

UBND TỈNH LÂM ĐỒNG
TRƯỜNG CAO ĐẲNG ĐÀ LẠT

GIÁO TRÌNH

MÔ ĐUN: XÂY DỰNG PHẦN MỀM QUẢN LÝ BÁN HÀNG

NGÀNH/NGHỀ: CÔNG NGHỆ THÔNG TIN (ÚDPM)

TRÌNH ĐỘ: CAO ĐẲNG

*(Ban hành kèm theo Quyết định số: /QĐ-CDNDL ngày ...tháng...năm...
của Hiệu trưởng Trường Cao đẳng Đà Lạt)*

LƯU HÀNH NỘI BỘ

Lâm Đồng, năm 2017

TUYÊN BỐ BẢN QUYỀN

Tài liệu này thuộc loại sách giáo trình nên các nguồn thông tin có thể được phép dùng nguyên bản hoặc trích dùng cho các mục đích về đào tạo và tham khảo.

Mọi mục đích khác mang tính lệch lạc hoặc sử dụng với mục đích kinh doanh thiếu lành mạnh sẽ bị nghiêm cấm.

LỜI GIỚI THIỆU

Trước sự phát triển không ngừng của các loại hình dịch vụ phục vụ, đặc biệt là sau khi hội nhập WTO các loại hình dịch vụ càng ngày càng phát triển, một trong những dịch vụ đó chính là dịch vụ vui chơi giải trí, Quán Café. Cũng chính điều này đã nảy sinh nhiều vấn đề trong công tác phục vụ và dịch vụ tại các Quán. Do lượng khách hàng mỗi ngày một tăng đồng thời thị hiếu của khách hàng và những đòi hỏi về dịch vụ ngày một cao thêm vào đó là số lượng bàn, Dịch vụ ngày được mở rộng, số lượng nhân viên càng ngày càng tăng theo và thêm vào đó thì việc đòi hỏi về quản lý cũng càng chặt chẽ hơn, số lượng công việc quản lý cũng gia tăng theo thời gian trong đó sức người có hạn chế vì điều đó ATO đã nghiên cứu khảo sát trên hàng trăm nhà hàng, Quán Café và đưa ra giải pháp đáp ứng yêu cầu quản lý. Chức năng quản lý Quán giúp bạn có thể quản lý tốt các dịch vụ, kinh doanh mà khách hàng sử dụng, thông tin sử dụng đồ uống, đồ ăn nhanh.... tại quán.

Đà Lạt, ngày 07 tháng 7 năm 2017

Tham gia biên soạn

Chủ biên: Ths. Phạm Đình Nam

MỤC LỤC

TRANG

LỜI GIỚI THIỆU

BÀI 1: THIẾT KẾ CƠ SỞ DỮ LIỆU	1
1. Khảo sát và phân tích bài toán quản lý bán hàng.....	1
1.1 Xác định luồng dữ liệu.....	1
1.1.1. Sơ đồ dữ liệu quản lý lương	1
1.1.2. Quản lý nhân viên	4
Trả lời YC	4
1.2. Mô tả ứng dụng.....	4
1.2.1. Quản lý thông tin.....	4
1.2.2. Xuất báo cáo	4
2. Lựa chọn hệ quản trị cơ sở dữ liệu.....	5
2.1 Phân tích ưu điểm, hạn chế của các hệ quản trị cơ sở dữ liệu mạnh nhất hiện nay	5
2.1.1 MySQL	5
2.1.2. SQL Server	5
2.2 Những điểm mạnh của hệ quản trị cơ sở dữ liệu được chọn (SQL Server)	6
3. Thiết kế cơ sở dữ liệu Quản lý bán hàng	6
3.1 Xác định các bảng.....	6
3.2 Xác định các thuộc tính của từng bảng.....	7
4. Thiết kế các Table trong CSDL	7
4.1 Xác định kiểu dữ liệu từng trường của bảng	7
4.2 Tạo bảng bằng công cụ SQL Management Studio	8
5. Tạo liên kết cho các Table trong CSDL.....	16
5.1 Xác định khóa chính, khóa ngoại	16
5.2 Thiết lập quan hệ cho các bảng bằng giao diện SQL Management Studio...	16
6. Nhập dữ liệu mẫu cho các Table.....	20
6.1 Nhập dữ liệu mẫu cho các bảng bên quan hệ 1	20
6.2 Nhập dữ liệu mẫu cho các bảng bên quan hệ n	21
7. Cài đặt cơ sở dữ liệu Quản lý bán hàng	21
7.1 Tạo user đăng nhập cơ sở dữ liệu	21

7.2 Cấp quyền user sử dụng cơ sở dữ liệu.....	23
7.3 Lưu trữ và quản lý cơ sở dữ liệu quản lý bán hàng	23
BÀI 2: THIẾT KẾ CÁC CHỨC NĂNG CHÍNH.....	27
1. Tạo project và kết nối cơ sở dữ liệu.....	27
1.1 Thiết lập chuỗi kết nối trong Properties của project.....	27
1.2 Khai báo các đối tượng dùng chung ADO.NET.....	27
1.3 Khởi tạo kết nối đến cơ sở dữ liệu.....	28
2. Thiết kế chức năng Cập nhật dữ liệu.....	29
2.1 Thiết kế form cập nhật dữ liệu cho Hàng hóa	29
2.2 Thiết kế form cập nhật dữ liệu cho khách hàng	29
2.3 Thiết kế form cập nhật dữ liệu cho Phiếu thu.....	30
2.4 Thiết kế form cập nhật dữ liệu cho Phiếu chi.....	31
2.5 Thiết kế form cập nhật dữ liệu cho Hóa đơn	31
2.6 Tạo các form giao diện tương ứng của từng chức năng	32
2.7 Xử lý các biến cố, sự kiện trên từng form	36
3 Thiết kế chức năng Xem thông tin.....	41
3.1 Thiết kế form nạp một phiếu đã nhập.....	41
3.1.1 Thiết kế form nạp danh sách phiếu nhập	41
3.1.2 Thiết kế form hiện thông tin chi tiết một phiếu đã nhập.....	43
3.2 Thiết kế form nạp một phiếu đã xuất.....	43
3.2.1 Thiết kế form nạp danh sách phiếu xuất	43
3.2.2 Thiết kế form hiện thông tin chi tiết một phiếu đã xuất.....	44
4. Thiết kế chức năng Tìm kiếm thông tin	44
4.1 Tạo form giao diện.....	44
4.2 Xử lý truy vấn, sự kiện	44
4.2.1 Viết lệnh tìm kiếm theo một tiêu chí riêng lẻ.....	44
4.2.2 Viết lệnh tìm kiếm theo nhiều điều kiện kết hợp	45
5. Thiết kế chức năng In.....	45
5.1 Tạo form In	45
5.1.1 Thêm nút lệnh in có sử dụng biểu tượng in	45
5.1.2 Tạo form chứa CrystalReportViewer để in báo cáo.....	45
5.2 Xây dựng thủ tục SQL để tạo nguồn dữ liệu cần in	46

5.2.1	Viết thủ tục lưu trữ	46
5.2.2	Sử dụng thủ tục lưu trữ từ chương trình.....	46
BÀI 3: THIẾT KẾ GIAO DIỆN CHƯƠNG TRÌNH.....		47
1.	Tạo form đăng nhập với người dùng.....	47
1.1	Tạo bảng lưu thông tin đăng nhập	47
1.2	Thiết kế giao diện đăng nhập.....	48
1.3	Viết lệnh đăng nhập	49
2.	Tạo form giao diện chính của chương trình.....	50
2.1	Thiết lập các thuộc tính của form chính	50
2.2	Thiết kế hệ thống menu, toolbar	50
2.2.1	Thiết kế hệ thống menu	50
2.2.2	Thiết kế hệ thống toolbar	51
3.	Tạo các liên kết đến các chức năng trong chương trình	51
3.1	Chức năng mở form	51
3.2	Chức năng trợ giúp	51
3.3	Chức năng thiết lập hệ thống	51
BÀI 4: THIẾT KẾ CÁC BÁO CÁO.....		52
1.	Xác định chức năng của báo cáo	52
1.1	Chức năng in danh sách	52
1.2	Chức năng in thống kê, tổng hợp.....	52
2.	Xác định, tạo nguồn dữ liệu cho báo cáo.....	53
2.1	Xây dựng nguồn dữ liệu ảo để thiết kế báo cáo	53
2.2.	Xác định nguồn dữ liệu thật để báo cáo	54
3.	Tạo báo cáo với công cụ	54
3.1	Sử dụng công cụ Report Wizard để tạo báo cáo.....	54
3.2	Thêm tiêu đề cho báo cáo	56
4.	Chỉnh sửa các thành phần, thuộc tính trong báo cáo	57
4.1	Định dạng ngày trong báo cáo	57
4.2	Kẽ viền cho báo cáo.....	57
5.	Tạo form truyền tham số.....	57
6.	Xử lý tham số với báo cáo	61
7.	Thiết kế các báo cáo chi tiết	62

7.1 Tạo thủ tục SQL để kết xuất dữ liệu.....	62
7.2 Tạo báo cáo sử dụng nguồn dữ liệu từ thủ tục SQL.....	72
BÀI 5: HOÀN THIÊN, ĐÓNG GÓI PHẦN MỀM.....	73
1. Kiểm thử phần mềm	73
2. Hiệu chỉnh phần mềm.....	74
2.1 Xác định nguyên nhân gây lỗi	74
2.2 Khắc phục lỗi.....	74
3. Xác định công cụ đóng gói	75
4. Đặt bảo mật cho cơ sở dữ liệu	75
4.1 Xây dựng thư viện mã hóa.....	75
4.2 Đặt mật khẩu cho cơ sở dữ liệu và code.....	75
5. Đóng gói phần mềm.....	76

GIÁO TRÌNH MÔ ĐUN

Tên mô đun: Xây dựng phần mềm quản lý bán hàng

Mã mô đun: MD24

Vị trí, tính chất, ý nghĩa và vai trò của mô đun:

- Vị trí: Mô đun này được học sau các mô đun Lập trình windows 2, Quản trị cơ sở SQL Server, Quản trị cơ sở dữ liệu với Access, Phân tích và thiết kế hệ thống thông tin.

- Tính chất: Là mô đun chuyên môn bắt buộc, yêu cầu sinh viên phải sử dụng thành thạo công cụ lập trình.

- Ý nghĩa và vai trò của mô đun: Tài liệu này được thiết kế theo từng mô đun/môn học thuộc hệ thống mô đun/môn học của một chương trình, để đào tạo hoàn chỉnh nghề Công nghệ thông tin (ứng dụng phần mềm) ở trình độ Cao đẳng, đồng thời được dùng làm Giáo trình cho học viên trong các khoá đào tạo ngắn hạn hoặc cho các công nhân kỹ thuật, các nhà quản lý và người sử dụng nhân lực tham khảo.

Mục tiêu của mô đun:

Về kiến thức:

- Phân tích thiết kế được cơ sở dữ liệu phục vụ bài toán quản lý trong thực tế;
- Trình bày được cách kết nối cơ sở dữ liệu và phương pháp tương tác dữ liệu;

Về kỹ năng:

- Tạo và quản trị được phần mềm ứng dụng trên các môi trường VB, Access,...;
- Thiết kế được giao diện cho ứng dụng;
- Viết được mã lệnh xử lý các sự kiện;
- Xây dựng được cơ sở dữ liệu quản lý bán hàng cho một doanh nghiệp vừa và nhỏ;
- Xây dựng được phần mềm để quản lý bán hàng với các tính năng chính như cập nhật dữ liệu, truy vấn dữ liệu, báo cáo,...;
- Kiểm thử, đánh giá được các chức năng của ứng dụng với các dữ liệu thực tế;

Về năng lực tự chủ và trách nhiệm:

- Có khả năng tự nghiên cứu, tự học, tham khảo tài liệu liên quan đến môn học để vận dụng vào hoạt động học tập.
- Vận dụng được các kiến thức tự nghiên cứu, học tập và kiến thức, kỹ năng đã được học để hoàn thiện các kỹ năng liên quan đến môn học một cách khoa học, đúng quy định.

Nội dung của mô đun:

BÀI 1: THIẾT KẾ CƠ SỞ DỮ LIỆU

Mã Bài: MD24_01

Giới thiệu:

Việc thiết kế cơ sở dữ liệu đã được các bạn thực hiện qua các mô đun như Quản trị CSDL với Microsoft Access, Quản trị CSDL với Microsoft SQL Server cũng như đã được học qua môn học phân tích và thiết kế hệ thống. Trong bài này chúng tôi nói về việc thiết kế cơ sở dữ liệu trên SQL Server.

Mục tiêu:

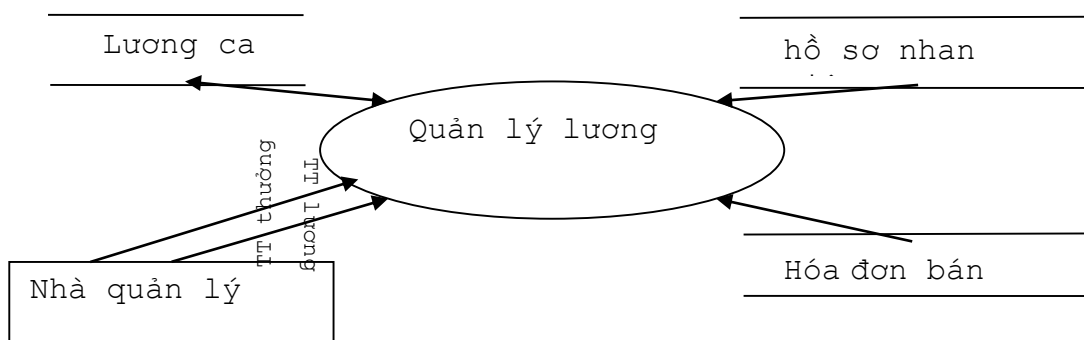
- Mô tả được đặc trưng, yêu cầu chung của bài toán quản lý bán hàng;
- Phân tích và thiết kế được hệ thống thông tin cho một số bài toán quản lý bán hàng;
- Thiết kế được cơ sở dữ liệu Quản lý bán hàng bằng Microsoft Access hoặc SQL Server;
- Đánh giá được tính hợp lý của cơ sở dữ liệu;
- Rèn luyện tính cẩn thận tỉ mỉ, khoa học, sáng tạo.

Nội dung chính:

1. Khảo sát và phân tích bài toán quản lý bán hàng

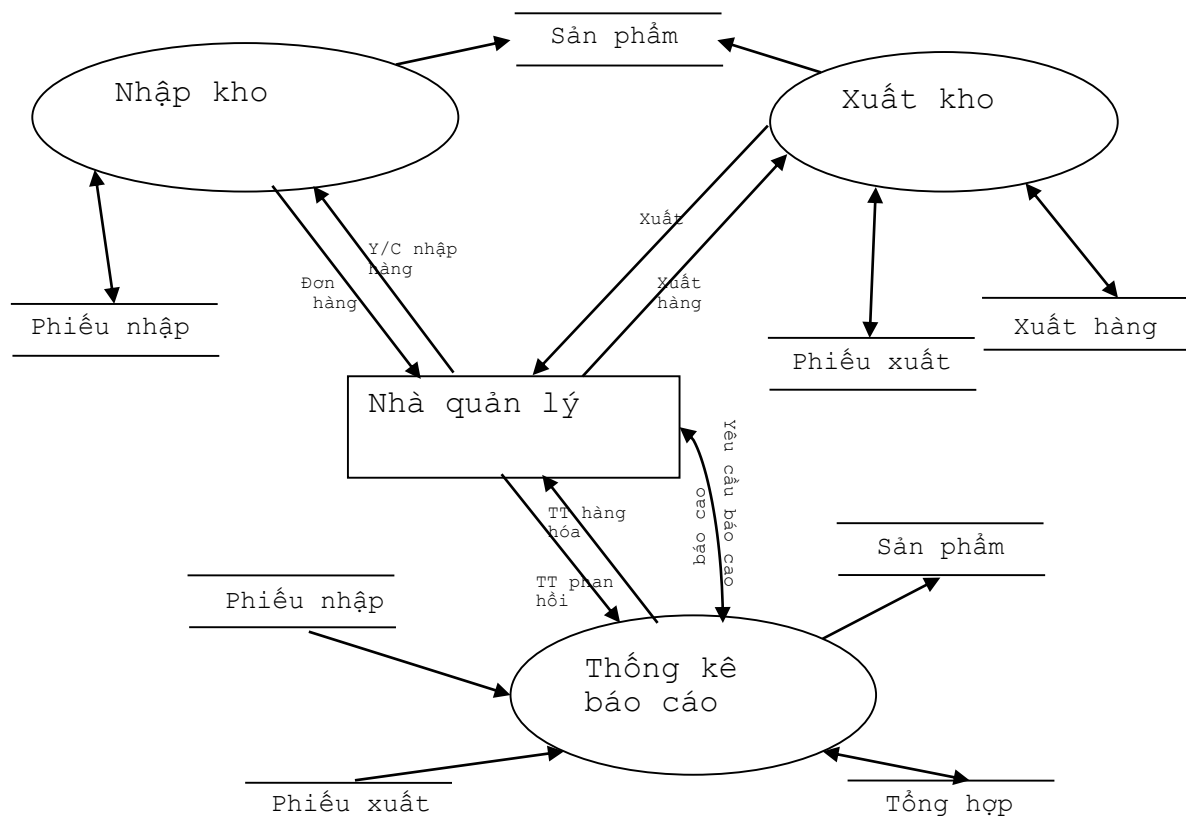
1.1 Xác định luồng dữ liệu

1.1.1. Sơ đồ dữ liệu quản lý lương

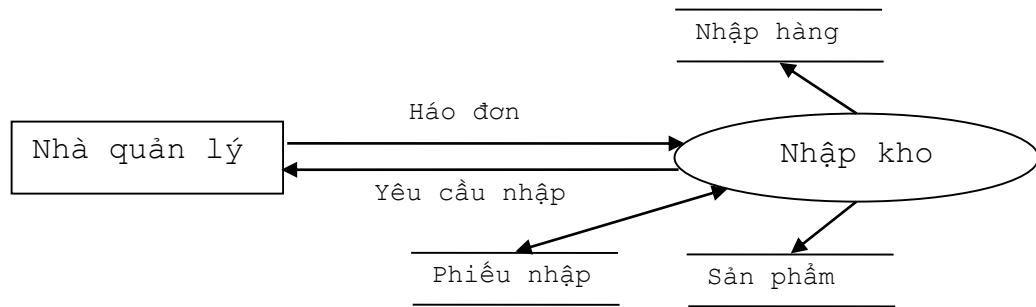


Sơ đồ dữ liệu quản lý kho

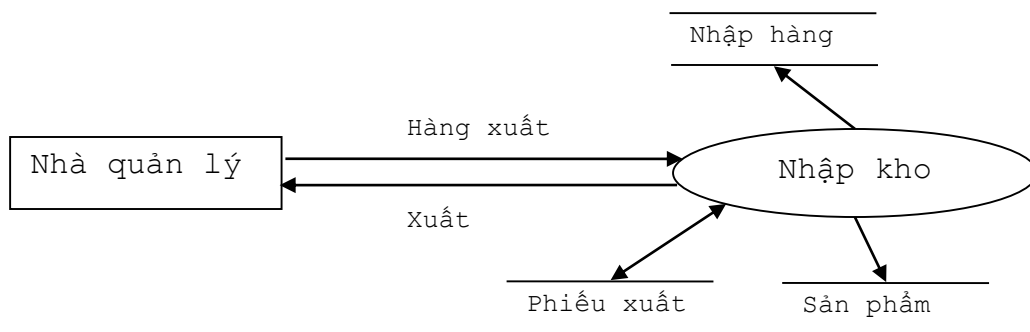
Sơ đồ luồng dữ liệu mức 0



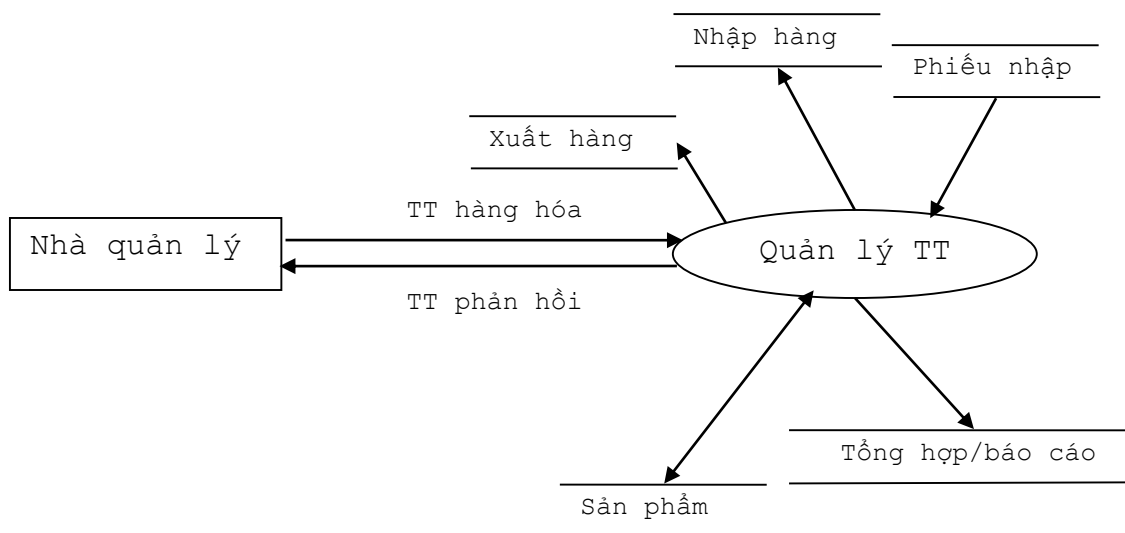
Sơ đồ luồng dữ liệu tiến trình nhập kho



Sơ đồ luồng dữ liệu tiến trình xuất kho

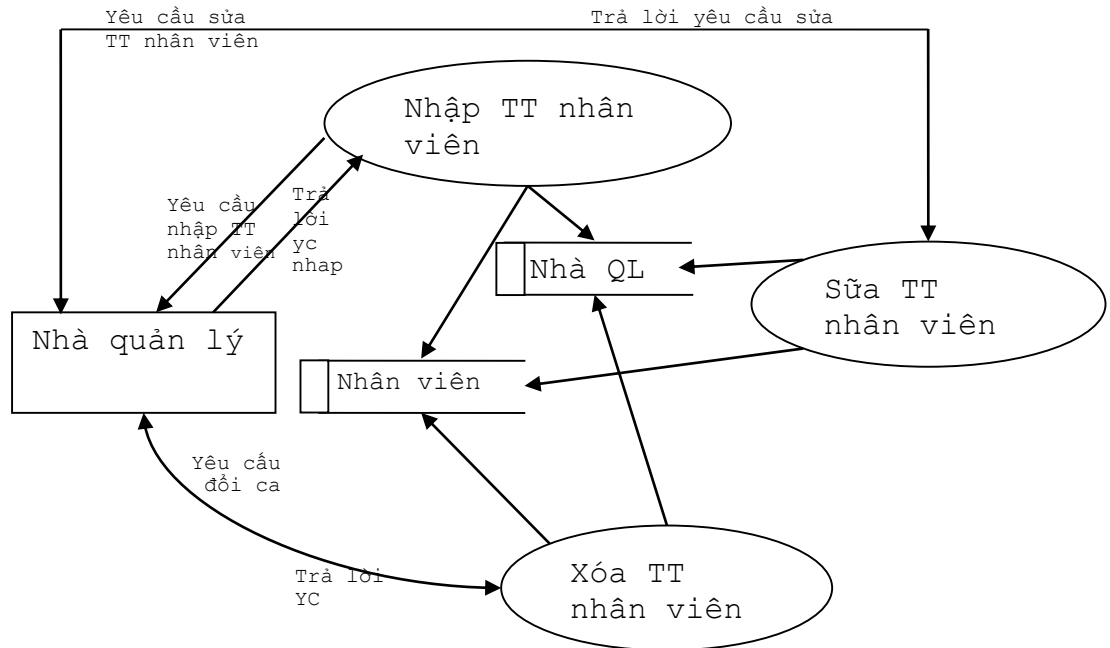


Sơ đồ dữ liệu tiến trình thống kê báo cáo



1.1.2. Quản lý nhân viên

Mô hình dưới đỉnh của quản lý thông tin Nhân viên



1.2. Mô tả ứng dụng

1.2.1. Quản lý thông tin

- Quản lý hóa đơn cho từng phòng gồm: Phí Phục Vụ, tiền ăn, tiền uống,...
- Quản lý loại bàn: mã số, loại bàn, giá bán, ...
- Quản lý xuất nhập các loại thực phẩm.
- Quản lý nhân viên: thông tin nhân viên, ca kíp làm việc của mỗi nhân viên, lương tiền thu được trong mỗi ca làm việc, ...
- Quản lý kho: xuất, nhập, tồn kho, báo cáo thu chi.
- Phần mềm ứng dụng hỗ trợ quản lý tính tiền, bán hàng và quản lý kho hàng cho các doanh nghiệp hoạt động kinh doanh trong lĩnh vực thương mại và dịch vụ.

1.2.2. Xuất báo cáo

- Xuất báo cáo theo ngày, tháng, quý, năm. Các báo cáo gồm báo cáo tổng quan và báo cáo chi tiết. Thống kê báo cáo lượng khách cho mỗi ngày, mỗi tháng và quý để có biện pháp thay đổi hoạt động của quán.

2. Lựa chọn hệ quản trị cơ sở dữ liệu

2.1 Phân tích ưu điểm, hạn chế của các hệ quản trị cơ sở dữ liệu mạnh nhất hiện nay

2.1.1 MySQL

MySQL là một trong những cơ sở dữ liệu có khả năng mở rộng phổ biến nhất hiện nay. Nó giàu các tính năng, là một sản phẩm mã nguồn mở mạnh mẽ trên các website và các ứng dụng online. Việc bắt đầu với MySQL là cực kì dễ dàng và các nhà phát triển dễ dàng tiếp cận với một lượng lớn các thông tin về cơ sở dữ liệu trên internet.

*** Ưu điểm của MySQL**

Dễ dàng sử dụng: MySQL có thể dễ dàng cài đặt. Với các công cụ bên thứ 3 làm cho nó càng dễ đơn giản hơn để có thể sử dụng.

Giàu tính năng: MySQL hỗ trợ rất nhiều chức năng SQL được mong chờ từ một hệ quản trị cơ sở dữ liệu quan hệ-cả trực tiếp lẫn gián tiếp.

Bảo mật: Có rất nhiều tính năng bảo mật, một số ở cấp cao đều được xây dựng trong MySQL.

Khả năng mở rộng và mạnh mẽ: MySQL có thể xử lý rất nhiều dữ liệu và hơn thế nữa nó có thể được mở rộng nếu cần thiết.

Nhanh: Việc đưa ra một số tiêu chuẩn cho phép MySQL để làm việc rất hiệu quả và tiết kiệm chi phí, do đó nó làm tăng tốc độ thực thi.

*** Nhược điểm của MySQL**

Giới hạn: Theo thiết kế, MySQL không có ý định làm tất cả và nó đi kèm với các hạn chế về chức năng mà một vào ứng dụng có thể cần.

Độ tin cậy: Cách các chức năng cụ thể được xử lý với MySQL (ví dụ tài liệu tham khảo, các giao dịch, kiểm toán,...) làm cho nó kém tin cậy hơn so với một số hệ quản trị cơ sở dữ liệu quan hệ khác.

Sự phát triển chậm: Mặc dù MySQL vẫn là một sản phẩm công nghệ mã nguồn mở phổ biến nhưng có những phản nản liên quan đến quá trình phát triển kể từ khi nó được mua lại.

2.1.2. SQL Server

*** Ưu điểm của SQL Server**

Tích hợp chặt chẽ trọn bộ hệ thống và công cụ phát triển phần mềm, đặc biệt ở mảng .NET. MSSQL còn hỗ trợ XML trực tiếp trong DB.

MS SQL Server có tính bảo mật cao, hệ thống xác thực người dùng của MSSQL cũng qua mặt MySQL.

Khả năng phục hồi (sau khi bị crash) của MSSQL tốt hơn MySQL. MSSQL cung cấp nhiều phương thức nhân bản nên việc phục hồi dữ liệu trên MSSQL dễ dàng hơn.

*** Nhược điểm của SQL Server**

Tính về mặt hiệu suất, có rất nhiều thử nghiệm và tường trình cho rằng MSSQL perform kém hơn MySQL nhiều mặt. MSSQL đòi hỏi tài nguyên rất lớn (CPU mạnh, nhiều RAM...), nếu không nó rất ì ạch. MySQL không đòi hỏi nhiều như MSSQL. MySQL có thể chạy trên các UNIX highend system và perform tốt hơn MSSQL trên Windows highend server trong nhiều trường hợp.

MSSQL là hệ quản trị có bản quyền, phải trả gần một ngàn rưỡi đô cho 1 license MSSQL standard và khi cần support, bạn phải trả thêm tiền support (tùy case). MS vẫn cung cấp bản MSSQL không thu phí dành cho mục đích development. Tài liệu về MSSQL cũng rất nhiều trên mạng.

2.2 Những điểm mạnh của hệ quản trị cơ sở dữ liệu được chọn (SQL Server)

Microsoft SQL Server 2008 là một hệ thống quản lý cơ sở dữ liệu sử dụng truy vấn SQL để trao đổi dữ liệu thực hiện các câu kết nối và truy vấn vào bên trong dữ liệu nhằm lưu trữ dữ liệu một cách an toàn hơn. Một hệ thống quản trị cơ sở dữ liệu bao gồm Databases, database engine và các ứng dụng dùng để quản lý dữ liệu. SQL Server 2008 được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn (Very Large Database Environment) lên đến Tera-Byte và có thể phục vụ cùng lúc cho hàng ngàn user. SQL Server 2008 có thể kết hợp "ăn ý" với các server khác như Microsoft Internet Information Server (IIS), E-Commerce Server, Proxy Server....SQL Server 2008 có rất nhiều phiên bản để cho từng đối tượng sử dụng. Hiện nay SQL Server đã phát triển lên đến phiên bản Microsoft SQL Server 2008, nhưng chúng tôi sử dụng phiên bản 2008 là vì cơ sở dữ liệu của phần mềm chúng tôi không lớn và do giới hạn về cấu hình máy, giá trị kinh tế....

3. Thiết kế cơ sở dữ liệu Quản lý bán hàng

3.1 Xác định các bảng

Ví dụ, với cơ sở dữ liệu quản lý bán hàng của một cửa hàng thương mại X chúng ta có thể xác định các bảng sau:

BAN
CHAMCONG
ChiTetNhapXuat
CHUCVU
GHICONG
HANGHOA
Ka
KHACHHANG
Khuvc
LoaiHangHoa
NHACUNGCAP
NHANVIEN
PHAT
PHIEUXUATNHAP
QuyDoiDonVi
Quyen
THUONG
USERS

Với mỗi bảng cần chỉ rõ:

- + Khoá chính (có thể gồm một hay nhiều thuộc tính).
- + Mô tả của tất cả các cột (trường).

3.2 Xác định các thuộc tính của từng bảng

Việc xác định thuộc tính của từng bảng phụ thuộc vào loại hàng hóa và đặt thù của công ty, với ví dụ về cửa hàng thương mại X nào đó chúng ta có thể liệt kê các thuộc tính của từng bảng như sau:

NHÀ CUNG CẤP (mã NCC, tên NCC, địa chỉ, điện thoại, fax)

HÀNGHOA (mã hàng, tên hàng, ĐVT, mô tả, mã NCC)

PHIẾUXUẤTNHẬP (số phiếu, ngày, số cửa hàng)

CHITIẾTXUẤTNHẬP (mã hàng, số phiếu, số lượng)

4. Thiết kế các Table trong CSDL

4.1 Xác định kiểu dữ liệu từng trường của bảng

Với mỗi cột (trường) cần phải có:

- + Một tên duy nhất (trong bảng lưu giữ nó).
- + Một mô tả ngắn gọn.
- + Một kiểu dữ liệu (ví dụ: integer, char, date, logical,...) phụ thuộc vào hệ quản trị cơ sở dữ liệu cụ thể cài đặt cơ sở dữ liệu)
- + Một kích thước (mặc định hay chỉ rõ tùy từng kiểu dữ liệu)

4.2 Tạo bảng bằng công cụ SQL Management Studio

Với việc xác định các bảng như trên, chúng ta có thể sử dụng công cụ SQL Management Studio để tạo bảng như sau:

```
CREATE TABLE [dbo].[QuyDoiDonVi] (
    [MaDV] [int] NOT NULL,
    [DVT] [nvarchar](20) NULL,
    [Loai] [int] NULL,
    [GiaTri] [float] NULL,
    CONSTRAINT [pr_DVT] PRIMARY KEY CLUSTERED
(
    [MaDV] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[CHUCVU] (
    [MaCV] [int] IDENTITY(1,1) NOT NULL,
    [TenChucVu] [nvarchar](25) NOT NULL,
    [HeSoLuong] [float] NOT NULL,
    [ChiTietCongViec] [nvarchar](100) NULL,
    CONSTRAINT [pkCHUCVU] PRIMARY KEY CLUSTERED
(
    [MaCV] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[NHACUNGCAP] (
    [MaNCC] [int] IDENTITY(1,1) NOT NULL,
    [TenNCC] [nvarchar](50) NOT NULL,
    [DiaChi] [nvarchar](100) NULL,
    [GmailNCC] [nvarchar](50) NULL,
    [SoDienThoai] [nvarchar](20) NULL,
```

```

        [ChietKhau] [float] NULL,
    CONSTRAINT [pk_NHACUNGCAP] PRIMARY KEY CLUSTERED
(
    [MaNCC] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[LoaiHangHoa] (
    [Maloi] [int] IDENTITY(1,1) NOT NULL,
    [TenLoaiHang] [nvarchar](20) NOT NULL,
    [GhiChu] [nvarchar](50) NULL,
    CONSTRAINT [pk_LoiDonHang] PRIMARY KEY CLUSTERED
(
    [Maloi] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Khuvuc] (
    [MaKhuVuc] [int] IDENTITY(1,1) NOT NULL,
    [TenKhuVuc] [nvarchar](20) NULL,
    [PhiPhucVu] [int] NULL,
    CONSTRAINT [pk_KhuVuc] PRIMARY KEY CLUSTERED
(
    [MaKhuVuc] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[KHACHHANG] (
    [MaKH] [int] IDENTITY(1,1) NOT NULL,

```

```

        [TenKH] [nvarchar](50) NOT NULL,
        [Loai] [int] NOT NULL,
        [DiaChi] [nvarchar](100) NOT NULL,
    CONSTRAINT [PkKhachHang] PRIMARY KEY CLUSTERED
(
        [MaKH] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Ka] (
        [MaKa] [int] NOT NULL,
        [TenKa] [nvarchar](20) NULL,
        [ThoiGian] [int] NULL,
        [ThoiGianBD] [nvarchar](10) NULL,
        [ThoiGianKT] [nvarchar](10) NULL,
        [ThoiGianLam] [nvarchar](100) NULL,
    CONSTRAINT [pk_KA] PRIMARY KEY CLUSTERED
(
        [MaKa] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[HANGHOA] (
        [MaSP] [int] IDENTITY(1,1) NOT NULL,
        [MaNCC] [int] NOT NULL,
        [TenSP] [nvarchar](50) NOT NULL,
        [SoLuongTon] [int] NULL,
        [MaLoai] [int] NULL,
        [MaKT] [int] NULL,
        [DonViTinh] [int] NULL,
        [DonGia] [int] NULL,

```

```

CONSTRAINT [pk_hanghoa] PRIMARY KEY CLUSTERED
(
    [MaSP] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[NHANVIEN] (
    [MaNV] [int] IDENTITY(1,1) NOT NULL,
    [MaCV] [int] NOT NULL,
    [TenNV] [nvarchar](35) NOT NULL,
    [NgaySinh] [smalldatetime] NOT NULL,
    [Diachi] [nvarchar](50) NULL,
    [GioiTinh] [bit] NOT NULL,
    [SoDienThoai] [nvarchar](20) NULL,
    [NgayVao] [smalldatetime] NOT NULL,
    CONSTRAINT [pk_khahhang] PRIMARY KEY CLUSTERED
(
    [MaNV] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[BAN] (
    [MaBan] [int] IDENTITY(1,1) NOT NULL,
    [TenBan] [nvarchar](20) NOT NULL,
    [MaKV] [int] NULL,
    CONSTRAINT [PkBAN] PRIMARY KEY CLUSTERED
(
    [MaBan] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]

```

```

) ON [PRIMARY]
GO

CREATE TABLE [dbo].[THUONG] (
    [MaNV] [int] NOT NULL,
    [MaCV] [int] NOT NULL,
    [NgayThang] [datetime] NULL,
    [TienThuong] [int] NOT NULL,
    [GhiChu] [nvarchar](100) NULL,
    [TenThuong] [nvarchar](50) NULL,
    [id] [int] IDENTITY(1,1) NOT NULL,
PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[CHAMCONG] (
    [TongCa] [int] NULL,
    [MaNV] [int] NULL,
    [Nam] [int] NULL,
    [Thang] [int] NULL,
    [MaCC] [varchar](50) NOT NULL,
    [TienPhat] [int] NULL,
    [TienThuong] [int] NULL,
CONSTRAINT [p_ChamCong] PRIMARY KEY CLUSTERED
(
    [MaCC] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]

```

GO

```
CREATE TABLE [dbo].[PHIEUXUATNHAP] (
    [MaPhieu] [int] IDENTITY(1,1) NOT NULL,
    [MaNCC] [int] NULL,
    [MaNV] [int] NOT NULL,
    [NgayThang] [smalldatetime] NOT NULL,
    [NoCo] [bit] NULL,
    [loai] [bit] NULL,
    [MaBan] [int] NULL,
    [MaKH] [int] NULL,
    [ThoiGianRA] [datetime] NULL,
    [TongTien] [float] NULL,
    [Khoa] [bit] NULL,
    [MaNV1] [nvarchar](50) NULL,
    CONSTRAINT [phieunhapxuat] PRIMARY KEY CLUSTERED
(
    [MaPhieu] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
```

GO

```
CREATE TABLE [dbo].[PHAT] (
    [MaNV] [int] NOT NULL,
    [MaCV] [int] NOT NULL,
    [NgayThang] [datetime] NULL,
    [TenPhat] [nvarchar](20) NULL,
    [Ghichu] [nvarchar](100) NULL,
    [TienPhat] [int] NULL,
    [id] [int] IDENTITY(1,1) NOT NULL,
    PRIMARY KEY CLUSTERED
(
    [id] ASC
```

```

) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[ChiTetNhapXuat] (
    [MaPhieu] [int] NOT NULL,
    [SoThuTu] [int] NOT NULL,
    [MaSP] [int] NULL,
    [SoLuong] [int] NULL,
    [DonGiaNhap] [int] NULL,
    [ThanhTien] [int] NULL,
    [DVT] [int] NULL,
    [SoLuongQT] [int] NULL,
    CONSTRAINT [pk_ChiTetNhapXuat] PRIMARY KEY CLUSTERED
(
    [MaPhieu] ASC,
    [SoThuTu] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[GHICONG] (
    [MaNV] [int] NOT NULL,
    [N1] [bit] NULL,
    [N2] [bit] NULL,
    [N3] [bit] NULL,
    [N4] [bit] NULL,
    [N5] [bit] NULL,
    [N6] [bit] NULL,
    [N7] [bit] NULL,
    [N8] [bit] NULL,
    [N9] [bit] NULL,
    [N10] [bit] NULL,

```

```

[N11] [bit] NULL,
[N12] [bit] NULL,
[N13] [bit] NULL,
[N14] [bit] NULL,
[N15] [bit] NULL,
[N16] [bit] NULL,
[N17] [bit] NULL,
[N18] [bit] NULL,
[N19] [bit] NULL,
[N21] [bit] NULL,
[N20] [bit] NULL,
[N22] [bit] NULL,
[N23] [bit] NULL,
[N24] [bit] NULL,
[N25] [bit] NULL,
[N26] [bit] NULL,
[N27] [bit] NULL,
[N28] [bit] NULL,
[N29] [bit] NULL,
[N30] [bit] NULL,
[N31] [bit] NULL,
[KA] [int] NOT NULL,
[MaCC] [varchar](50) NOT NULL,
CONSTRAINT [pk_GhiCong] PRIMARY KEY CLUSTERED
(
    [MaCC] ASC,
    [KA] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

5. Tạo liên kết cho các Table trong CSDL

5.1 Xác định khóa chính, khóa ngoại

Các quan hệ giữa các bảng sẽ được thiết lập bởi các khoá ngoại, kiểm tra để các ràng buộc khoá ngoại (nếu có) sẽ được cài đặt.

Phải đảm bảo rằng giữ trường khóa chính và khóa ngoại phải cùng kiểu dữ liệu và kích thước thì chúng ta mới thiết lập quan hệ cho bước sau được.

5.2 Thiết lập quan hệ cho các bảng bằng giao diện SQL Management Studio

```
/****** Object:  Default [DF__HANGHOA__SoLuong__489AC854]
*****/
ALTER TABLE [dbo].[HANGHOA] ADD DEFAULT ((0)) FOR [SoLuongTon]
GO
/****** Object:  Default [DF__KHACHHANG__Loai__498EEC8D]
*****/
ALTER TABLE [dbo].[KHACHHANG] ADD DEFAULT ((5)) FOR [Loai]
GO
/****** Object:  Default [DF__THUONG__NgayThan__4A8310C6]
*****/
ALTER TABLE [dbo].[THUONG] ADD DEFAULT (getdate()) FOR
[NgayThang]
GO
/****** Object:  Check [rbBanBaoGia_SoLuongTon] *****/
ALTER TABLE [dbo].[HANGHOA] WITH CHECK ADD CONSTRAINT
[rbBanBaoGia_SoLuongTon] CHECK (([SoLuongTon]>=(0)))
GO
ALTER TABLE [dbo].[HANGHOA] CHECK CONSTRAINT
[rbBanBaoGia_SoLuongTon]
GO
/****** Object:  ForeignKey [pk_ban_KhuVuc] *****/
ALTER TABLE [dbo].[BAN] WITH CHECK ADD CONSTRAINT
[pk_ban_KhuVuc] FOREIGN KEY ([MaKV])
REFERENCES [dbo].[KhuVuc] ([MaKhuVuc])
GO
ALTER TABLE [dbo].[BAN] CHECK CONSTRAINT [pk_ban_KhuVuc]
GO
```

```

/***** Object:  ForeignKey [fk_nv_CC]          *****/
ALTER TABLE [dbo].[CHAMCONG]  WITH CHECK ADD  CONSTRAINT
[fk_nv_CC] FOREIGN KEY([MaNV])
REFERENCES [dbo].[NHANVIEN] ([MaNV])
GO
ALTER TABLE [dbo].[CHAMCONG] CHECK CONSTRAINT [fk_nv_CC]
GO
/***** Object:  ForeignKey [fk_ctPhieu_dvt]      *****/
ALTER TABLE [dbo].[ChiTetNhapXuat]  WITH CHECK ADD  CONSTRAINT
[fk_ctPhieu_dvt] FOREIGN KEY([DVT])
REFERENCES [dbo].[QuyDoiDonVi] ([MaDV])
GO
ALTER TABLE [dbo].[ChiTetNhapXuat] CHECK CONSTRAINT
[fk_ctPhieu_dvt]
GO
/***** Object:  ForeignKey [fk_ctPhieu_hanghoa]    *****/
ALTER TABLE [dbo].[ChiTetNhapXuat]  WITH CHECK ADD  CONSTRAINT
[fk_ctPhieu_hanghoa] FOREIGN KEY([MaSP])
REFERENCES [dbo].[HANGHOA] ([MaSP])
GO
ALTER TABLE [dbo].[ChiTetNhapXuat] CHECK CONSTRAINT
[fk_ctPhieu_hanghoa]
GO
/***** Object:  ForeignKey [fk_ctPhieu_phieu]      *****/
ALTER TABLE [dbo].[ChiTetNhapXuat]  WITH CHECK ADD  CONSTRAINT
[fk_ctPhieu_phieu] FOREIGN KEY([MaPhieu])
REFERENCES [dbo].[PHIEUXUATNHAP] ([MaPhieu])
GO
ALTER TABLE [dbo].[ChiTetNhapXuat] CHECK CONSTRAINT
[fk_ctPhieu_phieu]
GO
/***** Object:  ForeignKey [f_gc_NHANVIEN]        *****/
ALTER TABLE [dbo].[GHICONG]  WITH CHECK ADD  CONSTRAINT
[f_gc_NHANVIEN] FOREIGN KEY([MaNV])
REFERENCES [dbo].[NHANVIEN] ([MaNV])

```

```

GO
ALTER TABLE [dbo].[GHICONG] CHECK CONSTRAINT [f_gc_NHANVIEN]
GO
/***** Object: ForeignKey [fk_GHICONG_CHAMCONG] *****/
ALTER TABLE [dbo].[GHICONG] WITH CHECK ADD CONSTRAINT
[fk_GHICONG_CHAMCONG] FOREIGN KEY ([MaCC])
REFERENCES [dbo].[CHAMCONG] ([MaCC])
GO
ALTER TABLE [dbo].[GHICONG] CHECK CONSTRAINT
[fk_GHICONG_CHAMCONG]
GO
/***** Object: ForeignKey [pr_ghicong_ka] *****/
ALTER TABLE [dbo].[GHICONG] WITH CHECK ADD CONSTRAINT
[pr_ghicong_ka] FOREIGN KEY ([KA])
REFERENCES [dbo].[Ka] ([MaKa])
GO
ALTER TABLE [dbo].[GHICONG] CHECK CONSTRAINT [pr_ghicong_ka]
GO
/***** Object: ForeignKey [f_NCC_HANGHOA] *****/
ALTER TABLE [dbo].[HANGHOA] WITH CHECK ADD CONSTRAINT
[f_NCC_HANGHOA] FOREIGN KEY ([MaNCC])
REFERENCES [dbo].[NHACUNGCAP] ([MaNCC])
ON UPDATE CASCADE
GO
ALTER TABLE [dbo].[HANGHOA] CHECK CONSTRAINT [f_NCC_HANGHOA]
GO
/***** Object: ForeignKey [fr_HangHoa_DVT] *****/
ALTER TABLE [dbo].[HANGHOA] WITH CHECK ADD CONSTRAINT
[fr_HangHoa_DVT] FOREIGN KEY ([DonViTinh])
REFERENCES [dbo].[QuyDoiDonVi] ([MaDV])
GO
ALTER TABLE [dbo].[HANGHOA] CHECK CONSTRAINT [fr_HangHoa_DVT]
GO
/***** Object: ForeignKey [pr_HangHoa_LoiHangHoa] *****/

```

```

ALTER TABLE [dbo].[HANGHOA] WITH CHECK ADD CONSTRAINT
[pr_HangHoa_LoaiHangHoa] FOREIGN KEY ([MaLoai])
REFERENCES [dbo].[LoaiHangHoa] ([Maloi])
GO

ALTER TABLE [dbo].[HANGHOA] CHECK CONSTRAINT
[pr_HangHoa_LoaiHangHoa]
GO

/***** Object: ForeignKey [rNHANVIEN] *****/
ALTER TABLE [dbo].[NHANVIEN] WITH CHECK ADD CONSTRAINT
[rNHANVIEN] FOREIGN KEY ([MaCV])
REFERENCES [dbo].[CHUCVU] ([MaCV])
ON UPDATE CASCADE
GO

ALTER TABLE [dbo].[NHANVIEN] CHECK CONSTRAINT [rNHANVIEN]
GO

/***** Object: ForeignKey [fk_Phlat_CV] *****/
ALTER TABLE [dbo].[PHAT] WITH CHECK ADD CONSTRAINT
[fk_Phlat_CV] FOREIGN KEY ([MaCV])
REFERENCES [dbo].[CHUCVU] ([MaCV])
GO

ALTER TABLE [dbo].[PHAT] CHECK CONSTRAINT [fk_Phlat_CV]
GO

/***** Object: ForeignKey [fk_Phlat_NV] *****/
ALTER TABLE [dbo].[PHAT] WITH CHECK ADD CONSTRAINT
[fk_Phlat_NV] FOREIGN KEY ([MaNV])
REFERENCES [dbo].[NHANVIEN] ([MaNV])
GO

ALTER TABLE [dbo].[PHAT] CHECK CONSTRAINT [fk_Phlat_NV]
GO

/***** Object: ForeignKey [fk_Phieu_nv] *****/
ALTER TABLE [dbo].[PHIEUXUATNHAP] WITH CHECK ADD CONSTRAINT
[fk_Phieu_nv] FOREIGN KEY ([MaNV])
REFERENCES [dbo].[NHANVIEN] ([MaNV])
GO

ALTER TABLE [dbo].[PHIEUXUATNHAP] CHECK CONSTRAINT [fk_Phieu_nv]

```

```

GO

/***** Object:  ForeignKey [fTHUONG_CHUCVU]          *****/
ALTER TABLE [dbo].[THUONG]  WITH CHECK ADD  CONSTRAINT
[fTHUONG_CHUCVU] FOREIGN KEY ([MaCV])
REFERENCES [dbo].[CHUCVU] ([MaCV])
GO

ALTER TABLE [dbo].[THUONG] CHECK CONSTRAINT [fTHUONG_CHUCVU]
GO

/***** Object:  ForeignKey [fTHUONG_NHANVIEN]        *****/
ALTER TABLE [dbo].[THUONG]  WITH CHECK ADD  CONSTRAINT
[fTHUONG_NHANVIEN] FOREIGN KEY ([MaNV])
REFERENCES [dbo].[NHANVIEN] ([MaNV])
GO

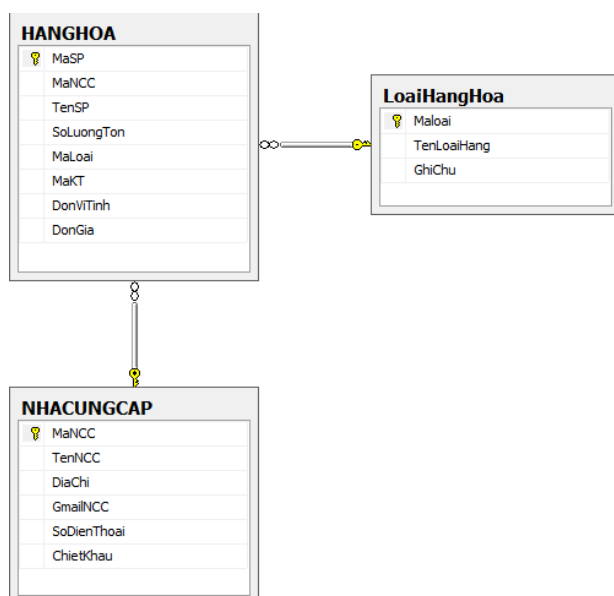
ALTER TABLE [dbo].[THUONG] CHECK CONSTRAINT [fTHUONG_NHANVIEN]
GO

```

6. Nhập dữ liệu mẫu cho các Table

6.1 Nhập dữ liệu mẫu cho các bảng bên quan hệ 1

Để nhập dữ liệu cho các bảng, chúng ta phải nhập dữ liệu cho các bảng bên quan hệ 1 trước. Ví dụ với các bảng có quan hệ sau:



Chúng ta phải nhập dữ liệu cho bảng LoaiHangHoa và bảng NhaCungCap trước, sau đó mới có thể nhập dữ liệu cho bảng HangHoa.

Khi nhập dữ liệu cho các bảng cần tránh sự trùng lặp về khóa chính.

```

INSERT [dbo].[NHACUNGCAP] ([MaNCC], [TenNCC], [DiaChi],
[GmailNCC], [SoDienThoai], [ChietKhau]) VALUES (1, N'Siêu thị
Bic C', N'Hồ Tùng Mậu', N'bicC@yahoo.com', N'0633123456', 0)
INSERT [dbo].[NHACUNGCAP] ([MaNCC], [TenNCC], [DiaChi],
[GmailNCC], [SoDienThoai], [ChietKhau]) VALUES (2, N'Chị Hoa',
N'Chợ Đà Lạt', N'hoathi@gmail.com', N'0988123321', 0)
GO
INSERT [dbo].[LoaiHangHoa] ([Maloi], [TenLoaiHang], [GhiChu])
VALUES (1, N'Ăn Sáng', N'Các món ăn và nước uống buổi sáng')
INSERT [dbo].[LoaiHangHoa] ([Maloi], [TenLoaiHang], [GhiChu])
VALUES (2, N'Ăn trưa', N'Các món ăn trưa')

```

6.2 Nhập dữ liệu mẫu cho các bảng bên quan hệ n

Như đã trình bày ở trên, khi nhập dữ liệu cho các bảng bên quan hệ bên n (bên nhiều), ngoài việc bảo toàn về sự không trùng lặp về khóa chính còn phải đảm bảo khóa ngoại phải tồn tại.

Ví dụ:

```

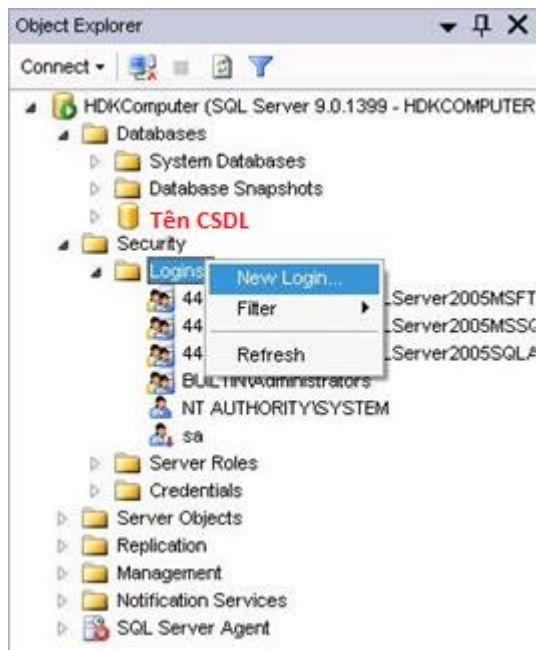
INSERT [dbo].[HANGHOA] ([MaSP], [MaNCC], [TenSP], [SoLuongTon],
[MaLoai], [MaKT], [DonViTinh], [DonGia]) VALUES (2, 1, N'Nước
Cocacola', 100, 1, NULL, N'1', 10000)
INSERT [dbo].[HANGHOA] ([MaSP], [MaNCC], [TenSP], [SoLuongTon],
[MaLoai], [MaKT], [DonViTinh], [DonGia]) VALUES (4, 1, N'Nước
Pesi', 50, 2, NULL, N'3', 5000)

```

7. Cài đặt cơ sở dữ liệu Quản lý bán hàng

7.1 Tạo user đăng nhập cơ sở dữ liệu

+ Vào Security, nhấp chuột phải lên Logins → chọn New login :



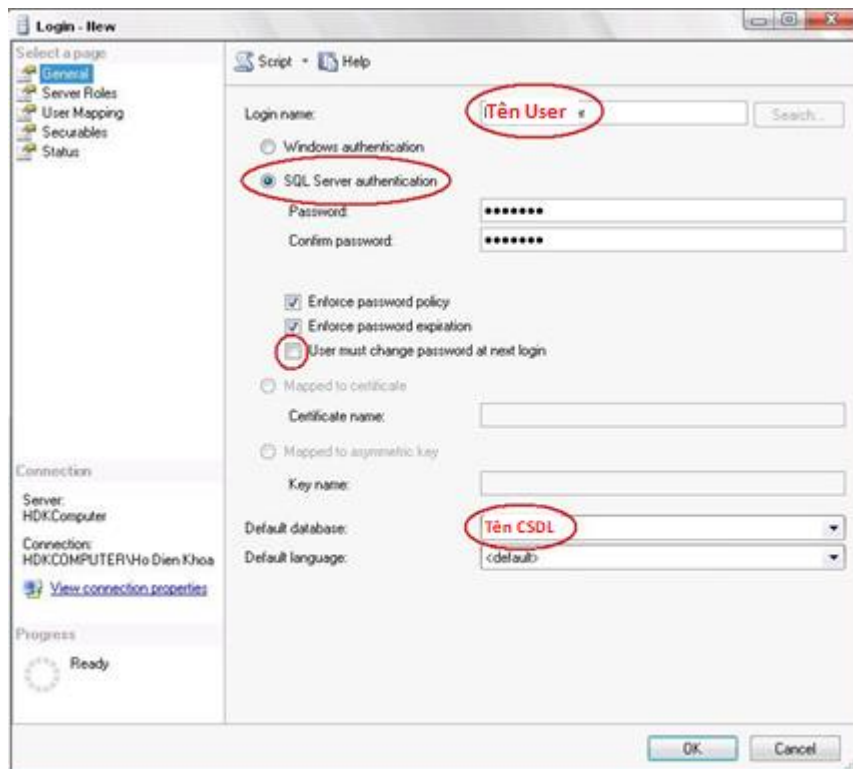
+ Trong tab “**General**”:

Login name: tên tài khoản đăng nhập. Trong VD này chúng ta chọn “**Tên User**”.

Chọn **SQL Sever authenticaton**. Nhập password của tài khoản & confirm password.

Bỏ lựa chọn “User must change password at next login”

Default database: chọn CSDL chúng ta vừa tạo là “**Tên CSDL**”

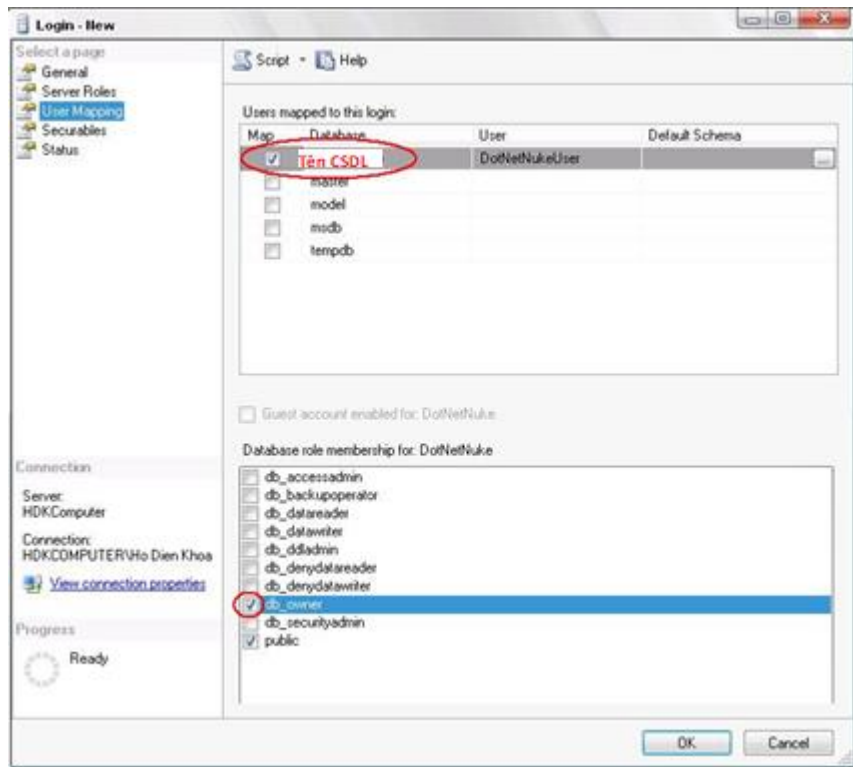


7.2 Cấp quyền user sử dụng cơ sở dữ liệu

+ Chuyển qua tab “User Mapping”:

Trong khung “Users mapped to this login”, check “Tên CSDL”

Trong khung “Database role membership for: Tên CSDL”, check “db_owner”



* **Lưu ý:** Phần “Authentication” phải chọn “SQL Server and Windows”

Nhấn phải chuột vào computer, chọn **Properties**

Chọn tab “**Security**”

Trong khung Security, chọn “SQL Server and Windows”

Chọn **OK**.

7.3 Lưu trữ và quản lý cơ sở dữ liệu quản lý bán hàng

Loại 1: Full backup

Loại backup này sẽ backup dữ liệu đầy đủ nhất, vì vậy máy chủ sẽ mất nhiều thời gian để thực hiện nếu database lớn

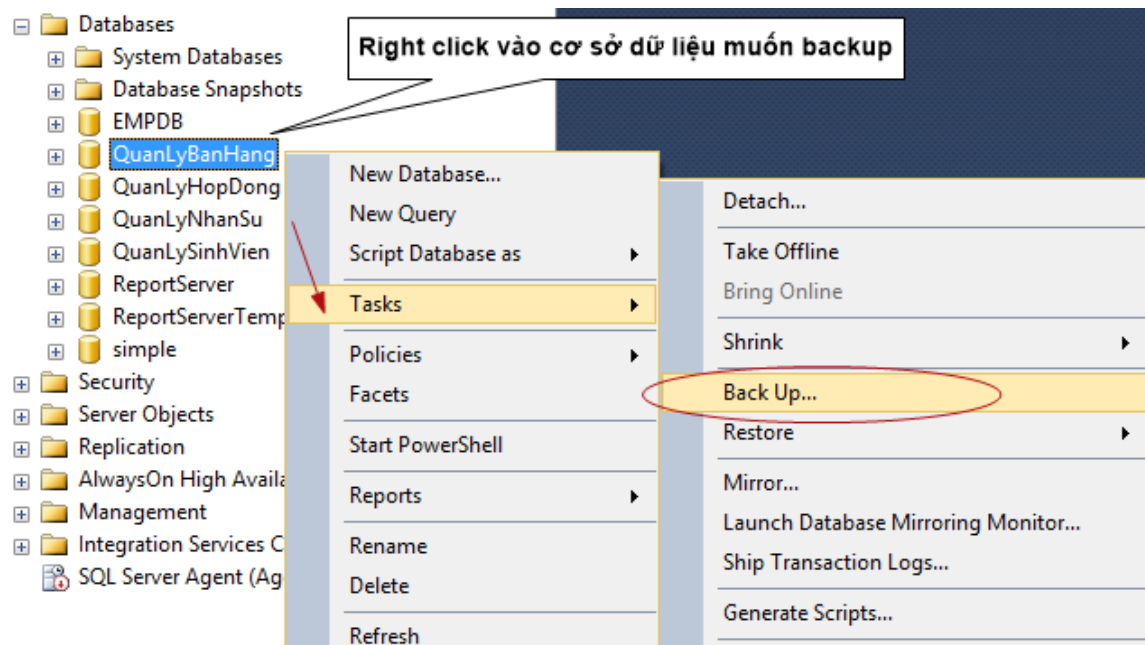
Cú pháp

```
backup database Tên_Cơ_Sở_Dữ_Liệu to disk =  
'Đường_dẫn\Tên_file.bak'
```

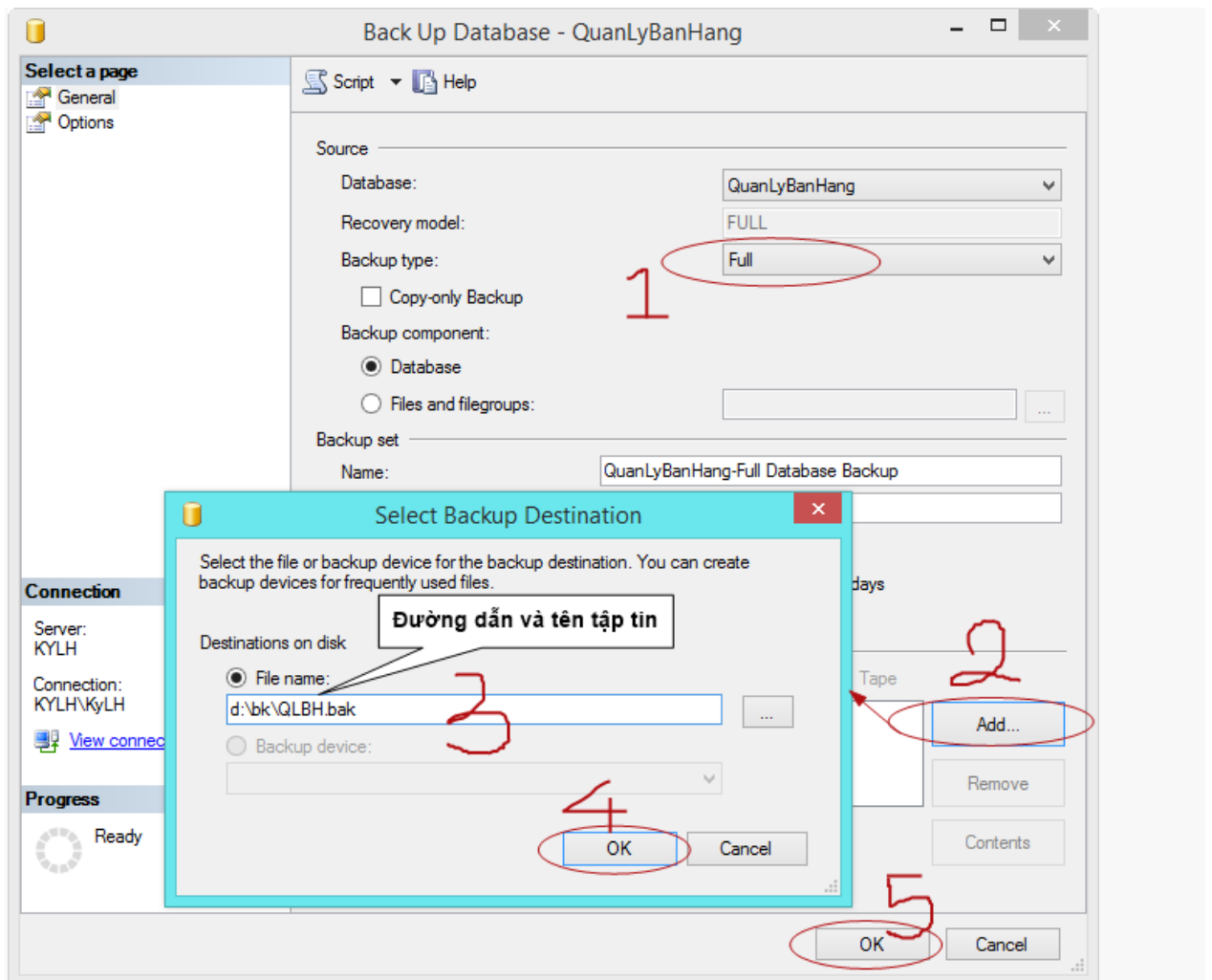
Ví dụ câu lệnh **full backup** cơ sở dữ liệu **QuanLyBanHang** và lưu vào “**d:\bk**” (Lưu ý chúng ta phải tạo thư mục bk trong ổ đĩa D trước khi thực thi câu lệnh sau):

```
backup database QuanLyBanHang to disk = 'd:\bk\QLBH.bak'
```

Thực hiện full backup bằng giao diện



Chọn full backup, tên tập tin



Loại 2: Different backup

Backup những dữ liệu phát sinh tính từ lần backup full gần nhất. Như vậy trước khi chúng ta thực hiện different backup thì full backup phải được thực hiện trước. Chính vì vậy, khi chúng ta sử dụng loại backup này sẽ tiết kiệm được thời gian backup dữ liệu.

Cú pháp

```
backup database Tên_Cơ_Sở_Dữ_Liệu to disk =  
'Đường_dẫn\Tên_File.bak' with differential
```

Ví dụ

```
backup database QuanLyBanHang to disk = 'D:\sql\QLBH_Diff.bak'  
with differential
```

Thực hiện different backup bằng giao diện

Source

Database:	QuanLyBanHang
Recovery model:	FULL
Backup type:	Differential

Loại 3: Log backup

Backup dữ liệu sử dụng tập tin log và log backup được sử dụng để backup dữ liệu tại thời điểm cụ thể ví dụ như mỗi giờ trong ngày.

Cú pháp

```
backup log Tên_Cơ_Sở_Dữ_Liệu to disk = 'Đường_dẫn\Tên_File.trn'
```

Ví dụ

```
backup log QuanLyBanHang to disk = 'D:\sql\QLBH.trn'
```

Thực hiện log backup bằng giao diện

Source

Database:	QuanLyBanHang
Recovery model:	FULL
Backup type:	Transaction Log

BÀI 2: THIẾT KẾ CÁC CHỨC NĂNG CHÍNH

Mã Bài: MD24_02

Giới thiệu:

Để xây dựng cho ứng dụng quản lý bán hàng nói chung rất đa dạng, trong phạm vi bài này chúng tôi chỉ xây dựng ứng dụng nhỏ để quản lý quán cafe.

Mục tiêu:

- Trình bày được một số cách thiết kế giao diện;
- Trình bày được giải thuật để thực hiện các chức năng của ứng dụng;
- Xây dựng được các giao diện thân thiện phù hợp công việc của từng chức năng phần mềm;
- Cài đặt được các giải thuật xử lý trong ứng dụng;
- Viết mã lệnh rõ ràng, sáng sủa, dễ đọc, dễ hiểu;
- Rèn luyện tính tích cực sáng tạo, tinh thần làm việc độc lập và hợp tác cao.

Nội dung chính:

1. Tạo project và kết nối cơ sở dữ liệu

1.1 Thiết lập chuỗi kết nối trong Properties của project

Trong App.config

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
  </configSections>
  <connectionStrings>
    <add name="QLCF.Properties.Settings.ChuoiKetNoi"
connectionString="Data Source=.;Initial Catalog=qlcf;User ID=sa"
      providerName="System.Data.SqlClient" />
  </connectionStrings>
</configuration>
```

1.2 Khai báo các đối tượng dùng chung ADO.NET

Đối với ngôn ngữ C#, việc khai báo các đối tượng ADO.NET dùng chung phải khai báo dạng thành phần static (tĩnh) của lớp.

```
using System.Data;
using System.Data.SqlClient;
```

```
namespace QLCF
{
    class Class1
    {
        static SqlCommand lenh = new SqlCommand();
        static SqlDataAdapter da = new SqlDataAdapter();
    }
}
```

1.3 Khởi tạo kết nối đến cơ sở dữ liệu

Để thuận lợi cho việc khởi tạo kết nối đến cơ sở dữ liệu, trước tiên là khai báo chuỗi:

```
string connStr = Properties.Settings.Default.ChuoiKetNoi;
```

Sau đó chúng ta có thể sử dụng phương thức tĩnh của lớp để khởi tạo kết nối một lần duy nhất đến cơ sở dữ liệu:

```
static private SqlConnection _ketnoi
                                = new SqlConnection(connStr);
static public SqlConnection ketnoi()
{
    if (_ketnoi.State == ConnectionState.Closed)
        _ketnoi.Open();
    return _ketnoi;
}
```

Hoặc chúng ta có thể không thông qua biến chuỗi, code hoàn thiện của lớp kết nối cơ sở dữ liệu như sau:

```
using System.Data;
using System.Data.SqlClient;
namespace QLCF
{
    class Class1
    {
        static SqlCommand lenh = new SqlCommand();
        static SqlDataAdapter da = new SqlDataAdapter();
    }
}
```

```

static private SqlConnection _ketnoi = new
SqlConnection(Properties.Settings.Default.ChuoiKetNoi );
static public SqlConnection ketnoi()
{
    if (_ketnoi.State == ConnectionState.Closed)
        _ketnoi.Open();
    return _ketnoi;
}
}
}

```

2. Thiết kế chức năng Cập nhật dữ liệu

2.1 Thiết kế form cập nhật dữ liệu cho Hàng hóa

Mã Sản Phẩm	Tên Sản Phẩm	Số Lượng Tồn	Đơn Vị Tính	Giá Bán
57	Bia Heineken 330ml	158	Chai	9165
58	Sapporo lon 330ml	60	Chai	0
59	Bia Tiger 320ml	0	Chai	0

Giao diện xét tồn cập nhập giá bán

Gridcontrol: hiển thị thông tin hàng hóa

Nút chức năng:

- + Cập nhập giá bán tự động: cập nhập giá bán theo giá nhập nhân với tỉ lệ giá
- + In DS: in danh sách các mặt hàng hiện có trên gridcontrol
- + Làm tươi: tải lại danh ds các mặt hàng
- + Lưu: lưu lại các thông tin đã chỉnh sửa trên gridcontrol
- + Thoát: thoát form

2.2 Thiết kế form cập nhật dữ liệu cho khách hàng

Mã Khách Hàng	Tên Khách Hàng	Giảm Giá	Địa Chỉ
1	Vip 1	4	-----
2	Vip 2	4	-----

Giao diện quản lý thông tin nhóm khách hàng

Gridcontrol: hiển thị thông tin nhóm khách hàng

Nút chức năng:

- + Làm tươi: tải lại danh sách các khách hàng

- + Thêm mới: thêm một dòng trống để nhập thông tin nhóm khách hàng
- + Lưu: lưu thông tin đã chỉnh sửa trên gridcontrol
- + Xóa: xóa một dòng đang chọn trên gridcontrol
- + Thoát: thoát from

2.3 Thiết kế form cập nhật dữ liệu cho Phiếu thu

+ Khoản Thu: Ghi chép lại cụ thể mình đã thu những khoản nào từ tài khoản nào và vào lúc nào.

Chi tiết Khoản Thu			
Tài khoản	Công quỹ	Số tiền	500,000.00
Khoản thu	Ứng hộ	Ngày thu	30/03/2015
Người được Thu	Nguyễn Văn Anh		
Ghi chú			
		 Thêm mới	

Phần này xem như một phần bài tập thực hành cho sinh viên.

Trong phần này, sinh viên cần in ra phiếu thu dạng như mẫu sau:

PHIẾU THU					Số: 1
Ngày 08 tháng 03 năm 2015					
Họ và tên:	Nguyễn Văn A				
Địa chỉ:	Quảng Nam				
Lý do nộp:	Photo, chứng thực				
Số tiền:	240,000.00 đ				
Bằng chữ:	Hai trăm bốn mươi nghìn đồng/				
Kèm theo:	chứng từ gốc				
					Ngày 08 tháng 03 năm 2015
Trưởng Nhóm	Kế toán trưởng	Người nộp tiền	Người lập phiếu	Thủ quỹ	
(Ký, họ tên, đóng dấu)	(Ký, họ tên)	(Ký, họ tên)	(Ký, họ tên)	(Ký, họ tên)	
Freelance Programmer	Kế Toán	Nguyễn Văn A	Người lập phiếu	Thủ Quỹ	
Đã nhận đủ số tiền (viết bằng chữ): Hai trăm bốn mươi nghìn đồng/					

2.4 Thiết kế form cập nhật dữ liệu cho Phiếu chi

Tương tự như phiếu thu, sinh viên tự thiết kế giao diện cho phiếu chi để tác nghiệp dựa vào kiến thức đã học. Kết quả cuối cùng sinh viên phải in ra được mẫu phiếu chi dạng như sau:

PHIẾU CHI		Số: 1
Ngày 08 tháng 03 năm 2015		
Họ và tên:	Trần Thị Em	
Địa chỉ:		
Lý do chi:	Tạm ứng lương tháng 03	
Số tiền:	1,500,000.00 đ	
Bằng chữ:	Một triệu năm trăm nghìn đồng./	
Kèm theo:	chứng từ gốc	
Trưởng Nhóm (Ký, họ tên, đóng dấu)		Người nhận tiền (Ký, họ tên)
		Thủ quỹ (Ký, họ tên)
Freelance Programmer		Trần Thị Em
		Thủ Quỹ
Đã nhận đủ số tiền (viết bằng chữ): Một triệu năm trăm nghìn đồng./		

2.5 Thiết kế form cập nhật dữ liệu cho Hóa đơn

Giao diện có dạng như sau:

The screenshot displays a software interface for managing restaurant bills. At the top, there are dropdown menus for 'Khu Vực' (Area) and 'Ban Số' (Table Number), along with a 'Chuyển Sang Bàn' (Transfer to Table) button. Below this, a tab control shows various areas: 'khu vực tang 1', 'khu vực tang 2', 'khu vực tang 3', 'khu vực tầng trệt', 'khu vực sân vườn', and 'dddd'. The 'khu vực tang 1' tab is active, showing a list of items like 'Ăn Sáng', 'Ăn Trưa', 'Ăn Tối', 'Nước ngọt', etc. A 'Giao diện bán hàng' (Sales Interface) window is open, displaying a 'Thông tin phiếu tính tiền' (Billing Information) section with fields for 'Bản Số', 'Ngày', 'Khu Vực', 'Giờ Vào', 'Số Phiếu', 'Giờ Ra', 'NV Phục Vụ', and 'Khách Hàng'. Below this is a table for items with columns for 'stt', 'Hàng Hóa', 'số Lượng', 'Đơn Vị', 'Đơn Giá', and 'Thành Tiền'. The table contains one row: '1 Bia Heineken 330ml', '1', 'Két(20)', '280000', and '280000'. At the bottom, there are fields for 'Tổng Tiền' (Total Amount), 'Giảm Giá' (Discount), 'Phí Phục Vụ' (Service Fee), and 'Thuế' (Tax), along with buttons for 'In Bếp', 'In Phiếu Tính Tiền', 'Tính Tiền', 'Làm Mới', and 'Thoát'.

Giao diện bán hàng

Tabcontrol: hiển thị chi tiết từng khu vực từng bàn

Nut chức năng: chuyển bàn (khi khách có nhu cầu chuyển bàn)

+ Giao diện lập phiếu:

Gridcontrol: hiển thị thông tin chi tiết phiếu xuất

Nút chức năng:

Thêm(+): thêm chi tiết phiếu xuất

Xóa(-): xóa chi tiết phiếu xuất

In bếp: in danh sách các món khách gọi xuống bếp chế biến

In phiếu tính tiền: liệt kê chi tiết mọi khoản thu chi tiết in phiếu tính tiền theo mẫu:

No Name

Địa Chỉ : ???????????????????/

PHIẾU TÍNH TIỀN

Số Phiếu : 36

Ngày Lập : 22/06/2013 Khu Vực : khu vực tầng 1

Thu Ngân : Duc Cong Bàn : Bàn Số 2

STT	Món	Đơn Giá	SL	Thành Tiền
1	Cafe Sữa Đá	12.000	1	12.000
2	Cafe Đen Nóng	8.000	1	8.000

Tổng Tiền : 20000

Giảm : 800

Phí Phục Vụ : 9000

Thuế : 2820

Thực Trả : 31020

Tính tiền: giải phóng trạng thái có khách cho bàn (thể hiện phiếu đã tính tiền)

2.6 Tạo các form giao diện tương ứng của từng chức năng

* Kho hàng

Thông Tin Hàng Hóa

Mã Sản Phẩm: 000 Số Lượng Tôn: 000

Nhà cung cấp: Nhà Cung Cấp Tổng Hợp Mã Menu: Ẩn Sáng

Tên Sản Phẩm: Kiểm Tồn: SP Không Qua Chế Biến

Đơn Vị Tính: Két(20)

Drag a column header here to group by that column

Mã Sản Phẩm	Nhà Cung Cấp	Tên Sản Phẩm	Số Lượng Tôn	Menu	Kiểm Tồn	Đơn Vị Tính
57	Nhà Cung Cấp Tổng...	Bia Heineken 330ml	158	Bia	SP Không Qua Chế Biến	Chai
58	Nhà Cung Cấp Tổng...	Sapporo lon 330ml	60	Bia	SP Không Qua Chế Biến	Chai
59	Nhà Cung Cấp Tổng...	Bia Tiger 320ml	0	Bia	SP Không Qua Chế Biến	Chai
60	Nhà Cung Cấp Tổng...	Bia 333 lon 330ml	60	Bia	SP Không Qua Chế Biến	Chai
61	Nhà Cung Cấp Tổng...					
62	Nhà Cung Cấp Tổng...					
63	Nhà Cung Cấp Tổng...					
64	Nhà Cung Cấp Tổng...					
65	Nhà Cung Cấp Tổng...					
66	Nhà Cung Cấp Tổng...					
67	Nhà Cung Cấp Tổng...					
68	Nhà Cung Cấp Tổng...					

Xét tồn, quản lý đơn giá

Drag a column header Tỷ lệ giá: 150 % Cập nhật giá tự động In DS SP Không Qua Chế Biến Làm tươi Lưu Thoát

Mã Sản Phẩm	Tên Sản Phẩm	Số Lượng Tôn	Đơn Vị Tính	Giá Bán
57	Bia Heineken 330ml	158	Chai	9165
58	Sapporo lon 330ml	60	Chai	0
59	Bia Tiger 320ml	0	Chai	0
60	Bia 333 lon 330ml	60	Chai	9166

Gridcontrol: hiển thị thông tin hàng hóa, số lượng tồn, loại menu, kiểm tồn, DVT

Nút chức năng:

- + Thêm: thêm hàng hóa vào kho hàng
- + Cập nhập: cập nhập thông tin hàng hóa trong kho
- + Xóa: xóa một mặt hàng hiện có trong kho hàng

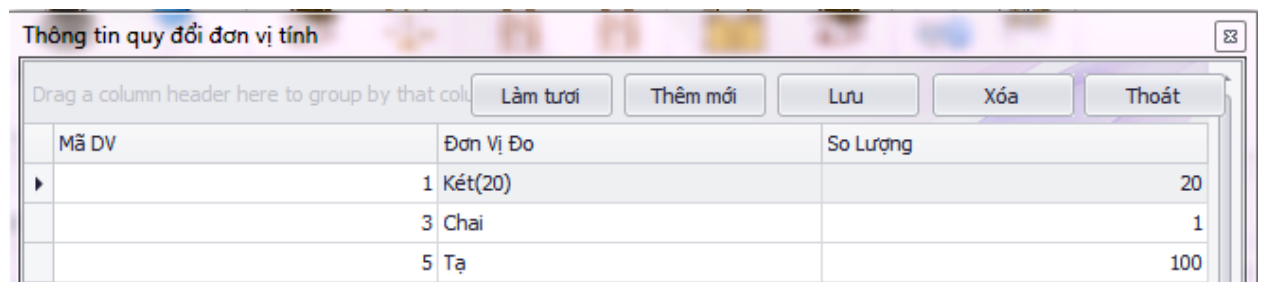
Xét tồn, cập nhập giá:

Gridcontrol: hiển thị thông tin hàng hóa

Nút chức năng:

- + Cập nhập giá bán tự động: cập nhập giá bán theo giá nhập nhân với tỉ lệ giá
- + In DS: in danh sách các mặt hàng hiện có trên gridcontrol
- + Làm tươi: tải lại danh ds các mặt hàng
- + Lưu: lưu lại các thông tin đã chỉnh sửa trên gridcontrol
- + Thoát: thoát form

* Giao diện đơn vị tính



Mã DV	Đơn Vị Đo	Số Lượng
1	Kết(20)	20
3	Chai	1
5	Tạ	100

Gridcontrol: hiển thị thông tin đơn vị tính mã đơn vị tính, đơn vị đo (tên đơn vị), số lượng(số lượng quy đổi thành đơn vị nhỏ nhất)

Nút chức năng:

- + Làm tươi: tải lại danh sách đơn vị tính
- + Thêm mới: thêm một dòng trống để nhập thông tin đơn vị đo
- + Lưu: lưu thông tin đã chỉnh sửa trên gridcontrol
- + Xóa: xóa một dòng đang chọn trên gridcontrol
- + Thoát: thoát from

* Giao diện quản lý thông tin nhà cung cấp

Nhà cung cấp				
Drag a column header here to group by that column				
			Làm tươi	thêm mới
			Lưu	Xóa
				Thoát
Mã NCC	Tên Nhà Cung Cấp	Địa Chỉ	Gmail	Số Điện Thoại
1	Nhà Cung Cấp Tổng Hợp	số 1 hoàng văn thụ F4 TP đà lạt	cocacola.com	0123456789
2	Công Ty TNHH pepsi	số 1 hoàng văn thụ F4 TP đà lạt	pepsi.com	0123456789

Gridcontrol: hiển thị thông tin nhà cung cấp

Nút chức năng:

- + Làm tươi: tải lại danh sách các nhà cung cấp
- + Thêm mới: thêm một dòng trống để nhập thông tin nhà cung cấp
- + Lưu: lưu thông tin đã chỉnh sửa trên gridcontrol
- + Xóa: xóa một dòng đang chọn trên gridcontrol
- + Thoát: thoát from

* Giao diện quản lý thông tin nhóm khách hàng

Khách hàng			
Drag a column header here to group by that column			
		Làm tươi	thêm mới
		Lưu	Xóa
			Thoát
Mã Khách Hàng	Tên Khách Hàng	Giảm Giá	Địa Chỉ
1	Vip 1	4	-----
2	Vip 2	4	-----

Gridcontrol: hiển thị thông tin nhóm khách hàng

Nút chức năng:

- + Làm tươi: tải lại danh sách các khách hàng
- + Thêm mới: thêm một dòng trống để nhập thông tin nhóm khách hàng
- + Lưu: lưu thông tin đã chỉnh sửa trên gridcontrol
- + Xóa: xóa một dòng đang chọn trên gridcontrol
- + Thoát: thoát from

* Giao diện quản lý loại menu

Quản lý thực đơn		
Drag a column header here to group by that column		
		làm tươi
		thêm mới
		Lưu
		Xóa
		Thoát
Mã Menu	Tên Menu	Ghi Chú
1	Ăn Sáng	
2	Ăn Trưa	
3	Ăn Tối	

Gridcontrol: hiển thị thông tin loại menu

Nút chức năng:

- + Làm tươi: tải lại danh sách các loại menu

- + Thêm mới: thêm một dòng trắng để nhập thông tin loại menu
- + Lưu: lưu thông tin đã chỉnh sửa trên gridcontrol
- + Xóa: xóa một dòng đang chọn trên gridcontrol
- + Thoát: thoát from

* Quản lý khu vực

Mã Khu Vực	Tên Khu Vực	Phí Phục Vụ
1	khu vuc tang 1	9000
2	khu vuc tang 2	1000

Giao diện quản lý khu vực bán hàng

Gridcontrol: hiển thị thông tin khu vực bán hàng, phí phục vụ

Nút chức năng:

- + Làm tươi: tải lại danh sách các khu vực bán hàng
- + Thêm mới: thêm một dòng trắng để nhập thông tin khu vực bán hàng
- + Lưu: lưu thông tin đã chỉnh sửa trên gridcontrol
- + Xóa: xóa một dòng đang chọn trên gridcontrol
- + Thoát: thoát from

* Giao diện quản lý bàn café

Mã Khu Vực	Mã Bàn	Tên Bàn
khu vuc tang 1	1	Bàn Số 1
khu vuc tang 1	2	Bàn Số 2

Gridcontrol: hiển thị thông tin bàn café

Combox: hiển thị bàn theo khu vực

Nút chức năng:

- + Hiện tất cả: hiển thị tất cả số bàn hiện có
- + Thêm mới: thêm một dòng trắng để nhập thông tin bàn café
- + Lưu: lưu thông tin đã chỉnh sửa trên gridcontrol
- + Xóa: xóa một dòng đang chọn trên gridcontrol
- + Thoát: thoát from

2.7 Xử lý các biến cố, sự kiện trên từng form

Các lệnh thêm, sửa, xóa thường lặp lại cho các giao diện. Để thuận tiện trong việc kế thừa các công việc thực hiện trước đó, chúng tôi đưa ra giải pháp xử lý dữ liệu theo mô hình 3 tầng.

Code của lớp xử lý dữ liệu như sau:

```
public abstract class SqlHelper
{

    private static Hashtable parasCache =
    Hashtable.Synchronized(new Hashtable());

    public static void ExecuteNonQuery(CommandType
cmdCommandType, string cmdCommandString, params SqlParameter[]
cmdParameters)
    {
        SqlCommand cmdCommand = new SqlCommand();
        SqlConnection connect = new
SqlConnection(QuanLyQuanCafe.DataAccess.Properties.Settings.Default.KetNoi);
        try
        {
            PrepareCommand(cmdCommand, connect, null,
cmdCommandType, cmdCommandString, cmdParameters);
            cmdCommand.ExecuteNonQuery();
            cmdCommand.Parameters.Clear();
        }
        catch (Exception e)
        {
            // throw ex; SqlException ex
        }
        finally
        {
            connect.Close();
            SqlConnection.ClearPool(connect);
        }
    }
}
```

```

    }
}

public static DataTable ExecuteData(CommandType
cmdCommandType, string cmdCommandString, params SqlParameter[]
cmdParameters)
{
    SqlCommand cmdCommand = new SqlCommand();
    SqlConnection connect = new
SqlConnection(QuanLyQuanCafe.DataAccess.Properties.Settings.Default.KetNoi);

    try
    {
        DataTable dattTopic = new DataTable();
        SqlDataAdapter dataTopic = new
SqlDataAdapter(cmdCommand);
        PrepareCommand(cmdCommand, connect, null,
cmdCommandType, cmdCommandString, cmdParameters);
        cmdCommand.ExecuteNonQuery();
        dataTopic.Fill(dattTopic);
        cmdCommand.Parameters.Clear();
        return dattTopic;
    }
    catch (SqlException ex)
    {
        //throw ex;
        return null;
    }
    finally
    {
        connect.Close();
        SqlConnection.ClearPool(connect);
    }
}

```

```

        public static void UpdateTB(DataTable tb, CommandType
cmdCommandType, string cmdCommandString, params SqlParameter[]
cmdParameters)
        {
            SqlCommand cmdCommand = new SqlCommand();
            SqlConnection connect = new
SqlConnection(QuanLyQuanCafe.DataAccess.Properties.Settings.Default.KetNoi);
            try
            {
                DataTable dattTopic = new DataTable();
                SqlDataAdapter dataTopic = new
SqlDataAdapter(cmdCommand);
                PrepareCommand(cmdCommand, connect, null,
cmdCommandType, cmdCommandString, cmdParameters);
                cmdCommand.ExecuteNonQuery();
                SqlCommandBuilder cb = new
SqlCommandBuilder(dataTopic);
                dataTopic.Update(tb);

            }
            catch (SqlException ex)
            {
                // throw ex;

            }
            finally
            {
                connect.Close();
                SqlConnection.ClearPool(connect);
            }
        }

```

```

        public static SqlDataReader ExecuteReader(CommandType
cmdCommandType, string cmdCommandString, params SqlParameter[]
cmdParameters)
        {
            SqlCommand cmdCommand = new SqlCommand();
            SqlConnection connConnection = new
SqlConnection(QuanLyQuanCafe.DataAccess.Properties.Settings.Defa
ult.KetNoi);
            try
            {
                PrepareCommand(cmdCommand, connConnection, null,
cmdCommandType, cmdCommandString, cmdParameters);
                SqlDataReader datrDataReader =
cmdCommand.ExecuteReader(CommandBehavior.CloseConnection);
                cmdCommand.Parameters.Clear();
                return datrDataReader;
            }
            catch (SqlException ex)
            {
                connConnection.Close();
                SqlConnection.ClearPool(connConnection);
                throw ex;
            }
        }

        private static void PrepareCommand(SqlCommand
cmdCommand, SqlConnection connConnection, SqlTransaction
trasTransaction, CommandType cmdCommandType, string
cmdCommandString, SqlParameter[] cmdParameters)
        {
            if (connConnection.State != ConnectionState.Open)
            {
                connConnection.Open();
            }
            cmdCommand.Connection = connConnection;

```

```

        cmdCommand.CommandText = cmdCommandString;
        if (trasTransaction != null)
        {
            cmdCommand.Transaction = trasTransaction;
        }

        cmdCommand.CommandType = cmdCommandType;
        if (cmdParameters != null)
        {
            foreach (SqlParameter para in cmdParameters)
            {
                cmdCommand.Parameters.Add(para);
            }
        }
    }

    public static void PrepareCommand(SqlCommand cmdCommand,
    CommandType cmdCommandType, string cmdCommandString,
    SqlParameter[] cmdParameters)
    {
        cmdCommand.Parameters.Clear();
        cmdCommand.CommandType = cmdCommandType;
        cmdCommand.CommandText = cmdCommandString;
        if (cmdParameters != null)
            foreach (SqlParameter para in cmdParameters)
                cmdCommand.Parameters.Add(para);
    }

    public static void CacheParameters(string cacheKey,
    params SqlParameter[] cmdParameters)
    {
        parasCache[cacheKey] = cmdParameters;
    }

```

```

        public static SqlParameter[] GetCachedParameters(string
cacheKey)
        {
            SqlParameter[] cachedParms =
(SqlParameter[])parasCache[cacheKey];

            if (cachedParms == null)
            {
                return null;
            }
            SqlParameter[] clonedParms = new
SqlParameter[cachedParms.Length];
            for (int i = 0, j = cachedParms.Length; i < j; i++)
            {
                clonedParms[i] =
(SqlParameter)((ICloneable)cachedParms[i]).Clone();
            }
            return clonedParms;
        }
    }

```

3 Thiết kế chức năng Xem thông tin

3.1 Thiết kế form nạp một phiếu đã nhập

3.1.1 Thiết kế form nạp danh sách phiếu nhập

*** Giao diện nhập hàng:**

Thông Tin Nhập Kho Mã PN : 29 Phiếu Nhập Mới N... Nhân Viên : Duc Cong Ngày Nhập : 12 Tháng Sáu 2013 Tổng Tiền : 1344409 VND Phiếu Ng : True Họ tên người giao : lê văn đồng Nhà cung cấp : Nhà Cung Cấp Tổng...		Thông tin chi tiết phiếu nhập Tên Hàng : Bia 333 lon 330ml Số Lượng : 3 Đơn Giá : 122222 DVT : Két(20)		Thanh toán Nhập Số Tiền Cần Thanh Toán : 0 Số Tiền Du :	
--	--	---	--	--	--

số thứ tự	mã Sản phẩm	đơn giá	Đơn vị tính	số lượng	Thành tiền
Mã Phiếu Nhập: 29					
1	Bia 333 lon 330ml	122222	Két(20)	3	366666
2	coca cola 390ml	122222	Két(20)	3	366666
3	Nước yến Nature 250ml	122222	Két(20)	3	366666
4	NTL Number 1 pet 370ml	122222	Két(20)	3	366666
5	Cam ép Twister 470ml	122222	Két(20)	3	366666
6	Sिंग chai	122222	Két(20)	3	366666
7	Dinh Snu 275ml	122222	Két(20)	3	366666

Gidcontrol: hiển thị thông tin chi tiết của phiếu nhập (mã phiếu, stt, sản phẩm, đơn giá, DVT, số lượng, thành tiền)

Nút chức năng:

Thêm phiếu nhập, thêm chi tiết phiếu nhập

Xóa phiếu nhập, xóa chi tiết phiếu nhập

Lưu: sao lưu những thông tin mà gridcontrol hiển thị

In: in phiếu nhập theo mẫu:

Tên Quản : No Name		Mẫu số: 01 - VT			
Bộ Phận : Kho		(Ban hành theo QĐ số: 48/2006/QĐ- BTC Ngày 14/9/2006 của Bộ trưởng BTC)			
PHIẾU NHẬP KHO Ngày 12 Tháng 05 Năm 2013 Số Phiếu: 32					
Họ tên Người Giao Hàng : aaaaaaa					
Theo Phiếu Nhập Số : 32		Ngày 12 Tháng 05 Năm 2013			
Của: Nhà Cung Cấp Tổng Hợp					
STT	Tên Hàng	DVT	Số Lượng	Đơn Giá	Thành Tiền
1	Bia Heieneken 330ml	Chai	10,00	2.555	25.550
Tổng :			10,00	2.555	25.550
Tổng Tiền (Ghi Bằng Chữ) : Hai nghìn, năm trăm năm mươi lăm VND					
Người lập phiếu (Ký, họ tên)		Người giao hàng (Ký, họ tên)		Thủ kho (Ký, họ tên)	
Kế toán trưởng (Hoặc bộ phận có nhu cầu nhận) (Ký, họ tên)					

Tính tiền: để nhận biết phiếu còn nợ hay không

Loại Phiếu
Xóa

Số Phiếu	Mã Nhà CC	Người Lập	Ngày Tháng	Phiếu Nợ	Bản	Khu vực	Thời Gian KT	Tổng Tiền	Phiếu Khóa	Nhân Viên
Loại Phiếu : Nhập										
Loại Phiếu : Xuất										
30		Nguyễn thu...	12/06/2013	☐	Bản Số 2	khu vực tan...	14/06/2013	1005012	☐	Duc Cong
31		Nguyễn thu...	13/06/2013	☐	Bản Số 5	khu vực tan...	14/06/2013	116521	☒	Duc Cong
33		Duc Cong	19/06/2013	☐	Bản Số 5	khu vực tan...	19/06/2013	601260	☒	Duc Cong
34		Duc Cong	19/06/2013	☒	Bản Số 1	khu vực tan...		0	☐	Duc Cong

Drag a column header here to group by that column
Xóa CT

STT	Sản Phẩm	Số Lượng	Đơn Giá	Thành Tiền	DVT
1	Bia Heieneken 330ml	1	33655	33655	3
2	Bia Heieneken 330ml	1	33655	33655	3
3	Bia Heieneken 330ml	1	33655	33655	3
4	Bia Heieneken 330ml	1	33655	33655	3
5	Bia Heieneken 330ml	1	33655	33655	3
6	Bia Heieneken 330ml	1	33655	33655	3
7	Bia Heieneken 330ml	1	33655	33655	3
8	Bia Heieneken 330ml	2	33655	67310	3

3.1.2 Thiết kế form hiện thông tin chi tiết một phiếu đã nhập

* Giao diện kiểm tra nhập/xuất kho

Gridcontrol trên: hiển thị phiếu nhập xuất

Gridcontrol dưới: hiển thị chi tiết phiếu nhập xuất

Nút chức năng:

+ Xóa: xóa phiếu nhập xuất (lưu ý: phải xóa hết chi tiết phiếu trước)

+ Xóa CT: xóa chi tiết nhập xuất

Phần chi tiết phiếu đã nhập/xuất được tích hợp chung trong giao diện nhập như hình trên.

3.2 Thiết kế form nạp một phiếu đã xuất

3.2.1 Thiết kế form nạp danh sách phiếu xuất

Giao diện tương tự giao diện nhập kho:

Thông Tin Nhập Kho Mã PN : 29 Phiếu Nhập Mới N... Nhân Viên : Dục Cong Ngày Nhập : 12 Tháng Sáu 2013 Tổng Tiền : 1344409 VND Phiếu Ng : True Họ tên người giao : Lê Văn Đồng Nhà cung cấp : Nhà Cung Cấp Tổng...		Thông tin chi tiết phiếu nhập Tên Hàng : Bia 333 lon 330ml Số Lượng : 3 Đơn Giá : 122222 DVT : Két(20)		Thanh toán Nhập Số Tiền Cần Thanh Toán : 0 Số Tiền Dư :	
---	--	---	--	--	--

số thứ tự	mã Sản phẩm	đơn giá	Đơn vị tính	số lượng	Thành tiền
Mã Phiếu Nhập: 29					
1	Bia 333 lon 330ml	122222	Két(20)	3	366666
2	coca cola 390ml	122222	Két(20)	3	366666
3	Nước yến Nature 250ml	122222	Két(20)	3	366666
4	NTL Number 1 pet 370ml	122222	Két(20)	3	366666
5	Cam ép Twister 470ml	122222	Két(20)	3	366666
6	Sिंग chai	122222	Két(20)	3	366666
7	Dum + Cnu 275ml	122222	Két(20)	3	366666

Chỉ chú ý khi xuất kho thì mã [loại] phiếu phải quy định khác với [loại] phiếu nhập.

3.2.2 Thiết kế form hiện thông tin chi tiết một phiếu đã xuất

Giao diện được tích hợp phía dưới giao diện xuất kho như hình trên.

4. Thiết kế chức năng Tìm kiếm thông tin

4.1 Tạo form giao diện

Mã	Tên	Địa chỉ	Điện thoại	#
Tìm kiếm ...				
00004	Lê Văn Đồng			
00005	Đào Văn Minh		0908	
00003	Long Hoàng Lâm			
00001	Hoàng Văn Phương			
00002	Nguyễn Văn Khánh	HD	012564872	

Việc tìm kiếm khách hàng có thể tùy biến theo nhu cầu của người sử dụng, tuy nhiên ở giao diện này chúng tôi chỉ yêu cầu đến việc tìm kiếm các thông tin cơ bản như tìm theo họ tên, mã khách hàng hoặc theo ngày mua hàng.

4.2 Xử lý truy vấn, sự kiện

4.2.1 Viết lệnh tìm kiếm theo một tiêu chí riêng lẻ

Tìm kiếm riêng lẻ là tìm theo một điều kiện trong mệnh đề Where của truy vấn Select trong T-SQL.

Chẳng hạn với việc chỉ tìm kiếm theo mã khách hàng, chúng ta sẽ đặt lệnh:

```
Select * From KhachHang Where MaMH=@MaKH
```

4.2.2 Viết lệnh tìm kiếm theo nhiều điều kiện kết hợp

Trong trường hợp cần tìm kiếm theo nhiều điều kiện kết hợp, chúng ta nên viết lệnh ngay trong Code để linh hoạt trong việc kết hợp nhiều điều kiện. Chẳng hạn chúng ta có thể sử dụng câu lệnh if để lập luận cho các tùy chọn của người sử dụng để đưa ra lệnh tìm kiếm tương ứng.

5. Thiết kế chức năng In

5.1 Tạo form In

5.1.1 Thêm nút lệnh in có sử dụng biểu tượng in

Để thêm một nút lệnh in có hình ảnh trên nút lệnh như một biểu tượng, chúng ta thao tác như sau:

Bước 1: Kéo thả biểu tượng Button trên ToolBox vào form cần in.



Bước 2: Đặt thuộc tính cần thiết:

+ Image: chọn hình ảnh biểu trưng cho máy in, hình ảnh này chúng ta có thể tìm kiếm dễ dàng trên mạng hoặc có thể tự sử dụng công cụ để vẽ.

Text: xóa text

5.1.2 Tạo form chứa CrystalReportViewer để in báo cáo

Ví dụ để in danh sách hàng hóa chúng ta tạo form mới, đặt tên là frmDanhSachHangHoa, thiết lập thuộc tính WindowState của form là Maximized, đưa vào form một điều khiển CrystalReportViewer (đặt tên là CrystalReportViewer1).

Tuy nhiên để in các danh sách khác chúng ta không cần phải tạo nhiều form dạng này mà có thể sử dụng chung 1 form để in tất cả các mẫu report:



5.2 Xây dựng thủ tục SQL để tạo nguồn dữ liệu cần in

5.2.1 Viết thủ tục lưu trữ

Vào Microsoft SQL Server, mở cơ sở dữ liệu quản lý bán hàng ra tạo một thủ tục (Stored Procedures) với cú pháp dạng như sau:

```
CREATE PROCEDURE sp_DanhSachHangHoa (@MaLoai nvarchar(5))
AS
BEGIN
SELECT LoaiHangHoa.MaLoai, LoaiHangHoa.TenLoai,
HangHoa.HangHoaID, HangHoa.TenHH, HangHoa.MaHieu, HangHoa.DVT
FROM HangHoa INNER JOIN LoaiHangHoa ON HangHoa.MaLoai =
LoaiHangHoa.MaLoai
WHERE tblLoaiHangHoa.MaLoai=@MaLoai
END
```

5.2.2 Sử dụng thủ tục lưu trữ từ chương trình

Sau đó từ chương trình Microsoft Visual Studio.NET chúng ta có thể sử dụng tên thủ tục để thực hiện truy vấn như sau:

```
cmd.CommandType = CommandType.StoredProcedure; //cmd là
SqlCommand
cmd.CommandText = "sp_DanhSachHangHoa";
cmd.Parameters.Clear() ;
cmd.Parameters.AddWithValue("@MaLoai", cboMaLoai.SelectedValue);
da.SelectCommand = cmd ; //da là SqlDataAdapter
ds.Clear(); //ds là DataSet
da.Fill(ds, "DanhSachHangHoa");
DataGridView1.DataSource = ds.Tables("DanhSachHangHoa") ;
```

BÀI 3: THIẾT KẾ GIAO DIỆN CHƯƠNG TRÌNH

Mã Bài: MD24_03

Giới thiệu:

Hệ thống giao diện cần đẹp mắt và dễ sử dụng để tạo hứng thú cho người làm việc, nhất là với phần mềm thương mại điều này là hết sức cần thiết để tạo ấn tượng ngay từ giai đoạn tiếp thị ban đầu đến khách hàng.

Mục tiêu:

- Trình bày được sơ đồ phân rã chức năng của ứng dụng;
- Xây dựng được giao diện đăng nhập cho người dùng đảm bảo tính bảo mật cho chương trình;
- Xây dựng giao diện chính của phần mềm, ở đó sẽ dễ dàng liên kết và thực hiện các chức năng chính của phần mềm;
- Rèn luyện tính tích cực, chủ động sáng tạo.

Nội dung chính:

1. Tạo form đăng nhập với người dùng

1.1 Tạo bảng lưu thông tin đăng nhập

```
CREATE TABLE [dbo].[USERS] (  
    [USERS] [nvarchar] (50) NOT NULL,  
    [PASS] [nvarchar] (60) NOT NULL,  
    [MaNV] [int] NOT NULL,  
    [Quyen] [int] NULL,  
    [Pass2] [nvarchar] (50) NULL,  
    [id] [int] IDENTITY(1,1) NOT NULL,  
    CONSTRAINT [pk_User] PRIMARY KEY CLUSTERED  
    (  
        [id] ASC  
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =  
    ON) ON [PRIMARY],  
    CONSTRAINT [pk_rangbuoc] UNIQUE NONCLUSTERED  
    (  
        [USERS] ASC  
    )
```

```

) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
GO

CREATE TABLE [dbo].[Quyên] (
    [Quyên] [int] NOT NULL,
    [TenQuyên] [nvarchar](20) NULL,
    [GhiChu] [nvarchar](1000) NULL,
PRIMARY KEY CLUSTERED
(
    [Quyên] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS =
ON) ON [PRIMARY]
) ON [PRIMARY]
GO

/***** Object: ForeignKey [fUsers_NhanVien] *****/
ALTER TABLE [dbo].[USERS] WITH CHECK ADD CONSTRAINT
[fUsers_NhanVien] FOREIGN KEY ([MaNV])
REFERENCES [dbo].[NHANVIEN] ([MaNV])
GO
ALTER TABLE [dbo].[USERS] CHECK CONSTRAINT [fUsers_NhanVien]
GO

```

1.2 Thiết kế giao diện đăng nhập

The image shows a Windows-style login dialog box. The title bar says 'Đăng nhập'. Inside, there's a group box 'THÔNG TIN TRUY CẬP HỆ THỐNG'. It contains two text boxes: the first is labeled 'tên truy cập' and the second is labeled 'mật khẩu'. Below these are two buttons: 'đăng nhập' (login) and 'thoát' (exit). At the bottom left, there's a blue underlined text 'Quyên password' (Forgot password).

Form đăng nhập cho phép người dùng try cập vào phần mềm ứng dụng

Cho phép người sử dụng dùng pass try cập thay đổi pass try cập và dùng pass dự phòng thay đổi pass try cập và pass dự phòng

ID: admin, pass try cập: admin

1.3 Viết lệnh đăng nhập

```
use = USERS.SelectByID(txtTenDangNhap.Text);  
if (use.USERS != null)  
{  
    if (use.PASS == md5(txtPass.Text))  
    {  
        this.DialogResult = DialogResult.OK;  
        use.Pass2 = "";  
        use.PASS = "";  
        txtPass.Text = "";  
        this.Close();  
    }  
    else  
    {  
        sldn = sldn - 1;  
        txtPass.Focus();  
        if (sldn == 0)  
            Close();  
    }  
}
```

```

        MessageBox.Show("Xin lỗi pass sai, số lần
        đăng nhập còn " + sldn + " lần");
        txtPass.Focus();
    }
}
else
{
    MessageBox.Show("Xin lỗi user này không có");
    txtTenDangNhap.Focus();
}
}
//Hàm mã hóa
string md5(string data)
{
    return
    BitConverter.ToString(encryptData(data)).Replace("-",
    "").ToLower();
}

```

2. Tạo form giao diện chính của chương trình

2.1 Thiết lập các thuộc tính của form chính

StartPosition: CenterParent

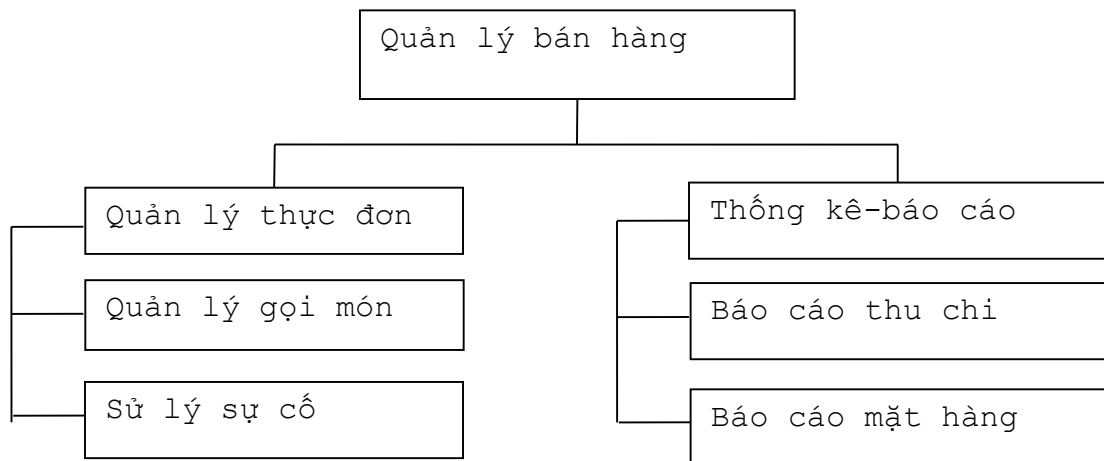
Text: Phần mềm quản lý bán hàng

Window State: Maximized

2.2 Thiết kế hệ thống menu, toolbar

2.2.1 Thiết kế hệ thống menu

Đưa vào form chính điều khiển MenuStrip và thiết kế hệ thống menu quản lý bán hàng như cấu trúc sau:



2.2.2 Thiết kế hệ thống toolbar

Tương tự như cấu trúc trên, nhưng chúng ta sử dụng điều khiển ToolStrip để thiết kế thanh công cụ cho giao diện chính.

3. Tạo các liên kết đến các chức năng trong chương trình

3.1 Chức năng mở form

Để mở các form từ hệ thống menu hoặc toolbar của phần mềm, chúng ta lần lượt viết code bằng cách nhấp đôi chuột vào menu hoặc toolbar chức năng tương ứng và thực hiện lệnh sau:

```
<tên form> f = new <tên form>();  
f.Show();
```

3.2 Chức năng trợ giúp

Để hỗ trợ cho người sử dụng, chúng ta cần thêm hai menu về việc giới thiệu phần mềm và hướng dẫn sử dụng.

3.3 Chức năng thiết lập hệ thống

Với những thông số có tính thay đổi giữa người sử dụng này với người sử dụng khác hoặc giữa các đơn vị sử dụng khác nhau chúng ta cần thiết kế một form để thiết lập các thông số này, giá trị của các thông số phải được lưu vào cơ sở dữ liệu. Bảng dữ liệu này có thể 2 trường Text và Value.

BÀI 4: THIẾT KẾ CÁC BÁO CÁO

Mã Bài: MD24_04

Giới thiệu:

Thiết kế báo cáo để in ấn là một phần không thể thiếu của phần mềm, việc lựa chọn công cụ Crystal Report của Microsoft chạy sử dụng công cụ XtraReport của DevExpress phụ thuộc vào sở thích của từng người.

Mục tiêu:

- Trình bày được tính năng, cách sử dụng của một số công cụ thiết kế báo cáo;
- Xác định đúng nguồn dữ liệu, các phương pháp xử lý dữ liệu tránh sai lệch cho kết quả báo cáo, thống kê;
- Xây dựng được giao diện báo cáo thân thiện, logic phù hợp với trình tự công việc;
- Rèn luyện tính tỉ mỉ, chính xác, khoa học.

Nội dung chính:

1. Xác định chức năng của báo cáo

1.1 Chức năng in danh sách

Với phương pháp tạo báo cáo như trình bày ở trên, chúng ta phải có hệ thống menu in danh sách từng danh mục khi có yêu cầu.

1.2 Chức năng in thống kê, tổng hợp

Một báo cáo cơ bản trong việc thống kê, tổng hợp là báo cáo tồn

04/06/13 14:54:07

22/06/13 14:54:07

Hiện thị

Sao Lưu

In Báo Cáo

Menu

Mã SP	Tên SP	SL DK	TT DK	SL Nhập	TT Nhập	SL Xuất	TT Xuất	SL CK	TT CK	
Menu: Ẩn Sáng										
Menu: Bia										
57	Bia Heieneken 330ml	10		91650	220	1344321	72	1243860	158	1448070
58	Sapporo lon 330ml	0	0		60	366666	0	0	60	0
59	Bia Tiger 320ml	0	0		0	0	0	0	0	0
60	Bia 333 lon 330ml	0	0		60	366666	0	0	60	549960
61	Bia Sài Gòn Special	0	0		0	0	0	0	0	0

Giao diện Báo cáo tồn kho

Gridcontrol: hiển thị thông tin sản phẩm, số lượng, thành tiền(đầu kỳ,nhập,xuất,cuối kỳ)

Nút chức năng:

- + Hiển thị: báo cáo tồn từ ngày tới ngày
- + Sao lưu: lưu thông trên gridcontrol hiển thị thành file hoặc in ra giấy
- + In báo cáo: in báo cáo tồn theo mẫu:

Quản Cà Phê : No Name....

Bộ Phận : Kiểm Kho

BÁO CÁO TỒN KHO

Từ Ngày 04 Tháng Sáu 2013 Đến Ngày 22 Tháng Sáu 2013

Vật Tư Hàng Hóa	Đầu Kỳ		Nhập Kho		Xuất Kho		Cuối Kỳ	
Tên Sản Phẩm	SL	TT	SL	TT	SL	TT	SL	TT
Bia Heieneken 330ml	10	91.650	220	1.344.321	72	1.243.860	158	1.448.070
Sapporo lon 330ml	0		60	366.666	0		60	
Cafe Đen Nóng	0		0		0		0	
Tổng:	10	91.650	840	5.133.159	72	1.243.860	778	6.580.950

Ngày 22 Tháng 06 Năm 2013

Người lập phiếu

(Ký, họ tên)

Thủ Kho

(Ký, họ tên)

Kế Toán

(Ký, họ tên)

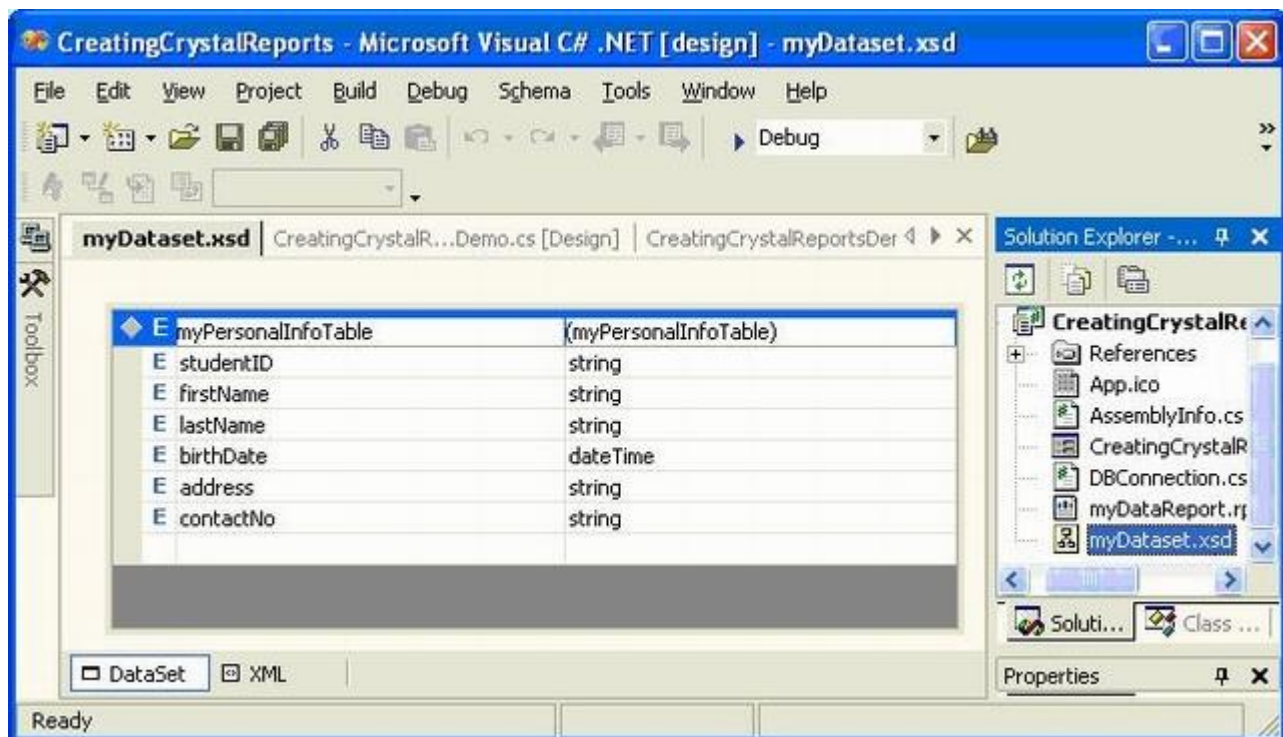
Thủ Trưởng Đơn Vị

(Ký, họ tên, đóng dấu)

2. Xác định, tạo nguồn dữ liệu cho báo cáo

2.1 Xây dựng nguồn dữ liệu ảo để thiết kế báo cáo

Đầu tiên vào menu Project → “Add New Item”, bạn add a Dataset Giống hình bên dưới.:



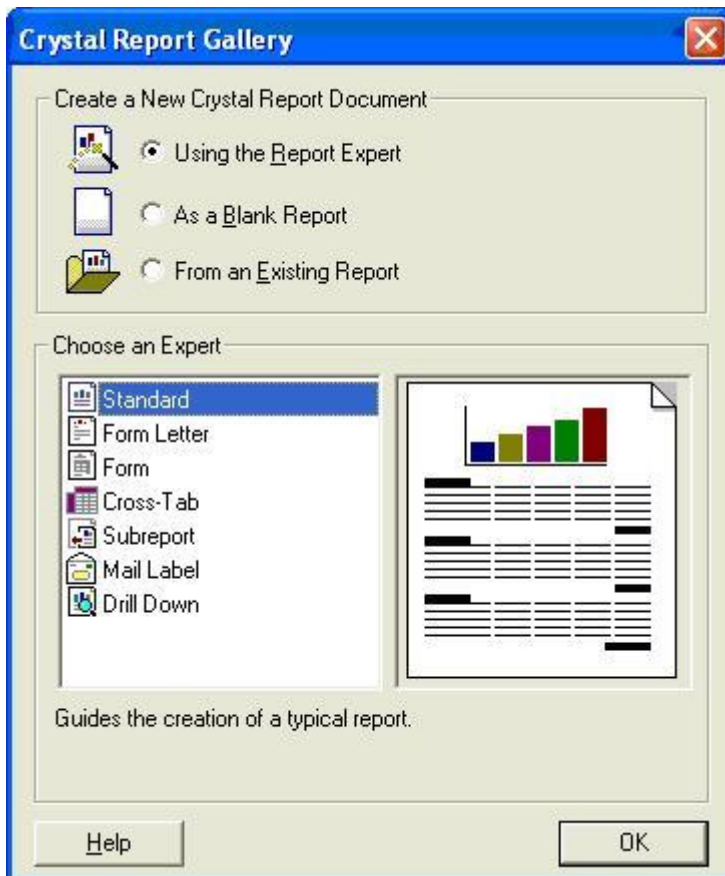
2.2. Xác định nguồn dữ liệu thật để báo cáo

```
String connString = @"Chuỗi kết nối";  
SqlConnection conn = new SqlConnection(connString);  
conn.Open();  
string query = "SELECT studentID, firstName, lastName,  
birthDate, address, contactNo FROM studentInfo";  
SqlDataAdapter SqlDA = new SqlDataAdapter(query, conn);
```

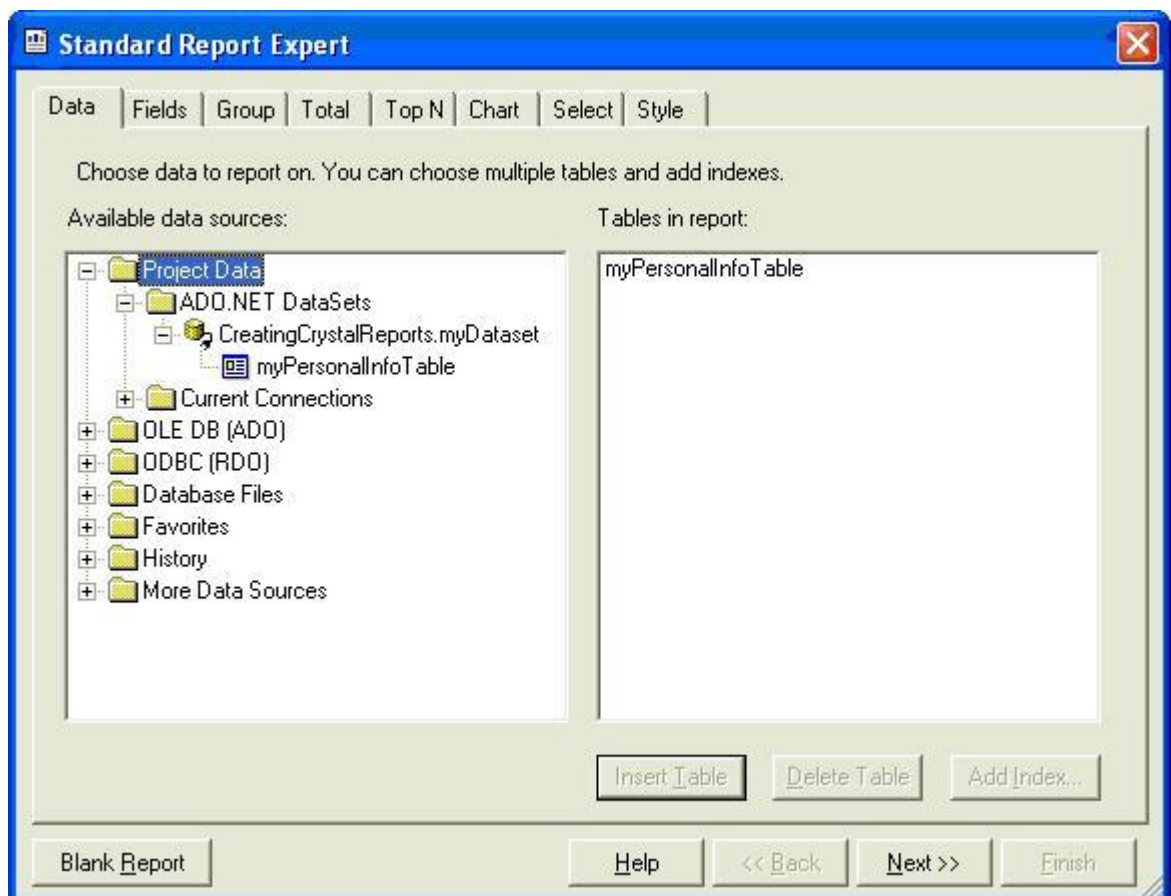
3. Tạo báo cáo với công cụ

3.1 Sử dụng công cụ Report Wizard để tạo báo cáo

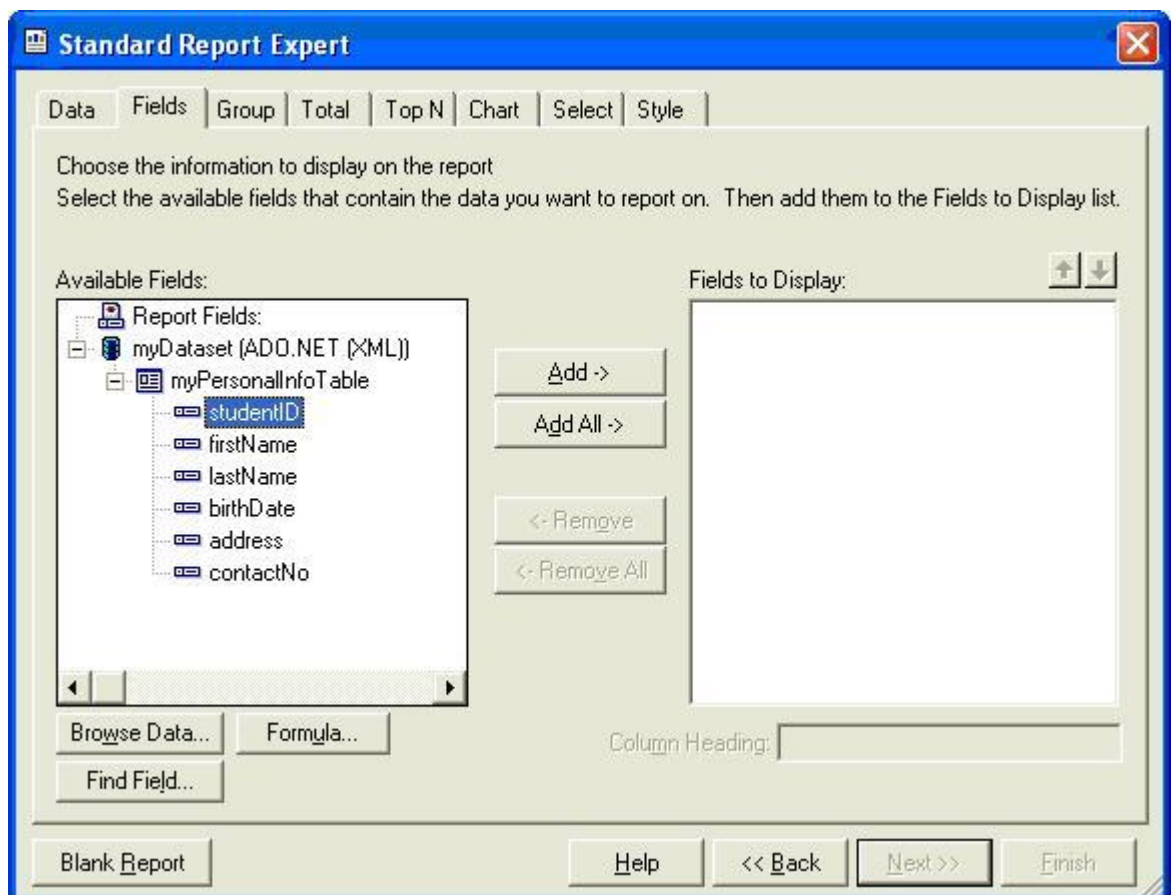
Sau khi tạo xong dataset bạn add a Crystal Report vào project bằng cách “Add New Item” trong “Project” menu. Sau đó chọn tên xuất hiện rồi nhấn nút ok.



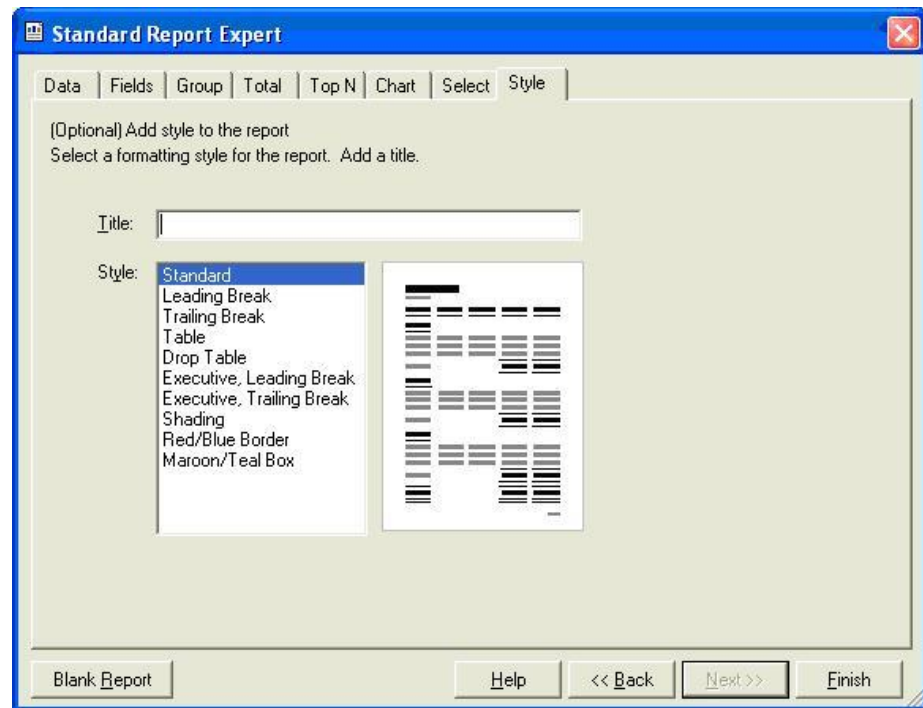
Màn hình sẽ hiển thị như sau bạn chọn mặt định rồi ấn OK
Sau đó chọn table rồi insert table.



sau đó chọn các fields hiển thị trong report --> nhấn nút add



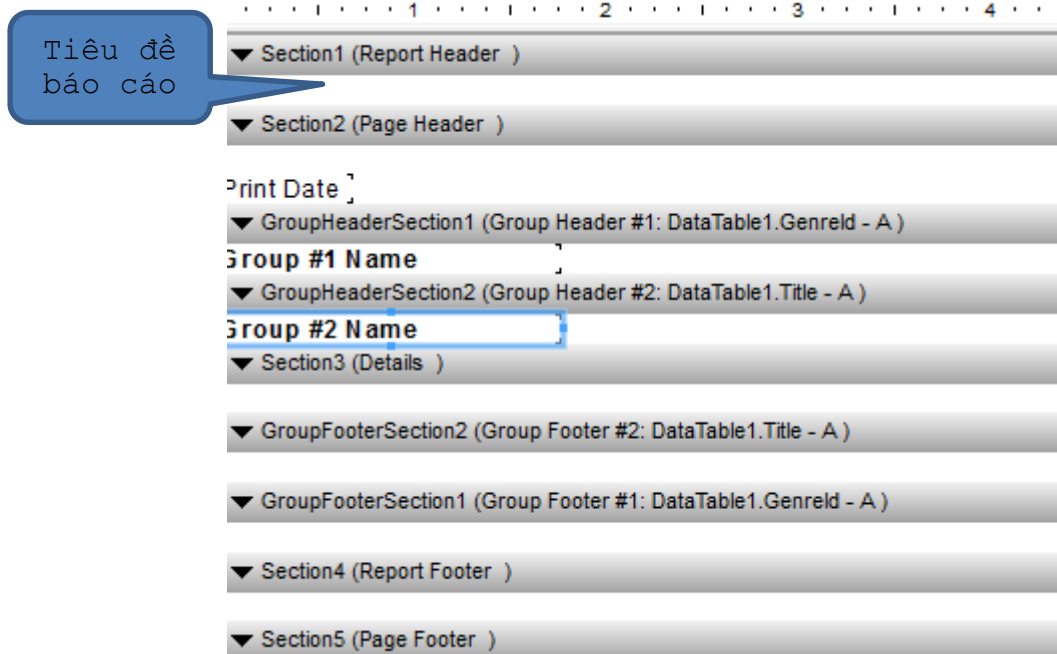
Xong xu bạn nhấn nút next-->cuối cùng chọn style kiểu bạn muốn report hiển thị



Kế tiếp bạn sẽ cho report hiển thị trên form

3.2 Thêm tiêu đề cho báo cáo

Trong trường hợp muốn thêm tiêu đề cho báo cáo thì thêm phần Report Header (Session 1) vào báo cáo.



4. Chỉnh sửa các thành phần, thuộc tính trong báo cáo

4.1 Định dạng ngày trong báo cáo

Để định dạng ngày trong Crystal Report bạn nháy chuột phải vào ô cần sửa và chọn Format Field, sau đó Customize Format.

Ví dụ: 3/13/2009 11:24:16PM cần chuyển về dạng dd/MM/yyyy

Tab DateTime chọn Order=Date, Tab Date chọn Order → DMY

+ Chọn lại phần format cho ngày tháng 2 số → OK

100,000,000.00 → cần loại bỏ từ dấu chấm và phần thập phân
Tab Number Bỏ phần Thousands Separator, phần Decimal chọn 1 → OK.

4.2 Kẻ viền cho báo cáo

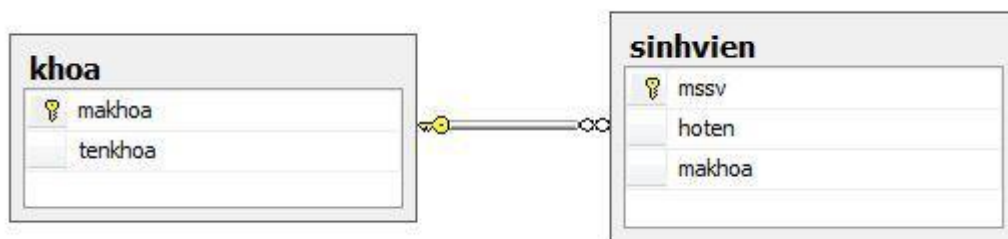
Để tạo viền cho báo cáo chúng ta có thể sử dụng đối tượng line để vẽ các đường ngang, dọc bao quanh các TextObject. Đây là một điểm yếu của Crystal report so với các công cụ thiết kế report của các hãng thứ ba.

5. Tạo form truyền tham số

Ngoài cách sử dụng StoredProcedure mà các bạn đã biết, thì các bạn còn có thể sử dụng Parameter Field để truyền thông tin ngắn gọn dễ dàng.

Xét một ví dụ như sau:

Với cơ sở dữ liệu:



Form:

Khoa: Khoa toán tin In báo cáo

Main Report

DANH SÁCH SINH VIÊN

Khoa toán tin

Mã số sinh viên	Họ tên sinh viên
0951010009	Hà Vĩ Biểu
0951012104	Luu Hậu Thanh Quang

Current Page No.: 1 Total Page No.: 1 Zoom Factor: 100%

Sau khi chọn khoa trên combobox và nhấn nút In báo cáo thì báo cáo in ra sẽ có tiêu đề tên khoa tương ứng, phía dưới là danh sách sinh viên thuộc khoa đó.

Chúng ta đã quen thuộc với những bài tập tạo báo cáo dạng này bằng cách sử dụng StoredProcedure đưa dữ liệu vào. Thông thường, chúng ta sẽ tạo ngay 1 StoredProcedure lấy mã số sinh viên và họ tên sinh viên theo tham số là mã khoa, cụ thể SQL thế này:

```
CREATE PROCEDURE Sp_BCSVTHEOKHOA
    @makhoa char(4)
AS
BEGIN
    select mssv, hoten from sinhvien where makhoa = @makhoa
END
```

Sau đó là khâu thiết kế báo cáo, chúng ta gặp ngay vấn đề là tham số tên khoa không có trong StoredProcedure trên.

ction1 (Report Header)

DANH SÁCH SINH VIÊN

???

ction2 (Page Header)

Mã số sinh viên Họ tên sinh viên

ction3 (Details)

mssv hoten

ction4 (Report Footer)

Vấn đề là làm thế nào thêm vào trường dữ liệu “tên khoa” đây?

Có một cách là ngay lập tức về lại bên cơ sở dữ liệu sửa lại StoredProcedure đó và kế tiếp là verify bên crystalreport. Chấp nhận rằng chúng ta sẽ truy vấn dư thừa.

	tenkhoa	mssv	hoten
1	Khoa thủy điện	09501012XXZ	Yugi Mutou
2	Khoa thủy điện	0951012XXX	Nguyễn Thủy Thủ
3	Khoa thủy điện	0951012XXY	Võ Thị Thủy Trang

Dư

Cách khác đó là truyền “tên khoa” từ form vào crystal report theo đường Parameter Field.

Cách thực hiện như sau:

Đầu tiên, tôi vào Report, bên cửa sổ Field Explorer, tạo mới một Parameter Field

Create Parameter Field

Parameter Field

Name:

Prompting text:

Value type:

Options

☐ Allow multiple values

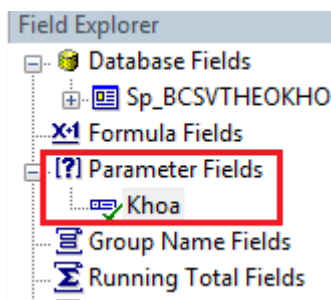
☒ Discrete value(s)

☐ Range value(s)

☐ Discrete and Range Values

☒ Allow editing of default values when there is more than one value

Đặt tên là “khoa”. Tôi bỏ trống phần prompting text, vì tôi không có dự định sử dụng prompt để người dùng nhập vào parameter, mà tôi sẽ truyền vào bằng code C#. Sau khi tạo xong, thì kéo Field Khoa đến vị trí cần hiển thị.



DANH SÁCH SINH VIÊN

?Khoa

Section2 (Page Header)

Section3 (Details)

Section4 (Report Footer)

Section5 (Page Footer)

Mã số sinh viên

Họ tên sinh viên

Lưu lại và về Form để thực hiện gán vào Parameter đó khi bấm vào nút in:

```

        private void btnPrintReport_Click(object sender,
EventArgs e)
        {
            clssForm1BLL.MaKhoa =
cboKhoa.SelectedValue.ToString();
            crystalReportSVTHEOKHOA1.SetDataSource(clssForm1BLL.
GetSinhVienByMaKhoa());

            ParameterDiscreteValue pa = new
ParameterDiscreteValue();
            pa.Value = cboKhoa.Text;

            ParameterFieldDefinition crParameterFieldDefinition
=
crystalReportSVTHEOKHOA1.DataDefinition.ParameterFields["khoa"];
            crParameterFieldDefinition.CurrentValues.Clear();
            crParameterFieldDefinition.CurrentValues.Add(pa);
            crParameterFieldDefinition.ApplyCurrentValues(crPara
meterFieldDefinition.CurrentValues);

            crystalReportViewer1.ReportSource =
crystalReportSVTHEOKHOA1;
        }

```

Những người bạn mới ở đây là:

- ParameterDiscreteValue : khai báo đối tượng thuộc lớp này để chứa dữ liệu rời rạc. Trong ví dụ này thì là tên khoa đang chọn được hiển thị trên combobox.
- ParameterFieldDefinition : khai báo đối tượng thuộc lớp này để tìm đến định nghĩa ParameterField muốn truyền trên báo cáo. Đó chính là ParameterField có tên là “khoa” (giống với tên ParameterField ban nãy tạo bên báo cáo). Sau đó là vài động tác xóa, thêm, sử dụng dữ liệu ta đã truyền vào theo đối tượng ParameterDiscreteValue.

6. Xử lý tham số với báo cáo

Chúng ta có thể viết ngắn gọn lại theo cách này:

```

        private void btnPrintReport_Click(object sender,
EventArgs e)
        {
            clssForm1BLL.MaKhoa =
cboKhoa.SelectedValue.ToString();
            crystalReportSVTHEOKHOA1.SetDataSource(clssForm1BLL.
GetSinhVienByMaKhoa());

            ParameterDiscreteValue pdv = new
ParameterDiscreteValue();
            pdv.Value = cboKhoa.Text;
            ParameterValues pv = new ParameterValues();
            pv.Add(pdv);
            crystalReportSVTHEOKHOA1.DataDefinition.ParameterFields["khoa"].
ApplyCurrentValues(pv);

            crystalReportViewer1.ReportSource =
crystalReportSVTHEOKHOA1;
        }
    }

```

7. Thiết kế các báo cáo chi tiết

7.1 Tạo thủ tục SQL để kết xuất dữ liệu

Với việc kết xuất dữ liệu cho việc nhập xuất tồn, chúng ta có thể có thủ tục sau:

```

CREATE proc [dbo].[selectBaoCao]
@NgayDau datetime,
@NgayCuoi datetime
--@tong int output
as
begin
SELECT      dbo.HANGHOA.MaSP,  dbo.HANGHOA.TenSP, HANGHOA.MaLoai ,
GETDATE() as NgayThang
--ton dau
            , ( (HANGHOA.SoLuongTon
+ -- cộng

```

```

--so luong xuất
case when (select
sum(CASE PHIEUXUATNHAP.loai WHEN 1 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
from PHIEUXUATNHAP,ChiTetNhapXuat
where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
and PHIEUXUATNHAP.NgayThang BETWEEN
@NgayCuoi AND GETDATE()
)>0 then (select
sum(CASE PHIEUXUATNHAP.loai WHEN 1 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
from PHIEUXUATNHAP,ChiTetNhapXuat
where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
and PHIEUXUATNHAP.NgayThang BETWEEN
@NgayCuoi AND GETDATE()
) else 0 end

- -- trừ
-- so luong nhập
case when (select
sum(CASE PHIEUXUATNHAP.loai WHEN 0 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
from PHIEUXUATNHAP,ChiTetNhapXuat
where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
and
PHIEUXUATNHAP.NgayThang BETWEEN @NgayCuoi AND GETDATE()
)>0 then (select
sum(CASE PHIEUXUATNHAP.loai WHEN 0 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
from PHIEUXUATNHAP,ChiTetNhapXuat
where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu

```

```

                                and
PHIEUXUATNHAP.NgayThang BETWEEN @NgayCuoi AND GETDATE()
                                ) else 0 end)
+ -- cộng
--so luong xuất
case when (select
sum(CASE PHIEUXUATNHAP.loai WHEN 1 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
from PHIEUXUATNHAP,ChiTetNhapXuat
where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
and PHIEUXUATNHAP.NgayThang BETWEEN
@NgayDau AND @NgayCuoi
)>0 then (select
sum(CASE PHIEUXUATNHAP.loai WHEN 1 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
from PHIEUXUATNHAP,ChiTetNhapXuat
where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
and PHIEUXUATNHAP.NgayThang BETWEEN @NgayDau AND
@NgayCuoi
) else 0 end

- -- trừ
-- so luong nhập
case when (select
sum(CASE PHIEUXUATNHAP.loai WHEN 0 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
from PHIEUXUATNHAP,ChiTetNhapXuat
where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
and PHIEUXUATNHAP.NgayThang BETWEEN @NgayDau
AND @NgayCuoi
)>0 then (select

```

```

sum(CASE PHIEUXUATNHAP.loai WHEN 0 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
from PHIEUXUATNHAP,ChiTetNhapXuat
where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
and PHIEUXUATNHAP.NgayThang BETWEEN @NgayDau
AND @NgayCuoi
) else 0 end) as tonDauKy

--Gia tri dau ky

, (( (HANGHOA.SoLuongTon
+ -- cộng
--so luong xuất
case when (select
sum(CASE PHIEUXUATNHAP.loai WHEN 1 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
from PHIEUXUATNHAP,ChiTetNhapXuat
where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
and PHIEUXUATNHAP.NgayThang BETWEEN
@NgayCuoi AND GETDATE()
)>0 then (select
sum(CASE PHIEUXUATNHAP.loai WHEN 1 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
from PHIEUXUATNHAP,ChiTetNhapXuat
where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
and PHIEUXUATNHAP.NgayThang BETWEEN
@NgayCuoi AND GETDATE()
) else 0 end

- -- trừ
-- so luong nhập

```

```

        case when (select
            sum(CASE PHIEUXUATNHAP.loai WHEN 0 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
            from PHIEUXUATNHAP,ChiTetNhapXuat
            where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
                and
PHIEUXUATNHAP.NgayThang BETWEEN @NgayCuoi AND GETDATE()
            )>0 then (select
            sum(CASE PHIEUXUATNHAP.loai WHEN 0 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
            from PHIEUXUATNHAP,ChiTetNhapXuat
            where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
                and
PHIEUXUATNHAP.NgayThang BETWEEN @NgayCuoi AND GETDATE()
            ) else 0 end)
+ -- cộng
--so luong xuất
        case when (select
            sum(CASE PHIEUXUATNHAP.loai WHEN 1 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
            from PHIEUXUATNHAP,ChiTetNhapXuat
            where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
                and PHIEUXUATNHAP.NgayThang BETWEEN
@NgayDau AND @NgayCuoi
            )>0 then (select
            sum(CASE PHIEUXUATNHAP.loai WHEN 1 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
            from PHIEUXUATNHAP,ChiTetNhapXuat
            where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
                and PHIEUXUATNHAP.NgayThang BETWEEN @NgayDau AND
@NgayCuoi

```

```

        ) else 0 end

-- -- trừ
-- so luong nhap
case when (select
    sum(CASE PHIEUXUATNHAP.loai WHEN 0 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
    from PHIEUXUATNHAP,ChiTetNhapXuat
    where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
    and PHIEUXUATNHAP.NgayThang BETWEEN @NgayDau
AND @NgayCuoi
    )>0 then (select
    sum(CASE PHIEUXUATNHAP.loai WHEN 0 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
    from PHIEUXUATNHAP,ChiTetNhapXuat
    where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
    and PHIEUXUATNHAP.NgayThang BETWEEN @NgayDau
AND @NgayCuoi
    ) else 0 end)* HANGHOA.DonGia) as GiaTriDauKy

-- so luong nhap
,case when (select
    sum(CASE PHIEUXUATNHAP.loai WHEN 0 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
    from PHIEUXUATNHAP,ChiTetNhapXuat
    where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
    and PHIEUXUATNHAP.NgayThang BETWEEN
@NgayDau AND @NgayCuoi
    )>0 then (select
    sum(CASE PHIEUXUATNHAP.loai WHEN 0 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
    from PHIEUXUATNHAP,ChiTetNhapXuat

```

```

        where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
        and PHIEUXUATNHAP.NgayThang BETWEEN @NgayDau
AND @NgayCuoi
    ) else 0 end as SoLuongNhap

-- so luong nhap
        ,case when (select
            sum(CASE PHIEUXUATNHAP.loai WHEN 0 THEN
ChiTetNhapXuat.ThanhTien ELSE 0 END )
            from PHIEUXUATNHAP,ChiTetNhapXuat
            where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
            and PHIEUXUATNHAP.NgayThang BETWEEN
@NgayDau AND @NgayCuoi
            )>0 then (select
            sum(CASE PHIEUXUATNHAP.loai WHEN 0 THEN
ChiTetNhapXuat.ThanhTien ELSE 0 END )
            from PHIEUXUATNHAP,ChiTetNhapXuat
            where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
            and PHIEUXUATNHAP.NgayThang BETWEEN @NgayDau
AND @NgayCuoi
            ) else 0 end as GiaTriNhap

-- so luong Xuat
        ,case when (select
            sum(CASE PHIEUXUATNHAP.loai WHEN 1 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
            from PHIEUXUATNHAP,ChiTetNhapXuat
            where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
            and PHIEUXUATNHAP.NgayThang BETWEEN
@NgayDau AND @NgayCuoi

```

```

        )>0 then (select
            sum(CASE PHIEUXUATNHAP.loai WHEN 1 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
            from PHIEUXUATNHAP,ChiTetNhapXuat
            where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
                and PHIEUXUATNHAP.NgayThang BETWEEN
@NgayDau AND @NgayCuoi
        ) else 0 end as SoLuongXuat

-- Gia tri Xuat
        ,case when (select
            sum(CASE PHIEUXUATNHAP.loai WHEN 1 THEN
ChiTetNhapXuat.ThanhTien ELSE 0 END )
            from PHIEUXUATNHAP,ChiTetNhapXuat
            where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
                and PHIEUXUATNHAP.NgayThang BETWEEN
@NgayDau AND @NgayCuoi
        )>0 then (select
            sum(CASE PHIEUXUATNHAP.loai WHEN 1 THEN
ChiTetNhapXuat.ThanhTien ELSE 0 END )
            from PHIEUXUATNHAP,ChiTetNhapXuat
            where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
                and PHIEUXUATNHAP.NgayThang BETWEEN
@NgayDau AND @NgayCuoi
        ) else 0 end as GiaTriXuat

--ton cuoi ky
        , (HANGHOA.SoLuongTon
+ -- cộng
--so luong xuất
        case when (select

```

```

sum(CASE PHIEUXUATNHAP.loai WHEN 1 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
from PHIEUXUATNHAP,ChiTetNhapXuat
where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
and PHIEUXUATNHAP.NgayThang BETWEEN
@NgayCuoi AND GETDATE()
)>0 then (select
sum(CASE PHIEUXUATNHAP.loai WHEN 1 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
from PHIEUXUATNHAP,ChiTetNhapXuat
where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
and PHIEUXUATNHAP.NgayThang BETWEEN
@NgayCuoi AND GETDATE()
) else 0 end

-- -- trừ
-- so luong nhap
case when (select
sum(CASE PHIEUXUATNHAP.loai WHEN 0 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
from PHIEUXUATNHAP,ChiTetNhapXuat
where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
and
PHIEUXUATNHAP.NgayThang BETWEEN @NgayCuoi AND GETDATE()
)>0 then (select
sum(CASE PHIEUXUATNHAP.loai WHEN 0 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
from PHIEUXUATNHAP,ChiTetNhapXuat
where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
and
PHIEUXUATNHAP.NgayThang BETWEEN @NgayCuoi AND GETDATE()

```

```

        ) else 0 end)
        as SoLuongCuoiKy
--Gia Tri Cuoi KY
        , ( (HANGHOA.SoLuongTon
        + -- cộng
        --so luong xuất
        case when (select
            sum(CASE PHIEUXUATNHAP.loai WHEN 1 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
            from PHIEUXUATNHAP,ChiTetNhapXuat
            where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
                and PHIEUXUATNHAP.NgayThang BETWEEN
@NgayCuoi AND GETDATE()
            )>0 then (select
            sum(CASE PHIEUXUATNHAP.loai WHEN 1 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
            from PHIEUXUATNHAP,ChiTetNhapXuat
            where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
                and PHIEUXUATNHAP.NgayThang BETWEEN
@NgayCuoi AND GETDATE()
            ) else 0 end

        - -- trừ
        -- so luong nhập
        case when (select
            sum(CASE PHIEUXUATNHAP.loai WHEN 0 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
            from PHIEUXUATNHAP,ChiTetNhapXuat
            where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
                and
PHIEUXUATNHAP.NgayThang BETWEEN @NgayCuoi AND GETDATE()
            )>0 then (select

```

```

sum(CASE PHIEUXUATNHAP.loai WHEN 0 THEN
ChiTetNhapXuat.SoLuongQT ELSE 0 END )
from PHIEUXUATNHAP,ChiTetNhapXuat
where ChiTetNhapXuat.MaSP = HANGHOA.MaSP and
ChiTetNhapXuat.MaPhieu = PHIEUXUATNHAP.MaPhieu
and
PHIEUXUATNHAP.NgayThang BETWEEN @NgayCuoi AND GETDATE()
) else 0 end)*HANGHOA.DonGia) as
giatritonCuoiKy

FROM      dbo.HANGHOA

group by HANGHOA.MaSP,
HANGHOA.TenSP, SoLuongTon, DonGia, HANGHOA.MaLoai
end

```

7.2 Tạo báo cáo sử dụng nguồn dữ liệu từ thủ tục SQL

```

cmd.CommandType = CommandType.StoredProcedure; //cmd là
SqlCommand
cmd.CommandText = "selectBaoCao";
cmd.Parameters.Clear() ;
cmd.Parameters.AddWithValue("@Tên_Tham_Số",Giá_trị);
da.SelectCommand = cmd ; //da là SqlDataAdapter
ds.Clear(); //ds là DataSet
da.Fill(ds, "Tên_bảng_ảo");
DataGridView1.DataSource = ds.Tables("Tên_bảng_ảo ") ;

```

BÀI 5: HOÀN THIỆN, ĐÓNG GÓI PHẦN MỀM

Mã Bài: MD24_05

Giới thiệu:

Công việc cuối cùng sau khi hoàn thiện sản phẩm là phải đóng gói để thuận lợi trong việc triển khai đến khách hàng.

Mục tiêu:

- Xác định được một số công cụ đóng gói sản phẩm;
- Kiểm tra, hiệu chỉnh nhằm đảm bảo phần mềm đạt được các yêu cầu đặt ra;
- Bảo mật cho cơ sở dữ liệu nhằm tránh những truy cập trái phép;
- Đóng gói phần mềm, tạo bộ cài đặt;
- Rèn luyện được nghiêm túc, cẩn thận, tỉ mỉ.

Nội dung chính:

1. Kiểm thử phần mềm

Ở đây chỉ nói đến việc Kiểm thử phần mềm cho ứng dụng máy tính và mobile, không bàn đến các nền tảng khác (nhúng, ...)

Kiểm thử phần mềm (software testing) là hoạt động nhằm tìm kiếm, phát hiện các lỗi của phần mềm được kiểm thử về thiết kế, mã nguồn, chức năng, dữ liệu, bảo mật, thân thiện với người dùng, tài liệu kèm theo, môi trường hoạt động, tốc độ hoạt động, khả năng tải của hệ thống, ... Thường được chia thành các nhóm là Nhóm thuộc về chức năng (Functionality), Nhóm không thuộc chức năng (Non-Functionality), Nhóm thuộc về cấu trúc (Structural) và Nhóm liên quan đến các thay đổi (Change Related). (Tham khảo thêm ISO9126)

Kiểm thử phần mềm còn hướng đến mục tiêu xa hơn mà tôi gọi nôm na là “phòng bệnh hơn chữa bệnh”. Tức là nâng cao khả năng kiểm soát và hạn chế các lỗi xảy ra khi phát triển phần mềm ngay từ ban đầu, chứ không đơn thuần chỉ là việc tìm những lỗi sẵn có khi nhóm phát triển đã đưa ra những phiên bản cụ thể của phần mềm.

Kiểm thử phần mềm là một ngành không quá mới tại Việt Nam. Tuy nhiên, hình thức tồn tại chủ yếu của Kiểm thử phần mềm tại Việt Nam là nằm phụ thuộc trong một nhóm phát triển phần mềm (vừa lập trình, vừa kiểm thử bởi một nhóm phát triển). Vấn đề đảm bảo chất lượng phần mềm cũng chưa được các công ty phát triển phần mềm trong nước coi trọng đúng mức. Do đó, mặc dù đã có ở Việt Nam từ lâu, nhưng chỉ một vài năm gần đây, Kiểm thử phần mềm chuyên nghiệp mới được biết đến như một ngành riêng biệt với tiềm năng phát triển lớn. Công ty

LogiGear Việt Nam hiện là công ty hàng đầu Việt Nam trong lĩnh vực kiểm thử phần mềm chuyên nghiệp

2. Hiệu chỉnh phần mềm

2.1 Xác định nguyên nhân gây lỗi

Có nhiều thuật ngữ tiếng Anh có thể dịch sang tiếng Việt là “lỗi”. Tuy nhiên, ý nghĩa của nó trong Kiểm thử hoàn toàn khác nhau:

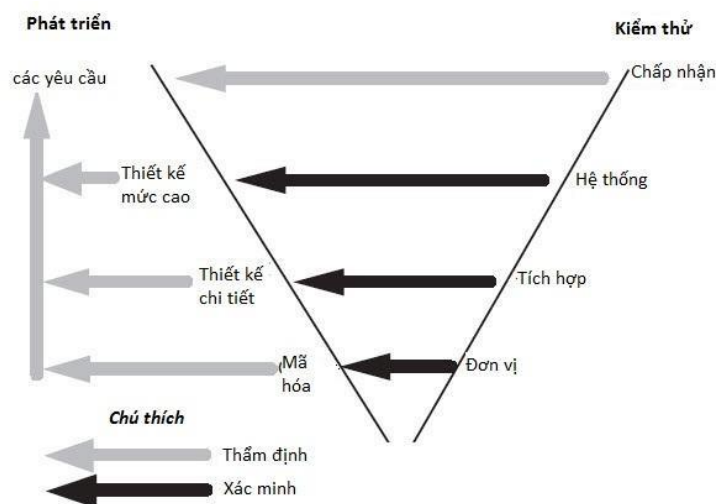
Error, Mistake: Thường là chỉ một lỗi của con người trong quá trình xây dựng phần mềm.

Bug, Defect, Fault: Là lỗi nằm trong mã nguồn, tài liệu của chương trình. Loại lỗi này có nhiều nguyên nhân như: do error của con người, do công nghệ phức tạp, áp lực công việc, do các thành phần của hệ thống tương tác với nhau, ... Kiểm thử viên chủ yếu là bắt các loại lỗi này.

Fail, Failure: Dùng để chỉ các lỗi dưới góc độ của hệ thống. Khi một hệ thống không thực hiện được chức năng cần thiết, hoặc thực hiện chức năng không được phép làm thì được gọi là fail/failure. Một bug có thể là nguyên nhân của nhiều fail khi hệ thống hoạt động.

2.2 Khắc phục lỗi

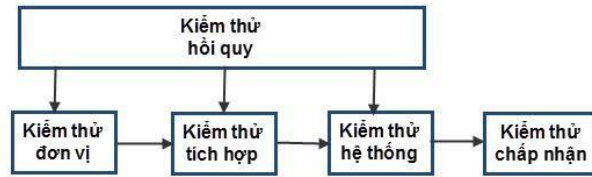
Các giai đoạn kiểm thử tương ứng với các giai đoạn khác nhau trong tiến trình phát triển phần mềm và được khái quát hóa qua mô hình chữ V như hình dưới gồm:



Các giai đoạn phát triển và kiểm thử trong mô hình chữ V

- 1) Kiểm thử đơn vị
- 2) Kiểm thử tích hợp
- 3) Kiểm thử hệ thống

4) Kiểm thử hồi quy



Kiểm thử hồi quy tại các mức kiểm thử phần mềm khác

5) Kiểm thử chấp nhận

3. Xác định công cụ đóng gói

Build và Release Version đều được dùng để chỉ một phiên bản của phần mềm. Tuy nhiên, ý nghĩa và trường hợp sử dụng thì khác nhau:

Build: Thường được dùng để chỉ phiên bản phần mềm trong quá trình phát triển tại nhóm dự án. Những thay đổi về chức năng giữa các build liên tiếp nhau thường là nhỏ (có thể là thêm một vài chức năng, sửa một số lỗi, tối ưu mã nguồn, ...). Các build được tạo ra trong quá trình phát triển phần mềm của nhóm dự án. Một ngày có thể có một hoặc nhiều build.

Release Version: Được dùng để chỉ một bản build. Tuy nhiên, bản build này sẽ được gửi đến cho khách hàng kiểm tra chấp nhận. Những thay đổi giữa các Release Version liên tiếp nhau thường là khá lớn. Phải có nhiều build được viết và kiểm thử tại nhóm dự án thì mới có một Release Version.

Trong bài này chúng ta chọn công cụ đóng gói của chính bộ phần mềm Microsoft Visual Studio .NET.

4. Đặt bảo mật cho cơ sở dữ liệu

4.1 Xây dựng thư viện mã hóa

Để đảm bảo về tính bảo mật, ngoài việc phải mã hóa mật khẩu trong code trước khi lưu vào cơ sở dữ liệu thì các thủ tục (Stored Procedures) và hàm (Functions) trong SQL cũng phải được mã hóa để tránh người sử dụng can thiệp gây ra những lỗi không mong muốn.

4.2 Đặt mật khẩu cho cơ sở dữ liệu và code.

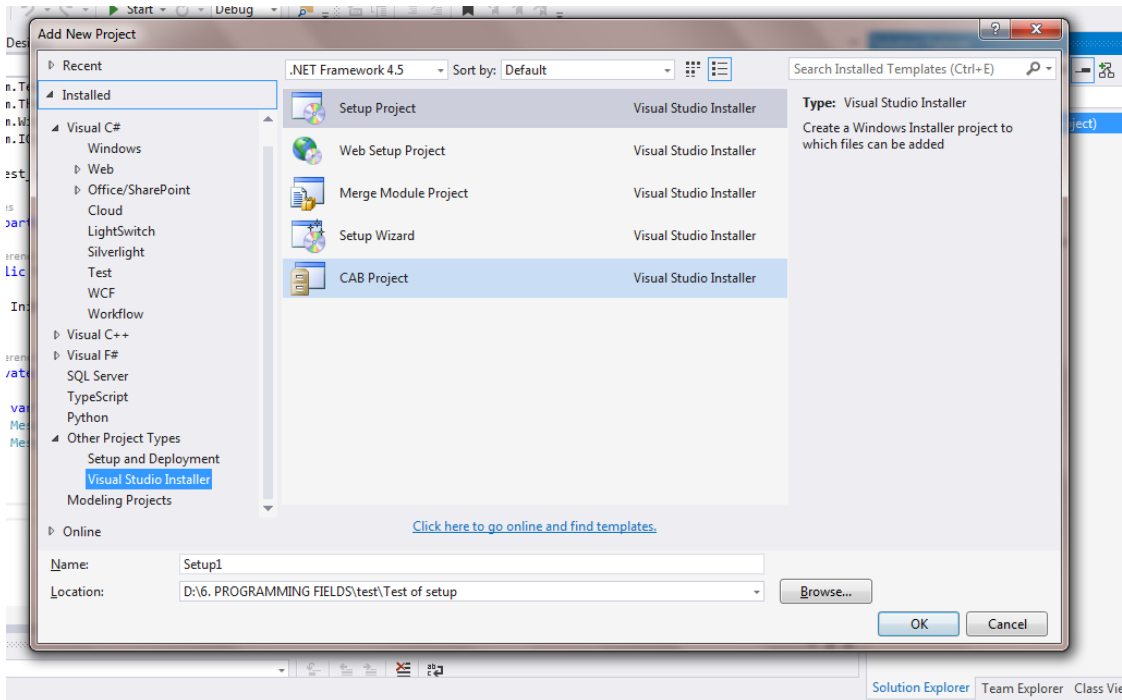
Với cơ sở dữ liệu SQL Server, để đảm bảo an toàn chúng ta không sử dụng user đăng nhập mặc định là sa mà nên tạo một user riêng, đồng thời tạo mật khẩu cho user này để đăng nhập cơ sở dữ liệu.

Đối với các user đăng nhập trong phần mềm cũng nên tạo mật khẩu mặc định không được quá đơn giản, tránh người sử dụng có thể đoán được. Đặc biệt không nên để trống.

5. Đóng gói phần mềm

B1. Tạo Setup project

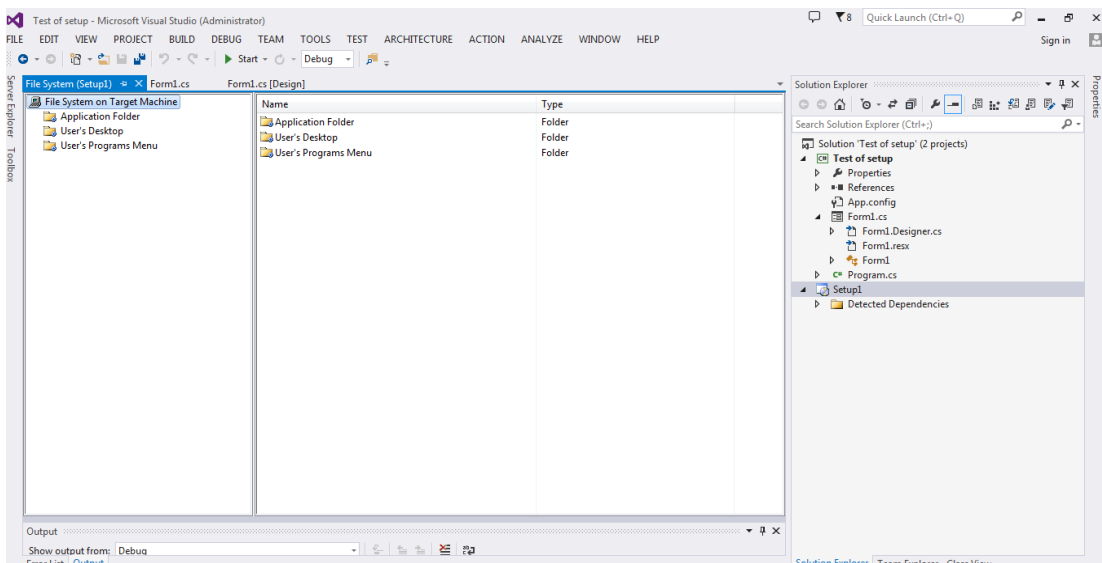
Bây giờ chúng ta sẽ tạo Setup project cho chương trình vừa viết ở trên. Chuột phải vào Solution "Test of Setup" => Add => New Project => Other project type => Setup Project:



Chú ý: đối với những phiên bản VS 2012 trở đi, chúng ta phải lên mạng và tìm cách down thêm Visual Studio Installer và cài đặt thì mới xuất hiện Setup Project trong Other Project Type được. Tải Visual Studio Installer TẠI ĐÂY.

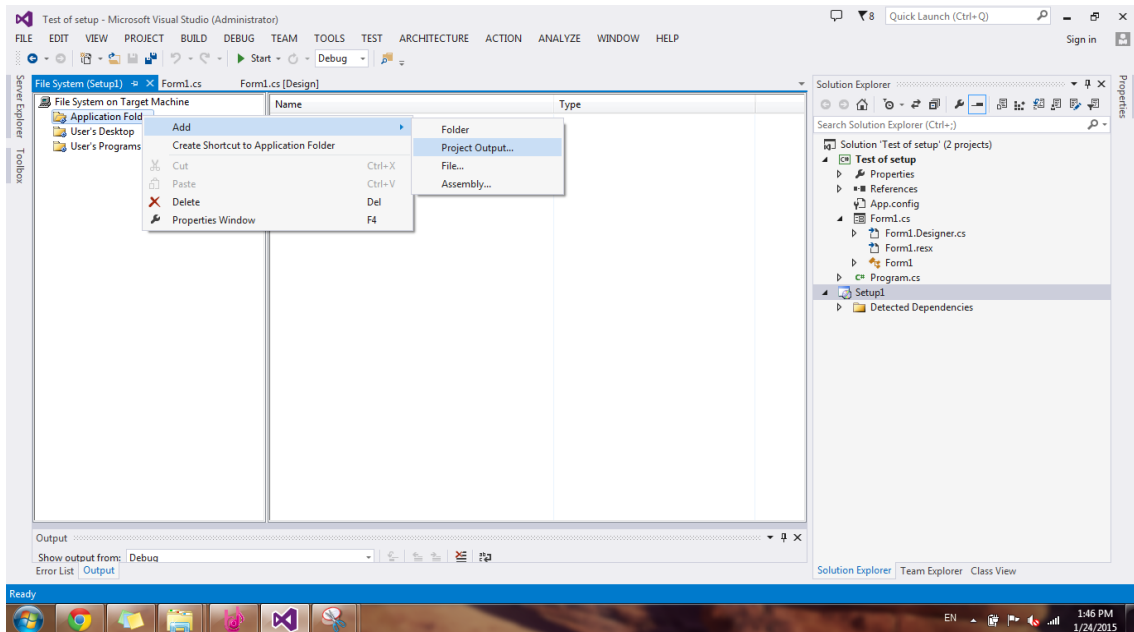
Tôi giữ nguyên tên là Setup1 cho nó đỡ lằng nhằng.

Kết quả, giao diện như sau:

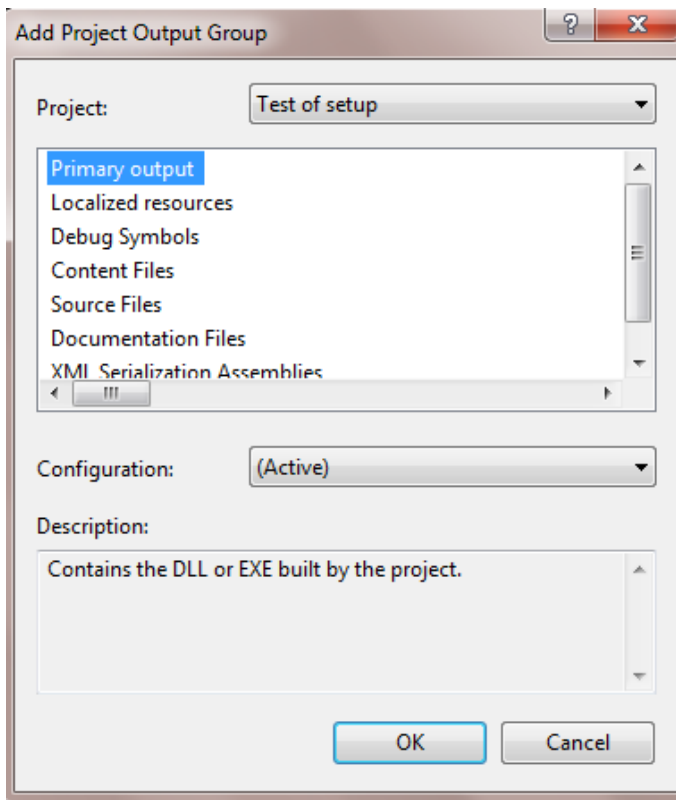


B2. Các bước cài đặt cơ bản

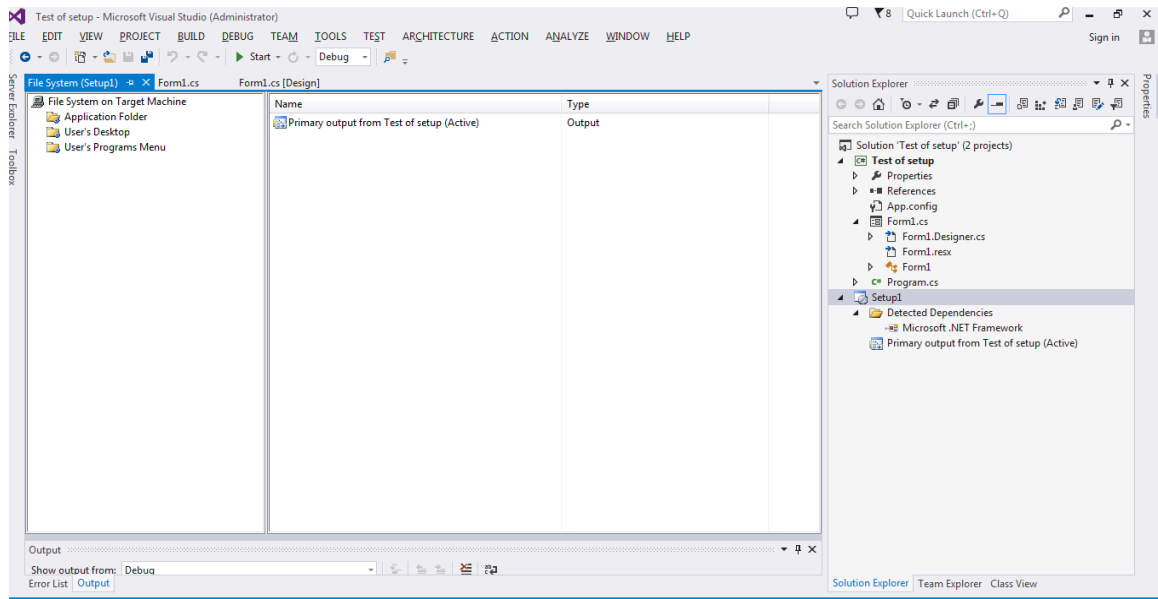
Bây giờ chúng ta cần tạo một Output cho setup. Chuột phải vào Application Folder => Add => Project Output... như bên dưới:



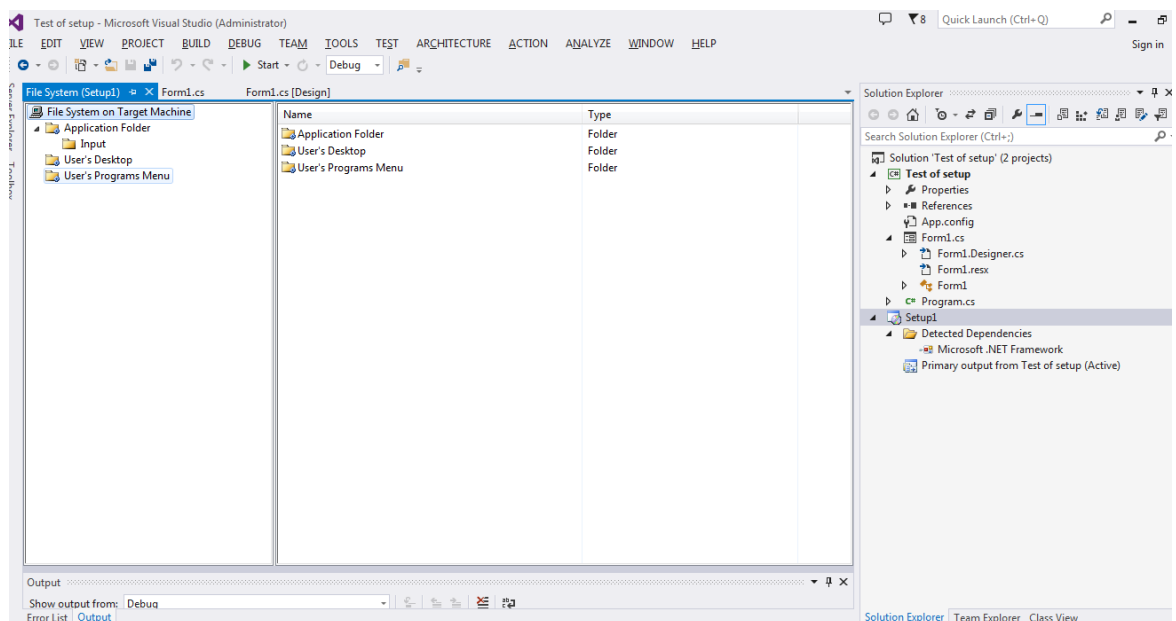
Sau đó chọn Primary Output => Ok:



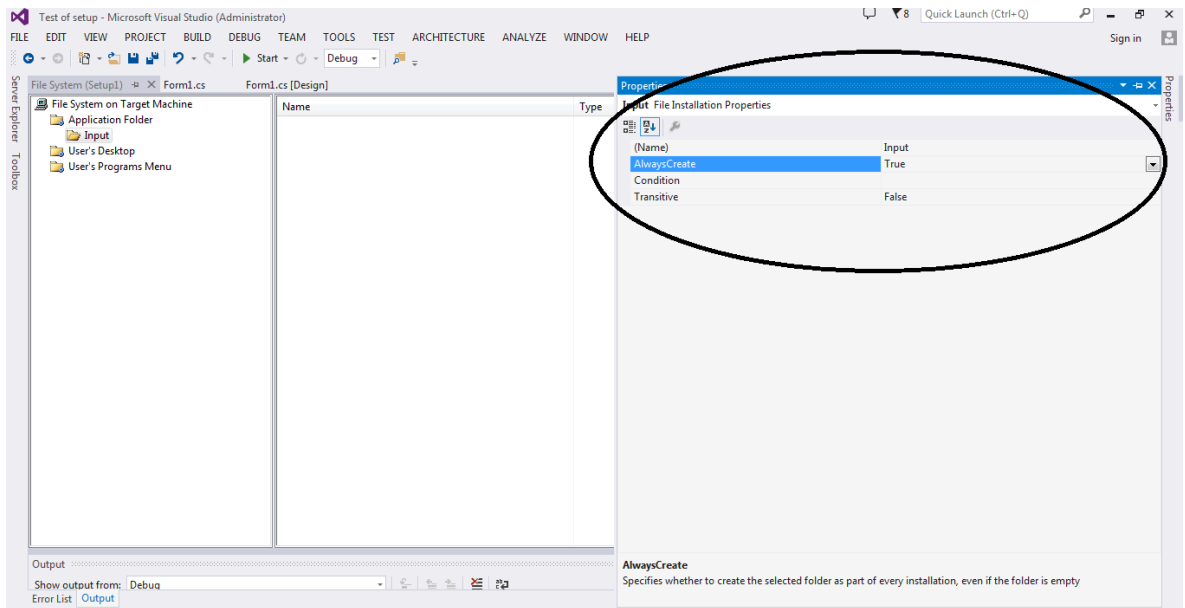
Và kết quả như sau:



Trong Project, chương trình của chúng ta sẽ phải lấy thông tin từ file Sample.txt để hiển thị ra ngoài màn hình. Do đó, file Sample.txt cũng phải đi kèm với file Setup luôn, có nghĩa là chúng ta cần phải add Sample.txt vào trong Setup Project (nếu không add thì chương trình của chúng ta sẽ bị lỗi). Để Add được thì chúng ta chuột phải vào Application Folder => Add => Folder => thế là một Folder mới xuất hiện, sửa thành Input, kết quả như sau:



Chuột phải vào thư mục Input => Properties => chọn thuộc tính AlwaysCreate là True:



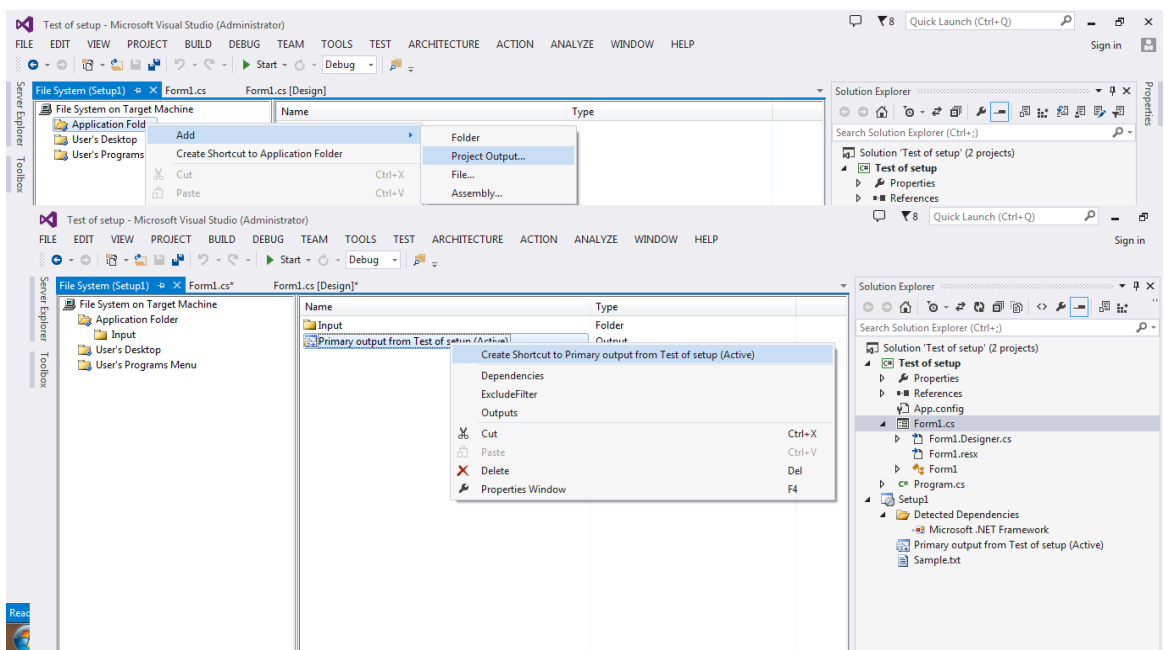
Làm như thế có nghĩa là: khi chạy file Setup thì chương trình luôn tạo một Folder tên là Input trong ổ C, còn nó ở chỗ nào thì ta chẳng quan tâm, chỉ cần biết khi mà cài xong thì chương trình cần dùng những gì từ thư mục Input đó thì nó sẽ tự biết tìm đến và lấy, thế thôi.

Chúng ta tiếp tục add Sample.txt vào trong thư mục Input để sau này khi phần mềm của mình đem sang máy khác cài thì nó còn biết đường mà tìm đến Input/Sample.txt.

Động tác Add như sau: chuột phải vào Input => Add => File => chọn đến file Sample.txt.

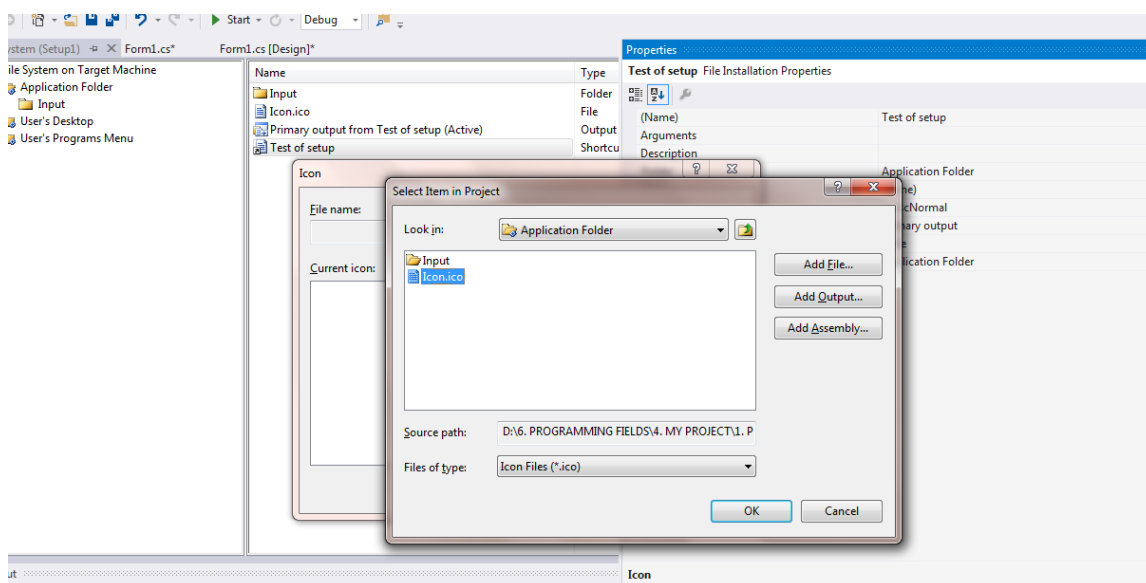
B3. Tạo Shortcut

Chuột phải vào Primary Output ở giữa màn hình và chọn Create Shortcut:



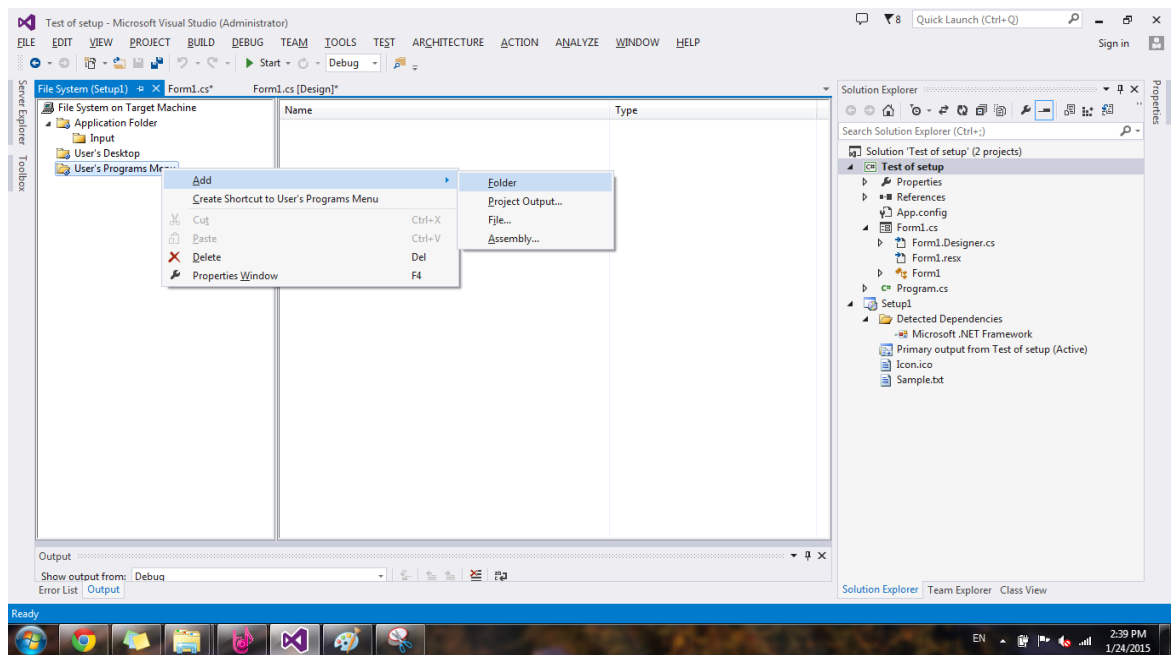
Sửa lại tên thành "Test of setup", cái shortcut này sẽ là cái xuất hiện trên nền desktop khi mà chúng ta cài đặt xong.

Bây giờ chúng ta cần thiết đặt Icon cho Shortcut: chuột phải vào Shortcut vừa mới tạo => Properties => trong cửa sổ Properties, chọn Icon => Browse => Browse => nháy kép vào Application Folder thì nút Add File sẽ có hiệu lực, bấm vào Add File => chọn đến một file ảnh có đuôi dạng .ico để làm Icon cho Shortcut, chọn được file .ico rồi thì bấm OK:



Sau đó chúng ta cắt cái Shortcut vừa tạo và paste vào trong thư mục User's Desktop

Tiếp theo, chúng ta sẽ quan tâm đến thư mục User Program Menu, nhìn cái tên thì biết ngay thư mục này liên quan đến các hoạt động ở Program Menu (cái Menu mà khi bấm Start thì hiện ra đấy). Bây giờ chúng ta sẽ Add thư mục và Short cut vào thư mục User Program Menu. Điều đó có nghĩa là khi mà chương trình của chúng ta được cài thì sau khi bấm Start, thư mục chương trình và shortcut chương trình của chúng ta sẽ nằm trên cả Program Menu nữa (như vậy trông nó mới chuyên nghiệp). Trước tiên là Add thư mục (tên thư mục thì tùy chọn, trong trường hợp này tôi cũng vẫn đặt tên là "Test of setup"), làm giống như add thư mục Input vào thư mục Application Folder:



Sau đó, chọn thư mục Test of setup vừa tạo, tiếp đó đưa chuột ra khung giữa chọn Create Shortcut (cái này là Shortcut nằm ở trên Program Menu), một hộp thoại sẽ ra, chúng ta nhấp kép vào thư mục Application Folder sau đó chọn Primary output from Test of setup (Active) và ấn OK. Sau đó đặt tên cho Shortcut đó là được.

TÀI LIỆU THAM KHẢO

- [1] Nguyễn Thị Ngọc Mai (2004), *Lập trình CSDL với Visual Basic 6.0*, NXB Lao động - Xã hội;
- [2] Đinh Xuân Lâm (2004), *Những bài thực hành Cơ sở dữ liệu Visual Basic căn bản*, NXB thống kê;
- [3] Nguyễn Ngọc Bình Phương – Nguyễn Quang Nam (2004), *Thủ thuật lập trình Visual basic 6.0*, NXB Giao thông vận tải.
- [4] Thuận Thành (2008), *Tự học lập trình ADO.Net*, NXB Thanh niên;
- [5] Nguyễn Tiên, Nguyễn Vũ Thịnh, Hồ Bích thủy (2009), *Professional ADO.NET lập trình và ứng dụng*, NXB Thống kê;
- [6] Dương Quang Thiện (2005), *.NET Toàn Tập - Tập 4: Lập trình căn cứ dữ liệu dùng ADO.NET*, NXB Thành phố Hồ Chí Minh;
- [7] Bill Hamilton (2007), *ADO.NET Cookbook*, O'Reilly.

