TRƯỜNG ĐẠI HỌC HẢI PHÒNG

**KHOA CÔNG NGHỆ THÔNG TIN**



****

**ĐỒ ÁN TỐT NGHIỆP**

**CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN**

**TÊN ĐỀ TÀI: NGHIÊN CỨU MÔ HÌNH LẬP TRÌNH MVC VÀ XÂY DỰNG WEBSITE TRẮC NGHIỆM ONLINE**

Sinh viên thực hiện: Nguyễn Bình Phương Anh

Ngày sinh: 18/07/1999

Lớp: CNTT1.K18 Khóa: 2017 - 2021

Giảng viên hướng dẫn: PGS.TS Lê Đắc Nhường

*Hải Phòng tháng 4 năm 2021*

TRƯỜNG ĐẠI HỌC HẢI PHÒNG

**KHOA CÔNG NGHỆ THÔNG TIN**



****

**ĐỒ ÁN TỐT NGHIỆP**

**CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN**

**TÊN ĐỀ TÀI: NGHIÊN CỨU MÔ HÌNH LẬP TRÌNH MVC VÀ XÂY DỰNG WEBSITE TRẮC NGHIỆM ONLINE**

Sinh viên thực hiện: Nguyễn Bình Phương Anh

Mã sinh viên: 173148201278

Ngày sinh: 18/07/1999

Lớp: CNTT1.K18 Khoá: 2017-2021

Giảng viên hướng dẫn: PGS.TS Lê Đắc Nhường

*Hải Phòng tháng 4 năm 2021*LỜI CẢM ƠN

Trước hết, em xin bày tỏ lòng biết ơn sâu sắc tới Thầy Lê Đắc Nhường, người đã trực tiếp hướng dẫn em làm báo cáo thực tập này.

Em xin chân thành cảm ơn sự giúp đỡ của toàn thể các giảng viên khoa công nghệ thông tin, trường Đại học Hải Phòng, cũng như sự chỉ bảo của các anh chị Công ty TNHH thương mại thế giới số Huy Hậu nơi em thực tập đã dìu dắt, dạy dỗ em cả về kiến thức chuyên môn và tinh thần học tập độc lập, sáng tạo để em có được kiến thức thực hiện báo cáo của mình.

Cuối cùng, em xin bày tỏ lòng biết ơn sâu đậm của mình tới gia đình, bạn bè những người luôn sát cánh bên em, tạo mọi điều kiện tốt nhất để em có thể thực hiện đồ án tốt nghiệp của mình.

Trong quá trình thực hiện báo cáo thực tập, mặc dù đã cố gắng hết sức song do thời gian và khả năng có hạn nên em không thể tránh khỏi những thiếu sót. Vì vậy, em rất mong nhận được sự thông cảm, chỉ bảo và giúp đỡ của các thầy giáo, cô giáo và các bạn.

Một lần nữa em xin chân thành cảm ơn!

# DANH MỤC CÁC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| **Từ viết tắt** | **Giải thích** |
| API | Application Programming Interface |
| ASP | Active Server Pages |
| CSDL | Cơ sở dữ liệu |
| CSS | Cascading Style Sheets |
| HTML | HyperText Markup Language |
| IDE | Integrated Developement Enviroment |
| MVC | Model-View-Controller |
| NET | Network Enabled Technologies |
| UC | Use Case |
| VS | Visual Studio |
| QTV | Quản trị viên |
| GV | Giáo viên |
| SV | Sinh viên |

# DANH MỤC HÌNH ẢNH

[Hình 2.1: Biểu đồ UC tổng quát 13](#_Toc68785208)

[Hình 2.2: Biểu đồ phân rã tác nhân SV 13](#_Toc68785209)

[Hình 2.3: Biểu đồ phân rã tác nhân GV 17](#_Toc68785210)

[Hình 2.4: Biểu đồ phân rã tác nhân QTV 21](#_Toc68785211)

[Hình 2.5: Biểu đồ tuần tự chức năng Đăng nhập – quản lý thông tin 32](#_Toc68785212)

[Hình 2.6: Biểu đồ tuần tự chức năng SV làm bài thi 32](#_Toc68785213)

[Hình 2.7: Biểu đồ tuần tự chức năng Xem điểm 33](#_Toc68785214)

[Hình 2.8: Biểu đồ tuần tự chức năng Quản lý GV 33](#_Toc68785215)

[Hình 2.9: Biểu đồ tuần tự chức năng Quản lý SV 34](#_Toc68785216)

[Hình 2.10: Biểu đồ tuần tự chức năng Quản lý ngành 34](#_Toc68785217)

[Hình 2.11: Biểu đồ tuần tự chức năng Quản lý khóa 35](#_Toc68785218)

[Hình 2.12: Biểu đồ tuần tự chức năng Quản lý lớp 35](#_Toc68785219)

[Hình 2.13: Biểu đồ tuần tự chức năng Quản lý môn 36](#_Toc68785220)

[Hình 2.14: Biểu đồ tuần tự chức năng Quản lý bài thi 36](#_Toc68785221)

[Hình 2.15: Biểu đồ hoạt động chức năng làm bài thi 37](#_Toc68785222)

[Hình 2.16: Biểu đồ hoạt động chức năng thiết lập đề thi 37](#_Toc68785223)

[Hình 2.17: Biếu đồ Lớp 38](#_Toc68785224)

[Hình 3.1: Giao diện đăng nhập 43](#_Toc68785225)

[Hình 3.2: Giao diện của QTV 43](#_Toc68785226)

[Hình 3.3: Giao diện của GV 44](#_Toc68785227)

[Hình 3.4: Giao diện của SV 44](#_Toc68785228)

[Hình 3.5: Giao diện Làm bài thi 45](#_Toc68785229)

# DANH MỤC CÁC BẢNG

[Bảng 1.1 Các phiên bản của ASP.NET MVC 3](#_Toc68785230)

[Bảng 1.2: Sự khách nhau giữa MVC và Webform 4](#_Toc68785231)

[Bảng 2.1: Các tác nhân và use case cơ bản của hệ thống 12](#_Toc68785232)

MỤC LỤC

[DANH MỤC CÁC CHỮ VIẾT TẮT iv](#_Toc69134969)

[DANH MỤC HÌNH ẢNH v](#_Toc69134970)

[DANH MỤC CÁC BẢNG v](#_Toc69134971)

[MỞ ĐẦU 1](#_Toc69134972)

[CHƯƠNG 1 CƠ SỞ LÝ THUYẾT 3](#_Toc69134973)

[1.1 Công nghệ ASP.NET MVC[1] 3](#_Toc69134974)

[1.1.1 Lịch sử ra đời của MVC 3](#_Toc69134975)

[1.1.2 Sự khác nhau giữa MVC và webform 4](#_Toc69134976)

[1.1.3 Kiến trúc sử dụng ASP.NET MVC 4](#_Toc69134977)

[1.1.4 Đặc điểm ASP.NET MVC Framework 6](#_Toc69134978)

[1.1.5 Ưu nhược điểm của mô hình MVC 7](#_Toc69134979)

[1.2 Ngôn ngữ mô hình hóa UML[3] 8](#_Toc69134980)

[1.2.1 Ngôn ngữ mô hình hóa 8](#_Toc69134981)

[1.2.2 Ngôn ngữ mô hình hóa UML 8](#_Toc69134982)

[CHƯƠNG 2 PHÂN TÍCH, THIẾT KẾ HỆ THỐNG 11](#_Toc69134983)

[2.1 Tổng quan về hệ thống 11](#_Toc69134984)

[2.2 Biểu đồ Use Case tổng quát 12](#_Toc69134985)

[2.3 Phân tích Use Case 13](#_Toc69134986)

[2.3.1 Các Use Case liên quan đến tác nhân Sinh viên 13](#_Toc69134987)

[2.3.2 Các Use Case liên quan đến tác nhân Giáo viên 17](#_Toc69134988)

[2.3.3 Các Use Case liên quan đến tác nhân Quản trị viên 21](#_Toc69134989)

[2.4. Xây dựng biểu đồ Tuần tự của hệ thống 32](#_Toc69134990)

[2.4.1 Các nhóm chức năng liên quan đến sinh viên 32](#_Toc69134991)

[2.4.2 Các nhóm chức năng liên quan đến giáo viên 33](#_Toc69134992)

[2.4.3 Các nhóm chức năng liên quan đến quản trị viên 33](#_Toc69134993)

[2.5 Biểu đồ hoạt động một số chức năng chính 37](#_Toc69134994)

[2.5.1 Biểu đồ hoạt động chức năng làm bài thi 37](#_Toc69134995)

[2.5.2 Biểu đồ hoạt động chức năng thiết lập đề thi 37](#_Toc69134996)

[2.6 Biểu đồ lớp 38](#_Toc69134997)

[CHƯƠNG 3 THỰC NGHIỆM XÂY DỰNG HỆ THỐNG 39](#_Toc69134998)

[3.1. Môi trường cài đặt 39](#_Toc69134999)

[3.1.1 Giới thiệu hệ quản trị SQL Server [2] 39](#_Toc69135000)

[3.1.2. Giới thiệu về Visual studio 2017 41](#_Toc69135001)

[3.2 Một số giao diện của Website 43](#_Toc69135002)

[3.2.1 Giao diện đăng nhập 43](#_Toc69135003)

[3.2.2 Giao diện của Quản trị viên 43](#_Toc69135004)

[3.2.3 Giao diện của Giáo viên 44](#_Toc69135005)

[3.2.4 Giao diện của Sinh viên 44](#_Toc69135006)

[3.2.5 Giao diện Làm bài thi 45](#_Toc69135007)

[3.2.6 Giao diện quản lý thông tin cá nhân 45](#_Toc69135008)

[KẾT LUẬN 46](#_Toc69135009)

[TÀI LIỆU THAM KHẢO 47](#_Toc69135010)

# MỞ ĐẦU

Hiện nay, hầu hết các cơ sở giáo dục, đào tạo vẫn đang áp dụng hình thức đánh giá kết quả học tập của học viên bằng phương pháp thi trên lớp, học viên làm bài trên giấy và giáo viên thu bài, chấm điểm vào bài thi cho học viên, và đề thi chủ yếu là tự luận. Đây là phương pháp còn nhiều nhược điểm về thời gian, chi phí cùng với độ chính xác và công bằng đối với học viên.

Theo Cục Khảo thí và Kiểm định chất lượng giáo dục của Bộ GD&ĐT, phương pháp làm kiểm tra, bài thi tự luận trước đây sẽ được dần thay đổi sang phương pháp thi trắc nghiệm. Trắc nghiệm được tiến hành thường xuyên ở các kỳ thi, kiểm tra để đánh giá kết quả học tập, giảng dạy đối với một phần của môn học, toàn bộ môn học, đối với cả một cấp học. Sự tiến bộ của phương pháp này ngày nay còn được hỗ trợ rất nhiều bởi công nghệ thông tin, nhiều website, phần mềm đã ra đời để hỗ trợ hình thức thi trắc nghiệm trực tuyến.

Trong đề tài này, em lựa chọn phân tích thiết kế một chương trình thi trắc nghiệm trực tuyến được viết trên nền web và sử dụng cơ sở dữ liệu, hoạt động trên môi trường mạng internet nhằm giúp giảm tải về mặt thời gian, chi phí cho việc tổ chức thi đồng thời đảm bảo tính khoa học, chính xác và áp dụng công nghệ, tính ưu việt của internet. Phần mềm này được thiết kế theo định hướng là một sản phẩm công cụ, do vậy nó có thể ứng dụng cho mọi môn học ở mọi cấp trong hệ thống giáo dục.

Do thời gian, trình độ cũng như kinh nghiệm còn nhiều hạn chế nên không thể tránh khỏi những thiếu xót trong khi thực hiện đề tài. Vì vậy, em rất mong nhận được sự góp ý của quý thầy, cô giáo để đề tài của em được hoàn thiện hơn.

**Mục tiêu nghiên cứu**

- Tìm hiểu công nghệ ASP.NET MVC, hệ quản trị cơ sở dữ liệu SQL Sever, IDE Visual Studio 2017.

- Thu thập các thông tin, các yêu cầu cần có của một website trắc nghiệm dùng để kiểm tra và đánh giá học sinh.

- Phân tích thiết kế hệ thống website dựa trên các thông tin thu thập được.Xây dựng hệ thống website quản lý và mua sắm online.

**Nội dung báo cáo gồm có 04 chương:**

* **Chương 1: Cơ sở lý thuyết**

Tìm hiểu về công nghệ ASP.NET MVC5, khái quát lên được ưu nhược điểm và tầm quan trọng của các công nghệ này trong việc xây dựng các hệ thống website hiện nay.

* **Chương 2: Phân tích hệ thống**

Khảo sát thực tế từ đó đề ra các mục tiêu cũng như các yêu cầu mà hệ thống cần đạt được.

* **Chương 3: Thiết kế hệ thống**

Từ việc khảo sát hệ thống ban đầu, xác định các chức năng cần có, luồng dữ liệu và các thực thể trong hệ thống từ đó thiết kế CSDL, xử lý các chức năng, thiết kế chương trình.

* **Chương 4: Thực nghiệm xây dựng hệ thống**

Từ dữ liệu có được qua phần phân tích, thiết kế hệ thống và ứng dụng từ việc tìm hiểu mô hình ASP.NET MVC, xác định môi trường triển khai và thiết kế hệ thống, đưa ra kết luận và rút kinh nghiệm.

# CƠ SỞ LÝ THUYẾT

## 1.1 Công nghệ ASP.NET MVC[1]

### 1.1.1 Lịch sử ra đời của MVC

Sự ra đời của giao diện đồ họa (GUI) và lập trình hướng đối tượng (Object Oriented Programming) cho phép lập trình viên làm việc với những thành phần đồ họa như những đối tượng đồ họa có thuộc tính và phương thức riêng của nó. Không dừng lại ở đó, những nhà nghiên cứu ở Xerox PARC còn đi xa hơn khi cho ra đời cái gọi là kiến trúc MVC (viết tắt của Model – View – Controller)

MVC được phát minh tại Xerox Parc vào những năm 70, bởi TrygveReenskaug. MVC lần đầu tiên xuất hiện công khai là trong Smalltalk-79

Thế hệ tiếp theo của MVC xuất hiện cùng với hệ điều hành NeXT và các phần mềm của nó.Kiến trúc này ngày càng được phát triển và hoàn thiện nhằm giải quyết các vấn đề cũng như các giải pháp cho quá trình phát triển phần mềm [8].

Vì vậy sau đó, lần lượt các MVC framework ra đời dựa trên mô hình MVC như: CodeIgniter, Zend, ASP.NET MVC …

Bảng 1.1 Các phiên bản của ASP.NET MVC

|  |  |
| --- | --- |
| **Thời gian phát hành** | **Phiên bản** |
| **2008** | ASP.NET MVC 1.0 |
| **2008** | ASP.NET MVC 2.0 (.NET 3.5) |
| **1/2010** | ASP.NET MVC 3.0 (.Net 4) |
| **15/8/2012** | ASP.NET MVC 4.0 (.Net 4.5) |
| **17/10/2013** | ASP.NET MVC 5.0 (.Net 5.0) |

### 1.1.2 Sự khác nhau giữa MVC và webform

Bảng 1.2: Sự khách nhau giữa MVC và Webform

|  |  |  |
| --- | --- | --- |
|  | **MVC** | **Webform** |
| Thành phần | ASP.net MVC chia ra làm 3 phần:  – Models  – View  – Controller.  Mọi tương tác của người dùng với Views sẽ được xử lý thông qua việc thực hiện các action hành động trong Controllers, không còn postback, lifecycle và events. | ASP.net WebForm sử dụng ViewState để quản lý các trang ASP.net đều có lifecycle, postback và dùng các web controls, các event để thực hiện các hành động cho UI (User Interface) . Khi có sự tương tác với người dùng nên hầu hết ASP.net WebForm xử lý chậm. |
| Việc kiểm tra (test), gỡ lỗi (debug) | Đối với MVC thì việc đó có thể sử dụng các unit test có thể thẩm định rất dễ dàng các Controllers thực hiện như thế nào. | Với ASP.net WebForm đều phải chạy tất cả các tiến trình của ASP.net, và sự thay đổi ID của bất kỳ Controls nào cũng ảnh hưởng đến ứng dụng. |

### 1.1.3 Kiến trúc sử dụng ASP.NET MVC

Như tên gọi MVC gồm 3 thành phần với các chức năng khác nhau [1]:

M (Model): là nơi chứa các nghiệp vụ tương tác với cơ sở dữ liệu. Nó bao gồm các class/ function xử lý nhiều nghiệp vụ như kết nối database, truy vấn dữ liệu, thêm – xóa – sửa dữ liệu…

V (View): Đảm nhận việc hiển thị thông tin, tương tác với người dùng, nơi chứa tất cả các đối tượng GUI như textbox, images… Hiểu một cách đơn giản, nó là tập hợp các form hoặc các file HTML.

C (Controller): Giữ nhiệm vụ nhận điều hướng các yêu cầu từ người dùng và gọi đúng những phương thức xử lý chúng… Chẳng hạn thành phần này sẽ nhận yêu cầu từ url và form để thao tác trực tiếp với Model.

#### 1.1.3.1 Model

Model là thành phần chủ yếu được sử dụng để thao tác xử lý dữ liệu. Trong các framework, Model vẫn thường sử dụng theo phương thức Active Record, một trong những design pattern. Chúng có tác dụng rút ngắn thời gian viết câu truy vấn cho người sử dụng. Biến những câu truy vấn phức tạp trở nên gần gũi và thân thiện với người sử dụng thông qua các thư viện được định nghĩa sẵn.

Model thường sẽ là các phương thức có trách nhiệm xử lý các tác vụ như: select, insert, update, delete các record trong database. Ứng với các lấy dữ liệu, model thường sử dụng mảng để gởi trả kết quả về. Vì mảng có thể cho phép model lưu trữ nhiều thông tin hơn, nên thường các bản ghi khi bóc tách chúng sẽ mang các dữ liệu của cơ sở dữ liệu một cách chi tiết.

Khi sử dụng Model, ta cũng cần tuân theo nguyên tắc chính của chúng là không xuất giá trị trực tiếp trong Model. Mà tất cả những dữ liệu ấy, phải đưa vào mảng và trả về theo phương thức. Và tiếp tục ở View ta sẽ sử dụng nó để lấy dữ liệu ra.

#### 1.1.3.2 View

View là phần hiển thị thông tin tương phản khi gửi và nhận yêu cầu. Trước đây, người ta thường sử dụng template để phân tách website thành 2 mảng riêng biệt. Một là giao diện và một là core. Việc chỉnh sửa giao diện trở nên đơn giản hơn đối so với cách viết thập cẩm lúc trước. Tuy nhiên, các thư viện này thực chất sẽ làm cho ứng dụng của chúng ta trở nên chậm chạp hơn bao giờ hết. Bởi chúng phải phiên dịch nhiều lần các kịch bản.

Vì vậy, View là một phần trong việc nâng cấp những hạn chế ấy. Chúng giúp giảm thiểu tối đa quá trình biên dịch nhiều lần. Và làm cho ứng dụng trở nên mạnh mẽ và chuyên nghiệp hơn nhiều so với cách lập trình thuần.

#### 1.1.3.3 Controller

Controller có trách nhiệm chính là điều hướng các yêu cầu của người sử dụng. Như vậy trên toàn ứng dụng của ta, tất cả các yêu cầu đều sẽ phải đi tới Controller. Và tại đây, ứng với các tham số người sử dụng truyền mà ta đưa họ đến một tác vụ nào đó trên ứng dụng.

Tại các tác vụ này, chúng sẽ thông qua lớp Model để làm việc và trả kết quả trở về Controller. Cuối cùng controller sẽ đẩy dữ liệu thao tác tới view. View là thành phần cuối cùng mà người sử dụng nhận được khi họ giở yêu cầu tới ứng dụng.

### 1.1.4 Đặc điểm ASP.NET MVC Framework

* Tiếp tục hỗ trợ các tính năng trong ASP.NET
  + Hỗ trợ sử dụng các các tập tin .ASPX, .ASCX, như là thành phần View.
  + Hỗ trợ đầy đủ các tính năng bảo mật của ASP.net: Form/Windows authenticate, URL authorization, membership/roles, output và data caching, section/profile state, configuration system, provider architecture.
* Tách rõ ràng các mối liên quan, mở ra khả năng test TDD (Test Driven Developer).
  + Có thể test unit trong ứng dụng mà không cần phải chạy Controllers cùng với tiến trình của ASP.NET và có thể dùng bất kỳ một unit testing framework nào như NUnit, MBUnit, MS Test,…
* Có khả năng mở rộng, mọi thứ trong MVC được thiết kế để dễ thay thế, dễ dàng tùy biến. Ánh xạ URL mạnh mẽ, cho phép xây dựng ứng dụng với những URL sạch.
* Không sử dụng mô hình post-back từ giao diện gửi đến server. Thay vào đó, chủ động đưa những post-back từ View đến thẳng lớp Controller.
* Hỗ trợ nhiều công cụ tạo View (Support for Multiple View Engines)
  + Cho phép chọn công cụ tạo view.
  + Các loại view engine
    - Web Forms (ASPX)
    - Razor
    - Hay một view engine nguồn mở như Spark, NHaml, NDjango.
* Hỗ trợ định tuyến
  + ASP.NET MVC Framework có một bộ máy ánh xạ URL thật sự mạnh mẽ.
  + Bộ máy này cung cấp phương pháp rất linh hoạt trong việc ánh xạ URL sang các Controller Classes.
  + Bạn có thể dễ dàng định ra các quy luật, cài đặt đường đi, ASP.NET dựa vào các quy luật đó để xác định Controller và action cần phải thực thi.
  + ASP.NET còn có khả năng phân tích URL, chuyển các thông số trong URL thành các tham số trong lời gọi hàm của Controller.
* Model Binding
  + Model Binding là tính năng thế mạnh của ASP.NET MVC (và bây giờ nó cũng được áp dụng cho cả Web Forms trong phiên bản ASP.NET 4.5).
  + Hỗ trợ bạn viết phương thức nhận một đối tượng tùy biến như là một tham số
  + Với sự hỗ trợ của Model Binding, bây giờ bạn chỉ cần tập trung vào việc cài đặt các nghiệp vụ logic, không cần phải bận tâm về việc suy nghĩ làm cách nào để ánh xạ dữ liệu từ người dùng sang các đối tượng .NET.
* Filters
  + Là tính năng mạnh trong ASP.NET MVC. Hỗ trợ cho việc kiểm tra tính hợp lệ trước khi một action method được gọi hoặc sau khi một action method thi hành.
* Razor View.

Từ ASP.net MVC 3 đi kèm với một công cụ View mới có tên là Razor với những lợi ích sau:

* + Cú pháp Razor là sạch sẽ và súc tích, đòi hỏi một số lượng tối thiểu các tổ hợp phím.
  + Việc tìm hiểu Razor tương đối dễ dàng vì nó dựa trên ngôn ngữ C# và Visual Basic.
  + Visual Studio bao gồm IntelliSense và mã cú pháp Razor được màu hóa.
  + Với Razor views có thể kiểm tra từng đơn vị mà không đòi hỏi bạn phải chạy các ứng dụng hoặc phải chạy website.

### 1.1.5 Ưu nhược điểm của mô hình MVC

**– Ưu điểm của MVC :** Cho thấy sự chuyên nghiệp trong lập trình và phân tích đối tượng, vì được chia các thành phần riêng biệt nên hoạt động độc lập tách biệt giúp phát triển ứng dụng nhanh hơn, đơn giản hơn và dễ nâng cấp, bảo trì hơn.

**– Nhược điểm của MVC:** Là mô hình làm việc rất bài bản nên với những ứng dụng nhỏ, sử dụng MVC rất tốn nhiều thời gian và gây ra nhiều phức tạp.

## 1.2 Ngôn ngữ mô hình hóa UML[3]

### 1.2.1 Ngôn ngữ mô hình hóa

Mô hình hóa là phương pháp đơn giản hóa các thành phần của hệ thống nhằm hiểu rõ hơn về hệ thống cần phát triển. Sản phẩm của quá trình này là các mô hình mô tả các khía cạnh khác nhau của hệ thống.

Đặc điểm mô hình:

* Hiển thị hệ thống như nó vốn có hoặc cần đạt tới.
* Kiểm chứng hệ thống bởi khách hàng.
* Cung cấp những chỉ dẫn để xây dựng hệ thống.
* Tài liệu hóa hệ thống

Ngôn ngữ mô hình hóa cho phép đặc tả, trực quan hóa và tài liệu hóa tiến trình phát triển phần mềm.

### 1.2.2 Ngôn ngữ mô hình hóa UML

UML, viết tắt của Unified Modelling Language, là ngôn ngữ mô hình hóa được tiêu chuẩn hóa bao gồm một bộ sơ đồ tích hợp, được phát triển để giúp các nhà phát triển hệ thống và phần mềm xác định, trực quan hóa, xây dựng và ghi lại các thành phần của hệ thống phần mềm, cũng như cho mô hình hóa kinh doanh và khác hệ thống phi phần mềm.

Ngoài ra, việc phát triển World Wide Web, trong khi làm cho một số thứ đơn giản hơn, đã làm trầm trọng thêm các vấn đề kiến ​​trúc này. Ngôn ngữ mô hình thống nhất (UML) được thiết kế để đáp ứng các nhu cầu này.

* Cung cấp cho người dùng một ngôn ngữ mô hình hóa trực quan sẵn sàng để sử dụng để họ có thể phát triển và trao đổi các mô hình có ý nghĩa.
* Cung cấp các cơ chế mở rộng và chuyên môn hóa để mở rộng các khái niệm cốt lõi.
* Độc lập với các ngôn ngữ lập trình cụ thể và các quy trình phát triển.
* Cung cấp một cơ sở chính thức để hiểu ngôn ngữ mô hình.
* Khuyến khích sự tăng trưởng của thị trường công cụ OO.
* Hỗ trợ các khái niệm phát triển cấp cao hơn như hợp tác, khung, mẫu và các thành phần.
* Tích hợp thực hành tốt nhất.

Điều đầu tiên cần chú ý về UML là có rất nhiều sơ đồ (mô hình) khác nhau để làm quen. Lý do cho điều này là có thể nhìn vào một hệ thống từ nhiều quan điểm khác nhau. Một sự phát triển phần mềm sẽ có nhiều bên liên quan đóng một phần.

UML cố gắng cung cấp một ngôn ngữ rõ ràng đến mức tất cả các bên liên quan đều có thể hưởng lợi từ ít nhất một sơ đồ UML.

Sơ đồ cấu trúc cho thấy cấu trúc tĩnh của hệ thống và các bộ phận của nó ở các mức độ trừu tượng và thực hiện khác nhau và cách chúng có liên quan với nhau. Các yếu tố trong sơ đồ cấu trúc đại diện cho các khái niệm có ý nghĩa của một hệ thống và có thể bao gồm các khái niệm trừu tượng, thế giới thực và thực hiện, có bảy loại sơ đồ cấu trúc như sau:

* Sơ đồ lớp
* Sơ đồ thành phần
* Sơ đồ triển khai
* Sơ đồ đối tượng
* Sơ đồ gói
* Sơ đồ cấu trúc hỗn hợp
* Sơ đồ hồ sơ

Biểu đồ hành vi cho thấy hành vi động của các đối tượng trong một hệ thống, có thể được mô tả như một chuỗi các thay đổi đối với hệ thống theo thời gian , có bảy loại sơ đồ hành vi như sau:

* Sử dụng sơ đồ trường hợp
* Sơ đồ hoạt động
* Sơ đồ máy nhà nước
* Sơ đồ trình tự
* Sơ đồ truyền thông
* Sơ đồ tổng quan tương tác
* Sơ đồ thời gian
* **Kết quả đạt được ở chương 1**:
* Hiểu về công nghệ ASP.NET MVC: biết được về lịch sử ra đời của MVC, sự khác nhau giữa MVC và webform, biết thêm về kiến trúc của ASP.NET MVC, sự tương tác giữa các thành phần trong mô hình MVC và các ưu nhược điểm của mô hình MVC.
* Hiểu rõ hơn về thương mại điện tử B2C.
* Cho thấy được tầm quan trọng của các công nghệ này trong việc xây dựng hệ thống website hiện nay.

Hiểu về ngôn ngữ mô hình hóa nói chung và ngôn ngữ mô hình hóa UML nói riêng, biết các sử dụng để áp dụng phân tích hệ thống phục vụ nghiên cứu

# PHÂN TÍCH, THIẾT KẾ HỆ THỐNG

## 2.1 Tổng quan về hệ thống

Hệ thống thi trực tuyến cho phép các học sinh của các lớp vào làm bài thi của mình trên máy tính của mình, các thông tin của hệ thống được đặt ở một server cố định.

Mỗi học sinh sẽ được cấp 1 tài khoản truy cập duy nhất trên hệ thống, tài khoản này bao gồm các thông tin: mã số học sinh (dùng để đăng nhập), mật khẩu (dùng để đăng nhập), họ và tên của học sinh, lớp và một phần tóm tắt thông tin của học sinh (ngày sinh, ảnh, các thông tin cá nhân khác). Mỗi học sinh chỉ thuộc vào một lớp nhất định, việc quản lý các học sinh sẽ thông qua đơn vị lớp mà học sinh đó học, lớp thì có: tên lớp, khóa học. Khóa học bao gồm các lớp cùng trong một niên khóa.

Khi học sinh đăng nhập vào hệ thống, tùy vào thông tin lớp mà học sinh đó học, hệ thống sẽ biết thời gian hôm nay học sinh đó thi môn nào và cho phép học sinh làm bài thi. Bài thi sẽ theo từng môn học trong một kì của lớp. Bài thi của một môn học sẽ do giáo viên dạy môn học đó cho lớp ra đề thi, đề thi bao gồm: thời gian thi, tên môn thi, thời gian làm bài, các câu hỏi. Câu hỏi sẽ bao gồm phần hỏi và phần thông tin trả lời để học sinh chọn (a,b,c,d) và điểm của câu hỏi. Mỗi đề thi đều có phần đáp án, bao gồm: tên môn thi, đáp án cho từng câu hỏi (a hoặc b hoặc c hoặc d). Sau khi học sinh nộp bài thi (hết thời gian, hệ thống tự chấm dứt hoặc học sinh nộp bài), hệ thống sẽ thực hiện lưu thông tin bài làm của học sinh, hệ thống chấm điểm bài thi khi người quản trị tương tác yêu cầu chấm bài thi, điểm của bài thi sẽ được lưu lại theo từng môn thi của học sinh.Sau khi có kết quả, học sinh có thể xem điểm các môn thi của mình

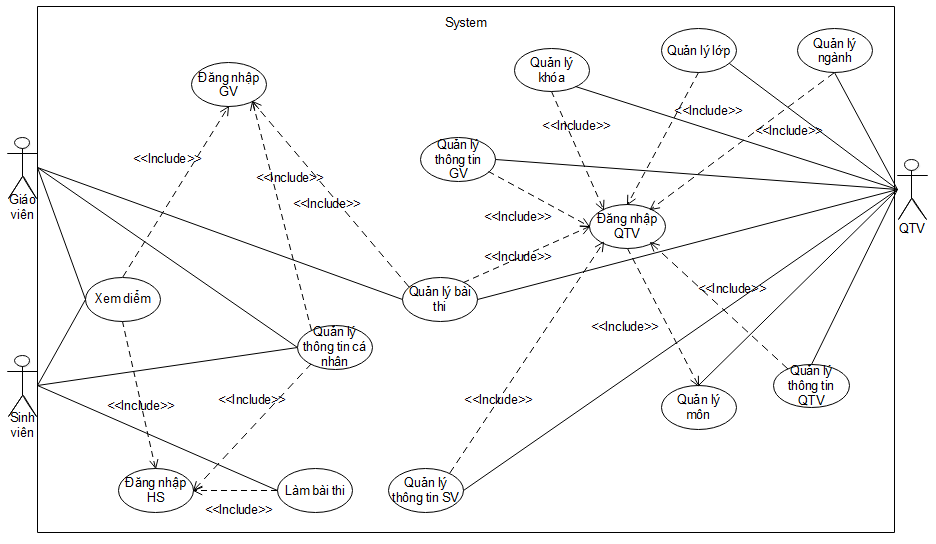
Giáo viên cũng được quản lý thông tin tương tự học sinh, giáo viên là những người giảng dạy các môn học cho các lớp, giáo viên có các thông tin: mã số giáo viên (để đăng nhập), mật khẩu (để đăng nhập), họ và tên, phần tóm tắt thông tin, dạy môn học nào cho các lớp nào Giáo viên sau khi đăng nhập vào hệ thống có quyền nhập đề thi cho các môn mình dạy, đề thi của một môn học có thể được dùng chung cho nhiều lớp mà giáo viên dạy hoặc sẽ riêng mỗi lớp sẽ có một đề, điều này sẽ do giáo viên qui định. Cùng với nhập đề thi giáo viên phải nhập thông tin của đáp án, sau khi lưu thông tin lại thì giáo viên không được phép sửa lại các thông tin trên. Một câu hỏi trong đề thi có tối đa 4 phương án trả lời, học sinh sẽ chọn 1 trong 4 phương án này, và đáp án sẽ lưu phương án trả lời đúng cho câu hỏi.

Quản trị hệ thống có quyền tạo, quản lý các tài khoản trên hệ thống cho học sinh và giáo viên, tạo, cập nhật, thay đổi thông tin cá nhân của các tài khỏan trên hệ thống Quản trị có quyền yêu cầu hệ thống chấm điểm bài thi theo từng môn học (đề thi) Quản trị có quyền thiết lập thời gian bắt đầu thi 1 đề thi nào đó của một lớp, để khi học sinh đăng nhập vào làm bài thi họ chỉ có 1 đề thi duy nhất để làm bài, quản trị viên có trách nhiệm nhập đúng thời gian thi để không có các môn thi cho một lớp bị đan chép thời gian thi. Quản trị có các chức năng quản lý lớp, khóa học, môn học, kỳ thi: tạo mới, cập nhật và trước mắt là không cho phép xóa các thông tin này. Khi xóa thông tin một học sinh thì đồng thời phải xóa toàn bộ thông tin điểm thi, bài làm của học sinh khỏi hệ thống

## 2.2 Biểu đồ Use Case tổng quát

Bảng 2.1: Các tác nhân và use case cơ bản của hệ thống

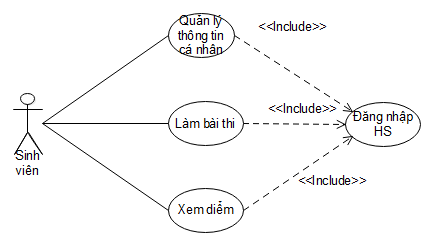
|  |  |  |  |
| --- | --- | --- | --- |
| Actor | QTV | Giáo viên | Sinh viên |
| Use case | - Đăng nhập  -Quản lý thông tin cá nhân | | |
|  | - Quản lý giáo viên  - Quản lý sinh viên  - Quản lý ngành  - Quản lý khóa  - Quản lý lớp  - Quản lý môn  - Quản lý bài thi | - Quản lý bài thi  - Xem điểm | - Làm bài thi  - Xem điểm |



Hình 2.1: Biểu đồ UC tổng quát

## 2.3 Phân tích Use Case

### 2.3.1 Các Use Case liên quan đến tác nhân Sinh viên



Hình 2.2: Biểu đồ phân rã tác nhân SV

#### 2.3.1.1 Use Case Đăng nhập SV

* + Tóm tắt: Đây là trường hợp sử dụng học sinh đăng nhập vào hệ thống để làm bài thi.
  + Tác nhân: Sinh viên
  + Liên quan: Không có các Use Case liên quan
  + Luồng sự kiện:
* **Luồng sự kiện chính:**
  + Học sinh chọn đăng nhập trên giao diện của hệ thống thi trực tuyến
  + Hệ thống hiển thị hộp thoại đăng nhập
  + Học sinh nhập ID tài khoản và mật khẩu đã được cấp
  + Hệ thống xác nhận mật khẩu
  + Hiển thị giao diện chức năng của học sinh
  + Kết thúc Use Case
* **Luồng sự kiện rẽ nhánh thứ nhất**
  + Học sinh hủy yêu cầu đăng nhập
  + Hệ thống bỏ qua hộp thoại đăng nhập, hiển thị giao diện ban đầu của hệ thống
  + Kết thúc Use Case
  + **Luồng rẽ nhánh thứ hai**
  + ID tài khoản và mật khẩu nhập vào không hợp lệ
  + Hệ thống từ chối đăng nhập
  + Hiển thị thông báo và hộp thoại đăng nhập
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ ba**
  + Hệ thống có lỗi trong quá trình xử lý xác nhận tài khoản
  + Hiển thị thông báo lỗi
  + Kết thúc Use Case

#### 2.3.1.2 Use Case Quản lý thông tin sinh viên

* + Tóm tắt: Đây là trường hợp sử dụng học sinh xem và thay đổi thông tin cá nhân của mình
  + Tác nhân: Sinh viên
  + Liên quan: Use Case này phải sử dụng Use Case Đăng nhập SV
  + Luồng sự kiện
* **Luồng sự kiện chính**
  + Học sinh chọn xem thông tin cá nhân
  + Hệ thống hiển thị thông tin cá nhân của học sinh
  + Học sinh có thể thay đổi một số thông tin: mật khẩu đăng nhập, các thông tin mô tả khác, nhập lại mật khẩu cũ
  + Học sinh chọn lưu thông tin thay đổi
  + Hệ thống xác nhận yêu cầu, kiểm tra dữ liệu và lưu thông tin mới
  + Hiển thị thông báo
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ nhất**
  + Học sinh hủy yêu cầu thay đổi thông tin
  + Hệ thống hiển thị lại thông tin ban đầu của học sinh
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ hai**
  + Mật khẩu cũ không hợp lệ
  + Hệ thống từ chối cập nhật
  + Hệ thống hiển thị lại thông tin đã nhập và yêu cầu nhập lại mật khẩu cũ
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ ba**
  + Hệ thống có lỗi trong quá trình xử lý
  + Hiển thị thông báo lỗi
  + Kết thúc Use Case

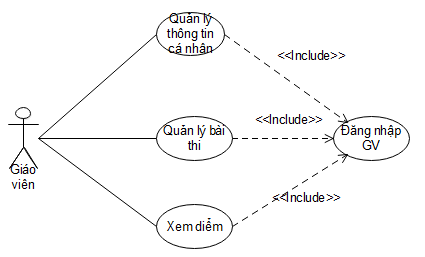
#### 2.3.1.3 Use Case Xem điểm

* + Tóm tắt: Đây là trường hợp sử dụng học sinh xem thông tin điểm các môn thi của mình
  + Tác nhân: Sinh viên
  + Liên quan: Use Case này phải sử dụng Use Case Đăng nhập SV
  + Luồng sự kiện:
* **Luồng sự kiện chính**
  + Học sinh chọn xem điểm
  + Hệ thống hiển thị thông tin điểm theo các môn thi của học sinh
  + Kết thúc Use Case
* **Luồng sự kiện rẽ nhánh**
  + Hệ thống có lỗi trong quá trình xử lý
  + Hiển thị thông báo lỗi
  + Kết thúc Use Case

#### 2.3.1.4 Use Case Thi trực tuyến

* + Tóm tắt: Đây là trường hợp sử dụng học sinh thực hiện bài thi của mình.
  + Tác nhân: Sinh viên
  + Liên quan: Use Case này phải sử dụng Use Case Đăng nhập SV
  + Luồng sự kiện
* **Luồng sự kiện chính**
* Học sinh chọn chức năng làm bài thi môn thi hiện tại
* Hệ thống hiển thị thông tin đề thi môn thi hiện tại của học sinh
* Học sinh chọn các câu trả lời trong đề thi.
* Kết thúc Use Case, học sinh chọn nộp bài hoặc hết thời gian làm bài.
* **Luồng rẽ nhánh thứ nhất**
  + Hiện tại không có môn thi nào cho học sinh
  + Hệ thống hiển thị thông báo không có môn thi
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ hai**
  + Hệ thống có lỗi trong quá trình xử lý
  + Hiển thị thông báo lỗi
  + Kết thúc Use Case.

### 2.3.2 Các Use Case liên quan đến tác nhân Giáo viên



Hình 2.3: Biểu đồ phân rã tác nhân GV

#### 2.3.2.1 Use Case Đăng nhập Giáo viên

* + Tóm tắt: Đây là trường hợp sử dụng giáo viên đăng nhập vào hệ thống
  + Tác nhân: Giáo viên
  + Liên quan: Không có các Use Case liên quan
  + Luồng sự kiện
* **Luồng sự kiện chính**
  + Giáo viên chọn đăng nhập trên giao diện quản lý của hệ thống thi trực tuyến
  + Hệ thống hiển thị hộp thoại đăng nhập
  + Giáo viên nhập ID tài khoản và mật khẩu đã được cấp, chọn đăng nhập vàochức năng giáo viên
  + Hệ thống xác nhận mật khẩu
  + Hiển thị giao diện chức năng của giáo viên
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ nhất**
  + Giáo viên hủy yêu cầu đăng nhập
  + Hệ thống bỏ qua hộp thoại đăng nhập, hiển thị giao diện ban đầu của hệthống
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ hai**
  + ID tài khoản và mật khẩu nhập vào không hợp lệ
  + Hệ thống từ chối đăng nhập
  + Hiển thị thông báo và hộp thoại đăng nhập
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ ba**
  + Hệ thống có lỗi trong quá trình xử lý xác nhận tài khoản
  + Hiển thị thông báo lỗi
  + Kết thúc Use Case

#### 2.3.2.2 Use Case Tạo đề thi

* + Tóm tắt: Đây là trường hợp sử dụng giáo viên tạo đề thi cho các lớp trên hệ thống
  + Tác nhân: Giáo viên
  + Liên quan: Use Case này phải sử dụng Use Case Đăng nhập GV
  + Luồng sự kiện
* **Luồng sự kiện chính**
* Giáo viên chọn chức năng tạo đề thi trên hệ thống các chức năng của giáo viên
* Hệ thống hiển thị giao diện chức năng tạo đề thi
* Giáo viên chọn kỳ thi, hệ thống sẽ hiển thị danh sách các môn học có kỳ thi vừa được chọn và do giáo viên đang làm việc với hệ thống giảng dạy
* Giáo viên chọn một môn học để tạo đề thi trong danh sách các môn học ở bước trên, hệ thống hiển thị giao diện tạo đề thi cùng danh sách các lớp có thi môn học được chọn trong kỳ thi hiện tại
* Giáo viên nhập các thông tin của đề thi: thời gian làm bài, từng câu hỏi,các phương án lựa chọn câu trả lời (có tối đa 7 phương án chọn), đáp án cho từng câu hỏi, điểm của từng câu hỏi
* Sau đó giáo viên tiếp tục chọn đề thi này dành cho các lớp nào, có thể 1 hoặc một vài hoặc tất cả các lớp trong danh sách hiển thị ra.
* Giáo viên chọn lưu thông tin, hệ thống xác nhận yêu cầu, xác nhận tính hợp lệ của dữ liệu, thực hiện lưu thông tin, hiển thị thông báo cho giáo viên (Quá trình tạo đề thi cho một môn học có thể được lặp lại như trên cho các lớp khác nhau)
* Kết thúc Use Case
* **Luồng rẽ nhánh thứ nhất**
  + Giáo viên hủy yêu cầu tạo đề thi
  + Hệ thống chuyển sang giao diện chức năng của giáo viên
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ hai**
  + Các thông tin giáo viên nhập vào không hợp lệ
  + Hệ thống hiển thị thông báo lỗi đối với các trường dự liệu không hợp lệ
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ ba**
  + Hệ thống lưu thông tin không thành công do lỗi xử lý
  + Hiển thị thông báo lỗi

Kết thúc Use Case

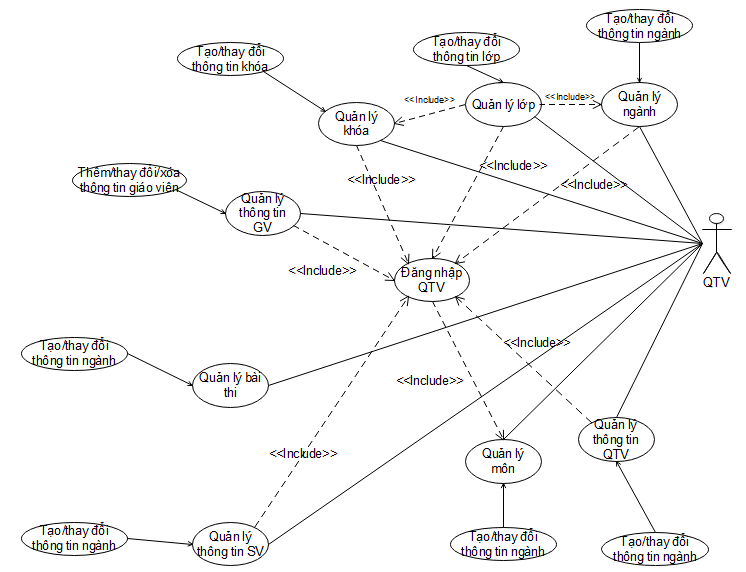
#### 2.3.2.3 Use Case Quản lý thông tin giáo viên

* + Tóm tắt: Đây là trường hợp sử dụng giáo viên xem và thay đổi thông tin cá nhân của mình
  + Tác nhân: Giáo viên
  + Liên quan: Use Case này phải sử dụng Use Case Đăng nhập GV
  + Luồng sự kiện.
* **Luồng sự kiện chính**
* Giáo viên chọn xem thông tin cá nhân
* Hệ thống hiển thị thông tin cá nhân của giáo viên
* Giáo viên có thể thay đổi một số thông tin: mật khẩu đăng nhập,các thông tin mô tả khác, nhập lại mật khẩu cũ
* Giáo viên chọn lưu thông tin thay đổi
* Hệ thống xác nhận yêu cầu, kiểm tra dữ liệu và lưu thông tin mới
* Hiển thị thông báo
* Kết thúc Use Case

**+ Luồng rẽ nhánh thứ nhất**

* + Giáo viên hủy yêu cầu thay đổi thông tin
  + Hệ thống hiển thị lại thông tin ban đầu của giáo viên
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ hai**
  + Mật khẩu cũ không hợp lệ
  + Hệ thống từ chối cập nhật
  + Hệ thống hiển thị lại thông tin đã nhập và yêu cầu nhập lại mật khẩu cũ
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ ba**
  + Hệ thống có lỗi trong quá trình xử lý
  + Hiển thị thông báo lỗi
  + Kết thúc Use Case

### 2.3.3 Các Use Case liên quan đến tác nhân Quản trị viên



Hình 2.4: Biểu đồ phân rã tác nhân QTV

#### 2.3.3.1 Use Case Đăng nhập vào quyền quản trị

* + Tóm tắt: Đây là trường hợp sử dụng quản trị viên đăng nhập vào hệ thống
  + Tác nhân: Quản trị viên
  + Liên quan: Không có các Use Case liên quan
  + Luồng sự kiện
* **Luồng sự kiện chính**
  + - Quản trị viên chọn đăng nhập trên giao diện quản lý của hệ thống thi trực tuyến
    - Hệ thống hiển thị hộp thoại đăng nhập
    - Quản trị viên nhập ID tài khoản và mật khẩu đã được cấp, chọn đăng nhập vào chức năng quản trị
    - Hệ thống xác nhận mật khẩu
    - Hiển thị giao diện chức năng của quản trị viên
    - Kết thúc Use Case
* **Luồng rẽ nhánh thứ nhất**
  + - Quản trị viên hủy yêu cầu đăng nhập
    - Hệ thống bỏ qua hộp thoại đăng nhập, hiển thị giao diện ban đầu của hệ thống
    - Kết thúc Use Case
* **Luồng rẽ nhánh thứ hai**
  + - ID tài khoản và mật khẩu nhập vào không hợp lệ
    - Hệ thống từ chối đăng nhập
    - Hiển thị thông báo và hộp thoại đăng nhập
    - Kết thúc Use Case
* **Luồng rẽ nhánh thứ ba**
  + Hệ thống có lỗi trong quá trình xử lý xác nhận tài khoản
  + Hiển thị thông báo lỗi
  + Kết thúc Use Case

#### 2.3.3.2 Use Case Tạo/Thay đổi thông tin khóa học

* + Tóm tắt: Đây là trường hợp sử dụng quản trị viên tạo hoặc thay đổi thông tin các khóa học
  + Tác nhân: Quản trị viên
  + Liên quan: Use Case này phải sử dụng Đăng nhập QTV
  + Luồng sự kiện
* **Luồng sự kiện chính**
* Trên giao diện quản lý khóa học, quản trị viên chọn thay đổi thông tin một khóa học nào đó hoặc tạo mới một khóa học.
  + Hệ thống hiển thị giao diện nhập thông tin khóa học. Nếu là chức năng thay đổi thông tin khóa học, hệ thống sẽ hiển thị thông tin của khóa học được chọn.
  + Quản trị viên nhập các thông tin: ID khóa học (nếu tạo mới), tên khóa học,sau đó chọn chức năng lưu thông tin.
  + Hệ thống xác nhận lại yêu cầu và dữ liệu vào.
  + Hệ thống lưu thông tin.
  + Hiển thị thông báo.
  + Kết thúc Use Case.
* **Luồng rẽ nhánh thứ nhất**
  + Quản trị viên hủy yêu cầu.
  + Hệ thống hiển thị giao diện quản lý khóa học.
  + Kết thúc Use Case.
* **Luồng rẽ nhánh thứ hai**
  + Thông tin nhập vào không hợp lệ, ID khóa học đã tồn tại (trong trường hợp tạo mới).
  + Hệ thống từ chối lưu thông tin, hiển thị thông báo.
  + Kết thúc Use Case.
* **Luồng rẽ nhánh thứ ba**
  + Hệ thống có lỗi trong quá trình xử lý.
  + Hiển thị thông báo.
  + Kết thúc Use Case.

#### 2.3.3.3 Use Case Tạo/Thay đổi thông tin lớp

* + Tóm tắt: Đây là trường hợp sử dụng quản trị viên tạo hoặc thay đổi thông tin lớp.
  + Tác nhân: Quản trị viên.
  + Liên quan: Use Case này phải sử dụng Đăng nhập QTV.
  + Luồng sự kiện.
* **Luồng sự kiện chính**
* Trên giao diện quản lý lớp, quản trị viên chọn thay đổi thông tin một lớp nào đó hoặc tạo mới một lớp.
  + Hệ thống hiển thị giao diện nhập thông tin lớp. Nếu là chức năng thay đổi thông tin lớp, hệ thống sẽ hiển thị thông tin của lớp được chọn.
  + Quản trị viên nhập các thông tin: Chọn khóa học của lớp, ID lớp (nếu tạo mới), tên lớp, chọn các môn học và giáo viên của lớp, sau đó chọn chức năng lưu thông tin.
  + Hệ thống xác nhận lại yêu cầu và dữ liệu vào.
  + Hệ thống lưu thông tin.
  + Hiển thị thông báo.
  + Kết thúc Use Case.
* **Luồng rẽ nhánh thứ nhất**
  + Quản trị viên hủy yêu cầu.
  + Hệ thống hiển thị giao diện quản lý lớp.
  + Kết thúc Use Case.
* **Luồng rẽ nhánh thứ hai**
  + Thông tin nhập vào không hợp lệ, ID lớp đã tồn tại (trong trường hợp tạo mới).
  + Hệ thống từ chối lưu thông tin, hiển thị thông báo.
  + Kết thúc Use Case.
* **Luồng rẽ nhánh thứ ba**
  + Hệ thống có lỗi trong quá trình xử lý.
  + Hiển thị thông báo.
  + Kết thúc Use Case.

#### 2.3.3.4 Use Case Tạo/Thay đổi thông tin môn học

* + Tóm tắt: Đây là trường hợp sử dụng quản trị viên tạo hoặc thay đổi thông tin môn học.
  + Tác nhân: Quản trị viên.
  + Liên quan: Use Case này phải sử dụng Đăng nhập QTV.
  + Luồng sự kiện.
* **Luồng sự kiện chính**
* Trên giao diện quản lý môn học, quản trị viên chọn thay đổi thông tin một môn học nào đó hoặc tạo mới một môn học.
  + Hệ thống hiển thị giao diện nhập thông tin môn học. Nếu là chức năng thay đổi thông tin môn học, hệ thống sẽ hiển thị thông tin của môn học được chọn.
  + Quản trị viên nhập các thông tin: ID môn học (nếu tạo mới), tên môn học, sau đó chọn chức năng lưu thông tin.
  + Hệ thống xác nhận lại yêu cầu và dữ liệu vào. Hệ thống lưu thông tin.
  + Hiển thị thông báo.
  + Kết thúc Use Case.
* **Luồng rẽ nhánh thứ nhất**
  + Quản trị viên hủy yêu cầu.
  + Hệ thống hiển thị giao diện quản lý môn học.
  + Kết thúc Use Case.
* **Luồng rẽ nhánh thứ hai**
  + Thông tin nhập vào không hợp lệ, ID môn học đã tồn tại (trong trường hợp tạo mới).
  + Hệ thống từ chối lưu thông tin, hiển thị thông báo.
  + Kết thúc Use Case.
* **Luồng rẽ nhánh thứ ba**
  + Hệ thống có lỗi trong quá trình xử lý.
  + Hiển thị thông báo.
  + Kết thúc Use Case.

#### 2.3.3.5 Use Case Tạo/Thay đổi thông tin giáo viên

* + Tóm tắt: Đây là trường hợp sử dụng quản trị viên tạo hoặc thay đổi thông tin một giáo viên.
  + Tác nhân: Quản trị viên.
  + Liên quan: Use Case này phải sử dụng Đăng nhập QTV.
  + Luồng sự kiện.
* **Luồng sự kiện chính**
  + Trên giao diện quản lý giáo viên, quản trị viên chọn thay đổi thông tin một giáo viên nào đó hoặc tạo mới thông tin một giáo viên. Hệ thống hiển thị giao diện nhập thông tin giáo viên. Nếu là chức năng thay đổi thông tin giáo viên, hệ thống sẽ hiển thị thông tin của giáo viên được chọn.
  + Quản trị viên nhập các thông tin: ID đăng nhập (nếu tạo mới), mật khẩu, các thông tin chi tiết khác, sau đó chọn chức năng lưu thông tin.
  + Hệ thống xác nhận lại yêu cầu và dữ liệu vào. Hệ thống lưu thông tin.
  + Hiển thị thông báo.
  + Kết thúc Use Case.
* **Luồng rẽ nhánh thứ nhất**
  + Quản trị viên hủy yêu cầu.
  + Hệ thống hiển thị giao diện quản lý giáo viên.
  + Kết thúc Use Case.
* **Luồng rẽ nhánh thứ hai**
  + Thông tin nhập vào không hợp lệ, ID đăng nhập đã tồn tại (trong trường hợp tạo mới).
  + Hệ thống từ chối lưu thông tin, hiển thị thông báo.
  + Kết thúc Use Case.
* **Luồng rẽ nhánh thứ ba**
  + Hệ thống có lỗi trong quá trình xử lý.
  + Hiển thị thông báo.
  + Kết thúc Use Case.

#### 2.3.3.6 Use Case Tạo/Thay đổi thông tin sinh viên

* + Tóm tắt: Đây là trường hợp sử dụng quản trị viên tạo hoặc thay đổi thông tin một học sinh.
  + Tác nhân: Quản trị viên.
  + Liên quan: Use Case này phải sử dụng Đăng nhập QTV.
  + Luồng sự kiện
* **Luồng sự kiện chính**
* Trên giao diện quản lý học sinh, quản trị viên chọn thay đổi thông tin một học sinh nào đó hoặc tạo mới thông tin một học sinh
  + Hệ thống hiển thị giao diện nhập thông tin học sinh. Nếu là chức năng thay đổi thông tin học sinh, hệ thống sẽ hiển thị thông tin của học sinh được chọn
  + Quản trị viên nhập các thông tin: ID đăng nhập (nếu tạo mới), mật khẩu, các thông tin chi tiết khác, lớp mà học sinh này học, sau đó chọn chức năng lưu thông tin
  + Hệ thống xác nhận lại yêu cầu và dữ liệu vào. Hệ thống lưu thông tin
  + Hiển thị thông báo
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ nhất**
  + Quản trị viên hủy yêu cầu
  + Hệ thống hiển thị giao diện quản lý học sinh
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ hai**
  + Thông tin nhập vào không hợp lệ, ID đăng nhập đã tồn tại (trong trườnghợp tạo mới)
  + Hệ thống từ chối lưu thông tin, hiển thị thông báo
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ ba**
  + Hệ thống có lỗi trong quá trình xử lý
  + Hiển thị thông báo
  + Kết thúc Use Case

#### 2.3.3.7 Use Case Xóa thông tin sinh viên

* + Tóm tắt: Đây là trường hợp sử dụng quản trị viên xóa thông tin một học sinh
  + Tác nhân: Quản trị viên
  + Liên quan: Use Case này phải sử dụng Use Case Đăng nhập QTV, Use Case Xóa điểm, Use Case Xóa bài thi
  + Luồng sự kiện
* **Luồng sự kiện chính**
  + Trên giao diện quản lý học sinh, quản trị viên chọn xóa thông tin một học sinh.
  + Hệ thống hiển thị giao diện thông tin học sinh.
  + Quản trị viên xác nhận lại yêu cầu
  + Hệ thống thực hiện xóa thông tin học sinh, xóa toàn bộ điểm, các bài thi đã có trong thời gian trước của học sinh
  + Hiển thị thông báo
* **Luồng rẽ nhánh thứ nhất**
  + Quản trị viên hủy yêu cầu
  + Hệ thống hiển thị giao diện quản lý học sinh
  + Kt thúc Use Case
* **Luồng rẽ nhánh thứ hai**
  + Hệ thống có lỗi trong quá trình xử lý
  + Hiển thị thông báo
  + Kết thúc Use Case

#### 2.3.3.8 Use Case Tạo/Thay đổi thông tin quản trị viên

* + Tóm tắt: Đây là trường hợp sử dụng quản trị viên tạo hoặc thay đổi thông tin một quản trị
  + Tác nhân: Quản trị viên
  + Liên quan: Use Case này phải sử dụng Đăng nhập QTV
  + Luồng sự kiện
* **Luồng sự kiện chính**
* Trên giao diện quản lý quản trị viên, quản trị viên chọn thay đổi thông tin một quản trị viên nào đó hoặc tạo mới thông tin một quản trị viên.
* Hệ thống hiển thị giao diện nhập thông tin quản trị viên. Nếu là chức năng thay đổi thông tin, hệ thống sẽ hiển thị thông tin của quản trị được chọn
* Quản trị viên nhập các thông tin: ID đăng nhập (nếu tạo mới), mật khẩu, các thông tin chi tiết khác, sau đó chọn chức năng lưu thông tin
  + Hệ thống xác nhận lại yêu cầu và dữ liệu vào. Hệ thống lưu thông tin
  + Hiển thị thông báo
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ nhất**
  + Quản trị viên hủy yêu cầu
  + Hệ thống hiển thị giao diện quản lý quản trị viên
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ hai**
  + Thông tin nhập vào không hợp lệ, ID đăng nhập đã tồn tại (trong trường hợp tạo mới)
  + Hệ thống từ chối lưu thông tin, hiển thị thông báo
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ ba**
  + Hệ thống có lỗi trong quá trình xử lý
  + Hiển thị thông báo
  + Kết thúc Use Case

#### 2.3.3.9 Use Case Tạo/Thay đổi thông tin kỳ thi

* + Tóm tắt: Đây là trường hợp sử dụng quản trị viên tạo hoặc thay đổi thông tin các kỳ thi
  + Tác nhân: Quản trị viên
  + Liên quan: Use Case này phải sử dụng Đăng nhập QTV
  + Luồng sự kiện
* **Luồng sự kiện chính**
* Trên giao diện quản lý các kỳ thi, quản trị viên chọn thay đổi thông tin một kỳ thi nào đó hoặc tạo mới một kỳ thi.
* Hệ thống hiển thị giao diện nhập thông tin kỳ thi. Nếu là chức năng thay đổi thông tin, hệ thống sẽ hiển thị thông tin của kỳ thi được chọn
* Quản trị viên nhập các thông tin: ID kỳ thi (nếu tạo mới), tên kỳ thi, các lớp sẽ tham gia vào kỳ thi, sau đó chọn chức năng lưu thông tin
  + Hệ thống xác nhận lại yêu cầu và dữ liệu vào. Hệ thống lưu thông tin
  + Hiển thị thông báo
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ nhất**
  + Quản trị viên hủy yêu cầu
  + Hệ thống hiển thị giao diện quản lý kỳ thi
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ hai**
  + Thông tin nhập vào không hợp lệ, ID kỳ thi đã tồn tại (trong trường hợp tạo mới)
  + Hệ thống từ chối lưu thông tin, hiển thị thông báo
  + Kết thúc Use Case
* **Luồng rẽ nhánh thứ ba**
  + Hệ thống có lỗi trong quá trình xử lý
  + Hiển thị thông báo
  + Kết thúc Use Case

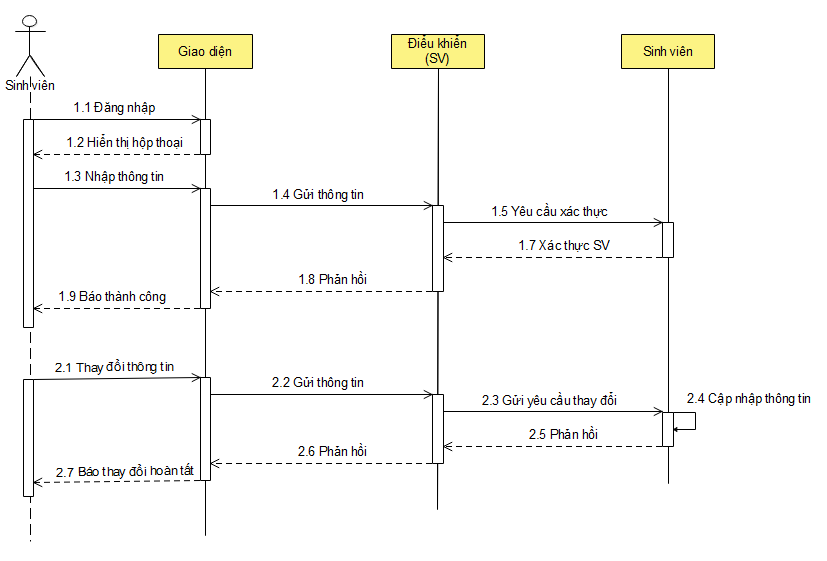
#### 2.3.3.10 Use case Thiết lập đề thi

* + Tóm tắt: Đây là trường hợp sử dụng quản trị viên thiết lập đề thi sắp tới cho một lớp nào đó
  + Tác nhân: Quản trị viên
  + Liên quan: Use Case này phải sử dụng Đăng nhập QTV
  + Luồng sự kiện
* **Luồng sự kiện chính**
* Trên giao diện quản lý các kỳ thi, quản trị viên chọn chức năng thiết lập thông tin đề thi.
* Hệ thống hiển thị giao diện quản lý thông tin đề thi
* Quản trị viên chọn một kỳ thi, chọn lớp, chọn đề thi theo môn học.
  + Hệ thống hiển thị các đề thi của môn học vừa được chọn theo từng lớp
  + Quản trị viên nhập các thông tin: ngày thi cho đề thi sau đó chọn chức năng lưu thông tin
  + Hệ thống xác nhận lại yêu cầu và dữ liệu vào. Hệ thống lưu thông tin
  + Hiển thị thông báo
  + Kết thúc Use Case
* **Luồng sự kiện rẽ nhánh**
  + Hệ thống có lỗi trong quá trình xử lý
  + Hiển thị thông báo
  + Kết thúc Use Case

## 2.4. Xây dựng biểu đồ Tuần tự của hệ thống

### 2.4.1 Các nhóm chức năng liên quan đến sinh viên

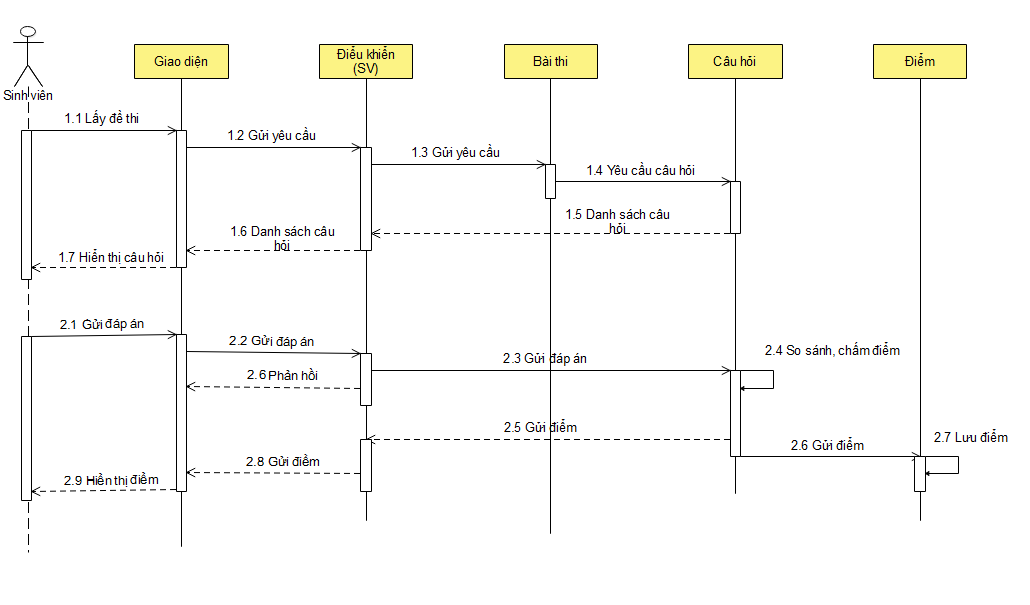
#### 2.4.1.1 Đăng nhập – quản lý thông tin



Hình 2.5: Biểu đồ tuần tự chức năng Đăng nhập – quản lý thông tin

- Biểu đồ mô tả quy trình hệ thống làm việc khi sinh viên đăng nhập, quản lý thông tin cá nhân.

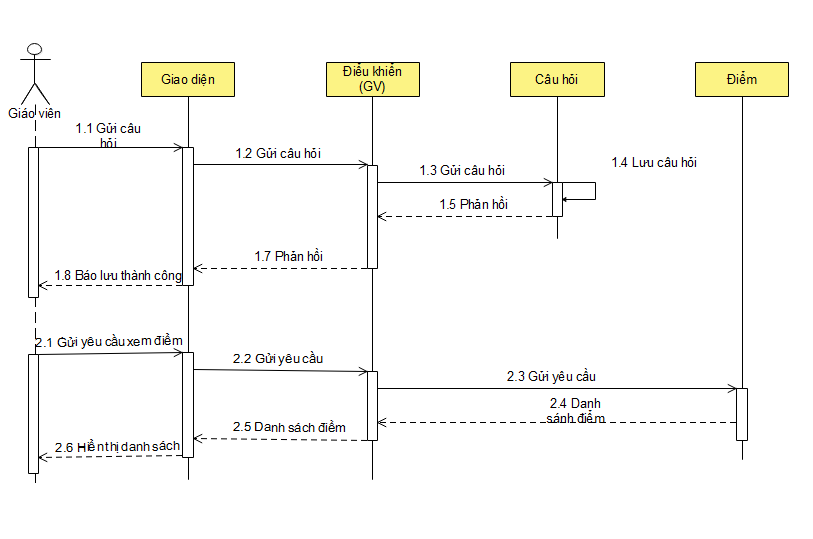
#### 2.4.1.2 Sinh viên làm bài thi



Hình 2.6: Biểu đồ tuần tự chức năng SV làm bài thi

- Biểu đồ mô tả quy trình hệ thống làm việc khi sinh viên làm bài thi trực tuyến.

### 2.4.2 Các nhóm chức năng liên quan đến giáo viên

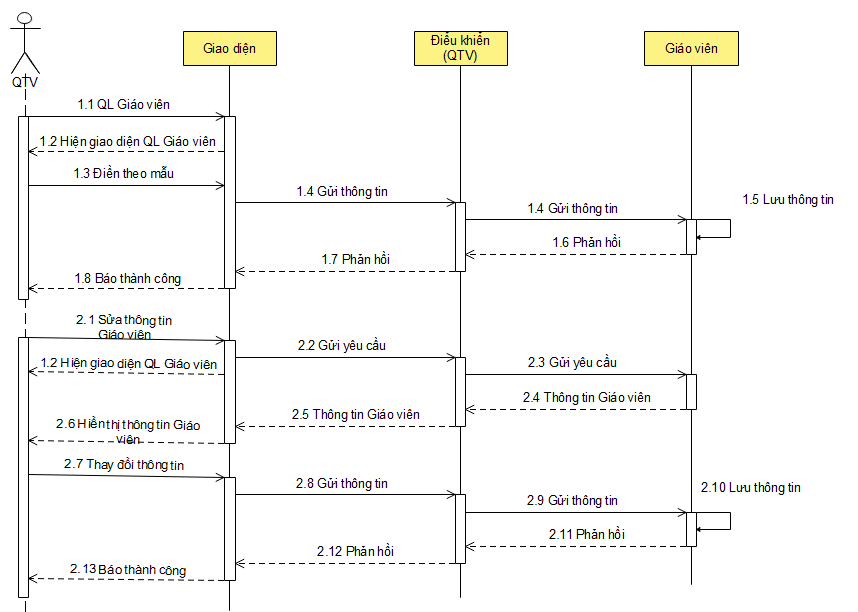


Hình 2.7: Biểu đồ tuần tự chức năng Xem điểm

- Biểu đồ mô tả quy trình hệ thống làm việc khi giáo viên xem điểm thi của các sinh viên

### 2.4.3 Các nhóm chức năng liên quan đến quản trị viên

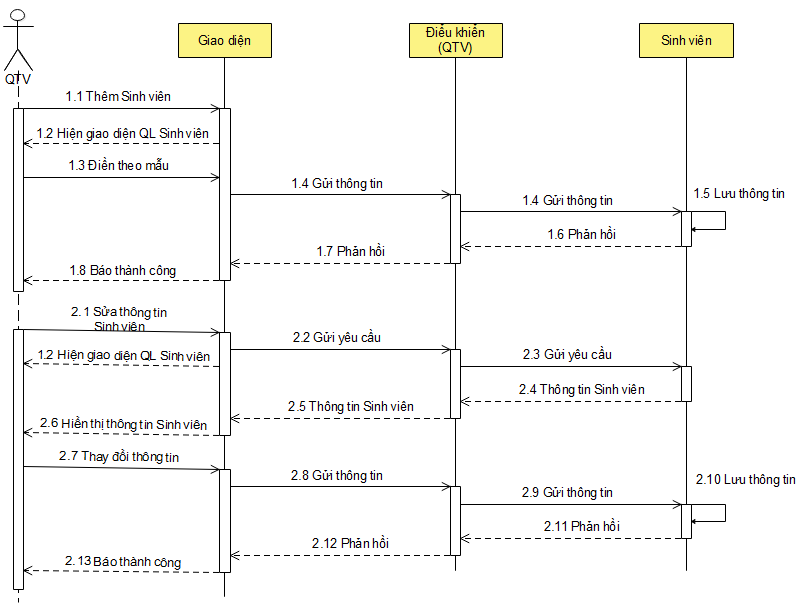
#### 2.4.3.1 Quản lý giáo viên



Hình 2.8: Biểu đồ tuần tự chức năng Quản lý GV

- Biểu đồ mô tả quy trình hệ thống làm việc khi quản trị viên thêm hoặc sửa thông tin giáo viên

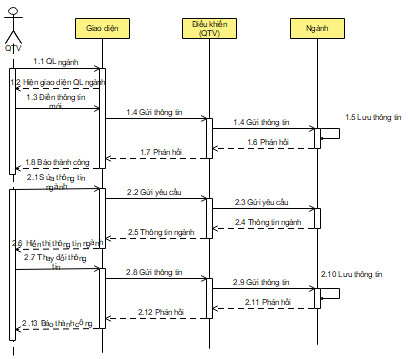
#### 2.4.3.2 Quản lý sinh viên



Hình 2.9: Biểu đồ tuần tự chức năng Quản lý SV

- Biểu đồ mô tả quy trình hệ thống làm việc khi quản trị viên thêm hoặc sửa thông tin sinh viên

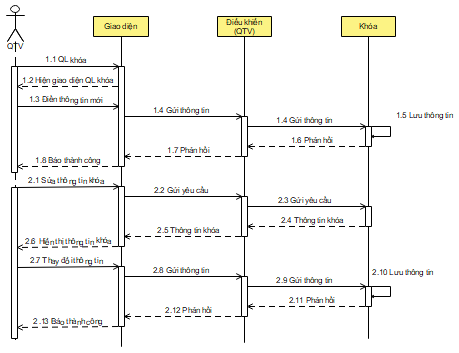
#### 2.4.3.3 Quản lý ngành



Hình 2.10: Biểu đồ tuần tự chức năng Quản lý ngành

- Biểu đồ mô tả quy trình hệ thống làm việc khi quản trị viên thêm hoặc sửa thông tin ngành học

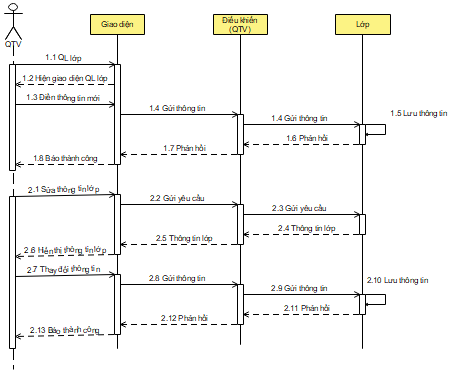
#### 2.4.3.4 Quản lý khóa



Hình 2.11: Biểu đồ tuần tự chức năng Quản lý khóa

- Biểu đồ mô tả quy trình hệ thống làm việc khi quản trị viên thêm hoặc sửa thông tin một khóa học

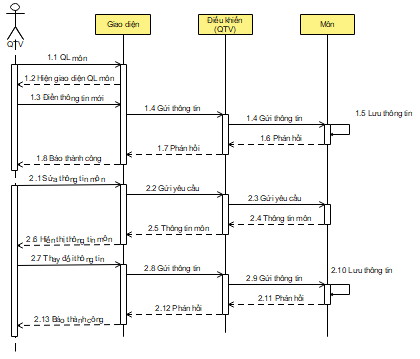
#### 2.4.3.5 Quản lý lớp



Hình 2.12: Biểu đồ tuần tự chức năng Quản lý lớp

- Biểu đồ mô tả quy trình hệ thống làm việc khi quản trị viên thêm hoặc sửa thông tin một lớp học

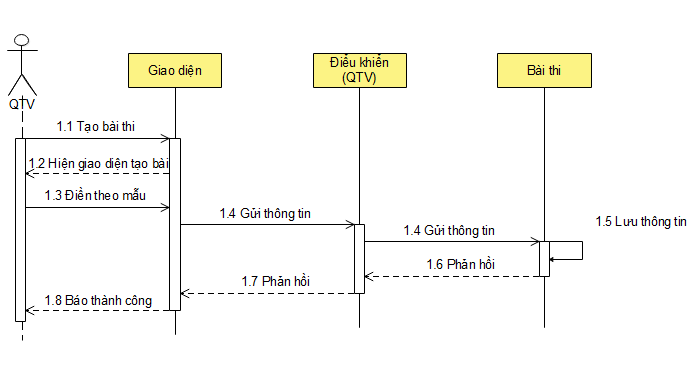
#### 2.4.3.6 Quản lý môn



Hình 2.13: Biểu đồ tuần tự chức năng Quản lý môn

- Biểu đồ mô tả quy trình hệ thống làm việc khi quản trị viên thêm hoặc sửa thông tin một môn học

#### 2.4.3.7 Quản lý bài thi

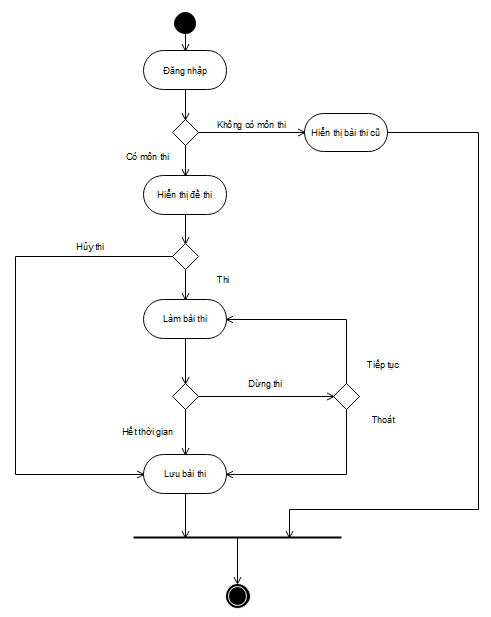


Hình 2.14: Biểu đồ tuần tự chức năng Quản lý bài thi

- Biểu đồ mô tả quy trình hệ thống làm việc khi quản trị viên thêm hoặc sửa thông tin một lkỳ thi

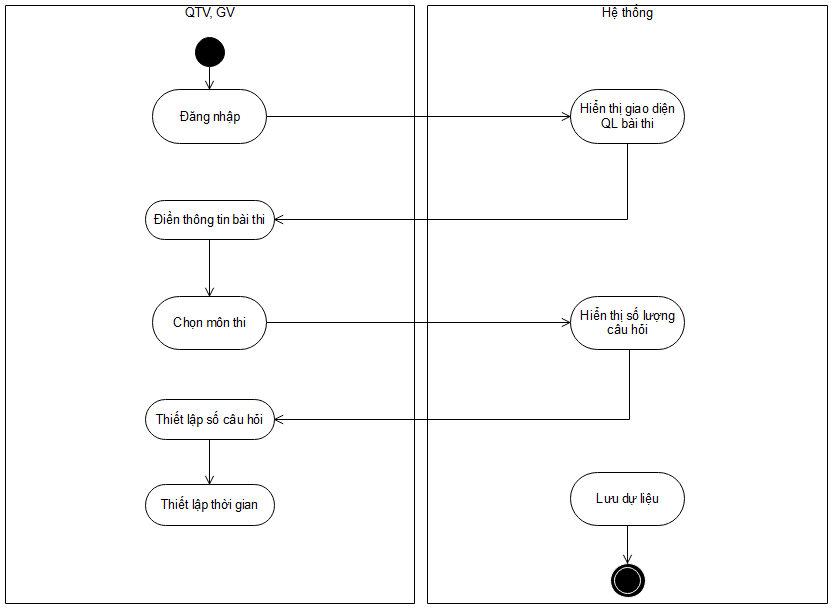
## 2.5 Biểu đồ hoạt động một số chức năng chính

### 2.5.1 Biểu đồ hoạt động chức năng làm bài thi



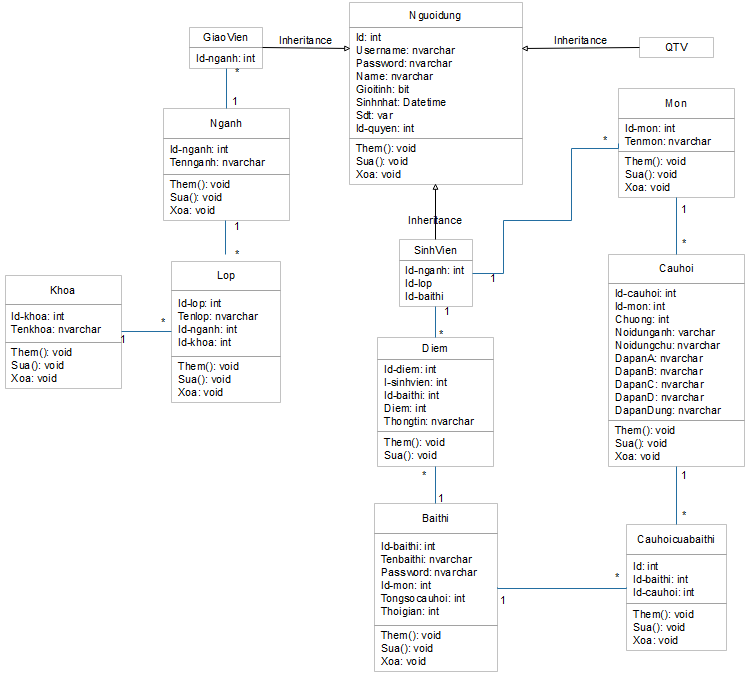
Hình 2.15: Biểu đồ hoạt động chức năng làm bài thi

### 2.5.2 Biểu đồ hoạt động chức năng thiết lập đề thi



Hình 2.16: Biểu đồ hoạt động chức năng thiết lập đề thi

## 2.6 Biểu đồ lớp



Hình 2.17: Biếu đồ Lớp

# THỰC NGHIỆM XÂY DỰNG HỆ THỐNG

## 3.1. Môi trường cài đặt

Việc lựa chọn công nghệ sao cho phù hợp với ứng dụng muốn phát triển là điều quan trọng. Với sự phát triển của công nghệ thông tin như hiện nay, nhiều công nghệ mới ra đời, xuất hiện nhiều ngôn ngữ mới để đáp ứng cho nhiều nhu cầu khác nhau thuộc các lĩnh vực khác nhau.

Với đề tài: “Nghiên cứu mô hình lập trình MVC và xây dựng Website trắc nghiệm online**”** em đã chọn sử dụng công cụ chính là Microsoft Visual Studio 2017 và SQL Server 2014.

### 3.1.1 Giới thiệu hệ quản trị SQL Server

SQL Server chính là một hệ quản trị dữ liệu quan hệ (Relational Database Management System – RDBMS) sử dụng câu lệnh SQL để trao đổi dữ liệu giữa máy cài SQL Server và máy Client. RDBMS gồm có: databases, datase engine và các chương trình ứng dụng dùng để quản lý các bộ phận trong RDBMS và những dữ liệu khác.

**Các thành phần cơ bản trong SQL Server**

Các thành cơ bản trong SQL Server gồm có: Reporting Services, Database Engine, Integration Services, Notification Services, Full Text Search Service,… Tất cả kết hợp với nhau tạo thành một giải pháp hoàn chỉnh giúp cho việc phân tích và lưu trữ dữ liệu trở nên dễ dàng hơn.

+ **Database Engine**: Đây là một engine có khả năng chứa dữ liệu ở các quy mô dưới dạng support và table. Ngoài ra, nó còn có khả năng tự điều chỉnh ví dụ: trả lại tài nguyên cho hệ điều hành khi một user log off và sử dụng thêm các tài nguyên của máy khi cần.

+ **Integration Services**: Là tập hợp các đối tượng lập trình và các công cụ đồ họa cho việc sao chép, di chuyển và chuyển đổi dữ liệu. Khi bạn làm việc trong một công ty lớn thì dữ liệu được lưu trữ ở nhiều nơi khác nhau như được chứa trong: Oracle, SQL Server, DB2, Microsoft Access,… và bạn chắc chắn sẽ có nhu cầu di chuyển dữ liệu giữa các server này. Ngoài ra, bạn còn muốn định dạng dữ liệu trước khi lưu vào database. Chắc chắn Integration Services sẽ giúp bạn giải quyết được công việc này dễ dàng.

+ **Analysis Services**: Đây là một dịch vụ phân tích dữ liệu rất hay của Microsoft. Dữ liệu khi được lưu trữ vào trong database mà bạn không thể lấy được những thông tin bổ ích thì coi như không có ý nghĩa gì. Chính vì thế, công cụ này ra đời giúp bạn trong việc phân tích dữ liệu một cách hiệu quả và dễ dàng bằng cách dùng kỹ thuật khai thác dữ liệu – datamining và khái niệm hình khối nhiều chiều – multi dimendion cubes.

+ **Notification Services**: Dịch vụ thông báo này là nền tảng cho sự phát triển và triển khai các ứng dụng soạn và gửi thông báo. Ngoài ra, dịch vụ này còn có chức năng gửi thông báo theo dịch thời đến hàng ngàn người đăng ký sử dụng trên nhiều loại thiết bị khác nhau.

+ **Reporting Services**: Là một công cụ tạo, quản lý và triển khai báo cáo bao gồm: server và client. Ngoài ra, nó còn là nền tảng cho việc phát triển và xây dựng các ứng dụng báo cáo.

+ **Full Text Search Service**: Là một thành phần đặc biệt trong việc truy vấn và đánh chỉ mục dữ liệu văn bản không cấu trúc được lưu trữ trong các cơ sở dữ liệu SQL Server.

+ **Service Broker**: Là một môi trường lập trình cho việc tạo ra các ứng dụng trong việc nhảy qua các Instance.

**Tại sao lại sử dụng SQL trong thiết kế web?**

SQL Server không phải là một hệ quản trị cơ sở dữ liệu độc lập mà nó chỉ là một thành phần với vai trò ngôn ngữ là công cụ giao tiếp giữa hệ cơ sở dữ liệu và người dùng. Chính vì thế nó được sử dụng trong các dịch vụ thiết kế web đẹp với chức năng giao tiếp với người dùng với các vai trò sau:

SQL là một ngôn ngữ đòi hỏi có tính tương tác cao: Người dùng có thể dễ dàng trao đổi với các tiện ích thông qua các câu lệnh của SQL đến cơ sở dữ liệu và nhận kết quả từ cơ sở dữ liệu.

SQL là một ngôn ngữ lập trình cơ sở dữ liệu: Các lập trình viên có thể xây dựng các chương trình ứng dụng giao tiếp với cơ sở dữ liệu bằng cách nhúng các câu lệnh SQL vào trong ngôn ngữ lập trình.

SQL là một ngôn ngữ lập trình quản trị cơ sở dữ liệu: Người quản trị cơ sở dữ liệu có thề quản lý, định nghĩa và điều khiển truy cập cơ sở dữ liệu thông qua SQL.

SQL là một ngôn ngữ lập trình cho các hệ thống chủ khách: SQL được sử dụng như là một công cụ giao tiếp với các trình ứng dụng trong hệ thống cơ sở dữ liệu khách chủ.

SQL là ngôn ngữ truy cập dữ liệu trên Internet: SQL được sử dụng với vai trò tương tác với dữ liệu trong hầu hết các máy chủ web và máy chủ Internet.

SQL là ngôn ngữ cơ sở dữ liệu phân tán: Với vai trò giao tiếp với các hệ thống trên mạng, gửi và nhận các yêu cầu truy xuất dữ liệu với nhau.

Nhìn chung SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ được sử dụng trong các chức năng tương tác giữa người dùng và dữ liệu. Chính vì thế, nó được sử dụng trong các dịch vụ thiết kế web rẻ và chuyên nghiệp.

### 3.1.2. Giới thiệu về Visual studio 2017

Visual Studio 2017 hỗ trợ các ngôn ngữ lập trình hướng đối tượng mới như: C# , VB.Net, J# và ngôn ngữ lập trình "siêu mạnh" là C++.Net, đều có những cải tiến đáng kể. Visual Studio 2017 là bộ công cụ phát triển phần mềm tích hợp mạnh mẽ với những tính năng hấp dẫn:

***Thiết kế giao diện, hỗ trợ viết mã (coding)***

Nếu đã từng sử dụng Visual Studio.Net chắc hẳn bạn rất thích thú với sự hỗ trợ viết mã IntelliSense. IntelliSense không những giúp viết mã nhanh hơn, đúng hơn mà còn giúp những lập trình viên mới làm quen với Visual Studio.Net nhanh chóng hơn. Đặc biệt, VS 2017 còn có khả năng sử dụng lại các đoạn mã mẫu. Không những thế, VS 2017 cho phép tạo ứng dụng với giao diện giống với giao diện của Outlook đầy hấp dẫn.

***Xây dựng ứng dụng Web nhanh hơn***

VS 2017 đã đưa vào công cụ phát triển Web mới là Visual Web Developer. Công cụ này cho phép tạo ra các ứng dụng Web được viết bằng ASP.NET 2.0. Với Visual Web Developer, có khoảng 50 điều khiển Web mới được tích hợp giúp phát triển ứng dụng Web nhanh và đơn giản hơn. Cải tiến lớn trong ASP.NET 2.0 là hỗ trợ tạo Master Page và cho phép các trang khác kế thừa lại Master Page. Một ưu điểm lớn nữa của Visual Web Developer là dễ dàng tạo một dự án ứng dụng Web (Web project) rất đơn giản.

Trước đây, khi tạo một dự án Web, bạn phải xác định một thư mục ảo (vitual directory) trong máy chủ Web IIS (Internet Information Services) và khi sao chép dự án này sang máy khác thì thật "mệt mỏi". Nhưng với VS 2017, tạo ứng dụng Web không cần những thao tác trên, và thực sự đơn giản như tạo một ứng dụng trên Windows bình thường.

**Tự động gợi ý các gói của NuGet**

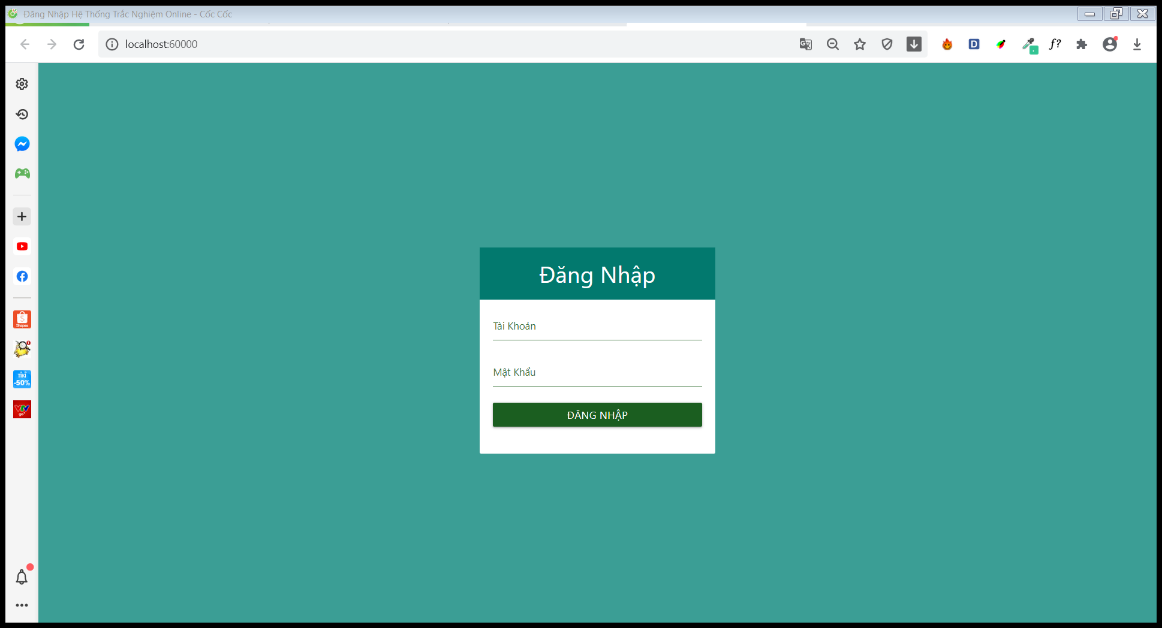
Khi chúng ta viết code, đôi khi chúng ta thường xuyên sử dụng các thư viện của hệ thống. Các thư viện này có thể đã được cài đặt bởi Nuget hoặc chưa được cài đặt.

Trong trường hợp gói thư viện chưa được cài đặt chúng ta có thể thực hiện Search trên Internet để chúng ta biết chúng ta cần cài đặt gói thư viện nào vào trong chương trình, ví dụ như khi chúng ta sử dụng lớp **DbContext**thì chúng ta sẽ phải biết chúng ta cần phải cài đặt gói thư viện gì vào hệ thống.

Trong bộ Visual Studio 2017 cho phép hệ thống tự động gợi ý các gói mở rộng thông qua việc sử dụng tên class, hoặc interface. Để thực hiện việc này chúng ta vào Tools > Options > Text Editor > C# > Advanced.

## 3.2 Một số giao diện của Website

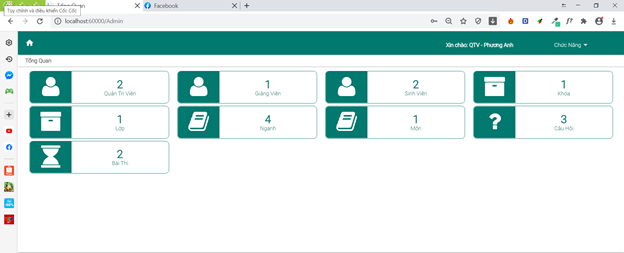
### 3.2.1 Giao diện đăng nhập



Hình 3.1: Giao diện đăng nhập

Mô tả: Giao diện đăng nhập chung cho người dùng. Sau khi đăng nhập, người dùng có thể thực hiện các chức năng mới quyền tương ứng của mình.

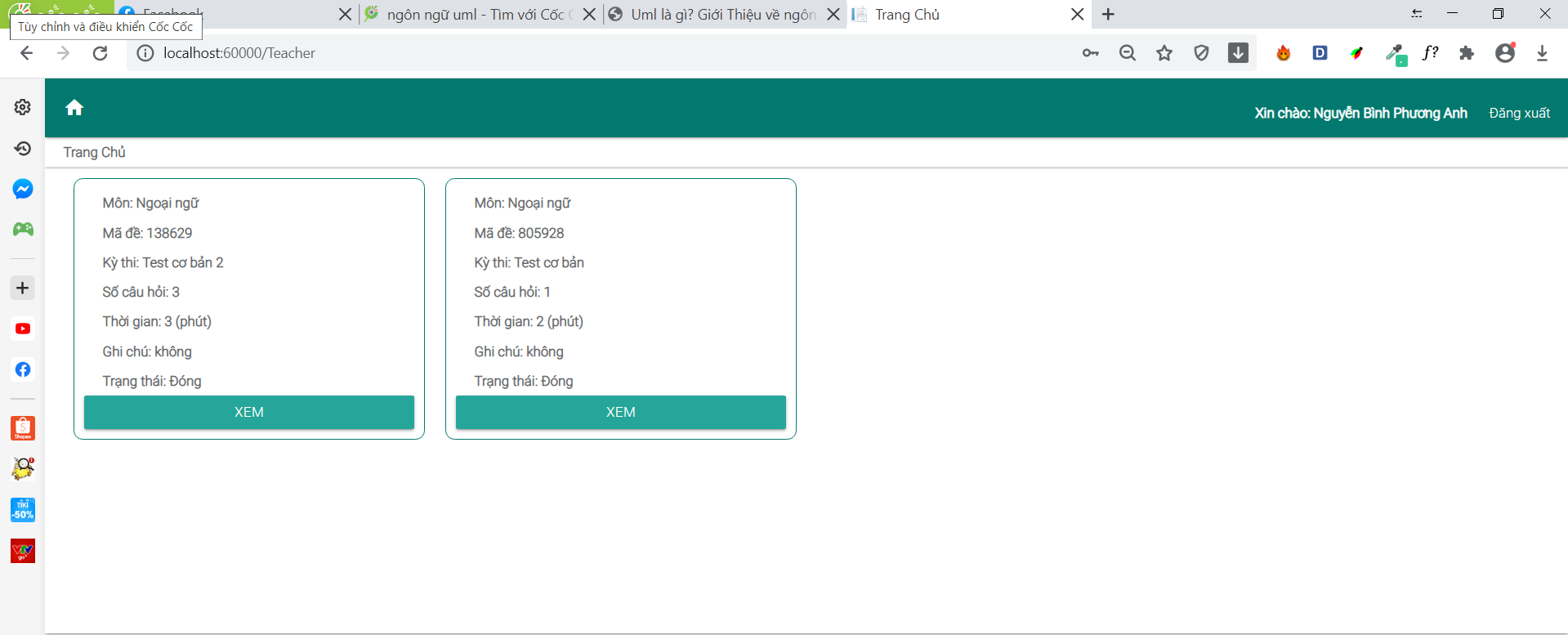
### 3.2.2 Giao diện của Quản trị viên



Hình 3.2: Giao diện của QTV

Mô tả: Giao diện chủ cho quản trị viên hiển thị các chức năng quản lý về các yếu tố trong toàn hệ thông như người dùng, ngành, môn, lớp, câu hỏi, bài thi ,… Từ đây quản trị viên truy cập vào để thực hiện các chức năng quản lý cụ thể

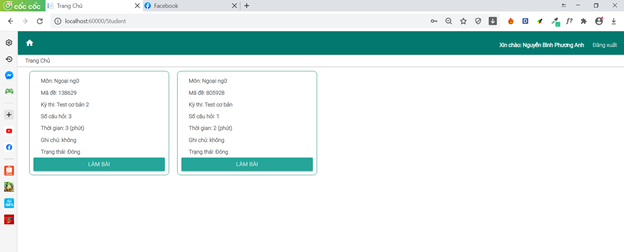
### 3.2.3 Giao diện của Giáo viên



Hình 3.3: Giao diện của GV

Mô tả: Giao diện của giáo viên cho phép giao viên xem kết quả các bài thi,xuất kết quả thi ra file excel, thực hiện quả lý thông tin cá nhân. Ngoài ra còn cho phép giao viên thực hiện chức năng quản lý câu hỏi và bài thi cho học sinh. Giáo viên có thể nhập câu hỏi thủ công hoặc nhập bằng file excel (theo mẫu file có sẵn).

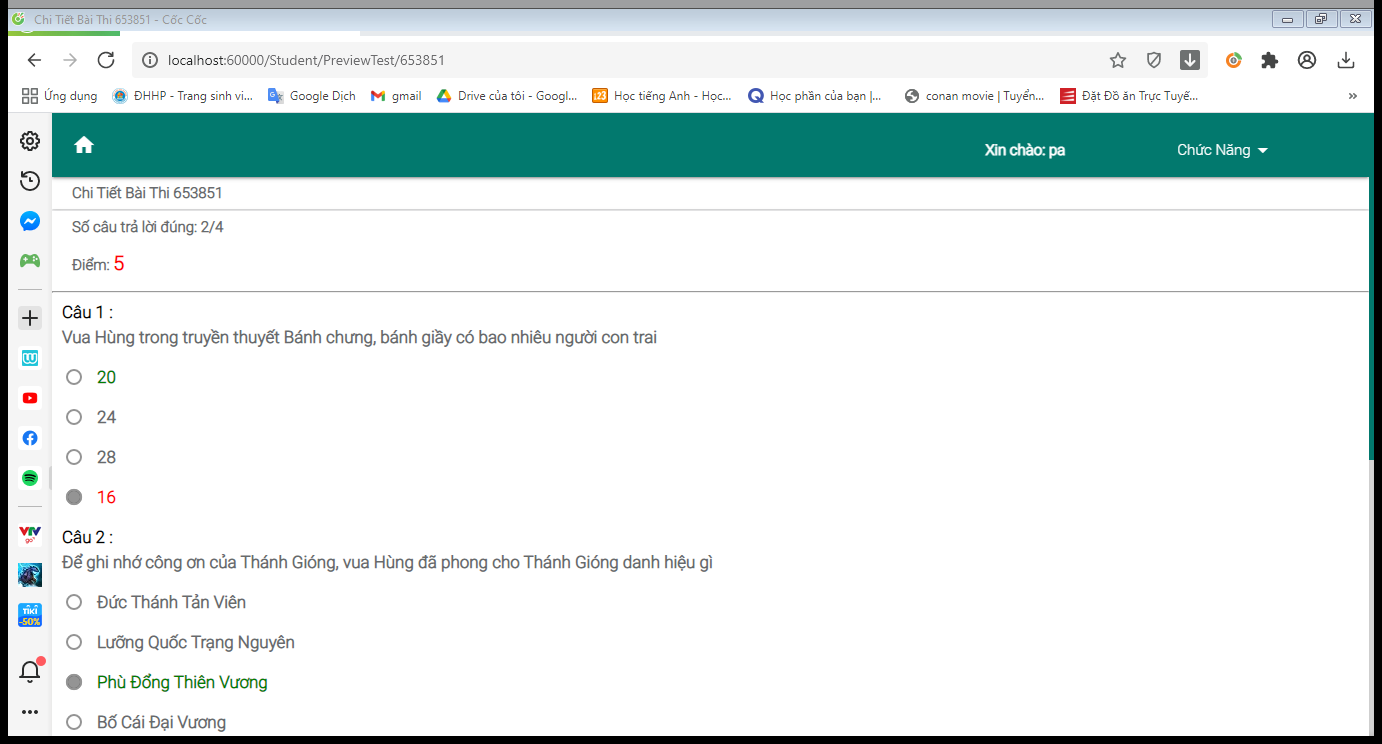
### 3.2.4 Giao diện của Sinh viên



Hình 3.4: Giao diện của SV

Mô tả: Giao diện chính của học sinh hiển thi, học sinh có thể chọn bài thi đã làm để xem lại kết quả, hoặc chọn làm bài thi mới, ngoài ra cho phép học sinh quản lý thông tin cá nhân.

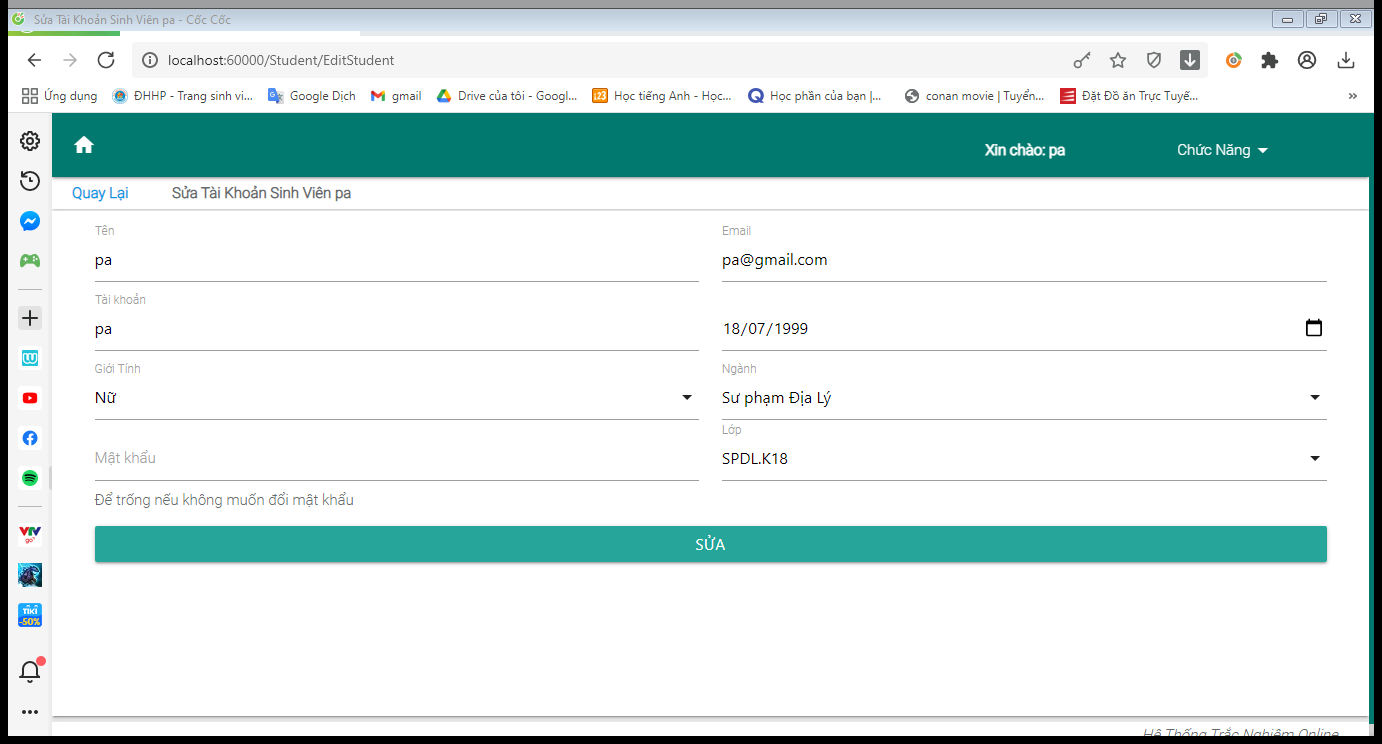
### 3.2.5 Giao diện Làm bài thi



Hình 3.5: Giao diện Làm bài thi

Mô tả: Giao diện làm bài thi đối với học sinh hiển thị các câu hỏi ngẫu nhiên trong bộ câu hỏi từ cơ sở dữ liệu. Mỗi câu hỏi có bốn đáp án, trong đó có một đáp án đúng, học sinh sau khi nộp bài sẽ được tính điểm, đáp án chọn sai và đúng cũng sẽ được hiển thị.

### 3.2.6 Giao diện quản lý thông tin cá nhân



Hình 3.6: Giao diện quản lý thông tin cá nhân

Mô tả: Giao diện quản lý thông tin các nhân cho phép người dùng quản lý thông tin của mình, hoặc thay đổi mật khẩu của tài khoản.

# KẾT LUẬN

Sau một thời gian tìm hiểu thông tin và các kiến thức cần có trong việc thiết kế website học tập online, cụ thể là Website kiểm tra trắc nghiệm online, em đã hoàn thiện sơ bộ đề tài: “Nghiên cứu mô hình lập trình MVC và xây dựng Website trắc nghiệm online”. Trong đề tài này, vận dụng các kiến thức căn bản đã tìm hiểu về Visual Studio 2017, SQL Server, công nghệ ASP.NET MVC cùng các thông tin thực tế tìm hiểu được, em đã hoàn thành được những chức năng sau:

* Xây dựng giao diện Website thân thiện với người dùng
* Đăng nhập cho quản trị viên, giáo viên, sinh viên
* Thực hiện các chức năng quản lý của quản trị viên như: quản lý sinh viên, quản lý giáo viên, quản lý ngành, môn …
* Thực hiện các chức năng của giáo viên như xem điểm
* Thực hiện quản lý câu hỏi và bài thi cho giáo viên
* Thực hiện xuất dữ liệu về điểm bài thi ra file excel
* Thực hiện các chức năng của sinh viên như làm bài thi, xem điểm thi
* Thực hiện thêm câu hỏi bằng file excel theo biểu mẫu có sẵn

**Hướng phát triển**

Trong thời gian tới, em sẽ cố gắng khắc phục những hạn chế trên để tiếp tục hoàn thiện đề tài mà mở rộng một số tính năng. Định hướng phát triển trong thời gian tới là bổ sung thêm các chức năng:

* Tối ưu các chức năng của quản trị viên
* Phân loại câu hỏi cho từng ngành, khóa học
* Thực hiện tính lực học của học sinh qua các bài thi

# TÀI LIỆU THAM KHẢO

[1] Lý thuyết về ASP.Net “[http//:dot.net.vn/](http://dot.net.vn/?fbclid=IwAR38omA_4IMF-EHgjL0yZlSsPYWb0eMlX3Ym1y1r6HEfDkSvHgHCtcV2-hM)” và Lê Hoàng Dũng, Tổng quan về ASP.NET MVC, Nhà xuất bản Khoa học và kỹ thuật, 2005.

[2] S.b.c.v.t.H. Nội, Giáo trình HTML và thiết kế website, Hà Nội, 2005.

[3] H. V. Đức, Giáo trình nhập môn UML, Thành Phố Hồ Chí Minh: Nhà Xuất Bản Lao Động, 2004.

[4] D. Q. Thiện, Sổ tay kỹ thuật Visual C#, Thành Phố Hồ Chí Minh, 2006.

[5] N. D. Phương, Bài giảng Kỹ thuật lập trình, Hồ Chí Minh: Nhà xuất bản Điện Biên, 2003.