TRƯỜNG ĐẠI HỌC ĐỒNG THÁP

**KHOA SƯ PHẠM TOÁN – TIN**



0018410634 – PHẠM NGUYỄN THANH DUY

**BÁO CÁO THỰC TẬP TỐT NGHIỆP**

**Đề tài**

**XÂY DỰNG WEBSITE QUẢN LÝ LAO ĐỘNG TRƯỜNG ĐẠI HỌC ĐỒNG THÁP**

NGÀNH: KHOA HỌC MÁY TÍNH

LỚP: ĐHCNTT18A

*Đồng Tháp, tháng 4 năm 2022*

TRƯỜNG ĐẠI HỌC ĐỒNG THÁP

**KHOA SƯ PHẠM TOÁN – TIN**



0018410634 – PHẠM NGUYỄN THANH DUY

**BÁO CÁO THỰC TẬP TỐT NGHIỆP**

**Đề tài**

**XÂY DỰNG WEBSITE QUẢN LÝ LAO ĐỘNG TRƯỜNG ĐẠI HỌC ĐỒNG THÁP**

NGÀNH: KHOA HỌC MÁY TÍNH

LỚP: ĐHCNTT18A

Cán bộ hướng dẫn: TRẦN PHƯỚC DƯ

Giảng viên hướng dẫn: Th.s NGUYỄN THỊ THANH THẢO

*Đồng Tháp, tháng 4 năm 2022*

# **LỜI CẢM ƠN**

Trước tiên, em xin gửi lời cảm ơn đến quý Thầy Cô bộ môn Công Nghệ Thông Tin trong khoa Sư phạm Toán – Tin đã trang bị cho em những kiến thức, kỹ năng cần thiết để hoàn thành báo cáo thực tập này.

Đặc biệt, em xin gửi lời cảm ơn chân thành đến cô NGUYỄN THANH THẢO đã tận tình hướng dẫn, giúp đỡ em trong suốt quá trình thực hiện và hoàn thành báo cáo: *“Xây dựng website quản lý lao động trường Đại học Đồng Tháp”.*

Và đặc biệt, em xin chân thành cảm ơn các anh chị trong ***Trung tâm Công Nghệ Thông Tin và Truyền Thông Đồng Tháp*** đã giúp đỡ em trong suốt thời gian thực tập vừa qua đồng thời tạo điều kiện cho em học hỏi được nhiều kiến thức mới để hoàn thành tốt bài thực tập.

Do kiến thức bản thân còn hạn chế nên trong quá trình thực hiện, hoàn thiện đề tài còn nhiều thiếu sót khi tìm hiểu, đánh giá và trình bày về đề tài. Rất mong sự quan tâm và đóng góp của Thầy Cô và Quý lãnh đạo của cơ quan nơi thực tập để đề tài được đầy đủ và hoàn chỉnh hơn.

Em xin chân thành cảm ơn!

*Đồng Tháp, ngày 08 tháng 4 năm 2022*

**Phạm Nguyễn Thanh Duy**

# **MỤC LỤC**

[**LỜI CẢM ƠN** i](#_Toc88850925)

[**MỤC LỤC** ii](#_Toc88850926)

[**DANH MỤC HÌNH** v](#_Toc88850927)

[**PHẦN MỞ ĐẦU** vii](#_Toc88850928)

[**I.** **Đặt vấn đề** vii](#_Toc88850929)

[**II.** **Mục tiêu nghiên cứu** vii](#_Toc88850930)

[**III.** **Phạm vi nghiên cứu** vii](#_Toc88850931)

[**IV.** **Phương pháp nghiên cứu** vii](#_Toc88850932)

[**V.** **Thời gian thực hiện** viii](#_Toc88850933)

[**PHẦN NỘI DUNG** 1](#_Toc88850934)

[**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT** 1](#_Toc88850935)

[**1.1** **Giới thiệu hệ quản trị cơ sở dữ liệu SQL Server** 1](#_Toc88850936)

[**1.1.1 Khái niệm** 1](#_Toc88850937)

[**1.1.2 Lịch sử ra đời** 1](#_Toc88850938)

[**1.1.3 Một vài ấn bản của SQL Server** 1](#_Toc88850939)

[**1.2** **Giới thiệu ngôn ngữ lập trình C# và .Net framework** 1](#_Toc88850940)

[**1.2.1** **Giới thiệu C#** 1](#_Toc88850941)

[**1.2.2** **Đặc trưng của ngôn ngữ C#** 1](#_Toc88850942)

[**1.2.3** **Nền tảng .NET** 2](#_Toc88850943)

[**1.3** **Giới thiệu Visual Studio** 2](#_Toc88850944)

[**CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG** 5](#_Toc88850945)

[**2.1 Mô tả bài toán** 5](#_Toc88850946)

[**2.2 Sơ đồ Use Case** 5](#_Toc88850947)

[**2.2.1 Sơ đồ tổng quát** 5](#_Toc88850948)

[**2.2.2 Sơ đồ thành phần** 6](#_Toc88850949)

[**2.2.2.1 Use Case Quản lý thông tin nhân khẩu** 6](#_Toc88850950)

[**2.2.2.2 Use Case Quản lý thông tin điểm tiêm** 7](#_Toc88850951)

[**2.2.2.3 Use Case Quản lý thông tin phiếu tiêm** 7](#_Toc88850952)

[**2.2.2.4 Use Case Quản lý đợt tiêm** 8](#_Toc88850953)

[**2.2.2.5 Use Case Quản lý thông tin phân bổ vaccine** 8](#_Toc88850954)

[**2.2.2.6 Use Case Báo cáo thống kê** 9](#_Toc88850955)

[**2.2.2.7 Use Case chi tiết** 10](#_Toc88850956)

[**2.3 Sơ đồ lớp** 10](#_Toc88850957)

[**2.4 Sơ đồ hoạt động** 15](#_Toc88850958)

[**2.4.1 Hoạt động đăng nhập** 15](#_Toc88850959)

[**2.4.2 Hoạt động thống kê** 15](#_Toc88850960)

[**2.4.3 Hoạt động thêm nhân khẩu mới** 16](#_Toc88850961)

[**2.5 Sơ đồ trạng thái** 17](#_Toc88850962)

[**2.5.1 Sơ đồ trạng thái thêm thông tin chi tiết tiêm** 17](#_Toc88850963)

[**2.5.2 Sơ đồ trạng thái lập báo cáo** 18](#_Toc88850964)

[**2.6 Sơ đồ tuần tự** 18](#_Toc88850965)

[**2.6.1 Sơ đồ tuần tự đăng nhập** 18](#_Toc88850966)

[**2.6.2 Sơ đồ tuần tự tìm kiếm vaccine** 19](#_Toc88850967)

[**2.6.3 Sơ đồ tuần tự quản lý nhân khẩu** 20](#_Toc88850968)

[**2.7 Sơ đồ thành phần** 20](#_Toc88850969)

[**2.8 Sơ đồ triển khai** 21](#_Toc88850970)

[**2.9 Thiết kế cơ sở dữ liệu** 21](#_Toc88850971)

[**CHƯƠNG 3: XÂY DỰNG PHẦN MỀM QUẢN LÝ THÔNG TIN TIÊM CHỦNG VACCINE COVID-19** 27](#_Toc88850972)

[**3.1** **Giao diện đăng nhập** 27](#_Toc88850973)

[**3.2** **Giao diện màn hình chính** 28](#_Toc88850974)

[**3.3** **Giao diện quản lý** 29](#_Toc88850975)

[**3.3.1** **Giao diện quản lý thông tin nhân khẩu** 29](#_Toc88850976)

[**3.3.2** **Giao diện quản lý thông tin điểm tiêm** 31](#_Toc88850977)

[**3.3.3** **Quản lý chi tiết tiêm** 32](#_Toc88850978)

[**3.3.4** **Quản lý thông tin đợt tiêm** 33](#_Toc88850979)

[**3.3.5** **Quản lý thông tin phân bổ vaccine** 34](#_Toc88850980)

[**3.4** **Giao diện báo cáo thống kê** 36](#_Toc88850981)

[**3.4.1** **Giao diện thống kê số người tiêm** 36](#_Toc88850982)

[**3.4.2** **Giao diện thống kê Vaccine** 37](#_Toc88850983)

[**3.4.3** **Giao diện báo cáo** 37](#_Toc88850984)

[**3.5** **Giao diện sao lưu – phục hồi dữ liệu** 38](#_Toc88850985)

[**PHẦN KẾT LUẬN** 40](#_Toc88850986)

[**I.** **Kết quả đạt được** 40](#_Toc88850987)

[**II.** **Hướng phát triển** 40](#_Toc88850988)

[**TÀI LIỆU THAM KHẢO** 41](#_Toc88850989)

# **DANH MỤC HÌNH**

# **PHẦN MỞ ĐẦU**

**---oOo---**

1. **Đặt vấn đề**

Hiện nay Công nghệ thông tin vô cùng phát triển thì mọi người đều sử dụng máy vi tính để làm việc. Công nghệ thông tin cũng được áp dụng rất nhiều vào các lĩnh vực mà điển hình là lĩnh vực quản lý. Như chúng ta đã biết tại các trường đại học, việc xây dựng quản lí lao động công ích của sinh viên thì rất quan trọng nhưng lại còn rất thủ công, chỉ xây dựng và lưu lại trên sổ sách gây mất thời gian, khó khăn trong việc thống nhất, theo dõi và quản lý cho nhà trường. Từ thực tế đó, việc xây dựng được website quản lý lao động là rất cần thiết. Vì vậy em đã chọn đề tài “Xây dựng website quản lý lao động Trường Đại học Đồng Tháp” với mục đích với nghiên cứu, tìm hiểu về và xây dựng website để có thể đáp ứng được nhu cầu quản lý cho nhà trường, giúp cho mọi người có thể tiết kiệm công sức, thời gian đi lại và giúp việc quản lý dễ dàng, tiện lợi hơn.

1. **Mục tiêu nghiên cứu**

Dựa trên tình hình thực tế, việc nghiên cứu này sẽ tạo ra một website hỗ trợ trong việc đăng ký lao động đối với sinh viên cũng như là quản lý phân công lao động đối với cán bộ quản lý ở trường Đại học Đồng Tháp.

1. **Phạm vi nghiên cứu**

Phạm vi nghiên cứu là khu vực Trường Đại học Đồng Tháp, Thành phố Cao Lãnh, tỉnh Đồng Tháp.

1. **Phương pháp nghiên cứu**

Để thực hiện đề tài, em đã sử dụng phương pháp nghiên cứu lý thuyết và phương pháp thực hành để có thể hoàn thành được mục tiêu của đề tài đã đề ra.

Cụ thể như sau:

* Phương pháp nghiên cứu lý thuyết:

+ Nghiên cứu những tài liệu liên quan đến nội dung cần thực hiện

* Phương pháp thực nghiệm: kết hợp với nghiên cứu lý thuyết và thực nghiệm

+ Tìm hiểu về ngôn ngữ lập trình C#, Asp.net core, Entity Framework core và viết chương trình.

+ Chạy thử nghiệm và đánh giá kết quả đạt được.

1. **Thời gian thực hiện**

|  |  |
| --- | --- |
| **Thời gian** | **Nội dung thực hiện** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**PHẦN NỘI DUNG**

**-----oOo-----**

**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT**

* 1. **Giới thiệu nền tảng .Net Core**

### **1.1.1 Khái niệm**

ASP.NET Core là một web framework mã nguồn đa nền tảng (cross-platform) cho việc xây dựng những ứng dụng hiện tại dựa trên kết nối đám mây, giống như web apps, IoT và backend cho mobile.

Ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên phiên bản đầy đủ của .NET Framework. Nó được thiết kế để cung cấp và tối ưu development framework cho những dụng cái mà được triển khai trên đám mây (clound) hoặc chạy on-promise.

Nó bao gồm các thành phần theo hướng module nhằm tối thiểu tài nguyên và chi phí phát triển, như vậy bạn giữ lại được sự mềm giẻo trong việc xây dựng giải pháp của bạn. Bạn có thể phát triển và chạy những ứng dụng ASP.NET Core đa nền tảng trên Windows, Mac và Linux.

### **1.1.2 Lịch sử ra đời**

ASP.NET đã được sử dụng từ nhiều năm để phát triển các ứng dụng web. Kể từ đó, framework này đã trải qua một sự thay đổi tiến hóa đều đặn và cuối cùng đã đưa chúng ta đến với ASP.NET Core 1.0 hậu duệ mới gần đây nhất của nó. ASP.NET Core không phải là phiên bản tiếp theo của ASP.NET 4.6. Đó là một framework hoàn toàn mới, may mắn rằng nó một dự án side-by-side tương tự với mọi thứ mà chúng ta biết. Nó thực ra được viết lại trên framework ASP.NET 4.6 hiện tại những kích thước nhỏ hơn và nhiều modular hơn.



Hình 1.1 Lịch sử phát triển của .Net core

### **1.1.3 Thành phần của .Net core**

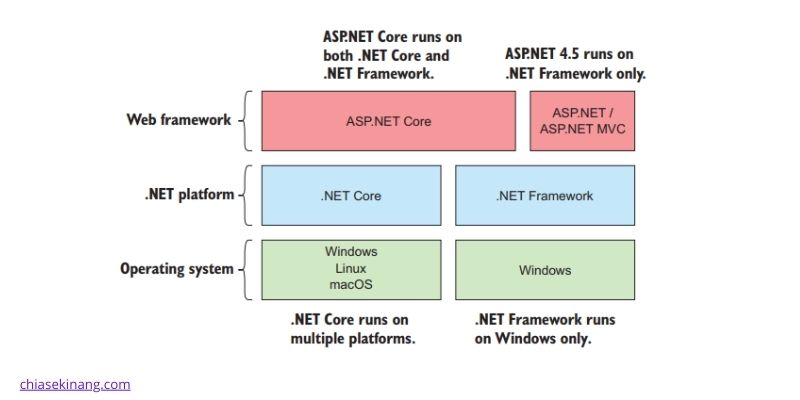
.NET Core bao gồm các phần sau:

.NET Core runtime: cung cấp một hệ thống kiểu, tải lắp ráp, trình thu gom rác, interop gốc và các dịch vụ cơ bản khác. Các thư viện khung .NET Core cung cấp các kiểu dữ liệu nguyên thủy, các kiểu thành phần ứng dụng và các tiện ích cơ bản.

ASP.NET Core runtime: cung cấp khung để xây dựng các ứng dụng kết nối internet, điện toán đám mây hiện đại, chẳng hạn như ứng dụng web, ứng dụng IoT và phụ trợ di động.

.NET Core SDK và trình biên dịch ngôn ngữ (Roslyn và F #) cho phép trải nghiệm nhà phát triển .NET Core.

Dotnet command, được sử dụng để khởi chạy các ứng dụng .NET Core và các lệnh CLI. Nó chọn thời gian chạy và lưu trữ thời gian chạy, cung cấp chính sách tải lắp ráp và khởi chạy các ứng dụng và công cụ.



Hình 1.2 Thành phần của .Net core

**1.1.4 Ưu điểm của .Net core**

ASP.NET Core đi kèm với những ưu điểm sau:

* ASP.NET Core có một số thay đổi kiến trúc dẫn đến modular framework nhỏ hơn.
* ASP.NET Core không còn dựa trên System.Web.dll. Nó dựa trên một tập hợp nhiều yếu tố của Nuget packages.
* Điều này cho phép bạn tối ưu ứng dụng của mình chỉ cần những NuGet packages cần thiết.
* Lợi ích của diện tích bề mặt ứng dụng nhỏ hơn thì bảo mật chặt chẽ hơn, giảm dịch vụ, cải thiện hiệu suất và giảm chi phí.

Với ASP.NET Core, bạn có thể nhận được các cải tiến sau:

* Xây dựng và chạy các ứng dụng ASP.NET Core đa nền tảng trên Windows, Mac và Linux.
* Được xây dựng trên **.NET Core**, hỗ trợ side-by-side app versioning.
* Công cụ mới giúp đơn giản hóa việc phát triển web hiện đại.
* Liên kết đơn các web stack như Web UI và API Web.
* Cấu hình dựa trên môi trường đám mây sẵn có.
* Được xây dựng dựa trên cho DI (Dependency Injection).
* Tag Helpers làm cho các Razor makup trở nên tự nhiên hơn với HTML.
* Có khả năng host trên IIS hoặc self-host.
  1. **Giới thiệu Entity Framework core**
     1. **Giới thiệu**

Entity Framework Core là phiên bản mới của Entity Framework sau EF 6.x. Nó là mã nguồn mở, nhẹ, có thể mở rộng và là phiên bản đa nền tảng của công nghệ truy cập dữ liệu Entity Framework. Entity Framework là một framework Object/Relational Mapping (O/RM - ánh xạ quan hệ/đối tượng). Đây là một cải tiến của ADO.NET, cung cấp cho các nhà phát triển một cơ chế tự động để truy cập và lưu trữ dữ liệu trong cơ sở dữ liệu. EF Core được dự định sẽ sử dụng với các ứng dụng .NET Core. Tuy nhiên, nó cũng có thể được sử dụng với các ứng dụng dựa trên .NET Framework 4.5+ tiêu chuẩn.

* + 1. **Cách tiếp cận phát triển của EF Core**

EF Core hỗ trợ hai cách tiếp cận phát triển: Code First và Database First. EF Core chủ yếu nhắm vào cách tiếp cận Code First và cung cấp ít hỗ trợ cho cách tiếp cận Database First vì trình thiết kế trực quan hoặc trình hướng dẫn cho mô hình DB không được hỗ trợ kể từ EF Core 2.0. Theo cách tiếp cận Code First, EF Core API tạo cơ sở dữ liệu và các bảng bằng cách sử dụng chuyển đổi (migration) dựa trên các quy ước và cấu hình được cung cấp trong các lớp thực thể của bạn. Cách tiếp cận này rất hữu ích trong thiết kế hướng miền (Domain Driven Design - DDD).

Diagram

Description automatically generated

Hình 1.3 Mô hình EF core

* + 1. **EF Core với EF 6**

Entity Framework Core là phiên bản mới và cải tiến của Entity Framework cho các ứng dụng .NET Core. EF Core là phiên bản mới hoàn toàn, vì vậy nó vẫn chưa đầy đủ như EF 6. EF Core tiếp tục hỗ trợ các tính năng và khái niệm giống như EF 6 như sau:

1. DbContext và Dbset.

2. Mô hình dữ liệu.

3. Truy vấn bằng cách sử dụng Linq-to-Entities.

4. Theo dõi thay đổi.

5. SaveChanges.

6. Chuyển đổi (Migration).

EF Core dần dần sẽ có hầu hết các tính năng của EF 6. Tuy nhiên, có một số tính năng của EF 6 không được hỗ trợ trong EF Core , chẳng hạn như:

1. EDMX/Graphical Visualization của mô hình.

2. Trình hướng dẫn mô hình dữ liệu thực thể (cho phương pháp Database First).

3. ObjectContext API.

4. Truy vấn bằng cách sử dụng Entity SQL.

5. Chuyển đổi tự động.

6. Kế thừa: Bảng trên mỗi kiểu (TPT)

7. Kế thừa: Bảng trên mỗi lớp con (TPC)

8. Mối quan hệ nhiều-nhiều.

9. Chia tách thực thể.

10. Kiểu dữ liệu không gian.

11. Ánh xạ stored procedure với DbContext cho hoạt động CUD (Create, Update, Delete).

12. Seed data.

* 1. **Giới thiệu Visual Studio**

Visual studio là một phần mềm hỗ trợ đắc lực hỗ trợ công việc lập trình website. Công cụ này được tạo lên và thuộc quyền sở hữu của ông lớn công nghệ Microsoft. Năm 1997, phần mềm lập trình nay có tên mã Project Boston. Nhưng sau đó, Microsoft đã kết hợp các công cụ phát triển, đóng gói thành sản phẩm duy nhất.

Visual Studio là hệ thống tập hợp tất cả những gì liên quan tới phát triển ứng dụng, bao gồm trình chỉnh sửa mã, trình thiết kế, gỡ lỗi. Tức là, bạn có thể viết code, sửa lỗi, chỉnh sửa thiết kế ứng dụng dễ dàng chỉ với 1 phần mềm Visual Studio mà thôi. Không dừng lại ở đó, người dùng còn có thể thiết kế giao diện, trải nghiệm trong Visual Studio như khi phát triển ứng dụng Xamarin, UWP bằng XAML hay Blend vậy.

Phần mềm Visual studio được chia thành 2 phiên bản Visual Studio Enterprise và Visual Studio Professional, các phiên bản cao cấp có tính phí này được sử dụng nhiều bởi các công ty chuyên về lập trình. Bên cạnh đó, Microsoft cũng cho ra mắt phiên bản Community (phiên bản miễn phí) của gói phần mềm, cung cấp cho người dùng những tính năng cơ bản nhất, phù hợp với các đối tượng lập trình không chuyên, mới tiếp cận tìm hiểu về công nghệ (đối tượng nghiên cứu, nhà phát triển cá nhân, hỗ trợ dự án mỡ, các tổ chức phi doanh nghiệp dưới 5 người dùng).

***• Các tính năng của Visual Studio***

+ Đa nền tảng: Phần mềm lập trình Visual Studio của Microsoft hỗ trợ sử dụng trên nhiều nền tảng khác nhau. Không giống như các trình viết code khác, Visual Studio sử dụng được trên cả Windows, Linux và Mac Systems. Điều này cực kỳ tiện lợi cho lập trình viên trong quá trình ứng dụng.

+ Đa ngôn ngữ lập trình: Không chỉ hỗ trợ đa nền tảng, Visual Studio cũng cho phép sử dụng nhiều ngôn ngữ lập trình khác nhau từ C#, F#, C/C++, HTML, CSS, Visual Basic, JavaScript,…Bởi vậy, Visual Studio có thể dễ dàng phát hiện và thông báo cho bạn khi các chương trình có lỗi.

+ Hỗ trợ website: Visual Studio code cũng hỗ trợ website, đặc biệt trong công việc soạn thảo và thiết kế web.

+ Kho tiện ích mở rộng phong phú: Mặc dù Visual Studio có hệ thống các ngôn ngữ hỗ trợ lập trình khá đa dạng. Nhưng nếu lập trình viên muốn sử dụng một ngôn ngữ khác, bạn có thể dễ dàng tải xuống các tiện ích mở rộng. Tính năng hấp dẫn này được hoạt động như một phần chương trình độc lập nên không lo làm giảm hiệu năng của phần mềm.

+ Lưu trữ phân cấp: Phần lớn các tệp dữ liệu đoạn mã của Visual Studio đều được đặt trong các thư mục tương tự nhau. Đồng thời, Visual Studio cũng cung cấp một số thư một cho các tệp đặc biệt để bạn lưu trữ an toàn, dễ tìm, dễ sử dụng hơn.

+ Kho lưu trữ an toàn: Với Visual Studio, bạn có thể hoàn toàn yên tâm về tính lưu trữ, bởi phần mềm đã được kết nối GIT và một số kho lưu trữ an toàn được sử dụng phổ biến hiện nay.

+ Màn hình đa nhiệm: Visual Studio sở hữu tính năng màn hình đa nhiệm, cho phép người dùng mở cùng lúc nhiều tập tin, thư mục dù chúng có thể không liên quan tới nhau.

+ Hỗ trợ viết code: Khi sử dụng code vào trong lập trình, với Visual Studio, công cụ này có thể đề xuất tới các lập trình viên một số tùy chọn thay thế nhằm điều chỉnh đôi chút để đoạn code áp dụng thuận tiện hơn cho người dùng.

+ Hỗ trợ thiết bị đầu cuối: Phần mềm Visual Studio cũng tích hợp các loại thiết bị đầu cuối, giúp người dùng không cần chuyển đổi giữa hai màn hình hay trở về thư mục gốc khi thực hiện một thao tác cần thiết nào đó.

+ Hỗ trợ Git: Do kết nối với GitHub nên Visual Studio cho phép hỗ trợ sao chép, kéo thả trực tiếp. Các mã code này sau đó cũng có thể thay đổi và lưu lại trên phần mềm.

+ Intellisense: Tính năng nhắc Intellisense được sử dụng hầu hết trong các phần mềm lập trình, bao gồm cả Visual Studio. Tuy nhiên, so với các trình viết mã, Visual Studio vẫn được đánh giá cao về tính chuyên nghiệp. Đặc biệt, tính năng này còn có thể phát hiện tất cả các đoạn mã không đầy đủ, nhắc lập trình viên, gợi ý sửa đổi, khai báo biến tự động trong trường hợp lập trình viên quên, giúp bổ sung cú pháp còn thiếu,…

+ Tính năng comment: Một tính năng cũng khá hay ho, hỗ trợ cho người lập trình trong trường hợp “nhớ nhớ quên quên” đó là tính năng bình luận. Tính năng này cho phép lập trình viên để lại nhận xét, giúp dễ dàng ghi nhớ công việc cần hoàn thành, không bỏ sót công đoạn nào.

**1.4 Hệ quản trị cơ sở dữ liệu SQL Server**

* ***Khái niệm***

SQL Server (viết tắt của cụm từ Structured Query Language) là một hệ thống quản lý cơ sở dữ liệu quan hệ (Relational Database Management System, viết tắt là RDBMS). SQL Server có khả năng hỗ trợ một số lượng lớn các quy trình xử lý giao dịch, ứng dụng doanh nghiệp và ứng dụng phân tích trong các công ty hoạt động trong lĩnh vực IT.

* ***Tính năng quản lý***

Về phương diện quản trị, Microsoft SQL Server gồm các dịch vụ tích hợp SQL Server, dịch vụ SQL Server Data Quality và dịch vụ SQL Server Master. Hai bộ công cụ dành riêng cho quản trị viên cơ sở dữ liệu và lập trình viên (SQL Server Data Tools) sử dụng trong việc phát triển hệ thống cơ sở dữ liệu cho SQL Server Management Studio, thực hiện nhiệm vụ triển khai, giám sát và quản lý các cơ sở dữ liệu.

* ***Vai trò của SQL server***

SQL Server không phải là một hệ quản trị cơ sở dữ liệu độc lập. Nó là thành phần với vai trò ngôn ngữ làm công cụ giao tiếp của cơ sở dữ liệu và người dùng.

* SQL là một ngôn ngữ đòi hỏi sự tương tác cao. Người dùng có thể dễ dàng trao đổi với tiện ích bằng câu lệnh SQL đến cơ sở dữ liệu và nhận lại kết quả từ đó.
* SQL là một ngôn ngữ lập trình cơ sở dữ liệu. Bằng cách nhúng các câu lệnh SQL trong ngôn ngữ lập trình, các lập trình viên có thể xây dựng được các chương trình ứng dụng giao tiếp với cơ sở dữ liệu
* SQL là một ngôn ngữ lập trình quản trị cơ sở dữ liệu. Quản trị viên cơ sở dữ liệu có thể quản lý và điều khiển các truy cập tới cơ sở dữ liệu thông qua SQL
* SQL được sử dụng như một công cụ giao tiếp với các trình ứng dụng trong hệ thống cơ sở dữ liệu khách – chủ.

## **1.5** **Giới thiệu ngôn ngữ lập trình C#**

C# (đọc là "C thăng" hay "C sharp" *("xi-sáp")*) là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) [hướng đối tượng](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng) được phát triển bởi [Microsoft](https://vi.wikipedia.org/wiki/Microsoft).

Microsoft phát triển C# dựa trên [C++](https://vi.wikipedia.org/wiki/C%2B%2B) và [Java](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)). C# được miêu tả là ngôn ngữ có được sự cân bằng giữa C++, [Visual Basic](https://vi.wikipedia.org/wiki/Visual_Basic), [Delphi](https://vi.wikipedia.org/wiki/Delphi_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) và [Java](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)).

C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng Windows Forms hay WPF ([Windows Presentation Foundatio](https://vi.wikipedia.org/wiki/Windows_Presentation_Foundation)n),…trở nên dễ dàng.

Đặc trưng của ngôn ngữ C#:

* Là một ngôn ngữ thuần hướng đối tượng.
* Là ngôn ngữ khá đơn giản, chỉ có khoảng 80 từ khóa và hơn mười mấy kiểu dữ liệu được dựng sẵn.
* C# không khuyến khích sử dụng con trỏ như trong C++ nhưng nếu bạn thực sự muốn sử dụng thì phải đánh dấu đây là mã không an toàn.
* C# đã loại bỏ đa kế thừa trong C++ mà thay vào đó C# sẽ hỗ trợ thực thi giao diện interface.

Ngoài ra, C# còn có những ưu điểm nổi bật như:

* Gần gũi với các ngôn ngữ lập trình thông dụng ( C++, Java, Pascal).
* Xây dựng dựa trên nền tảng của các ngôn ngữ lập trình mạnh nên thừa hưởng những ưu điểm của những ngôn ngữ đó.
* Cải tiến các khuyết điểm của C/C++ như con trỏ, các hiệu ứng phụ,…
* Dễ tiếp cận, dễ phát triển.
* Được sự chống lưng của .NET Framework.

Đi kèm với những điểm mạnh C# cũng có những nhược điểm:

* Nhược điểm lớn nhất của C# là chỉ chạy trên nền Windows và có cài .NET Framework.

Thao tác đối với phần cứng yếu hơn so với ngôn ngữ khác. Hầu hết phải dựa vào Windows.

## **1.6** **Phân tích thiết kế hệ thống hướng đối tượng và UML**

* **Khái niệm OOAD (Object Oriented Analysis and Design)**

Trong kỹ nghệ phần mềm để sản xuất được một sản phẩm phần mềm người ta chia quá trình phát triển sản phẩm ra nhiều giai đoạn như thu thập và phân tích yêu cầu, phân tích và thiết kế hệ thống, phát triển (coding), kiểm thử, triển khai và bảo trì. Trong đó, giai đoạn phân tích, thiết kế bao giờ cũng là giai đoạn khó khăn và phức tạp nhất. Giai đoạn này giúp chúng ta hiểu rõ yêu cầu đặt ra, xác định giải pháp, mô tả chi tiết giải pháp. Nó trả lời 2 câu hỏi What (phần mềm này làm cái gì?) và How (làm nó như thế nào?).

Để phân tích và thiết kế một phần mềm thì có nhiều cách làm, một trong những cách làm đó là xem hệ thống gồm những đối tượng sống trong đó và tương tác với nhau. Việc mô tả được tất cả các đối tượng và sự tương tác của chúng sẽ giúp chúng ta hiểu rõ hệ thống và cài đặt được nó. Phương thức này gọi là Phân tích thiết kế hướng đối tượng (OOAD)

* **Khái niệm về UML (Unified Modeling Language)**

UML là ngôn ngữ mô hình hóa hợp nhất dùng để biểu diễn hệ thống. Nói một cách đơn giản là nó dùng để tạo ra các bản vẽ nhằm mô tả thiết kế hệ thống. Các bản vẽ này được sử dụng để các nhóm thiết kế trao đổi với nhau cũng như dùng để thi công hệ thống (phát triển), thuyết phục khách hàng, các nhà đầu tư...

* **Tại sao lại là OOAD và UML?**

OOAD cần các bản vẽ để mô tả hệ thống được thiết kế, còn UML là ngôn ngữ mô tả các bản vẽ nên cần nội dung thể hiện. Do vậy, chúng ta phân tích và thiết kế theo hướng đối tượng và sử dụng UML để biểu diễn các thiết kế đó nên chúng thường đi đôi với nhau.

* **OOAD sử dụng UML**

UML sử dụng để vẽ cho nhiều lĩnh vực khác nhau như phần mềm, cơ khí, xây dựng,…trong phạm vi các bài viết này chúng ta chỉ nghiên cứu cách sử dụng UML cho phân tích và thiết kế hướng đối tượng trong ngành phần mềm. OOAD sử dụng UML bao gồm các thành phần sau:

* ***View (góc nhìn)***

Mỗi góc nhìn như thầy bói xem voi, nó không thể hiện hết hệ thống nhưng thể hiện rõ hệ thống ở một khía cạnh. Chính vì thế trong xây dựng có bản vẽ kiến trúc (nhìn về mặt kiến trúc), bản vẽ kết cấu (nhìn về mặt kết cấu), bản vẽ thi công (nhìn về mặt thi công).



Hình 1.4 Các View trong OOAD sử dụng UML

Trong đó,

* + UseCase View: cung cấp góc nhìn về các ca sử dụng giúp chúng ta hiểu hệ thống có gì? ai dùng và dùng nó như thế nào.
  + Logical View: cung cấp góc nhìn về cấu trúc hệ thống, xem nó được tổ chức như thế nào. Bên trong nó có gì.
  + Process View: cung cấp góc nhìn động về hệ thống, xem các thành phần trong hệ thống tương tác với nhau như thế nào.
  + Component View: Cũng là một góc nhìn về cấu trúc giúp chúng ta hiểu cách phân bổ và sử dụng lại các thành phần trong hệ thống ra sao.
  + Deployment View: cung cấp góc nhìn về triển khai hệ thống, nó cũng ảnh hưởng lớn đến kiến trúc hệ thống.

Tập hợp các góc nhìn này sẽ giúp chúng ta hiểu rõ hệ thống cần phân tích, thiết kế. Trong hình trên chúng ta thấy góc nhìn Use Case View nằm ở giữa và chi phối tất cả các góc nhìn còn lại. Chính vì thế chúng ta thường thấy các tài liệu nói về 4 view + 1 chứ không phải 5 view nhằm nhấn mạnh vai trò của UseCase View.

* ***Diagram (Bản vẽ)***

Diagram các bạn có thể dịch là sơ đồ. Tuy nhiên ở đây chúng ta sử dụng từ bản vẽ cho dễ hình dung. Các bản vẽ được dùng để thể hiện các góc nhìn của hệ thống.



Hình 1.5 Các bản vẽ trong OOAD sử dụng UML

Trong đó,

* Use Case Diagram: bản vẽ mô tả về ca sử dụng của hệ thống. Bản vẽ này sẽ giúp chúng ta biết được ai sử dụng hệ thống, hệ thống có những chức năng gì. Lập được bản vẽ này bạn sẽ hiểu được yêu cầu của hệ thống cần xây dựng.
* Class Diagram: bản vẽ này mô tả cấu trúc của hệ thống, tức hệ thống được cấu tạo từ những thành phần nào. Nó mô tả khía cạnh tĩnh của hệ thống.
* Object Diagram: Tương tự như Class Diagram nhưng nó mô tả đến đối tượng thay vì lớp (Class).
* Sequence Diagarm: là bản vẽ mô tả sự tương tác của các đối tượng trong hệ thống với nhau được mô tả tuần tự các bước tương tác theo thời gian.
* Collaboration Diagram: tương tự như sequence Diagram nhưng nhấn mạnh về sự tương tác thay vì tuần tự theo thời gian.
* State Diagram: bản vẽ mô tả sự thay đổi trạng thái của một đối tượng. Nó được dùng để theo dõi các đối tượng có trạng thái thay đổi nhiều trong hệ thống.
* Activity Diagram: bản vẽ mô tả các hoạt động của đối tượng, thường được sử dụng để hiểu về nghiệp vụ của hệ thống.
* Component Diagram: bản vẽ mô tả về việc bố trí các thành phần của hệ thống cũng như việc sử dụng các thành phần đó.
* Deployment Diagram: bản vẽ mô tả việc triển khai của hệ thống như việc kết nối, cài đặt, hiệu năng của hệ thống v.v…
* ***Notations (các ký hiệu)***

Notations là các ký hiệu để vẽ, nó như từ vựng trong ngôn ngữ tự nhiên. Bạn phải biết từ vựng thì mới ghép thành câu, thành bài được. Chúng ta sẽ tìm hiểu kỹ các notations trong từng bản vẽ sau này.



Hình 1.6 Ký hiệu về Use Case



Hình 1.7 Ký hiệu về Class



Hình 1.8 Ký hiệu về Actor

* ***Mechanisms (Rules)***

Mechanisms là các qui tắc để lập nên bản vẽ, mỗi bản vẽ có qui tắc riêng và bạn phải nắm được để tạo nên các bản vẽ thiết kế đúng. Các qui tắc này chúng ta sẽ bàn kỹ trong các bài về các bản vẽ.

**CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

**2.1 Mô tả bài toán**

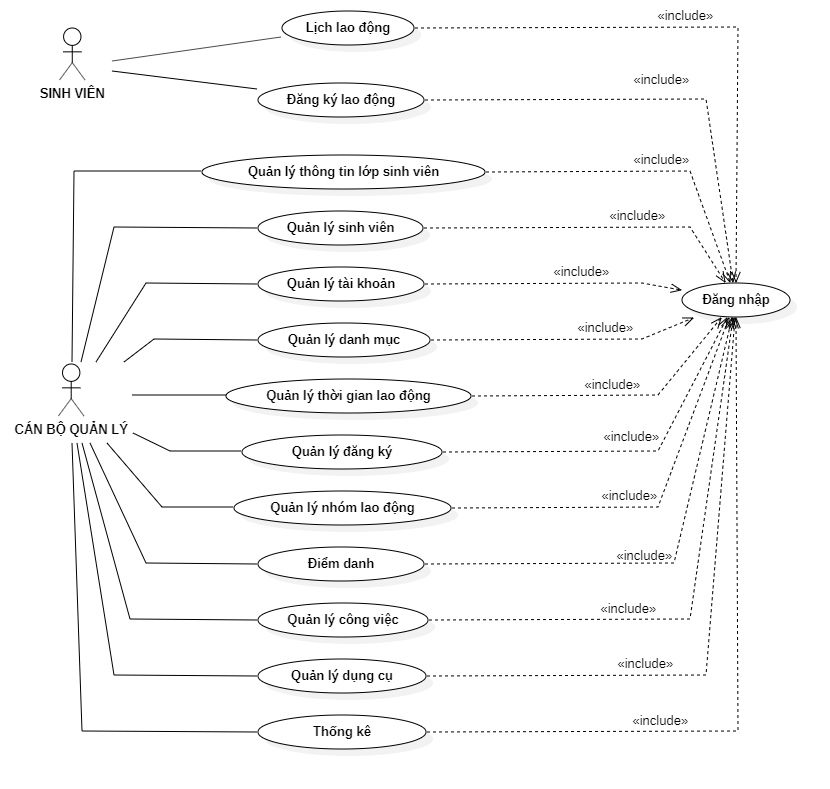
Sự phát triển của khoa học kỹ thuật và công nghệ là nền tảng cho việc ra đời những website có đầy đủ các chức năng quản lý, thống kê, tính toán chính xác và hiệu quả. (Nhất là trong việc quản lý sinh viên về số ngày, giờ lao động tại Trung tâm Quản lý Dịch vụ trực thuộc Trường Đại học Đồng Tháp. Mỗi năm học, Trung tâm phải quản lý, xếp lịch và đánh giá trên 1700 số giờ, ngày lao động của sinh viên toàn trường. Chúng tôi nhận thấy, nếu có website hỗ trợ việc đăng ký lao động trực tuyến và quản lý việc phân công công việc sẽ cải thiện được hiệu quả làm việc cho trung tâm và tạo thuận lợi cho sinh viên trong việc lựa chọn đăng ký thời gian lao động hợp lý. Ngoài ra, Website giúp cán bộ quản lý sinh viên biết chính xác số lần lao động của lớp hoặc kết quả lao động của từng sinh viên. Nghiệp vụ được mô tả như sau:

* Mỗi lần đăng ký lao động, cán bộ quản lý sẽ đưa ra một phiếu bao gồm các ngày trong tuần. (1 ngày: 2 buổi, 1 buổi sẽ cho phép tối đa 2 lớp đăng ký, 1 lớp chỉ được đăng ký 2 buổi/tuần).
* Sau khi đăng ký , tới ngày lao động phó lao động sẽ báo số lượng sinh viên có mặt rồi cán bộ quản lý tiến hành phân công công việc và giao dụng cụ cho lớp.
* Kết thúc buổi lao động, phó lao động kiểm tra dụng cụ và giao lại cho cán bộ quản lý. Từ đó, cán bộ quản lý sẽ ghi lại thông tin buổi lao động vào sổ. Trường hợp nếu có mất hoặc làm hỏng dụng cụ lao động, thì bên quản lý sẽ ghi nhận và bên phía lớp lao động phải đền bù. (số tiền tương ứng hoặc mua lại dụng cụ mới).
* Hằng năm, cán bộ quản lý sẽ có nhiệm vụ thống kê số ngày lao động của các lớp để ghi nhận về việc lao động đủ hay thiếu của mỗi lớp trong mỗi năm học làm cơ sở cho việc xét tốt nghiệp của các lớp sinh viên khi sắp kết thúc khóa đào tạo.

**2.2 Sơ đồ Use Case**

**2.2.1 Sơ đồ tổng quát**

Khái quát chức năng chính của hệ thống. Usecase dưới đây mô tả các chức năng chính một cách tổng quát để có thể dễ dàng nhìn thấy được trên quan điểm của các tác nhân.



Hình 2.1 Sơ đồ Use Case tổng quát

**2.2.2 Sơ đồ thành phần**

**2.2.2.1 Use Case Đăng ký lao động**

**Tác nhân:** Sinh viên

**Điều kiện:** phải đăng nhập vào hệ thống

**Mô tả:** Sinh viên sau khi đi đăng nhập vào hệ thống được quyền thực hiện chức năng đăng ký lao động. Khi đó sinh viên có thể thêm hoặc xóa thông tin đăng ký khi cần.

****

Hình 2.2 Sơ đồ Use Case đăng ký lao động

**2.2.2.2 Use Case Quản lý thông tin lớp sinh viên**

**Tác nhân:** Admin

**Điều kiện:** phải đăng nhập vào hệ thống

**Mô tả:** Admin sau khi đi đăng nhập vào hệ thống được quyền thực hiện chức năng quản lý thông tin lớp sinh viên. Admin có thể thực hiện việc thêm, sửa hoặc xóa các thông tin của lớp sinh viên.

****

Hình 2.3 Sơ đồ Use Case quản lý lớp sinh viên

**2.2.2.3 Use Case Quản lý tài khoản**

**Tác nhân:** Admin

**Điều kiện:** phải đăng nhập vào hệ thống

**Mô tả:** Admin sau khi đi đăng nhập vào hệ thống được quyền thực hiện chức năng quản lý tài khoản. Admin có thể thực hiện chức năng thêm, sửa hoặc xóa thông tin của các tài khoản.



Hình 2.4 Sơ đồ Use Case quản lý tài khoản

**2.2.2.4 Use Case Quản lý đăng ký**

**Tác nhân:** Admin

**Điều kiện:** phải đăng nhập vào hệ thống

**Mô tả:** Admin sau khi đi đăng nhập vào hệ thống được quyền thực hiện chức năng quản lý đăng ký. Admin có thể thực hiện sửa hoặc xóa thông tin đăng ký lao động của sinh viên.

****

Hình 2.5 Sơ đồ Use Case quản lý đăng ký

**2.2.2.5 Use Case Quản lý công việc**

**Tác nhân:** Admin

**Điều kiện:** phải đăng nhập vào hệ thống

**Mô tả:** Admin sau khi đi đăng nhập vào hệ thống được quyền thực hiện chức năng quản lý công việc. Admin có thể thực hiện việc thêm, sửa hoặc xóa thông tin của các công việc.

**Diagram

Description automatically generated**

Hình 2.6 Sơ đồ Use Case quản lý công việc

**2.2.2.6 Use Case Quản lý dụng cụ**

**Tác nhân:** Admin

**Điều kiện:** phải đăng nhập vào hệ thống

**Mô tả:** Admin sau khi đi đăng nhập vào hệ thống sẽ được quyền thực hiện chức năng quản lý dụng cụ. Admin có thể thực hiện việc thêm, sửa hoặc xóa thông tin dụng cụ lao động khi cần thiết.



Hình 2.7 Sơ đồ Use Case quản lý dụng cụ

### **2.2.2.7 Use Case quản lý danh mục**

**Tác nhân:** Admin

**Điều kiện:** phải đăng nhập vào hệ thống

**Mô tả:** Admin sau khi đi đăng nhập vào hệ thống sẽ được quyền thực hiện chức năng quản lý danh mục. Admin có thể thực hiện việc thêm, sửa hoặc xóa thông tin danh mục khi cần thiết.

Diagram

Description automatically generated

Hình 2.8 Sơ đồ Use Case quản lý danh mục

### **2.2.2.8 Use Case Quản lý sinh viên**

**Tác nhân:** Admin

**Điều kiện:** phải đăng nhập vào hệ thống

**Mô tả:** Admin sau khi đi đăng nhập vào hệ thống sẽ được quyền thực hiện chức năng quản lý sinh viên. Admin có thể thực hiện việc thêm, sửa hoặc xóa thông tin của sinh viên khi cần thiết.

Diagram

Description automatically generated

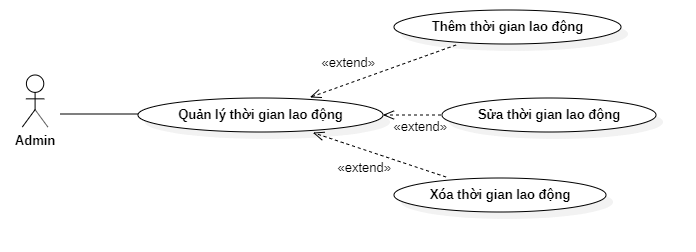
Hình 2.9 Sơ đồ Use Case quản lý sinh viên

### **2.2.2.9 Use Case Quản lý thời gian lao động**

**Tác nhân:** Admin

**Điều kiện:** phải đăng nhập vào hệ thống

**Mô tả:** Admin sau khi đi đăng nhập vào hệ thống sẽ được quyền thực hiện chức năng quản lý thời gian lao động. Admin có thể thực hiện việc thêm, sửa hoặc xóa thông tin về thời gian cho phép đăng ký lao động.



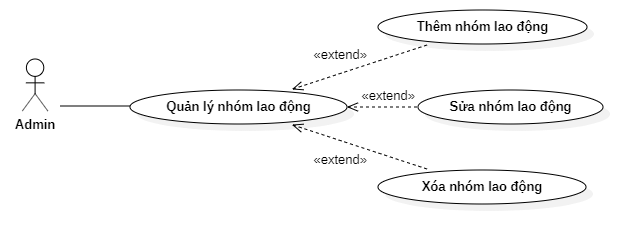
Hình 2.10 Sơ đồ Use Case quản lý thời gian lao động

### **2.2.2.10 Use Case Quản lý nhóm lao động**

**Tác nhân:** Admin

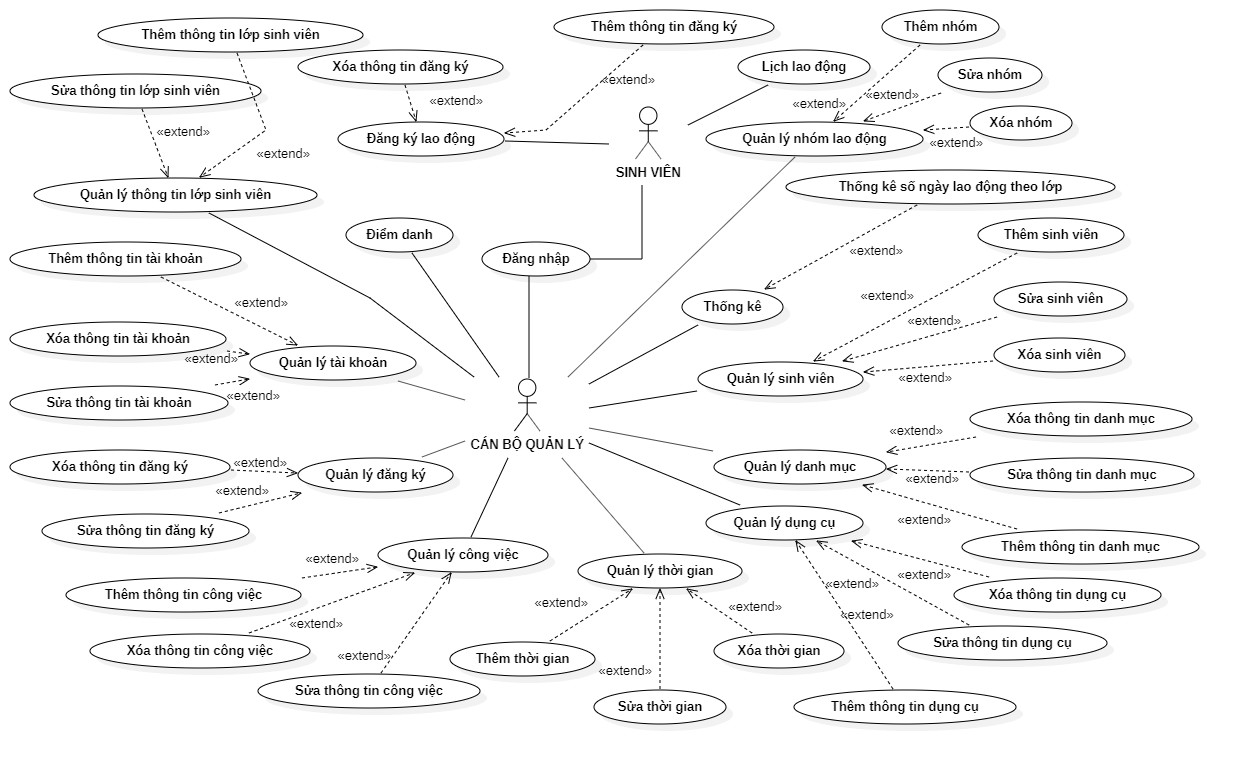
**Điều kiện:** phải đăng nhập vào hệ thống

**Mô tả:** Admin sau khi đi đăng nhập vào hệ thống sẽ được quyền thực hiện chức năng quản lý nhóm lao động. Admin có thể thực hiện việc thêm, sửa hoặc xóa thông tin các nhóm lao động.



Hình 2.11 Sơ đồ Use Case quản lý nhóm lao động

**2.2.3 Use Case chi tiết**

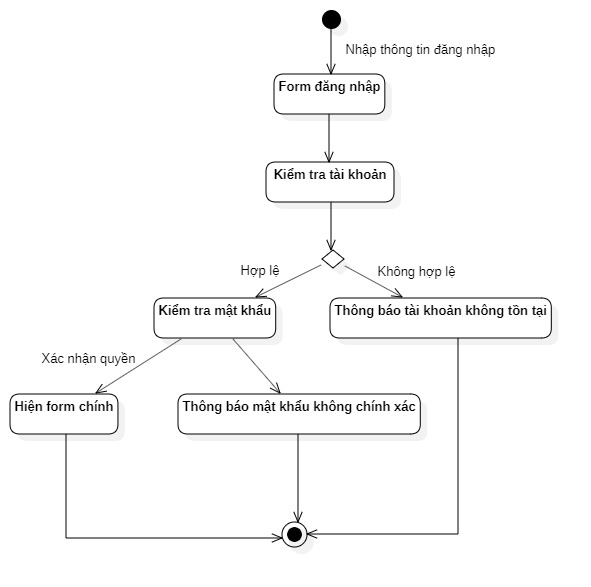


Hình 2.12 Sơ đồ Use Case chi tiết

**2.3 Sơ đồ hoạt động**

**2.3.1 Hoạt động đăng nhập**

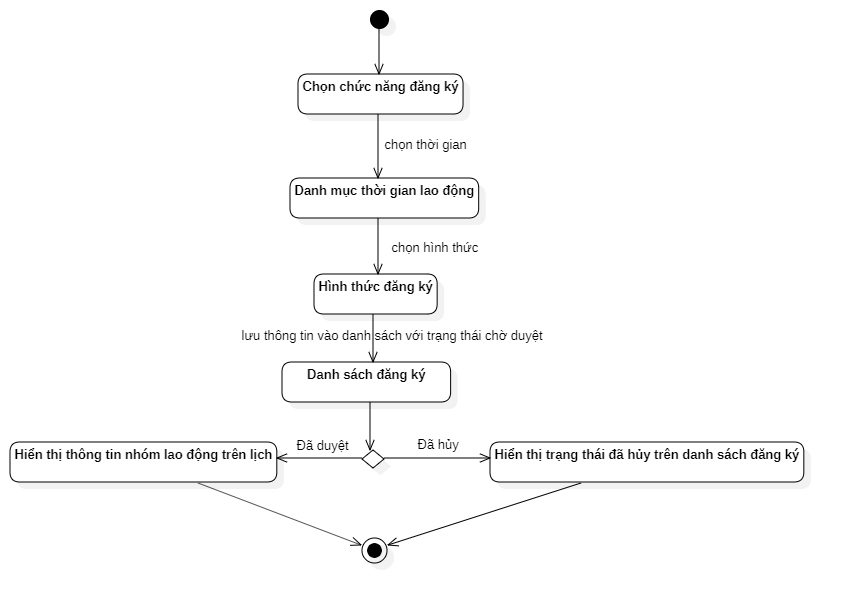
Người dùng yêu cầu đăng nhập, ở form đăng nhập người dùng sẽ nhập tên tài khoản và mật khẩu. Nếu tài khoản và mật khẩu hợp lệ thì “đăng nhập thành công”, hiện form hệ thống và kết thúc. Ngược lại, thông báo mật khẩu không chính xác và kết thúc. Còn nếu tài khoản không hợp lệ thì thông báo tài khoản không tồn tại và kết thúc.



Hình 2.13 Sơ đồ hoạt động Đăng nhập

**2.3.2 Hoạt động đăng ký**

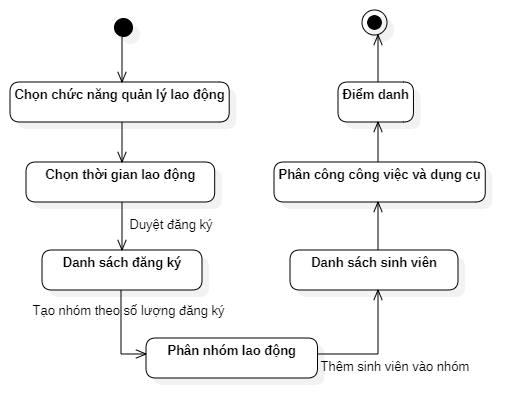
Người dùng sẽ đăng nhập với quyền user (sinh viên), chọn chức năng “Đăng ký lao động”. Hệ thống sẽ hiển thị danh sách các mốc thời gian cho phép người dùng đăng ký lao động, chọn mốc thời gian và chọn hình thức đăng ký lao động. Khi đó, thông tin đăng ký sẽ được hiển thị lên danh sách với trạng thái là Chờ duyệt. Sau khi cán bộ quản lý xem xét, duyệt đăng ký và phân công thì thông tin sẽ được hiển thị trên lịch lao động. Ngược lại, thông tin trên danh sách đăng ký sẽ chuyển trạng thái thành Đã hủy.

****

Hình 2.14 Sơ đồ hoạt động Đăng ký lao động

**2.3.3 Hoạt động quản lý đăng ký**

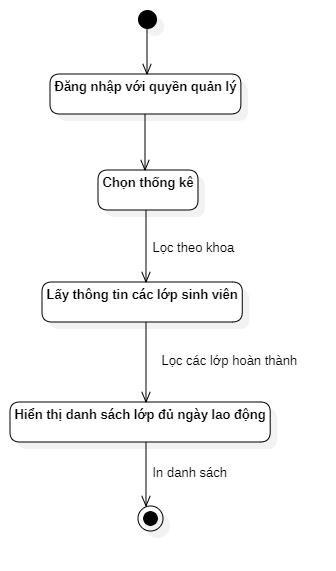
Người dùng sẽ đăng nhập với quyền admin (cán bộ quản lý), chọn chức năng “Quản lý lao động”. Người dùng chọn chức năng là quản lý lao động, tiếp đó chọn mốc thời gian cần quản lý, rồi xem xét và duyệt các phiếu đăng ký của sinh viên. Sau đó, người dùng sẽ phân nhóm lao động và thêm sinh viên vào các nhóm rồi phân công công việc, dụng cụ lao động. Cuối cùng là điểm danh và kết thúc.



Hình 2.15 Sơ đồ hoạt động quản lý lao động

**2.3.4 Hoạt động thống kê**

Người dùng sẽ đăng nhập với quyền admin (cán bộ quản lý), chọn chức năng “Thống kê”.



Hình 2.16 Sơ đồ hoạt động thống kê

**2.4 Sơ đồ tuần tự**

**2.4.1 Sơ đồ tuần tự đăng nhập**

**Diagram

Description automatically generated**

Hình 2.17 Sơ đồ tuần tự đăng nhập

**2.4.2 Sơ đồ tuần tự đăng ký lao động**

****

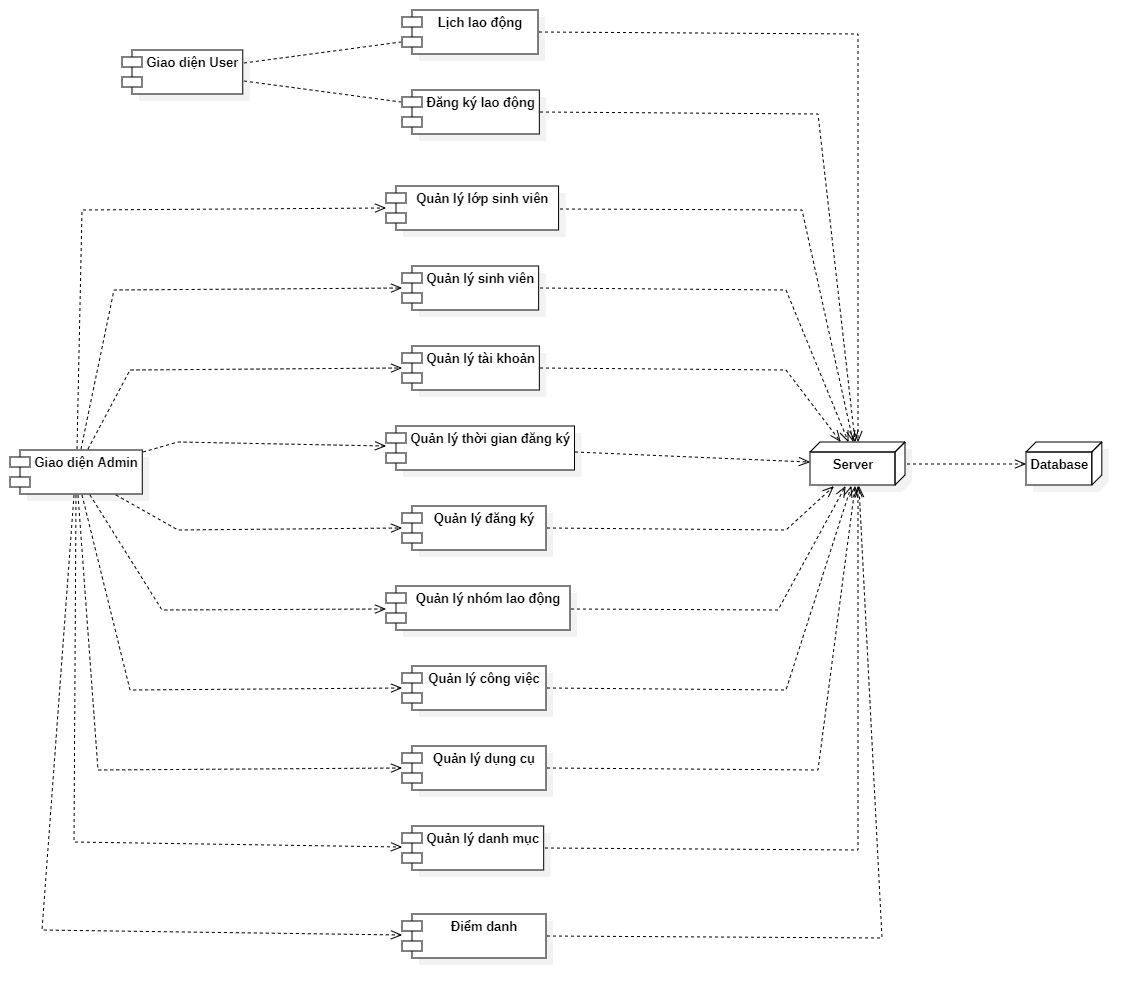
Hình 2.18 Sơ đồ tuần tự đăng ký lao động

**2.4.3 Sơ đồ tuần tự thống kê**

****

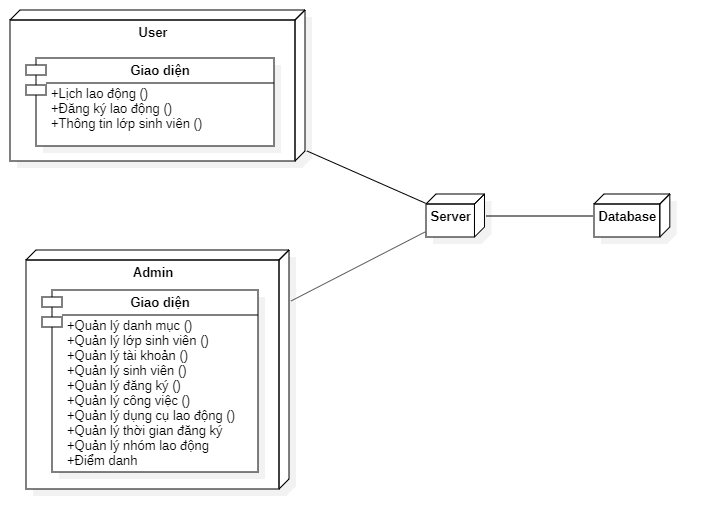
Hình 2.19 Sơ đồ tuần tự thống kê

**2.5 Sơ đồ thành phần**



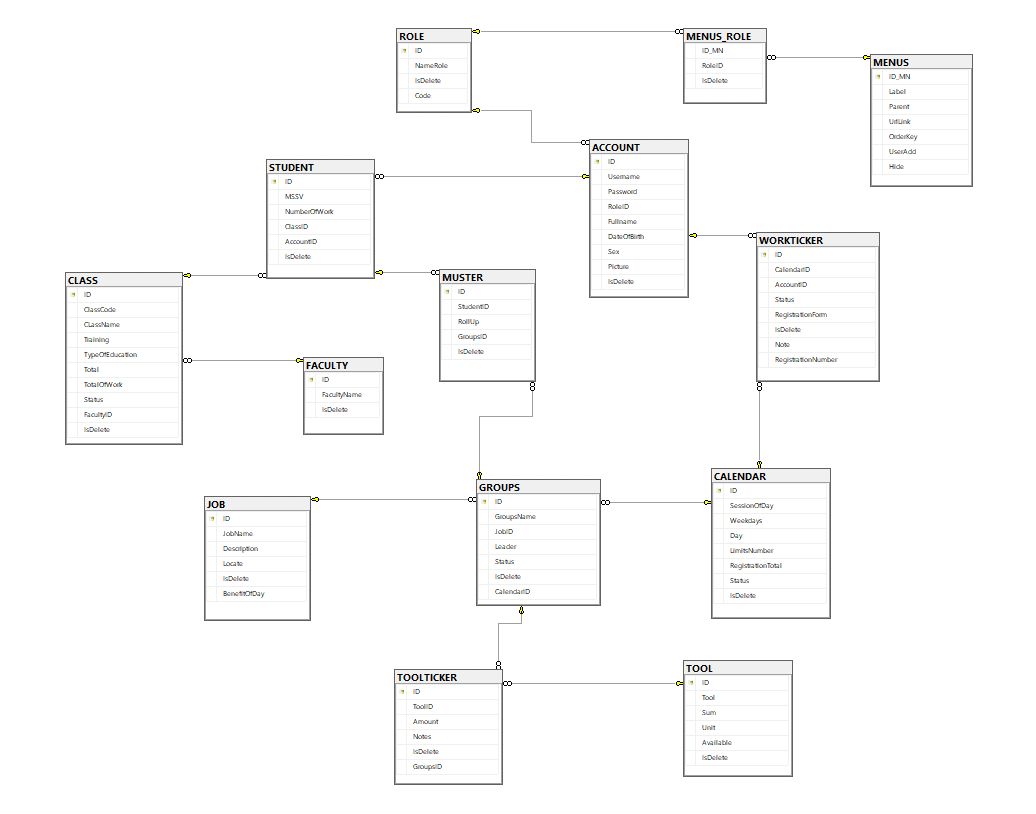
Hình 2.20 Sơ đồ thành phần

**2.6 Sơ đồ triển khai**



Hình 2.21 Sơ đồ triển khai

**2.7 Thiết kế cơ sở dữ liệu**

****

Hình 20 Diagram cơ sở dữ liệu

**CHƯƠNG 3: XÂY DỰNG WEBSITE QUẢN LÝ LAO ĐỘNG TRƯỜNG ĐẠI HỌC ĐỒNG THÁP**

* 1. **Giao diện chính**

**PHẦN KẾT LUẬN**

**-----oOo-----**

1. **Kết quả đạt được**

* Hoàn thành được mục tiêu đặt ra: Website quản lý lao động Trường Đại học Đồng Tháp
* Hiểu thêm về ASP.Net Core và Entity Framework Core.
* Phát triển khả năng lập trình trên C#.
* Nắm được quá trình phân tích, thiết kế, cài đặt một website hoàn chỉnh.

1. **Hướng phát triển**

* Tối ưu hóa tốc độ để tăng trải nghiệm người dùng.
* Mở rộng phạm vi đề tài và nghiên cứu xây dựng hoàn thiện website phù hợp với yêu cầu thực tế.

**TÀI LIỆU THAM KHẢO**

**1. Giáo trình, bài giảng**

[1]. **Phạm Nguyễn Cương, Hồ Tường Vinh**, *“Giáo trình Phân tích và thiết kế hệ thống Hướng đối tượng sử dụng UML”*, ĐHKHTN TP HCM, 2013.

[2]. Tài liệu lập trình C# – Trường Đại Học Khoa Học Tự Nhiên - Đại học quốc gia TP. HCM.

**2. Website**

[3]. https://docs.microsoft.com/vi-vn/aspnet/core/?view=aspnetcore-3.0.

[4]. https://docs.microsoft.com/en-us/ef/core/

[5]. <https://www.entityframeworktutorial.net/efcore/entity-framework-core.aspx>

[6]. https://docs.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection?view=aspnetcore-5.0