KIỂM THỬ PHẦN MỀM

Quản lý cửa hàng mỹ phẩm

Nhóm 10

Giảng viên: Ph.D Võ Phương Bình

1. Tài liệu yêu cầu
   1. Các giai đoạn

|  |  |  |
| --- | --- | --- |
| Tên bài tập | Nội dung | Thời gian |
| Giai đoạn 1 | | |
| Lab 01 | Test Scenario and Test Case (Manual Testing) | 29/03/2023 |
| Lab 01.2 | Test Scenario and Test Case (Đề tài nhóm) | 05/04/2023 |
| Lab 02 | Unit Test and Intergrated with C# | 12/04/2023 |
| Lab 02.2 | Unit Test and Intergrated with C# (Đề tài nhóm - Buổi 1) | 15/4/2023 |
|  | Unit Test and Intergrated with C# (Đề tài nhóm - Buổi 2) | 19/04/2023 |
| Giai đoạn 2 | | |
| Viết báo cáo | Viết báo cáo đề tài | 22/4/2023 |
| Báo cáo đề tài |  |  |

* 1. Giới thiệu

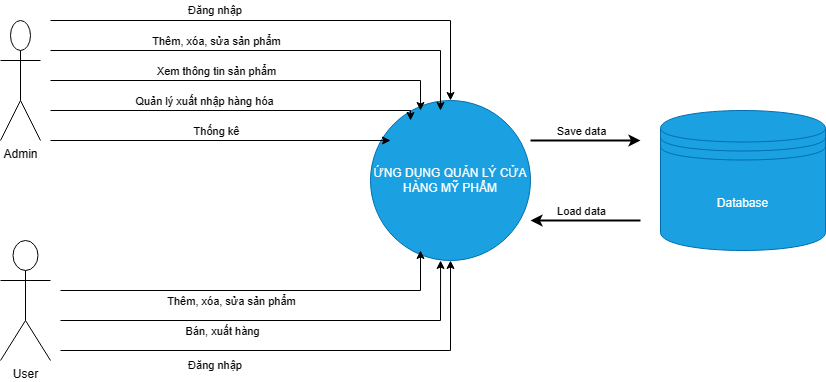
Dựa vào nhu cầu đời sống hiện tại, phương diện quản lý hàng hóa được số hóa rộng rãi. Ứng dụng quản lý cửa hàng mỹ phẩm là một ứng dụng được xây dựng sử dụng Winform cho giao diện, mô hình ADO.net, đáp ứng nhu cầu quản lý cho chủ các cửa hàng mỹ phẩm, ứng dụng cho phép chủ cửa hàng quản lý sản phẩm một cách rõ ràng thông qua ứng dụng không phải xem xét quá nhiều giấy tờ, ngoài ra còn có thể quản lý việc mua bán, xem doanh thu cửa hàng, quản lý nhân viên. Người nhân viên có các thao tác đơn giản để bán hàng và không thể có các chức năng quản lý. Thông qua đó chủ cửa hàng có thể kiểm soát việc mua, bán hàng hóa, cũng như doanh thu để có thể đề ra kế hoạch, định hướng cho cửa hàng.

Quản trị viên có thể:

* Đăng nhập
* Thêm/ xoá/ sửa sản phẩm
* Xem thông tin chi tiết sản phẩm
* Quản lý nhập/ xuất hàng
* Thống kê doanh thu sản phẩm bán chạy/ bán chậm

Nhân viên có thể:

* Đăng nhập
* Thực hiện chức năng bán hàng/ xuất hàng
* Thêm/ xoá/ sửa thông tin sản phẩm
  1. System Context Diagram



* 1. Tóm tắt tính năng

|  |  |
| --- | --- |
| ID | Mô tả tính năng |
| 1 | Quản lý có thể thêm, xoá, sửa sản phẩm |
| 2 | Quản lý có thể xem thông tin sản phẩm |
| 3 | Quản lý có thể thống kê doanh thu |
| 4 | Nhân viên có thể sửa thông tin sản phẩm, thực hiện các chức năng bán hàng như chọn sản phẩm, thêm sản phẩm, thanh toán,... |

* 1. Diagram

     Description automatically generatedUse Case Diagram
  2. Non-functional Requirements and Constraints
  3. Bảng chú thích
  4. Software Environments

- Ứng dụng được viết bằng ngôn ngữ lập trình C#

- Development environment: Microsoft Visual Studio 2019.

1. Class Design
   1. Use-Case Realizations with GRASP Patterns
      1. System Sequence Diagram (new)
      2. Operation Contract (new)

|  |  |
| --- | --- |
| ID | Tên hành động |
| CO-01 | login |
| CO-02 | enterPrimaryProduct |
| CO-03 | deletePrimaryProduct (new) |
| CO-04 | updatePrimaryProduct (new) |
| CO-05 | showPrimaryProduct |
| CO-06 | searchByName |
| CO-07 | viewProductInformation |
| CO-08 | enterSaleInformation |
| CO-09 | deleteProductInformation |
| CO-010 | updateProductInformation |
| CO-011 | showProductInformation |
| CO-012 | enterBill |
| CO-013 | deleteBill |
| CO-014 | updateBill |
| CO-015 | showBill |
| CO-016 | enterTransaction |
| CO-017 | deleteTransaction |
| CO-018 | updateTransaction |
| CO-019 | showTransaction |

* + - 1. login

|  |  |
| --- | --- |
| Hoạt động | login(userName: string, password: string) |
| Cross References |  |
| Preconditions |  |
| Post-conditions |  |

* + - 1. enterPrimaryProduct
      2. deletePrimaryProduct
      3. updatePrimaryProduct
      4. showPrimaryProduct
      5. searchByName
      6. viewProductInformation
      7. enterSaleInformation
      8. deleteProductInformation
      9. updateProductInformation
      10. showProductInformation
      11. enterBill
      12. deleteBill
      13. updateBill
      14. showBill
      15. enterTransaction
      16. deleteTransaction
      17. updateTransaction
      18. showTransaction
    1. Operation Sequence Diagram
       1. login
       2. enterPrimaryProduct
       3. deletePrimaryProduct
       4. updatePrimaryProduct
       5. showPrimaryProduct
       6. searchByName
       7. viewProductInformation
       8. enterSaleInformation
       9. deleteProductInformation
       10. updateProductInformation
       11. showProductInformation
       12. enterBill
       13. deleteBill
       14. updateBill
       15. showBill
       16. enterTransaction
       17. deleteTransaction
       18. updateTransaction
       19. showTransaction
  1. Design Class Model

Student / Advisor is a kind of Manager (extend).

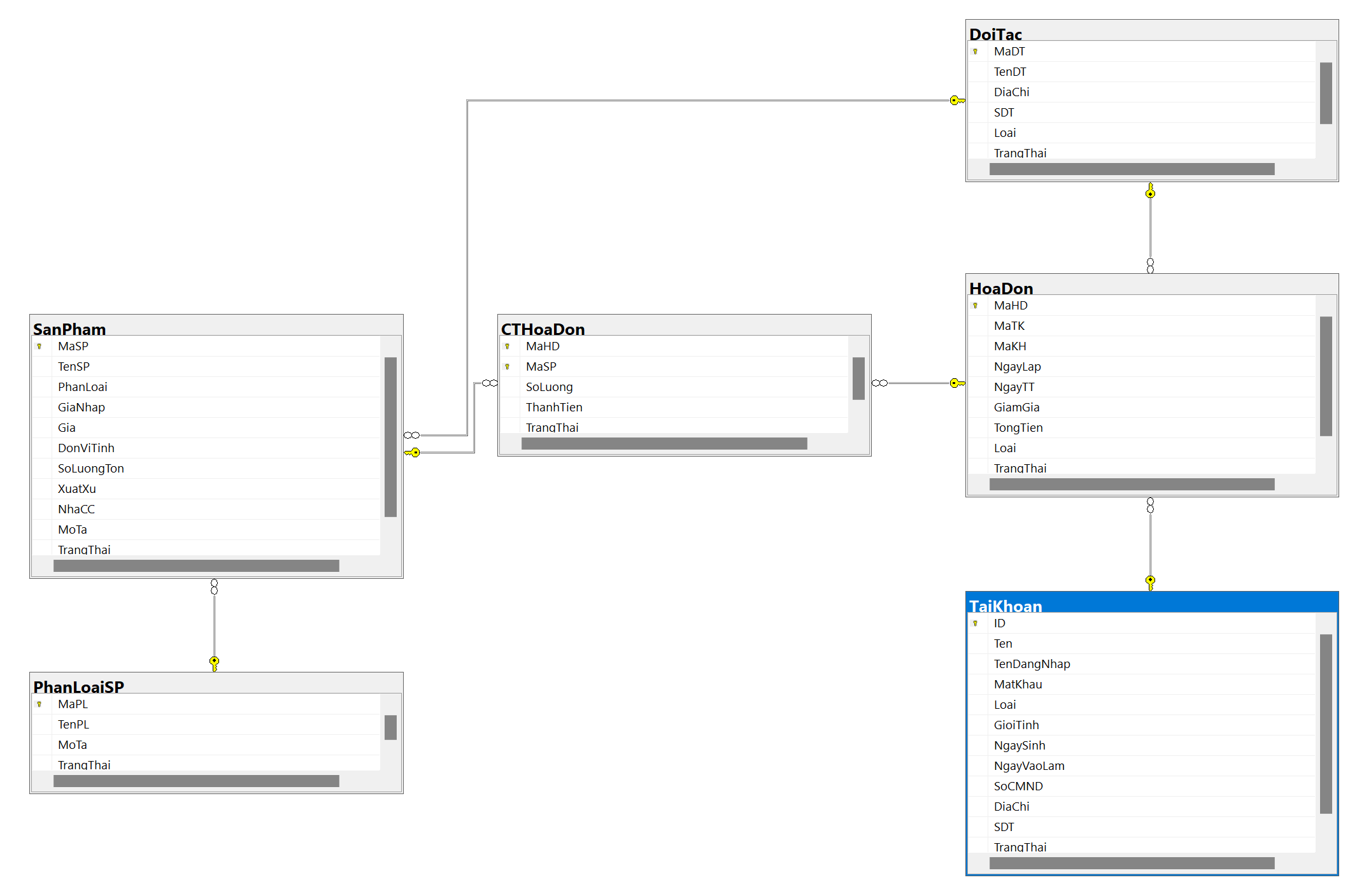
1. Implementation Class Model
   1. Show the difference between implementation class model and design class model
   2. Calculate Line of Code (new)
2. Programming
   1. Snapshots of system execution
   2. Source Code Listing
      1. Program.cs
3. Unit Testing
   1. Snapshots of testing result
   2. Unit Testing Code Listing
      1. AdminTest.cs
4. Measurement

Table

Description automatically generated

Appendix

SQL Server Database Tables



Test Data

Table

Description automatically generated

Table

Description automatically generated with medium confidence

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated