**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỒ ÁN LẬP TRÌNH C#**

**TRÊN WINDOWNS**

**ĐỀ TÀI**: QUẢN LÝ BÁN HÀNG ĐỒ PHƯỢT

**Giảng viên hướng dẫn :** ...Vũ Đình Ái....

**Lớp :**  15DTH09 ,15DTH10

**Họ tên sinh viên:** 1. LÊ PHƯƠNG THÀNH MSSV:

2.PHẠM PHÚ KHÔI MSSV: 1511060216

3.HỒ DUY THỌ MSSV: 1511060907

4.VÕ THỊ KIM THOA MSSV: 1511060646

5.NGUYỄN THỊ YẾN MSSV: 1511060709

Hà Nội - 2013

**MỤC LỤC**

[LỜI MỞ ĐẦU 1](#_Toc496650881)

[THÔNG TIN VỀ ĐỀ TÀI 2](#_Toc496650882)

[**I. TÊN ĐỒ ÁN** 2](#_Toc496650883)

[**II. NGƯỜI THỰC HIỆN** 2](#_Toc496650884)

[**III. YÊU CẦU CỦA ĐỒ ÁN.** 2](#_Toc496650885)

[CHƯƠNG I 3](#_Toc496650886)

[KHẢO SÁT HỆ THỐNG 3](#_Toc496650887)

[**I.1.MỤC TIÊU.** 3](#_Toc496650888)

[**I.2. KHẢO SÁT THỰC TẾ.** 3](#_Toc496650890)

[**I.3. YÊU CẦU CỦA HỆ THỐNG MỚI** 4](#_Toc496650892)

[**I.3.1. *PHÂN TÍCH YÊU CẦU NGHIỆP VỤ*** 4](#_Toc496650893)

[**I.3.2. *QUY TRÌNH NGHIỆP VỤ*** 4](#_Toc496650894)

[CHƯƠNG II 6](#_Toc496650895)

[CƠ SỞ LÝ THUYẾT 6](#_Toc496650896)

[**II.1. CƠ SỞ LÝ THUYẾT** 6](#_Toc496650897)

[***II.1.1 TỔNG QUAN VỀ .NET FRAMEWORK*** 6](#_Toc496650898)

[***II.1.2. MÔ HÌNH 3-TIERS VÀ 3-LAYER*** 14](#_Toc496650899)

[***II.1.3. NGÔN NGỮ C#*** 20](#_Toc496650900)

[CHƯƠNG III 22](#_Toc496650901)

[THIẾT KẾ HỆ THỐNG 22](#_Toc496650902)

[**III.1. THIẾT KẾ CƠ SỞ DỮ LIỆU** 22](#_Toc496650903)

[**III.2. THIẾT KẾ HỆ THỐNG** 22](#_Toc496650904)

[***III.2.1. CƠ SỞ DỮ LIỆU*** 22](#_Toc496650905)

[***III.2.2. MÔ HÌNH QUAN HỆ GIỮA CÁC BẢNG*** 26](#_Toc496650906)

[***III.2.3. CHỨC NĂNG TỪNG BẢNG*** 27](#_Toc496650907)

[***III.2.4. CÁC FORM GIAO DIỆN CỦA CHƯƠNG TRÌNH*** 29](#_Toc496650908)

[III2.4.1. Form Splash Screen 29](#_Toc496650909)

[III2.4.2. Form giao diện chính](#_Toc496650910) 30

[III.2.4.3.Form Khách hàng 32](#_Toc496650911)

[III.2.4.4. Form Nhà sản xuất 33](#_Toc496650912)

[III.2.4.5. Form Nhân viên 34](#_Toc496650913)

[III.2.4.6. Form Sản phẩm 35](#_Toc496650914)

[III.2.4.7. Form Hóa Đơn và Chi Tiết Hóa Đơn 36](#_Toc496650915)

[III. 2.4.8. Form tính tiền 37](#_Toc496650916)

[III.2.4.9. Frm Phiếu Nhập và Chi tiết phiếu nhập 38](#_Toc496650917)

[III.2.5.0. Các modules trong chương trình 39](#_Toc496650918)

[CHƯƠNG IV 40](#_Toc496650919)

[KẾT LUẬN 40](#_Toc496650920)

[**IV.1. ƯU ĐIỂM VÀ NHƯỢC ĐIỂM** 40](#_Toc496650921)

[***IV.1.1. ƯU ĐIỂM*** 40](#_Toc496650922)

[***IV.1.2. NHƯỢC ĐIỂM*** 40](#_Toc496650923)

[**IV.3. TÀI LIỆU THAM KHẢO** 40](#_Toc496650924)

[**IV.3. ĐÁNH GIÁ CỦA GIÁO VIÊN** 41](#_Toc496650925)

# 

# **LỜI MỞ ĐẦU**

Trong nền kinh tế thị trường của nước ta hiện nay, thông tin càng trở thành một yêu cầu bức thiết của các nhà quản lý, đòi hỏi thông tin phải đa dạng, phong phú, sâu sắc, chính xác càng trở nên bức xúc đối với mọi cấp quản lý. Chính vì vậy việc áp dụng tin học vào công tác quản lý sẽ tạo điều kiện xây dựng một hệ thống thông tin quản lý thoả mãn những tiêu chuẩn tối ưu nhất về phương diện dữ liệu và thao tác xử lý. Như vậy, việc áp dụng tin học vào công tác quản lý sẽ đáp ứng một cách toàn diện các yêu cầu về một hệ thống thông tin quản lý hoàn chỉnh. Mục đích cơ bản của việc xây dựng hệ thống quản lý bán hàng nhằm phục vụ cho việc mua bán của cửa hàng, nhằm cung cấp những thông tin đầy đủ, chính xác, nhanh chóng cho cửa hàng, cho khách hàng và đáp ứng được các yêu cầu của người tiêu dùng.

Như chúng ta đã biết, việc quản lý kinh doanh của rất nhiều cửa hàng nói chung và cửa hàng kinh doanh đồ phượt nói riêng vẫn tiến hành rất thủ công, việc ghi chép thông qua sổ sách gặp rất nhiều trở ngại, gây khó khăn cho người quản lý khi muốn xem xét tình trạng các mặt hàng còn hay hết. Cũng như rất khó có thể thống kê nhanh chóng được doanh thu của cửa hàng theo ngày ,theo tháng, theo quý …Từ thực tế như vậy mà em đã tiến hành xây dựng một phần mềm quản lý hoạt động kinh doanh cho mặt hàng đồ phượt.

Hệ thống là một chương trình phần mềm cho phép người sử dụng thực hiện một cách nhanh chóng, chính xác việc nhập dữ liệu, lưu trữ cập nhật thông tin các mặt hàng, thực hiện yêu cầu báo cáo bán hàng hay báo cáo tài chính của cửa hàng …với giao diện làm việc thân thiện ,tiện dụng đối với người sử dụng hệ thống.

Mong rằng với phần mềm này nhóm em có thể đóng góp được một phần nào đó vào sự phát triển kinh doanh của các mặt hàng phượt nói riêng và các mặt hàng kinh doanh khác nói chung.

Với sự cố gắng và nổ lực của nhóm, nhóm em đã xây dựng và hoàn thành đề tài này. Xong vẫn còn nhiều thiếu xót nhóm em rất mong nhận được sự đóng góp ý kiến của các thầy để cho đề tài của nhóm em được hoàn thiện hơn .

Nhóm em xin chân thành cảm ơn !

**THÔNG TIN VỀ ĐỀ TÀI**

**I. TÊN ĐỒ ÁN**

***QUẢN LÝ CÁC MẶT HÀNG PHƯỢT***

**II. NGƯỜI THỰC HIỆN**

1.LÊ PHƯƠNG THÀNH

2.PHẠM PHÚ KHÔI

3.HỒ DUY THỌ

4.VÕ THỊ KIM THOA

5.NGUYỄN THỊ YẾN

**III. YÊU CẦU CỦA ĐỒ ÁN.**

* Chương trình có khả năng lưu trữ cập nhập thông tin về các loại mặt hàng có trong cửa hàng.
* Cho phép tìm kiếm thông tin về mặt hàng .
* Thống kê được doanh thu và lãi xuất của cửa hàng (theo ngày, tháng, quý)
* Hệ thống mang tính bảo mật .
* Chương trình chạy trên môi trường WinDows.
* Các thao tác thực hiện công việc phải dễ dàng, thuận tiện.
* Font chữ hiển thị nhất quán
* Máy có cấu hình vừa phải cũng chạy được
* Chủ cửa hàng được toàn quyền với hệ thống .

**CHƯƠNG I**

**KHẢO SÁT HỆ THỐN****G**

**I.1.MỤC TIÊU.**

Việc xây dựng hệ thống quản lý bán hàng nhằm phục vụ cho việc mua bán của cửa hàng, nhằm cung cấp những thông tin đầy đủ, chính xác, nhanh chóng cho cửa hàng, cho khách hàng và đáp ứng được các yêu cầu của người tiêu dùng.

**I.2. KHẢO SÁT THỰC TẾ.**

Dựa trên quá trình khảo sát các hệ thống cửa hàng hiện tại và sự phản ánh của khách hàng ta thấy rằng hệ thống cửa hàng cũ có những đặc điểm sau:

* Các quy tắc quản lý cửa hàng:
* Quản lý hàng nhập về: quản lý Mã SP, Tên SP, Nhà cung cấp, Số lượng, Đơn giá
* Quản lý hàng bán ra: quản lý Mã SP, Tên SP, Số lượng, Đơn giá, Tổng tiền.
* Quản lý tài chính: Quản lý tổng chi, tổng thu, lãi suất thu được hàng tháng ( hàng quý, hàng năm )

Mặt hàng bán chạy nhất là mặt hàng bán được với số lượng nhiều nhất.

Hàng bán chậm nhất là hàng bán được nhỏ hơn 1/5 số lượng nhập về

* Hầu hết các cửa hàng đều sử dụng phương pháp thủ công để quản lý cửa hàng trong các khâu như: Lưu trữ, thống kê, nhập xuất và tìm kiếm... , nên tốn rất nhiều thời gian và công sức mà hiệu quả đạt được không cao, đôi khi dẫn đến những sai lầm không đáng có. Cụ thể
* Khi khách hàng đến mua hàng thì người chủ cửa hàng phải kiểm tra xem mặt hàng đó có còn hay không và kiểm tra giá cả mặt hàng đó.
* Trong quá trình thống kê, do lưu trữ các thông tin chủ yếu bằng sổ sách nên khó khăn trong việc tổng hợp các thông tin.
* Khi muốn nhập thêm hoặc thay đổi thông tin về hàng hóa thì việc cập nhật lại và lưu trữ rất phức tạp đôi khi sổ sách còn bị thất lạc nên những thông tin quang trọng có thể mất đi, gây ra khó khăn cho việc kiểm tra và giám sát sau này .

Đó là những thực tế còn tồn tại trong việc quản lý cửa hàng bằng phương pháp thủ công hiện nay. Với sụ phát triển của công nghệ hiện nay tại sao ta lại không ứng dụng nó vào trong thực tế này.

**I.3. YÊU CẦU CỦA HỆ THỐNG MỚI**

**I.3.1. *PHÂN TÍCH YÊU CẦU NGHIỆP VỤ***

* Do sự phát triển của nền kinh tế hàng hóa. nên phương pháp quản lý cửa hàng kiểu này không còn phù hợp nữa. Vì vậy yêu cầu đặt ra cần có một phương pháp quản lý mới để quản lý cửa hàng một cách tốt nhất đáp ứng được nhu cầu:
* Hệ thống cho phép cập nhâp và lưu trữ tất cả các thông tin liên quan đến cửa hàng một cách nhanh gọn, và chính xác.
* Cho phép tìm kiếm thông tin về hàng hóa một cách nhanh nhất đáp ứng như cầu của người sử dụng.
* Công việc thống kê chính xác và nhanh chóng
* Hệ thống giúp phân quyền cho người sử dụng nên mang tính bảo mật. Tránh được sự xâm nhập phá hoại từ bên ngoài.
* Hệ thống sử dụng trong viêc quản lý hàng hóa cho một cửa hàng vừa và nhỏ.

**I.3.2. *QUY TRÌNH NGHIỆP VỤ***

* Quy trình nhập hàng:

Bước 1: Nhân viên nhập hàng sẽ điền đầy đủ thông tin của sản phẩm

Bước 2: Kiểm tra các thông số kỹ thuật, chất lượng của sản phẩm

Bước 3: Tiến hành nhập sản phẩm vào hệ thống thông qua phần mềm quản lý bán hàng.

* Quy trình bán hàng

Bước 1: Nhân viên bán hàng sẽ tư vấn, hỗ trợ khách hàng thông tin về sản phẩm.

Bước 2: Sau khi khách hàng đã chọn được sản phẩm ưng ý, nhân viên bán hàng sẽ tiến hành lập hóa đơn với những thông tin của sản phẩm mà khách hàng lựa chọn.

Bước 3: Kiểm tra số lượng sản phẩm mà khách hàng yêu cầu trong kho hàng có đáp ứng được không? .Nếu còn đủ hàng thì tiến hành thêm hóa đơn và chuyển  xuống bộ phận giao hàng cho khách hàng. Ngược lại thông báo cho khách hàng và tick vào ô đã hết hàng để gửi 1 thông báo cho bộ phận nhập hàng biết.

**CHƯƠNG II**

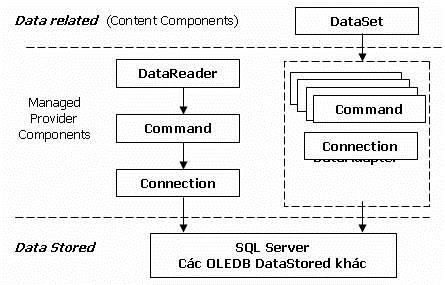
**CƠ SỞ LÝ THUYẾT**

**II.1. CƠ SỞ LÝ THUYẾT**

***II.1.1 TỔNG QUAN VỀ .NET FRAMEWORK***

***II.1.1.1*. Ado. Net:**

* ADO.NET là :
* ADO.NET là công cụ chứa các lớp, đối tượng cho phép truy cập, thao tác với cơ sở dữ liệu
* Viết tắt của ActiveX Data Object .NET (ADO.NET)
* ADO.NET là công nghệ của MS trên .NET Framework
* Được phát triển từ nền tảng ADO trên nền VB 6.0
* Cung cấp các lớp đối tượng và hàm thư viện phục vụ cho việc kết nối và xử lý dữ liệu
* Ưu điểm :
* ADO.NET được thiết kế làm việc với cả dữ liệu phi kết nối trong môi trường đa tầng (Muti-Tier)
* dễ dàng giao tiếp với các ứng dụng
* dễ dàng phát triển bằng nhiều ngôn ngữ khác nhau
* phép thực hiện mọi thao tác tới CSDL
* không làm tăng tải trên Server
* Tích hợp chặt chẽ với XML
* Tương tác với nhiều nguồn dữ liệu thông qua mô tả dữ liệu chung
* Tối ưu truy cập nguồn dữ liệu (OLE DB & SQL server)
* Nhược điểm :
* Tiêu tốn bộ nhớ vì phải lưu bản copy từ CSDL
* Kiến trúc ADO .Net :



Hình 1: Kiến trúc Ado.net

Kiến trúc ADO.NET có thể chia làm 2 phần chính:

* Managed Provider Component: bao gồm các đối tượng như DataAdapter, DataReader,… giữ nhiệm vụ làm việc trực tiếp với dữ liệu như database, file,…
* Content Component: bao gồm các đối tượng như DataSet, DataTable,… đại diện cho dữ liệu thực sự cần làm việc. DataReader là đối tượng mới, giúp truy cập dữ liệu nhanh chóng nhưng forward-only và read-only giống như ADO RecordSet sử dụng Server cursor, Open Forward Only.

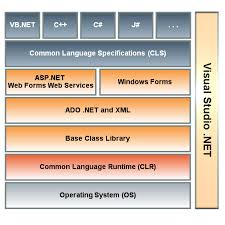
DataSet cũng là một đối tượng mới, không chỉ là dữ liệu, DataSet có thể coi là một bản sao gọn nhẹ của CSDL trong bộ nhớ với nhiều bảng và các mối quan hệ.   
DataAdapter là đối tượng kết nối giữa DataSet và CSDL, nó bao gồm 2 đối tượng Connection và Command để cung cấp dữ liệu cho DataSet cũng như cập nhật dữ liệu từ DataSet xuống CSDL.

* Quy trình kết nối đến CSDL :
* Tạo đối tượng Connection và xác định chuỗi kết nối (Connection String)
* Tạo đối tượng Command và xác định câu lệnh SQL
* Mở đối tượng Connection (Open)
* Thực hiện câu lệnh SQL và xử lý kết quả (Execute)
* Đóng đối tượng Connection (Close)

**II.1.1.2. Kiến trúc .NET Framework**

.NET framework là một platform mới làm đơn giản việc phát triển ứng dụng trong môi trường phân tán của Internet.

* Chương trình nền tảng cho công nghệ .NET
* Cung cấp tập hợp class library thường dùng
* Theo quan điểm của người lập trình, .NET có thể hiểu như môi trường thực thi mới và thư viện lớp cơ sở cải tiến.
* Môi trường thực thi là: Common Language Runtime- CLR. Vai trò chính CLR: locate, load, manage.NET types.



Hình 2: Kiến trúc .NET Framework

* Các thành phần chính của .NET Framework là:
* Common Language Runtime(CLR)
* Framework  Base Classes Library.
* Các phiên bản .NET Framework:
* .NET Framework 1.0 \_Visual Studio 2002
* .NET Framework 1.1 \_Visual Studio 2003
* .NET Framework 2.0 \_Visual Studio 2005
* .NET Framework 3.0 \_Visual Studio 2005
* .NET Framework 3.5 \_Visual Studio 2008
* .NET Framework 4.0 \_Visual Studio 2010
* .NET Framework 4.5 \_Visual Studio 2012

**II.1.1.3.Common Language Runtime(CLR)**

* CRL là môi trường thực thi cho ngôn ngữ chung và các dịch vụ trong quá trình thực thi.
* Ví dụ một số dịch vụ trong CRL
* Biên dịch
* Quản lý bộ nhớ
* Quản lý thread
* Bảo mật
* Kiểm tra an toàn kiểu

   Common Language Runtime trong .NET Framework quản lý sự thực hiện mã và cung cấp sự truy cập vào nhiều loại dịch vụ giúp cho quá trình phát triển được dễ dàng hơn. CLR đã được phát triển ở tầm cao hơn so với các runtime trước đây như VB-runtime chẳng hạn, bởi nó đạt được những khả nǎng như tích hợp các ngôn ngữ, bảo mật truy cập mã, quản lý thời gian sống của đối tượng và hỗ trợ gỡ lỗi.

* .NET Framework – Common Type System (CST)
* Mục đích hỗ trợ thực thi chéo ngôn ngữ
* Tất cả compiler hướng .NET đều phải tuân thủ theo CST
* Định nghĩa kiểu dữ liệu tiền định và có sẵn trong IL: tất cả ngôn ngữ .NET sẽ được sinh ra mã cuối trên cơ sở kiểu dữ liệu này.

**II.1.1.4. Framework Base Classes Library**

Đây là một thư viện lớp rất đồ sộ chứa những code  đã viết sẵn, cung cấp một

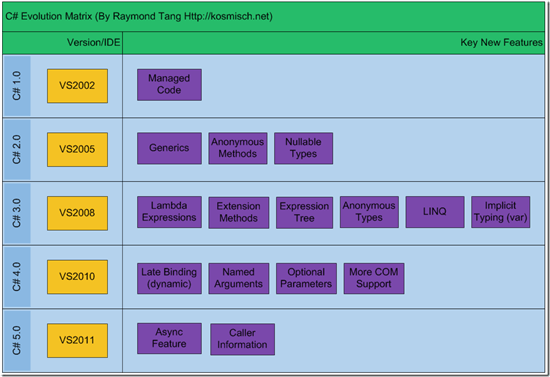
 API thiên đối tượng bao gồm tất cả các chức năng được bao trùm bởi .NET  Framework. Với hơn 5000 lớp khác nhau, Framework Classes Library cung

cấp những tiện nghi giúp triển khai nhanh chóng cácứng dụng trên desktop, client server, các dịch vụ web và các ứng dụng khác.

**II.1.1.5. Các phiên bản của .NET Framework**

Hình 3. Sự phát triển của .NET Framework

* .NET 1.1
* Regular Expressions là: Biểu thức chính quy cung cấp một kỹ thuật mạnh mẽ, linh hoạt và rất hiệu quả cho việc xử lý chuỗi. Bất cứ khi nào bạn cần lấy một chuỗi con từ một chuỗi cho trước, hãy nghĩ ngay đến việc sử dụng biểu thức chính quy. Lý thuyết về biểu thức chính quy trông có vẻ phức tạp lúc mới tiếp cận nhưng nó thực sự rất đáng giá khi nắm được. Chúng ta sẽ nghiên cứu về biểu thức chính quy trong các bài viết sau.
* XML Processing là: Lớp *XmlDocument*trong namespace *System.Xml* cung cấp các API để phân tích và đọc các tài liệu XML. Nó cũng cung cấp ngôn ngữ XPath dùng để trích xuất dữ liệu từ tài liệu XML.
* *HttpUtility*là: một lớp rất quan trọng dùng để mã hóa và giải mã tham số URL khi xử lý các web request. Đặc biệt nó có một phương thức vô cùng hữu dụng – *HtmlDecode*dùng để thay các thẻ HTML bằng các kí tự Unicode tương ứng (phòng tránh các lỗi bảo mật như XSS, SQL Injection…)
* String Builder là: một lớp hữu dụng khi xử lý việc kết hợp nhiều chuỗi (thường khi kết hợp lớn hơn 3 chuỗi), đặc biệt là khi kết hợp chuỗi trong vòng lặp lớn bởi vì kiến trúc bên trong String Builder dùng mảng kí tự và đã được Microsoft tối ưu hóa về hiệu năng.
* .NET 2.0
* Generics là: các thuật toán Generic được trừu tượng hóa và yêu cầu thành phần kiểu. Thành phần này có thể được mô tả bởi interface hoặc lớp trừu tượng (abstract class). VD: Generic *IEnumerable<T>* là một tập các object kiểu T. Generic rất hữu ích khi chúng ta cần viết những lớp chung như lớp base hoặc các phương thức dùng chung an toàn về kiểu.
* Event Log là: hệ thống ghi nhật ký thống nhất trên Windows. Thay vì quản lý việc tạo ra và xóa các file log hoặc lọc và phân loại dữ liệu log thì chúng ta có thể sử dụng lớp *EventLog* với các API được cấp sẵn để thay thế.
* Configuration Manager là: lớp *ConfigurationManager*chứa chuỗi kết nối và các hằng số bạn muốn thay đổi lúc chạy ứng dụng mà không phải biên dịch lại ứng dụng. VD: Trong ứng dụng Winform thì là file App.config còn ứng dụng ASP.NET là web.config.
* Nullable Types là: kiểu Nullable trong .NET có 2 kiểu dữ liệu là kiểu giá trị (Value Type) và kiểu tham chiếu (Reference Type). Kiểu giá trị là các kiểu cơ sở của .NET: int, long, byte, bool, short, char, structure… Kiểu tham chiếu là các đối tượng. Kiểu tham chiếu với giá trị null mang ý nghĩa là không chỉ đến đối tượng nào. Với kiểu giá trị có trường hợp khi lấy dữ liệu từ SQL Server một trường BIT có thể mang giá trị là true, false hoặc undefined vì vậy cần kiểu nullable để tương thích.NET 3.0
* XAML (Silverlight/WPF/Windows Phone/Windows RT) : cho phép chúng ta vẽ các giao diện ứng dụng giống như sử dụng HTML. Cách vẽ giao diện qua việc khai báo thẻ như HTML có khá nhiều ưu điểm mà nổi bật trong số đó chính là binding – một thành phần không thể thiếu của mô hình phát triển ứng dụng MVVM. Bên cạnh đó XAML được tăng tốc bởi GPU nên hiệu năng của ứng dụng khá tốt. XAML được áp dụng để thiết kế giao diện chính cho hầu hết các loại ứng dụng của Microsoft như: WPF/Silverlight/Windows Phone/Windows RT.
* WCF services là: web service giúp cho việc gọi các phương thức từ server trở nên dễ dàng hơn cho client. Trong .NET 3.0 Web services được chuẩn hóa thành WCF services với nhiều tính năng cao cấp hơn so với web service thông thương trong .NET 2.0
* Workflow là: một mô hình các hành động (activity). Các hoạt động này có thể là bất kỳ thứ gì từ tiến trình nghiệp vụ cho đến tiến trình build ứng dụng của Team Foundation Server. Ưu điểm của Workflow là các hành động được định nghĩa bởi XAML nên nó dễ dàng thay đổi bởi file cấu hình mà không cần phải biên dịch lại ứng dụng.
* Feed processing là: Lớp SyndicationFeed (nguồn cấp dữ liệu) hỗ trợ định dạng syndication RSS 2.0 và Atom 1.0.
* .NET 3.5
* Language Integrated Query (LINQ): là một trong những ngôn ngữ cách mạng cho việc xử lý dữ liệu, nó giống như SQL cho lập trình hướng đối tượng. Tất cả những store procedure được viết trong SQL Server thì giờ chúng ta hoàn toàn có thể viết trên code .NET, hết sức tiện lợi trong việc viết cũng như debug. Tuy nhiên nhược điểm khi sử dụng LINQ là hiệu năng sẽ thấp hơn so với khi sử dụng store procedure.
* Entity Framework là một thành phần không thể thiếu khi sử dụng LINQ. Một framework giúp chúng ta có thể mapping được ngay các bảng trong SQL server thành các đối tượng .NET để LINQ có thể sử dụng được.
* XML processing là: với sự xuất hiện của LINQ và lớp *Xdocument*trong namespace *System.Xml.Linq*việc xử lý XML trở nên dễ dàng, tiện lợi với hiệu năng cao hơn.
* Reactive Extensions (Rx)là: một thư viện dùng tạo các chương trình asynchronous dựa trên sự kiện và sử dụng oservable collection + LINQ để truy vấn. Rx được tạo ra để làm việc với data stream.
* .NET 4.0
* Managed Extensibility Framework (MEF) là: cho phép sử dụng các assembly (.dll) như là các module. Khi bạn cần tạo ra các hệ thống module hóa MEF sẽ đảm bảo việc kết nối giữa các module này.
* Parallel Extensions (PE) bao gồm Parallel LINQ (PLINQ) và Task Parallel Library (TPL). Khi bạn có nhiều việc cần làm thì có thể chọn hướng tiếp cận dựa trên task. Chia nhỏ công việc thành các việc nhỏ hơn và chạy chúng khi cần. Có thể chạy tuần tự hoặc song song mà không cần quan tâm đến việc quản lý thread. Lớp *Parallel* sẽ đảm nhận việc đếm xem có bao nhiêu bộ vi xử lý để chọn số thread hiệu quả nhất.
* Dynamic Language Runtime là: kiểu *dynamic* khi cần tương tác với môi trường thiếu kiểu, ngôn ngữ động… VD: Nó sẽ rất hữu ích khi tương tác với môi trường COM vì nó không cung cấp IntelliSense
* Lazy initialization là: khi chương trình chứa đối tượng chiếm nhiều thời gian để khởi tạo mà đối tượng này có thể không được dùng trong suốt chương trình thì chúng ta có thể dùng khởi tạo muộn *Lazy<T>,*đối tượng sẽ chỉ được khởi tạo khi truy cập nó lần đầu tiên
* .NET 4.5
* Asynchronous Pattern là: lập trình bất đồng bộ có thể thực hiện từ các phiên bản .NET trước nhưng phải đến .NET 4.5 với sự xuất hiện của async/await thì nó mới thực sự trở nên đơn giản với lập trình viên. Với .NET trước đây khi thread đợi server trả lời, nó không làm gì những vẫn tiêu tốn tài nguyên máy. Với async/await điều này không còn xảy ra vì nó thực sự không tạo ra thread mới khi thực hiện các lời gọi async.



Hình 4. .NET 4.5

***II.1.2. MÔ HÌNH 3-TIERS VÀ 3-LAYER***

**II.1.2.1. Mô hình 3 tầng (3-tiers)**

3-tiers là một kiến trúc kiểu client/server mà trong đó giao diện người dùng (UI-user interface), các quy tắc xử lý(BR-business rule hay BL-business logic), và việc lưu trữ dữ liệu được phát triển như những module độc lập, và hầu hết là được duy trì trên các nền tảng độc lập, và mô hình 3 tầng (3-tiers) được coi là một kiến trúc phần mềm và là một mẫu thiết kế đây là triển khai ứng dụng ở mức vật lý .Kiến trúc gồm 3 module chính và riêng biệt :

* Tầng Presentation: hiển thị các thành phần giao diện để tương tác với người dùng như tiếp nhận thông tin, thông báo lỗi, …
* Tầng Business Logic: thực hiện các hành động nghiệp vụ của phần mềm như tính toán, đánh giá tính hợp lệ của thông tin, … Tầng này còn di chuyển, xử lí thông tin giữa 2 tầng trên dưới.
* Tầng Data: nơi lưu trữ và trích xuất dữ liệu từ các hệ quản trị CSDL hay các file trong hệ thống. Cho phép tầng Business logic thực hiện các truy vấn dữ liệu .

Mọi người vẫn hay nhầm lẫn giữa tier và layer vì cấu trúc phân chia giống nhau (presentation, bussiness , data). Tuy nhiên, thực tế chúng hoàn toàn khác nhau. Nếu 3 tiers có tính vật lí thì 3 layer có tính logic. Nghĩa là ta phân chia ứng dụng thành các phần (các lớp) theo chức năng hoặc vai trò một cách logic. Các layer khác nhau được thực thi trong 1 phân vùng bộ nhớ của process. Vì thế nên một tier có thể có nhiều layer.

* Những ưu và nhược điểm

3-tiers là một kiến trúc phần mềm, có nghĩa là bạn có thể dùng nó để xây dựng nên bộ khung tổng thể của ứng dụng. Tuy nhiên bạn cần chú ý những ưu và nhược điểm sau đây để áp dụng nó một cách đúng đắn.

* Ưu điểm:
* Dễ dàng mở rộng, thay đổi quy mô của hệ thống: Khi cần tải lớn, người quản trị có thể dễ dàng thêm các máy chủ vào nhóm, hoặc lấy bớt ra trong trường hợp ngược lại.
* Nhược điểm:
* Việc truyền dữ liệu giữa các tầng sẽ chậm hơn vì phải truyền giữa các tiến trình khác nhau (IPC), dữ liệu cần phải được đóng gói -> truyền đi

-> mở gói trước khi có thể dùng được. Việc phát triển ứng dụng phức tạp hơn.

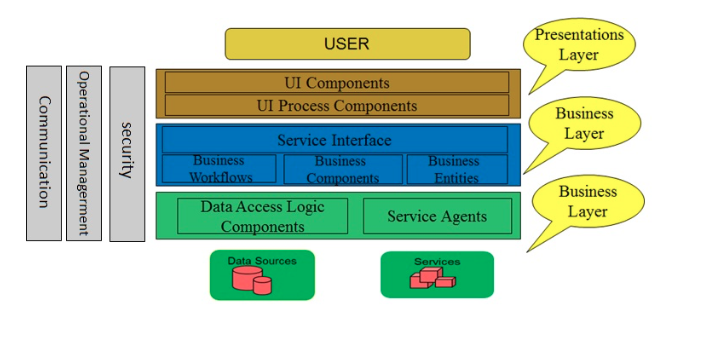
* Những công nghệ nào hỗ trợ xây dựng các ứng dụng 3-tiers

Tùy thuộc vào nền tảng, bạn có thể chọn một trong các công nghệ như EJB (J2EE), COM+ (Windows), hay cũng có thể dùng các máy chủ web như nền tảng xây dựng lớp giữa (dùng webservice). Tuy nhiên, EJB và COM+ là hai tùy chọn tốt nhất vì nó có nhiều công nghệ hỗ trợ như Object Pooling, Authentication và Authority, Resource management, Remote Object Access, Transaction…

Các công nghệ truyền thông điệp như JMS hay MSMQ cũng hỗ trợ nhiều trong việc tạo các lời gọi không đồng bộ.

**II.1.2.2. Giới thiệu mô hình 3-layer ( 3 lớp)**

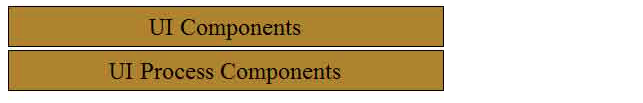
Layer là ta phân chia ứng dụng thành các phần (các lớp) theo chức năng hoặc vai trò một cách logic. Các layer khác nhau được thực thi trong 1 phân vùng bộ nhớ của process.

Hình 5. Mô hình layer

* Mô hình 3-layer gồm có 3 phần chính :
  + Presentation Layer (GUI) : Lớp này có nhiệm vụ chính giao tiếp với người dùng. Nó gồm các thành phần giao diện ( win form, web form,…) và thực hiện các công việc như nhập liệu, hiển thị dữ liêu, kiểm tra tính đúng đắn dữ liệu trước khi gọi lớp Business Logic Layer (BLL)
  + Business Logic Layer (BLL) : Layer này phân ra 2 thành nhiệm vụ :
* Đây là nơi đáp ứng các yêu cầu thao tác dữ liệu của GUI layer, xử lý chính nguồn dữ liệu từ Presentation Layer trước khi truyền xuống Data Access Layer và lưu xuống hệ quản trị CSDL.
* Đây còn là nơi kiểm tra các ràng buộc, tính toàn vẹn và hợp lệ dữ liệu, thực hiện tính toán và xử lý các yêu cầu nghiệp vụ, trước khi trả kết quả về Presentation Layer.
* Data Access Layer (DAL) : Lớp này có chức năng giao tiếp với hệ quản trị CSDL như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu ( tìm kiếm, thêm, xóa, sửa,…).
* Ưu điểm:
* Việc phân chia thành từng lớp giúp cho code được tường minh hơn. Nhờ vào việc chia ra từng lớp đảm nhận các chức năng khác nhau và riêng biệt như giao diện, xử lý, truy vấn thay vì để tất cả lại một chỗ. Nhằm giảm sự kết dính.
* Dễ bảo trì khi được phân chia, thì một thành phần của hệ thống sẽ dễ thay đổi. Việc thay đổi này có thể được cô lập trong 1 lớp, hoặc ảnh hưởng đến lớp gần nhất mà không ảnh hưởng đến cả chương trình.
* Dễ phát triển, tái sử dụng: khi chúng ta muốn thêm một chức năng nào đó thì việc lập trình theo một mô hình sẽ dễ dàng hơn vì chúng ta đã có chuẩn để tuân theo. Và việc sử dụng lại  khi có sự thay đổi giữa hai môi trường ( Winform sang Webfrom ) thì chỉ việc thay đổi lại lớp GUI.
* Dễ bàn giao. Nếu mọi người đều theo một quy chuẩn đã được định sẵn, thì công việc bàn giao, tương tác với nhau sẽ dễ dàng hơn và tiết kiệm được nhiều thời gian.
* Dễ phân phối khối lượng công việc. Mỗi một nhóm, một bộ phận sẽ nhận một nhiệm vụ trong mô hình 3 lớp. Việc phân chia rõ ràng như thế sẽ giúp các lập trình viên kiểm soát được khối lượng công việc của mình.
* Nhược điểm:
* Phải chia thành nhiều lớp trung gian nên việc sử lý dữ liệu chậm hơn so với kiểu một lớp truyền thống

**II.1.2.3. Phân tích chi tiết từng layer trong mô hình 3 lớp .**

* **Presentation Layer (GUI):**

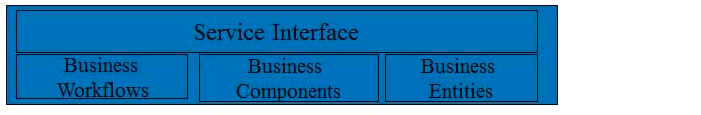


Có hai thành phần chính sau đây với những tác vụ cụ thể :

* UI Components : gồm các thành phần tạo nên giao diện của ứng dụng (GUI). Chúng chịu trách nhiệm thu nhận và hiển thị dữ liệu cho người dùng…

Ví dụ : textbox, button, combobox, …

* UI Process Components : là thành phần chịu trách nhiệm quản lý các quá trình chuyển đổi giữa các UI… Ví dụ : Sắp xếp quá trình kiểm tra thông tin khách hàng:
* 1.Hiển thị màn hình tra cứu ID
* 2.Hiển thị màn hình thông tin chi tiết khách hàng tương ứng
* 3.Hiển thị màn hình liên lạc với khách hàng.
* **Bussiness Layer (BLL) :**



Lớp này gồm 4 thành phần :

* Service Interface : là thành phần giao diện lập trình mà lớp này cung cấp cho lớp Presentation sử dụng.
* Bussiness Workflows : chịu trách nhiệm xác định và điều phối các quy trình nghiệp vụ gồm nhiều bước và kéo dài. Những quy trình này phải được sắp xếp và thực hiện theo một thứ tự chính xác.

Ví dụ : Thực hiện mua một đơn hàng trên tiki qua nhiều bước : kiểm tra gói hàng còn không?, tính tổng chi phí, cho phép giao dịch và sắp xếp việc giao hàng.

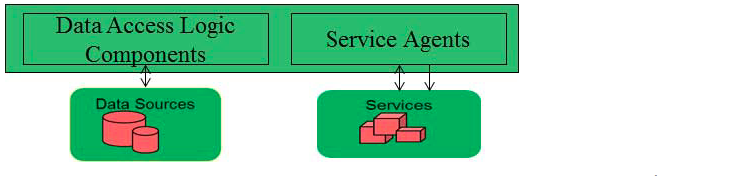
* Bussiness Components : chịu trách nhiệm kiểm tra các quy tắc nghiệp vụ, ràng buộc logic và thực hiện các công việc . Các thành phần này cũng thực hiện các dịch vụ mà Service Interface cung cấp và Business Workflows sẽ sử dụng nó.

Ví dụ : Tiếp tục ví dụ ở trên. Bạn sẽ cần một Bussiness Component để kiểm tra gói hàng có khả dụng không ? hay một component để tính tổng chi phí,…

* Bussiness Entities : thường được sử dụng như Data Transfer Objects ( DTO). Bạn có thể sử dụng để truyền dữ liệu giữa các lớp (Presentation và Data Layer). Chúng thường là cấu trúc dữ liệu ( DataSets, XML,… ) hay các lớp đối tượng đã được tùy chỉnh.

Ví dụ : tạo 1 class Student lưu trữ các dữ liệu về tên, ngày sinh, ID, lớp.

* **Data Layer (DAL) :**



* Data Access Logic Components : chịu trách nhiệm chính lưu trữ và truy xuất dữ liệu từ các nguồn dữ liệu (Data Sources) như XML, file system,… Hơn nữa còn tạo thuận lợi cho việc dễ cấu hình và bảo trì.
* Service Agents : giúp bạn gọi và tương tác với các dịch vụ từ bên ngoài một cách dễ dàng và đơn giản.

***II.1.3. NGÔN NGỮ C#***

C# được phát triển bởi Anders Hejlsberg và team của ông trong khi phát triển .Net Framework.

Ngôn ngữ C# là một trong số các ngôn ngữ được .NET Framework hỗ trợ , C# khá đơn giản, chỉ khoảng 80 từ khóa và hơn mười mấy kiểu dữ liệu được xây dựng sẵn. Tuy nhiên, ngôn ngữ C có ý nghĩa cao khi nó thực thi những khái niệm hiện đại. C# bao gồm tất cả những hỗ trợ cấu trúc, thành phần component, lập trình hướng đối tượng. Những tính chất đó hiện diện trong ngôn ngữ lập trình hiện đại.

* các lý do làm C# là ngôn ngữ lập trình chuyên nghiệp được sử dụng rộng rãi:
* Nó là hướng đối tượng.
  + Nó dễ dàng để học.
  + Nó là một ngôn ngữ được cấu trúc.
  + Nó tạo các chương trình hiệu quả.
  + Nó có thể được biên dịch trên nhiều nền tảng máy tính khác nhau.
  + Nó là một phần của .Net Framework.

Cấu trúc C# khá gần với các ngôn ngữ high-level truyền thống, C và C++, và là một ngôn ngữ lập trình hướng đối tượng. Nó có sự giống nhau mạnh mẽ với Java, nó có nhiều đặc điểm lập trình mạnh mẽ mà làm cho nó trở nên ưa thích với các lập trình viên trên toàn thế giới.

* các đặc điểm quan trọng của C#:
* Điều kiện Boolean
* Tự động dọn rác bởi Garbage-Collector (GC)
* Thư viện chuẩn (Standard Library)
* Hội Versioning
* Thuộc tính (Property) và sự kiện (Event)
* Các đại biểu (Delegate) và Quản lý sự kiện
* Dễ dàng để sử dụng Generic
* Indexer
* Biên dịch có điều kiện (Conditional Compilation)
* Đa luồng dễ dàng (Multithreading)
* LINQ và Lambda Expression
* Tích hợp với Windows

**CHƯƠNG III**

**THIẾT KẾ HỆ THỐNG**

**III.1. THIẾT KẾ CƠ SỞ DỮ LIỆU**

Cơ sở dữ liệu trong chương trình gồm 8 bảng

* Bảng khách hàng
* Bảng nhà sản xuất
* Bảng nhân viên
* Bảng sản phẩm
* Bảng hóa đơn
* Bảng chi tiết hóa đơn
* Bảng phiếu nhập
* Bảng chi tiết phiếu nhập

**III.2. THIẾT KẾ HỆ THỐNG**

***III.2.1. CƠ SỞ DỮ LIỆU***

* Cơ sở dữ liệu gồm 8 bảng sau:

**Bảng 1: KHÁCH HÀNG-**Khách hàng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên Trường** | **Kiểu dữ liệu** | **Field size** | **Khóa** | **Mô tả** |
| MA\_KH | varchar | 20 | Chính(PK) | Mã khách hàng |
| TEN\_KH | nvarchar | 100 |  | Tên khách hàng |
| DIACHI\_KH | nvarchar | 100 |  | Địa chỉ khách hàng |
| DT\_KH | varchar | 15 |  | Điện thoại khách hàng |

**Bảng 2: NHÀ SẢN XUẤT–** Nhà sản xuất

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên Trường** | **Kiểu dữ liệu** | **Field size** | **Khóa** | **Mô tả** |
| MA\_NSX | varchar | 20 | Chính(PK) | Mã nhà sản xuất |
| TEN\_NSX | nvarchar | 50 |  | Tên nhà sản xuất |
| DT\_NSX | varchar | 20 |  | Điện thoại nhà sản xuất |
| DIACHI\_NSX | nvarchar | 100 |  | Địa chỉ nhà sản xuất |

**Bảng 3: NHÂN VIÊN–** Nhân viên

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên Trường** | **Kiểu dữ liệu** | **Field size** | **Khóa** | **Mô tả** |
| MA\_NV | varchar | 20 | Chính(PK) | Mã nhân viên |
| TEN\_NV | nvarchar | 50 |  | Tên nhân viên |
| NS\_NV | Datetime |  |  | Ngày sinh nhân viên |
| GT\_NV | nvarchar | 5 |  | Giới tính nhân viên |
| NGAYVAOLAM\_NV | Datetime |  |  | Ngày vào làm |
| DIACHI\_NV | nvarchar | 100 |  | Địa chỉ nhân viên |

**Bảng 4: SẢN PHẨM**  - sản phẩm

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên Trường** | **Kiểu dữ liệu** | **Field size** | **Khóa** | **Mô tả** |
| Ma\_SP | varchar | 20 | Chính(PK) | Mã sản phẩm |
| TEN\_SP | nvarchar | 50 |  | Tên sản phẩm |
| GIA\_SP | float |  |  | Giá sản phẩm |
| NGUYENLIEU\_SP | nvarchar | 50 |  | Nguyên liệu sản phẩm |
| MA\_NSX | varchar | 20 | Ngoại(FK) | Mã nhà sản suất |

**Bảng 4: HÓA ĐƠN –hóa đơn**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên Trường** | **Kiểu dữ liệu** | **Field size** | **Khóa** | **Mô tả** |
| MA\_HD | varchar | 20 | Chính(PK) | Mã hóa đơn |
| NGAYLAP\_HD | Date/Time |  |  | Ngày vào lập hóa đơn |
| MA\_KH | varchar | 20 | Ngoại(FK) | Mã khách hàng |
| MA\_NV | varchar | 20 | Ngoại(FK) | Mã nhân viên |
| TONGTIEN | float |  |  | Tổng tiền |

**Bảng 5:CHI TIẾT HÓA ĐƠN-**Chi tiết hóa đơn

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên Trường** | **Kiểu dữ liệu** | **Field size** | **Khóa** | **Mô tả** |
| MA\_SP | varchar | 20 | chính(PK),phụ(FK) | Mã sản phẩm |
| MA\_HD | varchar | 20 | chính(PK),phụ(FK) | Mã hóa đơn |
| SOLUONG | tinyint |  |  | Số lượng chi tiết hóa đơn |
| THANHTIEN | float |  |  | Thành tiền chi tiết hóa đơn |
| DVT | nvarchar | 10 |  | Đơn vị tính |

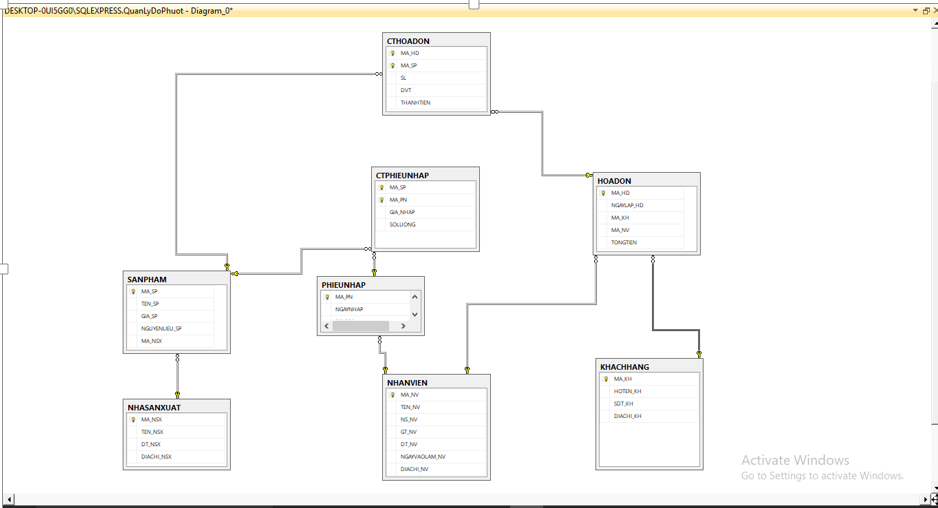
**Bảng 6: PHIẾU NHẬP – phiếu nhập**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên Trường** | **Kiểu dữ liệu** | **Field size** | **Khóa** | **Mô tả** |
| MA\_PN | varchar | 20 | chính(PK),Ngoại(FK) | Mã phiếu nhập |
| MA\_NV | varchar | 20 | chính(PK),Ngoại(FK) | Mã nhân viên |

**Bảng 7: CHI TIẾT PHIẾU NHẬP –chi tiết phiếu nhập**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên Trường** | **Kiểu dữ liệu** | **Field size** | **Khóa** | **Mô tả** |
| MA\_SP | varchar | 20 | Chính(PK),Ngoại(FK) | Mã sản phẩm |
| MA\_PN | varchar | 20 | Chính(PK),Ngoại(FK) | Mã phiếu nhập |
| GIA\_NHAP | float |  |  | Giá nhập |
| SOLUONG | tinyint |  |  | Số lượng |

***III.2.2. MÔ HÌNH QUAN HỆ GIỮA CÁC BẢNG***



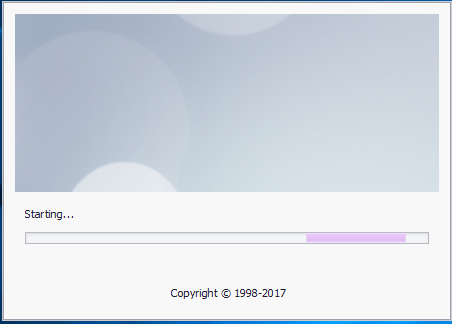
***III.2.3. CHỨC NĂNG TỪNG BẢNG***

* Mô tả các chức năng:

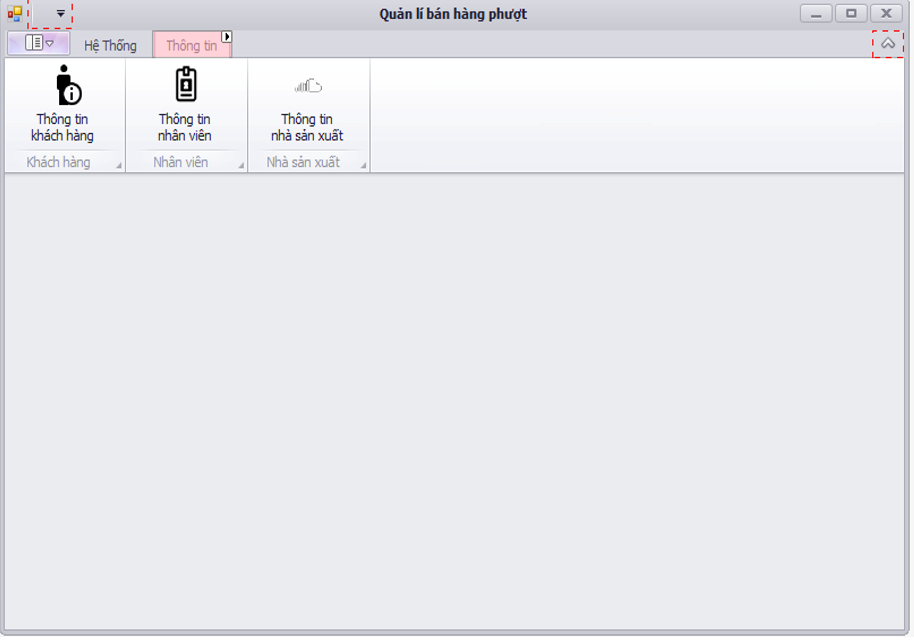
|  |  |  |
| --- | --- | --- |
| **Stt** | **Tên chức năng** | **ý nghĩa** |
| 1 | Cập nhật thông tin khách hàng | nhật thông tin như:   * Thêm * Sửa * Xóa * Lưu * Thoát * Tìm kiếm |
| 2 | Cập nhật thông tin nhà sản suất | Cập nhật thông tin nhà sản xuất như : |
| 3 | Cập nhật thông tin Nhân viên | Cập nhật thông tin Nhân viên như:   * Thêm * Sửa * Xóa * Lưu * Hủy |
| 4 | Cập nhật thông tin sản phẩm | * Sửa * Tìm * Làm mới * Xóa * Thêm * Thoát |
| 5 | Cập nhật hóa đơn | Nhận viên sẽ lập hoá đơn bán và thanh toán tiên với khách hàng |
| 6 | Cập nhật chi tiết hóa đơn | * Sửa |
| 7 | Lập phiếu nhập | Khi nhận được hàng từ nhà cung cấp thì nhân viên sẽ lập một phiếu nhập hàng. |
| 8 | Cập nhật chi tiết phiếu nhập | * Sửa |
| 9 | Tìm kiếm | Tìm kiếm theo yêu cầu thông tin người sử dụng |
| 10 | Tìm kiếm | Tìm kiếm thông tin về hoá đơn bán hàng,nhập hàng |
| 11 | Báo cáo nhập xuất | Báo cáo về tình hình bán hàng và nhập hàng của cửa hàng |

***III.2.4. CÁC FORM GIAO DIỆN CỦA CHƯƠNG TRÌNH***

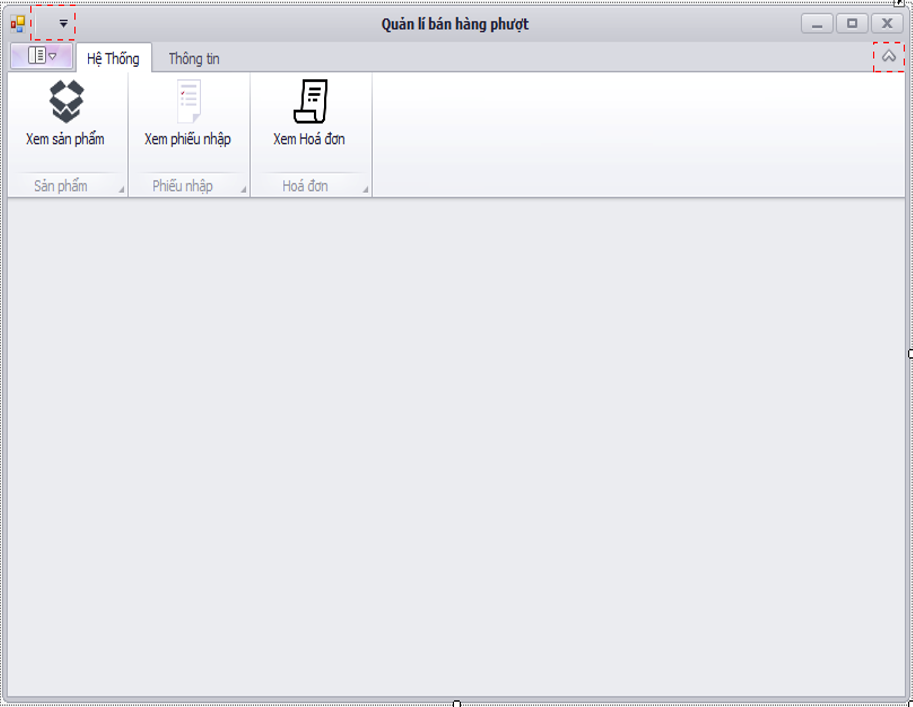
### III2.4.1. Form Splash Screen

****

### III2.4.2. Form giao diện chính

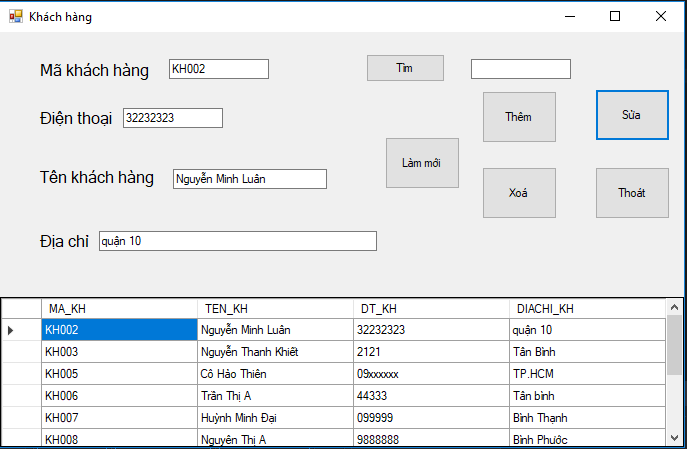


* Khi click vào Thông tin sẽ ra được :
  + Thông tin khách hang
  + Thông tin nhân viên
  + Thông tin nhà sản xuất



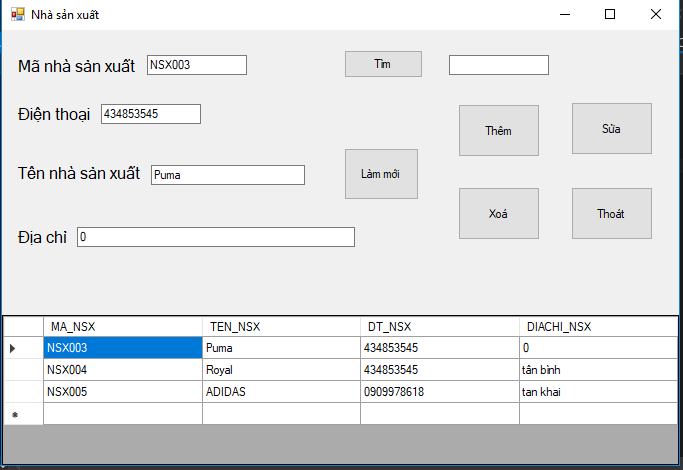
* Khi Click vào Hệ thống :
  + Xem sản phẩm
  + Xem phiếu nhập
  + Xem hóa đơn

### III.2.4.3.Form Khách hàng

****

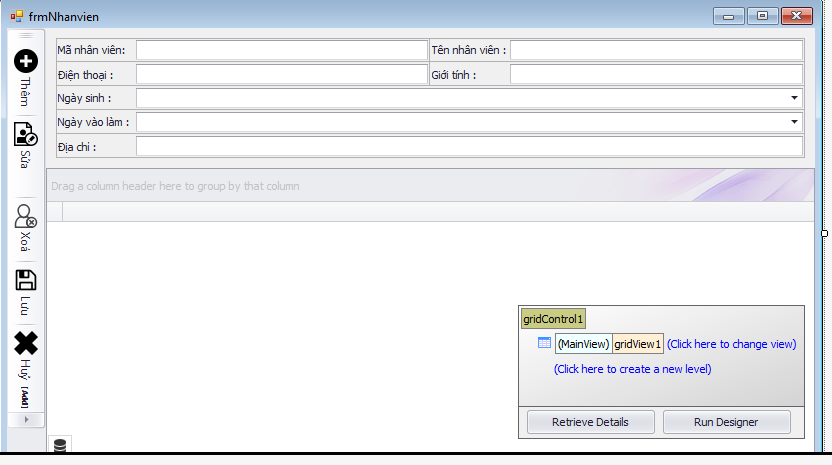
Đây là form giúp chủ cửa hàng có thể thêm ,xóa ,sửa ,tìm kiếm các khách hàng mới cũng như khách hàng cũ. Đây là bước tiền đề giúp chủ cửa hàng có thể mở rộng thêm mục khách hàng hay mua sắm tại cửa hàng thường xuyên để có được khuyến mãi cho người mua

### III.2.4.4. Form Nhà sản xuất



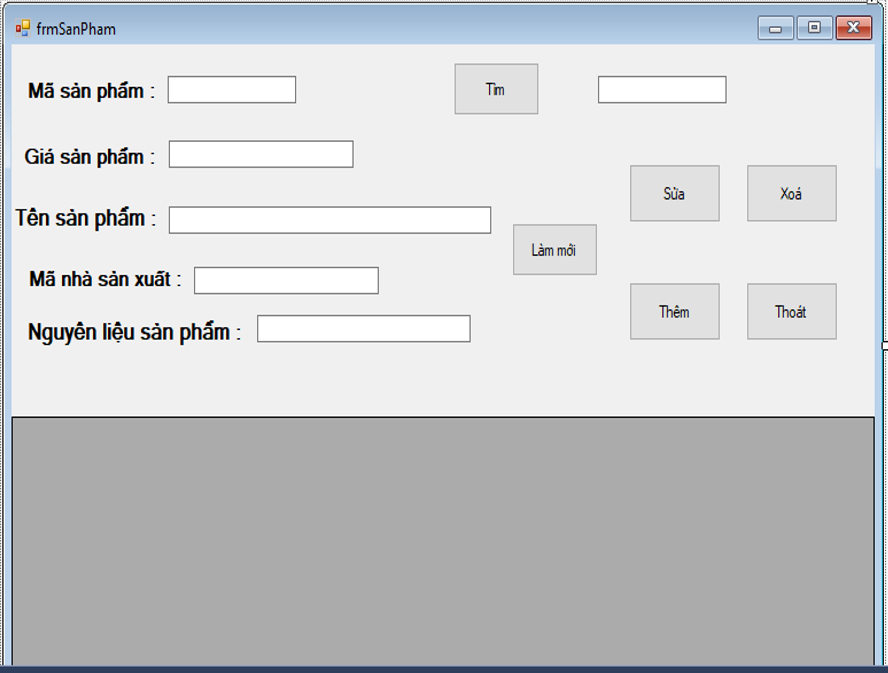
Đây là form để thêm xóa sửa các nhà cung cấp , đã và đang cung cấp cho cửa hàng

### III.2.4.5. Form Nhân viên

****

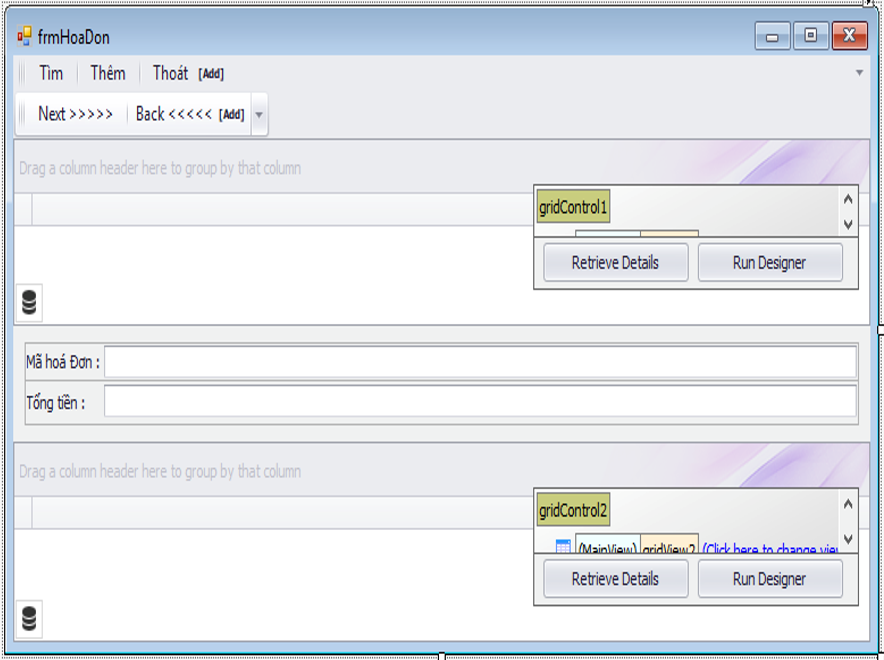
Quản lý nhân viên giúp chủ cửa hàng có thể nắm được những thông tin cá nhân của nhân viên và có thể sửa chữa cũng như ẩn nhân viên khỏi hệ thống nếu đã nghỉ làm.Cũng như sẽ có những phần thường dành cho những nhân viên gắn bó lâu dài

### III.2.4.6. Form Sản phẩm



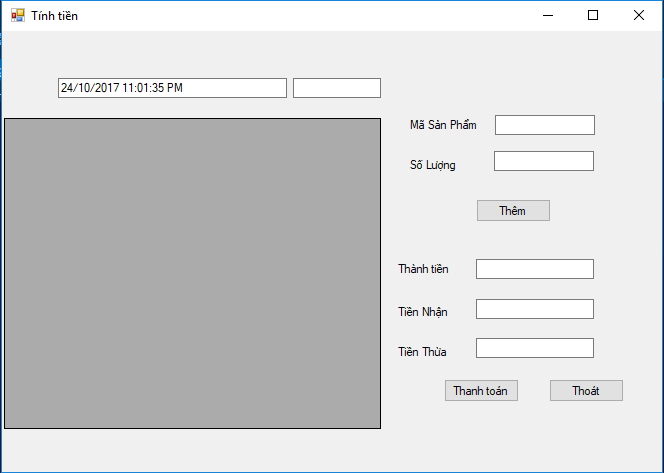
Form sản phẩm để lưu trữ lại các mặt hàng đã nhập về cửa hàng và tìm kiếm thông tin mặt hàng trong cửa hàng .

### III.2.4.7. Form Hóa Đơn và Chi Tiết Hóa Đơn



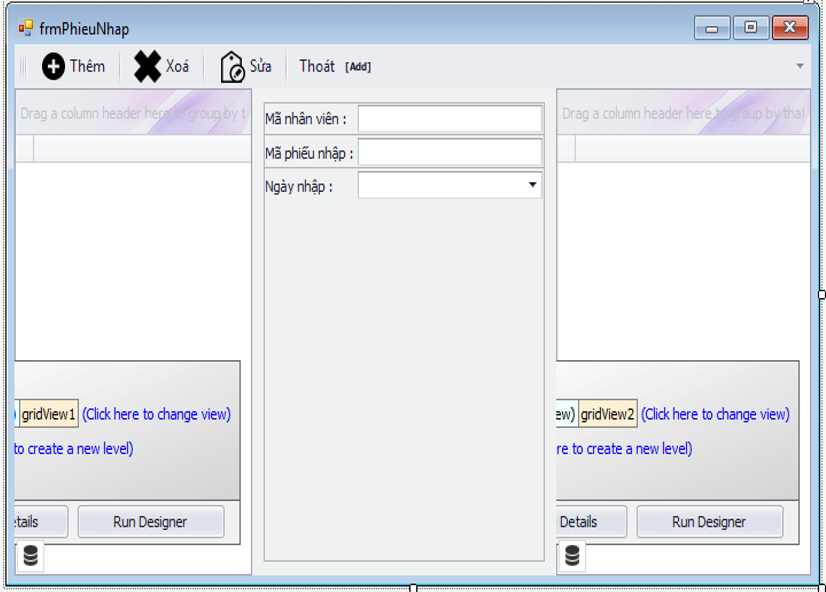
Đây là form lưu trữ lại các hóa đơn cũng như các chi tiết trong hóa đơn để thuận tiện trong việc kiểm tra đối chiếu hang hóa có trong cửa hàng nếu bị thâm hụt hoặc tránh trường hợp khách hàng khiếu nại về khai báo sai mã hàng dẫn đến sai thành tiền

### III. 2.4.8. Form tính tiền

****

Khi nhập mã sản phẩm ,số lượng và ấn thêm thì sẽ hiện ra món hàng khách hàng mua cũng như tổng tiền của sản phẩm (tổng tiền = số lượng \* đơn giá) sau đó thành thành tiền sẽ tính tổng các mặt hàng có trong đơn hàng ấy .Khi nhân viên nhập số tiền nhận và ấn thanh toán sẽ ra số tiền thừa của khách hàng.Mã hóa đơn bên phải ngày nhập hóa đơn sẽ tự động tăng lên nếu có đơn hàng khác

### III.2.4.9. Frm Phiếu Nhập và Chi tiết phiếu nhập



Đây là form thêm xóa sửa lưu trữ các mặt hàng nhập vào cửa hàng

### III.2.5.0. Các modules trong chương trình

* Dùng disconnection để kết nối với cơ sở dữ liệu

**CHƯƠNG IV**

**KẾT LUẬN**

**IV.1. ƯU ĐIỂM VÀ NHƯỢC ĐIỂM**

***IV.1.1. ƯU ĐIỂM***

* + Hình thức thông tin đa dạng và sống động
  + Giao diên thân thiện, dễ nhìn, màu sắc không đối lập
  + Thông tin được cập nhật và sửa đổi thường xuyên.

***IV.1.2. NHƯỢC ĐIỂM***

* Chưa xử lý được tối ưu các ràng buộc, còn một số tồn tại trong việc đặt tên và sử dụng linh hoạt các điều khiển.

**IV.3. TÀI LIỆU THAM KHẢO**

# Sách lập trình C# trên windown, biên soạn Mai Thu Ngọc, 2015

# Bài giảng lập trình C# trên windown, Ths. Nguyễn Hà Giang, 2010

* .Net Framework và lập trình visual studio :
  + <https://tungnt.net/tinh-nang-moi-cua-tung-phien-ban-net-framework/>
  + <https://123doc.org//document/2574133-tong-quan-ve-net-framework-va-moi-truong-lap-trinh-visual-studio.htm/>
* Mô hình 3-tiers :
  + <https://namdh.wordpress.com/2009/12/31/3-tiers-faq/>
* Mô hình 3 lớp :
  + <https://techtalk.vn/mo-hinh-3-lop-co-gi-hay.html>
* Bài: Ado-net
  + <https://www.mastercode.vn/blog/ado-net/bai-2-ket-noi-co-so-du-lieu-va-doc-du-lieu-cong-nghe-ado-net-voi-c.72>
  + http://tinhocthoidai.vn/gioi-thieu-ado-net-trong-c-net.html
* Bài : Tổng quan về ngôn ngữ lập trình c#:
  + <http://laptrinhc.edu.vn/tong-quan-ve-ngon-ngu-lap-trinh-c-sharp-2.html>
* Luận văn : đề tài quản lý nhà hàng máy tính
  + <http://luanvan.net.vn/luan-van/de-tai-quan-ly-cua-hang-ban-hang-may-tinh-46518/>
* Luận văn : đề tài quản lý bán hàng
  + <http://luanvan.net.vn/luan-van/xay-dung-phan-mem-quan-ly-ban-hang-60069>

**IV.3. ĐÁNH GIÁ CỦA GIÁO VIÊN**