



**ĐẠI HỌC
HOA SEN**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC HOA SEN
KHOA KHOA HỌC VÀ CÔNG NGHỆ**

KHOÁ LUẬN TỐT NGHIỆP

BÁO CÁO CÁ NHÂN

Tên đề tài:

XÂY DỰNG KIẾN TRÚC CÔNG THÔNG TIN TÌM VIỆC

Giảng viên hướng dẫn	:	TS. Trần Vũ Bình	
Sinh viên thực hiện	:	Lê Trung Hiếu	(09015L)
Lớp	:	QL092L	

Tháng 12 năm 2011

LỜI CẢM ƠN

Trước tiên, tôi xin chân thành cảm ơn TS. Trần Vũ Bình đã tận tình hướng dẫn, góp ý và định hướng nhóm chúng tôi trong quá trình thực hiện khoá luận tốt nghiệp này. Khoá luận đã được hoàn thành cũng chính là nhờ sự nhắc nhở, đôn đốc và sự giúp đỡ nhiệt tình của thầy.

Xin chân thành cảm ơn các thầy cô Khoa Khoa Học & Công Nghệ đã nhiệt tình giúp đỡ tôi trong suốt thời gian học tập tại trường. Chính thầy cô đã xây dựng cho tôi kiến thức nền tảng và những kiến thức chuyên môn để nhóm hoàn thành khoá luận này.

Tôi cũng xin cảm ơn gia đình và bạn bè đã luôn bên cạnh, cổ vũ và động viên tôi những lúc khó khăn để có thể vượt qua và hoàn thành tốt khoá luận này.

Mong rằng các kiến thức cũng như các kinh nghiệm tôi nhận được trong thời gian làm khoá luận vừa qua sẽ giúp ích cho tôi nhiều hơn trong công việc sau này.

Trân trọng.

MỤC LỤC

LỜI CẢM ƠN	1
MỤC LỤC.....	2
1. Những đóng góp của cá nhân trong đề án.....	4
1.1. Quản trị dự án bằng Tortoise SVN.....	4
1.2. Xây dựng giải pháp cây quyết định cho kiến trúc.....	4
1.3. Mô hình giải pháp so khớp thông tin cho kiến trúc	8
1.4. Nghiên cứu tìm hiểu về Design Patterns	9
1.5. Microsoft Business Intelligence (SQL Server Denali).....	10
1.5.1. Linked Server	10
1.5.2. Microsoft Analysis Services	11
1.5.3. Khắc phục một số khuyết điểm của Microsoft.AnalysisServices.dll...	11
1.5.4. MDX	11
1.6. Nghiên cứu công nghệ	12
1.6.1. ASP.NET MVC 3	12
1.6.2. LINQ	13
1.6.3. ADO.NET Framework Entity	14
1.6.4. ADOMD.NET	14
1.7. Công việc khác	14
1.7.1. Bảo mật cho mật khẩu bằng Password Salt	14
1.7.2. Triển khai quản lý người dùng và phân quyền trong MVC.....	15
2. Đánh giá kết quả thực hiện và hướng phát triển	15
2.1. Đánh giá kết quả đạt được	15

2.2. Những hạn chế	16
2.3. Hướng phát triển.....	16
3. Những kinh nghiệm có được sau khi thực hiện khoá luận	18
3.1. Về chuyên môn.....	18
3.2. Về kỹ năng làm việc nhóm	18
3.3. Về phong cách làm việc.....	18

1. Những đóng góp của cá nhân trong đề án

1.1. Quản trị dự án bằng Tortoise SVN

TortoiseSVN – Hệ thống quản lý phiên bản, đây là phần mềm mã nguồn mở dùng để quản lý và kiểm tra các phiên bản mã nguồn khác nhau trong quá trình phát triển phần mềm. Tortoise Subversion cũng còn được gọi là svn. Svn là lệnh dùng để thực hiện các chức năng Subversion trong môi trường thi hành lệnh trên các máy vi tính. Subversion được thiết kế với mục đích thay thế phần mềm quản lý phiên bản Concurrent Versioning System (CVS) đã cũ và có nhiều nhược điểm.

Sau khi được biết Google Code có hỗ trợ tính năng này hoàn toàn miễn phí nên giúp chúng tôi quản lý chặt chẽ cấu hình các tài liệu, mã nguồn của nhóm, khôi phục phiên bản cũ khi có sự cố hay dễ dàng nhận ra các thay đổi giữa các phiên bản. SVN hỗ trợ chúng tôi làm việc nhóm hiệu quả hơn.

1.2. Xây dựng giải pháp cây quyết định cho kiến trúc

Tôi tìm hiểu và đánh giá phương pháp khai thác dữ liệu bằng cây quyết định, nguyên tắc hoạt động của một số thuật toán xây dựng cây quyết định thường dùng và các độ đo mà các thuật toán này sử dụng. Sau đó, nghiên cứu xây dựng cây quyết định bằng Microsoft Analysis Services để xây dựng Data Mining Engine cho kiến trúc. Việc tiến hành xây dựng cây quyết định của kiến trúc bao gồm các giai đoạn theo thứ tự như sau:

1.2.1. Tiền xử lý dữ liệu

Dữ liệu đầu vào, cụ thể là taxonomy sẽ được Decision Tree Engine tiền xử lý để cho kết quả tối ưu nhất.

- Hệ thống sẽ tạo ra các view tương ứng đối với từng loại đối tượng trong taxonomy. Lọc các dòng dữ liệu lưu trữ “nhu cầu” hay “sản phẩm đáp ứng nhu cầu”, các “nhu cầu” hay “sản phẩm đáp ứng nhu cầu” này được phân nhóm theo tên và có một tiền tố (prefix) tương ứng (ví dụ “nhu cầu” có tiền tố là “NC” trước mỗi tên view).

- Dữ liệu lưu trữ dưới dạng taxonomy phải thông qua kỹ thuật Pivot Transformation của SQL Server nhằm giúp cho việc khai thác dữ liệu bằng cây quyết định thực hiện dễ dàng hơn.

- Lọc các ký tự đặc biệt mà MS Analysis Services không thể xử lý bao gồm “.,;”:\|*|?“&%\$!+=()[]{}<>” để đảm bảo giữ nguyên tên thuộc tính cho “nhu cầu” và “sản phẩm đáp ứng nhu cầu”. Việc mã hoá các ký tự đặc biệt khắc phục được khuyến điểm của Microsoft Analysis Services cũng như của “Business Intelligence Development Studio (BIDS)” (trong BIDS các ký tự đặc biệt không được giữ lại, tên các thuộc tính có ký tự đặc biệt sẽ bị BIDS lược bỏ)

Đầu vào: taxonomy

Đầu ra: các view lưu trữ cho từng “nhu cầu” và từng “sản phẩm đáp ứng nhu cầu”, có tiền tố để phân biệt giữa view lưu trữ “nhu cầu” và view lưu trữ “sản phẩm đáp ứng nhu cầu”. Các view được xử lý bằng kỹ thuật Pivot Transformation và được mã hoá các ký tự đặc biệt

1.2.2. Tạo Data source

Tạo data source dựa vào thông số kết nối đến Database Engine lưu trữ taxonomy.

Đầu vào: Thông số kết nối đến Database Engine lưu trữ taxonomy

Đầu ra: Data source object

1.2.3. Tạo Data source view

Dựa vào Data Source, tự động dựa vào các tiền tố được tạo ra ở bước tiền xử lý dữ liệu, từ đó tạo ra các Data source view. Các view có tiền tố phù hợp sẽ được tạo view hỗ trợ tạo Mining Structure.

Đầu vào: Data source object, prefix được định nghĩa ở bước tiền xử lý dữ liệu

Đầu ra: Data source view object

1.2.4. Định nghĩa các tham số thuật toán

Lựa chọn thuật toán xây dựng cây quyết định và các tham số tương ứng.

Các tham số thuật toán bao gồm:

- HoldoutMaxPercent: số lượng dữ liệu dùng để kiểm tra (đơn vị tính: %)

- SCORE_METHOD: Tên thuật toán dùng để xây dựng cây quyết định (Entropy, Bayesian with K2 Prior, hay Bayesian Dirichlet Equivalent (BDE) Prior)
- COMPLEXITY_PENALTY: kiểm soát mức độ phát triển của cây
- SPLIT_METHOD: Phương thức phân nhánh (Binary, Complete, Both)
- MAXIMUM_INPUT_ATTRIBUTES: Số thuộc tính đầu vào tối đa
- MAXIMUM_OUTPUT_ATTRIBUTES: Số thuộc tính đầu ra tối đa
- MINIMUM_SUPPORT: Chỉ định số lượng tối thiểu các trường hợp phải có các tập phổ biến trước khi thuật toán tạo ra một quy tắc.

Đầu vào: KHÔNG

Đầu ra: các tham số tham gia vào quá trình xây dựng cây quyết định

1.2.5. Tạo Mining Structures và Mining Models

Các tham số thuật toán được tạo ở giai đoạn trên sẽ được sử dụng để tạo ra Mining Structure và Mining Model. Data source và Data source view cùng với prefix sẽ xác định các model cần tiến hành xây dựng cây quyết định, các view có trong Data source view có prefix tương ứng sẽ được tiến hành xây dựng cây ở bước tiếp theo nhằm tránh sự lãng phí tài nguyên hệ thống.

Đầu vào: Data source, Data source view, tham số thuật toán, prefix được định nghĩa bước tiền xử lý dữ liệu

Đầu ra: Mining Structures và Mining Models được tạo

1.2.6. Tiến hành khai thác dữ liệu

Sau khi các Mining Structures và Mining Models được tạo ra, dữ liệu sẽ được nạp và xử lý, quá trình xử lý có thể kéo dài hàng giờ, hàng tuần hay hàng tháng phụ thuộc vào số lượng dữ liệu. Việc tiến hành khai thác dữ liệu trên một server độc lập với server chính sẽ góp phần đảm bảo hiệu năng của hệ thống khi xây dựng cây quyết định.

Đầu vào: Mining Structures và Mining Models được tạo

Đầu ra: Mining Models đã được xử lý

1.2.7. Xuất dữ liệu đã khai thác vào Database Engine

Sau khi dữ liệu đã được phân tích xong sẽ được lưu trữ về Server chính để dễ dàng truy xuất. Các ký tự đặc biệt được mã hoá ở giai đoạn tiền xử lý dữ liệu sẽ được giải mã ở giai đoạn này.

Một “Linked Server” được tạo ra trên Database Engine nhằm kết nối, lấy dữ liệu từ Analysis Services thông qua ngôn ngữ truy vấn MDX và ngôn ngữ truy vấn SQL. “Linked Server” là kỹ thuật liên kết các Server lại với nhau, giúp truy vấn dữ liệu từ một database đặt tại server khác (hoặc một instance khác của SQL Server).

Trong quá trình xuất dữ liệu, các Mining Model sẽ được liên kết với nhau thông qua node gốc của model (NODE_NAME = “0”, NODE_TYPE = 1), các node gốc của cây (NODE_CAPTION = “All”, NODE_TYPE = 2) sẽ được chuyển thành tên của “nhu cầu” hay “sản phẩm đáp ứng nhu cầu” để dễ dàng truy vấn.

Đầu vào: Mining Models đã được xử lý

Đầu ra: DecisionTreeNode và DecisionTreeNodeDistribution chứa kết quả khai thác (các cây quyết định đã được liên kết với nhau)

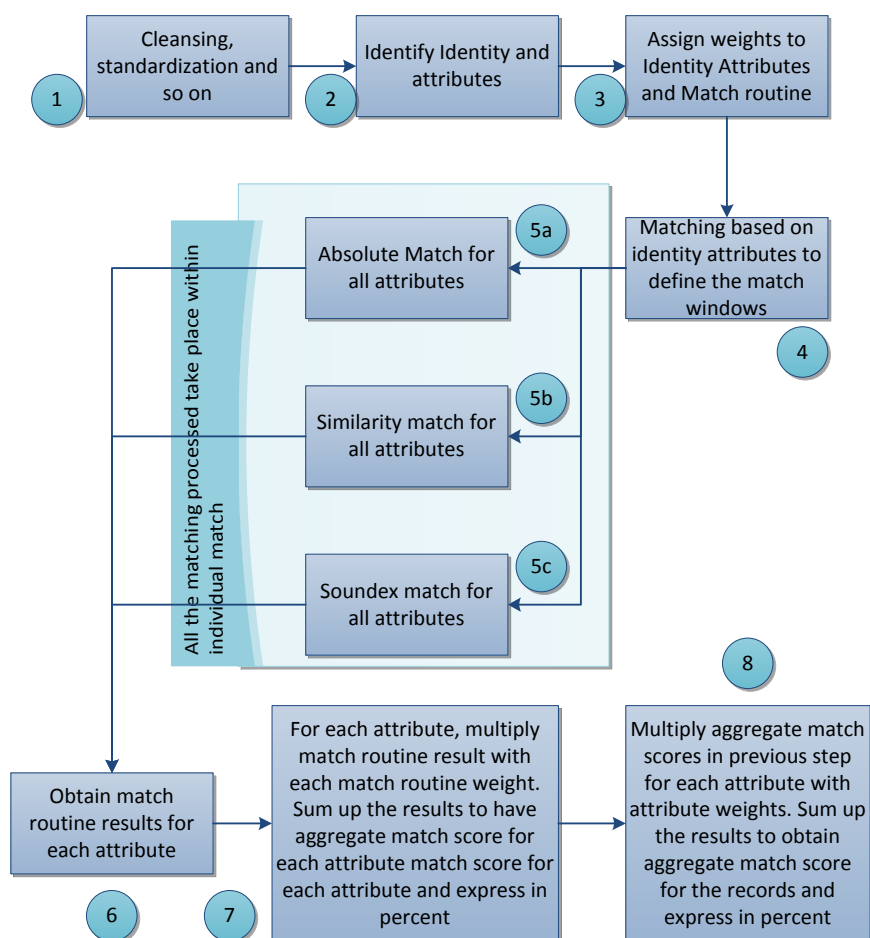
1.2.8. Dọn dẹp và đóng các kết nối

Sau khi cây quyết định được hoàn thành và lưu trữ trở về database chính, cần dọn dẹp các bảng dữ liệu tạm và đóng các kết nối sử dụng trong quá trình tạo cây.

Đầu vào: Kết quả của quá trình xây dựng cây quyết định.

Đầu ra: Dọn dẹp dữ liệu và đóng các kết nối. Giải pháp cho phần Data Mining Engine của tôi xây dựng hoàn toàn độc lập với công cụ “Business Intelligence Development Studio” – đây là môi trường làm việc với các mô hình khai thác dữ liệu mà Microsoft hỗ trợ.

1.3. Mô hình giải pháp so khớp thông tin cho kiến trúc



Hình 1. Mô hình so khớp thông tin

❖ **Bước 1:** Làm sạch và chuẩn hoá dữ liệu.

Lấy thông tin về nhu cầu và “sản phẩm đáp ứng nhu cầu”, chuẩn hoá và làm sạch thông tin chuẩn bị cho việc so khớp.

❖ **Bước 2:** Xác định các thuộc tính cần so khớp

Dựa vào những thuộc tính miêu tả nhu cầu để chọn ra những thuộc tính cần lấy ra từ “sản phẩm đáp ứng nhu cầu”.

❖ **Bước 3:** Xác định “Attribute Weight” (tầm quan trọng) cho từng thuộc tính và phương pháp so khớp cho mỗi kiểu dữ liệu trong khai thác thông tin. Ứng với từng loại kiểu dữ liệu sẽ có những phương pháp so khớp phù hợp, ứng với mỗi phương pháp sẽ có một trọng số “Match Routine Weight” tương ứng

❖ **Bước 4:** Dựa vào các thuộc tính cần so khớp định nghĩa các tiến trình so khớp

❖ **Bước 5:** Ứng với mỗi tiến trình thực hiện các phương pháp so khớp phù hợp cho mỗi kiểu dữ liệu. Có các phương pháp so khớp sau:

Absolute match: So khớp tuyệt đối, A phải hoàn toàn khớp với B. Kết quả so khớp giữa A và B là 0% hoặc 100%

Similarity match: So khớp có mức độ tương quan, A tương quan B và có một số điểm tương ứng. Kết quả so khớp giữa A và B trong khoảng từ 0% đến 100%

Soundex match: So khớp các từ có phát âm tương tự nhau. Kết quả so khớp giữa A và B trong khoảng từ 0% đến 100%

❖ **Bước 6:** Tập hợp kết quả so sánh cho từng thuộc tính

❖ **Bước 7:** Ứng với mỗi kết quả so khớp thuộc tính, kết quả so khớp của từng phương pháp so khớp sẽ được nhân với weight (tầm quan trọng) của mỗi phương pháp, sau đó tổng hợp để đưa ra số điểm cho mỗi thuộc tính. Kết quả được thể hiện dưới dạng phần trăm.

$$\text{Attribute Score (\%)} = \frac{\sum \text{Match Routine Score (\%)} * \text{Match Routine Weight}}{\sum \text{Match Routine Weight}}$$

❖ **Bước 8:** Nhân mỗi số điểm tổng hợp ở bước 7 với weight của từng thuộc tính được xác định ở bước 3. Tổng hợp để tính trung bình. Kết quả của bước 8 là mức độ so khớp giữa thông tin nhu cầu và “sản phẩm đáp ứng nhu cầu”

$$\text{Score (\%)} = \frac{\sum \text{Attribute Score (\%)} * \text{Attribute Weight}}{\sum \text{Attribute Weight}}$$

1.4. Nghiên cứu tìm hiểu về Design Patterns

Design Pattern là những thiết kế đã được sử dụng và được đánh giá tốt giúp giải quyết những vấn đề thiết kế thường gặp, Design pattern chú trọng việc giúp cho bản thiết kế có tính uyển chuyển, dễ nâng cấp, thay đổi.

Việc nghiên cứu tìm hiểu design patterns nhằm đóng góp ý kiến để đưa ra các giải pháp thiết kế kiến trúc phần mềm một cách hiệu quả, linh hoạt và khả năng tái sử dụng cao.

1.5. Microsoft Business Intelligence (SQL Server Denali)

Microsoft Business Intelligence được thiết kế xây dựng dựa trên nền tảng dữ liệu có khả năng mở rộng cho việc sắp xếp, phân tích, báo cáo dữ liệu và cung cấp các công cụ trực quan và mạnh mẽ để người dùng có thể sử dụng truy cập và phân tích các thông tin. Hạt nhân cuối cùng trong Microsoft BI đó là Microsoft SQL Server (hỗ trợ tốt nhất từ phiên bản 2008 trở lên). Microsoft Business Intelligence bao gồm các thành phần sau:

Thành phần	Mô tả
SQL Server Database Engine	Mang đến một phương tiện lưu trữ dữ liệu hiệu suất cao và có khả năng mở rộng cho các ấn bản dữ liệu rất lớn. Điều đó làm cho nó trở thành một lựa chọn lý tưởng cho việc hợp nhất dữ liệu doanh nghiệp từ toàn bộ doanh nghiệp vào một trung tâm dữ liệu để thuận tiện cho việc phân tích và báo cáo.
SQL Server Integration Services	Một nền tảng toàn diện cho việc trích rút, biến đổi và tải (ETL), các hành động này cho phép cư trú và đồng bộ kho lưu trữ dữ liệu của bạn với dữ liệu từ các nguồn khác nhau được sử dụng bởi các ứng dụng doanh nghiệp trong toàn bộ tổ chức.
SQL Server Analysis Services	Cung cấp phương tiện phân tích cho các giải pháp phân tích xử lý trực tuyến (OLAP), cụ thể gồm có việc thu nạp các tham số doanh nghiệp qua nhiều kích thước và các chỉ thị hiệu suất chính (KPI), và cho các giải pháp khai thác dữ liệu có sử dụng các thuật toán đặc biệt để nhận dạng mẫu, xu hướng và quan hệ trong dữ liệu doanh nghiệp.
SQL Server Reporting Services	Một giải pháp báo cáo mở cho phép tạo, xuất bản và phân phối các báo cáo doanh nghiệp chi tiết một cách dễ dàng cả bên trong và bên ngoài tổ chức.

Bảng 1. Các thành phần của Microsoft Business Intelligence

1.5.1. Linked Server

Linked Server là một server ảo được dùng để truy cập vào các database system khác (ví dụ liên kết với một máy chủ chứa database khác, liên kết với access, analysis services...). Một khi đã cài đặt, chúng ta có thể truy vấn dữ liệu dùng four-part name:

Xây dựng kiến trúc cổng thông tin tìm việc| Những đóng góp của cá nhân trong đề án

linked_server_name.catalog.schema.object_name. Trong đó catalog thường tương đương với database name, Schema tương đương với database owner và object_name tương đương với table hay view.

Tôi dùng Linked Server trong việc liên kết giữa hai Database Engine với nhau để trao đổi dữ liệu giữa hai máy chủ, liên kết giữa Database Engine và Analysis Services để có thể lấy dữ liệu từ Analysis Services thông qua ngôn ngữ MDX trên Database Engine.

1.5.2. Microsoft Analysis Services

Microsoft Analysis Services là một dịch vụ kèm theo trong bộ Microsoft SQL Server từ phiên bản 2005 trở đi, đây là công cụ để sử dụng, tích hợp và linh động giúp bạn xây dựng các khối dữ liệu đa chiều và cung cấp các chương trình ứng dụng truy xuất tới các khối dữ liệu. Các khối có thể được lưu trữ trong cơ sở dữ liệu quan hệ (ROLAP), hoặc lưu trữ theo dạng cấu trúc đa chiều riêng biệt, ở mức cao (MOLAP) hoặc kết hợp trộn lẫn của cả hai cấu trúc trên (HOLAP)

Microsoft Analysis Services cung cấp các giải thuật cho việc xây dựng các giải pháp khai thác dữ liệu.

1.5.3. Khắc phục một số khuyết điểm của Microsoft.AnalysisServices.dll

Microsoft Analysis Services có khuyết điểm trong việc xử lý các ký tự đặc biệt trong TagName. Tôi đưa ra giải pháp lọc các ký tự đặc biệt mà MS Analysis Services không thể xử lý bao gồm “.,”:’\^*|’?”&%\$!+=()[]{}<>” để đảm bảo giữ nguyên tên thuộc tính cho đối tượng. Việc mã hoá các ký tự đặc biệt khắc phục được khuyết điểm của Microsoft Analysis Services cũng như của “Business Intelligence Development Studio (BIDS)” (trong BIDS các ký tự đặc biệt không được giữ lại, tên các thuộc tính có ký tự đặc biệt sẽ bị BIDS lược bỏ)

1.5.4. MDX

Ngôn ngữ truy vấn MDX (MultiDimensional eXpressions) là dạng mở rộng của ngôn ngữ truy vấn SQL, áp dụng trong khai thác kho dữ liệu OLAP với khối dữ liệu đa chiều nhằm hỗ trợ cho việc tổng hợp thông tin và ra quyết định. Ngôn ngữ MDX

hỗ trợ mạnh cho tính toán trên khối dữ liệu đa chiều, nó hỗ trợ việc truy cập dữ liệu một cách trực quan và dễ dàng.

MDX được xây dựng cho các hệ thống phân tích dữ liệu chính vì thế nó có thể xử lý được những truy vấn một cách dễ dàng. MDX có thể xử lý dữ liệu một cách mềm dẻo, kết quả trả về của MDX cũng rất linh hoạt. Các phép truy vấn phức tạp trong SQL như pivot được sử dụng một cách dễ dàng trong MDX.

Cấu trúc của MDX giống như SQL nhưng mở rộng hơn để thao tác với cơ sở dữ liệu nhiều chiều. MDX phân biệt các tập hợp bằng việc dùng dấu ngoặc nhọn { } bao quanh giữa các bộ hay thành viên. Câu truy vấn MDX có thể chứa tới 128 chiều, nhưng chỉ có 5 chiều đầu tiên là có alias (tên phụ). Một trục có thể được tham chiếu bởi thứ tự của nó trong câu MDX hay bởi tên phụ của nó nếu có. Nếu đã từng sử dụng câu truy vấn SQL, bạn có thể nhận thấy mệnh đề FROM có thể chỉ đến nhiều bảng dữ liệu. Tuy nhiên, mệnh đề FROM trong MDX bị giới hạn ở một khối duy nhất.

Tôi sử dụng MDX kết hợp Linked Server nhằm chuyển dữ liệu đã phân tích trong Analysis Services về hệ thống cơ sở dữ liệu chính, hỗ trợ truy vấn các dữ liệu đã khai thác một cách nhanh chóng và hiệu quả hơn

1.6. Nghiên cứu công nghệ

1.6.1. ASP.NET MVC 3

ASP.NET MVC Framework cho phép người làm phần mềm xây dựng những Web application theo ý tưởng của MVC (Model-View-Controller). MVC là tên một phương pháp chia nhỏ một ứng dụng thành ba thành phần để cài đặt, mỗi thành phần đóng một vai trò khác nhau và ảnh hưởng lẫn nhau, đó là models, views, và controllers.

Models: Models trong các ứng dụng dựa trên MVC là những thành phần có nhiệm vụ lưu trữ thông tin, trạng thái của các đối tượng, thông thường nó là một lớp được ánh xạ từ một bảng trong cơ sở dữ liệu. Models thiết lập logic của phần dữ liệu của ứng dụng.

Views: chịu trách nhiệm hiển thị các thông tin lên cho người dùng thông qua giao diện (UI). Thông thường, các thông tin cần hiển thị được lấy từ thành phần Models.

Controllers: Controller là các thành phần dùng để quản lý tương tác người dùng, chịu trách nhiệm xử lý các tác động về mặt giao diện, các thao tác đối với models, và cuối cùng là chọn một view thích hợp để hiển thị ra màn hình. Trong ứng dụng MVC, Views 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.

MVC giúp tạo được các ứng dụng phân biệt rõ ràng các khía cạnh của ứng dụng (logic về nhập liệu, logic xử lý tác vụ và logic về giao diện). MVC chỉ ra mỗi loại logic kể trên nên được thiết lập ở đâu trên ứng dụng. Logic giao diện (UI logic) thuộc về views. Logic nhập liệu (input logic) thuộc về controller. Và logic tác vụ (Business logic – là logic xử lý thông tin, mục đích chính của ứng dụng) thuộc về model. Sự phân chia này giúp bạn giảm bớt được sự phức tạp của ứng dụng và chỉ tập trung vào mỗi khía cạnh cần được cài đặt ở mỗi thời điểm.

❖ Điểm mạnh

- Dễ dàng quản lý sự phức tạp của ứng dụng bằng cách chia ứng dụng thành ba thành phần model, view, controller. 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 ứng dụng hướng kiểm thử (TDD - test driven development)
- Nó 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
- Không sử dụng view state hoặc server-based form giúp lập trình viên dễ dàng quản lý hết các khía cạnh của một ứng dụng.
- Thực hiện unit test dễ dàng
- Friendly-URL (URL thân thiện, ngắn gọn).

1.6.2. LINQ

LINQ, viết tắt của Language Integrated Query, là một thành phần trong .NET Framework (từ 3.0 trở lên), đây là thư viện cho phép truy vấn trên nhiều kiểu dữ liệu khác nhau: Object collections, XML, LINQ to SQL (truy vấn cơ sở dữ liệu MS-SQL), LINQ to Dataset, LINQ to Entity (ADO.net Entity Framework), LINQ to SharePoint, LINQ to XML, LINQ to nHibernate, LINQ to XYZ, LINQ to Azure SQL Service....

❖ Điểm mạnh

- Nhanh hơn khi truy vấn dữ liệu nếu so sánh với ADO.NET.
- Giảm bớt số câu lệnh nặng tính thủ tục (procedural code) để thêm xoá sửa dữ liệu, giảm thiểu được nhiều dòng code

❖ **Điểm yếu**

- Về tổng quan, LINQ chậm hơn rất nhiều khi so sánh với T-SQL (Stored Procedure and compiled SQL query có tốc độ thực thi nhanh nhất vì nó gần với dữ liệu trên cơ sở dữ liệu nhất)

1.6.3. ADO.NET Framework Entity

ADO.NET Data Entity Framework được Microsoft phát triển như là một sự thay thế cho Linq To Sql

❖ **Điểm mạnh:**

- Ánh xạ mềm dẻo cấu trúc bảng trong cơ sở dữ liệu vào cấu trúc quan hệ, hướng đối tượng của các Object Entities trong .NET Business Layer
- Khả năng thực thi các câu lệnh dynamic SQL và stored procedure để tận dụng những tính năng lập trình đặc biệt của CSDL
- Giảm bớt số câu lệnh nặng tính thủ tục (procedural code) để thêm xoá sửa dữ liệu.
- Thể hiện những mối quan hệ phức tạp như: thừa kế, ...

1.6.4. ADOMD.NET

ADOMD.NET (ActiveX Data Objects Multi-Dimensional Library) cho phép bạn truy cập Online Analytical Processing (OLAP) Server Data từ ứng dụng của mình. ADOMD.NET cung cấp thư viện kết nối và thực hiện truy vấn trên Microsoft Analysis Services

Điểm yếu: Không debug câu lệnh truy vấn được

1.7. Công việc khác

1.7.1. Bảo mật cho mật khẩu bằng Password Salt

Nhằm mục đích mã hóa mật khẩu của người sử dụng trong cơ sở dữ liệu, tôi đã xây dựng một phương mã hóa bằng “Password Salt”, nhằm mục đích mã hóa và giả

mã password của người dùng sau đó lưu vào database, nhằm gia tăng độ tin cậy và bảo mật cho người dùng.

1.7.2. Triển khai quản lý người dùng và phân quyền trong MVC

Nghiên cứu triển khai cách tùy chỉnh cơ chế chứng thực người dùng và phân quyền theo roles trong MVC.

2. Đánh giá kết quả thực hiện và hướng phát triển

2.1. Đánh giá kết quả đạt được

- Kiến trúc giải quyết được nhu cầu lưu trữ thông tin nhiều, đa dạng, dễ dàng trong việc tìm kiếm và so khớp
- Tổ chức thông tin linh động dưới dạng tag kết hợp taxonomy. Thông tin được lưu trữ dưới dạng cây đa cấp và có mức độ tương quan giữa các tag.
- Hệ thống so khớp linh hoạt, kết hợp mức độ tương quan giữa các tag giải quyết được khuyết điểm của việc so sánh object.
- Kiến trúc dễ dàng triển khai, mở rộng, đồng thời hiệu năng của hệ thống vẫn được đảm bảo khi áp dụng cây quyết định và hệ thống so khớp để gợi ý cho actor xây dựng cây thông tin nhu cầu và “sản phẩm đáp ứng nhu cầu” một cách nhanh chóng và hiệu quả.
- Khắc phục hạn chế của “Microsoft Analysis Services¹”: không hỗ trợ xử lý ký tự đặc biệt. Hoàn toàn không phụ thuộc vào công cụ “SQL Server Business Intelligence Development Studio²”
- Ứng dụng kiến trúc vào bài toán JobZoom nhằm đánh giá khả năng tổ chức thông tin linh hoạt, hiệu quả của hệ thống so khớp, đồng thời đánh giá tính mở rộng, dễ dàng triển khai và tái sử dụng của kiến trúc khi áp dụng trong các lĩnh vực cần lưu trữ và so khớp thông tin.
- Nhóm đã xây dựng và phát triển JobZoom framework ứng dụng những điểm mạnh của bài toán kiến trúc:

¹ Microsoft Analysis Services: Thành phần của Microsoft SQL Server, dùng để khai thác dữ liệu

² SQL Server Business Intelligence Development Studio: Công cụ hỗ trợ khai thác dữ liệu của Microsoft SQL Server

- Người tìm việc tìm kiếm công việc phù hợp nhanh chóng, viết CV dễ dàng hơn dựa vào gợi ý từ kết quả khai thác dữ liệu và hệ thống so khớp
- Nhà tuyển dụng dễ dàng đăng tuyển yêu cầu công việc một cách chi tiết và lựa chọn ứng viên phù hợp với vị trí làm việc thông qua hệ thống so khớp và cây quyết định. Hệ thống so khớp giúp đánh giá ứng viên cho kết quả cụ thể (yêu cầu nào đạt hay không đạt và số điểm ứng viên đạt được)
- JobZoom Framework dễ dàng triển khai, mở rộng, kết hợp được với phương pháp khai thác dữ liệu “cây quyết định” tạo nền tảng cải thiện khả năng thích ứng của hệ thống với từng loại ngành nghề.

2.2. Những hạn chế

- Cây đa cấp còn ở mức trừu tượng thông qua bảng SimilarityTerm để thể hiện mức độ tương quan giữa các tag, việc xác định độ tương quan giữa các tag sẽ được triển khai dễ dàng dựa vào tính dễ dàng mở rộng của kiến trúc.
- Chưa được áp dụng vào thực tế nên bài toán kiến trúc vẫn còn ở mức trừu tượng, cần điều chỉnh để đảm bảo tính khách quan và linh hoạt của hệ thống so khớp. Dữ liệu kiểm tra vẫn chưa nhiều chưa chứng minh được hiệu quả của kiến trúc khi áp dụng vào hệ thống nhiều ngành nghề.
- Chưa triển khai được kiến trúc JobZoom framework trong thực tế nên khó kiểm tra tính khả thi và tốc độ tính toán, so khớp và xây dựng cây quyết định do nhóm phát triển khi triển khai với số lượng người dùng lớn. Tuy nhiên, hiệu năng của hệ thống được cải thiện thông qua việc xây dựng cây quyết định trên một hệ thống độc lập sau đó lưu trữ kết quả về database chính để truy vấn nhanh chóng, duyệt thông tin được lưu trữ dưới dạng cây nhanh hơn lưu trữ dưới dạng văn bản.

2.3. Hướng phát triển

2.3.1. Xác thực độ tin cậy của CV và yêu cầu tuyển dụng

Các đối tượng tham gia vào những cộng đồng trực tuyến thường là các đối tượng “ảo” và thường không được xác định tính chính xác, tính xác thực của các thông tin đăng tải. Việc xác thực độ tin cậy của CV và yêu cầu tuyển dụng sẽ giúp hệ thống vận hành tốt hơn, tránh rủi ro và tạo uy tín với người sử dụng. Với kiến trúc linh

hoạt của hệ thống ta có thể dễ dàng tích hợp công cụ xác định độ tin cậy của người dùng từ đó xác thực thông tin của họ.

2.3.2. Phát triển Semantic web

Với hệ thống hiện tại của chúng tôi chỉ mô hình lại kết quả phân tích về tương quan ngữ nghĩa (Semantic similarity) giữa các tag và tích hợp vào hệ thống bằng một interface cơ bản. Việc xây dựng và triển khai hệ thống Semantic web sẽ hỗ trợ người dùng tìm kiếm thông tin nhanh chóng, chuẩn xác, thông minh và trả ra kết quả so khớp chính xác hơn.

2.3.3. Đánh trọng số cho các thuộc tính dựa vào độ sâu của taxonomy

Phát triển cây đa cấp dựa vào semantic web, những tag có liên quan với những tag khác, khi matching sẽ được một số điểm tương ứng, số điểm này được dựa vào độ sâu của tag.

2.3.4. Phân tán dữ liệu

Ứng dụng những điểm mạnh của cơ sở dữ liệu phân tán nhằm tăng hiệu năng cho hệ thống.

2.3.5. Công cụ hỗ trợ lựa chọn ứng viên

Công cụ hỗ trợ lựa chọn ứng viên: phỏng vấn trực tuyến, thuyết trình qua internet... có thể được phát triển để hỗ trợ nhà tuyển dụng và ứng viên.

2.3.6. Thu thập thông tin việc làm tự động

Giúp thu thập thông tin việc làm từ các cổng thông tin tìm việc hiện tại.

2.3.7. Áp dụng quy trình tuyển dụng vào hệ thống

Kiến trúc đưa ra các ứng viên có số điểm cao cho nhà tuyển dụng quyết định, đây chỉ là một khâu trong quy trình tuyển dụng ứng viên. Hệ thống nên có thêm quy trình tuyển dụng, quản lý quy trình từ lúc ứng viên nộp hồ sơ đến khi ứng viên trở thành nhân viên của công ty, giúp nhà tuyển dụng dễ dàng trong việc quản lý quy trình tuyển dụng của mình.

3. Những kinh nghiệm có được sau khi thực hiện khoá luận

3.1. Về chuyên môn

- Nâng cao khả năng đọc hiểu các tài liệu tiếng Anh chuyên ngành
- Ứng dụng được kiến thức đã học trong môn Quản lý dự án phần mềm cùng với các kiến thức, kinh nghiệm có được trong quá trình làm việc để xác định sở trường và khả năng của các thành viên trong nhóm, từ đó phân chia công việc hợp lý.
- Nghiên cứu mới phải căn cứ vào các tài liệu, chứng cứ khoa học cùng những dẫn chứng cụ thể để có thể thuyết phục người đọc, đồng thời phải biết xác định một số hạn chế quan trọng trong các nghiên cứu trước để khắc phục, tránh cách làm dàn trải mà không đi sâu vào một vấn đề nào.
- Có thêm kinh nghiệm khi xây dựng kiến trúc phần mềm nghiệp vụ, áp dụng các design pattern nhằm thiết kế kiến trúc framework đảm bảo tính linh động, dễ mở rộng, hiệu năng hệ thống.

3.2. Về kỹ năng làm việc nhóm

- Phân chia công việc trong nhóm một cách hiệu quả, phù hợp với khả năng của từng cá nhân: đúng khả năng, sở trường, sở thích của mỗi người để đạt hiệu quả công việc tốt nhất.
- Tôn trọng ý kiến của các thành viên trong nhóm.
- Nâng cao khả năng và hiệu quả khi làm việc nhóm, khả năng đặt vấn đề và giải quyết vấn đề.
- Giải quyết mâu thuẫn giữa các thành viên trong nhóm.
- Hỗ trợ các thành viên trong nhóm.

3.3. Về phong cách làm việc

- Chủ động, nghiên cứu tìm hiểu thêm các kiến thức, công nghệ mới.
- Hoà đồng, nhiệt tình, tôn trọng các thành viên trong nhóm.
- Học hỏi phương pháp nghiên cứu khoa học cùng với phong cách làm việc khoa học của giảng viên hướng dẫn.
- Vận dụng phương pháp tư duy phản biện để đánh giá vấn đề.