ĐẠI HỌC QUỐC GIA TP.HCM

TRƯỜNG ĐẠI HỌC BÁCH KHOA

**KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH**

-----🙞🙜🕮🙞🙜-----



**BÁO CÁO THỰC TẬP TỐT NGHIỆP**

**ĐỀ TÀI:**

*Phát triển một framework hỗ trợ quản lý cho các công ty thương mại – dịch vụ – sản xuất vải.*

**GVHD:**

ThS. Nguyễn Thị Ái Thảo.

ThS. Nguyễn Đình Thành.

**GVPB:** TS. Trương Tuấn Anh.

**SVTH:** Võ Tiến Thiều 51203557

Trần Đắc Thế 51203532

TP.HCM, Tháng 05 năm 2016

**MỤC LỤC**

[I. GIỚI THIỆU ĐỀ TÀI 4](#_Toc452385872)

[1. Tổng quan về đề tài. 4](#_Toc452385873)

[2. Yêu cầu nghiệp vụ 4](#_Toc452385874)

[2.1. Quy trình quản lý chung 4](#_Toc452385875)

[2.2. Quy trình nhập hàng 5](#_Toc452385876)

[2.3. Quy trình bán hàng 6](#_Toc452385877)

[2.4. Quy trình quản lý thu chi 7](#_Toc452385878)

[2.5. Quy trình quản lý nhân viên 7](#_Toc452385879)

[II. KIẾN THỨC NỀN TẢNG 8](#_Toc452385880)

[1. MongoDB 8](#_Toc452385881)

[1.1. Giới thiệu về NoSQL 8](#_Toc452385882)

[1.2. MongoDB là gì ? 8](#_Toc452385883)

[1.3. Database trong MongoDB 9](#_Toc452385884)

[1.4. Collection trong MongoDB 9](#_Toc452385885)

[1.5. Document trong MongoBD 9](#_Toc452385886)

[1.6. MongoDB hoạt động như thế nào ? 9](#_Toc452385887)

[1.7. Ưu và nhược điểm của MongoDB 10](#_Toc452385888)

[2. C-Sharp ( C# ) 11](#_Toc452385889)

[2.1. C-Sharp là gi ? 11](#_Toc452385890)

[2.2. Đặc điểm ngôn ngữ 11](#_Toc452385891)

[2.3. Ưu và nhược điểm của C# 12](#_Toc452385892)

[3. ASP.NET MVC 4 12](#_Toc452385893)

[3.1. ASP.NET 12](#_Toc452385894)

[3.2. ASP.NET MVC 4 12](#_Toc452385895)

[3.3. Các tính năng của nền tảng ASP.NET MVC 14](#_Toc452385896)

[3.4. Lợi ích của ứng dụng web dựa trên mô hình MVC 15](#_Toc452385897)

[III. PHÂN TÍCH VÀ THIẾT KẾ 16](#_Toc452385898)

[1. Mô hình use-case 16](#_Toc452385899)

[2. Thiết kế cơ sở dữ liệu 24](#_Toc452385900)

[2.1. Lược đồ ERD 24](#_Toc452385901)

[2.2. Chi tiết kiểu dữ liệu 25](#_Toc452385902)

[3. Thiết kế famework 32](#_Toc452385903)

[3.1. Cấu trúc tổng quan 32](#_Toc452385904)

[3.2. Mô hình class diagram 33](#_Toc452385905)

[IV. HIỆN THỰC 56](#_Toc452385906)

[1. Công cụ phát triển 56](#_Toc452385907)

[2. Xây dựng framework 56](#_Toc452385908)

[V. DEMO ỨNG DỤNG XÂY DỰNG DỰA TRÊN FRAMEWORK 57](#_Toc452385909)

[1. Nghiệp vụ quản lý 57](#_Toc452385910)

[1.1. Nhân viên 57](#_Toc452385911)

[1.2. Nhà cung cấp 58](#_Toc452385912)

[1.3. Khách hàng 59](#_Toc452385913)

[2. Nghiệp vụ mua hàng 60](#_Toc452385914)

[2.1. Loại vải 60](#_Toc452385915)

[2.2. Loại vải màu 60](#_Toc452385916)

[2.3. Phiếu đặt hàng 61](#_Toc452385917)

[2.4. Hóa đơn mua hàng 62](#_Toc452385918)

[2.5. Thêm giá hàng hóa 62](#_Toc452385919)

[2.6. Lô hàng 63](#_Toc452385920)

[2.7. Cây vải 63](#_Toc452385921)

[2.8. Kho 64](#_Toc452385922)

[3. Nghiệp vụ bán hàng 65](#_Toc452385923)

[3.1. Đơn đặt hàng 65](#_Toc452385924)

[3.2. Thêm giá hàng hóa 65](#_Toc452385925)

[3.3. Hóa đơn bán hàng 66](#_Toc452385926)

[3.4. Thêm người vận chuyển cho hóa đơn 66](#_Toc452385927)

[4. Nghiệp vụ quản lý thu chi 67](#_Toc452385928)

[4.1. Thanh toán hóa đơn mua hàng 67](#_Toc452385929)

[4.2. Sổ thu chi 67](#_Toc452385930)

[VI. TỔNG KẾT VÀ ĐÁNH GIÁ 67](#_Toc452385931)

[TÀI LIỆU THAM KHẢO 68](#_Toc452385932)

# GIỚI THIỆU ĐỀ TÀI

## Tổng quan về đề tài.

Xây dựng một framework cho công ty dịch vụ buôn bán vải và sản xuất vải hỗ trợ các chức năng căn bản như:

* quản lý hàng hóa trong kho theo số mét, theo cây, theo màu, lô
* quản lý việc xuất, nhập hàng hóa
* quản lý đơn đặt hàng của khách hàng và đơn đặt hàng cho nhà cung cấp
* theo dõi công nợ của khách hàng
* quản lý giá cả hàng hóa theo thời giá
* quản lý thu chi theo thời gian
* quản lý thời gian làm việc, nghỉ phép của nhân viên
* xếp lịch giao nhận hàng hóa, theo dõi nguyên liệu và thành phẩm trong quá trình sản xuất.
* thống kê lịch sử giao dịch, các loại mặt hàng

## Yêu cầu nghiệp vụ

### 2.1. Quy trình quản lý chung

Một công ty chuyên phân phối vải cần quản lý quy trình nhập xuất hàng hóa, quản lý thông tin hàng hóa, công nợ, thu chi, nhân viên. Công ty có nhiều nhân viên được phân vào các loại sau: nhân viên bán hàng, giao hàng, nhập hàng, kế toán, thu ngân, quản lý. Mỗi nhân viên cần lưu trữ các thông tin sau: **mã số nhân viên, họ, tên, ngày sinh, địa chỉ, số điện thoại nội bộ, số di động, email, bằng cấp-chứng chỉ, ngày vào làm, lương**. Riêng nhân viên giao hàng cần lưu thêm thông tin số bằng lái.

Hàng hóa của công ty là những cây vải. Công ty kinh doanh nhiều loại vải khác nhau, phân biệt ở chất liệu vải. Thông tin cần lưu của loại vải gồm: **mã số, tên, nhà cung cấp**. Mỗi loại vải lại được nhuộm thành nhiều màu. Sản phẩm bán ra là những cây vải (vải sẽ được quấn thành một cây để bán). Cây vải có các thông tin: **mã số**, **loại vải, màu sắc, chiều dài, số lô**. Do trong quá trình nhuộm màu cho vải, người ta sẽ nhuộm theo lô. Cùng một loại vải nhuộm cùng một công thức màu, nhưng nhuộm ở 2 lô khác nhau, màu sau cũng sẽ không giống nhau 100%. Do đó những cây vải được nhuộm cùng một lô sẽ được đánh dấu số lô để biết được những cây nào cùng một lô nhuộm. Các cây vải sẽ được chứa trong các kho khác nhau. Thông tin kho gồm: **mã số, tên, địa chỉ, điện thoại, diện tích**. Trong quá trình nhập hàng từ nhà cung cấp, mỗi nhà cung cấp có thể nhập nhiều loại vải khác nhau, mỗi loại vải gồm nhiều cây vải với nhiều màu khác nhau, mỗi màu của mỗi loại vải sẽ được phân thành các lô nhuộm khác nhau. Quy trình nhập hàng được mô tả ở phần sau. Thông tin nhà cung cấp bao gồm: **mã số**, **tên, thông tin công ty (tên công ty, mã số thuế) – nếu có, thông tin ngân hàng (tên ngân hàng, số tài khoản) – có thể có nhiều tài khoản ngân hàng, số điện thoại bàn – nhiều số, số di động – nhiều số, email, địa chỉ**. Khách hàng mua hàng cũng có một quy trình bán hàng được mô tả ở phần sau. Mỗi khách hàng ứng với từng loại vải sẽ có giá bán khác nhau. Giá bán có thể thay đổi theo thời gian nên cần lưu lại thời điểm giá thay đổi để áp dụng giá đó cho tương lai cho đến khi giá mới được áp dụng*.* Thông tin khách hàng bao gồm**: mã số, họ tên, thông tin công ty (tên công ty, mã số thuế) – nếu có, thông tin ngân hàng (tên ngân hàng, số tài khoản) – có thể có nhiều tài khoản ngân hàng, số điện thoại bàn – nhiều số, số di động – nhiều số, email, địa chỉ**. Cần quản lý công nợ của nhà cung cấp và khách hàng.

### 2.2. Quy trình nhập hàng

Nhân viên nhập hàng có nhiệm vụ kiểm kê số lượng hàng ngày. Khi hụt hàng, nhân viên sẽ tạo ra phiếu đặt hàng. Phiếu đặt hàng bao gồm thông tin của kho nhận hàng, các loại vải, màu sắc, số lượng đặt (theo số mét hoặc theo cây vải), nhà cung cấp, thời điểm, tình trạng của phiếu đặt hàng. Lúc này trạng thái của phiếu đặt hàng đang là *chưa xác nhận*. Nếu nhà cung cấp không có hàng để bàn giao, tùy theo phản hồi của nhà cung cấp, nhân viên sẽ sửa lại cho phù hợp với số lượng hàng hóa mà nhà cung cấp có. Nếu không có tất cả mặt hàng, tình trạng phiếu sẽ là *Hủy*. Nếu lượng hàng không đủ, nhà cung cấp có thể giao trước một phần, phần còn lại sẽ được giao sau khi hàng đủ, lúc này trạng thái của phiếu đặt hàng là *đang giao*. Khi đã đủ hàng trạng thái là *hoàn thành*. Sau khi nhà cung cấp bàn giao hàng, kho sẽ xác nhận bằng phiếu nhập hàng. Phiếu nhập hàng bao gồm các thông tin về các sản phẩm (cây vải, loại vải, màu sắc, chiều dài, số lô, giá tiền), tổng tiền, nhà cung cấp, số lượng sản phẩm nhập, ngày tháng, nhân viên nhập hàng, phiếu nhập hàng này có ứng với phiếu đặt hàng nào không, nếu có cần lưu lại thông tin phiếu đặt hàng. Phiếu nhập hàng tương đương với hóa đơn bán hàng của nhà cung cấp. Thông thường các hóa đơn này sẽ được ghi vào công nợ của nhà cung cấp. Công ty sẽ không trả tiền liền. Thành tiền của hóa đơn mới sẽ được cộng dồn vào nợ cũ. Công ty thanh toán theo nhiều đợt. Cần lưu lại thông tin những đợt thanh toán này: số tiền, thời gian thanh toán, hình thức (ngân hàng hay tiền mặt), nếu chuyển qua ngân hàng cần lưu số tài khoản và ngân hàng giao dịch và người chuyển tiền, nếu chuyển tiền mặt cần lưu tên người nhận tiền. Giá mua từng mặt hàng sẽ được ghi trên hóa đơn mua hàng. Trong trường hợp công ty thanh toán tiền hóa đơn mua hàng liền cho nhà cung cấp, công ty có thể yêu cầu nhà cung cấp bớt tiền. Cần lưu lại số tiền được bớt trên hóa đơn mua hàng để biết số tiền thực tế mà công ty đã chi ra để trả cho nhà cung cấp. Trong trường hợp có gián đoạn trong quy trình, toàn bộ phiếu và hóa đơn liên quan sẽ có trạng thái là *Hủy*.

### 2.3. Quy trình bán hàng

Khách hàng đặt hàng qua điện thoại, email, hoặc ứng dụng. Thông tin đơn đặt hàng gồm: loại vải, màu sắc, số lượng, địa chỉ giao hàng, thời gian giao hàng, trạng thái. Ở thời điểm này trạng thái của đơn đặt hàng là *chưa xử lý*. Nếu đã đóng đầy đủ hàng theo yêu cầu của khách hàng trạng thái của đơn đặt hàng là *đã hoàn thàn*h. Trong trường hợp hàng có nhưng không đủ sẽ đóng trước một phần cho khách, phần còn lại sẽ giải quyết sau, trạng thái đơn đặt hàng lúc đó là *đang xử lý* . Khi xuất hàng cho khách hàng cần lưu lại hóa đơn bán hàng. Hóa đơn bán hàng chứa các thông tin: khách hàng, nhân viên bán hàng, ngày giờ bán, thông tin hàng hóa (cây vải, loại vải, màu sắc, số mét, đơn giá – tính theo mét, số lô), tổng tiền hóa đơn, tổng số cây vải. hóa đơn này được xuất theo đơn đặt hàng nào của khách hàng, lưu ý. Khách hàng có thể thanh toán hóa đơn liền, hoặc ghi nợ. Nếu ghi nợ hóa đơn mua hàng hiện tại sẽ được cộng dồn vào tổng nợ hiện tại của khách hàng để có tổng nợ mới. Khách hàng sẽ thanh toán từ từ. Cần lưu lại thông tin mỗi lần thanh toán của khách hàng: thời điểm thanh toán, số tiền, hình thức (tiền mặt, chuyển khoản), nếu trả tiền mặt thì lưu lại người nhận tiền, nếu chuyển khoản thì lưu lại người gởi tiền và tài khoản mà người đó gởi vào. Nếu khách hàng thanh toán tiền liền, công ty có thể bớt tiền cho khách hàng trên tổng tiền hóa đơn. Ví dụ: Tổng tiền hóa đơn là 77.400.000đ, khách hàng yêu cầu trả tiền liền, người quản lý có thể quyết định bớt theo thỏa thuân với khách hàng là 600.000đ, do đó số tiền thực tế thu được từ hóa đơn này là 76.800.000đ. Số tiền bớt này không theo một quy định nào hết chỉ bớt theo thỏa thuận với khách hàng nên cần lưu lại số tiền bớt cho khách hàng để biết được số tiền thu được thực tế từ hóa đơn. Sau khi xuất hàng, hàng hóa sẽ được vận chuyển tới địa chỉ mà khách hàng yêu cầu hoặc khách có thể tự tới công ty lấy hàng. Do đó cần lưu lại thông tin người nhận hàng. Trong trường hợp công ty chở thì cần lưu thêm thông tin nhân viên giao hàng. Trong trường hợp có sự gián đoạn trong quy trình bán hàng làm cho quy trình không thể hoàn thành. Toàn bộ các phiếu và hóa đơn liên quan sẽ được đưa về trạng thái *Hủy*.

### 2.4. Quy trình quản lý thu chi

Cuối ngày , công ty sẽ kiểm tra số tiền còn lại trong két. Số tiền này được tính theo số tiền công ty thu vào từ số tiền mà khách hàng thanh toán cho công ty trừ đi số tiền mà công ty chi ra để thanh toán cho nhà cung cấp. Các khoản chi còn bao gồm tiền lương của nhân viên (nếu hôm đó là ngày thanh toán lương cho nhân viên, còn bình thường không có khoản này), và một số mục chi lặt vặt. Cần lưu lại các khoản thu chi hàng ngày: **mã số,** **thời điểm, số tiền, diễn giải (ghi lại lý do thu/chi, nếu khoản thu/chi ứng với số tiền chi cho nhà cung cấp hoặc thu của khách hàng trong quy trình mua bán thì cần lưu lại mã số các khoản thanh toán đó).**

### 2.5. Quy trình quản lý nhân viên

Công ty có nhiều nhân viên được phân vào các loại sau: nhân viên bán hàng, giao hàng, nhập hàng, kế toán, thu ngân, quản lý. Mỗi **nhân viên** cần lưu trữ các thông tin sau: **mã số nhân viên, họ, tên, ngày sinh, địa chỉ, số điện thoại nội bộ, số di động, email, bằng cấp-chứng chỉ, ngày vào làm, lương**. Riêng nhân viên giao hàng cần lưu thêm thông tin số bằng lái.

Nhân viên bán hàng phụ trách việc lập phiếu đặt hàng cho khách hàng khi có yêu cầu, lập phiếu bán hàng cho khách hàng khi xuất hàng.

Nhân viên giao hàng phụ trách việc giao hàng khi có yêu cầu.

Nhân viên nhập hàng phụ trách việc lập phiếu đặt hàng và nhập hàng.

Nhân viên thu ngân lập hóa đơn thanh toán cho khách hàng và phiếu thanh toán nợ nhà cung cấp.

Ứng với mỗi nhân viên có một chức vụ và mức lương riêng. Lương sẽ được kế toán kết toán lại và trả cho nhân viên vào mỗi cuối tháng.

# KIẾN THỨC NỀN TẢNG

## MongoDB

### Giới thiệu về NoSQL

Cơ sở dữ liệu NoSQL (tên gốc là "Non SQL" hoặc "non relational") cung cấp một cơ chế để [lưu trữ](https://vi.wikipedia.org/wiki/B%E1%BB%99_nh%E1%BB%9B) và [truy xuất](https://vi.wikipedia.org/w/index.php?title=Data_retrieval&action=edit&redlink=1) dữ liệu được mô hình hóa khác với các quan hệ bảng được sử dụng trong các [cơ sở dữ liệu kiểu quan hệ](https://vi.wikipedia.org/w/index.php?title=Relational_database&action=edit&redlink=1). Các hệ thống NoSQL cũng đôi khi được gọi là "Not only SQL" (không chỉ là SQL) để nhấn mạnh rằng chúng có thể hỗ trợ các ngôn ngữ truy vấn dạng như [SQL](https://vi.wikipedia.org/wiki/SQL).

NoSQL lưu trữ dữ liệu của mình theo dạng cặp giá trị “key – value”. Sử dụng số lượng lớn các node để lưu trữ thông tin – Mô hình phân tán dưới sự kiểm soát phần mềm

Chấp nhận dữ liệu bị trùng lặp do một số node sẽ lưu cùng thông tin giống nhau

Một truy vấn sẽ được gửi tới nhiều máy cùng lúc, do đó khi một máy nào đó không phục vụ được sẽ không ảnh hưởng lắm đến chất lượng trả về kết quả

Tính nhất quán không theo thời gian thực: Sau mỗi thay đổi CSDL, không cần tác động ngay đến tất cả các CSDL liên quan mà được lan truyền theo thời gian.

Phi quan hệ – không có ràng buộc nào cho việc nhất quán dữ liệu.

### 1.2. MongoDB là gì ?

**MongoDB**là một mã nguồn mở và là một tập tài liệu dùng cơ chế NoSQL để truy vấn, nó được viết bởi ngôn ngữ C++. Chính vì được viết bởi C++ nên nó có khả năng tính toán với tốc độ cao chứ không giống như các hệ quản trị CSDL hiện nay.

Trong MongoDB dữ liệu được lưu dưới dạng JSON mà trong MongoDB gọi là BSON, chính vì thế nó có hiệu suất cao, tương tác nhanh và khả năng mở rộng tốt, nó hoạt động trên khái niệm **collection**và **document**. Mỗi một table trong SQL sử dụng thì trong MongoDB gọi là collection. Mỗi một record trong SQL sử dụng thì trong MongoDB được gọi là document.

### 1.3. Database trong MongoDB

Database là một ổ chứa dữ liệu ở mức vật lý, mỗi database sẽ có nhiều collection và được thiết lập lưu trữ ở một nơi trong máy chủ máy tính. Một máy chủ MongoDB thường có thể tạo nhiều cơ sở dữ liệu.

### 1.4. Collection trong MongoDB

**Collection trong MongoDB** là nhóm các tài liệu, nó tương đương với một bảng trong CSDL thông thường nên mỗi collection sẽ thuộc về một database duy nhất. Tuy nhiên nó có một sực khác biệt đó là nó không có ràng buộc Relationship như các hệ quản trị CSDL khác nên việc truy xuất rất nhanh, chính vì thế mỗi collection có thể chứa nhiều thể loại khác nhau không giống như table trong hệ quản trị mysql là các field cố định.

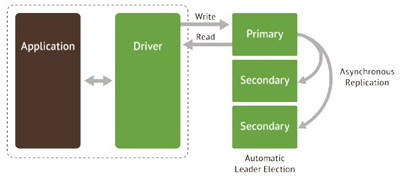
### 1.5. Document trong MongoBD

Document trong MongoDB có cấu trúc tương tự như kiểu dữ liệu JSON, nghĩa là sẽ có các cặp (key => giá trị) nên nó có tính năng động rất lớn. Document ta có thể hiểu nó giống như các record dữ liệu trong MYSQL, tuy nhiên nó có sự khác biệt là các cặp (key => value) có thể không giống nhau ở mỗi document.



*Hình 1: Ví dụ về một document*

### 1.6. MongoDB hoạt động như thế nào ?



*Hình 2: Mô hình hoạt động của MongoDB*

MongoDB hoạt động dưới một tiến trình ngầm service luôn mở một cổng (Cổng mặc định là 27017) để lắng nghe các yêu cầu truy vấn, thao tác từ các ứng dụng gửi vào sau đó mới tiến hành xử lý.

Mỗi một bản ghi của MongoDB được tự động gắn thêm một field có tên “\_id” thuộc kiểu dữ liệu ObjectId mà nó quy định để xác định được tính duy nhất của bản ghi này so với bản ghi khác, cũng như phục vụ các thao tác tìm kiếm và truy vấn thông tin về sau.

Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.

Mỗi khi có một truy vấn dữ liệu, bản ghi được cache (ghi đệm) lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.

Khi có yêu cầu thêm/sửa/xóa bản ghi, để đảm bảo hiệu suất của ứng dụng mặc định MongoDB sẽ chưa cập nhật xuống ổ cứng ngay, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng.

### 1.7. Ưu và nhược điểm của MongoDB

**Ưu điểm :**

* Dữ liệu lưu trữ phi cấu trúc, không có tính ràng buộc, toàn vẹn nên tính sẵn sàng cao, hiệu suất lớn và dễ dàng mở rộng lưu trữ.
* Dữ liệu được caching (ghi đệm) lên RAM, hạn chế truy cập vào ổ cứng nên tốc độ đọc và ghi cao.

**Nhược điểm :**

* Không ràng buộc, toàn vẹn nên không ứng dụng được cho các mô hình giao dịch yêu cầu độ chính xác cao.
* Không có cơ chế transaction (giao dịch) để phục vụ các ứng dụng ngân hàng.
* Dữ liệu được caching, lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn.
* Như đã giới thiệu ở trên, mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

## C-Sharp ( C# )

### 2.1. C-Sharp là gi ?

C# là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) [hướng đối tượng](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng) được phát triển bởi [Microsoft](https://vi.wikipedia.org/wiki/Microsoft), là phần khởi đầu cho kế hoạch [.NET](https://vi.wikipedia.org/w/index.php?title=.NET&action=edit&redlink=1) của họ. Tên của ngôn ngữ bao gồm ký tự thăng theo Microsoft nhưng theo [ECMA](https://vi.wikipedia.org/w/index.php?title=ECMA&action=edit&redlink=1) là **C#**, chỉ bao gồm dấu số thường. Microsoft phát triển C# dựa trên [C++](https://vi.wikipedia.org/wiki/C%2B%2B) và [Java](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)). C# được miêu tả là ngôn ngữ có được sự cân bằng giữa C++,[Visual Basic](https://vi.wikipedia.org/wiki/Visual_Basic), [Delphi](https://vi.wikipedia.org/wiki/Delphi_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) và Java.

### 2.2. Đặc điểm ngôn ngữ

C#, theo một hướng nào đó, là ngôn ngữ lập trình phản ánh trực tiếp nhất đến.NET Framework mà tất cả các chương trình.NET chạy, và nó phụ thuộc mạnh mẽ vào Framework này. Mọi dữ liệu cơ sở đều là đối tượng, được cấp phát và hủy bỏ bởi trình dọn rác Garbage-Collector (GC), và nhiều kiểu trừu tượng khác chẳng hạn như class, delegate, interface, exception, v.v, phản ánh rõ ràng những đặc trưng của.NET runtime.

Cấu trúc C# khá gần với các ngôn ngữ high-level truyền thống, C và C++ 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.

Dưới đây là các đặc điểm quan trọng của C#:

* + - Điều kiện Boolean
    - Tự động dọn rác bởi Garbage-Collector (GC)
    - Thư viện chuẩn (Standard Library)
      * Assembly
      * Property và sự kiện (Event)
      * Delegate 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

### 2.3. Ưu và nhược điểm của C#

**Ưu điểm :**

* + - * Nó là một ngôn ngữ lập trình hiện đại, mục đích tổng quát.
      * 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.

**Nhược điểm :**

* C# chỉ có thể chạy được trên nền Windows và .NETFrameword. Vì vậy nó hạn chế phạm vi các ứng dụng của C#. Đó là nhược điểm lớn nhất của C# với các ngôn ngữ lập trình khác.

## ASP.NET MVC 4

### 3.1. ASP.NET

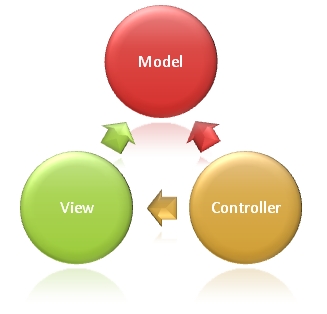
ASP.NET là một nền tảng ứng dụng web (*web application framework*) được phát triển và cung cấp bởi [Microsoft](https://vi.wikipedia.org/wiki/Microsoft), cho phép những người lập trình tạo ra những trang web động, những [ứng dụng web](https://vi.wikipedia.org/wiki/%E1%BB%A8ng_d%E1%BB%A5ng_web) và những [dịch vụ web](https://vi.wikipedia.org/wiki/D%E1%BB%8Bch_v%E1%BB%A5_web). Lần đầu tiên được đưa ra thị trường vào tháng 2 năm 2002 cùng với phiên bản 1.0 của.NET framework, là công nghệ nối tiếp của Microsoft's Active Server Pages(ASP). ASP.NET được biên dịch dưới dạng [Common Language Runtime](https://vi.wikipedia.org/w/index.php?title=Common_Language_Runtime&action=edit&redlink=1) (CLR), cho phép những người lập trình viết mã ASP.NET với bất kỳ ngôn ngữ nào được hỗ trợ bởi.NET language.

### 3.2. ASP.NET MVC 4

ASP.NET 4 MVC 4.0 là một framework để phát triển ứng dụng Web theo mô hình MVC được định nghĩa bởi Model-View-Controller. Và framework này khuyến khích lập trình viên lập trình những ứng dụng web với qui mô lớn, được chia ra thành từng phần là UI - phần giao tiếp với người dùng riêng và phần điều khiển và tương tác với CSDL là phần Model, và phần tương tác giữa CSDL và người dùng thông qua Controller.

Mẫu kiến trúc Model – View – Controller được sử dụng nhằm chi ứng dụng thành ba thành phần chính: model, view và controller. Nền tảng ASP.NET MVC giúp cho chúng ta có thể tạo được các ứng dụng web áp dụng mô hình MVC thay vì tạo ứng dụng theo mẫu ASP.NET Web Forsm. Nền tảng ASP.NET MVC có đặc điểm nổi bật là nhẹ (lighweigt), dễ kiểm thử phần giao diện (so với ứng dụng Web Forms), tích hợp các tính năng có sẵn của ASP.NET. Nền tảng ASP.NET MVC được định nghĩa trong namespace System.Web.Mvc và là một phần của name space System.Web.  
  
MVC là một mẫu thiết kế (design pattern) chuẩn mà nhiều lập trình viên đã quen thuộc. Một số loại ứng dụng web sẽ thích hợp với kiến trúc MVC. Một số khác vẫn thích hợp với ASP.NET Web Forms và cơ chế postbacks. Đôi khi có những ứng dụng kết hợp cả hai kiến trúc trên.

Nền tảng MVC bao gồm các thành phần dưới đây :



*Hình 3: Mẫu Model – View – Controller*

**Models**:

Các đối tượng Models là một phần của ứng dụng, các đối tượng này thiết lập logic của phần dữ liệu của ứng dụng. Thông thường, các đối tượng model lấy và lưu trạng thái của model trong CSDL. Ví dụ như, một đối tượng Product (sản phẩm) sẽ lấy dữ liệu từ CSDL, thao tác trên dữ liệu và sẽ cập nhật dữ liệu trở lại vào bảng Products ở SQL Server.  
  
Trong các ứng dụng nhỏ, model thường là chỉ là một khái niệm nhằm phân biệt hơn là được cài đặt thực thụ, ví dụ, nếu ứng dụng chỉ đọc dữ liệu từ CSDL và gởi chúng đến view, ứng dụng khong cần phải có tầng model và các lớp lien quan. Trong trường hợp này, dữ liệu được lấy như là một đối tượng model (hơn là tầng model).  
  
**Views**:

Views là các thành phần dùng để hiển thị giao diện người dùng (UI). Thông thường, view được tạo dựa vào thông tin dữ liệu model. Ví dụ như, view dùng để cập nhật bảng Products sẽ hiển thị các hộp văn bản, drop-down list, và các check box dựa trên trạng thái hiện tại của một đối tượng Product.  
  
**Controllers**:

Controller là các thành phần dùng để quản lý tương tác người dùng, làm việc với model và chọn view để hiển thị giao diện người dùng. Trong một ứng dụng MVC, view chỉ được dùng để hiển thị thông tin, controller chịu trách nhiệm quản lý và đáp trả nội dung người dùng nhập và tương tác với người dùng. Ví dụ, controller sẽ quản lý các dữ liệu người dùng gởi lên (query-string values) và gởi các giá trị đó đến model, model sẽ lấy dữ liệu từ CSDL nhờ vào các giá trị này.

### 3.3. Các tính năng của nền tảng ASP.NET MVC

Tách bạch các tác vụ của ứng dụng (logic nhập liệu, business logic, và logic giao diện), dễ dàng kiểm thử và mặc định áp dụng hướng phát triển TDD. Tất cả các tính năng chính của mô hình MVC được cài đặt dựa trên interface và được kiểm thử bằng cách sử dụng các đối tượng mocks, mock object là các đối tượng mô phỏng các tính năng của những đối tượng thực sự trong ứng dụng. Bạn có thể kiểm thử unit-test cho ứng dụng mà không cần chạy controller trong tiến trình ASP.NET, và điều đó giúp unit test được áp dụng nhanh chóng và tiện dụng. Bạn có thể sử dụng bất kỳ nền tảng unit-testing nào tương thích với nền tảng .NET.

MVC là một nền tảng khả mở rộng (extensible) & khả nhúng (pluggable). Các thành phần của ASP.NET MVC được thiết kể để chúng có thể được thay thế một cách dễ dàng hoặc dễ dàng tùy chỉnh. Bạn có thể nhúng thêm view engine, cơ chế định tuyến cho URL, cách kết xuất tham số của action-method và các thành phần khác. ASP.NET MVC cũng hỗ trợ việc sử dụng Dependency Injection (DI) và Inversion of Control (IoC). DI cho phép bạn gắn các đối tượng vào một lớp cho lớp đó sử dụng thay vì buộc lớp đó phải tự mình khởi tạo các đối tượng. IoC quy định rằng, nếu một đối tượng yêu cầu một đối tượng khác, đối tượng đầu sẽ lấy đối tượng thứ hai từ một nguồn bên ngoài, ví dụ như từ tập tin cấu hình. Và nhờ vậy, việc sử dụng DI và IoC sẽ giúp kiểm thử dễ dàng hơn.

ASP.NET MVC có thành phần ánh xạ URL mạnh mẽ cho phép bạn xây dựng những ứng dụng có các địa chỉ URL xúc tích và dễ tìm kiếm. Các địa chỉ URL không cần phải có phần mở rộng của tên tập tin và được thiết kế để hỗ trợ các mẫu định dạng tên phù hợp với việc tối ưu hóa tìm kiếm (URL) và phù hợp với lập địa chỉ theo kiểu REST.

Hỗ trợ sử dụng đặc tả (các thẻ) của các trang ASP.NET(.aspx), điều khiển người dùng (.ascx) và trang master page (.marter). Bạn có thể sử dụng các tính năng có sẵn của ASP.NET như là sử dụng lồng các trang master page, sử dụng in-line expression (<%= %>), sử dụng server controls, mẫu, data-binding, địa phương hóa (localization) và hơn thế nữa.

Hỗ trợ các tính năng có sẵn của ASP.NET như cơ chế xác thực người dùng, quản lý thành viên, quyền, output caching và data caching, seession và profile, quản lý tình trạng ứng dụng, hệ thống cấu hình…

ASP.NET MVC 3 còn bổ sung một view engine mới là Razor View Engine cho phép thiết lập các view nhanh chóng, dễ dàng và tốn ít công sức hơn so với việc sử dụng Web Forms view engine.

### 3.4. Lợi ích của ứng dụng web dựa trên mô hình MVC

Dễ dàng quản lý sự phức tạp của ứng dụng bằng cách chia ứng dụng thành ba thành phần model, view, controller

Nó không sử dụng view state hoặc server-based form. Điều này tốt cho những lập trình viên muốn quản lý hết các khía cạnh của một ứng dụng.

Nó sử dụng mẫu Front Controller, mẫu này giúp quản lý các requests (yêu cầu) chỉ thông qua một Controller. Nhờ đó bạn có thể thiết kế một hạ tầng quản lý định tuyến.

Hỗ trợ tốt hơn cho mô hình phát triển ứng dụng hướng kiểm thử (TDD).

Hỗ trợ tốt cho các ứng dụng được xây dựng bởi những đội có nhiều lập trình viên và thiết kế mà vẫn quản lý được tính năng của ứng dụng.

# PHÂN TÍCH VÀ THIẾT KẾ

## Mô hình use-case



*Hình 4: Mô hình use case nghiệp vụ nhập hàng.*

Mô tả mô hình use case nhập hàng:

* Use case “Xem danh sách tình trạng trong kho”: Cho nhân viên nhập hàng xem tình trạng hàng hoá trong các kho để quản lý lập đơn đặt hàng cho nhà cung cấp.
* Use case “Tìm kho”: Là use case mở rộng của use case “Xem danh sách tình trạng trong kho”, use case này cho phép nhân viên nhập hàng có thể tìm kho cụ thể nào đó trong danh sách các kho để xem chi tiết kho đó.
* Use case “Xem chi tiết thông tin hàng hoá trong kho”: Xem chi tiết một kho nào đó, use case này có các use case mở rộng “Theo lô”, “Theo loại vải”, “Theo loại vải màu” tương ứng từng use case thì người dùng có thể xem chi tiết hàng hoá trong kho được sắp xếp theo lô, loại vải, loại vải màu.
* Use case “Theo cây hoặc mét”: Là use case mở rộng của hai use case “Theo loại vải màu” và “Theo loại vải”: Cho xem chi tiết hàng hoá trong kho theo loại vải hoặc loại vải màu nhưng được sắp xếp thông tin theo đơn vị đo là cây hoặc mét của loại vải và loại vải màu.
* Use case “Danh sách phiếu nhập hàng”: Xem danh sách các phiếu nhập hàng đã được nhập vào kho.
* Use case “Theo tình trạng thanh toán”: use case mở rộng của “Danh sách phiếu nhập hàng” cho phép xem danh sách phiếu nhập hàng được sắp xếp theo tình trạng thanh toán cho phiếu nhập hàng.
* Use case “Tìm kiếm phiếu nhập hàng”: use case mở rộng của “Danh sách phiếu nhập hàng” cho phép tìm kiếm một phiếu nhập hàng củ thể trong danh sách phiếu nhập hàng.
* Use case “Xem chi tiết phiếu nhập hàng”: là use case mở rộng của “Danh sách phiếu nhập hàng” cho phép xem chi tiết cụ thể các thông tin trong phiếu nhập hàng.
* Use case “Xoá phiếu nhập hàng”: là use case mở rộng của “Xem chi tiết phiếu nhập hàng” cho phép xoá một phiếu nhập hàng.
* Use case “Danh sách phiếu đặt hàng”: Xem danh sách các phiếu đặt hàng.
* Use case “Theo tình trạng”: use case mở rộng của “Danh sách phiếu đặt hàng” cho phép xem danh sách phiếu đặt hàng được sắp xếp theo tình trạng cho phiếu đặt hàng.
* Use case “Tìm kiếm phiếu đặt hàng”: use case mở rộng của “Danh sách phiếu đặt hàng” cho phép tìm kiếm một phiếu đặt hàng củ thể trong danh sách phiếu đặt hàng.
* Use case “Xem chi tiết phiếu đặt hàng”: là use case mở rộng của “Danh sách phiếu đặt hàng” cho phép xem chi tiết cụ thể các thông tin trong phiếu đặt hàng.
* Use case “Xoá phiếu đặt hàng”: là use case mở rộng của “Xem chi tiết phiếu đặt hàng” cho phép xoá một phiếu đặt hàng.
* Use case “Theo thời gian”, “Theo tình trạng” và theo “số lượng”: là ba use case mở rộng của use case “Danh sách phiếu đặt hàng”, ba use case này cho người dùng xem danh sách phiếu đặt hàng được sắp xếp theo số lượng hàng đặt của phiếu hoặc theo thời gian đặt, tình trạng phiếu.
* Use case “Cập nhật trạng thái phiếu đặt hàng”: Người dùng cập nhật, thay đổi trạng thái của phiếu đặt hàng.
* Use case “Xem danh sách nhà cung cấp”: Xem danh sách các nhà cung cấp hàng hoá cho công ty.
* Use case “Xem chi tiết nhà cung cấp”: use case mở rộng của “Xem danh sách nhà cung cấp” cho người dùng xem chi tiết thông tin về một nhà cung cấp.
* Use case “Tạo phiếu đặt hàng”: Tạo một phiếu đặt hàng để nhập hàng về.
* Use case “Tạo phiếu nhập hàng”: Tạo một phiếu nhập hàng khi hàng được nhập về kho.



*Hình 5: Mô hình use case nghiệp vụ bán hàng.*

Mô tả mô hình use case nhập hàng:

* Use case “Xem danh sách tình trạng trong kho”: Cho nhân viên bán hàng xem tình trạng hàng hoá trong các kho để quản lý lập hoá đơn bán hàng cho Khách hàng.
* Use case “Tìm kho”: Là use case mở rộng của use case “Xem danh sách tình trạng trong kho”, use case này cho phép nhân viên bán hàng có thể tìm kho cụ thể nào đó trong danh sách các kho để xem chi tiết kho đó.
* Use case “Xem chi tiết thông tin hàng hoá trong kho”: Xem chi tiết một kho nào đó, use case này có các use case mở rộng “Theo lô”, “Theo loại vải”, “Theo loại vải màu” tương ứng từng use case thì người dùng có thể xem chi tiết hàng hoá trong kho được sắp xếp theo lô, loại vải, loại vải màu.
* Use case “Theo cây hoặc mét”: Là use case mở rộng của hai use case “Theo loại vải màu” và “Theo loại vải”: Cho xem chi tiết hàng hoá trong kho theo loại vải hoặc loại vải màu nhưng được sắp xếp thông tin theo đơn vị đo là cây hoặc mét của loại vải và loại vải màu.
* Use case “Danh sách hoá đơn bán hàng”: Xem danh sách hoá đơn bán hàng đã được lập.
* Use case “Theo tình trạng thanh toán”: use case mở rộng của “Danh sách hoá đơn bán hàng” cho phép xem danh sách hoá đơn bán hàng được sắp xếp theo tình trạng thanh toán cho hoá đơn bán hàng.
* Use case “Tìm kiếm hoá đơn bán hàng”: use case mở rộng của “Danh sách hoá đơn bán hàng” cho phép tìm kiếm một hoá đơn bán hàng cụ thể trong danh sách hoá đơn bán hàng.
* Use case “Xem chi tiết hoá đơn bán hàng”: là use case mở rộng của “Danh sách hoá đơn bán hàng” cho phép xem chi tiết cụ thể các thông tin trong hoá đơn bán hàng.
* Use case “Xoá hoá đơn bán hàng”: là use case mở rộng của “Xem chi tiết hoá đơn bán hàng” cho phép xoá một hoá đơn bán hàng.
* Use case “Danh sách phiếu đặt hàng của Kh”: Xem danh sách các phiếu đặt hàng mà khách hàng đã lập.
* Use case “Theo tình trạng”: use case mở rộng của “Danh sách phiếu đặt hàng của Kh” cho phép xem danh sách phiếu đặt hàng được sắp xếp theo tình trạng cho phiếu đặt hàng.
* Use case “Tìm kiếm phiếu đặt hàng”: use case mở rộng của “Danh sách phiếu đặt hàng của Kh” cho phép tìm kiếm một phiếu đặt hàng củ thể trong danh sách phiếu đặt hàng.
* Use case “Xem chi tiết phiếu đặt hàng”: là use case mở rộng của “Danh sách phiếu đặt hàng của Kh” cho phép xem chi tiết cụ thể các thông tin trong phiếu đặt hàng.
* Use case “Xoá phiếu đặt hàng”: là use case mở rộng của “Xem chi tiết phiếu đặt hàng của Kh” cho phép xoá một phiếu đặt hàng.
* Use case “Cập nhật trạng thái phiếu đặt hàng”: Người dùng cập nhật, thay đổi trạng thái của phiếu đặt hàng.
* Use case “Xem thông tin Kh”: use case mở rộng của “Xem chi tiết phiếu đặt hàng của Kh” cho xem chi tiết khách hàng thuộc phiếu đặt hàng.
* Use case “Xem chi tiết nhà cung cấp”: use case mở rộng của “Xem danh sách nhà cung cấp” cho người dùng xem chi tiết thông tin về một nhà cung cấp.
* Use case “Tạo phiếu đặt hàng cho Kh”: Tạo một phiếu đặt hàng.
* Use case “Tạo hoá đơn hàng”: Tạo một hoá đơn hàng khi hàng được xuất kho.
* Use case “Thêm khách hàng”: Thêm một khách hàng.
* Use case “Nhập giá SP cho kh”: Nhập giá cả các mặt hàng bán cho khách hàng, từng khách hàng có một giá riêng.



*Hình 6: Mô hình use case nghiệp vụ Thanh toán.*

Mô tả mô hình use case nghiệp vụ thanh toán:

* Use case “Xem danh sách khách hàng”: Cho xem danh sách khách hàng đã mua hàng của công ty.
* Use case “Xem danh sách nhà cung cấp”: Cho xem danh sách nhà cung cấp mà công ty đã mua hàng.
* Use case “Xem số nợ hiện tại”: use case mở rộng của “Xem danh sách khách hàng” và “Xem danh sách nhà cung cấp” cho xem danh sách cùng khoản nợ với công ty.
* Use case “Xem chi tiết”: use case mở rộng của “Xem danh sách khách hàng ” và “Xem danh sách nhà cung cấp” cho xem chi tiết một khách hàng hoặc nhà cung cấp nào đó.
* Use case “Thanh toán”: Thanh toán nợ cho nhà cung cấp.
* Use case “kiểm tra thanh toán”: kiểm tra thanh toán mà khách hàng đã thanh toán cho công ty.
* Use case “Lập phiếu thanh toán”: lập phiếu thanh toán cho khách hàng hoặc nhà cung cấp.
* Use case “Ghi vào sổ thu chi”: mọi phiếu thanh toán được lập đều được tự động ghi vào sổ thu chi.



*Hình 7: Mô hình use case nghiệp vụ quản lý.*

Mô tả mô hình use case nghiệp vụ quản lý:

* Use case “Quản lý nhân viên”: quản lý tất cả thông tin liên quan đến nhân viên.
* Use case “Quản lý nhà cung cấp”: quản lý tất cả thông tin liên quan đến nhà cung cấp.
* Use case “Quản lý khách hàng”: quản lý tất cả thông tin liên quan đến khách hàng.
* Use case “Sổ thu chi”: quản lý các khoản thu chi của công ty.
* Use case “Xem danh sách nhân viên”: Xem danh sách nhân viên của công ty.
* Use case “Tìm nhân viên”: Tìm kiếm nhân viên nào đó trong danh sách các nhân viên.
* Use case “Khoá nhân viên”: Khoá một nhân viên không cho phép sử dụng hệ thống.
* Use case “Mở khoá nhân viên”: Mở khoá một nhân viên.
* Use case “Xem danh sách khách hàng”: Xem danh sách khách hàng của công ty.
* Use case “Tìm khách hàng”: Tìm kiếm khách hàng nào đó trong danh sách khách hàng.
* Use case “Khoá khách hàng”: Khoá một khách không cho phép tương tác với hệ thống và bán hàng cho khách hàng này.
* Use case “Mở khoá khách hàng”: Mở khoá một khách hàng.
* Use case “Xem danh sách nhà cung cấp”: Xem danh sách nhà cung cấp của công ty.
* Use case “Tìm nhà cung cấp”: Tìm kiếm nhà cung cấp nào đó trong danh sách nhà cung cấp.
* Use case “Khoá nhà cung cấp”: Khoá một nhà cung cấp không cho phép tương tác với hệ thống và mua hàng của nhà cung cấp này.
* Use case “Mở khoá nhà cung cấp”: Mở khoá một nhà cung cấp.
* Các use case “Xem chi tiết”, “Thêm”, “Sửa”, “Xoá”: là các use case mở rộng cho phép xem chi tiết thông tin, thêm, sửa, xoá thông tin.

## Thiết kế cơ sở dữ liệu

### Lược đồ ERD



*Hình 8: Lược đồ ERD*

### Chi tiết kiểu dữ liệu

Ánh xạ ERD vào hệ cơ sở dữ liệu MongoDB thì có các kiểu dữ liệu tương ứng cho các thực thể như sau.

1. Nhân viên

Tên Collection là *employees*

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| ID | id | ObjectId |
| Họ tên | first\_name | String |
| Tên | last\_name | String |
| Email | email | String |
| Địa chỉ | address | String |
| Di động | cell\_phone | List<Phone> |
| Nội bộ | desk\_phone | List<Phone> |
| CMND | identity\_card\_number | String |
| Lương | salary | int |
| Ngày sinh | birth\_date | Date |
| Ngày vào lam | join\_date | Date |
| Ngày nghỉ | leaves | List<Leave> |
| Bằng lái | driver\_lincense\_number | String |

1. Khách hàng

Tên Collection là customers

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| ID | id | ObjectId |
| Họ tên | first\_name | String |
| Tên | last\_name | String |
| Email | email | String |
| Địa chỉ | address | String |
| Di động | cell\_phone | List<Phone> |
| Điện thoại bàn | desk\_phone | List<Phone> |
| Công ty đại diện | company\_name | String |
| Ngân hàng | bank | List<Bank> |
| MST | tax\_number | String |
| Hàng trả | goods\_return | List<GoodsReturn> |
| Giá | prices | List<Prices> |

1. Hoá đơn bán hàng

Tên Collection là salesinvoices

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| ID | id | ObjectId |
| Thời điểm | time | Date |
| Cách thanh toán | payment\_method | String |
| Chiết khấu | discount\_percent | Int |
| Nhân viên tạo | employee\_id | ObjectId |
| Khách hàng | customer\_id | ObjectId |
| Phiếu đặt hàng | sales\_order\_id | ObjectId |
| Giao hàng | delivery | List<Delivery> |

1. Phiếu đặt hàng

Tên Collection là salesorders

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| Id | id | ObjectId |
| Thời điểm | time | Date |
| Khách hàng | customer\_id | ObjectId |
| Hàng bán | goods | List<Goods> |

1. Loại vải

Tên Collection là fabrics

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| Id | id | ObjectId |
| Tên | Name | String |
| Mô tả | description | string |

1. Loại vải màu

Tên Collection là fabricscolors

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| Mã màu | id | ObjectId |
| Tên màu | name | string |
| Loại vải | fabrics\_id | ObjectId |

1. Cây vải

Tên Collection là fabricsrolls

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| Id | id | ObjectId |
| Chiều dài | lenght | int |
| Loại vải màu | fabrics\_color\_id | ObjectId |
| Lô nhuộm | plots\_id | ObjecId |
| Hoá đơn bán hàng | sales\_invoice\_id | ObjectId |
| Hoá đơn mua hàng | purchase\_invoice\_id | ObjectId |

1. Lô nhuộm

Tên Collection là fabricsplotss

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| Id | id | ObjectId |
| Ngày nhuộm | time | Date |

1. Phiếu đặt mua hàng

Tên Collection là purchaseorders

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| Id | id | ObjectId |
| Thời điểm | time | Date |
| Nhân viên | employee\_id | ObjectId |
| Hàng mua | goods | List<Goods> |
| Trạng thái | status | int |

1. Nhà cung cấp

Tên Collection là suppliers

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| ID | id | ObjectId |
| Họ tên | first\_name | String |
| Tên | last\_name | String |
| Email | email | String |
| Địa chỉ | address | String |
| Di động | cell\_phone | List<Phone> |
| Điện thoại bàn | desk\_phone | List<Phone> |
| Công ty đại diện | company\_name | String |
| Ngân hàng | bank | List<Bank> |
| MST | tax\_number | String |
| Hàng trả | goods\_return | List<GoodsReturn> |
| Giá | prices | List<Prices> |

1. Thanh toán

Tên Collection là payments

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| ID | id | ObjectId |
| Số tiền | money | double |
| Hình thức thanh toán | payment\_method | Int |
| Người nhận hoặc gửi | recipient\_or\_sender | String |
| Tên Ngần hàng | bank\_name | String |
| Tài khoản | account\_number | String |
| Thu hoặc chi | payment\_type | Int |
| Thời gian | time | Date |
| Nhân viên | employee\_id | ObjectId |
| Khách hàng | customer\_id | ObjectId |
| Nhà cung cấp | supplier\_id | ObjectId |

1. Kho

Tên Collection là warehouses

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| ID | id | ObjectId |
| Tên | name | String |
| Diện tích | areage | int |
| Địa chỉ | address | String |

1. Hoá đơn mua hàng

Tên Collection là purchaseinvoices

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| ID | id | ObjectId |
| Thời điểm | time | Date |
| Cách thanh toán | payment\_method | String |
| Chiết khấu | discount\_percent | Int |
| Nhân viên tạo | employee\_id | ObjectId |
| Nhà cung cấp | supplier\_id | ObjectId |
| Phiếu đặt mua hàng | purchase\_order\_id | ObjectId |
| Kho | warehouse\_id | ObjectId |

1. Sổ thu chi

Tên Collection là records

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| ID | id | ObjectId |
| Số tiền | money | double |
| Thời điểm | time | Date |
| Diễn giải | note | string |

1. Ngân hàng

Tên model là Bank, được lưu thẳng vào document trong collection của khách hàng và nhà cung cấp.

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| Tên ngân hàng | bank\_name | String |
| Tài khoản | account\_number | string |

1. Giao hàng

Tên model là Delivery, được lưu thẳng vào document trong collection của hoá đơn bán hàng.

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| Nhân viên giao hàng | shipper\_id | ObjectId |
| Kiểu đóng gói | pack\_method | String |
| Địa chỉ | address | String |
| Số lượng | total | int |
| Thời điểm | time | Date |

1. Điện thoại

Tên model là Phone, được lưu thẳng vào document trong collection của nhân viên, khách hàng và nhà cung cấp.

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| Số điện thoại | phone | String |

1. Ngày nghỉ

Tên model là Leave, được lưu thẳng vào document trong collection của nhân viên.

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| Ngày nghỉ | date | Date |

Hàng hoá

Tên model là Leave, được lưu thẳng vào document trong collection của Hoá đơn mua hàng và hoá đơn bán hàng.

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| Loại vải màu | fabrics\_color\_id | ObjectId |
| Tổng mét | total\_metter | int |
| Tổng cây | total\_roll | int |

1. Giá

Tên model là Prices, được lưu thẳng vào document trong collection của Khách hàng và nhà cung cấp.

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| Loại vải màu | fabrics\_color\_id | ObjectId |
| Thời gian | time | Date |
| Giá | price | int |

1. Hàng trả

Tên model là GoodsReturn, được lưu thẳng vào document trong collection của Khách hàng và nhà cung cấp.

|  |  |  |
| --- | --- | --- |
| Tên thuộc tính | Tên lưu trong document | Kiểu dữ liệu |
| Cây vải | Fabrics\_roll\_id | ObjectId |
| Ngày trả | date | Datea |

## Thiết kế famework

### Cấu trúc tổng quan



*Hình 9: Cấu trúc tổng quan framework*

Mô tả:

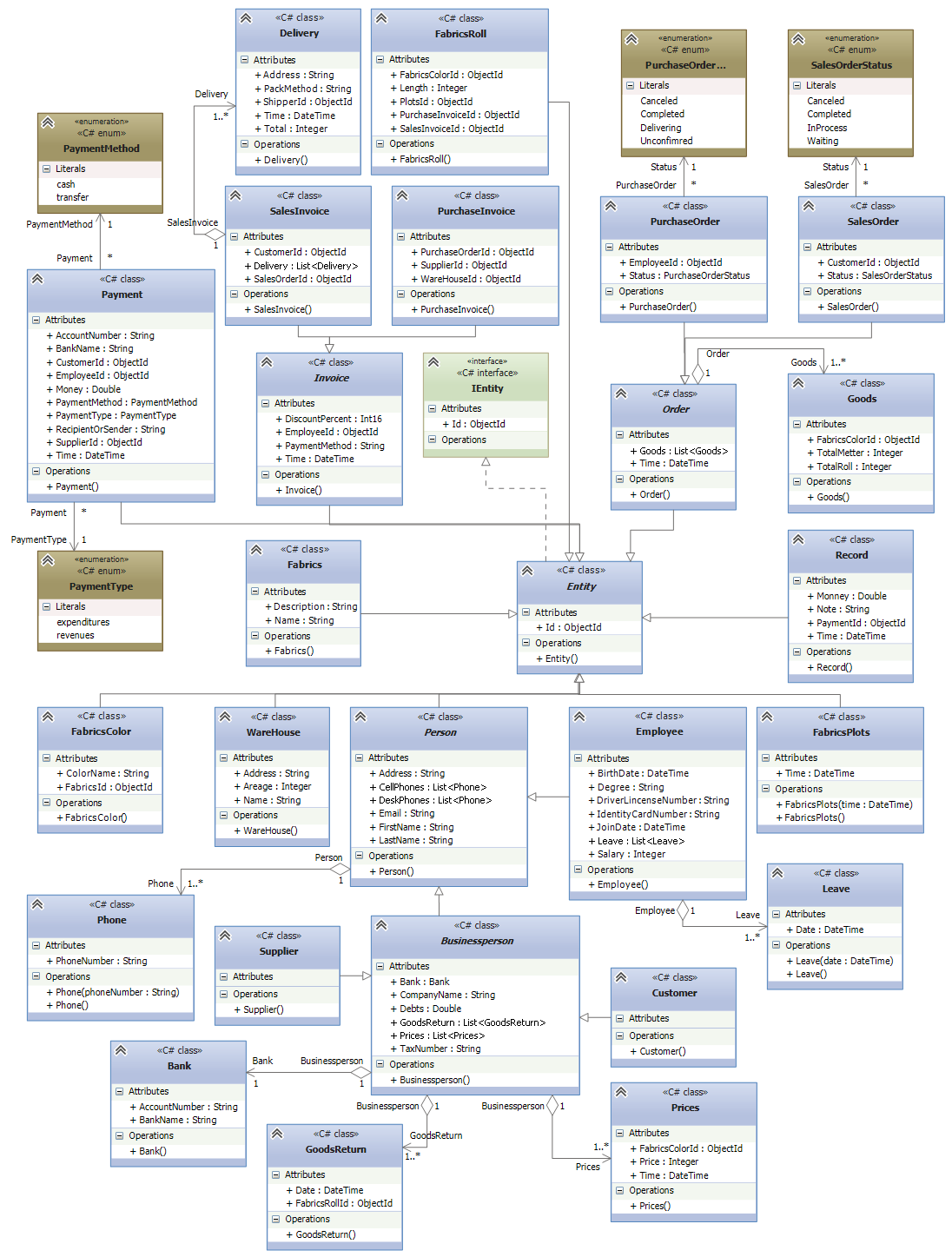
Framework gồm các Model tương ứng với các thực thể trong nghiệp vụ mua bán vải.

Service là các lớp cung cấp các dịch vụ liên quan đến Model và tương tác với cơ sở dữ liệu.

Connection là lớp liên kết tạo kết nối tới database.

### Mô hình class diagram

**Mô hình class diagram của Entities**



*Hình 10: Mô hình class diagram cho Entities*

Mô tả class diagram:

1. Class Bank

Mô tả: là lớp model đại diện cho các đối tượng Ngân hàng.

Gồm các thuộc tính:

* BankName : string
* AccountNumber : string

Gồm các phương thức:

* Bank : Contructor

Tham số : Không.

Trả về : Không.

1. Class Businessperson

Mô tả: kế thừa class Person, là lớp trừu tượng cho đối tượng là người liên quan đến kinh doanh.

Gồm các thuộc tính:

* CompanyName : string
* TaxNumber : string
* Bank : Bank
* GoodsReturn : List<GoodsReturn>
* Prices : List<Prices>

Gồm các phương thức:

* Businessperson : Contructor

Tham số : Không.

Trả về : Không.

1. Class Customer

Mô tả: kế thừa class Businessperson, là lớp model cho đối tượng là khách hàng.

Gồm các thuộc tính: không có thuộc tính riêng.

Gồm các phương thức:

* Customer : Contructor

Tham số : Không.

Trả về : Không.

1. Class Delivery

Mô tả: là lớp model cho đối tượng vận chuyển hàng hoá.

Gồm các thuộc tính:

* ShipperId : ObjectId
* PackMethod : string
* Address : string
* Total : int
* Time : DateTime

Gồm các phương thức:

* Delivery : Contructor

Tham số : Không.

Trả về : Không.

1. Class Employee

Mô tả: kế thừa class Person, là lớp model cho đối tượng là nhân viên công ty.

Gồm các thuộc tính:

* BirthDate : DateTime
* IdentityCardNumber : string
* Salary : int
* JoinDate : DateTime
* Degree : string
* DriverLincenseNumber : string
* Leave : List<Leave>

Gồm các phương thức:

* Employee : Contructor

Tham số : Không.

Trả về : Không.

1. Class Entity

Mô tả: implement interface IEntity, là lớp trừu tượng cho đối tượng chính có Id.

Gồm các thuộc tính:

* Id : ObjectId

Gồm các phương thức: Không.

1. Class Fabrics

Mô tả: kế thừa class Entity, là lớp model cho đối tượng Loại vải.

Gồm các thuộc tính:

* Name : string
* Dscription : string

Gồm các phương thức:

* Fabrics : Contructor

Tham số : Không.

Trả về : Không.

1. Class FabricsColor

Mô tả: kế thừa class Entity, là lớp model cho đối tượng Loại vải màu.

Gồm các thuộc tính:

* ColorName : string
* FabricsId : ObjectId

Gồm các phương thức:

* FabricsColor : Contructor

Tham số : Không.

Trả về : Không.

1. Class FabricsPlots

Mô tả: kế thừa class Entity, là lớp model cho đối tượng Lô Vải.

Gồm các thuộc tính:

* Time : DateTime

Gồm các phương thức:

* FabricsPlots : Contructor

Tham số : Không.

Trả về : Không.

1. Class FabricsRoll

Mô tả: kế thừa class Entity, là lớp model cho đối tượng Cây vải.

Gồm các thuộc tính:

* Length : int
* FabricsColorId : ObjectId
* PlotsId : ObjectId
* PurchaseInvoiceId : ObjectId
* SalesInvoiceId : ObjectId

Gồm các phương thức:

* FabricsRoll : Contructor

Tham số : Không.

Trả về : Không.

1. Class Goods

Mô tả: là lớp model cho đối tượng hàng hoá.

Gồm các thuộc tính:

* FabricsColorId : ObjectId
* TotalMetter : int
* TotalRoll : int

Gồm các phương thức:

* Goods : Contructor

Tham số : Không.

Trả về : Không.

1. Class GoodsReturn

Mô tả: là lớp model cho đối tượng hàng trả.

Gồm các thuộc tính:

* FabricsRollId : ObjectId
* Date : DateTime

Gồm các phương thức:

* GoodsReturn : Contructor

Tham số : Không.

Trả về : Không.

1. Interfacce IEntity

Mô tả: Là interface đại diện cho các thực thể chính.

Gồm các thuộc tính:

* Id : ObjectId

Gồm các phương thức: không có.

1. Class Invoice

Mô tả: kế thừa class Entity, là lớp trừu tượng cho đối tượng hoá đơn.

Gồm các thuộc tính:

* Time : DateTime
* PaymentMethod : string
* DiscountPercent : short
* EmployeeId : ObjectId

Gồm các phương thức:

* Invoice : Contructor

Tham số : Không.

Trả về : Không.

1. Class PurchaseInvoice

Mô tả: kế thừa class Invoice, là lớp model cho đối tượng hoá đơn mua hàng.

Gồm các thuộc tính:

* WareHouseId : ObjectId
* SupplierId : Objectid
* PurchaseOrderId : ObjectId

Gồm các phương thức:

* PurchaseInvoice : Contructor

Tham số : Không.

Trả về : Không.

1. Class SalesInvoice

Mô tả: kế thừa class Invoice, là lớp model cho đối tượng hoá đơn bán hàng.

Gồm các thuộc tính:

* CustomerId : ObjectId
* SalesOrderId : Objectid
* Delivery : List<Delivery>

Gồm các phương thức:

* SalesInvoice : Contructor

Tham số : Không.

Trả về : Không.

1. Class Leave

Mô tả: là lớp model cho đối tượng Nghỉ phép.

Gồm các thuộc tính:

* Date : DateTime

Gồm các phương thức:

* Leave: Contructor

Tham số : Không.

Trả về : Không.

1. Class Order

Mô tả: kế thừa class Entity, là lớp trừu tượng cho đối tượng đơn đặt hàng.

Gồm các thuộc tính:

* Time : DateTime
* Goods : List<Goods>

Gồm các phương thức:

* Order : Contructor

Tham số : Không.

Trả về : Không.

1. Class Payment

Mô tả: kế thừa class Entity, là lớp model cho đối tượng thanh toán.

Gồm các thuộc tính:

* Time : DateTime
* Money : double
* PaymentMethod : PaymentMethod
* RecipentOrSender : string
* BankName : string
* AccountNumber : string
* PaymentType : PaymentType
* EmployeeId : ObjectId
* CustomerId : ObjectId
* SupplierId : ObjectId

Gồm các phương thức:

* Payment : Contructor

Tham số : Không.

Trả về : Không.

1. Class Person

Mô tả: kế thừa class Entity, là lớp trừu tượng cho đối tượng là con người.

Gồm các thuộc tính:

* CellPhones : List<Phone>
* DeskPhones : List<Phone>
* FirstName : string
* LastName : string
* Email : string
* Address : string

Gồm các phương thức:

* Person : Contructor

Tham số : Không.

Trả về : Không.

1. Class Phone

Mô tả: là model cho đối tượg số điện thoại.

Gồm các thuộc tính:

* PhoneNumber : string

Gồm các phương thức:

* Phone : Contructor

Tham số : Không.

Trả về : Không.

1. Class Prices

Mô tả: là lớp model cho đối tượng giá bán cho từng khách hàng, và giá nhập theo từng nhà cung cấp.

Gồm các thuộc tính:

* Time : DateTime
* FabricsColorId : ObjectId
* Price : int

Gồm các phương thức:

* Prices : Contructor

Tham số : Không.

Trả về : Không.

1. Class PurchaseOrder

Mô tả: kế thừa class Order, là lớp model cho đối tượng phiếu đặt mua hàng.

Gồm các thuộc tính:

* EmployeeId : ObjectId
* Status : PurchaseOrderStatus

Gồm các phương thức:

* PurchaseOrder : Contructor

Tham số : Không.

Trả về : Không.

Class Class SalesOrder

Mô tả: kế thừa class Order, là lớp model cho đối tượng phiếu đặt hàng.

Gồm các thuộc tính:

* CustomerId : ObjectId
* Status : SalesOrderStatus

Gồm các phương thức:

* SalesOrder : Contructor

Tham số : Không.

Trả về : Không.

1. Class Record

Mô tả: kế thừa class Entity, là lớp model cho đối tượng sổ thu chi.

Gồm các thuộc tính:

* PaymentId : ObjectId
* Money : double
* Time : DateTime
* Note : string

Gồm các phương thức:

* Record : Contructor

Tham số : Không.

Trả về : Không.

1. Class Supplier

Mô tả: kế thừa class Businessperson, là lớp model cho đối tượng nhà cung cấp.

Gồm các thuộc tính: không

Gồm các phương thức:

* Supplier : Contructor

Tham số : Không.

Trả về : Không.

1. Class WareHouse

Mô tả: kế thừa class Entity, là lớp model cho đối tượng kho hàng.

Gồm các thuộc tính:

* Name : string
* Areage : int
* Address : string

Gồm các phương thức:

* WareHouse : Contructor

Tham số : Không.

Trả về : Không.

**Mô hình class diagram của Services**



*Hình 11: Mô hình class diagram cho Services*

Mô tả class diagram:

1. BusinesspersonService

Mô tả: kế thừa class PersonService, là class trừu tượng cung cấp các dịch vụ cho đối tượng là người kinh doanh.

Thuộc tính : không.

Gồm các phương thức:

* AddFabricsPrice

Tham số :

* id : string
* price : Prices

Trả về : không.

* AddFabricsPrice

Tham số :

* id : string
* price : List<Prices>

Trả về : không.

1. Connection

Mô tả: là class cung cấp các dịch vụ kết nối tới database.

Thuộc tính :

* \_connection : MongoClient

Gồm các phương thức:

* Connection : Contructor

Tham số : không

Trả về : không.

* CreateConnection

Tham số :

* connectionString: string

Trả về : không.

1. CustomerService

Mô tả: kế thừa class BusinesspersonService, là class cung cấp các dịch vụ cho đối tượng khách hàng.

Thuộc tính : không

Gồm các phương thức:

* CustomerService: Contructor

Tham số : không

Trả về : không.

* GetDebt

Tham số :

* id : string

Trả về : kiểu double, số nợ của khách hàng.

1. Database

Mô tả: kế thừa class BusinesspersonService, là class cung cấp các dịch vụ cho đối tượng khách hàng.

Thuộc tính :

* database: IMongoDatabase

Gồm các phương thức:

* Database: Contructor

Tham số : không

Trả về : không.

1. EmployeeService

Mô tả: kế thừa class PersonService, là class cung cấp các dịch vụ cho đối tượng nhân viên.

Thuộc tính : không

Gồm các phương thức:

* EmployeeService: Contructor

Tham số : không

Trả về : không.

* AddLeave

Tham số :

* id : string
* leave : Leave

Trả về : không.

* AddLeave

Tham số :

* id : string
* leave : List<Leave>

Trả về : không.

* Géthipper

Tham số : không

Trả về : List<Employee>, danh sách nhân viên giao hàng.

1. EntityService

Mô tả: kế thừa class Database vả hiện thực interface IEntityService, là class trừu tượng cung cấp các dịch vụ cơ bản như thêm, sửa, xoá, select cơ bản trên collection.

Thuộc tính :

* Collection : IMongoCollection

Gồm các phương thức:

* EntityService: Contructor

Tham số : không

Trả về : không.

* CreateIndex

Tham số :

* fields : string[]

Trả về : không.

* Create

Tham số :

* entity : Template

Trả về : không.

* Create

Tham số :

* entity : List<Template>

Trả về : không.

* Delete

Tham số :

* id : string

Trả về : bool.

* GetById

Tham số :

* id: string

Trả về : Template.

* GetAll

Tham số : không.

Trả về : List<Template>.

* Update

Tham số :

* entity: Template

Trả về : không.

1. FabricsColorService

Mô tả: kế thừa class EntityService, là class cung cấp các dịch vụ cho đối tượng loại vải màu.

Thuộc tính : không

Gồm các phương thức:

* FabricsColorService: Contructor

Tham số : không

Trả về : không.

* GetFabricsColorForFabricsType

Tham số :

* fabricsTypeId : string

Trả về : kiểu List<FabricsColor>, Danh sách các loại vải màu.

1. FabricsPlotsService

Mô tả: kế thừa class EntityService, là class cung cấp các dịch vụ cho đối tượng lô nhuộm.

Thuộc tính : không

Gồm các phương thức:

* FabricsPlotsService: Contructor

Tham số : không

Trả về : không.

* GetFabricsPlotsByDate

Tham số :

* fromDate : DateTime
* toDate : DateTime

Trả về : kiểu List<FabricsPlots>, danh sách lô nhuộm trong khoảng thời gian.

1. FabricsRollService

Mô tả: kế thừa class EntityService, là class cung cấp các dịch vụ cho đối tượng cây vải.

Thuộc tính : không

Gồm các phương thức:

* FabricsRollService: Contructor

Tham số : không

Trả về : không.

* GetFabricsRollForSalesInvoice

Tham số :

* salesInvoiceId : string

Trả về : kiểu List<FabricsRoll>, danh sách các cây vải theo hoá đơn bán hàng.

* GetFabricsRollForPurchaseInvoice

Tham số :

* purchaseInvoiceId : string

Trả về : kiểu List<FabricsRoll>, danh sách các cây vải theo hoá đơn mua hàng.

* GetFabricsRollForFabricsColor

Tham số :

* fabricsColorId : string

Trả về : kiểu List<FabricsRoll>, danh sách các cây vải theo loại vải màu.

* GetFabricsRollForFabricsPlots

Tham số :

* plotsId: string

Trả về : kiểu List<FabricsRoll>, danh sách các cây vải theo lô nhuộm.

1. FabricsService

Mô tả: kế thừa class EntityService, là class cung cấp các dịch vụ cho đối tượng loại vải.

Thuộc tính : không

Gồm các phương thức:

* FabricsService: Contructor

Tham số : không

Trả về : không.

1. IEntityService

Mô tả: là một interface.

Thuộc tính : không

Gồm các phương thức:

* Create

Tham số :

* entity : Template

Trả về : không.

* Delete

Tham số :

* id : string

Trả về : bool.

* GetById

Tham số :

* id: string

Trả về : Template.

* Update

Tham số :

* entity: Template

Trả về : không.

1. InvoiceService

Mô tả: kế thừa class EntityService, là class trừu tượng cung cấp các dịch vụ cho đối tượng hoá đơn.

Thuộc tính : không

Gồm các phương thức:

* GetFabricsRoll

Tham số :

* id : string

Trả về : kiểu List<FabricsRoll>, danh sách các cây vải thuộc hoá đơn.

* GetTotalMoney

Tham số :

* entity : Template

Trả về : kiểu double, tổng tiền hoá đơn.

* GetInvoiceByDate

Tham số :

* id : string

Trả về : kiểu List<FabricsRoll>, danh sách các hoá đơn trong khoảng thời gian.

1. OrderService

Mô tả: kế thừa class EntityService, là class trừu tương cung cấp các dịch vụ cho đối tượng đơn đặt hàng.

Thuộc tính : không

Gồm các phương thức:

* AddGoods

Tham số :

* id : string
* goods : Goods

Trả về : không.

* AddGoods

Tham số :

* id : string
* goods : List<Goods>

Trả về : không.

* GetOrderByDate

Tham số :

* fromDate : DateTime
* toDate : DateTime

Trả về : List<Template>.

1. PamentService

Mô tả: kế thừa class EntityService, là class cung cấp các dịch vụ cho đối tượng thanh toán.

Thuộc tính : không

Gồm các phương thức:

* PaymentService: Contructor

Tham số : không

Trả về : không.

* Create

Tham số :

* entity : Payment

Trả về : không.

* GetPaymentForSupplier

Tham số :

* id : string

Trả về : List<Payment>, danh sách phiếu thanh toán cho nhà cung cấp.

* GetPaymentForCustomer

Tham số :

* id : string

Trả về : List<Payment>, danh sách phiếu thanh toán của khách hàng.

* GetPaymentByDate

Tham số :

* fromDate : DateTime
* toDate : DateTime

Trả về : List<Payment>, danh sách phiếu thanh toán trong khoảng thời gian.

1. PersonService

Mô tả: kế thừa class EntityService, là class trừu tượng cung cấp các dịch vụ cho đối tượng là con người.

Thuộc tính : không

Gồm các phương thức:

* AddCellPhone

Tham số :

* id : string
* phone : Phone

Trả về : không.

* AddCellPhone

Tham số :

* id : string
* phones : List<Phone>

Trả về : không.

* AddDeskPhone

Tham số :

* id : string
* phone : Phone

Trả về : không.

* AddDeskPhone

Tham số :

* id : string
* phones : List<Phone>

Trả về : không.

1. PurchaseInvoiceService

Mô tả: kế thừa class EntityService, là class cung cấp các dịch vụ cho đối tượng hoá đơn mua hàng.

Thuộc tính : không

Gồm các phương thức:

* PurchaseInvoiceService: Contructor

Tham số : không

Trả về : không.

* GetInvoiceForWareHouse

Tham số :

* id : string

Trả về : List<PurchaseInvoice>, danh sách hoá đơn theo kho nhập vào.

GetInvoiceForSupplier

Tham số :

* id : string

Trả về : List<PurchaseInvoice>, danh sách hoá đơn theo nhà cung cấp.

* GetFabricsRoll

Tham số :

* id : string

Trả về : List<FabricsRoll>, danh sách cây vải trong hoá đơn.

* GetTotalMoney

Tham số :

* entity : PurchaseInvoice

Trả về : double, tổng tiền hoá đơn.

1. PurchaseOrderService

Mô tả: kế thừa class EntityService, là class cung cấp các dịch vụ cho đối tượng đơn đặt mua hàng.

Thuộc tính : không

Gồm các phương thức:

* PurchaseOrderService: Contructor

Tham số : không

Trả về : không.

* Create

Tham số :

* entity : PurchaseOrder

Trả về : không.

* UpdateStatus

Tham số :

* status: PurchaseOrderStatus
* id : string

Trả về : không.

1. RecordService

Mô tả: kế thừa class EntityService, là class cung cấp các dịch vụ cho đối tượng sổ thu chi.

Thuộc tính : không

Gồm các phương thức:

* RecordService : Contructor

Tham số : không

Trả về : không.

* GetRecordByDate

Tham số :

* fromDate : DateTime
* toDate : DateTime

Trả về : List<Record>, danh sách các khoản thu chi trong khoảng thời gian.

1. SalesInvoiceService

Mô tả: kế thừa class EntityService, là class cung cấp các dịch vụ cho đối tượng hoá đơn bán hàng.

Thuộc tính : không

Gồm các phương thức:

* SalesInvoiceService : Contructor

Tham số : không

Trả về : không.

* GetInvoiceForCustomer

Tham số :

* id : string

Trả về : List<SalesInvoice>, danh sách hoá đơn theo nhà khách hàng.

* GetFabricsRoll

Tham số :

* id : string

Trả về : List<FabricsRoll>, danh sách cây vải trong hoá đơn.

* GetTotalMoney

Tham số :

* entity : PurchaseInvoice

Trả về : double, tổng tiền hoá đơn.

* AddDelivery

Tham số :

* id : string
* delivery : Delivery

Trả về : không.

* AddDelivery

Tham số :

* id : string
* delivery: List< delivery>

Trả về : không.

* GetDelivery

Tham số :không.

Trả về : List<Delivery>, danh sách giao hàng của hoá đơn.

1. SalesOrderService

Mô tả: kế thừa class EntityService, là class cung cấp các dịch vụ cho đối tượng đơn đặt hàng.

Thuộc tính : không

Gồm các phương thức:

* SalesOrderService : Contructor

Tham số : không

Trả về : không.

* Create

Tham số :

* entity : SalesOrder

Trả về : không.

* UpdateStatus

Tham số :

* status: SalesOrderStatus
* id : string

Trả về : không.

1. SupplierService

Mô tả: kế thừa class EntityService, là class cung cấp các dịch vụ cho đối tượng nhà cung cấp.

Thuộc tính : không

Gồm các phương thức:

* SupplierService : Contructor

Tham số : không

Trả về : không.

* GetDebt

Tham số :

* id : string

Trả về : kiểu double, số nợ nhà cung cấp.

1. WareHouseService

Mô tả: kế thừa class EntityService, là class cung cấp các dịch vụ cho đối tượng kho chứa.

Thuộc tính : không

Gồm các phương thức:

* WareHouseService : Contructor

Tham số : không

Trả về : không.

* GetFabricsRoll

Tham số :

* id : string

Trả về : kiểu List<FabricsRoll>, danh sách các cây vải trong kho.

* GetRemainFabricsRoll

Tham số :

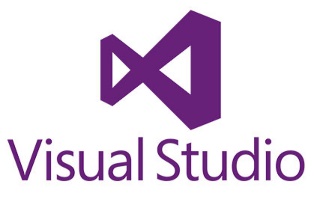
* idWareHouse : string

Trả về : kiểu List<FabricsRoll>, danh sách các cây vải còn lại trong kho.

# HIỆN THỰC

## Công cụ phát triển

**Visual Studio 2012**



Microsoft Visual Studio là một [môi trường phát triển tích hợp](https://vi.wikipedia.org/wiki/M%C3%B4i_tr%C6%B0%E1%BB%9Dng_ph%C3%A1t_tri%E1%BB%83n_t%C3%ADch_h%E1%BB%A3p) (IDE) từ [Microsoft](https://vi.wikipedia.org/wiki/Microsoft). Nó được sử dụng để phát triển [chương trình máy tính](https://vi.wikipedia.org/wiki/Ch%C6%B0%C6%A1ng_tr%C3%ACnh_m%C3%A1y_t%C3%ADnh) cho Microsoft Windows, cũng như các trang web, các ứng dụng web và các dịch vụ web. Visual Studio sử dụng nền tảng phát triển phần mềm của Microsoft như [Windows API](https://vi.wikipedia.org/w/index.php?title=Windows_API&action=edit&redlink=1), [Windows Forms](https://vi.wikipedia.org/w/index.php?title=Windows_Forms&action=edit&redlink=1), [Windows Presentation Foundation](https://vi.wikipedia.org/wiki/Windows_Presentation_Foundation), [Windows Store](https://vi.wikipedia.org/wiki/Windows_Store)và [Microsoft Silverlight](https://vi.wikipedia.org/wiki/Microsoft_Silverlight).



**Git là** một trong những Hệ thống Quản lý Phiên bản Phân tán, vốn được phát triển nhằm quản lý mã nguồn (source code) của Linux. Trên **Git**, ta có thể lưu trạng thái của file dưới dạng lịch sử cập nhật.

## Xây dựng framework

Dựa trên các thiết kế xây dựng framework cho công ty dịch vụ buôn bán và sản xuất vải.

Quá trình xây dựng cho ra source code là project visual studio có tên FabricsFactoryMethodPattern được ghi và đĩa chương trình đính kèm bản báo cáo.

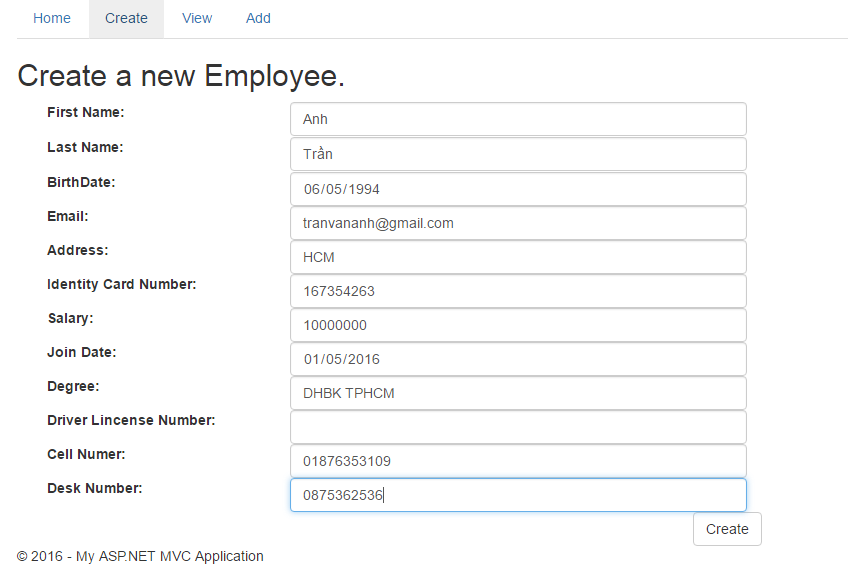
Khi build project ở trên ta có file FabricsFactoryMethodPattern.dll sẽ là thư viện được sử dụng khi xây dựng ứng dụng.

Demo cho framework được xây dựng dựa trên ASP.NET. Project demo có tên FabricsWebApplication được ghi vào đĩa đính kèm.

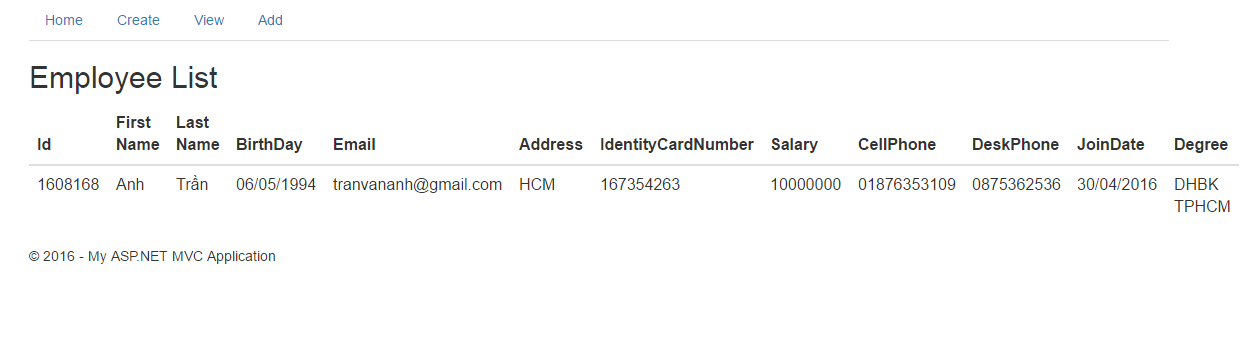
# DEMO ỨNG DỤNG XÂY DỰNG DỰA TRÊN FRAMEWORK

## Nghiệp vụ quản lý

### Nhân viên

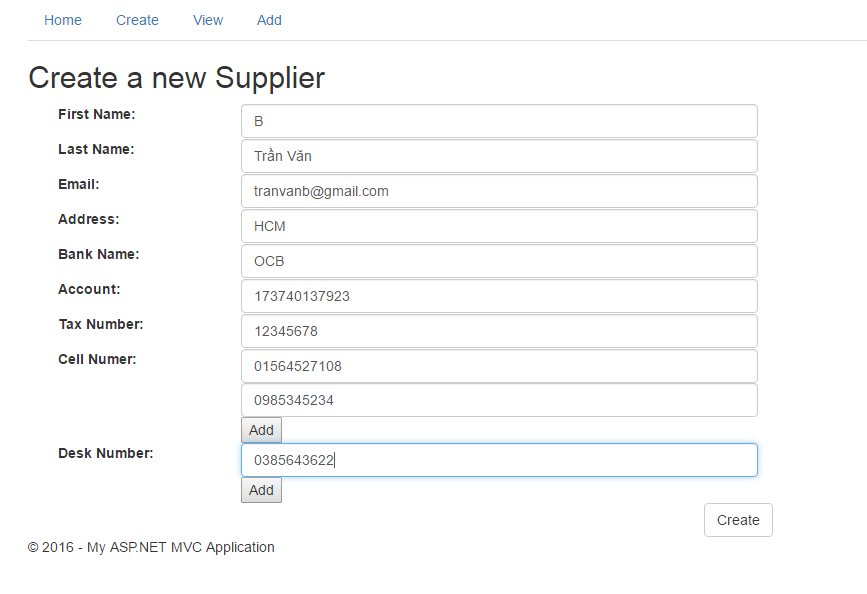


*Hình 12: Tạo 1 nhân viên*

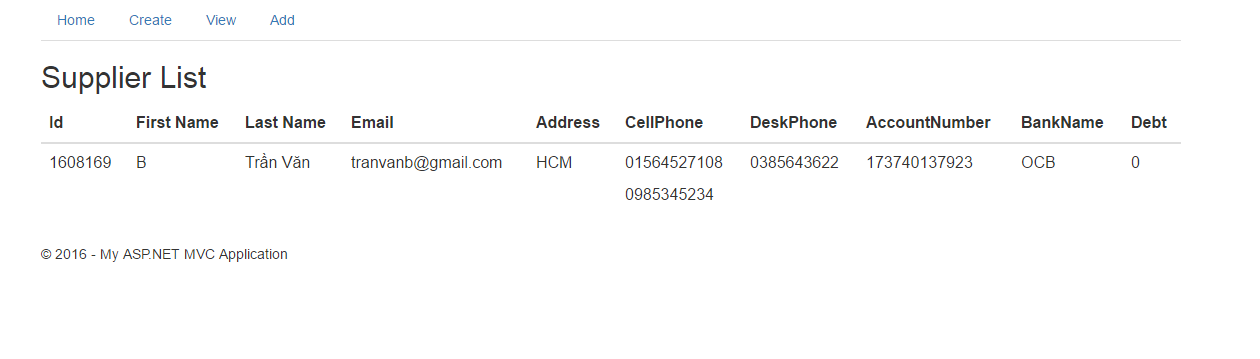


*Hình 13 : Danh sách nhân viên*

### 1.2. Nhà cung cấp

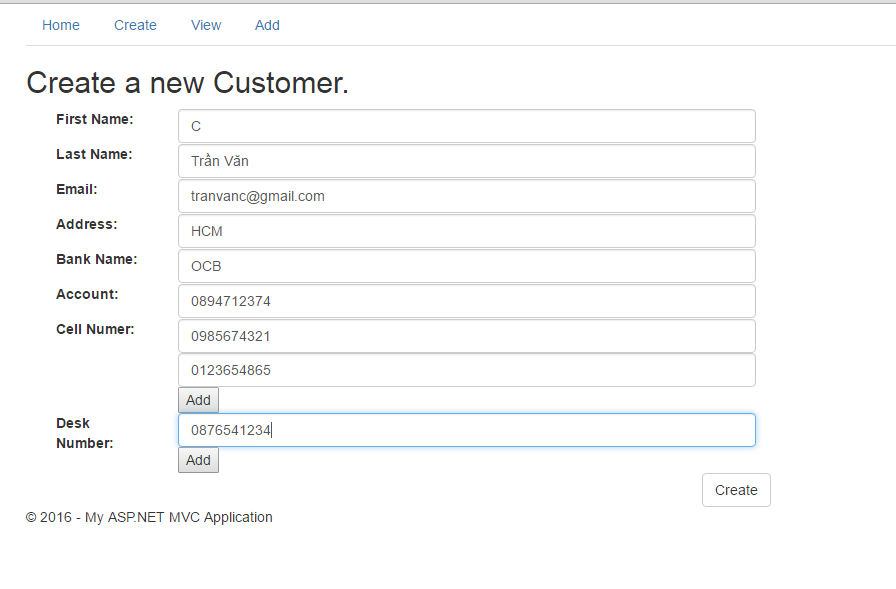


*Hình 14: Tạo 1 nhà cung cấp*

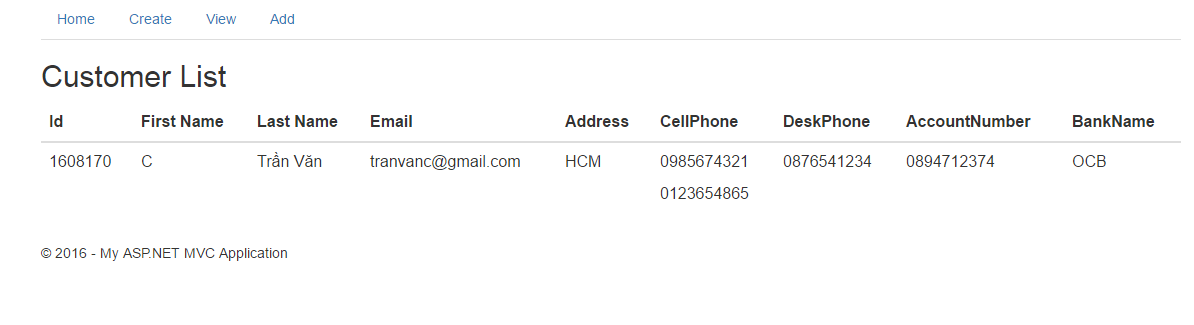
**

*Hình 15: Danh sách nhà cung cấp*

### 1.3. Khách hàng



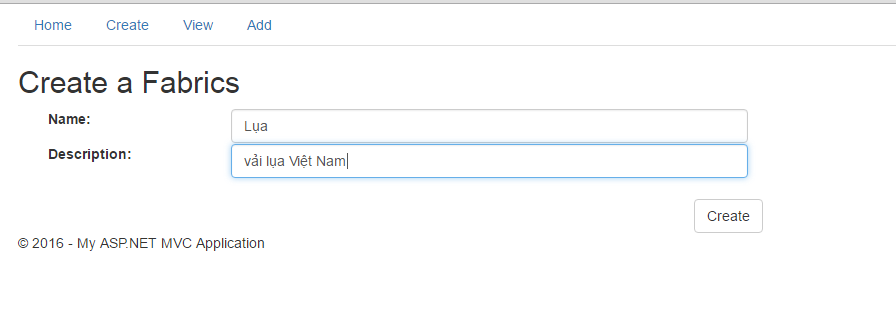
*Hình 16: Tạo 1 khách hàng*

**

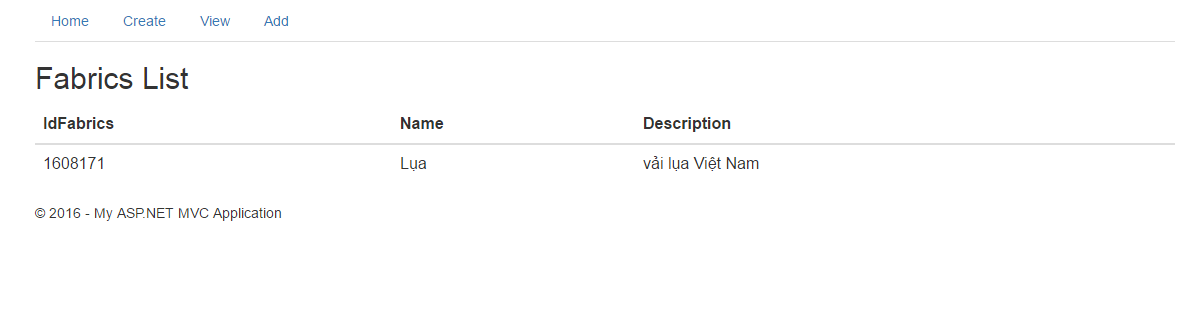
*Hình 17 : Danh sách khách hàng*

## Nghiệp vụ mua hàng

### Loại vải

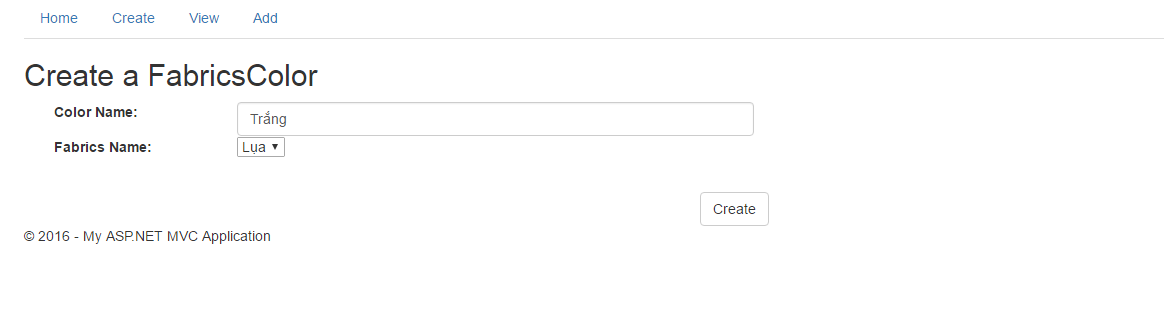


*Hình 18 : Tạo loại vải*

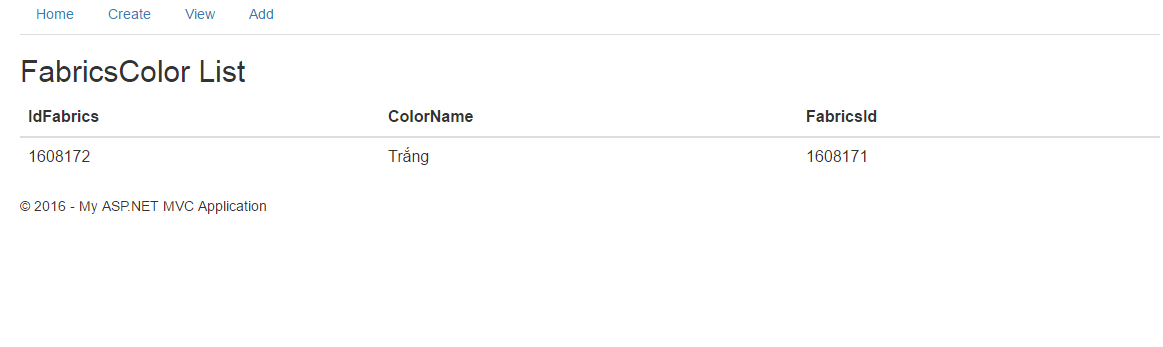
**

*Hình 19 : Danh sách loại vải*

### Loại vải màu

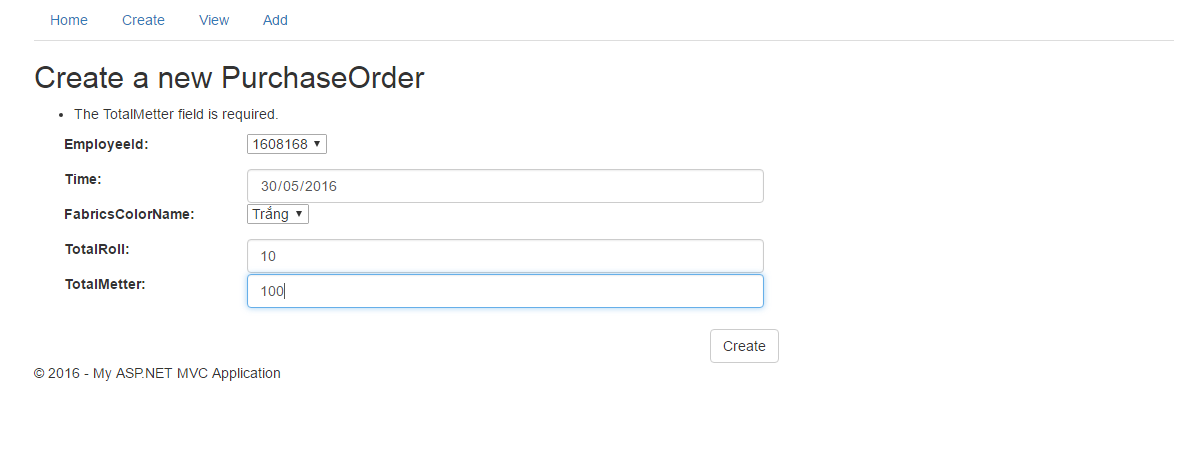


*Hình 20 : Tạo loại vải màu*

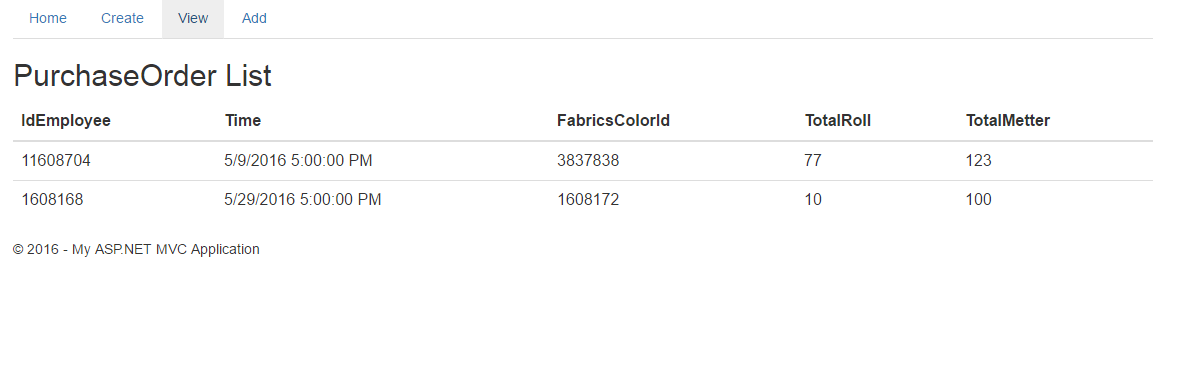
**

*Hình 21 : Danh sách loại vải màu*

### Phiếu đặt hàng

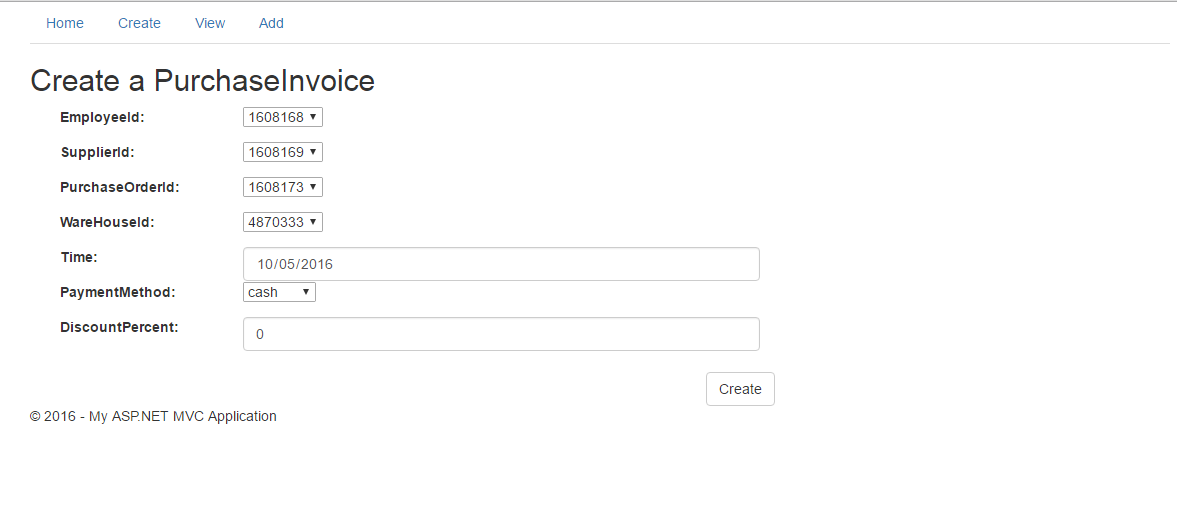


*Hình 22 : Tạo phiếu đặt hàng*

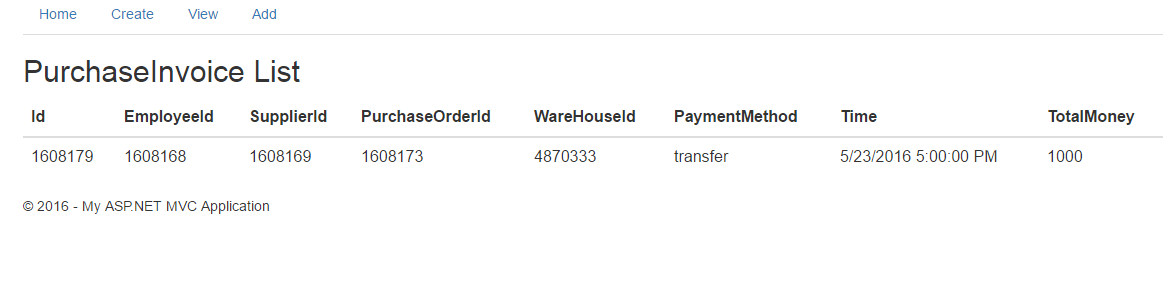
**

*Hình 23 : Danh sách phiếu đặt hàng*

### 2.4. Hóa đơn mua hàng

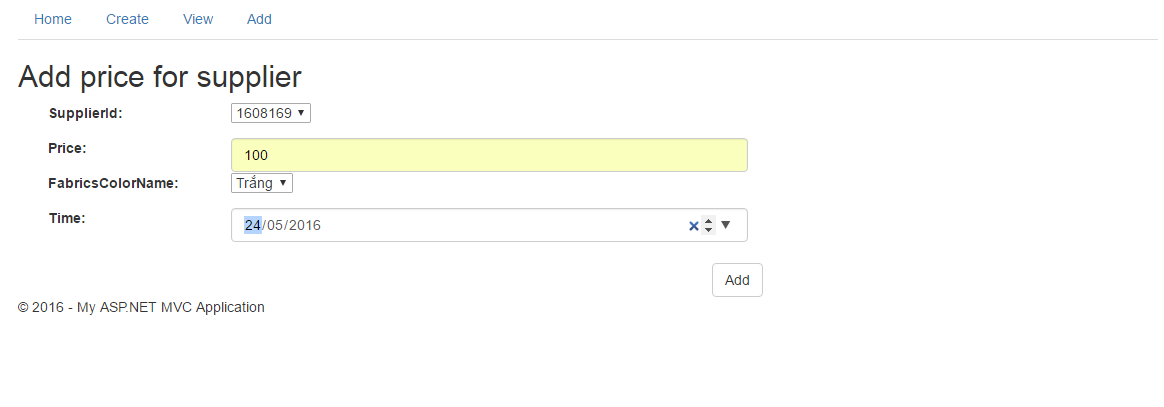


*Hình 24 : Tạo hóa đơn mua hàng*

**

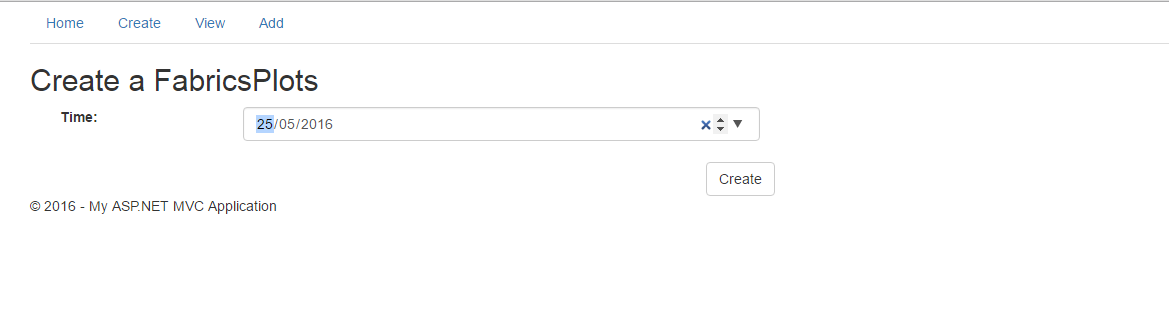
*Hình 25 : Danh sách hóa đơn mua hàng*

### Thêm giá hàng hóa

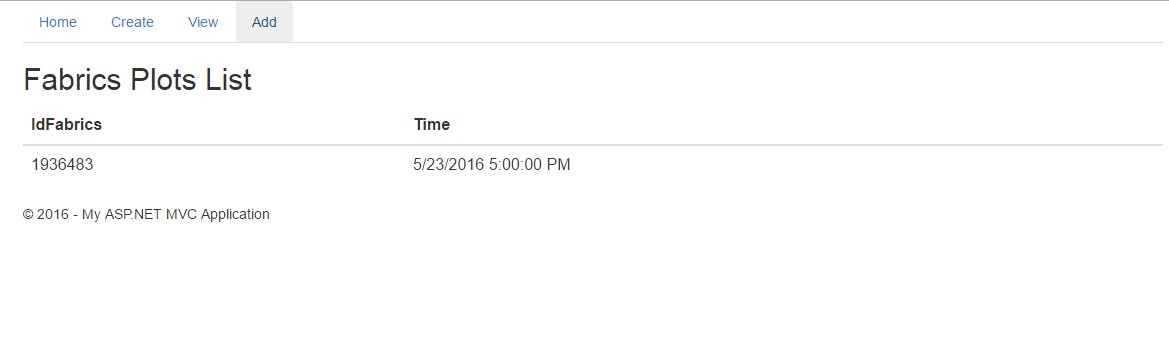
**

*Hình 26 : Thêm giá cho nhà cung cấp*

### Lô hàng

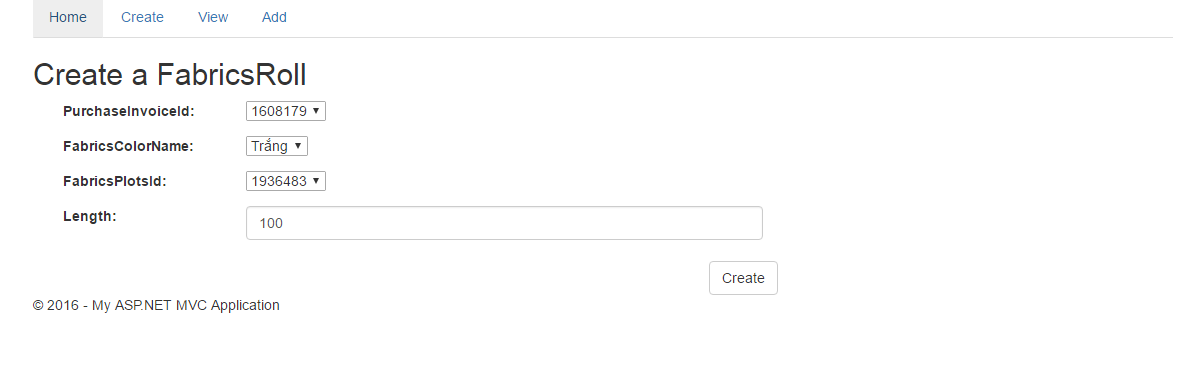
**

*Hình 27 : Tạo lô hàng*

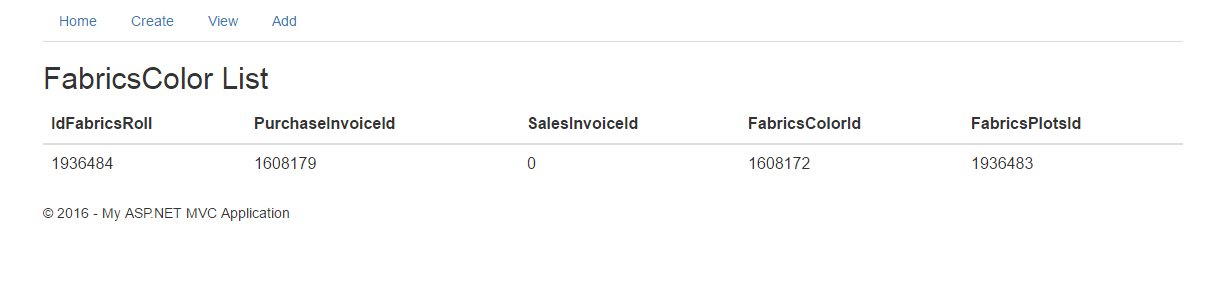
**

*Hình 28 : Danh sách lô hàng*

### Cây vải

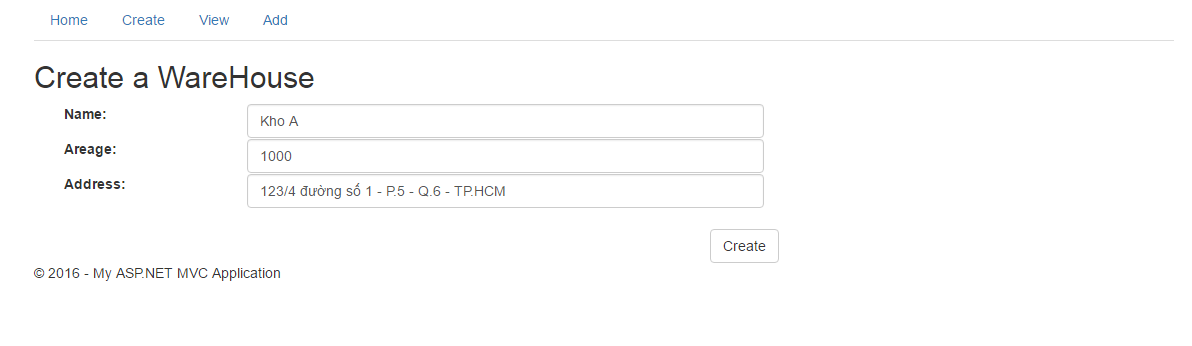
**

*Hình 29 : Tạo cây vải*

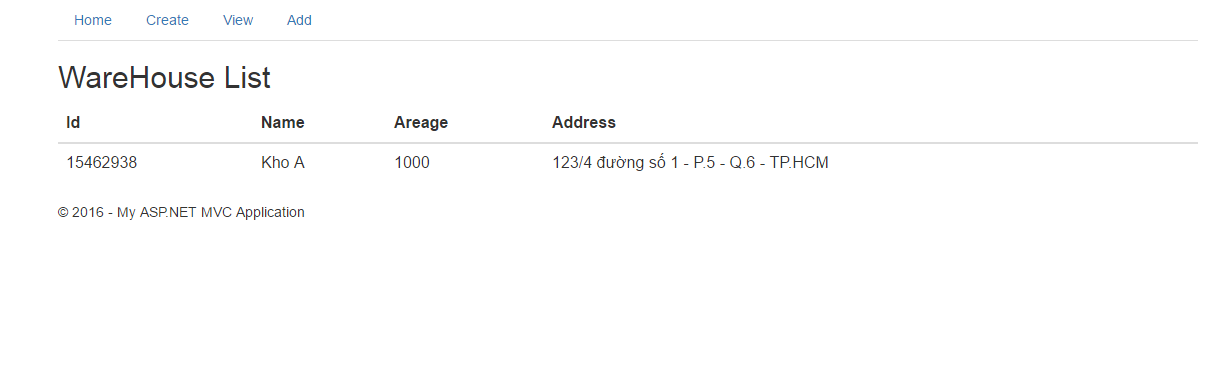
**

*Hình 30 : Danh sách cây vải*

### Kho



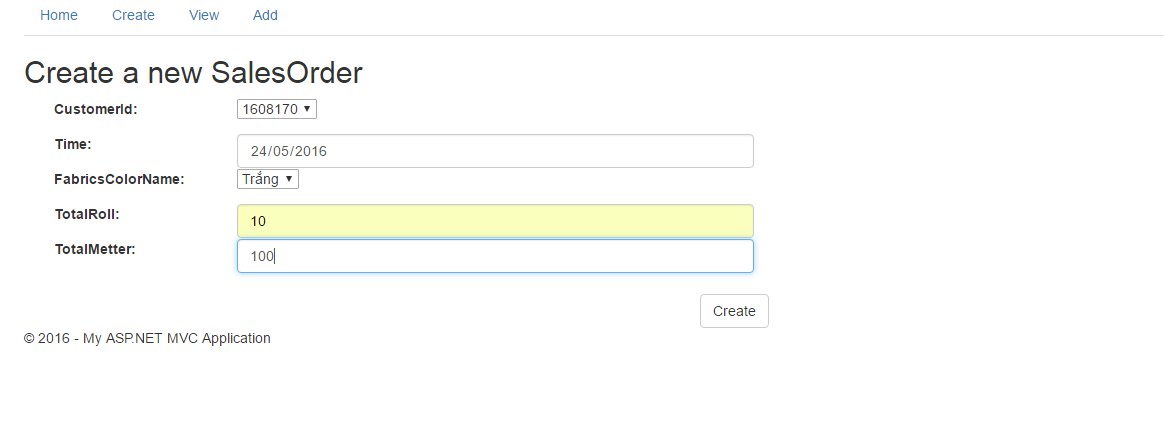
*Hình 31 : Tạo kho*

**

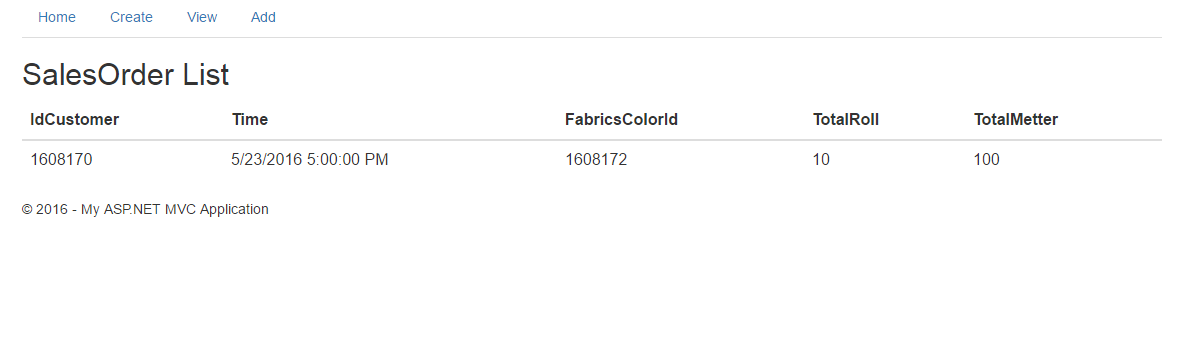
*Hình 32: Danh sách kho*

## Nghiệp vụ bán hàng

### 3.1. Đơn đặt hàng

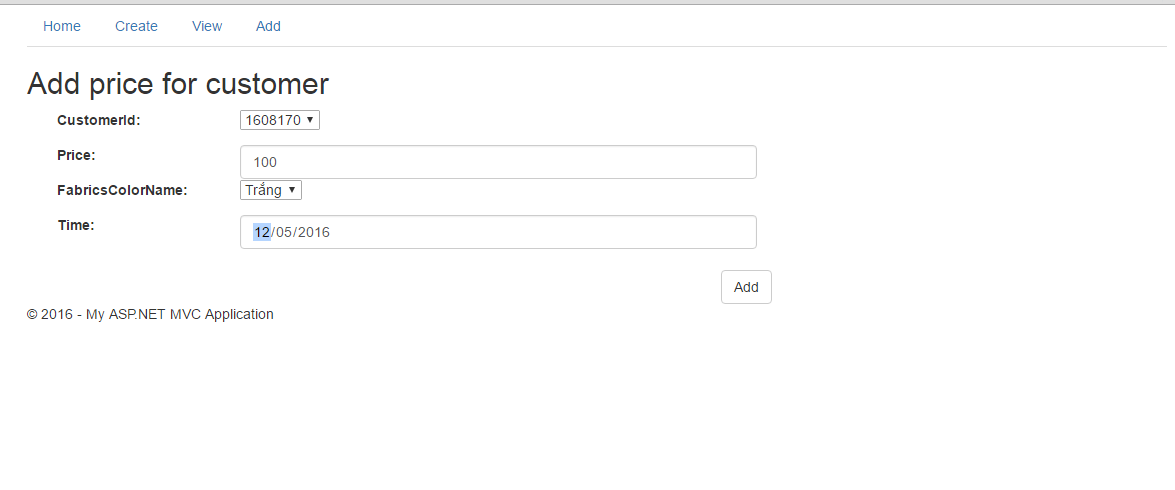


*Hình 33: Tạo đơn đặt hàng*

**

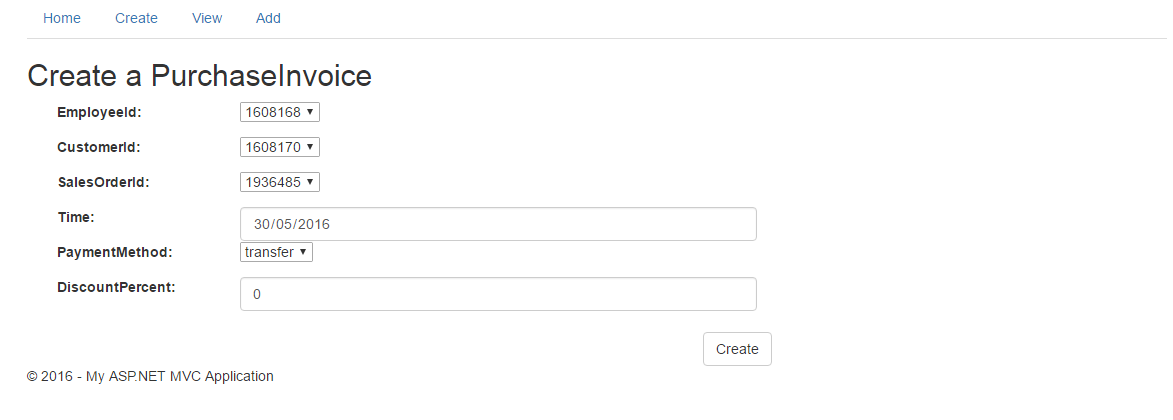
*HÌnh 34 : Danh sách đơn đặt hàng*

### 3.2. Thêm giá hàng hóa

**

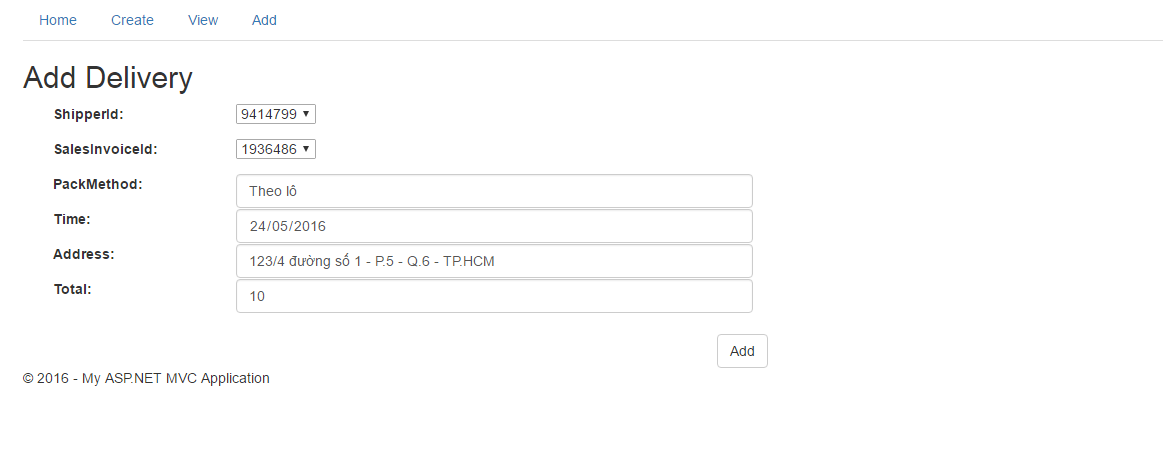
*Hình 35: Thêm giá cho khách hàng*

### 3.3. Hóa đơn bán hàng

**

*Hình 36 : Tạo hóa đơn bán hàng*

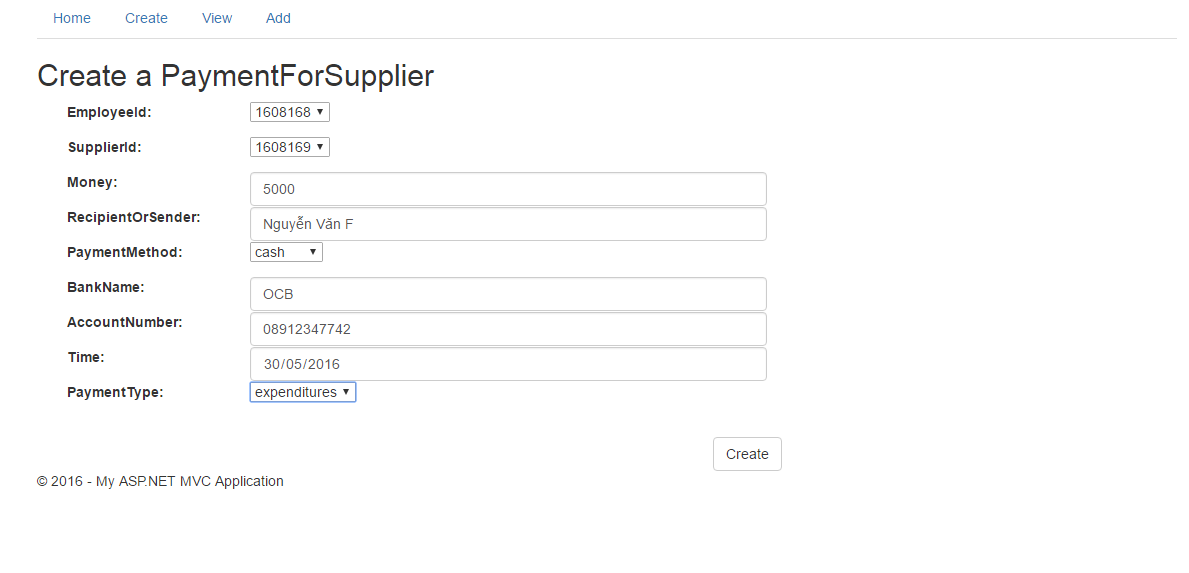
### 3.4. Thêm người vận chuyển cho hóa đơn

**

*Hình 37: Thêm người vận chuyển cho hóa đơn mua hàng*

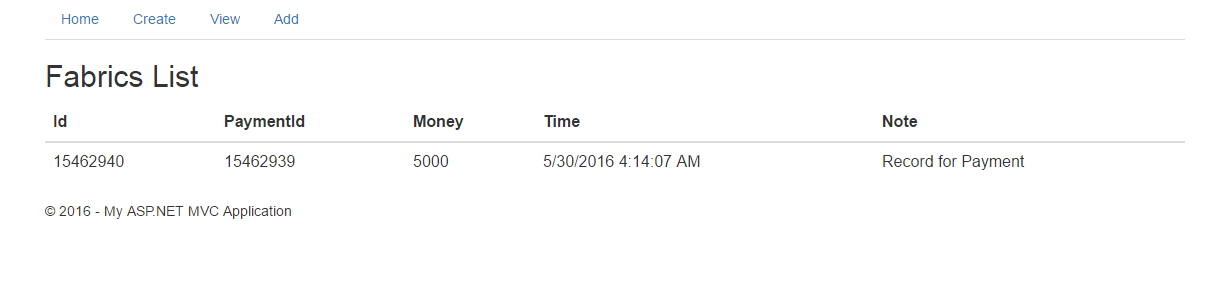
## Nghiệp vụ quản lý thu chi

### 4.1. Thanh toán hóa đơn mua hàng



*Hình 38: Thanh toán hóa đơn mua hàng*

### 4.2. Sổ thu chi



*Hình 39 : Danh sách sổ thu chi*

# TỔNG KẾT VÀ ĐÁNH GIÁ

Xây dựng cấu trúc framework hoàn chỉnh dễ mở rộng dựa trên mô hình factory pattern.

Tuy nhiên các dịch vụ cơ bản đáp ứng nhu cầu cho một công ty sản xuất vải là chưa hoàn chỉnh.

# TÀI LIỆU THAM KHẢO

[1] Tổng quan về MongoDB (http://freetuts.net/tong-quan-ve-mongodb-203.html)

[2] NoSQL (https://vi.wikipedia.org/wiki/NoSQL)

[3] Tổng quan về ASP.NET MVC (<https://support.microsoft.com/vi-vn/kb/2645095>)

[4] UML Class Diagram (<http://kipalog.com/posts/UML-Class-Diagram>)

[5] [Series bài dịch Design Patterns for Dummies](https://haihth.wordpress.com/2013/02/20/series-bai-dich-design-patterns-for-dummies/) (https://haihth.wordpress.com/2013/02/20/series-bai-dich-design-patterns-for-dummies/)

[6] C# (https://vi.wikipedia.org/wiki/C\_th%C4%83ng)