TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TIỂU LUẬN GIỮA KÌ**

**LẬP TRÌNH WEB VÀ ỨNG DỤNG**

**ASP.NET Core**

*Người hướng dẫn*: **TS VŨ ĐÌNH HỒNG**

*Người thực hiện*: **LÊ TRẦN PHÚ – 52100572**

**NGUYỄN THÀNH HỒNG ĐỨC - 52100014**

Lớp **: 21050201**

Khoá  **: 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2014**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TIỂU LUẬN GIỮA KÌ**

**LẬP TRÌNH WEB VÀ ỨNG DỤNG**

**ASP.NET Core**

*Người hướng dẫn*: **TS VŨ ĐÌNH HỒNG**

*Người thực hiện*: **LÊ TRẦN PHÚ – 52100572**

**NGUYỄN THÀNH HỒNG ĐỨC - 52100014**

Lớp **: 21050201**

Khoá  **: 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2014**

LỜI CẢM ƠN

Chúng em xin được gửi lời cảm ơn chân thành đến quý thầy cô trong khoa Công nghệ thông tin trường Đại học Tôn Đức Thắng đã tạo điều kiện cho chúng em có cơ hội được tham gia học và thực hiện tiểu luận môn Lập trình Web và ứng dụng.

Đây là một môn học mà chúng em cảm thấy rất quan trọng trong định hướng tương lại của chúng em.

Bên cạnh đó chúng em cũng xin chân thành cảm ơn thầy Vũ Đình Hồng đã tận tình chỉ dạy và hướng dẫn chúng em trong quá trình học tập cũng như hoàn thành bài báo cáo này. Thầy là người đã dẫn dắt chúng em để chúng em có thể hiểu hơn về môn học này trong thời gian vừa qua.

Tuy vậy, về phần kiến thức bản thân chúng em cũng còn nhiều phần hạn chế, vì vậy sẽ không thể tránh khỏi một số sai sót không đáng có, kính mong nhận được những ý kiến đóng góp từ các thầy cô.

Trân trọng.

ĐỒ ÁN ĐƯỢC HOÀN THÀNH

TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của chúng tôi và được sự hướng dẫn của  thầy Vũ Đình Hồng. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực  và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu  phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn  khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của  các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc. **Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm  về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến  những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có). *\*

*TP. Hồ Chí Minh, ngày tháng 4 năm 2023*

*Tác giả*

*(Ký tên và ghi rõ họ tên)*

*Lê Trần Phú*

*Nguyễn Thành Hồng Đức*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

MỤC LỤC

**No table of contents entries found.**

**DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

**CÁC KÝ HIỆU**

*f Tần số của dòng điện và điện áp (Hz)*

*p Mật độ điện tích khối (C/m3)*

**CÁC CHỮ VIẾT TẮT**

MVC Model-View-Controller

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 2.1: Kiến trúc FTP 9](#_Toc133226414)

**DANH MỤC BẢNG**

[Bảng 3.1 Ví dụ cho chèn bảng 1](#_Toc387689363)

CHƯƠNG 1 – MÔ HÌNH MVC

1.1 Mô hình MVC là gì?

MVC là viết tắt cho Model-View-Controller.

Mô hình MVC là một mô hình kiến trúc phần mềm được sử dụng rộng rãi trong phát triển các ứng dụng web.

Mô hình này được thiết kế để phân tách ứng dụng thành 3 thành phần khác nhau, được kết nối với nhau và mỗi phần đều có một nhiệm vụ riêng và độc lập với các thành phần khác.

Việc phân tách như vậy giúp cho quá trình phát triển, bảo trì và mở rộng ứng dụng trở nên dễ dàng hơn.

1.2 Sự ra đời của mô hình MVC

MVC được tiến sĩ Trygve Reenskaug đưa vào ngôn ngữ lập trình Smalltalk-76 khi ông đến trung tâm Nghiên cứu Xerox Palo Alto (PARC) vào giữa năm 1970.

Mô hình này được nghiên cứu bởi các nhân viên Xerox PARC để hỗ trợ phát triển ứng dụng desktop.

Năm 1988, các bài báo “The Journal of Object Technology” – JOT mang lại bước tranh toàn cảnh về MVC mang liệu sự hiệu quả tốt nhất.

Đến những năm 1990, khi các ứng dụng web bắt đầu phát triển mạnh mẽ, mô hình này được áp dụng rộng rãi trong phát triển các ứng dụng web.

Trong thời gian gần đây, mô hình MVC được sử dụng rộng rãi trong các ứng dụng web hiện đại.

1.3 Kiến trúc, các thành phần của mô hình MVC



Hình 1: Các thành phần trong mô hình MVC.

Các thành phần chính của mô hình MVC bao gồm:

1. Model: là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng và là cầu nối giữa 2 thành phần bên dưới là View và Controller. Một model là dữ liệu được sử dụng bởi chương trình. Đây có thể là cơ sở dữ liệu hay một đối tượng đơn giản.
2. View (Giao diện): thể hiện giao diện người dùng. View hiển thị thông tin cho người dùng và cung cấp một giao diện để tương tác với ứng dụng.
3. Controller (Điều khiển): là phần quan trọng nhất trong mô hình, nó liên kết Model và View. Là bộ phận logic xử lý các yêu cầu và sự kiện từ người dùng. Controller xử lý các yêu cầu này và thay đổi trạng thái của Model tương ứng. Controller cũng có thể chịu trách nhiệm cho việc truy xuất cơ sở dữ liệu.

1.4 Luồng xủ lý trong MVC



Hình 2: Luồng xử lý trong MVC.

Quá trình xử lý trong mô hình MVC như sau:

1. Người dùng sẽ tương tác với hệ thống thông qua View.
2. Khi này request của người dùng sẽ được chuyển đến Controller.
3. Controller tiếp nhận request và xử lý theo hướng được yêu cầu.
4. Nếu request là chuyển trang mà không phải gửi và nhận dữ liệu thì Controller sẽ chuyển hướng browser và kết thúc luồng.
5. Nếu request là yêu cầu liên quan đến dữ liệu thì Controller sẽ giao tiếp với Model.
6. Model tương tác với database để lấy dữ liệu và gửi về Controller.
7. Controller gửi dữ liệu trở lại cho View.
8. View sẽ hiển thị dữ liệu tương ứng lên giao diện của người dùng.

1.5 Ưu, nhược điểm của mô hình MVC

1.5.1 Ưu điểm

Mô hình MVC được chia thành các phần độc lập nên source code trong quá trình phát triển ứng dụng sẽ dễ đọc, dễ hiểu hơn, ngoài ra trong việc nâng cấp, bảo trì cũng sẽ tiện lợi hơn.

Các thành phần trong mô hình MVC được tách biệt nên việc tái sử dụng cho các ứng dụng các là rất tiện lợi vì không cần phải thay đổi quá nhiều.

Mô hình MVC giúp tiết kiệm băng thông hơn vì không sử dụng viewstate. Người dùng có thể gửi, nhận dữ liệu một cách liên tục. Điều này giúp website hoạt động ổn định.

Việc kiểm tra lỗi sẽ dễ dàng hơn và sản phẩm đầu ra cũng sẽ uy tín hơn.

Các lập trình viên có thể làm việc cùng lúc, công việc sẽ không ảnh hưởng đến nhau.

Mô hình MVC cho phép linh hoạt động trên các nền tảng ngôn ngữ khác nhau.

1.5.1 Nhược điểm

Mô hình MVC có kiến trúc khá phức tạp cho nên việc triển khai sẽ phức tạp hơn so với các mô hình khác.

Thời gian xây dựng các thư viện, cấu trúc cũng bị gia tăng đáng kể, dẫn đến việc tăng chi phí sản xuất.

Mô hình MVC chỉ phù hợp với các ứng dụng có quy mô lớn, vì thời gian xây dựng nền tảng khá lâu nên khi sử dụng mô hình này với một số ứng dụng nhỏ không mang lại hiệu quả cao.

CHƯƠNG 2 – ASP.NET CORE

2.1 ASP.NET Core là gì?

ASP.NET Core là một tập hợp các thư viện như một framework mã nguồn mở, hiệu suất cao và đa nền tảng, được dùng để xây dựng các ứng dụng hiện đại được kích hoạt bởi Cloud. ASP.NET Core được sử dụng trong việc xây dựng các web apps và dịch vụ, ứng dụng IOT và Mobile Backend.

ASP.NET Core chạy trên cơ sở là tập hợp của các gói Nuget Packages. Với các Nuget Packages cho phép tối ưu thiết kế ứng dụng, vì vậy ứng dụng được thiết kế cho ra sản phẩm có dung lượng nhỏ hơn, bảo mật tốt hơn và hiệu quả ổn định hơn.

Quá trình phát triển, xây dựng và bảo trì cũng được đơn giản hóa.

Với ứng dụng được phát triển ra, có thể sử dụng được trên Windows, macOS và cả Linux.

ASP.NET Core được thiết kế lại từ ASP.NET 4.x, gồm các thay đổi về kiến trúc tạo ra một framework gọn nhẹ và linh hoạt hơn

2.2 Sự ra đởi của ASP.NET Core

ASP.NET Core là một phiên bản mã nguồn mở của ASP.NET và là phiên bản cải tiến của nó.

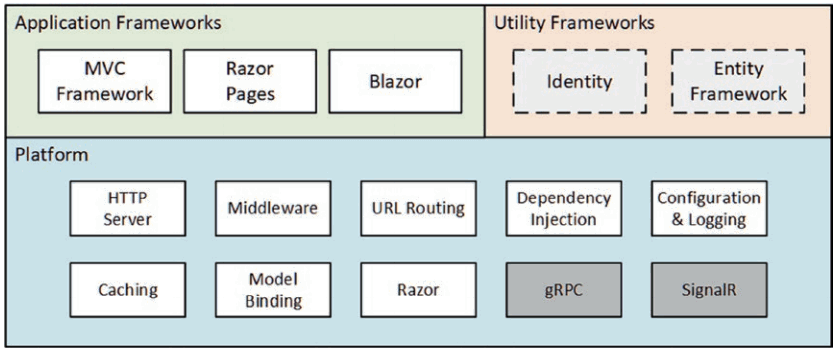
Được phát triển với Microsoft và được ra mắt lần đầu tiên vào năm 2016.

Với phiên bản ASP.NET chỉ chạy trên nền tảng Windows thì ASP.NET Core có thể linh hoạt chạy trên .NET Framework trên Windows và .NET đa nền tảng.

Từ phiên bản 3 trở đi thì ASP.NET Core chỉ hoạt động trên .NET Core mà không còn hỗ trợ .NET Framework nữa.

2.3 Các thành phần của ASP.NET Core

ASP.NET Core bao các thành phần cơ bản sau:



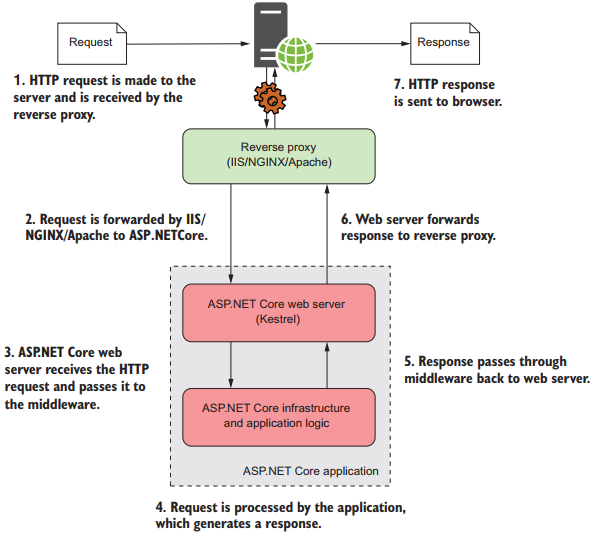
Hình 3: Kiến trúc ASP.NET Core

1. Application Frameworks: chứa một số framework như MVC Framework, Razor Pages, ... Đây là những framework giúp xây dựng các dạng khác nhau của ứng dụng web. Đây là những mẫu kiến trúc được xây dựng sẵn nhằm hỗ trợ cho việc phát triển. Việc chọn framework sẽ tùy vào yêu cầu của dự án. Trong quá trình phát triển có thể chuyển đổi hoặc kết hợp các framework lại với nhau.
2. Utility Frameworks: chứa những framework hỗ trợ cho ứng dụng, bảo mật và cơ sở dữ liệu, ví dụ như Entity Framework Core và ASP.NET Identity. Hai framework này là không bắt buộc nhưng thường được sử dụng trong hầu hết các ứng dụng ASP.NET Core.
3. Platform là những gì tạo nên nền tảng chung mà mọi ứng dụng ASP.NET Core đều cần đến. Bên trong có chứa nhiều thành phần cấp thấp cần cho việc nhận và xử lý truy vấn HTTP, cũng như tạo ra các phản hồi phù hợp. Các thành phần trong Platform khá đa dạng, ví dụ như: HTTP Server, Middleware, Razor, ...

2.4 Cơ chế hoạt động của ASP.NET Core

ASP.NET Core hoạt động dựa trên một pipeline xử lý yêu cầu HTTP nhẹ, hiệu suất cao và linh hoạt. Pipeline này được hiểu như là một loạt các middleware, mỗi middleware sẽ xử lý một phần sau đó chuyển đến middleware tiếp theo hoặc kết thúc yêu cầu. Một quy trình có thể được hiểu như sau:

1. ASP.NET Core nhận yêu cầu HTTP được gửi đến ứng dụng web thông qua middleware xử lý yêu cầu trong pipeline.Middleware có thể thực hiện các chức năng như xác thực, ghi log, nén, mã hóa đường truyền.
2. Yêu cầu HTTP này sẽ được chuyển đến một Controller tương ứng. Controller này sẽ xử lý yêu cầu bằng cách truy xuất dữ liệu từ mô hình và tạo ra kết quả hiển thị trên giao diện của người dùng.
3. Controller sẽ thực hiện truy xuất hoặc lưu dữ liệu bằng cách gọi các phương thức.
4. Kết quả sẽ được trả về giao diện của người dùng thông qua View.
5. Khi kết quả được trả về cho người dùng, middleware cuối cùng sẽ được gọi để thực hiện chức năng ghi log và nén.



Hình 4: Mô hình hoạt động của ASP.NET Core.

2.5 Ưu và nhược điểm của ASP.NET Core

2.5.1 Ưu điểm của ASP.NET Core

Đa nền tảng: ASP.NET Core có thể chạy trên Windows, Linux và macOS giúp việc phát triển trở nên dễ dàng hơn và đáp ứng nhu cầu cảu người dùng.

Hiệu suất cao: ASP.NET Core được tối ưu hóa để cung cấp hiệu suất cao và tốc độ xử lý nhanh hơn.

ASP.NET Core được tích hợp nhiều công nghệ như Entity, Framework Core, ... dễ hỗ trợ trong quá trình phát triển.

Tính bảo mật cao: với các tính năng bảo mật hiện đại như xác thực và phân quyền, bảo vệ các cuộc tấn công độc hại và bảo mật dữ liệu.

Tính mở rộng cao: việc sử dụng ASP.NET Core giúp cho việc phát triển và mở rộng trở nên dễ dàng hơn với các tiện ích được hỗ trợ như Dependency Injection, Middleware, ...

Linh hoạt: ta có thể sử dụng nhiều ngôn ngữ lập trình khác nhau, cho phép các nhà phát triển sử dụng ngôn ngữ lập trình ưa thích của mình để phát triển ứng dụng.

2.5.2 Nhược điểm của ASP.NET Core

Việc nghiên cứu và sử dụng ASP.NET Core là khá khó khăn vì đây là một framework khá rộng và phức tạp, đòi hỏi kiến thức chuyên sâu. Vì vậy gây khó khăn ít nhiều cho các lập trình viên mới bắt đầu.

Vì là một công nghệ mới nên luôn được phát triển liên tục qua thời gian, vì vậy có thể dẫn đến những thay đổi trong quá trình sử dụng framework trên các phiên bản mới, dẫn đến việc cập nhật, bảo trì ứng dụng trở nên khó khăn.

Entity Framework Core là một công nghệ phổ biển trong ASP.NET Core nhưng trong các dự án lớn, việc này có thể giảm hiệu suất.

Một số tính năng còn chưa được hoàn thiện, vì đây là một công nghệ mới cho nên việc xảy ra lỗi trong quá trình sử dụng là điều khó tránh phải.

Tuy nhiên những nhược điểm trên có thể được xử lý trong quá trình phát triển, ngoài ra ưu điểm mà nó mang lại quá lớn nên vẫn được các nhà phát triển lựa chọn sử dụng.

**TÀI LIỆU THAM KHẢO**

**Tiếng Việt**

1. [MVC là gì? Ứng dụng của mô hình MVC trong lập trình (vietnix.vn)](https://vietnix.vn/tim-hieu-mo-hinh-mvc-la-gi/)
2. [Tìm hiểu mô hình MVC dành cho người mới bắt đầu: Cấu trúc và ví dụ - Viblo](https://viblo.asia/p/tim-hieu-mo-hinh-mvc-danh-cho-nguoi-moi-bat-dau-cau-truc-va-vi-du-V3m5WLDyKO7)
3. [Cách hoạt động của ứng dụng ASP.NET Core | Tự học ICT (tuhocict.com)](https://tuhocict.com/cach-hoat-dong-cua-ung-dung-aspnet-core/)
4. [ASP.NET Core là gì? Những lợi ích khi sử dụng ASP.NET Core (itnavi.com.vn)](https://itnavi.com.vn/blog/aspnet-core-la-gi)

**Tiếng Anh**

1. [MVC Framework - Introduction (tutorialspoint.com)](https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm#:~:text=The%20Model%2DView%2DController%20(,development%20aspects%20of%20an%20application.)
2. [MVC Framework Introduction - GeeksforGeeks](https://www.geeksforgeeks.org/mvc-framework-introduction/)
3. [ASP.NET MVC Architecture (tutorialsteacher.com)](https://www.tutorialsteacher.com/mvc/mvc-architecture)
4. [The Model View Controller Pattern – MVC Architecture and Frameworks Explained (freecodecamp.org)](https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/)
5. [MVC - MDN Web Docs Glossary: Definitions of Web-related terms | MDN (mozilla.org)](https://developer.mozilla.org/en-US/docs/Glossary/MVC)

**PHỤ LỤC**

Phần này bao gồm những nội dung cần thiết nhằm minh họa hoặc hỗ trợ cho nội dung luận văn như số liệu, biểu mẫu, tranh ảnh. . . . nếu sử dụng những câu trả lời cho một *bảng câu hỏi thì bảng câu hỏi mẫu này phải được đưa vào phần Phụ lục ở dạng nguyên bản* đã dùng để điều tra, thăm dò ý kiến; **không được tóm tắt hoặc sửa đổi**. Các tính toán mẫu trình bày tóm tắt trong các biểu mẫu cũng cần nêu trong Phụ lục của luận văn. Phụ lục không được dày hơn phần chính của luận văn