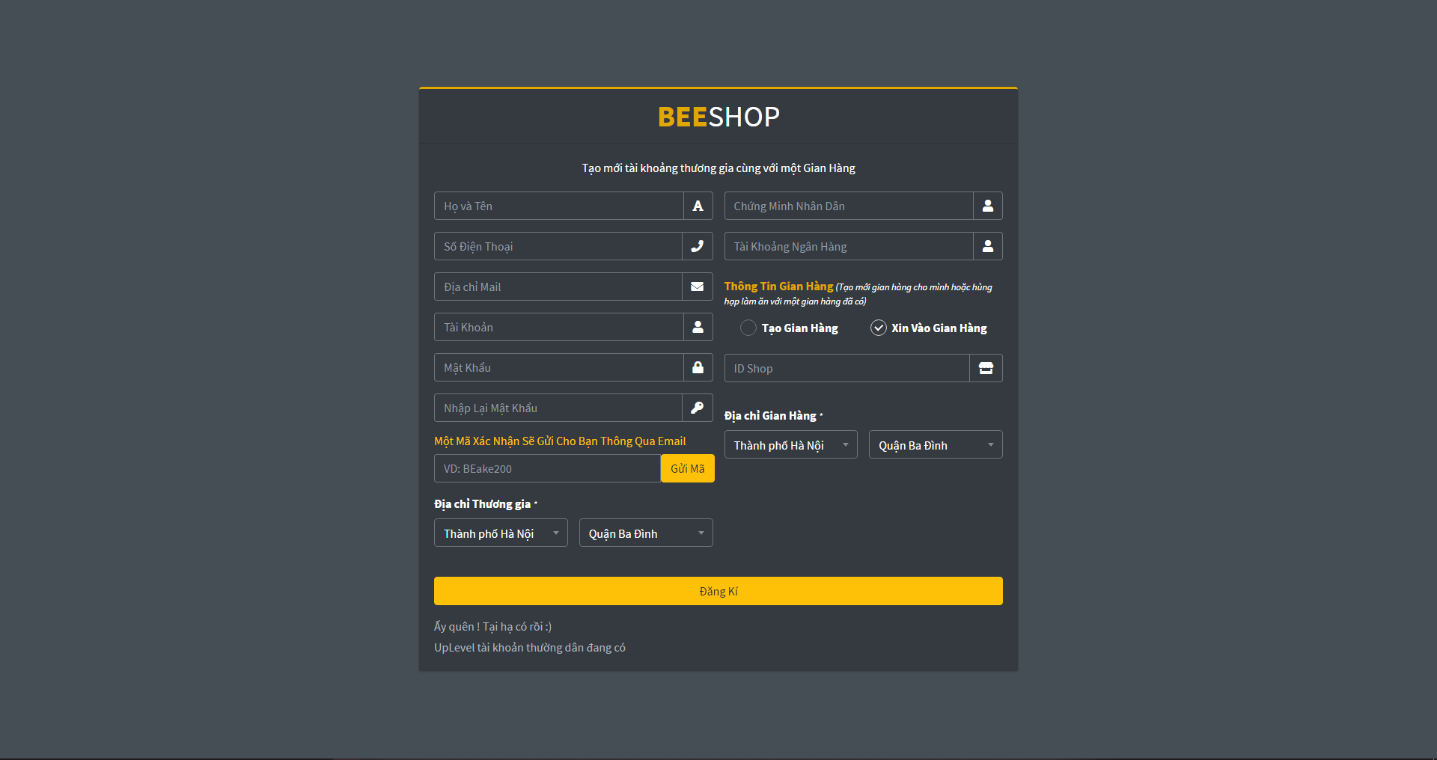


Hình 4. Giao diện trang mua hàng

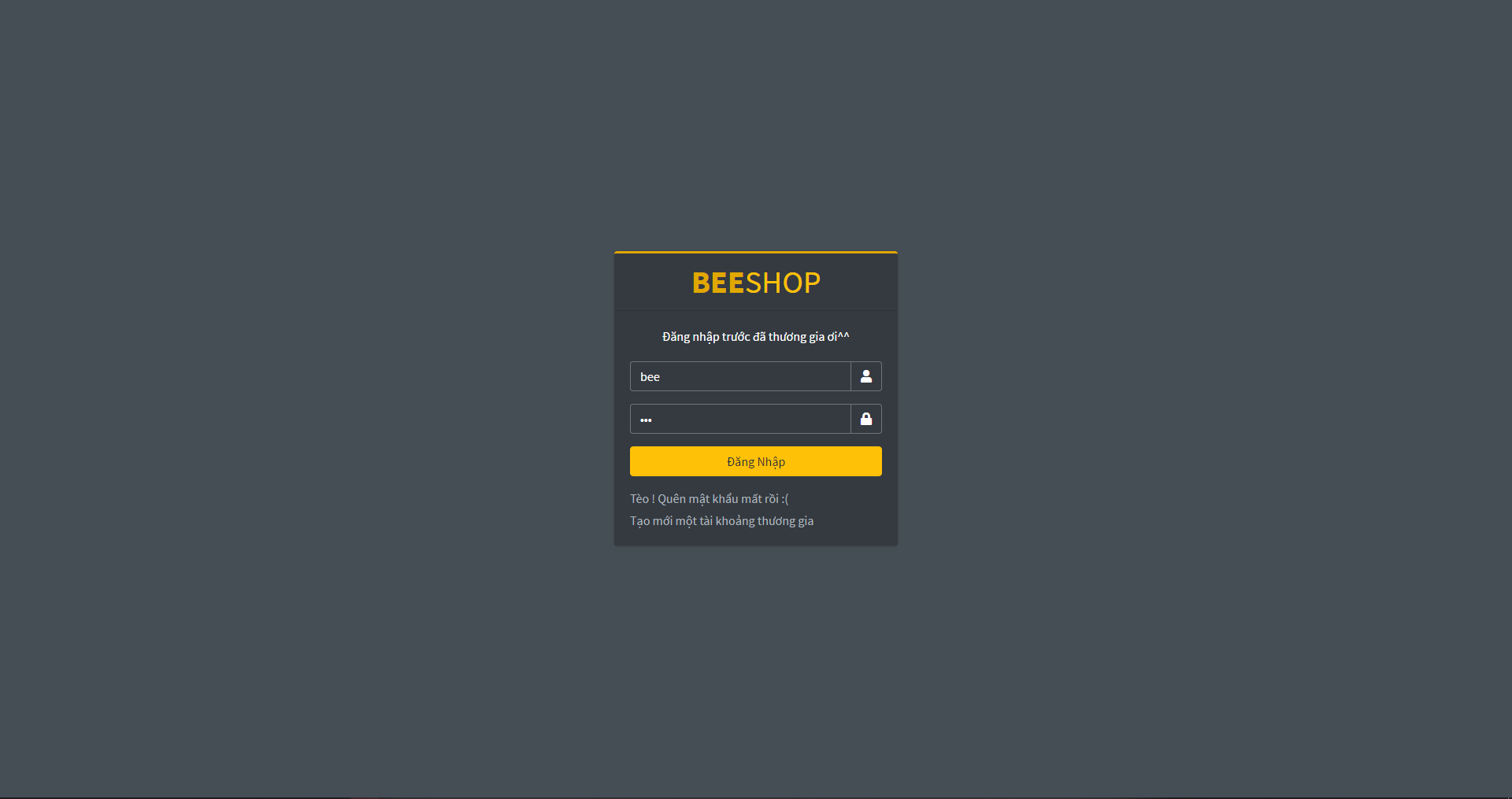


Hình 5. Đơn hàng

1. Giao diện trang bán hàng



Hình 6. Giao diện trang đăng ký



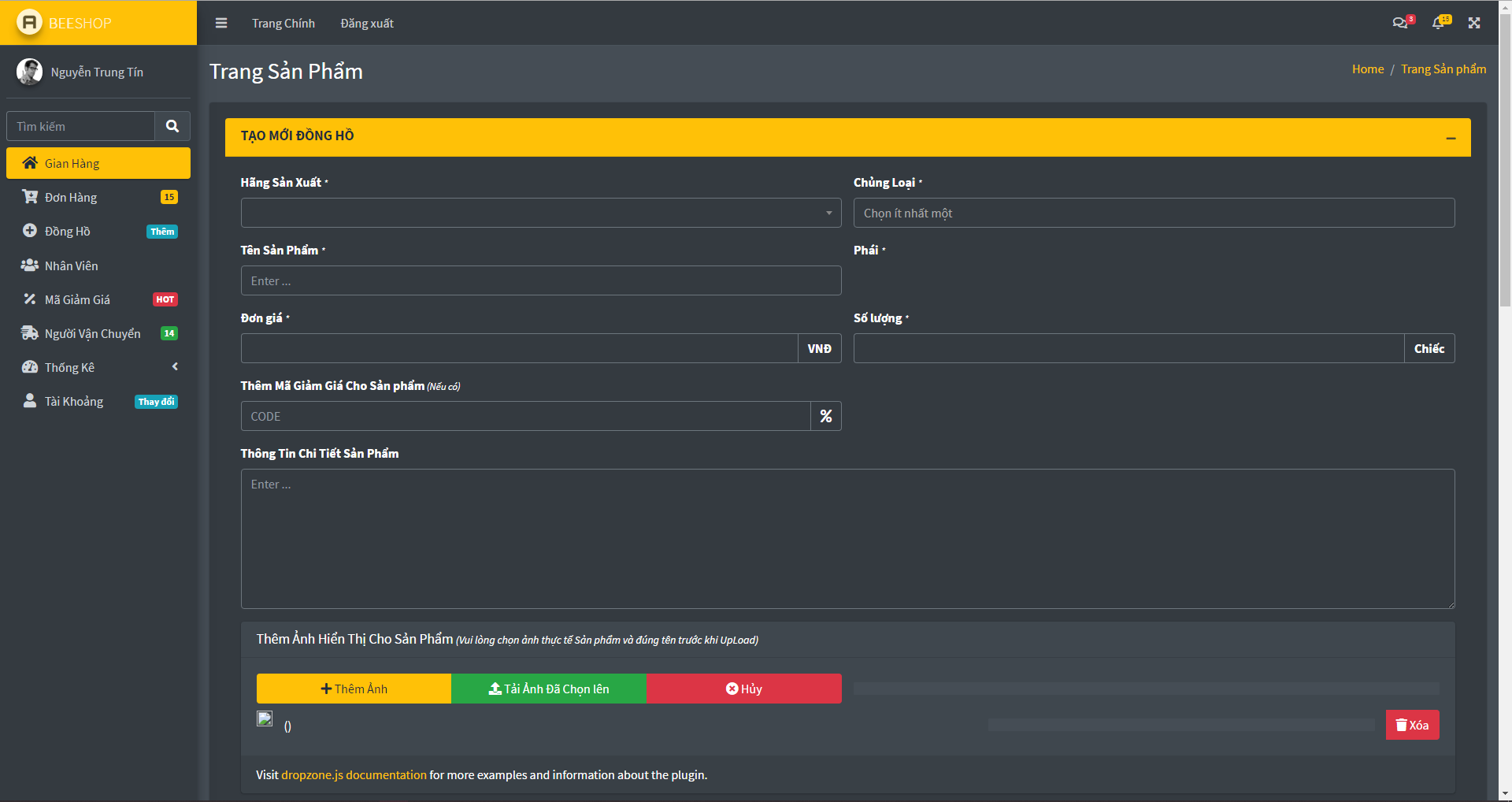
Hình 7. Giao diện trang đăng nhập



Hình 8. Giao diện thống kê

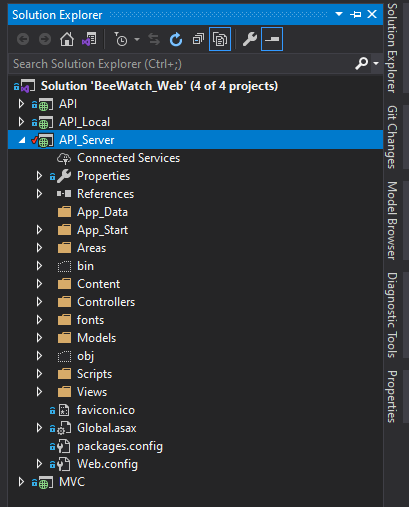


Hình 9. Giao diện thống kê



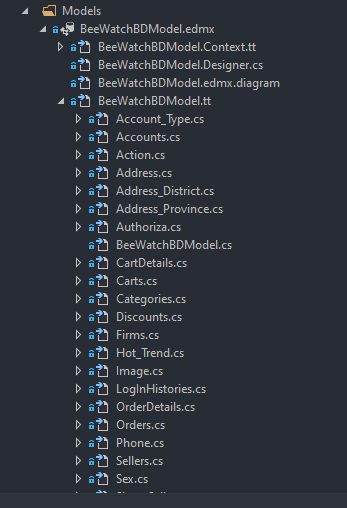
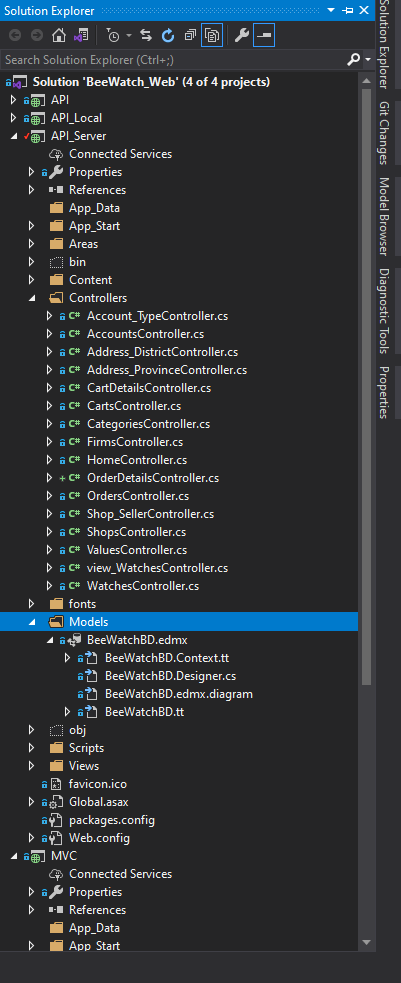
Hình 10. Trang Sản phẩm

1. Tổ chức các lớp trong chương trình

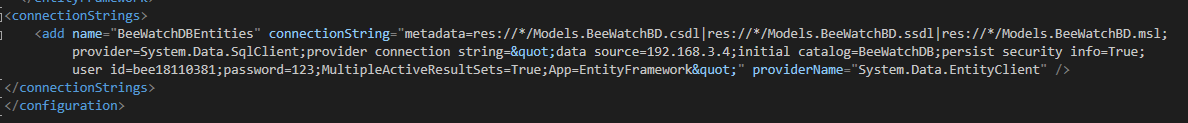
* Xây dựng chương trình theo ASP.NET MVC và có sử dụng API.

Hình 11. Tổ chức các project trong solution

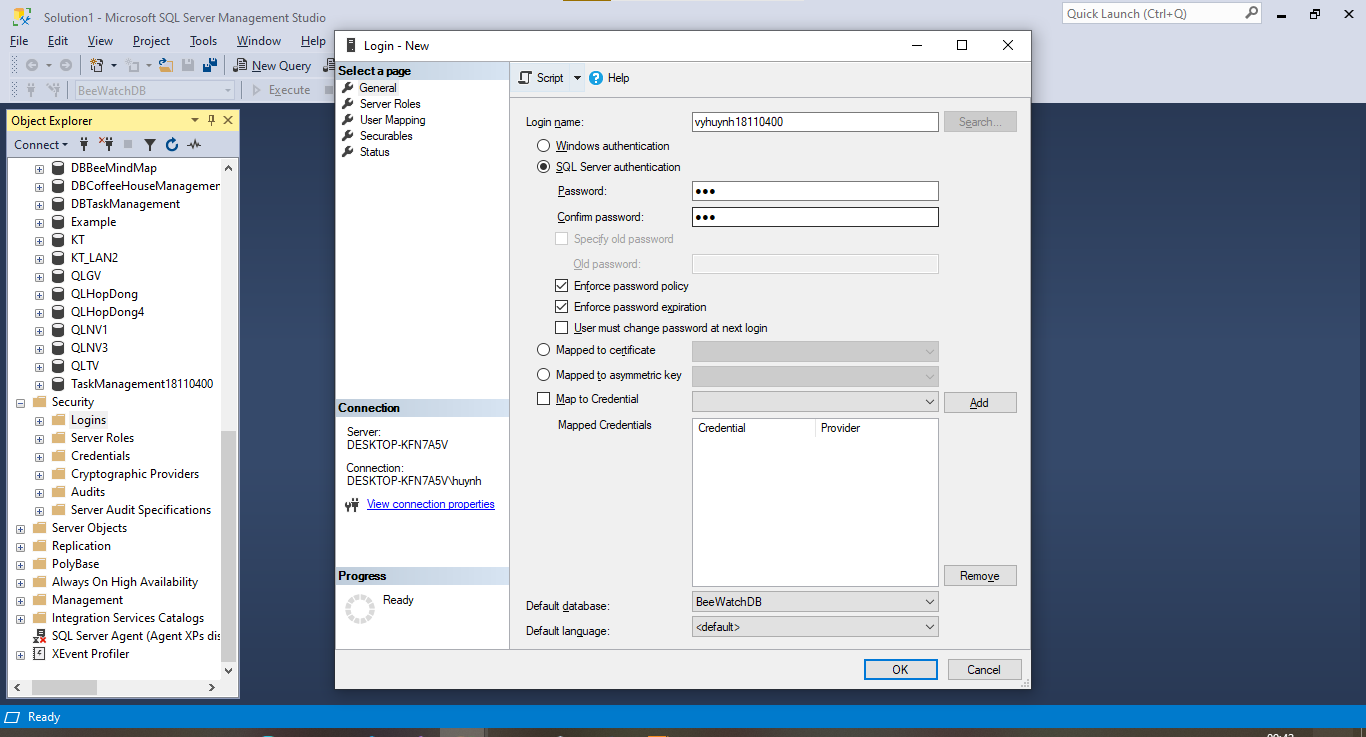
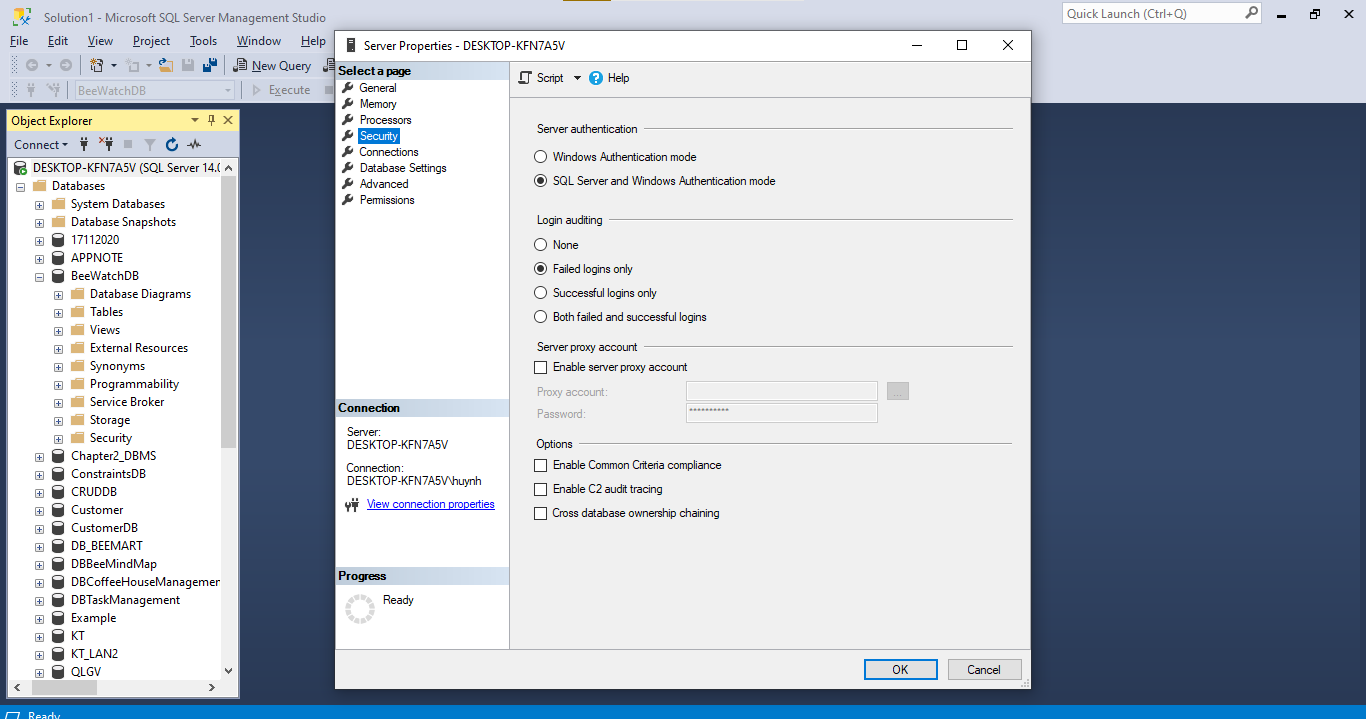
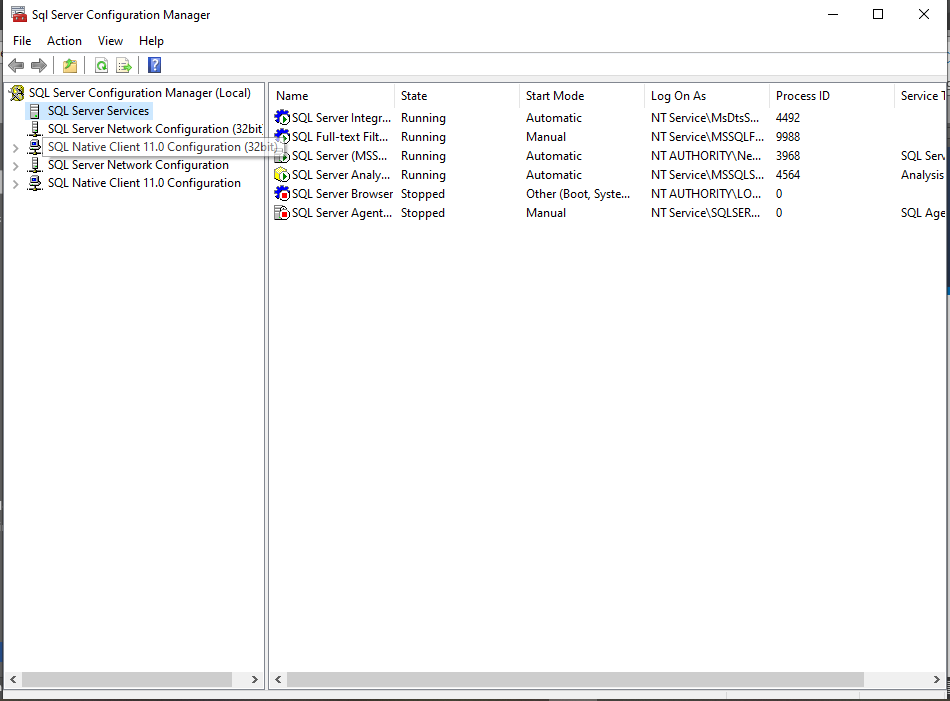
* Trang web được xây dựng theo mô hình Database First, các class được ánh xạ từ database được lưu giữ trong Model của API:

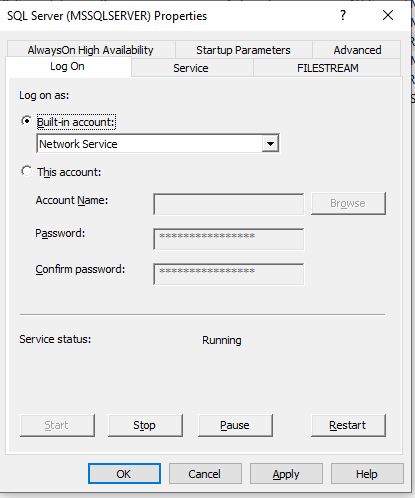


Hình 12. Models

* Views: tầng giao diện hiển thị cho người dùng nhìn thấy, hiển thị dữ liệu được truy xuất từ tầng Models
* Controllers: tầng giúp kết nối giữa tầng Models và tầng Views trong mô hình MVC.
* Kết nối CSDL thông qua Connection strings trong Web.config:

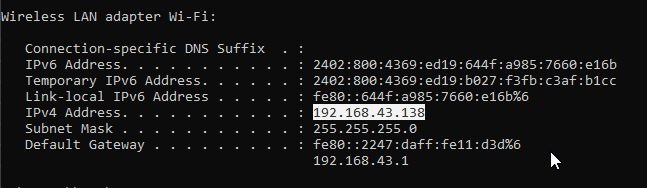
Hình 13. Chuỗi ConnectionStrings

* 1. Kết nối CSDL từ xa
* Tạo login từ SQL Server
* Cho phép đăng nhập bằng tài khoản
* Mở SQL Server Configuration Manager

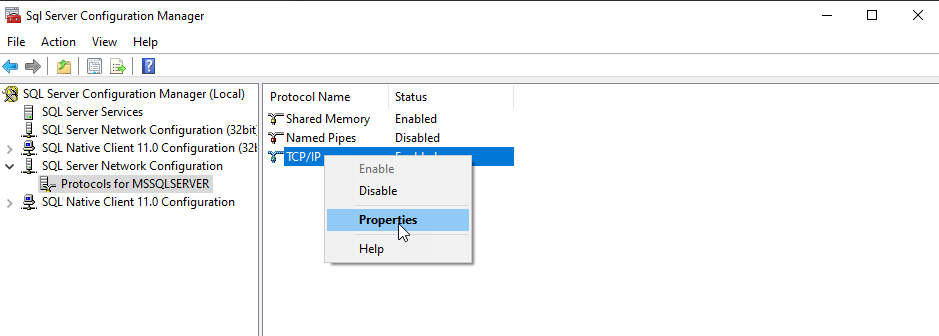
Chọn Properties của MSSQLSERVER, chọn Built-in account – Network Service

* Cấu hình TCP/IP – IP Address:

Mở CMD, gõ ipconfig để lấy địa chỉ Ipv4

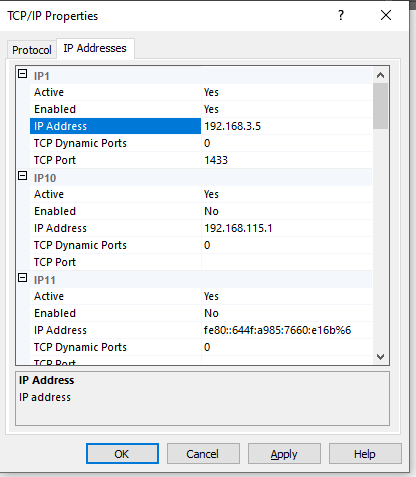
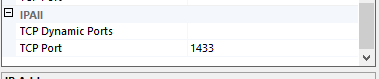


Chọn Properties của TCP/IP

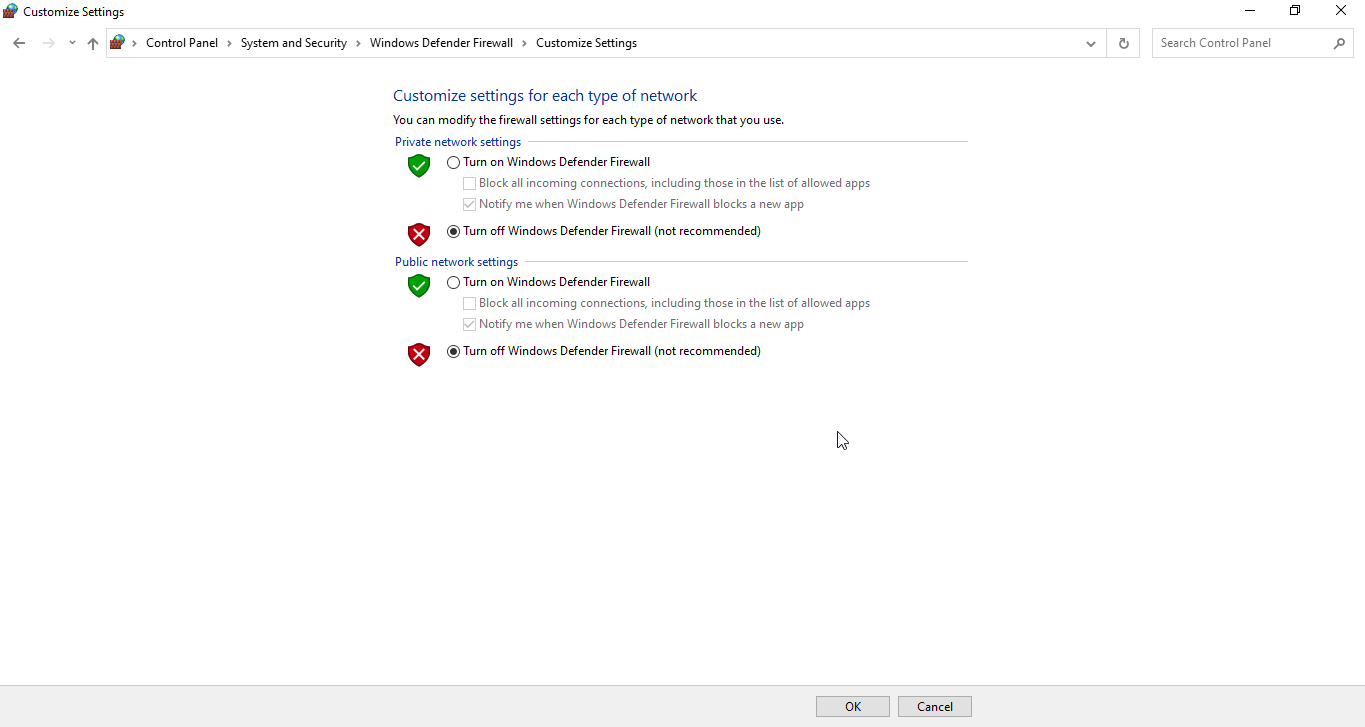


Protocol -> Enable -> Yes

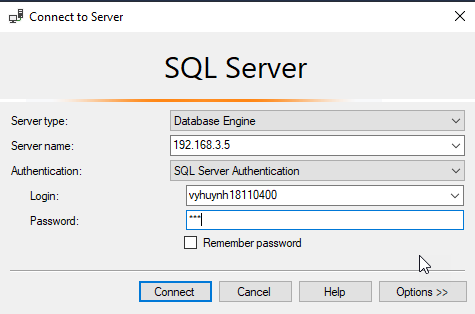
Cấu hình IP Address



* Tắt Firewall



* Đăng nhập SQL Server bằng account được tạo



CHƯƠNG 6. TỔNG KẾT

1. Đóng góp đề tài

Xây dựng trang web bán đồng hồ đẹp mắt, trực quan, thân thiện với người dùng. Trang web được áp dụng các kiến thức của Hệ quản trị CSDL, cụ thể:

* Phân tích, nắm bắt được nhu cầu sử dụng trang web để thiết kế CSDL phù hợp.
* Xây dựng CSDL chặt chẽ từ lược đồ ERD.
* Phân cấp, phân quyền cho các đối tượng sử dụng CSDL.
* Tìm hiểu và nắm rõ về các khái niệm trong SQL Sever.
* Có khả năng mở rộng CSDL khi có nhu cầu.
* Kết nối và chạy được trên nền tảng web.
* Tạo thống kê với Chart để dễ dàng quản lý.

1. Các tính năng bổ sung, đặc biệt trong đề tài
2. Gửi mail

Gửi mail xác nhận cho người dùng khi quên mật khẩu và muốn reset lại pass

1. Sử dụng Cursor

Sử dụng Cursor thay đổi Status của bảng Discounts, mã giảm giá có thời gian quy định sử dụng trong khoảng thời gian nhất định, dùng cursor để duyệt từng mã giảm giá có trong CSDL để kiểm tra HSD.

|  |
| --- |
| CREATE PROC sp\_ChangeStatusDiscount @Ngay DATE  AS  SET NOCOUNT ON  BEGIN  DECLARE @id int, @code nchar(10), @datefrom date, @dateto date  SET @Ngay = GETDATE()  ---lay danh sach discount dua vao con tro cursor\_date  DECLARE cursor\_date CURSOR FOR  SELECT Discounts.id, Code, DateFrom, DateTo  FROM Discounts  OPEN cursor\_date  FETCH NEXT FROM cursor\_date INTO @id, @code, @datefrom, @dateto  WHILE @@FETCH\_STATUS = 0  BEGIN  IF(@dateto < @Ngay)  BEGIN  UPDATE Discounts SET Status = 0  WHERE @id = id  END  FETCH NEXT FROM cursor\_date INTO @id, @code, @datefrom, @dateto  END  CLOSE cursor\_date  DEALLOCATE cursor\_date  END  GO |

1. Phân trang cho bảng

Mỗi trang sẽ giới hạn số sản phẩm được hiển thị lên, ở đây, dùng function phân trang để lấy dữ liệu theo yêu cầu truy xuất đổ lên web.

|  |
| --- |
| CREATE FUNCTION fn\_PhanTrang(@num\_offset int, @num\_next int)  RETURNS TABLE  AS  RETURN  SELECT \* FROM Watches  ORDER BY id OFFSET (@num\_offset \* @num\_next - @num\_next) ROWS FETCH NEXT @num\_next ROWS ONLY;  GO |

1. Thêm login kết nối server

|  |
| --- |
| CREATE PROC [dbo].[SQLAccount\_Create](@username nvarchar(20),@password nvarchar(max),@namerole nvarchar(50))  AS  BEGIN  DECLARE @createUser nvarchar(max)  DECLARE @createLogin nvarchar(max)  DECLARE @addSellerRole nvarchar(max)  DECLARE @addAdminRole nvarchar(max)  DECLARE @error int  SET @error=0  SET @createLogin='create login '+ @username + ' with password = ' + Char(39) + @password + Char(39)  SET @createUser='create user '+ @username +' for login '+ @username  SET @addSellerRole='Sp\_addRoleMember' + Char(39) +'seller\_role' + Char(39) + ',' + Char(39) + @username + Char(39)  SET @addAdminRole='Sp\_addRoleMember' + Char(39) +'db\_owner' + Char(39) + ',' + Char(39) + @username + Char(39)  BEGIN TRY  BEGIN TRANSACTION    EXEC (@createLogin)  EXEC (@createUser)  IF(@namerole = N'Admin')  EXEC (@addAdminRole)  IF(@namerole = N'Bán hàng')  EXEC (@addSellerRole)  COMMIT  END TRY  BEGIN CATCH  -- Test XACT\_STATE for 0, 1, or -1.  -- If 1, the transaction is committable.  -- If -1, the transaction is uncommittable and should be rolled back.  -- XACT\_STATE = 0 means there is no transaction and a commit or rollback operation would generate an error.    -- Test whether the transaction is uncommittable.  IF (XACT\_STATE()) = -1  BEGIN  ROLLBACK TRANSACTION;  END;  END CATCH  END |

1. Xóa login kết nối server

|  |
| --- |
| CREATE PROC [dbo].[SQLAccount\_KillConnection] @username nvarchar(20)  AS  BEGIN  DECLARE @killConnection NVARCHAR(MAX) = N'';  SELECT @killConnection += N'KILL ' + CONVERT(VARCHAR(11), session\_id) + N';'  FROM sys.dm\_exec\_sessions  WHERE login\_name = @username  EXEC sys.sp\_executesql @killConnection  END  GO |

1. Các hạn chế

* Các chức năng trong trang web chưa hoàn thiện nên chưa thể sử dụng được tất cả các câu truy vấn.
* Một số truy vấn vẫn còn chưa được tối ưu.
* Chưa đảm bảo tất cả các ngoại lệ có thể xảy ra trong thực tế.

1. Hướng phát triển

Nhóm đã cố gắng phát triển và xây dựng nền tảng đề tài theo yêu cầu. Do thời gian có hạn chỉ mới đi vào các chức năng cơ bản. Với đề tài này, nhóm có hướng phát triển xây dựng được toàn bộ các chức năng cần có, chỉnh sửa, mở rộng và tối ưu chức năng để hoàn thiện chương trình.

TÀI LIỆU THAM KHẢO

[1] Stackoverflow , Stack Exchange Inc, <https://stackoverflow.com/>

[2] Microsoft Studio Document, Language C#, <https://docs.microsoft.com/en-us/visualstudio/?view=vs-2017> .

[3] AdminLTE Bootstrap Admin Dashboard Template, [Free Bootstrap Admin Template | AdminLTE.IO](https://adminlte.io/) .

[4] SQL Server Tutorial, [SQL Server Tutorial](https://www.sqlservertutorial.net/).

PHỤ LỤC

Hướng dẫn sử dụng

(File đính kèm)