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

**BÁO CÁO CÁ NHÂN**

**XÂY DỰNG KIẾN TRÚC**

**CỔNG THÔNG TIN TÌM VIỆC**

Giảng viên hướng dẫn : TS. Trần Vũ Bình

Sinh viên thực hiện : Lê Dương Công Phúc (09025L)

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 khóa luận vừa qua sẽ giúp ích cho tôi nhiều hơn trong công việc sau này.

Trân trọng.

# **MỤC LỤC**

[LỜI CẢM ƠN 1](#_Toc312626653)

[MỤC LỤC 2](#_Toc312626654)

[1. Những đóng góp của cá nhân trong đề án 4](#_Toc312626655)

[1.1. Tìm hiểu, phân tích, đánh giá các website tìm việc hiện nay 4](#_Toc312626656)

[1.2. Tìm hiểu các cơ sở lý thuyết về xây dựng kiến trúc phần mềm 5](#_Toc312626657)

[1.2.1. Lý thuyết cơ bản về kiến trúc phần mềm 5](#_Toc312626658)

[1.2.2. 4+1 View architecture model 6](#_Toc312626659)

[1.2.3. Design pattern 6](#_Toc312626660)

[1.3. Tìm hiểu và áp dụng cây quyết định vào thiết kế hệ thống 7](#_Toc312626661)

[1.3.1. Tiền xử lý dữ liệu 7](#_Toc312626662)

[1.3.2. Xuất dữ liệu đã khai thác vào Database Engine 7](#_Toc312626663)

[1.4. Giải pháp tổ chức thông tin linh hoạt đảm bảo khả năng so khớp 7](#_Toc312626664)

[1.5. Giải pháp thiết kế kiến trúc tổng quan của kiến trúc hệ thống 7](#_Toc312626665)

[1.5.1. Thiết kế giải pháp kiến trúc tổng quan của framework 7](#_Toc312626666)

[1.5.2. Thiết kế giải pháp giải quyết vấn đề hiệu năng của hệ thống 7](#_Toc312626667)

[1.5.3. Thiết kế và triển khai thư viện lập trình (framework library) 7](#_Toc312626668)

[1.6. Thử nghiệm, đánh giá giải pháp framework 7](#_Toc312626669)

[1.7. Nghiên cứu công nghệ 8](#_Toc312626670)

[1.7.1. ASP.NET MVC 3 8](#_Toc312626671)

[1.7.2. ADO.NET Framework Entity 8](#_Toc312626672)

[1.7.3. Microsoft SQL Server 2008 R2 Business Intelligene 8](#_Toc312626673)

[1.8. Công việc khác 8](#_Toc312626674)

[2. Đánh giá kết quả thực hiện và hướng phát triển 8](#_Toc312626675)

[2.1. Đánh giá kết quả đạt được 8](#_Toc312626676)

[2.2. Những hạn chế 8](#_Toc312626677)

[2.3. Hướng phát triển 8](#_Toc312626678)

[2.3.1. Xác thực độ tin cậy của CV và yêu cầu tuyển dụng 8](#_Toc312626679)

[2.3.2. Phát triển Semantic web 8](#_Toc312626680)

[2.3.3. Đánh trọng số cho các thuộc tính dựa vào độ sâu của taxonomy 8](#_Toc312626681)

[2.3.4. Phân tán dữ liệu 8](#_Toc312626682)

[2.3.5. Công cụ hỗ trợ lựa chọn ứng viên 8](#_Toc312626683)

[2.3.6. Thu thập thông tin việc làm tự động 8](#_Toc312626684)

[2.3.7. Áp dụng quy trình tuyển dụng vào hệ thống 8](#_Toc312626685)

[3. Những kinh nghiệm có được sau khi thực hiện khoá luận 8](#_Toc312626686)

[3.1. Về chuyên môn 8](#_Toc312626687)

[3.2. Về kỹ năng làm việc nhóm 8](#_Toc312626688)

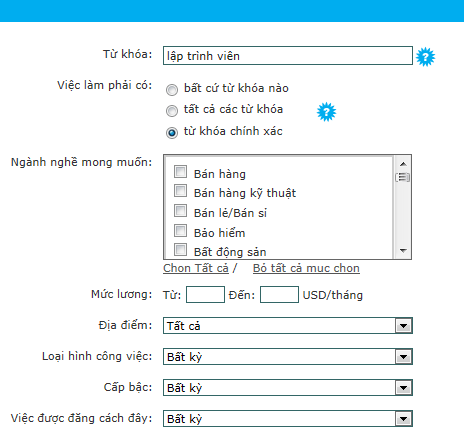
[3.3. Về phong cách làm việc 8](#_Toc312626689)

# Những đóng góp của cá nhân trong đề án

## Tìm hiểu, phân tích, đánh giá các website tìm việc hiện nay

Tìm hiểu các website tìm kiếm việc nổi tiếng trong và ngoài nước bao gồm: VietnamWorks.com, CareerLink, Anphabe, SimplyHired, Linkedin, MediaBistro, Monster, Craigslist, Yahoo! Hotjobs, CareerBuilder, Dice.. để tìm hiểu nghiệp vụ về đăng tải thông tin tuyển dụng, viết hồ sơ, tìm kiếm về cả hai gốc độ của ứng viên có nhu cầu kiếm việc và nhà tuyển dụng. Kết hợp tìm hiểu nghiệp vụ với tìm hiểu về các chức năng của hệ thống đó phục vụ cho việc đánh giá, phân tích đặc điểm, xu hướng các website tìm việc hiện nay. Qua tìm hiểu, đánh giá theo các tiêu chí cụ thể (theo đặc điểm, theo các chức năng nghiệp vụ cụ thể) cộng với cơ sở đánh giá của các tổ chức đánh giá website, tôi và nhóm đã phân tích và rút ra kết luận sau:

* Thông tin phục vụ cho nghiệp vụ đăng tải và tìm kiếm việc của các website rất đa dạng tùy theo đặc điểm riêng trong chiến lược phát triển của từng website (mạng xã hội việc làm (social networking), máy tìm kiếm việc (job search engine) , phân loại việc làm (classified), job for media types,..) từ đó đặt vấn đề: làm thế nào thiết kế một kiến trúc phần chung giúp tổ chức thông tin đa dạng một cách linh hoạt có thể áp dụng cho nhiều phần mềm có nghiệp vụ đăng tải và tìm kiếm thông tin nói chung và các website tìm kiếm việc làm nói riêng.
* Nghiệp vụ tổ chức thông tin phục vụ cho nghiệp vụ tìm kiếm thông tin còn hạn chế. Trên các website tìm kiếm việc, người sử dụng phải sử dụng các công cụ tìm kiếm còn nhiều hạn chế, phải thủ công lựa chọn các thuộc tính, tiêu chí tìm kiếm. Từ những phân tích, nhóm chúng tôi suy nghĩ đến vấn đề: làm thế nào tổ chức thông tin linh hoạt để hỗ trợ cho khả năng tìm kiếm, so khớp thông tin một cách từ động, giúp các ứng viên dễ dàng được gợi ý các công việc phù hợp với hồ sơ của mình và giúp cho các nhà tuyển dụng dễ dàng tìm ra những ứng viên phù hợp với đặc điểm của công việc.



Hình 1. Ví dụ về chức năng tìm kiếm công việc còn thủ công, ít linh hoạt

## Tìm hiểu các cơ sở lý thuyết về xây dựng kiến trúc phần mềm

### Lý thuyết cơ bản về kiến trúc phần mềm

Bám sát với mục tiêu và các bài toán cụ thể của đề tài chú trọng về “xây dựng kiến trúc phần mềm” áp dụng vào các phần mềm có nghiệp vụ đăng tải, tìm kiếm, so khớp thông tin nói chung và các website tìm kiếm việc làm nói riêng.

***Tôi bắt đầu tìm hiểu lý thuyết về kiến trúc phần mềm để nắm rõ những khái niệm cơ bản, có cái nhìn tổng quan về bài toán của đề tài về mặt kiến trúc, qua giúp đỡ các thành viên trong nhóm giải quyết các bài toán đúng hướng, đúng mục tiêu đã đặt ra.***

Tóm lược về kiến trúc phần mềm: bao gồm các phần tử phần mềm, các thuộc tính và mối quan hệ giữa chúng. Kiến trúc phần mềm giúp việc quyết định ở mức cao trong thiết kế phần mềm dễ dàng hơn và cho phép tái sử dụng các thành phần và mẫu thiết kế (design pattern) của các dự án.

**Software Architecture = {Components, Connectors, Constraint, Analysis}**

* Components: là các thành phần có đặc điểm riêng trong hệ thống.
* Connectors: là phương tiện trung gian cho các tương tác của các thành phần.
* Contraints: là các ràng buộc cho các thành phần (component) và các kết nối (connector).
* Framework model: cung cấp mô hình giải quyết các bài toán kiến trúc phần mềm cụ thể (domain-specific).

***Sau khi nắm rõ các lý thuyết cơ sở kiến trúc phần mềm, tôi đã đánh giá, phân tích các bài toán trong đề tài và đưa ra giải pháp tổng quan về kiến trúc chung hệ thống “Tổ chức thông tin linh hoạt đảm bảo khả năng so khớp, và đưa ra quyết định” nói chung và giải pháp JobZoom framework nói riêng có thể ứng dụng thực tế vào các website tìm kiếm việc làm.***

### 4+1 architecture view model

Trong lý thuyết về kiến trúc phần mềm đã chỉ ra về việc áp dụng mô hình 4+1 architecture view model để thiết kế giải pháp kiến trúc phần mềm. ***Tôi đã tìm hiểu thật kỹ về mô hình thiết kế này, áp dụng những kiến thức phân tích thiết kế cơ bản đã học vào mô hình này để giải quyết các bài toán cụ thể, qua các gốc độ kiến trúc khác nhau: người dùng cuối (end-user), lập trình viên (developer), tích hợp hệ thống (integrator) và người triển khai hệ thống (system engineer).***



Hình 2. Ví dụ về chức năng tìm kiếm công việc còn thủ công, ít linh hoạt

### Design pattern

Trong thiết kế kiến trúc phần mềm ở mức chi tiết (detail design) các đối tượng trong lập trình phần mềm (classes), áp dụng các mẫu thiết kế (design pattern) giúp giải quyết các vấn đề, những bài toán thường gặp. Design Pattern là những mẫu thiết kế đã được sử dụng, được đánh giá tốt giúp giải quyết những vấn đề thiết kế cụ thể. Design pattern chú trọng việc giúp cho bản thiết kế có tính uyển chuyển, dễ nâng cấp, thay đổi.

***Tôi đã nghiên cứu tìm hiểu design patterns nhằm áp dụng vào việc thiết kế các classes lập trình (diagram) giải quyết các bài toán cụ thể với mục tiêu giúp thiết kế kiến trúc hiệu quả, linh hoạt và khả năng tái sử dụng cao.***

Design patterns có ba nhóm chính:

* Structural: Cung cấp cơ chế xử lý những lớp không thể thay đổi (lớp thư viện của third party…), ràng buộc muộn (lower coupling) và cung cấp các cơ chế khác để kế thừa.
* Creational: Khắc phục các vấn đề khởi tạo đối tượng, hạn chế sự phụ thuộc vào nền tảng (platform).
* Behavioral: Che dấu hiện thực của đối tượng, che dấu giải thuật, hỗ trợ việc thay đổi cấu hình đối tượng một cách linh động

## Tìm hiểu và áp dụng cây quyết định vào thiết kế hệ thống

Giải pháp kiến trúc của đề tài chúng tôi là tổ chức tất cả các thông tin của hệ thống một cách linh hoạt, đảm bảo khả năng so khớp và khả năng khai thác dữ liệu nhằm tạo ra các tri thức, các thông tin hữu ích, mang tính chất quyết định. Trong phạm vi của đề tài mà chúng tôi định nghĩa: hệ thống tìm kiếm việc mà qua đó, nhà tuyển dụng có thể dễ dàng đưa ra các quyết định lựa chọn ứng viên của một công việc cụ thể. Sau khi tìm hiểu các kỹ thuật phân loại thông tin (classification) như cây quyết định, decision tree, Naïve Bayes, phân tích nghiệp vụ của bài toán, chúng tôi quyết định lựa chọn và áp dụng cây quyết định vào đề tài.

Cây quyết định là một trong những thuật toán điển hình hỗ trợ phân loại và hồi quy, sử dụng rất tốt các mô hình dự đoán. Thuật toán này sẽ khảo sát sự ảnh hưởng của mỗi thuộc tính trong tập dữ liệu và kết quả của thuộc tính dự đoán, sau đó sử dụng các thuộc tính đầu vào với các quan hệ rõ ràng để tạo thành một nhóm phân hoá (các node). Sự liên kết với nhau theo mức độ phụ thuộc giữa các node với nhau thiết lập nên cấu trúc dạng cây.

* Trong phạm vi đề tài này, tôi đã tìm hiểu lý thuyết cơ bản về cây quyết định, để áp dụng một cách phù hợp hiệu quả.
* Ngoài ra, tôi còn tìm hiểu kỹ thuật xây dựng cây quyết định Microsoft Decision Tree từ giải pháp khai thác dữ liệu Business Intelligence của công cụ Microsoft SQL Server 2008 R2.

## Giải pháp tổ chức thông tin linh hoạt đảm bảo khả năng so khớp

### Thiết kế lưu trữ thông tin linh hoạt theo Tag đáp ứng khả năng so khớp thông tin nội tại.

Đã đề cập trong bài toán thứ nhất, Trong các phần mềm, thông tin của một đối tượng trong thế giới thực (real object) được mô hình hóa thành các đối tượng, các thực thể trong phần mềm (software object). Có nhiều cách để mô hình hóa (schema mapping) một đối tượng trong thế thực: mối quan hệ 1-1 (một đối tượng thực được biểu diễn bằng một đối tượng duy nhất trong phần mềm, mối quan hệ tập hợp (một đối tượng thực được biểu diễn bằng cách cấu thành bởi nhiều đối tượng trong phần mềm), quan hệ kế thừa… Không những chỉ tổ chức thông tin trong phần mềm (software object) mà còn phải tổ chức về mặt lưu trữ các đối tượng đó xuống cơ sở dữ liệu thì cũng có nhiều cách để tổ chức.

Áp dụng giải pháp tổ chức thông tin theo cây phân cấp (hierarchy) trong các phương pháp phân loại và tổ chức thông tin (taxonomy): cây phân cấp là dạng cấu trúc linh hoạt, trung gian có thể đáp ứng về các schema thông tinh linh động của nghiệp vụ đăng tải và so khớp thông tin, đồng thời hierarchy là cấu trúc dễ dàng linh động mở rộng thay đổi sau này: faced, network giúp giải quyết các bài toán chuyên sâu hơn về khai thác thông tin và trí tuệ nhân tạo

Tìm hiểu cơ sở lý thuyết từ tổ chức thông tin theo Tag và cây phân cấp (hierarchy), đánh giá, nghiên cứu, tôi triển khai giải pháp lưu trữ tất cả thông tin thành một table TagAttribute duy nhất

### Triển khai giải pháp schema mapping trong framework library (API)

Vấn đề: trong phần trên đã trình bày khó khăn về kiến trúc chung cho việc tổ chức thông tin linh hoạt đáp ứng cho nhiều mô hình thông tin của các website. Lập trình viên khi xây dựng các mới các website hoặc các website đã có có thể sử dụng API giải pháp chúng tôi để triển khai, nâng cấp website của họ giải quyết các bài toán nghiệp vụ đăng tải và so khớp thông tin. Vì vậy giải pháp schema mapping ra được cung cấp trong API.

***Tôi đã tìm hiểu, phân tích một số kỹ thuật để giải quyết vấn đề tượng tự trong các thư viện lập trình (ADO.NET Entity framwork (schema mapping), Web service (mapping các Contract với nhau) để rút ra giải pháp dùng kỹ thuật Custom Attribute, một đối tượng trong .NET dùng để đánh dấu, khai báo (declaration) các Classes, Methods, Properties trong lập trình. Kết hợp với các kỹ thuật Reflection mã nguồn, tôi đã xây dựng giải pháp mapping linh động không phụ thuộc tổ chức thông tin của website.***



## Giải pháp thiết kế kiến trúc tổng quan của kiến trúc hệ thống

### Thiết kế giải pháp kiến trúc tổng quan của framework

### Thiết kế giải pháp giải quyết vấn đề hiệu năng của hệ thống

### Thiết kế và triển khai thư viện lập trình (framework library)

## Thử nghiệm, đánh giá giải pháp framework

## Nghiên cứu công nghệ

### ASP.NET MVC 3

### ADO.NET Framework Entity

### Microsoft SQL Server 2008 R2 Business Intelligene

## Công việc khác

# Đánh giá kết quả thực hiện và hướng phát triển

## Đánh giá kết quả đạt được

## Những hạn chế

## Hướng phát triển

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

### Phát triển Semantic web

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

### Phân tán dữ liệu

### Công cụ hỗ trợ lựa chọn ứng viên

### Thu thập thông tin việc làm tự động

### Áp dụng quy trình tuyển dụng vào hệ thống

# Những kinh nghiệm có được sau khi thực hiện khoá luận

## Về chuyên môn

## Về kỹ năng làm việc nhóm

## 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.