# **CHƯƠNG 1. CƠ SỞ LÝ THUYẾT**

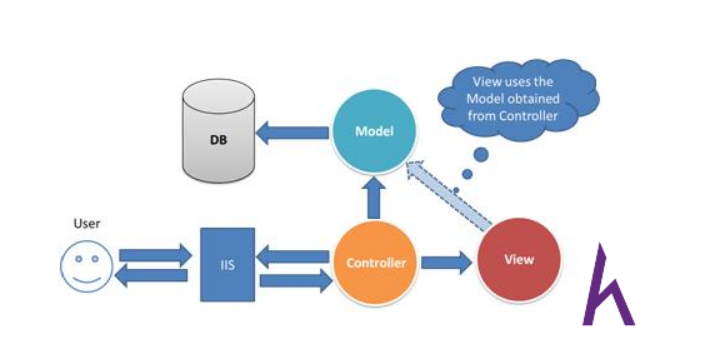
## **Giới thiệu về ASP.NET Core**

### 1.1.1. Khái niệm

ASP.NET Core là một phiên bản mới của ASP.NET, được phát hành bởi Microsoft và là một mã nguồn mở trên GitHub. ASP.NET Core được sử dụng để phát triển khuôn khổ website và có thể thích ứng với nhiều trình duyệt khác nhau như Windows, Mac hoặc Linux kể cả trên nền tảng MVC.

ASP.NET Core được thiết kế để tối ưu development framework cho những ứng dụng cái mà được chạy on-promise hay được triển khai trên đám mây. ASP.NET Core bao gồm các thành phần theo hướng module với mục đích tối thiểu tài nguyên và tiết kiệm chi phí khi phát triển. Đồng thời, ASP.NET Core cũng là một mã nguồn mở, một xu thế mà các ngôn ngữ lập trình hiện nay hướng đến.

### 1.2.2. Cơ chế hoạt động của ASP.Net Core MVC

****

Hình 1.1 Cơ chế hoạt động ASP.NET Core MVC

User gửi một yêu cầu tới server bằng cách truyền vào 1 URL trong browser.

Yêu cầu đó được gửi tới Controller đầu tiên, Controller sẽ xử lý yêu cầu, nếu yêu cầu cần truy xuất dữ liệu thì Controller sẽ chuyển qua tầng Model.

Tại tầng Model, dữ liệu được truy xuất từ database và sau đó truyền qua View thông qua Controller.

Controller sẽ giúp dữ liệu được chuyển từ Model qua View.

View là tầng cuối cùng giao tiếp với User, mọi dữ liệu sẽ được hiển thị cho. User thông qua tầng View.

### 1.2.3. Ưu điểm của ASP.NET Core

* Là khung mẫu và gọn gàng hơn kiến trúc nhiều tầng.
* Là một khung công tác với mã nguồn mở.
* Dễ dàng tạo ứng dụng ASP.NET đa nền tảng trên Windows, Mac và Linux.
* Cấu hình là môi trường sẵn sàng cho đám mây
* Khả năng lưu trữ trên: IIS, HTTP.sys, Kestrel, Nginx, Apache và Docker.

## **Giới thiệu ngôn ngữ C#**

C# là một ngôn ngữ lập trình mạnh mẽ và đa mục đích, được phát triển bởi Microsoft vào những năm 2000. C# là một phần của nền tảng phát triển phần mềm .NET Framework, và nó được sử dụng rộng rãi để xây dựng ứng dụng từ các ứng dụng máy tính đơn giản cho đến các ứng dụng di động, ứng dụng web và hệ thống phần mềm phức tạp.

Dưới đây là một số đặc điểm chính của ngôn ngữ C#:

*Tính đa nền tảng*: C# là một ngôn ngữ đa nền tảng, có thể chạy trên nhiều hệ điều hành như Windows, macOS và Linux. Điều này cho phép nhà phát triển viết mã một lần và chạy ứng dụng trên nhiều nền tảng khác nhau, giảm thiểu công sức phát triển và tăng tính linh hoạt.

*Tính dễ học và dễ sử dụng*: C# có cú pháp gần gũi với ngôn ngữ C++ và Java, điều này làm cho việc học và chuyển đổi từ các ngôn ngữ khác sang C# trở nên dễ dàng. Nó hỗ trợ mô hình lập trình hướng đối tượng (OOP), cho phép nhà phát triển xây dựng các đối tượng có tính chất, phương thức và sự tương tác giữa chúng.

*Công cụ phát triển tích hợp*: C# được tích hợp chặt chẽ với Visual Studio, một môi trường phát triển tích hợp (IDE) mạnh mẽ từ Microsoft. Visual Studio cung cấp các công cụ hỗ trợ mạnh mẽ như gỡ lỗi, thông báo lỗi cú pháp, hoàn thành mã tự động và nhiều tính năng khác để tăng hiệu suất và sức sáng tạo của nhà phát triển.

*Hỗ trợ rộng rãi cho các loại ứng dụng*: C# có thể được sử dụng để phát triển rất nhiều loại ứng dụng, bao gồm ứng dụng máy tính, ứng dụng di động (qua Xamarin), ứng dụng web (qua ASP.NET và Blazor), ứng dụng trò chơi và nhiều hơn nữa. Điều này cho phép nhà phát triển tận dụng sức mạnh của C# để xây dựng các ứng dụng đa dạng và đáp ứng nhu cầu của nhiều lĩnh vực khác nhau.

*Hệ sinh thái mạnh mẽ*: C# có một cộng đồng lập trình viên đông đảo và hệ sinh thái phong phú. Có rất nhiều tài liệu, sách, diễn đàn và nguồn học trực tuyến để hỗ trợ nhà phát triển trong việc học và phát triển ứng dụng bằng C#. Microsoft và cộng đồng cũng cung cấp các thư viện và framework mạnh mẽ như Entity Framework, ASP.NET Core, Xamarin và nhiều công nghệ khác để giúp nhà phát triển nhanh chóng xây dựngcác ứng dụng và giảm thiểu công việc lặp lại.

*Tính bảo mật và kiểm soát lỗi*: C# cung cấp các tính năng bảo mật tích hợp như kiểm soát truy cập, quản lý bộ nhớ tự động và xử lý ngoại lệ. Điều này giúp ngăn chặn các lỗ hổng bảo mật và giảm rủi ro lỗi trong quá trình phát triển và triển khai ứng dụng.

*Hỗ trợ cho phát triển đội ngũ*: C# được thiết kế để hỗ trợ phát triển đội ngũ và làm việc nhóm. Nó cung cấp các công cụ như quản lý mã nguồn, hệ thống kiểm soát phiên bản và tích hợp liên kết dữ liệu, giúp các nhà phát triển làm việc cùng nhau và duy trì mã nguồn dễ dàng.

*Tương thích và tích hợp*: C# tương thích tốt với các ngôn ngữ và công nghệ khác trong hệ sinh thái .NET, bao gồm VB.NET, F#, ASP.NET và nhiều hơn nữa. Nó cũng có khả năng tích hợp với các công nghệ khác như SQL Server, Azure và các dịch vụ web khác, tạo điều kiện cho việc phát triển ứng dụng phức tạp và kết nối với các hệ thống khác.

*Hỗ trợ và cập nhật liên tục*: C# được Microsoft hỗ trợ và phát triển liên tục. Cập nhật và phiên bản mới được phát hành thường xuyên để cải thiện hiệu suất, tăng tính năng và sửa các lỗi. Điều này đảm bảo rằng ngôn ngữ C# luôn giữ được tính tương thích và đáp ứng các xu hướng công nghệ mới.

Các phiên bản của ngôn ngữ C#:

C# 1.0 (2002): Đây là phiên bản đầu tiên của ngôn ngữ C#. Nó được giới thiệu cùng với .NET Framework 1.0. C# 1.0 cung cấp các tính năng cơ bản như lớp, đối tượng, kế thừa, giao diện, các kiểu dữ liệu cơ bản, xử lý ngoại lệ, tham số hóa kiểu và quản lý bộ nhớ tự động.

C# 2.0 (2005): Phiên bản này được phát hành cùng với .NET Framework 2.0 và Visual Studio 2005. C# 2.0 bổ sung nhiều tính năng mới như kiểu đối tượng tùy chỉnh, phạm vi (partial types), phạm vi tùy chỉnh (nullable types), phạm vi tùy chỉnh (anonymous methods), kiểu dữ liệu var, phạm vi mở rộng (extension methods) và lớp generic.

C# 3.0 (2007): Được phát hành cùng với .NET Framework 3.5 và Visual Studio 2008, C# 3.0 mang đến một số tính năng mới quan trọng như cú pháp LINQ (Language Integrated Query), biểu thức lambda, khởi tạo đối tượng tự động (automatic properties), khởi tạo đối tượng kiểu vô danh (anonymous types) và phạm vi mở rộng LINQ (LINQ extension methods).

C# 4.0 (2010): Phiên bản này được phát hành cùng với .NET Framework 4.0 và Visual Studio 2010. C# 4.0 bổ sung các tính năng như đặc tả tên tham số tùy chọn (optional parameter), tham số tùy chỉnh (named parameter), kiểu động (dynamic type), kiểu phạm vi động (covariant return type) và phạm vi nghịch đảo (contravariant type parameter).

C# 5.0 (2012): Được phát hành cùng với .NET Framework 4.5 và Visual Studio 2012, C# 5.0 đưa ra các tính năng như phương thức bất đồng bộ (asynchronous methods), từ khóa await/async, khóa đè (caller info attribute) và biểu thức lambda với các tham số tùy chọn.

C# 6.0 (2015): Phiên bản này được phát hành cùng với .NET Framework 4.6 và Visual Studio 2015. C# 6.0 mang đến các cải tiến như biểu thức null-conditional (null-conditional expressions), chuỗi kiểu (string interpolation), khai báo using tĩnh (static using directives), khai báo tên tùy chỉnh (custom name declarations), khởi tạo chỉ mục tĩnh (static index initializers) và các tính năng khác.

C# 7.0 (2017): Phiên bản này được phát hành cùng với .NET Core 2.0 và Visual Studio 2017. C# 7.0 bổ sung nhiều tính năng mới như tuple, mẫu khớp (pattern matching), phương thức mở rộng chỉ mục (indexers), tham số khởi tạo (ref locals and returns), bắt ngoại lệ (throw expressions) và các cải tiến khác.

C# 8.0 (2019): Được phát hành cùng với .NET Core 3.0 và Visual Studio 2019, C# 8.0 giới thiệu các tính năng như: danh sách không hoàn chỉnh vì giới hạn độ dài. Để biết thêm thông tin chi tiết về các phiên bản và lịch sử phát triển của ngôn ngữ C#, bạn có thể tham khảo tài liệu chính thức từ Microsoft hoặc các nguồn tài liệu lập trình C# khác.

Tóm lại, C# là một ngôn ngữ lập trình mạnh mẽ, đa nền tảng và dễ học. Nó cung cấp các công cụ và hệ sinh thái phong phú để phát triển ứng dụng từ đơn giản đến phức tạp, đáp ứng nhu cầu của nhiều lĩnh vực khác nhau. Với tính linh hoạt, tính bảo mật và khả năng tích hợp, C# là một lựa chọn tốt cho các nhà phát triển muốn xây dựng các ứng dụng đa dạng trên nhiều nền tảng và môi trường phát triển.

## **Giới thiệu về Entity Framework**

Entity Framework (EF) là một khung làm việc mã nguồn mở được phát triển bởi Microsoft cho nền tảng .NET. Nó là một công cụ hỗ trợ ánh xạ đối tượng-quan hệ (ORM) cho phép phát triển ứng dụng dựa trên cơ sở dữ liệu quan hệ một cách dễ dàng và hiệu quả.

EF cung cấp một cách tiếp cận trừu tượng hóa dữ liệu, cho phép nhà phát triển làm việc với đối tượng trong mã nguồn và tự động ánh xạ chúng với cơ sở dữ liệu. Điều này giúp giảm bớt công việc lặp lại và mã code phức tạp, đồng thời tăng cường khả năng bảo trì và linh hoạt của ứng dụng.

Một số lợi ích chính của Entity Framework bao gồm:

*Tính trừu tượng hóa*: EF ẩn đi chi tiết cơ sở dữ liệu và cho phép nhà phát triển tập trung vào thiết kế và phát triển đối tượng. Điều này giúp tăng tính trừu tượng và linh hoạt của mã nguồn.

*Lập trình theo mô hình Code First*: EF cho phép nhà phát triển xây dựng cơ sở dữ liệu bằng cách viết mã nguồn trực tiếp, mà không cần tạo ra cơ sở dữ liệu trước. Điều này giúp tăng tốc quá trình phát triển và giảm bớt công việc quản lý phiên bản cơ sở dữ liệu.

*Tích hợp LINQ*: EF tích hợp với ngôn ngữ truy vấn LINQ (Language Integrated Query), cho phép nhà phát triển truy vấn dữ liệu bằng cú pháp gần gũi và linh hoạt. Điều này giúp tạo ra mã nguồn dễ đọc, dễ bảo trì và hiệu quả.

*Quản lý thay đổi tự động*: EF theo dõi và quản lý các thay đổi với cơ sở dữ liệu một cách tự động. Khi có sự thay đổi trong đối tượng, EF sẽ tự động tạo và thực thi câu truy vấn cập nhật tương ứng, giúp giảm bớt công việc lặp lại và đảm bảo tính nhất quán của cơ sở dữ liệu.

*Hỗ trợ đa dạng cơ sở dữ liệu*: EF hỗ trợ nhiều hệ quản trị cơ sở dữ liệu quan hệ như SQL Server, MySQL, PostgreSQL, Oracle và SQLite. Điều này cho phép nhà phát triển lựa chọn cơ sở dữ liệu phù hợp với yêu cầu dự án.

Các phiên bản của Entity Framework:

*Entity Framework 1.0 (2008)*: Đây là phiên bản đầu tiên của Entity Framework, được giới thiệu cùng với .NET Framework 3.5 SP1. Phiên bản này mang lại khả năng tạo bản đồ đối tượng-quan hệ (object-relational mapping) cơ bản và hỗ trợ truy vấn LINQ to Entities.

*Entity Framework 4.0 (2010)*: Phiên bản này được phát hành cùng với .NET Framework 4.0 và Visual Studio 2010. EF 4.0 giới thiệu một số cải tiến quan trọng như hỗ trợ POCO (Plain Old CLR Object) và Code First Development, cho phép lập trình viên tạo các đối tượng mô hình dữ liệu mà không cần sử dụng các lớp tạo mô hình (model-first) hoặc cơ sở dữ liệu hiện có (database-first).

*Entity Framework 5.0 (2012)*: Phiên bản này được phát hành cùng với .NET Framework 4.5 và Visual Studio 2012. EF 5.0 tập trung vào việc cải thiện hiệu suất và hỗ trợ cho các tính năng mới như hỗ trợ cho truy vấn không đồng bộ (asynchronous querying) và truy vấn batch (batch querying).

Entity Framework 6.0 (2013): Được phát hành cùng với .NET Framework 4.5.1 và Visual Studio 2013, EF 6.0 tập trung vào việc cải thiện hiệu suất và khả năng mở rộng, bao gồm cache thứ cấp, hỗ trợ cho các kiểu dữ liệu mới như datetime2 và enum, hỗ trợ truy vấn không đồng bộ toàn diện và nhiều tính năng khác.

Entity Framework Core (EF Core) 1.0 (2016): Phiên bản này là phiên bản đầu tiên của Entity Framework Core, là một phiên bản tái thiết và đa nền tảng của Entity Framework. EF Core 1.0 được phát triển để chạy trên .NET Core và hỗ trợ một loạt các cơ sở dữ liệu, bao gồm cả SQL Server, SQLite, MySQL và PostgreSQL.

Entity Framework Core (EF Core) 2.0 (2017): Phiên bản này được phát hành cùng với .NET Core 2.0 và tiếp tục cải thiện hiệu suất và tính năng của EF Core. EF Core 2.0 bổ sung hỗ trợ cho các tính năng như Tự động phát hiện thay đổi (Automatic Change Tracking), Hỗ trợ cơ sở dữ liệu quan hệ (Relational Database Support) và Hỗ trợ truy vấn không đồng bộ (Asynchronous Query Support).

Entity Framework Core (EF Core) 3.0 (2019): Phiên bản này được phát hành cùng với .NET Core 3.0 và EF Core 3.0 tập trung vào việc cải thiện hiệu suất và khả năng mở rộng, bao gồm hỗ trợ cho CSDL không quan hệ (NoSQL), sử dụng kiểu dữ liệu JSON, hỗ trợ Index tùy chỉnh và nhiều cải tiến khác.

Entity Framework Core (EF Core) 5.0 (2020): Phiên bản này được phát hành cùng với .NET 5.0 và EF Core 5.0 tập trung vào việc cải thithi hiệu suất và tính năng của EF Core. EF Core 5.0 bổ sung hỗ trợ cho nhiều kiểu cơ sở dữ liệu mới, bao gồm SQLite, SQL Server, MySQL, PostgreSQL, Oracle và nhiều hơn nữa. Nó cũng cung cấp nhiều cải tiến hiệu suất, bao gồm truy vấn không đồng bộ, kéo dài truy vấn và cache thứ cấp.

Entity Framework đã trở thành một công cụ phổ biến và phát triển mạnh mẽ trong việc phát triển các ứng dụng dựa trên .NET. Với khả năng giảm thiểu công việc lặp lại, tăng cường hiệu suất và linh hoạt, EF cung cấp một cách tiếp cận tiện ích và hiệu quả cho việc làm việc với cơ sở dữ liệu quan hệ trong môi trường .NET.

## Giới thiệu về cơ sở dữ liệu SQL Server

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) phát triển bởi Microsoft. Nó cung cấp một nền tảng mạnh mẽ và đáng tin cậy để lưu trữ, truy vấn và quản lý dữ liệu trong môi trường doanh nghiệp.

SQL Server cung cấp nhiều tính năng và lợi ích quan trọng:

Quản lý dữ liệu: SQL Server cho phép người dùng tạo, sửa đổi và xóa cơ sở dữ liệu, bảng, chỉ mục và ràng buộc. Nó cung cấp các công cụ quản lý mạnh mẽ để giúp người quản trị cơ sở dữ liệu theo dõi và duy trì tính toàn vẹn của dữ liệu.

Truy vấn dữ liệu: SQL Server hỗ trợ ngôn ngữ truy vấn SQL (Structured Query Language), cho phép người dùng truy vấn dữ liệu từ cơ sở dữ liệu. Với SQL Server, người dùng có thể viết các câu truy vấn phức tạp để truy xuất, lọc và sắp xếp dữ liệu một cách linh hoạt và hiệu quả.

Bảo mật và quản lý quyền truy cập: SQL Server cung cấp các công cụ và tính năng để bảo vệ dữ liệu và quản lý quyền truy cập. Người dùng có thể thiết lập các chính sách bảo mật, quản lý người dùng, vai trò và phân quyền để kiểm soát quyền truy cập vào dữ liệu.

Tích hợp dịch vụ: SQL Server cung cấp tích hợp với các dịch vụ và công nghệ khác của Microsoft như .NET Framework, Integration Services, Analysis Services và Reporting Services. Điều này giúp tạo ra các ứng dụng và giải pháp phức tạp hơn, từ việc xử lý dữ liệu đến phân tích và báo cáo.

Khả năng mở rộng và hiệu suất cao: SQL Server hỗ trợ khả năng mở rộng và có thể xử lý tải công việc lớn. Nó cung cấp các tính năng như phân vùng dữ liệu, sao chép dữ liệu và nhân rộng, tối ưu hóa truy vấn và đánh chỉ mục để đảm bảo hiệu suất cao và khả năng mở rộng của hệ thống.

Các phiên bản của SQL Server:

SQL Server 7.0 (1998): Đây là phiên bản đầu tiên của SQL Server dành cho môi trường Windows NT. Nó cung cấp nhiều tính năng quan trọng như truy vấn dựa trên ANSI SQL, xử lý ngoại lệ, giao dịch ACID và hỗ trợ dữ liệu có cấu trúc.

SQL Server 2000: Phiên bản này được phát triển trong giai đoạn nâng cấp từ SQL Server 7.0 và được phát hành vào năm 2000. SQL Server 2000 bổ sung nhiều tính năng mới như hỗ trợ XML, dịch vụ quản lý SQL Server, dịch vụ tích hợp Full-Text Search và tính năng phân tích dữ liệu dạng OLAP (Online Analytical Processing).

SQL Server 2005: Được phát hành vào năm 2005, SQL Server 2005 mang đến nhiều cải tiến quan trọng, bao gồm hỗ trợ CLR (Common Language Runtime) để phát triển các stored procedure và hàm bằng các ngôn ngữ lập trình .NET, tích hợp dịch vụ thông tin với tính năng truy vấn XML nâng cao, tính năng snapshot isolation, và quản lý dữ liệu dạng XML.

SQL Server 2008: Phiên bản này được phát hành vào năm 2008 và giới thiệu một số tính năng mới như dịch vụ báo cáo (Reporting Services), dịch vụ tích hợp toàn văn (Full-Text Search), quản lý chính sách (Policy-Based Management), ghi lại thay đổi dữ liệu (Change Data Capture) và Phân tích dữ liệu dạng tabular (Tabular Data Analysis).

SQL Server 2012: Được phát hành vào năm 2012, SQL Server 2012 bổ sung các tính năng như AlwaysOn Availability Groups (cho phép cấu hình cao hơn và khả năng chịu lỗi), các tính năng Business Intelligence (bao gồm Power View, PowerPivot và Power Query), Columnstore Indexes (tăng hiệu suất truy vấn dữ liệu phân tích), và tính năng Semantic Search.

SQL Server 2014: Phiên bản này được phát hành vào năm 2014 và giới thiệu tính năng In-Memory OLTP (Hekaton), cải thiện hiệu suất truy vấn và xử lý giao dịch. Ngoài ra, SQL Server 2014 cũng cung cấp tính năng backup và restore trực tuyến, tích hợp Azure Cloud và tăng cường tính năng AlwaysOn.

SQL Server 2016: Được phát hành vào năm 2016, SQL Server 2016 mang đến các tính năng như Query Store (lưu trữ và quản lý kế hoạch truy vấn), JSON support, R Services (phân tích dữ liệu thông qua R language), Always Encrypted (bảo mật dữ liệu), và Stretch Database (lưu trữ dữ liệu trên Azure).

SQL Server 2017: Phiên bản này được phát hành vào năm 2017 và giới thiệu tính năng như hỗ trợ Linux, Graph Database (hỗ trợ lưu trữ và truy vấn dữ liệu đồ thị), Adaptive Query Processing (tối ưu hóa truy vấn), và Machine Learning Services (hỗ trợ Python và R).

SQL Server 2019: Được phát hành vào năm 2019, SQL Server 2019 mang đến các tính năng như Big Data Clusters (kết hợp dữ liệu cấutrúc và phi cấu trúc trong một hệ thống duy nhất), hỗ trợ dự án dữ liệu lớn (data lakes), Enhanced PolyBase (truy vấn dữ liệu từ các nguồn khác nhau), hỗ trợ kịch bản Python trong stored procedure và SQL Graph (truy vấn dữ liệu đồ thị).

SQL Server đã trở thành một hệ quản trị cơ sở dữ liệu phổ biến và được sử dụng rộng rãi trong các môi trường doanh nghiệp. Với tính năng mạnh mẽ, khả năng mở rộng và tích hợp, SQL Server cung cấp một nền tảng đáng tin cậy cho việc quản lý và truy xuất dữ liệu trong các ứng dụng và hệ thống quan trọng.

# **CHƯƠNG 2. KHẢO SÁT, PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

## **Khảo sát hệ thống**

### Giới thiệu về hệ thống

### Các yêu cầu của hệ thống

#### Yêu cầu chức năng

Các chức năng chính:

* Đăng ký tài khoản, đăng nhập, sửa thông tin cá nhân

1. Thí sinh

* Đăng ký thi, thanh toán lệ phí thi online
* Xem lịch sử thi, lịch thi, làm bài thi khi tài khoản đã xác thực

1. Người đóng góp câu hỏi

* Nhập câu hỏi qua excel
* Thêm, sửa, xóa câu hỏi
* Cập nhật câu hỏi khi câu hỏi chưa được xác nhận

1. Người duyệt câu hỏi:

* Duyệt câu hỏi đề thi
* Bình luận lí do khi câu hỏi không hợp lệ

1. Quản lý tạo ngân hàng câu hỏi

* Theo dõi tiến độ : Tạo câu hỏi, Đánh giá câu hỏi
* Quyết định tài khoản nào là Người đóng góp câu hỏi, Người đánh giá câu hỏi

1. Quản lý cuộc thi

* Tạo kỳ thi, phòng thi
* Xem danh sách tài khoản dự thi
* Xem kết quả thi chung hoặc của riêng 1 thí sinh
* Xác thực tài khoản của thí sinh, nếu thông tin sai thì gửi thông báo đến thí sinh
* Tạo danh sách dự thi để đối chiếu thí sinh trước khi vào phòng thi gồm: ảnh, thông tin thí sinh.
* Xem tiến độ tạo đề
* Bổ nhiệm quản lý đề thi

1. Quản lý đề thi

* Quản lý cấu trúc đề thi
* Đánh giá đề thi ( đề thi được tạo tự động)

1. Người quản trị tài khoản

* Xem danh sách các tài khoản
* Cấp quyền cho các tài khoản: quản lý cuộc thi.

#### Yêu cầu phi chức năng

* **Yêu cầu bảo mật**

Đối với những yêu cầu về bảo mật, em sẽ tập chung vào hai hình thức chính đó là bảo mật theo sự phân quyền và bảo mật qua việc mã hóa mật khẩu khi người dùng đăng nhập.

* **Yêu cầu về hiệu năng**

Một website cơ bản không nhất thiết là phải theo quy chuẩn cơ bản. Nó có nhiều trang con được liên kết với nhau và các trang sẽ sử dụng script ví dụ như C# cần truy vấn cơ sở dữ liệu và thực hiện chức năng. Tuy nhiên, việc thiết kế và phát triển website chuyên nghiệp luôn luôn theo một khuôn khổ tiêu chuẩn, chẳng hạn như mô hình MVC (Model - View - Controller). Trong mô hình này, website được chia thành 3 lớp: lớp thấp nhất là lớp Model sẽ xử lý tất cả các truy vấn cơ sở dữ liệu. Lớp Controller thực hiện các chức năng web thông qua những tiện ích cơ sở dữ liệu trong lớp Model. Lớp View xử lý giao diện người dùng mà người xem có thể thấy, bằng cách sử dụng các chức năng được xây dựng trong lớp điều khiển. Kiểu kiến trúc này giúp trang web làm việc ổn định hơn, dễ dàng hơn để mở rộng, nâng cấp, hoặc mở rộng.

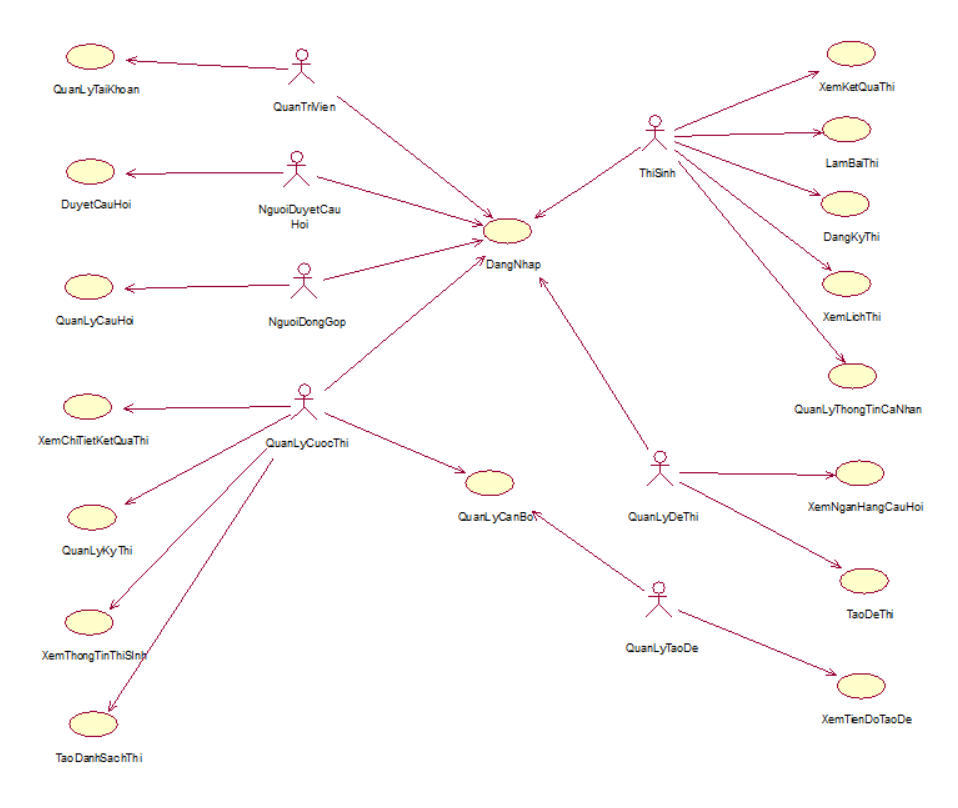
Đối với một trang web bán hàng online, lượng truy cập của khách hàng sẽ rất nhiều trong cũng một thời điểm, vì vậy hệ thống cần phải có khả năng chịu tải phù hợp, có thể xử lý các request của khách hàng một cách nhanh nhất. Kèm theo đó những tương tác của người sử dụng đối với trang web cần được phản hồi nhanh chóng, vì vậy kĩ thuật tải AJAX cho phép nội dung được tải sẵn trước khi người dùng thực sự xem nội dung đó, điều này sẽ giúp người dùng có trải nghiệm mượt mà hơn.

Tóm lại, về mặt hiệu năng của hệ thống, em sẽ được xây dựng dựa trên mô hình MVC, sử dụng tối đa khả năng của AJAX, kèm theo đó là có một thiết kế về cơ sở dữ liệu phù hợp.

## Phân tích hệ thống

### Mô hình use case

#### Biểu đồ use case



Hình 2.1. Biểu đồ use case tổng quan của hệ thống

#### Biểu đồ use case phân rã

### 2.2.2. Đặc tả use case

1. Đặc tả Use case Đăng ký

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC | Tên use case | Đăng ký |
| Tác nhân | Thí sinh, quản lý hồ sơ, người đóng góp, quản lý thi, kiểm duyệt viên, người chấm thi | | |
| Mô tả | Cho phép các tác nhân đăng ký tài khoản | | |
| Sự kiện kích hoạt | Kích vào nút “Đăng ký” | | |
| Tiền điều kiện | Không có | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | # | Thực hiện bởi | Hành động | | 1 | Các tác nhân | Kích vào “Đăng ký” | | 2 | Hệ thống | Hiển thị màn hình đăng ký | | 3 | Các tác nhân | Nhập thông tin Họ và tên, Ngày sinh, Giới tính, Số CCCD, Ngày cấp, Nơi cấp, Quốc tịch, Địa chỉ, Số điện thoại, Email, Mật khẩu, Nghề nghiệp, Ảnh bản thân, Ảnh mặt trước CCCD, Ảnh mặt sau CCCD và kích nút “Đăng ký” | | 4 | Hệ thống | Hiển thị màn hình gồm các thông tin đăng ký và yêu cầu các tác nhân xác nhận | | 5 | Các tác nhân | Kích nút “Xác nhận” | | 6 | Hệ thống | Thêm thông tin vào bảng Account | | | |
| Lồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | # | Thực hiện bởi | Hành động | | 4b | Hệ thống | Nếu tác nhân bỏ trống trường thông tin thì hiển thị thông báo yêu cầu nhập thông tin | | 6b | Hệ thống | Hiển thị thông báo Email đã tồn tại | | | |
| Hậu điều kiện | Thông tin các tác nhân được thêm vào bảng Account | | |
| Điểm mở rộng | Không có | | |

1. Đặc tả Use case Quản lý thông tin cá nhân

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC01 | Tên use case | Quản lý thông tin cá nhân |
| Tác nhân | Thí sinh, quản lý hồ sơ, người đóng góp, quản lý thi, kiểm duyệt viên, người chấm thi | | |
| Mô tả | Cho phép các tác nhân xem, sửa thông tin cá nhân | | |
| Sự kiện kích hoạt | Kích nút “Thông tin cá nhân” | | |
| Tiền điều kiện | Đăng nhập thành công | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | # | Thực hiện bởi | Hành động | | 1 | Các tác nhân | Kích nút “Thông tin cá nhân” | | 2 | Hệ thống | Lấy thông tin cá nhân của tác nhân gồm Mã định danh, Họ và tên, Ngày sinh, Giới tính, Địa chỉ, Số điện thoại, Nghề nghiệp từ bảng Account và hiển thị lên màn hình | | 3 | Các tác nhân | Kích nút “Sửa thông tin” | | 4 | Hệ thống | Cho phép sửa các trường thông tin | | 5 | Các tác nhân | Nhập thông tin cần sửa và kích nút “Cập nhật” | | 6 | Hệ thống | Cập nhật thông tin tác nhân vào bảng Account | | | |
| Lồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | # | Thực hiện bởi | Hành động | | 5b | Hệ thống | Nếu tác nhân bỏ trống trường thông tin thì hiển thị thông báo yêu cầu nhập thông tin | | | |
| Hậu điều kiện | Thông tin được cập nhật vào bảng Account | | |
| Điểm mở rộng | Không có | | |

1. Mô tả Use case Xem thông báo

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC01 | Tên use case | Xem thông báo |
| Tác nhân | Thí sinh, quản lý hồ sơ, người đóng góp, quản lý thi, kiểm duyệt viên, người chấm thi | | |
| Mô tả | Cho phép các tác nhân xem thông báo | | |
| Sự kiện kích hoạt | Kích vào nút “Thông báo” | | |
| Tiền điều kiện | Đăng nhập thành công | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | # | Thực hiện bởi | Hành động | | 1 | Các tác nhân | Kích vào nút “Thông báo” | | 2 | Hệ thống | Lấy thông tin Nội dung, Thời gian, Người gửi từ bảng Notification và trả ra màn hình | | | |
| Lồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | # | Thực hiện bởi | Hành động | | 2b | Hệ thống | Thông báo “Không có thông báo” nếu không có dữ liệu trong bảng Notification | | | |
| Hậu điều kiện | Không có | | |
| Điểm mở rộng | Không có | | |

1. Mô tả use case Quản lý tài khoản

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC01 | Tên use case | Quản lý tài khoản |
| Tác nhân | Quản trị viên | | |
| Mô tả | Cho phép quản trị viên xem, phân quyền cho tài khoản | | |
| Sự kiện kích hoạt | Kích nút “Quản lý tài khoản” | | |
| Tiền điều kiện | Đăng nhập bằng tài khoản quản trị viên | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | # | Thực hiện bởi | Hành động | | 1 | Hệ thống | Lấy danh sách Họ và tên, Email, Vai trò từ bảng Account và hiển thị ra màn hình | | 2 | Quản trị viên | Tìm tài khoản trong danh sách tài khoản, kích nút sửa vai trò | | 3 | Hệ thống | Hiển thị danh sách vai trò | | 4 | Quản trị viên | Chọn vai trò và nhấn nút “Cập nhật” | | 5 | Hệ thống | Cập nhật Vai trò trong bảng Account | | | |
| Lồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | # | Thực hiện bởi | Hành động | | 2b | Quản trị viên | Nhập email vào ô tìm kiếm và kích “Tìm kiếm” | | 3b | Hệ thống | Tìm kiếm tài khoản gồm các thông tin Họ và tên, Email, Vai trò từ bảng Account và hiển thị ra màn hình | | | |
| Hậu điều kiện | Cập nhật vai trò vào bảng Account | | |
| Điểm mở rộng | Không có | | |

4. Đặc tả Use case Xem thông tin thí sinh

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC01 | Tên use case | Xem thông tin thí sinh |
| Tác nhân | Quản lý thi | | |
| Mô tả | Cho phép quản lý thi | | |
| Sự kiện kích hoạt |  | | |
| Tiền điều kiện |  | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | # | Thực hiện bởi | Hành động | |  |  |  | |  |  |  | |  |  |  | |  |  |  | | | |
| Lồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | # | Thực hiện bởi | Hành động | |  |  |  | |  |  |  | |  |  |  | | | |
| Hậu điều kiện |  | | |
| Điểm mở rộng |  | | |

### 2.2.3. Biểu đồ trình tự

### 2.2.4. Biểu đồ lớp

2.2.5. Thiết kế cơ sở dữ liệu

2.2.5.1. Mô hình dữ liệu quan hệ

2.2.5.2. Chi tiết các bảng