**ĐẠI HỌC QUỐC GIA TP.HCM**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

--------- 🙡 ⯎ 🙣 ---------

****

**ĐỀ CƯƠNG LUẬN VĂN TỐT NGHIỆP**

**Đề tài:**

**XÂY DỰNG ỨNG DỤNG VỚI NHIỀU PHIÊN BẢN**

**CHO WINDOWS VÀ WEB**

**VỚI MÔ HÌNH MODEL-VIEW-PRESENTER VÀ .NET FRAMEWORK**

Giáo viên hướng dẫn: **Phạm Thi Vương**

Sinh viên thực hiện – nhóm:

**Vương Hà Thanh Mẫn 06520282**

**Nguyễn Minh Toàn 06520493**

Lớp: **CNPM01** Khóa: **2006-2010**

Thành phố Hồ Chí Minh - Tháng 09 Năm 2010

MỤC LỤC

[Phần 1: Giới thiệu chung 1](#_Toc271330642)

[1.1 Tóm tắt đề tài 1](#_Toc271330643)

[1.2 Sự hình thành của đề tài 1](#_Toc271330644)

[1.3 Mục tiêu và mục đích của đề tài 3](#_Toc271330645)

[1.4 Phạm vi của đề tài 4](#_Toc271330646)

[Phần 2: Tổng quan về các vấn đề nghiên cứu 5](#_Toc271330647)

[2.1 Cách thức chuyển đổi giữa phần mềm Windows và web hiện tại 5](#_Toc271330648)

[2.2 Mẫu thiết kế Model-View-Presenter 5](#_Toc271330649)

[2.3 .NET Framework 5](#_Toc271330650)

[2.3.1 WPF – Windows Presentation Foundation 5](#_Toc271330651)

[2.3.2 Silverlight 5](#_Toc271330652)

[2.3.3 WCF – Windows Communication Foundation 5](#_Toc271330653)

[2.3.4 ASP.NET MVC 5](#_Toc271330654)

[2.4 Ứng dụng quản lý ảnh Imagow 5](#_Toc271330655)

[Phần 3: Thực hiện đề tài 5](#_Toc271330656)

[3.1 Phương pháp nghiên cứu và thực hiện 5](#_Toc271330657)

[3.2 Kế hoạch thực hiện 6](#_Toc271330658)

[Phần 4: Bố cục dự kiến của luận văn 7](#_Toc271330659)

[Phần 5: Kết quả dự kiến 7](#_Toc271330660)

[PHỤ LỤC 7](#_Toc271330661)

[CÁC THUẬT NGỮ VÀ CHỮ VIẾT TẮT 7](#_Toc271330662)

[TÀI LIỆU THAM KHẢO 7](#_Toc271330663)

DANH MỤC CÁC HÌNH ẢNH

DANH MỤC CÁC BẢNG BIỂU

# Giới thiệu chung

## Tóm tắt đề tài

Đề tài “**Xây dựng ứng dụng với nhiều phiên bản cho Windows và Web với mô hình Model-View-Presenter và công nghệ .NET**” tập trung nghiên cứu xây dựng ứng dụng theo thiết kế MVP sao cho có thể cài đặt nhiều phiên bản của ứng dụng cho hai nền tảng Windows và web với chi phí thấp nhất. Kết hợp với các tiện ích của công nghệ .NET, đề tài xây dựng giải pháp cho các nhà phát triển trước đòi hỏi ứng dụng ngày nay có thể vừa chạy trên nền web vửa chạy trên nền Windows.

## Sự hình thành của đề tài

* **Thực trạng hiện tại:**

Hiện nay xu hướng xây dựng và sử dụng các ứng dụng trực tuyến ngày càng phát triển. Có một sự chuyển dịch lớn các ứng dụng dạng Windows form truyền thống sang các ứng dụng web. Giải pháp thường được sử dụng là tham khảo mã nguồn cũ và viết thêm phiên bản mới cho môi trường web. Khả năng tái sử dụng các **tầng dữ liệu (Model)** và **tầng xử lý nghiệp vụ (Business logic)** là rất hạn chế, nếu có thì cũng chỉ là các thư viện dùng chung. Phần lớn là phải viết lại hoàn toàn.

Ứng dụng web có điểm hạn chế là phụ thuộc vào tốc độ đường truyền và không tận dụng hết các tài nguyên của hệ thống. Đa phần khi dùng một ứng dụng web và một ứng dụng dạng Windows form thì người dùng thường cảm thấy ứng dụng Windows form sử dụng thoải mái hơn. Chẳng hạn như phần mềm Picasa có hai phiên bản trên web và trên Windows. Vì vậy nhu cầu xây dựng các phiên bản ngoại tuyến dạng Windows form cho các ứng dụng web là không nhỏ. Với tình trạng tương tự như trên, nhà phát triển gần như phải xây dựng ứng dụng lại từ đầu.

Điều này làm chi phí phát triển, kiểm thử và bảo trì sản phẩm tăng lên. Chỉ cần đặt một tình huống đơn giản chúng ta có thể thấy điểm hạn chế rất lớn của phương pháp truyền thống. Ví dụ như sản phẩm cần có các tính năng mới trên cả phiên bản web và Windows thì cần phải cập nhật thay đổi trên cả hai bộ mã nguồn. Sự tồn tại hai bộ mã nguồn khác nhau cho cùng một sản phẩm làm tăng mức độ rủi ro của dự án và có thể dẫn đến sự không tương đồng giữa các phiên bản, gây khó khăn cho người sử dụng.

* **Yêu cầu đặt ra:**

Để giải quyết vấn đề này đòi hỏi các nhà phát triển cần lập chiến lược cho sản phẩm ngay từ đầu và thiết kế ứng dụng sao cho có thể dễ dàng chuyển đổi qua lại giữa các nền tảng.

* **Lý do chọn đề tài và hướng giải quyết:**

Trước những thực trạng và yêu cầu đặt ra đó, chúng em quyết định đi tìm giải pháp cho bài toán bằng cách sử dụng các mẫu thiết kế (design pattern) và các công nghệ hiện tại. Mẫu thiết kế chính được sử dụng trong đề tài là **Model-View-Presenter** và công nghệ được dùng để cài đặt là **.NET Framework**.

Về mặt lập trình, hai phiên bản web và Windows của cùng một sản phẩm có rất nhiều điểm tương đồng ở các tầng xử lý nghiệp vụ và tầng dữ liệu. Điểm khác biệt lớn nhất chính là ở tầng giao diện hiển thị cho người dùng. Các ứng dụng Windows và web sở dĩ khó chuyển đổi qua lại vì tầng xử lý của chúng gắn quá chặt và phụ thuộc nhiều vào tầng giao diện. Mô hình **Model-View-Controller (MVC)** là một mô hình rất quen thuộc với các lập trình viên ngày nay và có lẽ chúng ta cũng không cần nói nhiều về nó. Tương tự như mô hình MVC, mô hình **Model-View-Presenter (MVP)** cũng phân thành các tầng một cách rõ ràng, chỉ có điều là MVP hướng đến việc tách biệt hoàn toàn tầng xử lý và tầng dữ liệu ra khỏi tầng giao diện thông qua một giao diện chung cho tầng giao diện. Vì thế ứng dụng xây dựng theo mô hình MVP có thể đạt đến mức độ “độc lập giao diện”.

**.NET Framework** là một trong những nền tảng đang rất phát triển hiện nay với ưu điểm là cung cấp rất nhiều công cụ lập trình mạnh và hỗ trợ rất tốt trên nền Windows, tốc độ xây dựng ứng dụng nhanh và dễ dàng. Với rất nhiều thư viện cùng các công cụ hỗ trợ xây dựng các ứng dụng từ Windows form, Windows service, web cho đến dịch vụ web.

## Mục tiêu và mục đích của đề tài

* **Mục tiêu:**
* Nghiên cứu lý thuyết về mô hình **MVP** và các cách áp dụng nó để tìm ra một giải pháp phù hợp cho nhu cầu chuyển đổi qua lại giữa ứng dụng web và Windows.
* Tìm hiểu lý thuyết và cách sử dụng các công nghệ **WPF, Silverlight, WCF** cùng các công nghệ có liên quan của **.NET 4.0** để phục vụ cho việc cài đặt.
* Xây dựng ứng dụng quản lý ảnh với các phiên bản chạy trên nền web và Windows form.
* Kết hợp mô hình **MVP** với mô hình **ASP.NET MVC** khi xây dựng phiên bản web cho ứng dụng. Sở dĩ cần kết hợp MVP với mô hình ASP.NET MVC vì đa số các ứng dụng web hiện nay đều được xây dựng dựa trên các Framework MVC được xây dựng sẵn, việc thay đổi các bộ thư viện này là không thể mà chỉ có thể kết hợp với MVP để hướng đến mục tiêu có thể sử dụng chung tầng xử lý khi xây dựng ứng dụng trên Windows.
* Đánh giá tính hiệu quả của việc sử dụng mô hình MVP về các mặt: thời gian phát triển, khả năng tái sử dụng các thành phần khi chuyển đổi qua lại giữa ứng dụng web và Windows, mức độ dễ dàng trong việc kiểm thử và bảo trì hệ thống.
* **Mục đích:**
* Đề tài không nhằm vào việc tạo ra một mô hình lập trình hoàn toàn mới mà là đề xuất một cách tiếp cận mang tính ứng dụng (best practice) bằng cách kết hợp các mẫu thiết kế và các công nghệ lại với nhau để xây dựng các ứng dụng dễ dàng chuyển đổi giữa môi trường web và Windows.
* Cho thấy được tính hiệu quả và tầm quan trọng của các mẫu thiết kế (design pattern) trong phát triển phần mềm.

## Phạm vi của đề tài

* Đề tài tập trung nghiên cứu các mẫu thiết kế MVC và MVP.
* Sử dụng ngôn ngữ lập trình C# cùng các công nghệ có liên quan trong .NET 4.0 làm công cụ cài đặt.
* Đề tài không tập trung sâu vào các công nghệ mà chỉ nghiên cứu ở mức độ sử dụng.
* Cài đặt ứng dụng ở mức độ vửa phải là ứng dụng quản lý ảnh với cách chức năng cơ bản để cụ thể hóa các nghiên cứu.

# Tổng quan về các vấn đề nghiên cứu

## Cách thức chuyển đổi giữa phần mềm Windows và web hiện tại

## Mẫu thiết kế Model-View-Presenter

## .NET Framework

### WPF – Windows Presentation Foundation

### Silverlight

### WCF – Windows Communication Foundation

### ASP.NET MVC

## Ứng dụng quản lý ảnh Imagow

# Thực hiện đề tài

## Phương pháp nghiên cứu và thực hiện

* Tiến hành khảo sát một số ứng dụng web và Windows để tìm ra những khó khăn trong việc chuyển đổi nền tảng (Windows và web) của các ứng dụng này. Xác định các điểm chung và riêng của ứng dụng Web và Windows.
* Nghiên cứu bằng thực nghiệm: nghiên cứu kiến thức lý thuyết và thực hiện các ví dụ và ứng dụng nhỏ cho kĩ thuật đã nghiên cứu. Ví dụ sau khi nghiên cứu lý thuyết mẫu thiết kế MVP, bắt tay vào xây dựng một ứng dụng nhỏ dạng “Hello world” cài đặt theo mô hình này. Bên cạnh đó còn tham khảo các ứng dụng mẫu đơn giản được thiết kế theo mô hình MVP và sử dụng các kĩ thuật có liên quan.
* Xây dựng cấu trúc tổng quát của ứng dụng theo mô hình hướng đối tượng (OOD); xây dựng tầng xử lý nghiệp vụ (Presenter) và tầng truy xuất dữ liệu (Model) là các thành phần dùng chung; sau đó tiến hành cài đặt chi tiết cho từng phiên bản web và Windows.

## Kế hoạch thực hiện

|  |  |
| --- | --- |
| Thời gian | Công việc thực hiện |
| Tháng 09/2010 | * Thu thập tài liệu và hoàn thành đề cương luận văn. * Tìm hiểu lý thuyết và thực hành các mẫu thiết kế MVC và MVP. * Hoàn thành tài liệu về mô hình MVC và MVP. |
| Tháng 10/2010 | * Khảo sát một số ứng dụng web và Windows. Ghi lại kết quả khảo sát. * Xây dựng cấu trúc MVP tổng quát và cài đặt các ứng dụng nhỏ dạng “Hello world” dựa trên bộ khung này để tối ưu hóa dần. * Nghiên cứu các công nghệ WPF, Silverlight, WCF. * Tìm hiểu ASP.NET MVC và tìm giải pháp kết hợp với mô hình MVP đã xây dựng. * Hoàn thành các tài liệu về những kĩ thuật và công nghệ đã tìm hiểu. |
| Tháng 11/2010 –  Tháng 12/2010 | * Tiến hành phân tích và thiết kế phần mềm quản lý ảnh **Imagow** (tên mã của chương trình). * Hoàn thành các tài liệu cho chương trình: đặc tả yêu cầu, thiết kế các module và các lớp, thiết kế dữ liệu, thiết kế giao diện,… * Cài đặt Imagow cho môi trường Windows theo mô hình MVP. |
| Tháng 01/2011 –  Tháng 02/2011 | * Hoàn thiện Imagow cho Windows. * Kiểm thử và sửa lỗi. * Tiến hành cài đặt Imagow phiên bản web dựa trên phiên bản Imagow cho Windows. |
| Thang 03/2011 | * Hoàn chỉnh luận văn, báo cáo thử và chỉnh sửa. * Bảo vệ luận án. |

# Bố cục dự kiến của luận văn

# Kết quả dự kiến

* Xây dựng được mô hình MVP tổng quát và giải pháp kết hợp MVP với các bộ thư viện MVC dùng để xây dựng ứng dụng web (cụ thể là ASP.NET MVC).
* Xây dựng được ứng dụng quản lý ảnh với hai phiên bản web và Windows.
* Khả năng tái sử dụng mã nguồn của chương trình là khoảng 70-80% khi xây dựng ứng dụng trên nền tảng khác và thời gian xây dựng được rút ngắn khoảng phân nửa so với mô hình truyền thống.
* Chương trình có thể dễ dàng kiểm thử, bảo trì và nâng cấp với chi phí chỉ bằng một nửa so với mô hình truyền thống.

PHỤ LỤC

CÁC THUẬT NGỮ VÀ CHỮ VIẾT TẮT

TÀI LIỆU THAM KHẢO

* **Tiếng Anh:**