**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**



**ĐỒ ÁN MÔN HỌC**

**LẬP TRÌNH TRỰC QUAN**

**XÂY DỰNG PHẦN MỀM QUẢN LÝ BÁN HÀNG**

**Giảng viên hướng dẫn : ThS. Nguyễn Tấn Toàn**

**Sinh viên thực hiện 1 : Tô Thế Kiệt**

**Mã sinh viên 1 : 21522263**

**Sinh viên thực hiện 2 : Vũ Hoàng Nhật Minh**

**Mã sinh viên 2 :**

**Sinh viên thực hiện 3 : Ngô Hữu Thiên**

**Mã sinh viên 3 :**

**Sinh viên thực hiện 4 : Đào Anh Tú**

**Mã sinh viên 4 :**

**Lớp : IT008.N13.PMCL**

**Bộ môn : Lập trình trực quan**

Tp HCM, tháng … năm 2023

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  **KHOA CÔNG NGHỆ PHẦN MỀM**  🙡★🙣 | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  **Độc lập – Tự do – Hạnh phúc**  🙡★🙣 |

**NHIỆM VỤ ĐỒ ÁN MÔN HỌC**

Lớp: **IT008.N13.PMCL**

Tên đề tài: **XÂY DỰNG PHẦN MỀM QUẢN LÝ CỬA HÀNG QUẦN ÁO**

Giảng viên giảng dạy: **ThS. Nguyễn Tấn Toàn**

Thời gian thực hiện: **10/2022 đến 01/2023**

Nhiệm vụ đồ án môn học:

1. Xây dựng CSDL trong SQL Server.
2. Thiết kế giao diện phần mềm.
3. Lập trình xử lý phần mềm với các chức năng sau:
   * Đăng ký và đăng nhập
   * Quản lý danh mục
   * Quản lý kết quả tổng hợp
   * Thống kê và tìm kiếm
   * Quản lý người dùng.
4. Nộp file nén (\*.rar) lưu sản phẩm đề tài bao gồm:

* File báo cáo word (\*.docx)
* Thư mục chứa dự án (project), các class thư viện, CSDL, hình ảnh, ...)

*Tp.HCM, ngày tháng năm 2023*

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **GIẢNG VIÊN GIẢNG DẠY**  *(Ký và ghi rõ họ tên)*  **ThS. Nguyễn Tấn Toàn** | |
| **BẢNG PHÂN CÔNG THỰC HIỆN ĐỒ ÁN MÔN HỌC** | | | |
| 1. Lên ý tưởng (Thế Kiệt) | | | 1. Thiết kế ERD, mô hình dữ liệu quan hệ (Thế Kiệt) |
| 1. Phân công công việc (Thế Kiệt) | | | 1. Thiết kế cơ sở dữ liệu (Thế Kiệt) |
| 1. Thiết kế giao diện các màn hình sau:  * Màn hình đăng nhập (Thế Kiệt) * Màn hình đăng ký (Thế Kiệt) * Màn hình chính (Anh Tú) * Màn hình quản lý đơn hàng (Anh Tú) * Màn hình quản lý sản phẩm (Anh Tú) * Màn hình quản lý khách hàng (Hữu Thiên) * Màn hình quản lý nhập hàng (Hữu Thiên) * Màn hình quản lý nhân viên (Hữu Thiên) * Màn hình quên mật khẩu (Thế Kiệt) | | | 1. Thiết kế giao diện các màn hình sau:  * Màn hình trang chủ (Anh Tú) * Màn hình thống kê (Hữu Thiên) * Màn hình cài đặt thông tin tài khoản (Nhật Minh) * Màn hình đổi mật khẩu (Nhật Minh) * Màn hình gửi thông báo tới khách hàng (Thế Kiệt) * Màn hình gửi báo lỗi tới bộ phận phát triển phần mềm (Thế Kiệt) |
| 1. Code xử lý chức năng các màn hình sau:  * Màn hình đăng ký (Nhật Minh) * Màn hình chính (Anh Tú) * Màn hình quản lý đơn hàng (Nhật Minh) * Màn hình quản lý sản phẩm (Thế Kiệt) * Màn hình quản lý khách hàng (Hữu Thiên) * Màn hình quản lý nhân viên (Thế Kiệt) * Màn hình quên mật khẩu (Nhật Minh) | | | 1. Code xử lý chức năng các màn hình sau:  * Màn hình đăng nhập (Nhật Minh) * Màn hình trang chủ (Thế Kiệt) * Màn hình quản lý nhập hàng (Hữu Thiên) * Màn hình thống kê (Hữu Thiên) * Màn hình cài đặt thông tin tài khoản (Thế Kiệt) * Màn hình đổi mật khẩu (Thế Kiệt) * Màn hình gửi thông báo tới khách hàng (Nhật Minh) * Màn hình gửi báo lỗi tới bộ phận phát triển phần mềm (Nhật Minh) |
| 1. Test chức năng | | | 1. Test chức năng |
| 1. Soạn, chỉnh sửa file báo cáo | | | 1. Soạn và chỉnh sửa file báo cáo |

**LỜI CẢM ƠN**

Sự phát triển như vũ bão của công nghệ thông tin (CNTT) đã tác động mạnh mẽ và to lớn đến mọi mặt đời sống kinh tế xã hội. Ngày nay, CNTT đã trở thành một trong những động lực quan trọng nhất của sự phát triển. Với khả năng số hóa mọi loại thông tin, máy tính trở thành phương tiện xử lý thông tin thống nhất và đa năng, thực hiện được nhiều chức năng khác nhau trên mọi dạng thông tin thuộc mọi lĩnh vực như: nghiên cứu, quản lý, kinh doanh, giáo dục, ...

Những ứng dụng của CNTT trong lĩnh vực quản lý là những ứng dụng vô cùng quan trọng. Nó không những giải phóng công sức cho những người quản lý mà còn đem lại sự chính xác và nhanh nhạy trong quản lý.

Là sinh viên CNTT, trong đồ án môn Lập trình trực quan này, nhóm chúng em đã chọn và thực hiện đồ án “XÂY DỰNG ỨNG DỤNG QUẢN LÝ BÁN HÀNG”.

Nhóm xin gửi lời cảm ơn chân thành đến thầy Nguyễn Tấn Toàn đã tận tình giảng dạy, hướng dẫn chúng em trong suốt thời gian học vừa qua và các bạn học đã góp ý và giúp đỡ nhóm trong quá trình thực hiện đồ án này.

Do kiến thức và thời gian thực hiện hạn chế, đồ án của nhóm vẫn còn nhiều thiếu sót. Nhóm rất mong nhận được góp ý của thầy và các bạn để đồ án của nhóm được hoàn thiện.

**Nhóm sinh viên thực hiện**

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

Tp.HCM, ngày … tháng … năm 2023

**GVHD**

**ThS.** **Nguyễn Tấn Toàn**

**MỤC LỤC**

[Chương 1: GIỚI THIỆU ĐỀ TÀI 11](#_Toc91884983)

[1.1. Tên đề tài: 11](#_Toc91884984)

[1.2. Mô tả đề tài: 11](#_Toc91884985)

[1.3. Lý do chọn đề tài: 11](#_Toc91884986)

[1.4. Các chức năng chính của đề tài: 11](#_Toc91884987)

[1.5. Công nghệ sử dụng: 11](#_Toc91884988)

[1.6. Môi trường lập trình: 11](#_Toc91884989)

[1.7. Công cụ hỗ trợ (nếu có): 11](#_Toc91884990)

[Chương 2: GIỚI THIỆU CÔNG NGHỆ 13](#_Toc91884991)

[2.1. Giới thiệu C#: 13](#_Toc91884992)

[2.1.1. Định nghĩa: 13](#_Toc91884993)

[2.1.2. Đặc điểm: 13](#_Toc91884994)

[2.2. Tổng quan Entity Framework: 13](#_Toc91884995)

[2.2.2. Lý do sử dụng Entity Framework: 13](#_Toc91884996)

[2.4. Công nghệ WPF: 14](#_Toc91884997)

[2.4.1. Giới thiệu WPF: 14](#_Toc91884998)

[2.4.2. Ưu điểm của WPF: 14](#_Toc91884999)

[Chương 3: THIẾT KẾ CƠ SỞ DỮ LIỆU 15](#_Toc91885000)

[3.1. Khảo sát hiện trạng: 15](#_Toc91885001)

[3.2. Mô tả tân từ: 15](#_Toc91885002)

[3.3. Mô hình thực thể kết hợp: 16](#_Toc91885003)

[3.4. Mô hình CSDL quan hệ: 17](#_Toc91885004)

[3.5. Database diagram trong SQL: 18](#_Toc91885005)

[3.6. Cấu trúc các bảng dữ liệu trong SQL: 19](#_Toc91885006)

[3.7. Dữ liệu mẫu: 21](#_Toc91885007)

[Chương 4: XÂY DỰNG ỨNG DỤNG 23](#_Toc91885008)

[4.1. Màn hình chính: 23](#_Toc91885009)

[4.1.1. Giao diện: 23](#_Toc91885010)

[4.1.2. Mô tả chức năng: 24](#_Toc91885011)

[4.1.3. Kỹ thuật xử lý: 24](#_Toc91885012)

[4.2. Màn hình đăng nhập: 26](#_Toc91885013)

[4.2.1. Giao diện: 26](#_Toc91885014)

[4.2.2. Mô tả chức năng: 27](#_Toc91885015)

[4.2.3. Kỹ thuật xử lý: 27](#_Toc91885016)

[4.3. Màn hình đăng ký: 29](#_Toc91885017)

[4.3.1. Giao diện: 29](#_Toc91885018)

[4.3.2. Mô tả chức năng: 29](#_Toc91885019)

[4.3.3. Kỹ thuật xử lý: 29](#_Toc91885020)

[4.4. Quản lý danh mục: 32](#_Toc91885021)

[4.4.1. Màn hình danh mục trang chủ: 32](#_Toc91885022)

[4.4.1.1. Giao diện: 32](#_Toc91885023)

[4.4.1.2. Mô tả chức năng: 33](#_Toc91885024)

[4.4.1.3. Kỹ thuật xử lý: 33](#_Toc91885025)

[4.4.2. Màn hình danh mục đơn hàng: 34](#_Toc91885026)

[4.4.2.1. Giao diện: 34](#_Toc91885027)

[4.4.2.2. Mô tả chức năng: 35](#_Toc91885028)

[4.4.2.3. Kỹ thuật xử lý: 35](#_Toc91885029)

[4.4.3. Màn hình danh mục sản phẩm: 37](#_Toc91885030)

[4.4.3.1. Giao diện: 37](#_Toc91885031)

[4.4.3.2. Mô tả chức năng: 38](#_Toc91885032)

[4.4.3.3. Kỹ thuật xử lý: 38](#_Toc91885033)

[4.4.4. Màn hình danh mục khách hàng: 42](#_Toc91885034)

[4.4.4.1. Giao diện: 42](#_Toc91885035)

[4.4.4.2. Mô tả chức năng: 42](#_Toc91885036)

[4.4.4.3. Kỹ thuật xử lý: 42](#_Toc91885037)

[4.4.5. Màn hình danh mục nhập hàng: 45](#_Toc91885038)

[4.4.5.1. Giao diện: 45](#_Toc91885039)

[4.4.5.2. Mô tả chức năng: 45](#_Toc91885040)

[4.4.5.3. Kỹ thuật xử lý: 45](#_Toc91885041)

[4.4.6. Màn hình danh mục thống kê: 47](#_Toc91885042)

[4.4.6.1. Giao diện: 47](#_Toc91885043)

[4.4.6.2. Mô tả chức năng: 48](#_Toc91885044)

[4.4.6.3. Kỹ thuật xử lý: 48](#_Toc91885045)

[4.4.7. Màn hình danh mục quản lý: 56](#_Toc91885046)

[4.4.7.1. Giao diện: 56](#_Toc91885047)

[4.4.7.2. Mô tả chức năng: 56](#_Toc91885048)

[4.4.7.1. Kỹ thuật xử lý: 57](#_Toc91885049)

[4.4.8. Màn hình danh mục cài đặt: 59](#_Toc91885050)

[4.4.8.1. Giao diện: 59](#_Toc91885051)

[4.4.8.2. Mô tả chức năng: 59](#_Toc91885052)

[4.4.8.3. Kỹ thuật xử lý: 60](#_Toc91885053)

[4.6. Quản lý đơn hàng: 62](#_Toc91885054)

[4.6.1. Màn hình chi tiết đơn hàng: 62](#_Toc91885055)

[4.6.1.1. Giao diện: 62](#_Toc91885056)

[4.6.1.2. Mô tả chức năng: 62](#_Toc91885057)

[4.6.1.3. Kỹ thuật xử lý: 62](#_Toc91885058)

[4.6.2. Màn hình tạo đơn hàng: 64](#_Toc91885059)

[4.6.2.1. Giao diện: 64](#_Toc91885060)

[4.6.2.2. Mô tả chức năng: 64](#_Toc91885061)

[4.6.2.3. Kỹ thuật xử lý: 65](#_Toc91885062)

[4.7. Quản lý sản phẩm: 70](#_Toc91885063)

[4.7.1. Màn hình chi tiết sản phẩm: 70](#_Toc91885064)

[4.7.1.1. Giao diện: 70](#_Toc91885065)

[4.7.1.2. Mô tả chức năng: 70](#_Toc91885066)

[4.7.1.3. Kỹ thuật xử lý: 70](#_Toc91885067)

[4.7.2. Màn hình thêm sản phẩm mới: 72](#_Toc91885068)

[4.7.2.1. Giao diện: 72](#_Toc91885069)

[4.7.2.2. Mô tả chức năng: 73](#_Toc91885070)

[4.7.2.3. Kỹ thuật xử lý: 73](#_Toc91885071)

[4.8. Quản lý khách hàng: 75](#_Toc91885072)

[4.8.1. Màn hình chi tiết khách hàng: 75](#_Toc91885073)

[4.8.1.1. Giao diện: 75](#_Toc91885074)

[4.8.1.2. Mô tả chức năng: 75](#_Toc91885075)

[4.8.1.3. Kỹ thuật xử lý: 75](#_Toc91885076)

[4.8.2. Màn hình thêm khách hàng: 76](#_Toc91885077)

[4.8.2.1. Giao diện: 76](#_Toc91885078)

[4.8.2.2. Mô tả chức năng: 76](#_Toc91885079)

[4.8.2.3. Kỹ thuật xử lý: 77](#_Toc91885080)

[4.9. Quản lý nhân viên: 78](#_Toc91885081)

[4.9.1. Màn hình chi tiết nhân viên: 78](#_Toc91885082)

[4.9.1.1. Giao diện: 78](#_Toc91885083)

[4.9.1.2. Mô tả chức năng: 78](#_Toc91885084)

[4.9.1.3. Kỹ thuật xử lý: 78](#_Toc91885085)

[4.9.2. Màn hình thêm nhân viên: 80](#_Toc91885086)

[4.9.2.1 Giao diện: 80](#_Toc91885087)

[4.9.2.2. Mô tả chức năng: 80](#_Toc91885088)

[4.9.2.3. Kỹ thuật xử lý: 80](#_Toc91885089)

[4.10. Quản lý nhập hàng: 83](#_Toc91885090)

[4.10.1. Màn hình chi tiết phiếu nhập: 83](#_Toc91885091)

[4.10.1.1. Giao diện: 83](#_Toc91885092)

[4.10.1.2. Mô tả chức năng: 83](#_Toc91885093)

[4.10.1.3. Kỹ thuật xử lý: 83](#_Toc91885094)

[4.10.2. Màn hình thêm phiếu nhập: 84](#_Toc91885095)

[4.10.2.1. Giao diện: 84](#_Toc91885096)

[4.10.2.2. Mô tả chức năng: 85](#_Toc91885097)

[4.10.2.3. Kỹ thuật xử lý: 85](#_Toc91885098)

[4.11. Màn hình đổi mật khẩu: 88](#_Toc91885099)

[4.11.1. Giao diện: 88](#_Toc91885100)

[4.11.2. Mô tả chức năng: 88](#_Toc91885101)

[4.11.3. Kỹ thuật xử lý: 89](#_Toc91885102)

[4.12. Màn hình quên mật khẩu: 90](#_Toc91885103)

[4.12.1. Giao diện: 90](#_Toc91885104)

[4.12.2. Mô tả chức năng: 90](#_Toc91885105)

[4.12.3. Kỹ thuật xử lý: 90](#_Toc91885106)

[KẾT LUẬN 92](#_Toc91885107)

[ Ưu điểm của đồ án: 92](#_Toc91885108)

[ Hạn chế của đồ án: 92](#_Toc91885109)

[ Hướng phát triển của đồ án: 92](#_Toc91885110)

[TÀI LIỆU THAM KHẢO 93](#_Toc91885111)

**MỤC LỤC HÌNH ẢNH**

[Hình 3.1. Mô hình thực thể kết hợp 16](#_Toc91800840)

[Hình 3.2. Mô hình CSDL quan hệ 17](#_Toc91800841)

[Hình 3.3. Database diagram trong SQL 18](#_Toc91800842)

[Hình 3.4. Dữ liệu mẫu 22](#_Toc91800843)

[Hình 4.1. Giao diện màn hình chính khi đăng nhập dưới quyền quản lý 23](#_Toc91800844)

[Hình 4.2. Giao diện màn hình chính khi đăng nhập dưới quyền nhân viên 23](#_Toc91800845)

[Hình 4.3. Giao diện màn hình đăng nhập 26](#_Toc91800846)

[Hình 4.4. Giao diện màn hình đăng ký 29](#_Toc91800847)

[Hình 4.5. Giao diện màn hình danh mục trang chủ 32](#_Toc91800848)

[Hình 4.6. Giao diện màn hình danh mục đơn hàng 34](#_Toc91800849)

[Hình 4.7. Giao diện màn hình danh mục sản phẩm 37](#_Toc91800850)

[Hình 4.8. Giao diện màn hình danh mục khách hàng 42](#_Toc91800851)

[Hình 4.9. Giao diện màn hình danh mục nhập hàng 45](#_Toc91800852)

[Hình 4.10. Giao diện màn hình danh mục thống kê trang 1 47](#_Toc91800853)

[Hình 4.11. Giao diện màn hình danh mục thống kê trang 2, chu kì theo năm 47](#_Toc91800854)

[Hình 4.12. Giao diện màn hình danh mục thống kê trang 2 chu kì theo tháng 48](#_Toc91800855)

[Hình 4.13. Giao diện màn hình danh mục quản lý 56](#_Toc91800856)

[Hình 4.14. Giao diện màn hình danh mục cài đặt 59](#_Toc91800857)

[Hình 4.15. Giao diện màn hình chi tiết hóa đơn 62](#_Toc91800858)

[Hình 4.16. Giao diện màn hình tạo hóa đơn 64](#_Toc91800859)

[Hình 4.17. Giao diện màn hình chi tiết sản phẩm 70](#_Toc91800860)

[Hình 4.18. Giao diện màn hình thêm sản phẩm mới 72](#_Toc91800861)

[Hình 4.19. Giao diện màn hình chi tiết khách hàng 75](#_Toc91800862)

[Hình 4.20. Giao diện màn hình thêm khách hàng 76](#_Toc91800863)

[Hình 4.21. Giao diện màn hình chi tiết nhân viên 78](#_Toc91800864)

[Hình 4.22. Giao diện màn hình thêm nhân viên 80](#_Toc91800865)

[Hình 4.23. Giao diện màn hình chi tiết phiếu nhập 83](#_Toc91800866)

[Hình 4.24. Giao diện màn hình thêm phiếu nhập 84](#_Toc91800867)

[Hình 4.25. Giao diện màn hình đổi mật khẩu 88](#_Toc91800868)

[Hình 4.26. Giao diện màn hình quên mật khẩu 90](#_Toc91800869)

Chương 1:   
GIỚI THIỆU ĐỀ TÀI

* 1. Tên đề tài:

XÂY DỰNG ỨNG DỤNG QUẢN LÝ BÁN HÀNG

* 1. Mô tả đề tài:
* Để việc quản lý kinh doanh trở nên thuận tiện và dễ dàng hơn, chúng em đã tiến hành phân tích và triển khai thực hiện một dự án đó là “Phần mềm Quản lý bán hàng”.
* Một ứng dụng quản lý với đầy đủ các chức năng cơ bản, giao diện bắt mắt, dễ thao tác, thân thiện với người dùng sẽ thích hợp với các doanh nghiệp vừa và nhỏ.
* Việc xây dựng ứng dụng quản lý sẽ giúp các doanh nghiệp quản lý hoạt động kinh doanh một cách hiệu quả, với quy trình rõ ràng, cải thiện hiệu quả và năng suất của nhân viên, tránh sai sót và thất thoát thông tin, và dễ dàng giám sát, theo dõi mục tiêu của doanh nghiệp.[8]
  1. Lý do chọn đề tài:
* Hiện nay, đời sống của người dân ngày càng phát triển, nhu cầu mua sắm và buôn bán tăng cao.
* Khi các doanh nghiệp nằm ở quy mô nhỏ hoặc mới thành lập, việc quản lý bằng phương pháp thủ công như excel có thể chưa gặp nhiều vấn đề, nhưng khi doanh nghiệp phát triển, việc sắp xếp, quản lý dữ liệu sẽ trở nên phức tạp và gây khó khăn cho các chủ doanh nghiệp.
  1. Các chức năng chính của đề tài:
* Quản lý sản phẩm
* Quản lý hóa đơn
* Quản lý khách hàng
* Quản lý nhân viên
* Quản lý nhập hàng
* Đăng nhập và đăng xuất
* Báo cáo và thống kê
* Gửi thông báo khuyến mãi tới toàn bộ khách hàng
* Gửi báo lỗi tới bộ phận phát triển phần mềm
* Cài đặt
  1. Công nghệ sử dụng:
* Sử dụng ngôn ngữ C# kết hợp CSDL SQL Server
* Sử dụng ngôn ngữ XAML
* Entity Framework
* WPF
  1. Môi trường lập trình:
* Microsoft Visual Studio 2022
  1. Công cụ hỗ trợ (nếu có):
* Figma: thiết kế giao diện cho phần mềm
* Github: quản lý source code
* Google Drive: quản lý file
* Facebook, Microsoft Teams: dùng để liên lạc
* Material Design in XAML Toolkit: thiết kế giao diện bên trong chương trình
* SfChart: vẽ biểu đồ

Chương 2:   
GIỚI THIỆU CÔNG NGHỆ

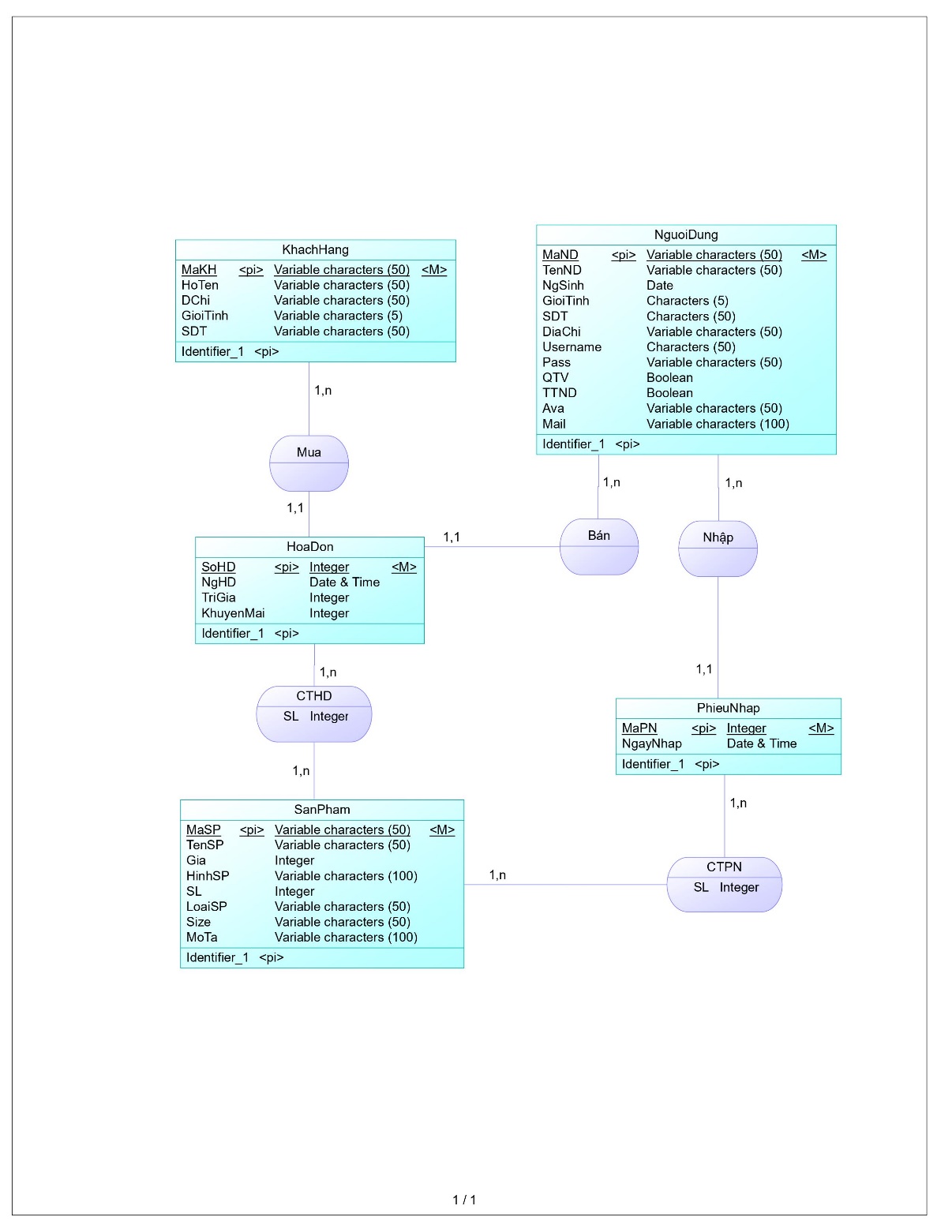
* 1. Giới thiệu C#:
     1. Định nghĩa:
  + C# (hay C sharp) là một ngôn ngữ lập trình đơn giản, được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000.
  + C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.[2]
    1. Đặc điểm:
* Hiện đại và dễ dàng.
* Mã nguồn mở và nhanh.
* Đa nền tảng.
* An toàn.
* Đa năng.
* Tiến hóa.[3]
  1. Tổng quan Entity Framework:
     1. Định nghĩa Entity Framework:
  + Entity Framework (EF) là một ánh xạ quan hệ đối tượng (ORM), là một loại công cụ giúp đơn giản hóa ánh xạ giữa các đối tượng trong phần mềm của bạn với các bảng và cột của cơ sở dữ liệu quan hệ.
  + Entity Framework (EF) là một framework ORM mã nguồn mở độc lập với .NET Framework.
  + Entity Framework đảm nhiệm việc tạo các kết nối cơ sở dữ liệu và thực thi các lệnh, cũng như lấy kết quả truy vấn và tự động ánh xạ các kết quả đó thành các đối tượng trong ứng dụng của bạn.
  + Entity Framework giúp theo dõi các thay đổi của đối tượng và cập nhật các thay đổi đó trở lại cơ sở dữ liệu cho bạn.[4]
    1. Lý do sử dụng Entity Framework:
* Nếu bạn thực hiện truy vấn dữ liệu. Bạn có thể sử dụng Linq to Entities để thao tác với objects được sinh ra từ Entity Framework, nghĩa là bạn không phải viết code SQL.
* Việc update các classes, commands dễ dàng mỗi khi cơ sở dữ liệu có sự thay đổi, điều này giúp bạn tiết kiệm thời gian đáng kể.
* Entity Framework sẽ tự động tạo ra các classes, commands tương ứng cho việc select, insert, update,delete dữ liệu từ cơ sở dữ liệu quan hệ.
* Entity Framework tự động tạo ra các classes cho việc truy xuất cơ sở dữ liệu giúp lập trình viên giảm được thời gian viết code thao tác với database. Hỗ trợ bạn không phải mất quá nhiều thời gian cho việc viết code để thao tác với database.[5]
  1. SQL Server:
     1. Định nghĩa
  + SQL server hay còn được gọi là Microsoft SQL Server, nó từ viết tắt của MS SQL Server. Đây chính là một loại phần mềm đã được phát triển bởi Microsoft và nó được sử dụng để có thể dễ dàng lưu trữ cho những dữ liệu dựa theo tiêu chuẩn RDBMS.
  + SQL Server có khả năng cung cấp đầy đủ các công cụ cho việc quản lý từ giao diện GUI đến sử dụng ngôn ngữ cho việc truy vấn SQL. Điểm mạnh của SQL điểm mạnh của nó là có nhiều nền tảng được kết hợp cùng như: ASP.NET, C# để xây dựng Winform cũng chính nó có khả năng hoạt động độc lập. Tuy nhiên, SQL Server thường đi kèm với việc thực hiện riêng các ngôn ngữ SQL, T-SQL,...

2.3.2. Tính năng của SQL Server

* Giúp người sử dụng có thể duy trì việc lưu trữ bền vững.
* Cho phép bạn tạo ra nhiều cơ sở dữ liệu hơn.
* Có khả năng phân tích dữ liệu bằng SSAS
* Nó có khả năng bảo mật cao
* Việc tạo ra được các báo cáo bằng SSRS — SQL Server Reporting Services sẽ được dễ dàng hơn.
* Các quá trình sẽ được thực hiện bằng SSIS — SQL Server Integration Services.
  1. Công nghệ WPF:
     1. Giới thiệu WPF:
* WPF viết tắt của Windows Presentation Foundation, là hệ thống API mới hỗ trợ việc xây dựng giao diện đồ hoạ trên nền Windows.
* Được xem như thế hệ kế tiếp của WinForms, WPF tăng cường khả năng lập trình giao diện của lập trình viên bằng cách cung cấp các API cho phép tận dụng những lợi thế về đa phương tiện hiện đại.
  + 1. Ưu điểm của WPF:
* Nó mới hơn và do đó phù hợp hơn với các tiêu chuẩn hiện tại.
* Microsoft đang sử dụng nó cho rất nhiều ứng dụng mới
* Nó linh hoạt hơn, có thể làm nhiều việc hơn mà không phải viết hoặc mua các control mới.
* Khi bạn cần sử dụng các control của bên thứ 3, các nhà phát triển các control này có thể sẽ tập trung hơn vào WPF vì nó mới hơn.
* XAML giúp dễ dàng tạo và chỉnh sửa GUI của bạn và cho phép công việc được phân chia giữa một nhà thiết kế (XAML) và một lập trình viên (C #, VB.NET, v.v.)
* Databinding, cho phép bạn có được một sự tách biệt hơn giữa data và layout.[7]

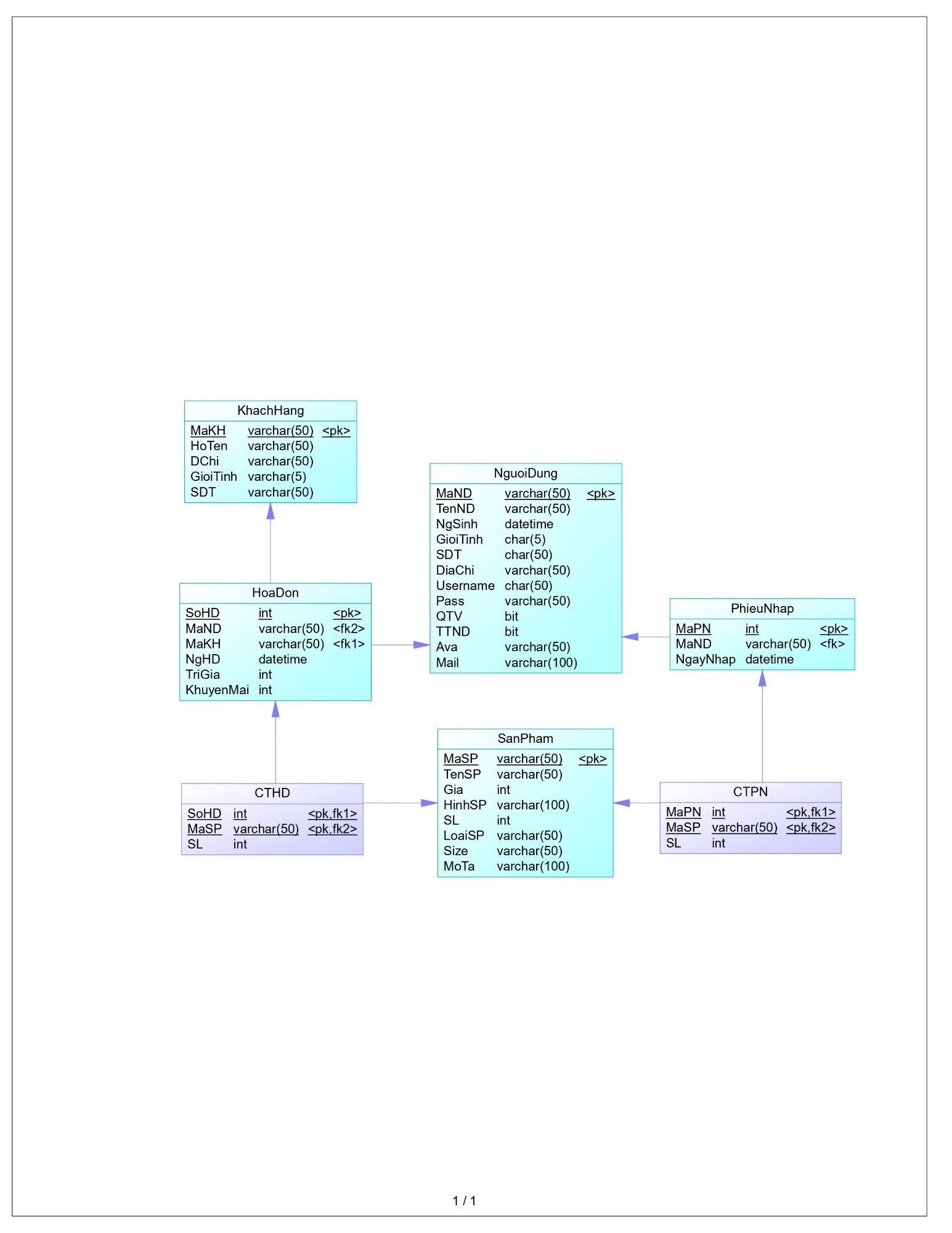
Chương 3:   
THIẾT KẾ CƠ SỞ DỮ LIỆU

* 1. Khảo sát hiện trạng:
* Một cửa hàng quần áo cần quản lý nhiều nhân viên, sản phẩm, khách hàng và việc mua hàng, nhập hàng.
* Mỗi nhân viên có một tài khoản đăng nhập riêng lưu trữ thông tin cá nhân (mã người dùng, họ tên, giới tính, địa chỉ, ngày sinh, số điện thoại, chức vụ, ảnh đại diện, địa chỉ mail) và thông tin tài khoản (tên đăng nhập, mật khẩu, tình trạng), mỗi chức vụ có quyền truy cập hệ thống riêng.
* Sản phẩm cần lưu trữ thông tin gồm: mã sản phẩm, tên sản phẩm, giá, mô tả, hình ảnh, số lượng, loại sản phẩm, kích cỡ.
* Khách hàng cần lưu trữ thông tin: mã khách hàng, họ tên, địa chỉ, số điện thoại, giới tính.
* Khi mua hàng, khách hàng sẽ nhận một hóa đơn tính tiền, trong đó sẽ có số hóa đơn, ngày mua, nhân viên bán, khách hàng mua, trị giá và khuyến mãi bao nhiêu. Mỗi hóa đơn gồm có sản phẩm và số lượng mua.
* Khi nhập hàng, nhân viên cần có phiếu nhập, thông tin phiếu nhập bao gồm mã phiếu nhập, ngày nhập, nhân viên nhập. Mỗi phiếu nhập gồm có sản phẩm và số lượng nhập.
  1. Mô tả tân từ:
* Mỗi khách hàng có một mã khách hàng duy nhất để phân biệt. Mỗi mã khách hàng xác định tất cả các thuộc tính của khách hàng đó, bao gồm: họ tên khách hàng, địa chỉ, số điện thoại, giới tính.
* Mỗi người dùng có một mã người dùng duy nhất để phân biệt. Mỗi mã người dùng xác định tất cả các thuộc tính của người dùng và thông tin tài khoản của người dùng đó, bao gồm: tên người dùng, ngày sinh, giới tính, số điện thoại, địa chỉ, tên đăng nhập, mật khẩu, quyền quản trị, tình trạng, ảnh đại diện và địa chỉ mail.
* Mỗi sản phẩm có một mã sản phẩm duy nhất để phân biệt. Mỗi mã sản phẩm xác định tất cả các thuộc tính còn lại của sản phẩm đó, bao gồm: tên sản phẩm, giá, mô tả, hình sản phẩm, số lượng, loại sản phẩm, kích cỡ.
* Mỗi hóa đơn có một số hóa đơn duy nhất để phân biệt. Mỗi số hóa đơn xác định tất cả các thuộc tính còn lại của hóa đơn, bao gồm: nhân viên bán, người mua, ngày mua, trị giá và khuyến mãi.
* Chi tiết trong mỗi hóa đơn gồm có những sản phẩm gì với số lượng là bao nhiêu.
* Mỗi phiếu nhập có một mã phiếu duy nhất để phân biệt. Mỗi mã phiếu xác định tất cả các thuộc tính còn lại của phiếu nhập, bao gồm: ngày nhập, nhân viên nhập.
* Chi tiết trong mỗi phiếu nhập gồm có những sản phẩm gì với số lượng nhập là bao nhiêu.
  1. Mô hình thực thể kết hợp:

****

Hình 3.1. Mô hình thực thể kết hợp

* 1. Mô hình CSDL quan hệ:

****

Hình 3.2. Mô hình CSDL quan hệ

* 1. Database diagram trong SQL:

******

Hình 3.3. Database diagram trong SQL

* 1. Cấu trúc các bảng dữ liệu trong SQL:

**KHACHHANG – Danh sách khách hàng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Field Size** | **Descriptions** |
| **MAKH** | varchar | 50 | Mã khách hàng |
| HOTEN | nvarchar | 50 | Tên khách hàng |
| GIOITINH | nvarchar | 5 | Giới tính |
| DCHI | nvarchar | 50 | Địa chỉ |
| SDT | varchar | 50 | Số điện thoại |

**NGUOIDUNG – Danh sách nhân viên/người dùng**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Field Size** | **Descriptions** |
| **MAND** | varchar | 50 | Mã người dùng |
| TENND | nvarchar | 50 | Tên người dùng |
| NGSINH | smalldatetime |  | Ngày sinh |
| GIOITINH | nvarchar | 5 | Giới tính |
| SDT | char | 50 | Số điện thoại |
| DIACHI | nvarchar | 50 | Địa chỉ |
| USERNAME | char | 50 | Tên đăng nhập |
| PASS | nvarchar | max | Mật khẩu |
| QTV | bit |  | Quyền  (0: user, 1: admin) |
| TTND | bit |  | Tình trạng  (0: nghỉ việc,  1: đang làm việc) |
| AVA | varchar | max | Ảnh đại diện |
| MAIL | varchar | 100 | Địa chỉ mail |

**SANPHAM – Danh sách sản phẩm**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Field Size** | **Descriptions** |
| **MASP** | varchar | 50 | Mã sản phẩm |
| TENSP | nvarchar | 50 | Tên sản phẩm |
| GIA | int |  | Giá sản phẩm |
| MOTA | nvarchar | max | Mô tả |
| HINHSP | nvarchar | max | Đường dẫn hình sản phẩm |
| SL | int |  | Số lượng |
| LOAISP | nvarchar | 50 | Loại sản phẩm |
| SIZE | nvarchar | 50 | Kích cỡ |

**HOADON – Danh sách hóa đơn**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Field Size** | **Descriptions** |
| **SOHD** | int |  | Số hóa đơn |
| MAND | varchar | 50 | Mã người dùng |
| MAKH | varchar | 50 | Mã khách hàng |
| NGHD | smalldatetime |  | Ngày mua |
| TRIGIA | int |  | Trị giá |
| KHUYENMAI | int |  | Khuyến mãi |

**PHIEUNHAP – Danh sách phiếu nhập**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Field Size** | **Descriptions** |
| **MAPN** | int |  | Mã phiếu nhập |
| MAND | varchar | 50 | Mã người dùng |
| NGAYNHAP | smalldatetime |  | Ngày nhập |

**CTHD – Chi tiết hóa đơn**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Field Size** | **Descriptions** |
| **SOHD** | int |  | Số hóa đơn |
| **MASP** | varchar | 50 | Mã sản phẩm |
| SL | int |  | Số lượng |

**CTPN – Chi tiết phiếu nhập**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Field Type** | **Field Size** | **Descriptions** |
| **MAPN** | int |  | Mã phiếu nhập |
| **MASP** | varchar | 50 | Mã sản phẩm |
| SL | int |  | Số lượng |

Chương 4:   
XÂY DỰNG ỨNG DỤNG

* 1. Màn hình chính:
     1. Giao diện:

Graphical user interface, application

Description automatically generated

Nhấn để đăng xuất

Nhấn để chuyển danh mục

Hình 4.1. Giao diện màn hình chính khi đăng nhập dưới quyền quản lý

Graphical user interface, application

Description automatically generated

Hình 4.2. Giao diện màn hình chính khi đăng nhập dưới quyền nhân viên

* + 1. Mô tả chức năng:
  + Khi đăng nhập thành công, sẽ xuất hiện màn hình chính.
  + Nhấn nút  sẽ đóng chương trình.
  + Nhấn nút  sẽ thu nhỏ chương trình.
  + Nhấn các nút trên thanh điều hướng để chuyển danh mục.
  + Nhấn nút đăng xuất để trở về màn hình đăng nhập.
    1. Kỹ thuật xử lý:
* Hiệu chỉnh các thuộc tính trong bảng Properties:

 ,  : tắt thanh tiêu đề mặc định của window

A picture containing Teams

Description automatically generated : chỉnh nền cửa sổ về trong suốt để bỏ đi các góc nhọn và hiển thị các góc bo tròn trong thiết kế giao diện.

* Source code xử lý:
* **Chức năng chuyển danh mục:**

void switchtab(MainWindow p)

{

int index = int.Parse(Name);

switch (index)

{

case 0:

{

\_Loadwd(p);

p.Main.NavigationService.Navigate(new HomeView());

break;

}

case 1:

{

\_Loadwd(p);

p.Main.NavigationService.Navigate(new OrderView());

break;

}

case 2:

{

\_Loadwd(p);

p.Main.NavigationService.Navigate(new ProductsView());

break;

}

case 3:

{

\_Loadwd(p);

p.Main.NavigationService.Navigate(new CustomerView());

break;

}

case 4:

{

\_Loadwd(p);

p.Main.NavigationService.Navigate(new ImportView());

break;

}

case 5:

{

\_Loadwd(p);

p.Main.NavigationService.Navigate(new ReportView());

break;

}

case 6:

{

\_Loadwd(p);

p.Main.NavigationService.Navigate(new QLNVView());

break;

}

case 7:

{

\_Loadwd(p);

p.Main.NavigationService.Navigate(new SettingView());

break;

}

default:

{

break;

}

}

}

* **Kiểm tra quyền của người dùng:**

public void LoadQuyen(MainWindow p)

{

p.Quyen.Text = User.QTV ? "Quản lý" : "Nhân viên";

}

* **Tải dữ liệu cho màn hình:**

void \_Loadwd(MainWindow p)

{

if (LoginViewModel.IsLogin)

{

string a = Const.TenDangNhap;

User = DataProvider.Ins.DB.NGUOIDUNGs.Where(x => x.USERNAME == a).FirstOrDefault();

Const.ND = User;

SetQuanLy = User.QTV ? Visibility.Visible : Visibility.Collapsed;

Const.Admin = User.QTV;

Ava = User.AVA;

LoadTenND(p);

}

}

* **Chức năng đăng xuất:**

void LogOut(MainWindow p)

{

LoginWindow login = new LoginWindow();

login.Show();

p.Close();

}

* **Tải tên người dùng:**

public void LoadTenND(MainWindow p)

{

p.TenDangNhap.Text = string.Join(" ", User.TENND.Split().Reverse().Take(2).Reverse());

}

1. 1. Màn hình đăng nhập:
      1. Giao diện:

Graphical user interface, application

Description automatically generated

Nhấn để đăng nhập

Nhấn để đăng ký tài khoản

Nhấn để mở cửa sổ lấy lại mật khẩu

Điền thông tin đăng nhập

Hình 4.3. Giao diện màn hình đăng nhập

* + 1. Mô tả chức năng:
* Khi khởi chạy sẽ xuất hiện form đăng nhập.
* Khi nhấn nút đăng nhập, nếu người dùng nhập thông tin đăng nhập hợp lệ sẽ hiện màn hình chính và các mục menu theo phân quyền. Nếu không, hiện thông báo thông tin đăng nhập không hợp lệ.
* Khi nhấn nút đăng ký sẽ xuất hiện cửa sổ đăng ký.
* Khi nhấn nút quên mật khẩu sẽ xuất hiện cửa sổ quên mật khẩu.
  + 1. Kỹ thuật xử lý:
* Hiệu chỉnh các thuộc tính trong bảng Properties:

 : chỉnh nền cửa sổ về trong suốt để bỏ đi các góc nhọn và hiển thị các góc bo tròn trong thiết kế giao diện.

 ,  : tắt thanh tiêu đề mặc định của WPF.



 : không thể điều chỉnh kích thước màn hình.

* Source code xử lý:
* **Nhấn nút đăng nhập:**

public void login(LoginWindow p)

{

try

{

if (p == null) return;

string PassEncode = MD5Hash(Base64Encode(Password));

var accCount = DataProvider.Ins.DB.NGUOIDUNGs.Where(x => x.USERNAME == Username && x.PASS == PassEncode).Count();

if (accCount > 0)

{

IsLogin = true;

Const.TenDangNhap = Username;

MainWindow mainWindow = new MainWindow();

mainWindow.Show();

Username = "";

p.Hide();

}

else

{

MessageBox.Show("Tên đăng nhập hoặc mật khẩu không đúng!", "Thông báo", MessageBoxButton.OK);

}

}

catch

{

MessageBox.Show("Mất kết nối đến cơ sở dữ liệu!", "Thông báo", MessageBoxButton.OK);

}

}

* **Nhấn nút đăng kí:**

void \_RegisterCommand(LoginWindow parameter)

{

RegisterView registerView = new RegisterView();

registerView.ShowDialog();

}

* **Mã hóa password bằng base64**[11] **và MD5**[12]**:**

public static string Base64Encode(string plainText)

{

var plainTextBytes = System.Text.Encoding.UTF8.GetBytes(plainText);

return System.Convert.ToBase64String(plainTextBytes);

}

public static string MD5Hash(string input)

{

StringBuilder hash = new StringBuilder();

MD5CryptoServiceProvider md5provider = new MD5CryptoServiceProvider();

byte[] bytes = md5provider.ComputeHash(new UTF8Encoding().GetBytes(input));

for (int i = 0; i < bytes.Length; i++)

{

hash.Append(bytes[i].ToString("x2"));

}

return hash.ToString();

}

* **Mở cửa sổ quên mật khẩu:**

void \_ForgetPassCommand(LoginWindow parameter)

{

ForgetPassView forgetPassView = new ForgetPassView();

forgetPassView.ShowDialog();

}

* 1. Màn hình đăng ký:
     1. Giao diện:

Graphical user interface, application, Teams

Description automatically generated

Nhấn chọn giới tính

Ngày sinh không thể nhập tay, chỉ có thể chọn ở đây

Nhấn để chọn ảnh đại diện

Nhấn để đăng ký tài khoản

Hình 4.4. Giao diện màn hình đăng ký

* + 1. Mô tả chức năng:
* Khi nhấn nút đăng ký ở màn hình đăng nhập sẽ hiển thị form đăng ký.
* Khi nhấn nút Thêm ảnh đại diện hiện ra cửa sổ file để chọn hình ảnh.
* Khi nhấn nút Đăng ký, nếu thông tin hợp lệ sẽ hiện thông báo đăng ký thành công, ngược lại sẽ hiện thông báo thông tin không hợp lệ.
  + 1. Kỹ thuật xử lý:
* Hiệu chỉnh các thuộc tính trong bảng Properties:

 : chỉnh nền cửa sổ về trong suốt để bỏ đi các góc nhọn và hiển thị các góc bo tròn trong thiết kế giao diện.

 ,  : tắt thanh tiêu đề mặc định của WPF.



 : không thể điều chỉnh kích thước màn hình.

* Source code xử lý:
* **Nhấn nút đăng ký:**

void \_Register(RegisterView parameter)

{

if (parameter.TenND.Text == "" || parameter.GT.Text == "" || parameter.NS.SelectedDate == null || parameter.SDT.Text == "" || parameter.User.Text == "" || Password == ""||parameter.Mail.Text=="")

{

MessageBox.Show("Bạn chưa nhập đầy đủ thông tin !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

int dem = DataProvider.Ins.DB.NGUOIDUNGs.Where(p => p.USERNAME == parameter.User.Text).Count();

if (dem > 0)

{

MessageBox.Show("Tên đăng nhập đã tồn tại !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

foreach(NGUOIDUNG temp in DataProvider.Ins.DB.NGUOIDUNGs)

{

if(temp.MAIL==parameter.Mail.Text)

{

MessageBox.Show("Email này đã được sử dụng !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

}

string match = @"\w+([-+.']\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*";

Regex reg = new Regex(match);

if (!reg.IsMatch(parameter.Mail.Text))

{

MessageBox.Show("Email không hợp lệ !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

string match1 = @"^((09(\d){8})|(086(\d){7})|(088(\d){7})|(089(\d){7})|(01(\d){9}))$";

Regex reg1 = new Regex(match1);

if (!reg1.IsMatch(parameter.SDT.Text))

{

MessageBox.Show("Số điện thoại không hợp lệ !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

MessageBoxResult h = System.Windows.MessageBox.Show("Bạn muốn đăng ký tài khoản ?", "THÔNG BÁO", MessageBoxButton.YesNoCancel, MessageBoxImage.Question);

if (h == MessageBoxResult.Yes)

{

NGUOIDUNG temp = new NGUOIDUNG();

temp.MAND = rdma();

temp.TENND = parameter.TenND.Text;

temp.GIOITINH = parameter.GT.Text;

temp.DIACHI = parameter.DC.Text;

temp.NGSINH = (DateTime)parameter.NS.SelectedDate;

temp.MAIL=parameter.Mail.Text;

temp.SDT = parameter.SDT.Text;

temp.QTV = false;

temp.TTND = true;

temp.USERNAME = parameter.User.Text;

temp.PASS = LoginViewModel.MD5Hash(LoginViewModel.Base64Encode(Password));

if (linkaddimage == "/Resource/Image/addava.png")

temp.AVA = "/Resource/Image/addava.png";

else

temp.AVA = "/Resource/Ava/" + temp.MAND + ((linkaddimage.Contains(".jpg")) ? ".jpg" : ".png").ToString();

DataProvider.Ins.DB.NGUOIDUNGs.Add(temp);

DataProvider.Ins.DB.SaveChanges();

try

{

File.Copy(linkaddimage, Const.\_localLink + @"Resource\Ava\" + temp.MAND + ((linkaddimage.Contains(".jpg")) ? ".jpg" : ".png").ToString(), true);

}

catch { }

MessageBox.Show("Chúc mừng bạn đã đăng ký thành công !", "THÔNG BÁO", MessageBoxButton.OK);

parameter.User.Clear();

parameter.password.Clear();

parameter.TenND.Clear();

parameter.GT.SelectedItem = null;

parameter.NS.SelectedDate = null;

parameter.SDT.Clear();

parameter.DC.Clear();

linkaddimage = "/Resource/Image/addava.png";

parameter.HinhAnh1.ImageSource = new BitmapImage(new Uri(Const.\_localLink + linkaddimage));

}

}

* **Tạo ngẫu nhiên mã người dùng:**

bool check(string m)

{

foreach (NGUOIDUNG temp in DataProvider.Ins.DB.NGUOIDUNGs)

{

if (temp.MAND == m)

return true;

}

return false;

}

string rdma()

{

string ma;

do

{

Random rand = new Random();

ma = "NV" + rand.Next(0, 10000).ToString();

} while (check(ma));

return ma; }

* **Thêm ảnh đại diện:**

void \_AddImage(ImageBrush img)

{

OpenFileDialog open = new OpenFileDialog();

open.Filter = "Image Files(\*.jpg; \*.png)|\*.jpg; \*.png";

if (open.ShowDialog() == true)

{

if (open.FileName != "")

linkaddimage = open.FileName;

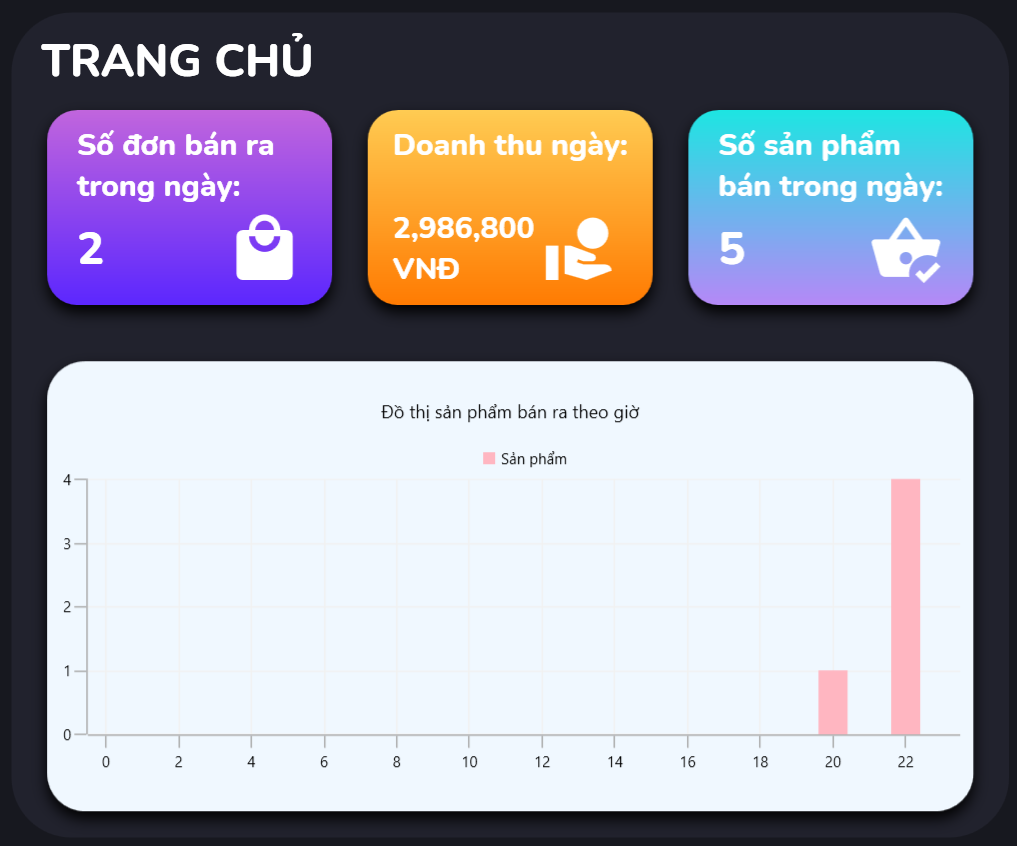
};

Uri fileUri = new Uri(linkaddimage);

img.ImageSource = new BitmapImage(fileUri);

}

* 1. Quản lý danh mục:
     1. Màn hình danh mục trang chủ:
        1. Giao diện:



Rê chuột vào để hiển thị dữ liệu

Hình 4.5. Giao diện màn hình danh mục trang chủ

* + - 1. Mô tả chức năng:
* Khi rê chuột vào cột trong đồ thị sẽ hiển thị số sản phẩm tương ứng.
  + - 1. Kỹ thuật xử lý:
* Source code xử lý:
* **Số đơn trong ngày:**

public void SoDon(HomeView p)

{

int count = DataProvider.Ins.DB.HOADONs.Where(x => x.NGHD.Day == DateTime.Now.Day && x.NGHD.Month == DateTime.Now.Month && x.NGHD.Year == DateTime.Now.Year).Count();

p.DonNgay.Text = count.ToString();

}

* **Doanh thu ngày:**

public void LoadDT(HomeView p)

{

long total = 0;

if (DataProvider.Ins.DB.HOADONs.Where(x => x.NGHD.Day == DateTime.Now.Day && x.NGHD.Month == DateTime.Now.Month && x.NGHD.Year == DateTime.Now.Year).Select(x => x.TRIGIA).Count() != 0)

{

total = DataProvider.Ins.DB.HOADONs.Where(x => x.NGHD.Day == DateTime.Now.Day && x.NGHD.Month == DateTime.Now.Month && x.NGHD.Year == DateTime.Now.Year).Select(x => x.TRIGIA).Sum();

DoanhThu = total.ToString("#,###") + " VNĐ";

}

else DoanhThu = "0 VNĐ";

p.DTNgay.Text = DoanhThu;

}

* **Số sản phẩm bán ra trong ngày:**

void \_LoadSP(HomeView p)

{

int count = 0;

if (DataProvider.Ins.DB.HOADONs.Where(x => x.NGHD.Day == DateTime.Now.Day && x.NGHD.Month == DateTime.Now.Month && x.NGHD.Year == DateTime.Now.Year).Count() > 0)

count = DataProvider.Ins.DB.CTHDs.Where(x => x.HOADON.NGHD.Day == DateTime.Now.Day && x.HOADON.NGHD.Month == DateTime.Now.Month && x.HOADON.NGHD.Year == DateTime.Now.Year).Sum(x => x.SL);

p.SPNgay.Text = count.ToString();

}

* **Đồ thị sản phẩm bán ra theo giờ:**

public void LineChart(HomeView p)

{

var query = from a in DataProvider.Ins.DB.CTHDs

join b in DataProvider.Ins.DB.HOADONs on a.SOHD equals b.SOHD

where a.SOHD == b.SOHD

select new HomeViewModel()

{

SL = a.SL,

Ngay = b.NGHD,

SanPham = a.MASP

};

Data = new List<Result>();

for (int h = 0; h < 24; h++)

{

int value = 0;

if (query.Where(x => x.Ngay.Hour == h && x.Ngay.Day == DateTime.Now.Day && x.Ngay.Month == DateTime.Now.Month && x.Ngay.Year == DateTime.Now.Year).Select(x => x.SL).Count() > 0)

{

value = query.Where(x => x.Ngay.Hour == h && x.Ngay.Day == DateTime.Now.Day && x.Ngay.Month == DateTime.Now.Month && x.Ngay.Year == DateTime.Now.Year).Select(x => x.SL).Sum();

}

Result result = new Result(h, value);

Data.Add(result);

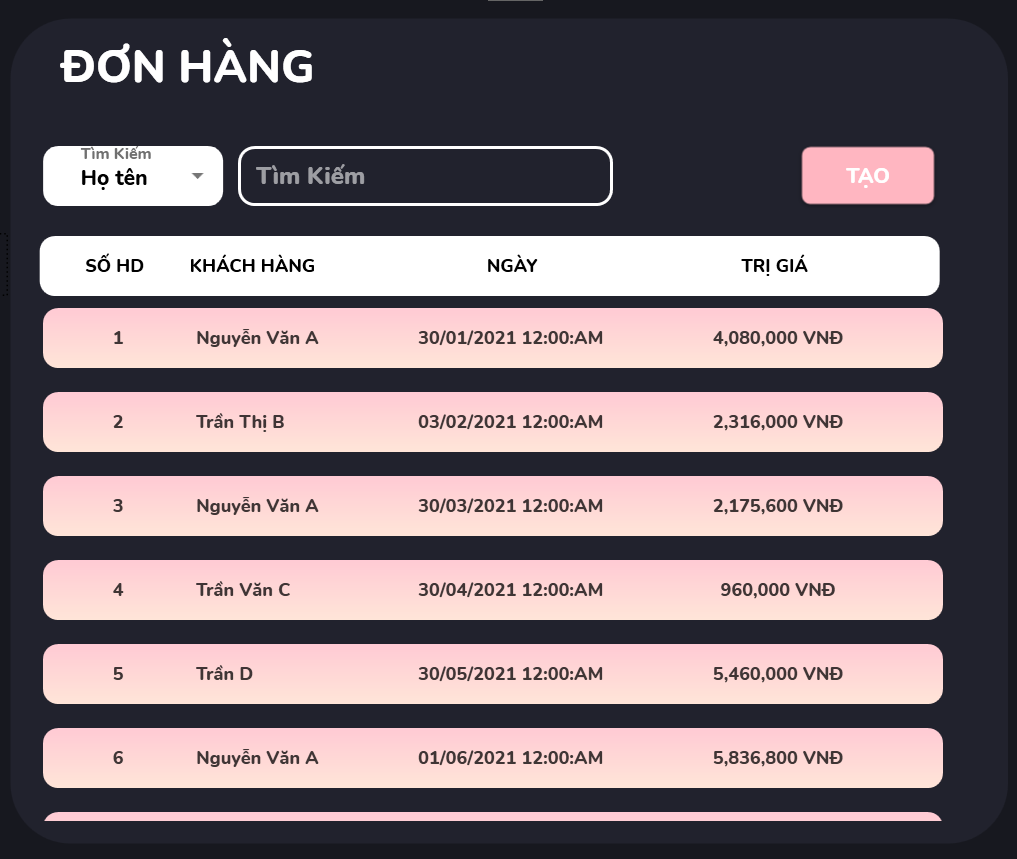
}

p.Chart.ItemsSource = Data;

}

* + 1. Màn hình danh mục đơn hàng:
       1. Giao diện:

Nhấn mở cửa sổ tạo đơn hàng



Chọn trong danh sách để hiển thị cửa sổ thông tin chi tiết

Nhập dữ liệu tìm kiếm theo thuộc tính đã chọn

Chọn thuộc tính để tìm kiếm

Hình 4.6. Giao diện màn hình danh mục đơn hàng

* + - 1. Mô tả chức năng:
* Khi nhấn nút Tạo sẽ xuất hiện form tạo hóa đơn.
* Khi nhấn vào một hóa đơn bất kì trong danh sách sẽ hiện form chi tiết hóa đơn.
* Chọn thuộc tính cần tìm kiếm bằng combobox. Khi gõ ký tự vào thanh tìm kiếm sẽ hiển thị các hóa đơn có ký tự trùng khớp với ký tự trong thanh tìm kiếm theo thuộc tính trong combobox.
  + - 1. Kỹ thuật xử lý:
* Source code xử lý:
* **Load thuộc tính trong combobox khi khởi chạy danh mục:**

void \_LoadCsCommand(OrderView parameter)

{

parameter.cbxChon.SelectedIndex = 0;

}

* **Nhấn nút Tạo:**

void \_OpenAdd(OrderView paramater)

{

AddOrderView addOrder = new AddOrderView();

addOrder.SoHD.Text=rdma().ToString();

addOrder.ShowDialog();

listHD = new ObservableCollection<HOADON>(DataProvider.Ins.DB.HOADONs);

paramater.ListViewHD.ItemsSource = listHD;

paramater.ListViewHD.Items.Refresh();

}

* **Tạo ngẫu nhiên số hóa đơn khi thêm hóa đơn** (tương tự như khi thêm người dùng)
* **Tìm kiếm hóa đơn theo thuộc tính tự chọn:**

void \_SearchCommand(OrderView paramater)

{

ObservableCollection<HOADON> temp = new ObservableCollection<HOADON>();

if (paramater.txbSearch.Text != "")

{

switch (paramater.cbxChon.SelectedItem.ToString())

{

case "Số HD":

{

try

{

foreach (HOADON s in listHD)

{

if (s.SOHD == int.Parse(paramater.txbSearch.Text))

{

temp.Add(s);

}

}

}

catch { }

break;

}

case "Họ tên":

{

foreach (HOADON s in listHD)

{

if (s.KHACHHANG.HOTEN.ToLower().Contains(paramater.txbSearch.Text.ToLower()))

{

temp.Add(s);

}

}

break;

}

case "Ngày":

{

foreach (HOADON s in listHD)

{

if (s.NGHD.ToString("dd/MM/yyyy").Contains(paramater.txbSearch.Text))=

temp.Add(s);

}

break;

}

default:

{

foreach (HOADON s in listHD)

{

if (s.KHACHHANG.HOTEN.ToLower().Contains(paramater.txbSearch.Text.ToLower()))

temp.Add(s);

}

break;

}

}

paramater.ListViewHD.ItemsSource = temp;

}

else

paramater.ListViewHD.ItemsSource = listHD;

}

* **Khởi chạy cửa sổ thông tin hóa đơn:**

void \_Detail(OrderView parameter)

{

DetailOrder detailOrder = new DetailOrder();

HOADON temp = (HOADON) parameter.ListViewHD.SelectedItem;

detailOrder.MaND.Text = temp.NGUOIDUNG.MAND;

detailOrder.TenND.Text= temp.NGUOIDUNG.TENND;

detailOrder.Ngay.Text = temp.NGHD.ToString("dd/MM/yyyy hh:mm tt");

detailOrder.SoHD.Text = temp.SOHD.ToString();

detailOrder.MaKH.Text = temp.MAKH.ToString();

detailOrder.TenKH.Text = temp.KHACHHANG.HOTEN;

detailOrder.KM.Text = temp.KHUYENMAI.ToString()+"%";

List<HienThi> list = new List<HienThi>();

foreach(CTHD a in temp.CTHDs)

{

list.Add(new HienThi(a.MASP, a.SANPHAM.TENSP, a.SANPHAM.SIZE, a.SL,a.SANPHAM.GIA, a.SL \* a.SANPHAM.GIA));

}

detailOrder.ListViewSP.ItemsSource = list;

detailOrder.GG.Text= "- "+String.Format("{0:0,0}", (temp.TRIGIA\*100/(100-temp.KHUYENMAI)) \* temp.KHUYENMAI/100) + " VND";

detailOrder.TT.Text= String.Format("{0:0,0}", temp.TRIGIA) + " VND";

detailOrder.TT1.Text = String.Format("{0:0,0}", temp.TRIGIA) + " VND";

detailOrder.ShowDialog();

parameter.ListViewHD.SelectedItem=null;

listHD = new ObservableCollection<HOADON>(DataProvider.Ins.DB.HOADONs);

parameter.ListViewHD.ItemsSource = listHD;

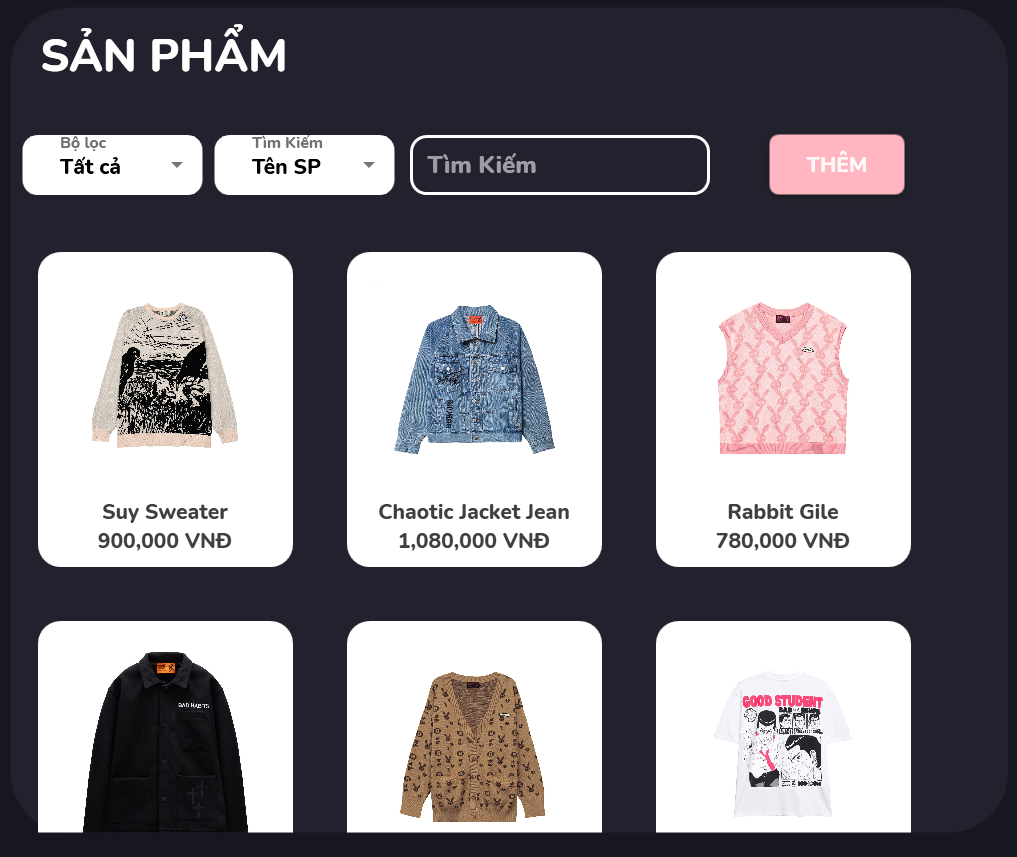
\_SearchCommand(parameter);

}

* + 1. Màn hình danh mục sản phẩm:
       1. Giao diện:

Tìm kiếm theo thuộc tính đã chọn

Chọn thuộc tính



Chọn trong danh sách để hiển thị thông tin chi tiết

Mở cửa sổ thêm sản phẩm

Chọn bộ lọc cho danh sách

Hình 4.7. Giao diện màn hình danh mục sản phẩm

* + - 1. Mô tả chức năng:
* Khi nhấn nút Thêm sẽ xuất hiện form thêm sản phẩm.
* Khi nhấn vào một sản phẩm bất kì trong danh sách sẽ hiện form chi tiết sản phẩm.
* Khi đổi giá trị trong combobox Bộ lọc thì danh sách sẽ hiển thị những sản phẩm có loại sản phẩm ứng với loại trong combobox.
* Chọn tìm kiếm theo tên sản phẩm hoặc theo giá trong combobox tìm kiếm. Khi tìm kiếm theo tên thì danh sách sẽ hiển thị các sản phẩm có tên trùng khớp từng ký tự trong ô tìm kiếm không phân biệt hoa thường. Khi tìm kiếm theo giá thì danh sách sẽ hiển thị các sản phẩm có giá nhỏ hơn hoặc bằng giá trong thanh tìm kiếm.
  + - 1. Kỹ thuật xử lý:
* Source code xử lý:
* **Load các thuộc tính khi khởi chạy danh mục:**

void \_LoadCsCommand(ProductsView parameter)

{

listTK = new ObservableCollection<string>() { "Tên SP", "Giá SP" };

listSP1 = new ObservableCollection<SANPHAM>(DataProvider.Ins.DB.SANPHAMs.Where(p => p.SL >= 0));

listSP = new ObservableCollection<SANPHAM>(listSP1.GroupBy(p => p.TENSP).Select(grp => grp.FirstOrDefault()));

parameter.cbxChon.SelectedIndex = 0;

parameter.cbxChon1.SelectedIndex = 0;

\_Filter(parameter);

\_SearchCommand(parameter);

}

* **Bộ lọc loại sản phẩm:**

void \_Filter(ProductsView parameter)

        {

            switch(parameter.cbxChon1.SelectedIndex.ToString())

            {

                case "0":

                    {

                        listSP = new ObservableCollection<SANPHAM>(listSP1.GroupBy(p => p.TENSP).Select(grp => grp.FirstOrDefault()));

                        parameter.ListViewProduct.ItemsSource = listSP;

                        break;

                    }

                case "1":

                    {

                        listSP = new ObservableCollection<SANPHAM>(listSP1.GroupBy(p => p.TENSP).Select(grp => grp.FirstOrDefault()).Where(p=>p.LOAISP== "Áo sơ mi"));

                        parameter.ListViewProduct.ItemsSource = listSP;

                        break;

                    }

                case "2":

                    {

                        listSP = new ObservableCollection<SANPHAM>(listSP1.GroupBy(p => p.TENSP).Select(grp => grp.FirstOrDefault()).Where(p => p.LOAISP == "Áo thun"));

                        parameter.ListViewProduct.ItemsSource = listSP;

                        break;

                    }

                case "3":

                    {

                        listSP = new ObservableCollection<SANPHAM>(listSP1.GroupBy(p => p.TENSP).Select(grp => grp.FirstOrDefault()).Where(p => p.LOAISP == "Áo trùm đầu"));

                        parameter.ListViewProduct.ItemsSource = listSP;

                        parameter.ListViewProduct.ItemsSource = listSP;

                        break;

                    }

                case "4":

                    {

                        listSP = new ObservableCollection<SANPHAM>(listSP1.GroupBy(p => p.TENSP).Select(grp => grp.FirstOrDefault()).Where(p => p.LOAISP == "Áo khoác"));

                        parameter.ListViewProduct.ItemsSource = listSP;

                        break;

                    }

                case "5":

                    {

                        listSP = new ObservableCollection<SANPHAM>(listSP1.GroupBy(p => p.TENSP).Select(grp => grp.FirstOrDefault()).Where(p => p.LOAISP == "Áo tay dài"));

                        parameter.ListViewProduct.ItemsSource = listSP;

                        break;

                    }

                case "6":

                    {

                        listSP = new ObservableCollection<SANPHAM>(listSP1.GroupBy(p => p.TENSP).Select(grp => grp.FirstOrDefault()).Where(p => p.LOAISP == "Quần"));

                        parameter.ListViewProduct.ItemsSource = listSP;

                        break;

                    }

                case "7":

                    {

                        listSP = new ObservableCollection<SANPHAM>(listSP1.GroupBy(p => p.TENSP).Select(grp => grp.FirstOrDefault()).Where(p => p.LOAISP == "Phụ kiện"));

                        parameter.ListViewProduct.ItemsSource = listSP;

                        break;

                    }

            }

        }

* **Chức năng tìm kiếm theo thuộc tính trong combobox:**

void \_SearchCommand(ProductsView paramater)

{

ObservableCollection<SANPHAM> temp = new ObservableCollection<SANPHAM>();

if (paramater.txbSearch.Text != "")

{

switch (paramater.cbxChon.SelectedItem.ToString())

{

case "Tên SP":

{

foreach (SANPHAM s in listSP)

{

if (s.TENSP.ToLower().Contains(paramater.txbSearch.Text.ToLower()))

{

temp.Add(s);

}

}

break;

}

case "Giá SP":

{

try

{

foreach (SANPHAM s in listSP)

{

if (s.GIA <= int.Parse(paramater.txbSearch.Text))

{

temp.Add(s);

}

}

}

catch { }

break;

}

default:

{

foreach (SANPHAM s in listSP)

{

if (s.TENSP.ToLower().Contains(paramater.txbSearch.Text.ToLower()))

{

temp.Add(s);

}

}

break;

}

}

paramater.ListViewProduct.ItemsSource = temp;

}

else

paramater.ListViewProduct.ItemsSource = listSP;

}

* **Khởi chạy cửa sổ thông tin chi tiết sản phẩm:**

void \_DetailPd(ProductsView paramater)

{

DetailProduct detailProduct = new DetailProduct();

SANPHAM temp = (SANPHAM)paramater.ListViewProduct.SelectedItem;

detailProduct.TenSP.Text = temp.TENSP;

detailProduct.GiaSP.Text = string.Format("{0:0,0}", temp.GIA) + " VNĐ";

detailProduct.LoaiSP.Text = temp.LOAISP;

string SL = listSP1.Where(p => p.TENSP == temp.TENSP&&p.SL>=0).Select(p=>p.SL).Sum().ToString();

detailProduct.SLSP.Text = "Số lượng: " + SL;

detailProduct.kichco.ItemsSource = new ObservableCollection<SANPHAM>(listSP1.Where(p => p.TENSP == temp.TENSP&&p.SL>=0));

detailProduct.Mota.Text = temp.MOTA;

Uri fileUri = new Uri(temp.HINHSP);

detailProduct.HinhAnh.Source = new BitmapImage(fileUri);

detailProduct.ShowDialog();

listSP1 = new ObservableCollection<SANPHAM>(DataProvider.Ins.DB.SANPHAMs.Where(p => p.SL >= 0));

paramater.ListViewProduct.SelectedItem = null;

\_Filter(paramater);

\_SearchCommand(paramater);

}

* **Nhấn nút Thêm:**

void \_AddPdCommand(ProductsView paramater)

{

AddProductView addProductView = new AddProductView();

addProductView.MaSp.Text=rdma();

addProductView.ShowDialog();

listSP1 = new ObservableCollection<SANPHAM>(DataProvider.Ins.DB.SANPHAMs.Where(p=>p.SL>=0));

\_Filter(paramater);

\_SearchCommand(paramater);

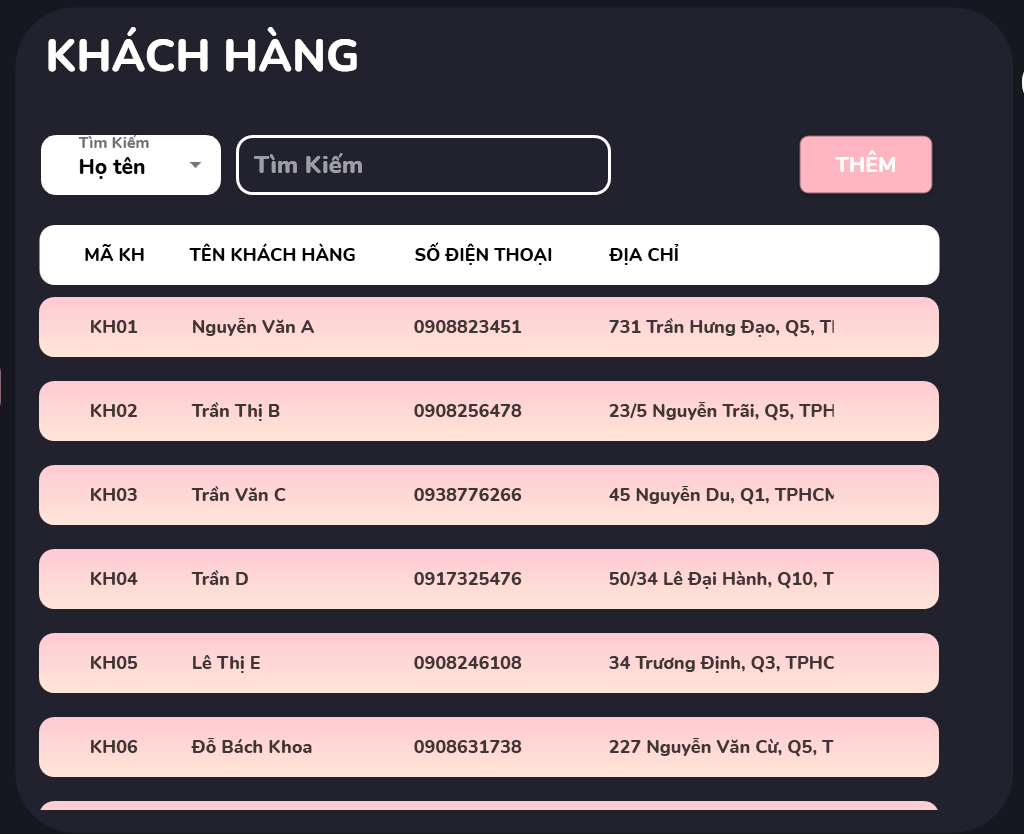
}

* **Tạo ngẫu nhiên mã sản phẩm khi thêm sản phẩm** (tương tự như khi thêm người dùng).
  + 1. Màn hình danh mục khách hàng:
       1. Giao diện:

Nhấn mở cửa sổ thêm khách hàng

Nhập dữ liệu tìm kiếm theo thuộc tính đã chọn

Chọn thuộc tính để tìm kiếm



Chọn trong danh sách để hiển thị cửa sổ thông tin chi tiết

Hình 4.8. Giao diện màn hình danh mục khách hàng

* + - 1. Mô tả chức năng:
* Khi nhấn nút Thêm sẽ xuất hiện form thêm khách hàng.
* Khi nhấn vào một khách hàng bất kì trong danh sách sẽ hiện form chi tiết thông tin khách hàng.
* Chọn thuộc tính cần tìm kiếm bằng combobox. Khi gõ ký tự vào thanh tìm kiếm sẽ hiển thị các khách hàng có ký tự trùng khớp với ký tự trong thanh tìm kiếm theo thuộc tính trong combobox.
  + - 1. Kỹ thuật xử lý:
* Source code xử lý:
* **Load thuộc tính trong combobox khi khởi chạy danh mục:**

void \_LoadCsCommand(CustomerView parameter)

{

parameter.cbxChon.SelectedIndex = 0;

}

* **Chức năng tìm kiếm theo thuộc tính trong combobox:**

void \_SearchCommand(CustomerView paramater)

{

ObservableCollection<KHACHHANG> temp = new ObservableCollection<KHACHHANG>();

if (paramater.txbSearch.Text != "")

{

switch (paramater.cbxChon.SelectedItem.ToString())

{

case "Mã KH":

{

foreach (KHACHHANG s in listKH)

{

if (s.MAKH.Contains(paramater.txbSearch.Text))

temp.Add(s);

}

break;

}

case "Họ tên":

{

foreach (KHACHHANG s in listKH)

{

if (s.HOTEN.ToLower().Contains(paramater.txbSearch.Text.ToLower()))

temp.Add(s);

}

break;

}

case "SĐT":

{

foreach (KHACHHANG s in listKH)

{

if (s.SDT.Contains(paramater.txbSearch.Text))

temp.Add(s);

}

break;

}

default:

{

foreach (KHACHHANG s in listKH)

{

if (s.HOTEN.Contains(paramater.txbSearch.Text))

temp.Add(s);

}

break;

}

}

paramater.ListViewKH.ItemsSource = temp;

} else

paramater.ListViewKH.ItemsSource = listKH; }

* **Khởi chạy cửa sổ chi tiết thông tin khách hàng:**

void \_DetailCs(CustomerView paramater)

{

DetailCustomerView detailCustomerView = new DetailCustomerView();

KHACHHANG temp=(KHACHHANG) paramater.ListViewKH.SelectedItem;

detailCustomerView.MaKH.Text = temp.MAKH;

detailCustomerView.TenKH.Text = temp.HOTEN;

detailCustomerView.SDT.Text = temp.SDT;

detailCustomerView.GT.Text = temp.GIOITINH;

detailCustomerView.DC.Text = temp.DCHI;

int doanhso=0;

foreach(HOADON a in DataProvider.Ins.DB.HOADONs)

{

if (a.MAKH == temp.MAKH)

doanhso += a.TRIGIA;

}

detailCustomerView.DS.Text = String.Format("{0:0,0}", doanhso) + " VND"; ;

string hang = "Đồng";

if (doanhso > 2000000 && doanhso <= 5000000)

hang = "Bạc";

else if (doanhso > 5000000 && doanhso <= 10000000)

hang = "Vàng";

else if(doanhso>10000000)

hang = "Kim cương";

detailCustomerView.Rank.Text = hang;

detailCustomerView.ShowDialog();

listKH = new ObservableCollection<KHACHHANG>(DataProvider.Ins.DB.KHACHHANGs);

paramater.ListViewKH.ItemsSource = listKH;

paramater.ListViewKH.SelectedItem = null;

}

* **Nhấn nút Thêm:**

void \_AddCs(CustomerView paramater)

{

AddCustomerView addCustomerView = new AddCustomerView();

addCustomerView.MaKH.Text=rdma();

addCustomerView.ShowDialog();

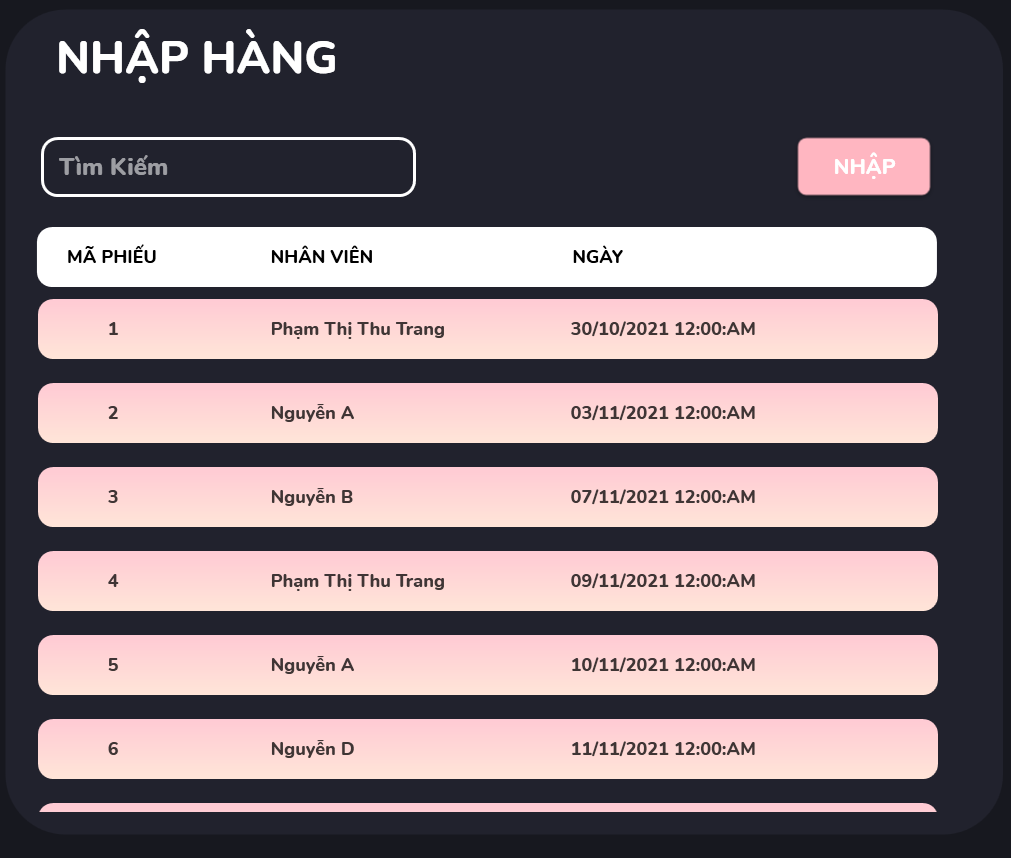
listKH = new ObservableCollection<KHACHHANG>(DataProvider.Ins.DB.KHACHHANGs);

paramater.ListViewKH.ItemsSource = listKH;

}

* **Tạo mã khách hàng ngẫu nhiên khi thêm khách hàng:** (tương tự như khi thêm người dùng).
  + 1. Màn hình danh mục nhập hàng:
       1. Giao diện:

Nhấn mở cửa sổ nhập hàng



Tìm kiếm theo tên nhân viên

Chọn trong danh sách để hiển thị cửa sổ thông tin chi tiết

Hình 4.9. Giao diện màn hình danh mục nhập hàng

* + - 1. Mô tả chức năng:
* Khi nhấn nút Nhập sẽ xuất hiện form nhập hàng.
* Khi nhấn vào một phiếu nhập bất kì trong danh sách sẽ hiện form chi tiết phiếu nhập.
* Khi gõ ký tự vào thanh tìm kiếm sẽ hiện các phiếu nhập với tên nhân viên ứng với từng ký tự trên thanh tìm kiếm không phân biệt hoa thường.
  + - 1. Kỹ thuật xử lý:
* Source code xử lý:
* **Nhấn nút nhập:**

void \_OpenAdd(ImportView paramater)

{

AddImportView addOrder = new AddImportView();

addOrder.MaPN.Text = rdma().ToString();

addOrder.ShowDialog();

listPN = new ObservableCollection<PHIEUNHAP>(DataProvider.Ins.DB.PHIEUNHAPs);

}

* **Tạo ngẫu nhiên mã phiếu nhập** (tương tự như khi thêm người dùng)
* **Nhấn phiếu nhập trong danh sách:**

void \_Detail(ImportView p)

{

DetailImport detailImport = new DetailImport();

PHIEUNHAP temp = (PHIEUNHAP)p.ListViewPN.SelectedItem;

detailImport.MaND.Text = temp.NGUOIDUNG.MAND;

detailImport.TenND.Text = temp.NGUOIDUNG.TENND;

detailImport.Ngay.Text = temp.NGAYNHAP.ToString("dd/MM/yyyy hh:mm tt");

detailImport.MaPN.Text = temp.MAPN.ToString();

List<Display> list = new List<Display>();

int tong = 0;

foreach (CTPN a in temp.CTPNs)

{

list.Add(new Display(a.MASP, a.SANPHAM.TENSP, a.SANPHAM.SIZE, (int)((float)a.SANPHAM.GIA\* 5/6), a.SL,(int)((float)(a.SL\*a.SANPHAM.GIA)\*5/6)));

tong += (int)((float)(a.SL \* a.SANPHAM.GIA) \* 5/6);

}

detailImport.ttn.Text = String.Format("{0:#,###}"+" VND",tong);

detailImport.ListViewSP.ItemsSource = list;

detailImport.ShowDialog();

p.ListViewPN.SelectedItem = null;

listPN = new ObservableCollection<PHIEUNHAP>(DataProvider.Ins.DB.PHIEUNHAPs);

\_SearchCommand(p);

}

* T**ìm kiếm phiếu nhập theo tên nhân viên:**

void \_SearchCommand(ImportView p)

{

ObservableCollection<PHIEUNHAP> temp = new ObservableCollection<PHIEUNHAP>();

if (p.txbSearch.Text != "")

{

foreach (PHIEUNHAP s in listPN)

{

if (s.NGUOIDUNG.TENND.ToLower().Contains(p.txbSearch.Text.ToLower()))

{

temp.Add(s);

}

}

p.ListViewPN.ItemsSource = temp;

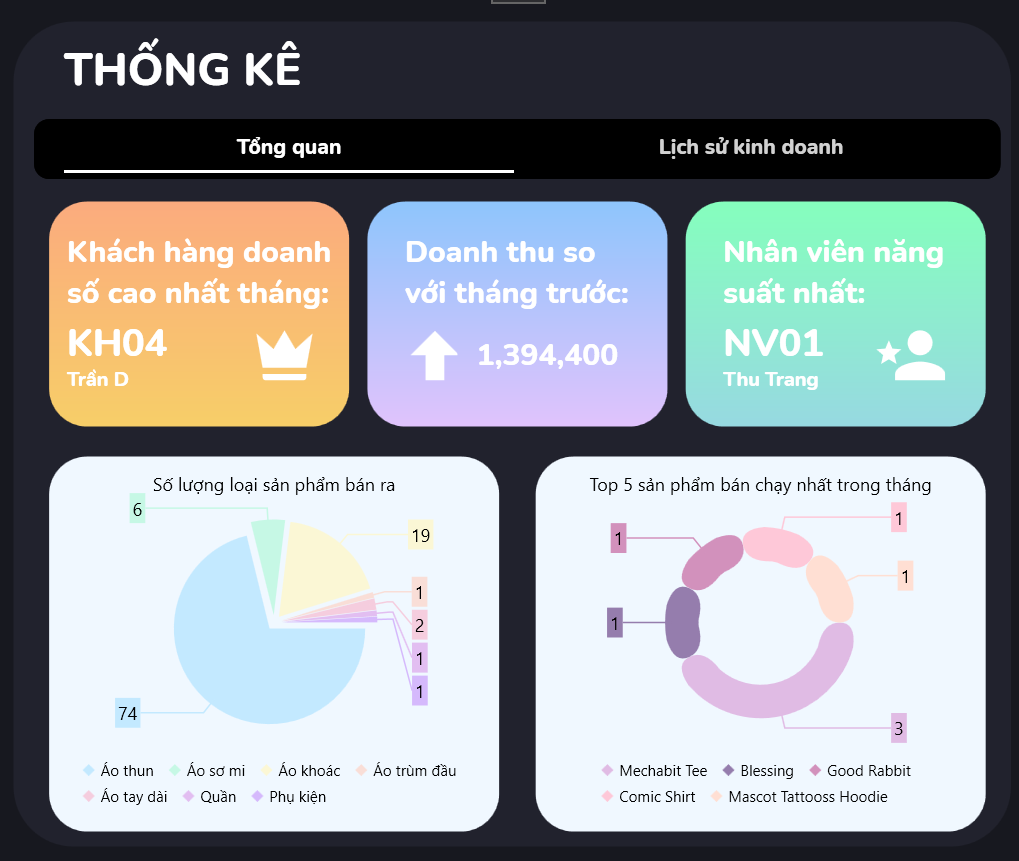
}

else

p.ListViewPN.ItemsSource = listPN;

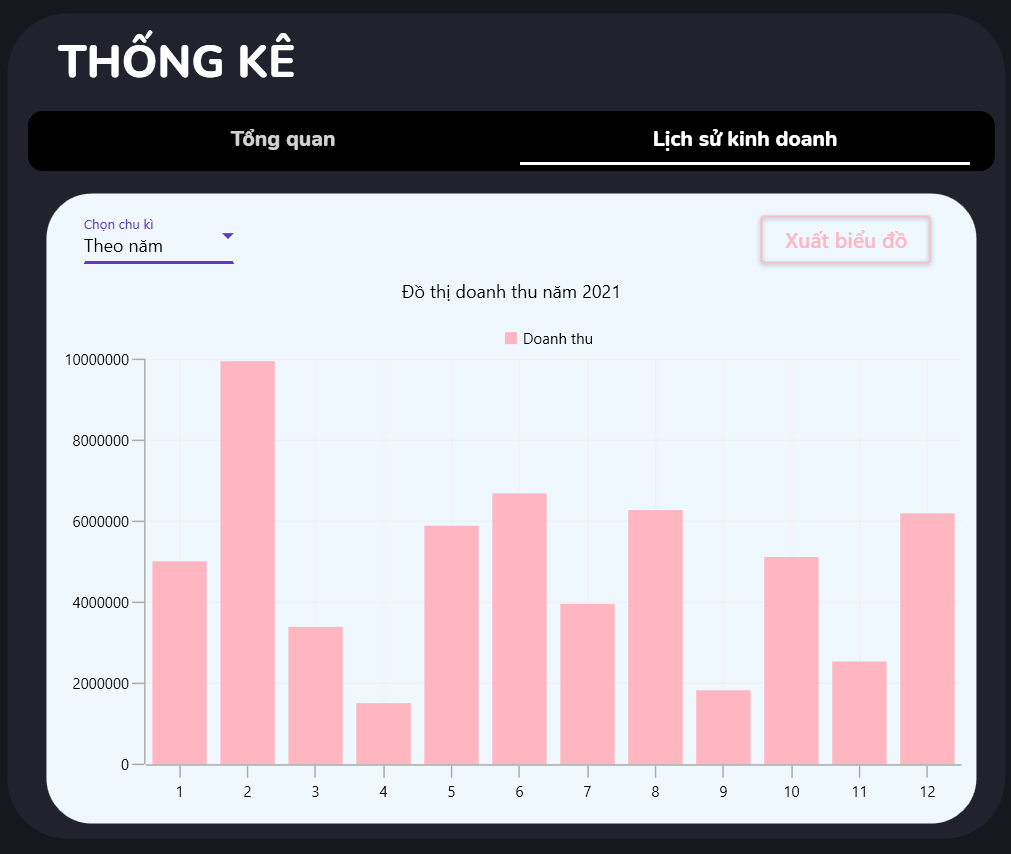
}

* + 1. Màn hình danh mục thống kê:
       1. Giao diện:



Nhấn chọn danh mục

Hình 4.10. Giao diện màn hình danh mục thống kê trang 1



Nhấn để xuất biểu đồ và gửi qua mail

Chọn chu kì cho biểu đồ

Hình 4.11. Giao diện màn hình danh mục thống kê trang 2, chu kì theo năm



Hình 4.12. Giao diện màn hình danh mục thống kê trang 2 chu kì theo tháng

* + - 1. Mô tả chức năng:
* Khi nhấn nút trên thanh điều hướng danh mục thống kê sẽ hiện lên trang tương ứng.
* Khi chọn mục bất kì trong combobox biểu đồ sẽ thay đổi theo mục đã chọn.
* Khi rê chuột vào cột trong biểu đồ cột sẽ hiện giá trị tương ứng.
* Khi nhấn nút Xuất biểu đồ sẽ lưu biểu đồ doanh thu theo tháng, biểu đồ loại sản phẩm và top 5 sản phẩm bán chạy nhất về máy và gửi qua mail.
  + - 1. Kỹ thuật xử lý:
* Source code xử lý:
* **Khởi chạy màn hình:**

public void \_loadwd(ReportView p)

{

p.Combobox.SelectedIndex = 1;

ColChart(p);

SetMain = Visibility.Visible;

SetBills = Visibility.Hidden;

}

* **Chuyển trang:**

void switchtab(ReportView p)

{

int index = int.Parse(Name);

switch (index)

{

case 0:

{

SetMain = Visibility.Visible;

SetBills = Visibility.Hidden;

break;

}

case 1:

{

listHD = new ObservableCollection<HOADON>(DataProvider.Ins.DB.HOADONs);

SetMain = Visibility.Hidden;

SetBills = Visibility.Visible;

break;

}

default:

{

break;

}

}

}

* **Khách hàng có doanh số cao nhất tháng:**

public void KHCount(ReportView p)

{

MaxSell = 0;

foreach (HOADON hd in DataProvider.Ins.DB.HOADONs.Where(x => x.NGHD.Month == DateTime.Now.Month))

{

int temp = DataProvider.Ins.DB.HOADONs.Where(x => x.MAKH == hd.MAKH).Count();

if (MaxSell < temp)

{

MaxSell = temp;

BestKH = hd.MAKH;

KHName = hd.KHACHHANG.HOTEN;

}

}

if (MaxSell == 0)

{

BestKH = "";

KHName = "(chưa có)";

}

p.MaxKH.Text = BestKH;

p.KHName.Text = KHName;

}

* **Doanh thu so với tháng trước:**

public void DTTrend(ReportView p)

{

if(DataProvider.Ins.DB.HOADONs.Where(x => x.NGHD.Month == DateTime.Now.Month).Count()==0)

{

ThisMonth = 0;

}

else

{

ThisMonth = DataProvider.Ins.DB.HOADONs.Where(x => x.NGHD.Month == DateTime.Now.Month).Sum(x => x.TRIGIA);

}

if(DataProvider.Ins.DB.HOADONs.Where(x => x.NGHD.Month == DateTime.Now.Month-1).Count() == 0)

{

LastMonth = 0;

}

else

{

LastMonth = DataProvider.Ins.DB.HOADONs.Where(x => x.NGHD.Month == DateTime.Now.Month - 1).Sum(x => x.TRIGIA);

}

long temp = ThisMonth - LastMonth;

if (temp >= 0)

{

p.DTTrend.Text = "+" + temp.ToString("#,### VNĐ");

p.DTTrend.Foreground = new SolidColorBrush(Color.FromRgb(255, 255, 255));

Up = Visibility.Visible;

Down = Visibility.Collapsed;

}

else

{

p.DTTrend.Text = temp.ToString("#,### VNĐ");

p.DTTrend.Foreground = new SolidColorBrush(Color.FromRgb(255, 255, 255));

Up = Visibility.Collapsed;

Down = Visibility.Visible;

}

}

* **Nhân viên năng suất nhất:**

public void NVCount(ReportView p)

{

MaxNV = int.MinValue;

foreach (HOADON hd in DataProvider.Ins.DB.HOADONs)

{

int temp = DataProvider.Ins.DB.HOADONs.Where(x => x.MAND == hd.MAND).Count();

if (MaxNV < temp)

{

MaxNV = temp;

NVBest = hd.MAND;

}

}

NVName = DataProvider.Ins.DB.NGUOIDUNGs.Where(x => x.MAND == NVBest).Select(x => x.TENND).FirstOrDefault();

p.NVBest.Text = NVBest;

p.NVName.Text = string.Join(" ", NVName.Split().Reverse().Take(2).Reverse());

}

* **Biểu đồ loại sản phẩm:**

void PieChart(ReportView p)

{

int ts = 0, sh = 0, jk = 0, sw = 0, ac = 0, sp = 0, ho = 0;

if (DataProvider.Ins.DB.CTHDs.Where(x => x.SANPHAM.LOAISP == "Áo thun").Count() > 0)

ts = DataProvider.Ins.DB.CTHDs.Where(x => x.SANPHAM.LOAISP == "Áo thun").Sum(x => x.SL);

if (DataProvider.Ins.DB.CTHDs.Where(x => x.SANPHAM.LOAISP == "Áo sơ mi").Count() > 0)

sh = DataProvider.Ins.DB.CTHDs.Where(x => x.SANPHAM.LOAISP == "Áo sơ mi").Sum(x => x.SL);

if (DataProvider.Ins.DB.CTHDs.Where(x => x.SANPHAM.LOAISP == "Áo khoác").Count() > 0)

jk = DataProvider.Ins.DB.CTHDs.Where(x => x.SANPHAM.LOAISP == "Áo khoác").Sum(x => x.SL);

if (DataProvider.Ins.DB.CTHDs.Where(x => x.SANPHAM.LOAISP == "Áo tay dài").Count() > 0)

sw = DataProvider.Ins.DB.CTHDs.Where(x => x.SANPHAM.LOAISP == "Áo tay dài").Sum(x => x.SL);

if (DataProvider.Ins.DB.CTHDs.Where(x => x.SANPHAM.LOAISP == "Phụ kiện").Count() > 0)

ac = DataProvider.Ins.DB.CTHDs.Where(x => x.SANPHAM.LOAISP == "Phụ kiện").Sum(x => x.SL);

if (DataProvider.Ins.DB.CTHDs.Where(x => x.SANPHAM.LOAISP == "Quần").Count() > 0)

sp = DataProvider.Ins.DB.CTHDs.Where(x => x.SANPHAM.LOAISP == "Quần").Sum(x => x.SL);

if (DataProvider.Ins.DB.CTHDs.Where(x => x.SANPHAM.LOAISP == "Áo trùm đầu").Count() > 0)

ho = DataProvider.Ins.DB.CTHDs.Where(x => x.SANPHAM.LOAISP == "Áo trùm đầu").Sum(x => x.SL);

Reviews = new List<Review>();

Review r1 = new Review()

{

Type = "Áo thun",

Num = ts

};

Review r2 = new Review()

{

Type = "Áo sơ mi",

Num = sh

};

Review r3 = new Review()

{

Type = "Áo khoác",

Num = jk

};

Review r4 = new Review()

{

Type = "Áo trùm đầu",

Num = ho

};

Review r5 = new Review()

{

Type = "Áo tay dài",

Num = sw

}; Review r6 = new Review()

{

Type = "Quần",

Num = sp

};

Review r7 = new Review()

{

Type = "Phụ kiện",

Num = ac

};

Reviews.Add(r1);

Reviews.Add(r2);

Reviews.Add(r3);

Reviews.Add(r4);

Reviews.Add(r5);

Reviews.Add(r6);

Reviews.Add(r7);

p.Pie.ItemsSource = Reviews;

p.Pie.AdornmentsInfo = new Syncfusion.UI.Xaml.Charts.ChartAdornmentInfo()

{

ShowLabel = true,

ShowConnectorLine = true,

Margin = new Thickness(2)

};

p.Pie.ExplodeOnMouseClick = true;

}

* **Biểu đồ top 5 sản phẩm bán chạy nhất:**

void DonutChart(ReportView p)

{

var query = from a in DataProvider.Ins.DB.CTHDs

join b in DataProvider.Ins.DB.SANPHAMs on a.MASP equals b.MASP

where a.MASP == b.MASP && a.HOADON.NGHD.Month == DateTime.Now.Month && a.HOADON.NGHD.Year == DateTime.Now.Year

select new ReportViewModel()

{

SL = a.SL,

MaSP = a.MASP,

TenSP = b.TENSP,

Ngay = a.HOADON.NGHD

};

string sp1 = "", sp2 = "", sp3 = "", sp4 = "", sp5 = "";

int max1 = 0;

foreach (ReportViewModel obj in query)

{

int temp = query.Where(x => x.TenSP == obj.TenSP && x.Ngay.Month == DateTime.Now.Month && x.Ngay.Year == DateTime.Now.Year).Sum(x => x.SL);

if (max1 < temp)

{

max1 = temp;

sp1 = obj.TenSP;

}

}

int max2 = 0;

foreach (ReportViewModel obj in query)

{

if (obj.TenSP == sp1) continue;

int temp = query.Where(x => x.TenSP == obj.TenSP && x.Ngay.Month == DateTime.Now.Month && x.Ngay.Year == DateTime.Now.Year).Sum(x => x.SL);

if (max2 < temp)

{

max2 = temp;

sp2 = obj.TenSP;

}

}

int max3 = 0;

foreach (ReportViewModel obj in query)

{

if (obj.TenSP == sp1 || obj.TenSP == sp2) continue;

int temp = query.Where(x => x.TenSP == obj.TenSP && x.Ngay.Month == DateTime.Now.Month && x.Ngay.Year == DateTime.Now.Year).Sum(x => x.SL);

if (max3 < temp)

{

max3 = temp;

sp3 = obj.TenSP;

}

}

int max4 = 0;

foreach (ReportViewModel obj in query)

{

if (obj.TenSP == sp1 || obj.TenSP == sp2 || obj.TenSP == sp3) continue;

int temp = query.Where(x => x.TenSP == obj.TenSP && x.Ngay.Month == DateTime.Now.Month && x.Ngay.Year == DateTime.Now.Year).Sum(x => x.SL);

if (max4 < temp)

{

max4 = temp;

sp4 = obj.TenSP;

}

}

int max5 = 0;

foreach (ReportViewModel obj in query)

{

if (obj.TenSP == sp1 || obj.TenSP == sp2 || obj.TenSP == sp3 || obj.TenSP == sp4) continue;

int temp = query.Where(x => x.TenSP == obj.TenSP && x.Ngay.Month == DateTime.Now.Month && x.Ngay.Year == DateTime.Now.Year).Sum(x => x.SL);

if (max5 < temp)

{

max5 = temp;

sp5 = obj.TenSP;

}

}

Reviews = new List<Review>();

Review r1 = new Review()

{

Type = sp1,

Num = max1

};

Review r2 = new Review()

{

Type = sp2,

Num = max2

};

Review r3 = new Review()

{

Type = sp3,

Num = max3

};

Review r4 = new Review()

{

Type = sp4,

Num = max4

};

Review r5 = new Review()

{

Type = sp5,

Num = max5

};

Reviews.Add(r1);

Reviews.Add(r2);

Reviews.Add(r3);

Reviews.Add(r4);

Reviews.Add(r5);

p.Donut.ItemsSource = Reviews;

p.Donut.AdornmentsInfo = new Syncfusion.UI.Xaml.Charts.ChartAdornmentInfo()

{

ShowLabel = true,

ShowConnectorLine = true,

Margin = new Thickness(2)

};

}

* **Biểu đồ cột:**

public void ColChart(ReportView p)

{

if (p.Combobox.SelectedIndex == 0)

{

p.SampleChart.Header = "Đồ thị doanh thu năm " + DateTime.Now.Year.ToString();

var query = DataProvider.Ins.DB.HOADONs.Select(x => new ReportViewModel()

{

Tien = x.TRIGIA,

Ngay = x.NGHD

});

YDatas = new List<YData>();

for (int h = 1; h < 13; h++)

{

long value = 0;

if (query.Where(x => x.Ngay.Month == h && x.Ngay.Year == DateTime.Now.Year).Select(x => x.Tien).Count() > 0)

{

value = query.Where(x => x.Ngay.Month == h && x.Ngay.Year == DateTime.Now.Year).Select(x => x.Tien).Sum();

}

YData result = new YData(h, value);

YDatas.Add(result);

}

}

else

{

p.SampleChart.Header = "Đồ thị doanh thu tháng " + DateTime.Now.Month.ToString() + "/" + DateTime.Now.Year.ToString();

var query = DataProvider.Ins.DB.HOADONs.Select(x => new ReportViewModel()

{

Tien = x.TRIGIA,

Ngay = x.NGHD

});

YDatas = new List<YData>();

for (int h = 1; h <= 31; h++)

{

long value = 0;

if (query.Where(x => x.Ngay.Day == h && x.Ngay.Month == DateTime.Now.Month && x.Ngay.Year == DateTime.Now.Year).Select(x => x.Tien).Count() > 0)

{

value = query.Where(x => x.Ngay.Day == h && x.Ngay.Month == DateTime.Now.Month && x.Ngay.Year == DateTime.Now.Year).Select(x => x.Tien).Sum();

}

YData result = new YData(h, value);

YDatas.Add(result);

}

}

p.ColChart.ItemsSource = YDatas;

}

* **Xuất biểu đồ và gửi qua mail:**

private void SaveImageEncoder\_Click(ReportView p)

{

p.Combobox.SelectedIndex = 0;

ColChart(p);

MessageBoxResult d = System.Windows.MessageBox.Show("In báo cáo ?", "THÔNG BÁO", MessageBoxButton.YesNoCancel, MessageBoxImage.Question);

if (d == MessageBoxResult.Yes)

{

try

{

p.SampleChart.Save(Const.\_localLink + @"Resource/Chart/doanh\_thu.jpg");

p.PChart.Save(Const.\_localLink + @"Resource/Chart/loai\_san\_pham.jpg");

p.DChart.Save(Const.\_localLink + @"Resource/Chart/top5\_san\_pham.jpg");

MailMessage message = new MailMessage("clothesmanagement1412@gmail.com", Const.ND.MAIL, "Báo cáo thống kê", "Đây là biểu đồ thống kê bạn vừa xuất. Trân trọng !");

Attachment attachment = new Attachment(Const.\_localLink + @"Resource/Chart/doanh\_thu.jpg");

Attachment attachment1 = new Attachment(Const.\_localLink + @"Resource/Chart/loai\_san\_pham.jpg");

Attachment attachment2 = new Attachment(Const.\_localLink + @"Resource/Chart/top5\_san\_pham.jpg");

message.Attachments.Add(attachment);

message.Attachments.Add(attachment1);

message.Attachments.Add(attachment2);

SmtpClient smtpClient = new SmtpClient("smtp.gmail.com", 587);

smtpClient.EnableSsl = true;

smtpClient.Credentials = new NetworkCredential("clothesmanagement1412@gmail.com", "doanlttq");

smtpClient.Send(message);

MessageBox.Show("Đã gửi báo cáo, vui lòng kiểm tra mail !", "Thông báo");

}

catch

{

MessageBox.Show("Không thể xử lý vì tệp đang được mở trong cửa sổ khác !", "Thông báo", MessageBoxButton.OK, MessageBoxImage.Error); ;

}

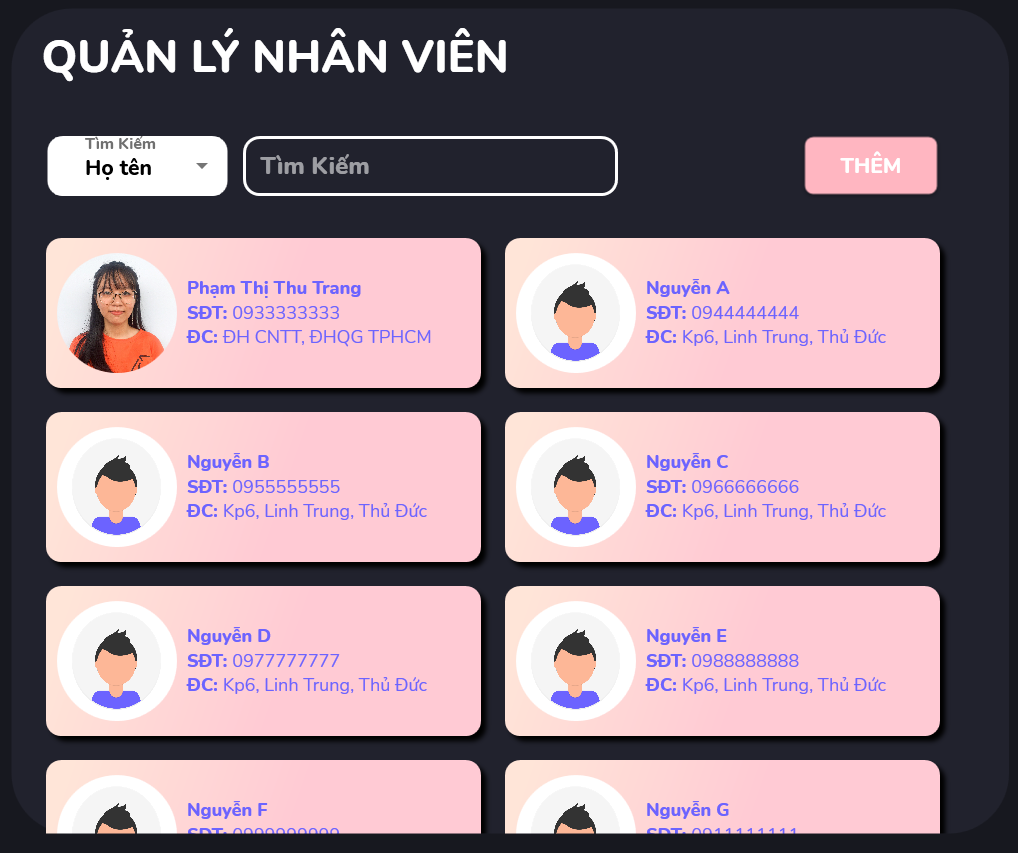
}

}

* + 1. Màn hình danh mục quản lý:

* + - 1. Giao diện:

Chọn thuộc tính để tìm kiếm



Nhấn mở cửa sổ thêm nhân viên

Tìm kiếm theo thuộc tính đã chọn

Chọn trong danh sách để hiển thị cửa sổ thông tin chi tiết

Hình 4.13. Giao diện màn hình danh mục quản lý

* + - 1. Mô tả chức năng:
* Màn hình hiển thị danh sách các quản lý và nhân viên của cửa hàng, không tính bản thân.
* Khi nhấn nút Thêm sẽ xuất hiện form thêm người dùng.
* Khi nhấn vào một nhân viên bất kì trong danh sách sẽ hiện form chi tiết thông tin nhân viên.
* Chọn thuộc tính cần tìm kiếm bằng combobox. Khi gõ ký tự vào thanh tìm kiếm sẽ hiển thị các nhân viên có ký tự trùng khớp với ký tự trong thanh tìm kiếm theo thuộc tính trong combobox.
  + - 1. Kỹ thuật xử lý:
* Source code xử lý:
* **Load combobox và danh sách nhân viên khi khởi chạy màn hình:**

void \_LoadCsCommand(QLNVView parameter)

{

parameter.cbxChon.SelectedIndex = 0;

listND = new ObservableCollection<NGUOIDUNG>(DataProvider.Ins.DB.NGUOIDUNGs.Where(p => p.TTND == true && p.MAND != Const.ND.MAND));

}

* **Chức năng tìm kiếm theo thuộc tính trong combobox:**

void \_SearchCommand(QLNVView paramater)

{

ObservableCollection<NGUOIDUNG> temp = new ObservableCollection<NGUOIDUNG>();

if (paramater.txbSearch.Text != "")

{

switch (paramater.cbxChon.SelectedItem.ToString())

{

case "Mã NV":

{

foreach (NGUOIDUNG s in listND)

{

if (s.MAND.Contains(paramater.txbSearch.Text))

{

temp.Add(s);

}

}

break;

}

case "Họ tên":

{

foreach (NGUOIDUNG s in listND)

{

if (s.TENND.ToLower().Contains(paramater.txbSearch.Text.ToLower()))

{

temp.Add(s);

}

}

break;

}

case "SĐT":

{

foreach (NGUOIDUNG s in listND)

{

if (s.SDT.Contains(paramater.txbSearch.Text))

{

temp.Add(s);

}

}

break;

}

default:

{

foreach (NGUOIDUNG s in listND)

{

if (s.TENND.Contains(paramater.txbSearch.Text))

{

temp.Add(s);

}

}

break;

}

}

paramater.ListViewND.ItemsSource = temp;

}

else

paramater.ListViewND.ItemsSource = listND;

}

* **Khởi chạy cửa sổ thông tin chi tiết nhân viên:**

void \_DetailND(QLNVView paramater)

{

DetailNDView detailNDView=new DetailNDView();

NGUOIDUNG temp= (NGUOIDUNG)paramater.ListViewND.SelectedItem;

detailNDView.MaND.Text = temp.MAND;

detailNDView.TenND.Text = temp.TENND;

detailNDView.SDT.Text = temp.SDT;

detailNDView.GT.Text = temp.GIOITINH;

detailNDView.NS.Text = temp.NGSINH.ToString();

Uri fileUri = new Uri(temp.AVA);

detailNDView.HinhAnh.ImageSource = new BitmapImage(fileUri);

detailNDView.DC.Text = temp.DIACHI;

detailNDView.QTV.Text = temp.QTV == true ? "Quản lý" : "Nhân viên";

detailNDView.ShowDialog();

listND = new ObservableCollection<NGUOIDUNG>(DataProvider.Ins.DB.NGUOIDUNGs.Where(p => p.TTND == true && p.MAND != Const.ND.MAND));

paramater.ListViewND.ItemsSource = listND;

paramater.ListViewND.SelectedItem = null;

}

* **Khởi chạy màn hình thêm nhân viên:**

void \_AddND(QLNVView parameter)

{

AddNDView addNDView = new AddNDView();

addNDView.MaND.Text=rdma();

addNDView.ShowDialog();

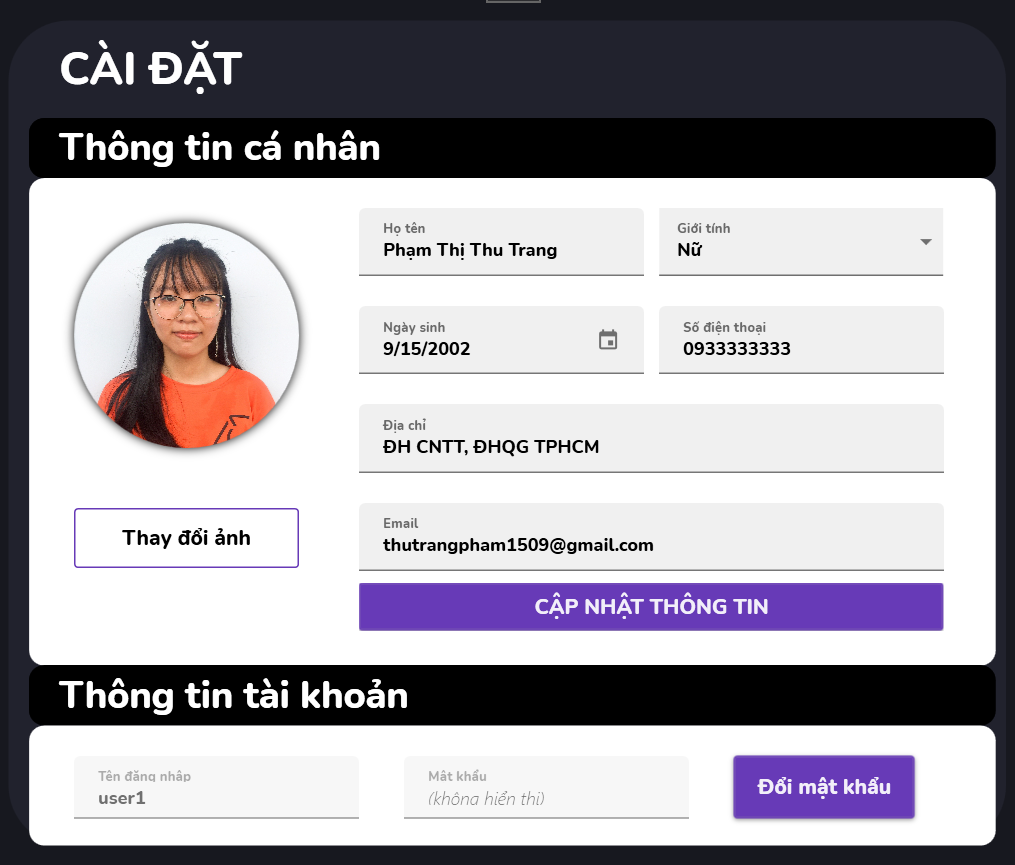
listND = new ObservableCollection<NGUOIDUNG>(DataProvider.Ins.DB.NGUOIDUNGs.Where(p => p.TTND == true && p.MAND != Const.ND.MAND));

parameter.ListViewND.ItemsSource = listND;

parameter.ListViewND.Items.Refresh();

}

* **Tạo ngẫu nhiên mã người dùng khi thêm nhân viên** (tương tự như khi thêm người dùng)
  + 1. Màn hình danh mục cài đặt:
       1. Giao diện:



Nhấn chọn ảnh đại diện

Nhấn khởi chạy cửa sổ đổi mật khẩu

Điền thông tin nếu cần thay đổi

Nhấn để cập nhật thông tin đã thay đổi phía trên

Hình 4.14. Giao diện màn hình danh mục cài đặt

* + - 1. Mô tả chức năng:
* Khi nhấn nút Thay đổi ảnh sẽ hiện ra cửa sổ file để chọn hình ảnh.
* Khi nhấn nút Cập nhật thông tin, ảnh và các thông tin ở các ô phía trên và ảnh đại diện sẽ được lưu lại.
* Khi nhấn nút Đổi mật khẩu sẽ khởi chạy form đổi mật khẩu.
  + - 1. Kỹ thuật xử lý:
* Source code xử lý:
* **Khởi chạy màn hình cài đặt:**

void \_Loadwd(SettingView p)

{

if (LoginViewModel.IsLogin)

{

string a = Const.TenDangNhap;

User = DataProvider.Ins.DB.NGUOIDUNGs.Where(x => x.USERNAME == a).FirstOrDefault();

Ava = User.AVA;

Name = User.TENND;

DoB = User.NGSINH.ToString();

DiaChi = User.DIACHI;

GioiTinh = (User.GIOITINH == "Nam") ? 0 : 1;

SDT = User.SDT;

TenTK = User.USERNAME;

Mail = User.MAIL;

}

}

* **Nhấn nút Đổi mật khẩu:**

void \_ChangePass()

{

ChangePassword change = new ChangePassword();

change.ShowDialog();

}

* **Nhấn nút Thay đổi ảnh:**

void \_AddImage(ImageBrush p)

{

OpenFileDialog open = new OpenFileDialog();

open.Filter = "Image Files(\*.jpg; \*.png)|\*.jpg; \*.png";

if (open.ShowDialog() == true)

{

Ava = open.FileName;

}

p.ImageSource = new BitmapImage(new Uri(Ava));

}

* **Nhấn nút cập nhật thông tin:**

void \_UdpateInfo(SettingView p)

{

foreach (NGUOIDUNG temp2 in DataProvider.Ins.DB.NGUOIDUNGs)

{

if (temp2.MAIL == p.Mail.Text&&p.Mail.Text!=Const.ND.MAIL)

{

MessageBox.Show("Email này đã được sử dụng !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

}

string match = @"\w+([-+.']\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*";

Regex reg = new Regex(match);

if (!reg.IsMatch(p.Mail.Text))

{

MessageBox.Show("Email không hợp lệ !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

var temp = DataProvider.Ins.DB.NGUOIDUNGs.Where(pa => pa.USERNAME == TenTK).FirstOrDefault();

temp.TENND = p.NameBox.Text;

temp.SDT = p.SDTBox.Text;

temp.DIACHI = p.AddressBox.Text;

temp.GIOITINH = p.GTBox.Text;

temp.NGSINH = (DateTime)p.DateBox.SelectedDate;

temp.MAIL = p.Mail.Text;

string rd = StringGenerator();

if (User.AVA != Ava)

temp.AVA = "/Resource/Ava/" + rd + (Ava.Contains(".jpg") ? ".jpg" : ".png").ToString();

DataProvider.Ins.DB.SaveChanges();

try

{

if (User.AVA != Ava)

File.Copy(Ava, Const.\_localLink + @"Resource/Ava/" + rd + (Ava.Contains(".jpg") ? ".jpg" : ".png").ToString(), true);

}

catch { }

MessageBox.Show("Cập nhật thành công!", "Thông báo");

}

* **Tạo chuỗi ngẫu nhiên để đặt tên file:**

static string StringGenerator()

{

Random rd = new Random();

int length = rd.Next(5, 20);

StringBuilder str\_build = new StringBuilder();

Random random = new Random();

char letter;

for (int i = 0; i < length; i++)

{

double flt = random.NextDouble();

int shift = Convert.ToInt32(Math.Floor(25 \* flt));

letter = Convert.ToChar(shift + 65);

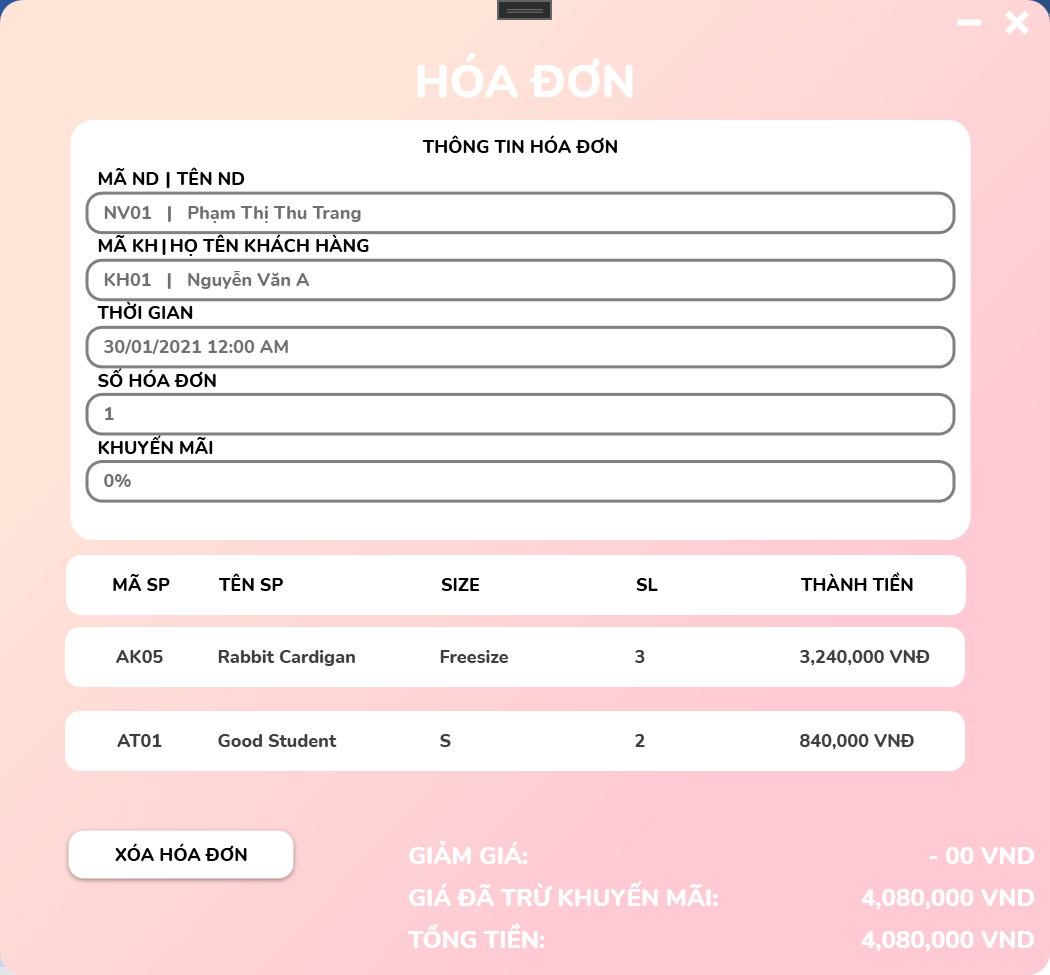
str\_build.Append(letter);

}

return str\_build.ToString();

}

* 1. Quản lý đơn hàng:
     1. Màn hình chi tiết đơn hàng:
        1. Giao diện:



Nhấn để xóa hóa đơn

Hình 4.15. Giao diện màn hình chi tiết hóa đơn

* + - 1. Mô tả chức năng:
* Nhấn nút xóa hóa đơn để xóa hóa đơn đang chọn hiện tại.
  + - 1. Kỹ thuật xử lý:
* Hiệu chỉnh các thuộc tính trong bảng Properties:

Graphical user interface, text

Description automatically generated

*Tạo màu gradient cho background*



*Format kiểu tiền khi hiện trên màn hình*

* Source code xử lý:
* **Xóa đơn hàng đang chọn hiện tại:**

void \_DeleteOrder(DetailOrder parameter)

{

MessageBoxResult h = System.Windows.MessageBox.Show("Bạn muốn xóa hóa đơn này?", "THÔNG BÁO", MessageBoxButton.YesNoCancel,MessageBoxImage.Question);

if (h == MessageBoxResult.Yes)

{

foreach (HOADON temp in DataProvider.Ins.DB.HOADONs)

{

if (temp.SOHD == int.Parse(parameter.SoHD.Text))

{

foreach (CTHD temp1 in temp.CTHDs)

{

foreach (SANPHAM temp2 in DataProvider.Ins.DB.SANPHAMs)

{

if (temp1.MASP == temp2.MASP)

{

if(temp2.SL==-1)

temp2.SL += temp1.SL+1;

else if(temp2.SL>=0)

temp2.SL += temp1.SL;

}

}

}

DataProvider.Ins.DB.HOADONs.Remove(temp);

}

}

DataProvider.Ins.DB.SaveChanges();

}

}

* + 1. Màn hình tạo đơn hàng:
       1. Giao diện:

Graphical user interface, application

Description automatically generated

Nhấn để thanh toán hóa đơn

Xóa sản phẩm đã chọn trong danh sách bên dưới

Thêm sản phẩm vào hóa đơn

Nhập số lượng

Chọn sản phẩm

Chọn khách hàng trong danh sách

Hình 4.16. Giao diện màn hình tạo hóa đơn

* + - 1. Mô tả chức năng:
* Nhấn vào ô khách hàng để hiển thị danh sách khách hàng và chọn khách hàng cho hóa đơn đang tạo.
* Nhấn vào ô sản phẩm để hiển thị danh sách sản phẩm và chọn sản phẩm cho hóa đơn đang tạo.
* Nhấn vào ô số lượng để nhập số lượng sản phẩm khách hàng muốn mua.
* Nhấn nút thêm để thêm sản phẩm vừa chọn vào hóa đơn và danh sách hiển thị sản phẩm khách hàng đã mua.
* Nhấn nút xóa để xóa sản phẩm vừa chọn trong danh sách hiển thị sản phẩm đã mua ra khỏi hóa đơn và danh sách hiển thị sản phẩm khách hàng đã mua.
* Nhấn nút thanh toán để lưu hóa đơn hiện tại vào database và xuất hiện thông báo in hóa đơn.
  + - 1. Kỹ thuật xử lý:
* Source code xử lý:
* **Tải dữ liệu mặc định cho hóa đơn:**

void \_Loadwd(AddOrderView paramater)

{

LKH= DataProvider.Ins.DB.KHACHHANGs.ToList();

LSP = DataProvider.Ins.DB.SANPHAMs.Where(p=>p.SL>=0).ToList();

paramater.KH.ItemsSource = LKH;

paramater.SP.ItemsSource = LSP;

paramater.MaND.Text = Const.ND.MAND;

paramater.TenND.Text = Const.ND.TENND;

paramater.Ngay.Text = DateTime.Now.ToString("dd/MM/yyyy hh:mm tt");

paramater.TT.Text = String.Format("{0:0,0}", tongtien) + " VND";

paramater.TT1.Text = String.Format("{0:0,0}", tongtien) + " VND";

paramater.GG.Text = "- " + String.Format("{0:0,0}", tienkm) + " VND";

km = 0;

paramater.khuyenmai.Text = km.ToString()+"%";

}

* **Tải thông tin khuyến mãi của khách hàng:**

void \_chooseKH(AddOrderView parameter)

{

KHACHHANG temp = (KHACHHANG)parameter.KH.SelectedItem;

if(temp!=null)

{

int doanhso = 0;

foreach (HOADON a in DataProvider.Ins.DB.HOADONs)

{

if (a.MAKH == temp.MAKH)

doanhso += a.TRIGIA;

}

km = 0;

if (doanhso > 2000000 && doanhso <= 5000000)

km = 2;

else if (doanhso > 5000000 && doanhso <= 10000000)

km = 5;

else if (doanhso > 10000000)

km = 10;

parameter.khuyenmai.Text = km.ToString() + "%";

}

else

{

km = 0;

parameter.khuyenmai.Text = "0%";

}

}

* **Thêm sản phẩm vào hóa đơn:**

void \_AddSP(AddOrderView paramater)

{

if(paramater.SP.SelectedItem==null)

{

System.Windows.MessageBox.Show("Bạn chưa chọn sản phẩm để thêm !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if(paramater.SoHD.Text=="")

{

System.Windows.MessageBox.Show("Bạn chưa nhập số hóa đơn !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (paramater.SL.Text == "")

{

System.Windows.MessageBox.Show("Bạn chưa nhập số lượng sản phẩm !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

try {

int so = int.Parse(paramater.SL.Text);

}

catch {

MessageBox.Show("Số lượng không hợp lệ !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return ;

}

if (int.Parse(paramater.SL.Text) < 0)

{

MessageBox.Show("Số lượng sản phẩm không hợp lệ !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (paramater.KH.SelectedItem == null)

{

System.Windows.MessageBox.Show("Bạn chưa chọn khách hàng !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

SANPHAM a = (SANPHAM)paramater.SP.SelectedItem;

if (a.SL >= int.Parse(paramater.SL.Text))

{

foreach (HienThi temp in LHT)

{

if (temp.MaSp == a.MASP)

{

temp.SL += int.Parse(paramater.SL.Text);

temp.Tong = temp.SL \* a.GIA;

foreach (CTHD temp1 in LCTHD)

{

if (temp1.MASP == a.MASP)

{

temp1.SL += int.Parse(paramater.SL.Text); ;

}

}

tongtien += int.Parse(paramater.SL.Text) \* a.GIA\*(100-km)/100;

tienkm+= int.Parse(paramater.SL.Text) \* a.GIA \* km / 100;

paramater.TT.Text = String.Format("{0:0,0}", tongtien) + " VND";

paramater.TT1.Text = String.Format("{0:0,0}", tongtien) + " VND";

paramater.GG.Text="- "+ String.Format("{0:0,0}", tienkm) + " VND";

paramater.ListViewSP.ItemsSource = LHT;

paramater.ListViewSP.Items.Refresh();

paramater.SP.SelectedItem = null;

paramater.SL.Text = "";

return;

}

}

HienThi b = new HienThi(a.MASP, a.TENSP, a.SIZE, int.Parse(paramater.SL.Text),a.GIA, int.Parse(paramater.SL.Text) \* a.GIA);

CTHD cthd = new CTHD()

{

MASP = a.MASP,

SL = int.Parse(paramater.SL.Text),

SANPHAM = a,

SOHD = int.Parse(paramater.SoHD.Text),

};

tongtien += int.Parse(paramater.SL.Text) \* a.GIA\*(100-km)/100;

tienkm += int.Parse(paramater.SL.Text) \* a.GIA \* km / 100;

paramater.TT.Text = String.Format("{0:0,0}", tongtien) + " VND";

paramater.TT1.Text = String.Format("{0:0,0}", tongtien) + " VND";

paramater.GG.Text = "- " + String.Format("{0:0,0}", tienkm) + " VND";

LCTHD.Add(cthd);

LHT.Add(b);

paramater.ListViewSP.ItemsSource = LHT;

paramater.ListViewSP.Items.Refresh();

paramater.SP.SelectedItem = null;

paramater.SL.Text = "";

}

else

System.Windows.MessageBox.Show("Sản phẩm tồn kho không đủ cung cấp !", "THÔNG BÁO");

}

* **Xóa sản phẩm trong hóa đơn:**

void \_DeleteSP(AddOrderView paramater)

{

if(paramater.ListViewSP.SelectedItem==null)

{

System.Windows.MessageBox.Show("Bạn chưa chọn sản phẩm để xóa !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

MessageBoxResult h = System.Windows.MessageBox.Show(" Bạn có chắc muốn xóa sản phẩm.", "THÔNG BÁO", MessageBoxButton.YesNoCancel,MessageBoxImage.Question);

if (h == MessageBoxResult.Yes)

{

HienThi a = (HienThi)paramater.ListViewSP.SelectedItem;

tongtien -= a.Tong\*(100-km)/100;

tienkm-= a.Tong \* km / 100;

paramater.TT.Text = String.Format("{0:0,0}", tongtien) + " VND";

paramater.TT1.Text = String.Format("{0:0,0}", tongtien) + " VND";

paramater.GG.Text = "- " + String.Format("{0:0,0}", tienkm) + " VND";

LHT.Remove(a);

foreach (CTHD b in LCTHD)

{

if (b.MASP == a.MaSp && b.SL == a.SL)

{

LCTHD.Remove(b);

break;

}

}

paramater.SP.ItemsSource = LSP;

paramater.SP.Items.Refresh();

paramater.ListViewSP.Items.Refresh();

}

else

return;

}

* **Lưu hóa đơn vào database:**

void \_SaveHD(AddOrderView paramater)

{

DataProvider.Ins.DB.SaveChangesAsync();

if(paramater.KH.SelectedItem==null||paramater.ListViewSP.Items.Count==0)

{

System.Windows.MessageBox.Show("Thông tin hóa đơn chưa đầy đủ !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

MessageBoxResult h = System.Windows.MessageBox.Show(" Bạn muốn thanh toán ?", "THÔNG BÁO", MessageBoxButton.YesNoCancel,MessageBoxImage.Question);

if (h == MessageBoxResult.Yes)

{

KHACHHANG a = (KHACHHANG)paramater.KH.SelectedItem;

int tonggia = 0;

foreach (HienThi b in LHT)

{

tonggia += b.Tong;

}

double tien = (double)(1-(double)km/100)\*tonggia;

HOADON temp = new HOADON()

{

SOHD = int.Parse(paramater.SoHD.Text),

MAKH = a.MAKH,

MAND = Const.ND.MAND,

NGHD = DateTime.Now,

CTHDs = new ObservableCollection<CTHD>(LCTHD),

TRIGIA = (int)tien,

KHACHHANG = a,

NGUOIDUNG = Const.ND,

KHUYENMAI = km

};

foreach(CTHD s in LCTHD)

{

foreach (SANPHAM x in DataProvider.Ins.DB.SANPHAMs)

{

if (x.MASP == s.SANPHAM.MASP)

{

x.SL -= s.SL;

}

}

}

DataProvider.Ins.DB.HOADONs.Add(temp);

DataProvider.Ins.DB.SaveChanges();

MessageBoxResult d = System.Windows.MessageBox.Show(" Bạn có muốn in hóa đơn ?", "THÔNG BÁO", MessageBoxButton.YesNoCancel,MessageBoxImage.Question);

if (d == MessageBoxResult.Yes)

{

print(paramater);

}

tongtien = 0;

km = 0;

tienkm = 0;

LSPSelected.Clear();

paramater.KH.SelectedItem = null;

LHT.Clear();

LCTHD.Clear();

paramater.ListViewSP.ItemsSource = LHT;

paramater.TT.Text = tongtien.ToString();

paramater.GG.Text = "- " + tienkm.ToString();

paramater.TT1.Text = tongtien.ToString();

paramater.SoHD.Text=rdma().ToString();

MessageBox.Show("Thanh toán hóa đơn thành công !", "THÔNG BÁO");

}

else

return;

}

* **In hóa đơn:**

void print(AddOrderView parameter)

{

KHACHHANG temp = (KHACHHANG)parameter.KH.SelectedItem;

PrintOrderView printOrderView = new PrintOrderView();

printOrderView.Height = 300 + 35 \* LHT.Count;

printOrderView.TenKH.Text = temp.HOTEN;

printOrderView.dc.Text = temp.DCHI;

printOrderView.sdt.Text = temp.SDT;

printOrderView.ngay.Text = DateTime.Now.ToString("dd/MM/yyyy hh:mm tt");

printOrderView.sohd.Text = parameter.SoHD.Text;

printOrderView.ListSP.ItemsSource = LHT;

printOrderView.tt.Text = parameter.TT.Text;

printOrderView.gg.Text = parameter.GG.Text;

printOrderView.tt1.Text = parameter.TT1.Text;

try

{

PrintDialog printDialog = new PrintDialog();

if (printDialog.ShowDialog() == true)

{

printDialog.PrintVisual(printOrderView.PrintView, "BILL");

}

}

finally

{

}

MessageBox.Show("In Hóa đơn thành công !","THÔNG BÁO");

}

* 1. Quản lý sản phẩm:
     1. Màn hình chi tiết sản phẩm:
        1. Giao diện:

Graphical user interface

Description automatically generated

Nhấn xem mô tả sản phẩm. Quản trị viên có thể sửa mô tả

Quản trị viên có thể đổi tên sản phẩm

Nhấn để xóa sản phẩm khỏi danh sách hiển thị

Nhấn để cập nhật thông tin

Đăng nhập dưới quyền nhân viên sẽ không có 2 nút này

Hình 4.17. Giao diện màn hình chi tiết sản phẩm

* + - 1. Mô tả chức năng:
* Nhấn nút cập nhật để cập nhật lại thông tin của sản phẩm.
* Nhấn nút xóa để xóa sản phẩm ra khỏi danh sách hiển thị sản phẩm.
* Chỉ quản trị viên mới có thể cập nhật và xóa sản phẩm.
  + - 1. Kỹ thuật xử lý:
* Hiệu chỉnh các thuộc tính trong bảng Properties:

: cho chữ khi nhập tự động xuống hàng khi không còn đủ chỗ hiển thị.

: cho phép nhấn Enter để xuống dòng trong khi nhập.

: hiện thanh cuộn nội dụng.

* Source code xử lý:
* **Kiểm tra quyền người dùng:**

void \_Loadwd(DetailProduct parmater)

{

if(Const.Admin)

{

parmater.TenSP.IsEnabled = true;

parmater.Mota.IsEnabled = true;

parmater.btncapnhat.Visibility = Visibility.Visible;

parmater.btnxoa.Visibility = Visibility.Visible;

}

else

{

parmater.TenSP.IsEnabled = false;

parmater.Mota.IsEnabled = false;

parmater.Mota.Height = 200;

}

}

* **Cập nhật thông tin sản phẩm:**

void \_UpdateProduct(DetailProduct p)

{

MessageBoxResult h = System.Windows.MessageBox.Show("Bạn muốn cập nhật sản phẩm ?", "THÔNG BÁO", MessageBoxButton.YesNoCancel,MessageBoxImage.Question);

if (h == MessageBoxResult.Yes)

{

if (string.IsNullOrEmpty(p.TenSP.Text) || string.IsNullOrEmpty(p.Mota.Text) || string.IsNullOrEmpty(p.Mota.Text))

{

MessageBox.Show("Thông tin chưa đầy đủ !", "THÔNG BÁO");

}

else

{

foreach (SANPHAM a in DataProvider.Ins.DB.SANPHAMs.Where(pa => (pa.TENSP == TenSP1&&pa.SL>=0)))

{

a.TENSP = p.TenSP.Text;

a.MOTA = p.Mota.Text;

a.MOTA = p.Mota.Text;

}

DataProvider.Ins.DB.SaveChanges();

MessageBox.Show("Cập nhật sản phẩm thành công !", "THÔNG BÁO");

}

}

}

}

* **Xóa sản phẩm ra khỏi danh sách hiển thị:**

void \_DeleteProduct(DetailProduct parameter)

{

MessageBoxResult h = System.Windows.MessageBox.Show("Bạn muốn xóa sản phẩm ?", "THÔNG BÁO", MessageBoxButton.YesNoCancel,MessageBoxImage.Question);

if (h == MessageBoxResult.Yes)

{

foreach (SANPHAM a in DataProvider.Ins.DB.SANPHAMs.Where(pa => (pa.TENSP == TenSP1 && pa.SL >= 0)))

{

a.SL = -1;

}

DataProvider.Ins.DB.SaveChanges();

MessageBox.Show("Xóa sản phẩm thành công !", "THÔNG BÁO");

}

}

* + 1. Màn hình thêm sản phẩm mới:
       1. Giao diện:

Graphical user interface, application

Description automatically generated

Nhấn để thêm sản phẩm

Nhấn để chọn hình ảnh cho sản phẩm

Hình 4.18. Giao diện màn hình thêm sản phẩm mới

* + - 1. Mô tả chức năng:
* Nhấn vào hình ảnh để chọn hình ảnh cho sản phẩm.
* Nhấn nút Thêm sản phẩm để thêm sản phẩm vào database. Nếu không nhập đầy đủ thông tin hoặc sai kiểu dữ liệu sẽ báo lỗi.
  + - 1. Kỹ thuật xử lý:
* Hiệu chỉnh các thuộc tính trong bảng Properties:

: làm viền đổi màu LightPink khi di chuyển chuột vào ô nhập.

: cho phép nhấn Enter để xuống dòng trong khi nhập.

: tự động xuống hàng khi không đủ chỗ hiển thị.

* Source code xử lý:
* **Mở và tải ảnh sản phẩm:**

void \_AddImage(Image img)

{

OpenFileDialog open = new OpenFileDialog();

open.Filter = "Image Files(\*.jpg; \*.png)|\*.jpg; \*.png";

if (open.ShowDialog() == true)

{

linkimage = open.FileName;

};

Uri fileUri = new Uri(linkimage);

img.Source = new BitmapImage(fileUri);

}

* **Thêm sản phẩm vào database và danh sách hiển thị:**

void \_AddProduct(AddProductView paramater)

{

if(string.IsNullOrEmpty(paramater.MaSp.Text)|| string.IsNullOrEmpty(paramater.TenSp.Text) || string.IsNullOrEmpty(paramater.LoaiSp.Text) || string.IsNullOrEmpty(paramater.GiaSp.Text) || string.IsNullOrEmpty(paramater.SizeSp.Text) || string.IsNullOrEmpty(paramater.SlSp.Text) || linkimage== "/Resource/Image/add.png")

{

MessageBox.Show("Bạn chưa nhập đủ thông tin.", "Thông Báo", MessageBoxButton.OK, MessageBoxImage.Error);

}

else

{

MessageBoxResult h = System.Windows.MessageBox.Show("Bạn muốn thêm sản phẩm mới ?", "THÔNG BÁO", MessageBoxButton.YesNoCancel,MessageBoxImage.Question);

if (h == MessageBoxResult.Yes)

{

if (DataProvider.Ins.DB.SANPHAMs.Where(p => p.MASP == paramater.MaSp.Text).Count() > 0)

{

MessageBox.Show("Mã sản phẩm đã tồn tại.", "Thông Báo");

}

else

{

SANPHAM a = new SANPHAM();

a.MASP = paramater.MaSp.Text;

a.TENSP = paramater.TenSp.Text;

a.GIA = int.Parse(paramater.GiaSp.Text);

a.LOAISP = paramater.LoaiSp.Text;

a.SL = int.Parse(paramater.SlSp.Text);

a.SIZE = paramater.SizeSp.Text;

a.MOTA = paramater.MotaSp.Text;

a.HINHSP = "/Resource/ImgProduct/" + "product\_" + paramater.MaSp.Text + ((linkimage.Contains(".jpg")) ? ".jpg" : ".png").ToString();

DataProvider.Ins.DB.SANPHAMs.Add(a);

DataProvider.Ins.DB.SaveChanges();

try

{

File.Copy(linkimage, \_localLink + @"Resource\ImgProduct\" + "product\_" + paramater.MaSp.Text + ((linkimage.Contains(".jpg")) ? ".jpg" : ".png").ToString(), true);

}

catch { }

MessageBox.Show("Thêm sản phẩm mới thành công !", "THÔNG BÁO");

paramater.MaSp.Text = rdma();

paramater.TenSp.Clear();

paramater.LoaiSp.SelectedItem = null;

paramater.GiaSp.Clear();

paramater.SlSp.Clear();

paramater.SizeSp.SelectedItem = null;

Uri fileUri = new Uri(Const.\_localLink+ "/Resource/Image/add.png");

paramater.HinhAnh.Source=new BitmapImage(fileUri);

paramater.MotaSp.Clear();

}

}

* 1. Quản lý khách hàng:
     1. Màn hình chi tiết khách hàng:
        1. Giao diện:

Graphical user interface

Description automatically generated with medium confidence

Nhấn để cập nhật thông tin

Hình 4.19. Giao diện màn hình chi tiết khách hàng

* + - 1. Mô tả chức năng:
* Nhấn nút cập nhật để cập nhật thông tin của khách hàng (họ tên, số điện thoại, giới tính, địa chỉ).
  + - 1. Kỹ thuật xử lý:
* Source code xử lý:
* **Chức năng cập nhật thông tin khách hàng:**

void \_Update(DetailCustomerView p)

{

MessageBoxResult h = System.Windows.MessageBox.Show(" Bạn muốn cập nhật thông tin ?", "THÔNG BÁO", MessageBoxButton.YesNoCancel,MessageBoxImage.Question);

if (h == MessageBoxResult.Yes)

{

if (string.IsNullOrEmpty(p.TenKH.Text) || string.IsNullOrEmpty(p.SDT.Text) || string.IsNullOrEmpty(p.GT.Text) || string.IsNullOrEmpty(p.DC.Text))

{

MessageBox.Show("Thông tin chưa đầy đủ !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

}

else

{

var temp = DataProvider.Ins.DB.KHACHHANGs.Where(pa => pa.MAKH == MaKH);

foreach (KHACHHANG a in temp)

{

a.HOTEN = p.TenKH.Text;

a.SDT = p.SDT.Text;

a.GIOITINH = p.GT.Text;

a.DCHI = p.DC.Text;

}

DataProvider.Ins.DB.SaveChanges();

MessageBox.Show("Cập nhật thông tin thành công !", "THÔNG BÁO");

}

}

}

* + 1. Màn hình thêm khách hàng:
       1. Giao diện:

Graphical user interface, text, application

Description automatically generated

Nhấn để thêm khách hàng

Hình 4.20. Giao diện màn hình thêm khách hàng

* + - 1. Mô tả chức năng:
* Cho phép nhập thông tin của khách hàng.
* Nhấn nút thêm khách hàng để thêm khách hàng vào database và danh sách hiển thị khách hàng.
  + - 1. Kỹ thuật xử lý:
* Source code xử lý
* **Chức năng thêm khách hàng:**

void \_AddCsCommand(AddCustomerView paramater)

{ if(paramater.TenKH.Text==""||paramater.SDT.Text==""||paramater.GT.SelectedItem==null||paramater.DC.Text=="")

{

MessageBox.Show("Bạn chưa nhập đủ thông tin !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

MessageBoxResult h = System.Windows.MessageBox.Show(" Bạn muốn thêm khách hàng ?", "THÔNG BÁO", MessageBoxButton.YesNoCancel);

if (h == MessageBoxResult.Yes) {

if (string.IsNullOrEmpty(paramater.MaKH.Text) || string.IsNullOrEmpty(paramater.TenKH.Text) || string.IsNullOrEmpty(paramater.SDT.Text) || string.IsNullOrEmpty(paramater.GT.Text) || string.IsNullOrEmpty(paramater.DC.Text))

{

MessageBox.Show("Thông tin chưa đầy đủ !", "THÔNG BÁO");

}

else

{

if (DataProvider.Ins.DB.KHACHHANGs.Where(p => p.MAKH == paramater.MaKH.Text).Count() > 0) {

MessageBox.Show("Mã khách hàng đã tồn tại !", "THÔNG BÁO");

}

else {

KHACHHANG temp = new KHACHHANG();

temp.MAKH = paramater.MaKH.Text.ToString();

temp.HOTEN = paramater.TenKH.Text.ToString();

temp.SDT = paramater.SDT.Text.ToString();

temp.GIOITINH = paramater.GT.Text.ToString();

temp.DCHI = paramater.DC.Text.ToString();

DataProvider.Ins.DB.KHACHHANGs.Add(temp);

DataProvider.Ins.DB.SaveChanges();

MessageBox.Show("Thêm khách hàng thành công.", "THÔNG BÁO");

paramater.MaKH.Text = rdma();

paramater.TenKH.Clear();

paramater.SDT.Clear();

paramater.GT.SelectedItem=null;

paramater.GT.Items.Refresh();

paramater.DC.Clear();

}

}

}

}

* 1. Quản lý nhân viên:
     1. Màn hình chi tiết nhân viên:
        1. Giao diện:

Graphical user interface, application

Description automatically generated

Nhấn để xóa nhân viên

Nhấn để đặt mật khẩu về giá trị mặc định

Nhấn để cập nhật quyền

Hình 4.21. Giao diện màn hình chi tiết nhân viên

* + - 1. Mô tả chức năng:
* Nhấn nút cập nhật để cập nhật lại quyền của nhân viên.
* Nhấn nút xóa để xóa nhân viên ra khỏi danh sách hiển thị.
* Nhấn nút đặt lại mật khẩu để khởi tạo mặc định mật khẩu của tài khoản nhân viên (mật khẩu mặc định là 123456).
  + - 1. Kỹ thuật xử lý:
* Source code xử lý:
* **Chức năng cập nhật:**

void \_UpdateNDCommand(DetailNDView p)

{

MessageBoxResult h = System.Windows.MessageBox.Show("Bạn muốn cập nhật thông tin ?", "THÔNG BÁO", MessageBoxButton.YesNoCancel,MessageBoxImage.Question);

if (h == MessageBoxResult.Yes)

{

foreach (NGUOIDUNG a in DataProvider.Ins.DB.NGUOIDUNGs.Where(pa => pa.TTND == true && pa.MAND != Const.ND.MAND))

{

if (a.MAND == p.MaND.Text)

{

if (p.QTV.Text == "Quản lý")

a.QTV = true;

else

a.QTV = false;

}

}

DataProvider.Ins.DB.SaveChanges();

MessageBox.Show("Cập nhật thành công !", "THÔNG BÁO");

}

}

* **Chức năng xóa:**

void \_DeleteNDCommand(DetailNDView p)

{

MessageBoxResult h = System.Windows.MessageBox.Show("Bạn muốn xóa người dùng này ?", "THÔNG BÁO", MessageBoxButton.YesNoCancel,MessageBoxImage.Question);

if (h == MessageBoxResult.Yes)

{

foreach (NGUOIDUNG a in DataProvider.Ins.DB.NGUOIDUNGs.Where(pa => pa.TTND == true && pa.MAND != Const.ND.MAND))

{

if (a.MAND == p.MaND.Text)

{

a.TTND = false;

}

}

DataProvider.Ins.DB.SaveChanges();

MessageBox.Show("Xóa người dùng thành công !", "THÔNG BÁO");

}

}

* **Chức năng đặt lại mật khẩu:**

void \_ResetPass(DetailNDView p)

{

MessageBoxResult h = System.Windows.MessageBox.Show("Bạn muốn đặt lại mật khẩu ?", "THÔNG BÁO", MessageBoxButton.YesNoCancel, MessageBoxImage.Question);

if (h == MessageBoxResult.Yes)

{

foreach(NGUOIDUNG temp in DataProvider.Ins.DB.NGUOIDUNGs)

{

if (temp.MAND == p.MaND.Text)

temp.PASS = LoginViewModel.MD5Hash(LoginViewModel.Base64Encode("123456"));

}

DataProvider.Ins.DB.SaveChanges();

MessageBox.Show("Đặt lại mật khẩu thành công !", "THÔNG BÁO");

}

}

* + 1. Màn hình thêm nhân viên:
       1. Giao diện:

Graphical user interface, application

Description automatically generated

Nhấn để thêm nhân viên

Nhấn chọn ảnh đại diện

Hình 4.22. Giao diện màn hình thêm nhân viên

* + - 1. Mô tả chức năng:
* Nhấn nút thêm để thêm nhân viên vào database và danh sách hiển thị nhân viên.
* Nhấn nút thêm avatar để tải ảnh lên.
  + - 1. Kỹ thuật xử lý:
* Source code xử lý:
* **Chức năng tải ảnh lên:**

void \_AddImage(ImageBrush img)

{

OpenFileDialog open = new OpenFileDialog();

open.Filter = "Image Files(\*.jpg; \*.png)|\*.jpg; \*.png";

if (open.ShowDialog()== true)

{

if(open.FileName!="")

linkaddimage = open.FileName;

};

Uri fileUri = new Uri(linkaddimage);

img.ImageSource = new BitmapImage(fileUri);

}

* **Chức năng thêm nhân viên:**

void \_AddND(AddNDView addNDView)

{

MessageBoxResult h = System.Windows.MessageBox.Show("Bạn muốn thêm người dùng ?", "THÔNG BÁO", MessageBoxButton.YesNoCancel,MessageBoxImage.Question);

if (h == MessageBoxResult.Yes)

{

if (String.IsNullOrEmpty(addNDView.MaND.Text) || String.IsNullOrEmpty(addNDView.TenND.Text) || String.IsNullOrEmpty(addNDView.SDT.Text) || String.IsNullOrEmpty(addNDView.GT.Text) || String.IsNullOrEmpty(addNDView.QTV.Text) || addNDView.NS.SelectedDate == null)

{

MessageBox.Show("Bạn chưa nhập đầy đủ thông tin !", "THÔNG BÁO");

return;

}

NGUOIDUNG temp = new NGUOIDUNG();

foreach (NGUOIDUNG a in DataProvider.Ins.DB.NGUOIDUNGs.Where(p=>p.TTND==true))

{

if (addNDView.MaND.Text == a.MAND)

{

MessageBox.Show("Mã ND đã tồn tại !", "THÔNG BÁO");

return;

}

}

foreach (NGUOIDUNG temp5 in DataProvider.Ins.DB.NGUOIDUNGs)

{

if (temp5.MAIL == addNDView.Mail.Text)

{

MessageBox.Show("Email này đã được sử dụng !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

}

string match = @"\w+([-+.']\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*";

Regex reg = new Regex(match);

if (!reg.IsMatch(addNDView.Mail.Text))

{

MessageBox.Show("Email không hợp lệ !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

string match1 = @"^((09(\d){8})|(086(\d){7})|(088(\d){7})|(089(\d){7})|(01(\d){9}))$";

Regex reg1 = new Regex(match1);

if (!reg1.IsMatch(addNDView.SDT.Text))

{

MessageBox.Show("Số điện thoại không hợp lệ !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

temp.MAND = addNDView.MaND.Text;

temp.TENND = addNDView.TenND.Text;

temp.SDT = addNDView.SDT.Text;

temp.DIACHI = addNDView.DC.Text;

temp.GIOITINH = addNDView.GT.Text;

temp.MAIL=addNDView.Mail.Text;

temp.NGSINH = (DateTime)addNDView.NS.SelectedDate;

if (addNDView.QTV.Text == "Quản lý")

temp.QTV = true;

else

temp.QTV = false;

temp.TTND = true;

temp.USERNAME = addNDView.MaND.Text;

temp.PASS = LoginViewModel.MD5Hash(LoginViewModel.Base64Encode(addNDView.MaND.Text));

if (linkaddimage == "/Resource/Image/addava.png")

temp.AVA = "/Resource/Image/addava.png";

else

temp.AVA = "/Resource/Ava/" + addNDView.MaND.Text + ((linkaddimage.Contains(".jpg")) ? ".jpg" : ".png").ToString();

DataProvider.Ins.DB.NGUOIDUNGs.Add(temp);

try

{

File.Copy(linkaddimage, Const.\_localLink + @"Resource\Ava\" + temp.MAND + ((linkaddimage.Contains(".jpg")) ? ".jpg" : ".png").ToString(), true);

}

catch { }

DataProvider.Ins.DB.SaveChanges();

MessageBox.Show("Thêm người dùng thành công !", "THÔNG BÁO");

addNDView.MaND.Text = rdma();

addNDView.TenND.Clear();

addNDView.GT.SelectedItem = null;

addNDView.GT.Items.Refresh();

addNDView.QTV.SelectedItem = null;

addNDView.QTV.Items.Refresh();

addNDView.NS.SelectedDate = null;

addNDView.SDT.Clear();

addNDView.DC.Clear();

linkaddimage = Const.\_localLink + "/Resource/Image/addava.png";

Uri fileUri = new Uri(linkaddimage);

addNDView.HinhAnh1.ImageSource = new BitmapImage(fileUri);

}

* 1. Quản lý nhập hàng:

4.10.1. Màn hình chi tiết phiếu nhập:

4.10.1.1. Giao diện:

Table

Description automatically generated

Nhấn để xóa phiếu nhập

Hình 4.23. Giao diện màn hình chi tiết phiếu nhập

4.10.1.2. Mô tả chức năng:

* Nhấn nút xóa phiếu nhập để xóa phiếu nhập ra khỏi database và danh sách hiển thị phiếu nhập.

4.10.1.3. Kỹ thuật xử lý:

* Source code xử lý:
* Chức năng xóa phiếu nhập:

void \_Delete(DetailImport parameter)

{

MessageBoxResult h = System.Windows.MessageBox.Show("Bạn muốn xóa phiếu nhập này?", "THÔNG BÁO", MessageBoxButton.YesNoCancel, MessageBoxImage.Question);

if (h == MessageBoxResult.Yes)

{

foreach (PHIEUNHAP temp in DataProvider.Ins.DB.PHIEUNHAPs)

{

if (temp.MAPN == int.Parse(parameter.MaPN.Text))

{

foreach(CTPN temp1 in temp.CTPNs)

{

foreach(SANPHAM temp2 in DataProvider.Ins.DB.SANPHAMs)

{

if(temp1.MASP==temp2.MASP)

{

if(temp2.SL-temp1.SL<0)

{

MessageBox.Show("Không thể xóa phiếu nhập vì sản phẩm nhập đã được bán !","THÔNG BÁO",MessageBoxButton.OK,MessageBoxImage.Error);

return;

}

else

temp2.SL -= temp1.SL;

}

}

}

DataProvider.Ins.DB.PHIEUNHAPs.Remove(temp);

}

}

DataProvider.Ins.DB.SaveChanges();

}

}

* + 1. Màn hình thêm phiếu nhập:

4.10.2.1. Giao diện:

Graphical user interface, application

Description automatically generated

Xóa sản phẩm đã chọn trong danh sách bên dưới

Thêm sản phẩm vào phiếu nhập

Nhấn để xác nhận nhập hàng

Nhập số lượng

Chọn sản phẩm

Hình 4.24. Giao diện màn hình thêm phiếu nhập

* + - 1. Mô tả chức năng:
* Nhấn nút thêm để thêm sản phẩm cần nhập vào phiếu nhập.
* Nhất nút xóa để xóa sản phẩm đã chọn trong danh sách sản phẩm nhập.
* Nhấn nút xác nhận để lưu phiếu nhập vào database và danh sách hiển thị.
  + - 1. Kỹ thuật xử lý:
* Source code xử lý:
* Chức năng thêm:

void \_AddSP(AddImportView paramater)

{

if (paramater.MaPN.Text == "")

{

System.Windows.MessageBox.Show("Bạn chưa nhập mã phiếu nhập!", "THÔNG BÁO",MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

foreach (PHIEUNHAP s in DataProvider.Ins.DB.PHIEUNHAPs)

{

if (int.Parse(paramater.MaPN.Text) == s.MAPN)

{

System.Windows.MessageBox.Show("Mã phiếu nhập đã tồn tại !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

}

try

{

if (int.Parse(paramater.SL.Text) < 10)

{

System.Windows.MessageBox.Show("Số lượng nhập không được nhỏ hơn 10!", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

}

catch

{

System.Windows.MessageBox.Show("Số lượng nhập không hợp lệ!", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

SANPHAM a = (SANPHAM)paramater.SP.SelectedItem;

foreach (Display display in paramater.ListViewSP.Items)

{

if (display.MaSp == a.MASP)

{

display.SL += int.Parse(paramater.SL.Text);

display.Tiennhap = display.SL \* (int)(a.GIA \* 5/6);

foreach (CTPN ct in LCTPN)

{

if (ct.MASP == display.MaSp)

ct.SL = display.SL;

}

goto There;

}

}

Display b = new Display(a.MASP, a.TENSP, a.SIZE, (int)((float)a.GIA \* 5/6), int.Parse(paramater.SL.Text), (int)((float)(int.Parse(paramater.SL.Text) \* a.GIA) \* 5/6));

CTPN ctpn = new CTPN()

{

MASP = a.MASP,

SL = int.Parse(paramater.SL.Text),

SANPHAM = a,

MAPN = int.Parse(paramater.MaPN.Text),

};

LCTPN.Add(ctpn);

LHT.Add(b);

There:

tongtien += int.Parse(paramater.SL.Text) \* (int)(a.GIA \* 5/6);

paramater.ListViewSP.ItemsSource = LHT;

paramater.ListViewSP.Items.Refresh();

paramater.SP.ItemsSource = LSP;

paramater.SP.Items.Refresh();

paramater.SP.SelectedItem = null;

paramater.SL.Text = "";

paramater.TT.Text = tongtien.ToString("#,### VNĐ");

}

* Chức năng xóa:

void \_DeleteSP(AddImportView paramater)

{

if (paramater.ListViewSP.SelectedItem == null)

{

System.Windows.MessageBox.Show("Bạn chưa chọn sản phẩm !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

MessageBoxResult h = System.Windows.MessageBox.Show(" Bạn có chắc muốn xóa sản phẩm.", "THÔNG BÁO", MessageBoxButton.YesNoCancel,MessageBoxImage.Question);

if (h == MessageBoxResult.Yes)

{

Display a = (Display)paramater.ListViewSP.SelectedItem;

tongtien -= a.Tiennhap;

paramater.TT.Text = String.Format("{0:0,0}", tongtien) + " VND";

LHT.Remove(a);

foreach (SANPHAM b in LSPSelected)

{

if (b.MASP == a.MaSp)

{

LSPSelected.Remove(b);

break;

}

}

foreach (CTPN b in LCTPN)

{

if (b.MASP == a.MaSp && b.SL == a.SL)

{

LCTPN.Remove(b);

break;

}

}

paramater.ListViewSP.Items.Refresh();

}

else

return;

}

* Chức năng xác nhận:

void \_SavePN(AddImportView paramater)

{

if (paramater.ListViewSP.Items.Count == 0)

{

System.Windows.MessageBox.Show("Thông tin phiếu nhập chưa đầy đủ !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

MessageBoxResult h = System.Windows.MessageBox.Show("Xác nhận nhập hàng?", "THÔNG BÁO", MessageBoxButton.YesNoCancel,MessageBoxImage.Question);

if (h == MessageBoxResult.Yes)

{

PHIEUNHAP temp = new PHIEUNHAP()

{

MAPN = int.Parse(paramater.MaPN.Text),

MAND = Const.ND.MAND,

NGAYNHAP = DateTime.Now,

CTPNs = new ObservableCollection<CTPN>(LCTPN),

};

foreach (CTPN s in LCTPN)

{

foreach (SANPHAM x in DataProvider.Ins.DB.SANPHAMs)

{

if (x.MASP == s.SANPHAM.MASP)

{

x.SL += s.SL;

}

}

}

DataProvider.Ins.DB.PHIEUNHAPs.Add(temp);

DataProvider.Ins.DB.SaveChanges();

System.Windows.MessageBox.Show("Nhập hàng thành công", "THÔNG BÁO");

LHT = new ObservableCollection<Display>();

paramater.MaPN.Clear();

LCTPN = new ObservableCollection<CTPN>();

paramater.ListViewSP.ItemsSource = LHT;

LSP = DataProvider.Ins.DB.SANPHAMs.Where(p => p.SL >= 0).ToList();

paramater.SP.Items.Refresh();

}

else

return;

}

* 1. Màn hình đổi mật khẩu:
     1. Giao diện:

Graphical user interface

Description automatically generated

Nhấn để đóng cửa sổ

Nhấn lưu mật khẩu mới

Hình 4.25. Giao diện màn hình đổi mật khẩu

* + 1. Mô tả chức năng:
* Khi nhấn nút Lưu, nếu điền thông tin hợp lệ (đúng mật khẩu cũ, mật khẩu mới khác mật khẩu cũ, mật khẩu nhập lại giống mật khẩu mới) sẽ hiện thông báo thành công và tự động đóng cửa sổ, ngược lại sẽ báo lỗi.
* Khi nhấn nút Thoát sẽ đóng cửa sổ.
  + 1. Kỹ thuật xử lý:
* Hiệu chỉnh các thuộc tính trong bảng Properties:

Text

Description automatically generated

*Tạo viền cho cửa sổ*

Graphical user interface, text

Description automatically generated

*Tạo hiệu ứng đổ bóng*

* Source code xử lý:
* **Kiểm tra hợp lệ và lưu mật khẩu mới:**

void SaveNewPass(ChangePassword p)

{

string a = Const.TenDangNhap;

User = DataProvider.Ins.DB.NGUOIDUNGs.Where(x => x.USERNAME == a).FirstOrDefault();

try

{

if (User.PASS != MD5Hash(Base64Encode(OldPass)))

{

MessageBox.Show("Mật khẩu cũ không đúng!", "Thông báo", MessageBoxButton.OK, MessageBoxImage.Warning);

}

else if (Password == OldPass)

{

MessageBox.Show("Mật khẩu mới không được giống mật khẩu cũ!", "Thông báo", MessageBoxButton.OK, MessageBoxImage.Warning);

}

else if (Password != NewPass)

{

MessageBox.Show("Mật khẩu nhập lại không đúng!", "Thông báo", MessageBoxButton.OK, MessageBoxImage.Warning);

}

else

{

User.PASS = MD5Hash(Base64Encode(Password));

DataProvider.Ins.DB.SaveChanges();

MessageBox.Show("Đổi mật khẩu thành công!", "Thông báo");

p.Close();

}

}

catch

{

MessageBox.Show("Vui lòng nhập thông tin!", "Thông báo", MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

* 1. Màn hình quên mật khẩu:
     1. Giao diện:

Graphical user interface, text, application, chat or text message

Description automatically generated

Nhấn để gửi mật khẩu mới về mail đã đăng ký

Hình 4.26. Giao diện màn hình quên mật khẩu

* + 1. Mô tả chức năng:
* Khi nhấn nút lấy lại mật khẩu thì mật khẩu của tài khoản sẽ được gửi về mail mà người dùng đăng ký tài khoản. Nếu mail chưa được đăng ký sẽ báo lỗi.
  + 1. Kỹ thuật xử lý:
* Hiệu chỉnh các thuộc tính trong bảng Properties:

Text

Description automatically generated

*Tạo viền cho cửa sổ*

* Source code xử lý:
* **Kiểm tra hợp lệ và gửi mật khẩu về mail:**

void \_SendPass(ForgetPassView parameter)

{

int dem = DataProvider.Ins.DB.NGUOIDUNGs.Where(p => p.MAIL == parameter.mail.Text).Count();

if(dem==0)

{

MessageBox.Show("Email này chưa được đăng lý !", "THÔNG BÁO", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

Random rand = new Random();

string newpass = rand.Next(100000, 999999).ToString();

foreach(NGUOIDUNG temp in DataProvider.Ins.DB.NGUOIDUNGs)

{

if(temp.MAIL==parameter.mail.Text)

{

temp.PASS = LoginViewModel.MD5Hash(LoginViewModel.Base64Encode(newpass));

break;

}

}

DataProvider.Ins.DB.SaveChanges();

string nd = "Vui lòng nhập mật khẩu " + newpass + " để đăng nhập. Trân trọng !";

MailMessage message=new MailMessage("clothesmanagement1412@gmail.com", parameter.mail.Text,"Lấy lại mật khẩu",nd);

SmtpClient smtpClient = new SmtpClient("smtp.gmail.com",587);

smtpClient.EnableSsl = true;

smtpClient.Credentials = new NetworkCredential("clothesmanagement1412@gmail.com","doanlttq");

smtpClient.Send(message);

MessageBox.Show("Đã gửi mật khẩu vào Email đăng ký !", "Thông báo");

}

KẾT LUẬN

* + Ưu điểm của đồ án:
* Hoàn thiện về mặt giao diện và tính năng giúp tăng tính tiện dụng và thân thiện cho người dùng.
* Giao diện bắt mắt, cuốn hút người dùng.
* Có biểu đồ báo cáo, điều đó giúp thể hiện trực quan dữ liệu, khiến cho người dùng dễ dàng hơn trong việc thống kê.
  + Hạn chế của đồ án:
* Phản hồi chậm khi lần đầu chuyển sang danh mục sản phẩm.
  + Hướng phát triển của đồ án:
* Liên kết với các cửa hàng khác trong khu vực thông qua Internet.
* Thêm chức năng quản lý lương cho nhân viên.
* Thêm chức năng quản lý xuất/nhập kho.
* Khuyến mãi cho khách hàng theo vào dịp lễ, dịp sinh nhật.
* Quản lý chi tiết lợi nhuận, chi phí trong mục thống kê.

TÀI LIỆU THAM KHẢO

[1] Trần Thị Hồng Yến (2021)*, Bài giảng môn Lập trình trực quan*, Khoa CNPM, Trường ĐH. Công Nghệ Thông Tin.

[2] Tìm hiểu về ngôn ngữ lập trình C#:

<https://codegym.vn/blog/2020/06/29/c-la-gi-tim-hieu-ve-ngon-ngu-lap-trinh-c/>

[3] Tìm hiểu C#: <https://viettuts.vn/csharp>

[4] Entity Framework là gì: <https://chiasekinang.com/entity-framework-la-gi/>

[5] Giới thiệu Entity Framework:

<https://comdy.vn/entity-framework/gioi-thieu-entity-framework/>

[6] Tìm hiểu SQL Server: <https://itnavi.com.vn/blog/sql-server>

[7] WPF là gì: <https://chiasekinang.com/wpf-la-gi/>

[8] 6 lý do bạn nên sử dụng phần mềm quản lý doanh nghiệp:

<https://tripione.vn/6-l-y-do-ban-nen-su-dung-phan-mem-quan-ly-doanh-nghiep/>

[9] Howkteam – Lập trình WPF cơ bản: <https://www.youtube.com/playlist?list=PL33lvabfss1ywgHcDF2aB8YBxwtj1_Rjk>

[10] Howkteam – Lập trình phần mềm quản lý kho: <https://www.youtube.com/playlist?list=PL33lvabfss1zfGxCcTIYr5IjsyweWWtAO>

[11] Chuyển đổi chuỗi sang base64:

<https://stackoverflow.com/questions/11743160/how-do-i-encode-and-decode-a-base64-string>

[12] Chuyển đổi chuỗi sang MD5:

<https://coderwall.com/p/4puszg/c-convert-string-to-md5-hash>