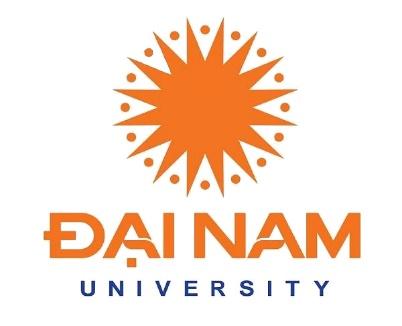
TRƯỜNG ĐẠI HỌC ĐẠI NAM

**KHOA CÔNG NGHỆ THÔNG TIN**

**----------------------------------**

**BÁO CÁO THỰC TẬP**

**Đề tài:**

**XÂY DỰNG PHẦN MỀM QUẢN LÝ CÔNG NHÂN**

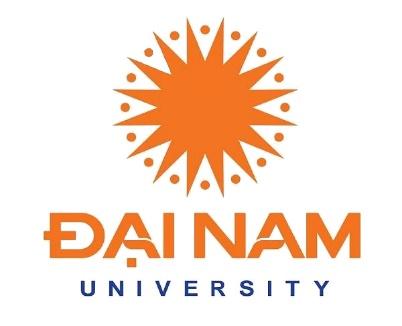
|  |  |  |
| --- | --- | --- |
| Sinh viên thực hiện | : |  |
| Giảng viên hướng dẫn | : |  |
| Lớp | : |  |
| MSV | : |  |
| Hệ | : |  |

Hà Nội, Tháng 2 năm 2023

TRƯỜNG ĐẠI HỌC ĐẠI NAM

**KHOA CÔNG NGHỆ THÔNG TIN**

**----------------------------------**



**Đề tài:**

**XÂY DỰNG PHẦN MỀM QUẢN LÝ CÔNG NHÂN**

CHUYÊN NGÀNH: CÔNG NGHỆ THÔNG TIN

Người hướng dẫn:

Hà Nội, Tháng 2 năm 2023

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

Hà Nội, ngày … tháng … năm 2023

**Giảng viên hướng dẫn**

**NHẬN XÉT CỦA HỘI ĐỒNG PHẢN BIỆN**

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..…..

Hà Nội, ngày … tháng … năm 2023

**T/M hội đồng phản biện**

**LỜI CAM ĐOAN**

Tôi xin cam đoan đây là công trình nghiên cứu khoa học độc lập của riêng tôi dưới sự hướng dẫn khoa học của ThS. Lê Chí Luận. Các nội dung nghiên cứu, số liệu và thực nghiệm trong đề tài “Xây dựng phần mềm quản lý công nhân” của tôi là hoàn toàn trung thực và chưa từng được công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được cá nhân tôi thu thập từ các nguồn khác nhau có ghi rõ nguồn gốc. Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung bài tiểu luận của mình.

Hà Nội, ngày … tháng … năm 2023

**Sinh viên thực hiện**

**LỜI CẢM ƠN**

Trân trọng cảm ơn các thầy cô giáo trong ban lãnh đạo trường Đại học Đại Nam, các thầy cô giáo là giảng viên khoa Công nghệ thông tin đã tạo điều kiện và môi trường để tôi được thực hiện đề tài nghiên cứu thực tập tốt nghiệp này.

Tôi xin được gửi lời cảm ơn chân thành tới thầy giáo ThS. Lê Chí Luận đã luôn dõi theo, tận tình chỉ dẫn, chỉ bảo, hỗ trợ kịp thời và chia sẻ cho tôi những kinh nghiệm quý báu để tôi có thể hoàn thành tốt đề tài nghiên cứu này.

Đặc biệt, xin được gửi lời cảm ơn tới các thầy cô giáo, các anh chị phòng Công tác Sinh viên trường Đại học Đại Nam đã tạo điều kiện cho tôi được tìm hiểu về nghiệp vụ, cũng như cung cấp cho tôi các tài liệu, văn bản cần thiết phục vụ cho đề tài.

Trân trọng cảm ơn!

Hà Nội, ngày … tháng … năm 2023

**Sinh viên thực hiện**

**MỤC LỤC**

[**DANH MỤC HÌNH ẢNH 8**](#_Toc128606212)

[**DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT 9**](#_Toc128606213)

[**LỜI MỞ ĐẦU 11**](#_Toc128606214)

[**CHƯƠNG 1. GIỚI THIỆU VỀ CÔNG TY THỰC TẬP 14**](#_Toc128606215)

[1.1. Tổng quan về Evotek 14](#_Toc128606216)

[1.2. Lĩnh vực hoạt động 15](#_Toc128606217)

[**CHƯƠNG 2: KIẾN THỨC CƠ BẢN 17**](#_Toc128606218)

[2.1. Giới thiệu về C# và ASP.NET CORE 17](#_Toc128606219)

[2.1.1. Giới thiệu về ngôn ngữ lập trình C# 17](#_Toc128606220)

[2.1.1.1. C# là gì? 17](#_Toc128606221)

[2.1.1.2. Đặc trưng của ngôn ngữ lập trình C# 18](#_Toc128606222)

[2.1.2. Giới thiệu về ASP.NET CORE 19](#_Toc128606223)

[2.1.2.1. Tổng quan về ASP.NET CORE 19](#_Toc128606224)

[2.1.2.2. Tại sao sử dụng ASP.NET CORE 20](#_Toc128606225)

[2.1.3. Giới thiệu mô hình Model – View – Controller (MVC) 21](#_Toc128606226)

[2.1.4. Giới thiệu về jQuery Ajax 23](#_Toc128606227)

[2.1.4.1. AJAX là gì? 23](#_Toc128606228)

[2.1.4.2. jQuery là gì? 23](#_Toc128606229)

[2.1.4.3. jQuery làm được gì cho AJAX? 24](#_Toc128606230)

[2.1.5. Thư viện đọc, ghi dữ liệu từ bảng tính EPPlus 24](#_Toc128606231)

[2.1.6. Giới thiệu Visual Studio 2019 25](#_Toc128606232)

[2.2. Hệ quản trị cơ sở dữ liệu SQL Server 28](#_Toc128606233)

[2.2.1. Giới thiệu về cơ sở dữ liệu 28](#_Toc128606234)

[2.2.1.1. Dữ liệu là gì? 28](#_Toc128606235)

[2.2.1.2. Cơ sở dữ liệu là gì? 28](#_Toc128606236)

[2.2.1.3. Các mô hình cơ sở dữ liệu 29](#_Toc128606237)

[2.2.2. Giới thiệu công cụ Microsoft SQL Server Management Studio 18 32](#_Toc128606238)

[2.2.2.1. Lịch sử ra đời SQL Server và các phiên bản 33](#_Toc128606239)

[2.2.2.2. Các thành phần cơ bản trong SQL Server 34](#_Toc128606240)

[2.2.2.3. Database Engine 35](#_Toc128606241)

[2.2.2.4. Replication 35](#_Toc128606242)

[2.2.2.5. Integration Services (DTS) 35](#_Toc128606243)

[2.2.2.6. Analysis Services 35](#_Toc128606244)

[2.2.2.7. Notification Services 36](#_Toc128606245)

[2.2.2.8. Reporting  Services 36](#_Toc128606246)

[2.2.2.9. Full Text Search Service 36](#_Toc128606247)

[2.2.2.10. Service Broker 36](#_Toc128606248)

[2.2.3. Mục đích sử dụng cơ sở dữ liệu 36](#_Toc128606249)

[2.3. Kiểm tra kết nối SQL Server và Visual Studio 2019 37](#_Toc128606250)

[**CHƯƠNG 3: NỘI DUNG THỰC HIỆN 43**](#_Toc128606251)

[3.1. Phân tích yêu cầu bài toán và thiết kế database 43](#_Toc128606252)

[3.1.1. Các chức năng cơ bản của hệ thống 43](#_Toc128606253)

[3.1.2. Mô hình phân cấp chức năng 43](#_Toc128606254)

[3.1.3. Phát hiện thực thể 43](#_Toc128606255)

[3.1.4. Mô hình ERD 43](#_Toc128606256)

[3.1.5. Mô hình quan hệ 44](#_Toc128606257)

[3.1.6. Ý tưởng thiết kế 44](#_Toc128606258)

[3.2. Phần mềm quản lý công nhân 49](#_Toc128606259)

[*3.2.1. Giao diện 49*](#_Toc128606260)

[*3.2.2.* Khu vực menu 49](#_Toc128606261)

[3.2.3. Khu vực điều hướng 49](#_Toc128606262)

[3.2.4. Giao diện quản lý công trình 50](#_Toc128606264)

[3.2.5. Giao diện quản lý nhân viên 53](#_Toc128606265)

[3.2.6. Giao diện quản lý phòng ban 54](#_Toc128606266)

[3.2.7. Giao diện quản lý chức vụ 54](#_Toc128606267)

[3.2.8. Giao diện phân công 55](#_Toc128606268)

[3.2.9. Giao diện quản lý tài khoản 55](#_Toc128606269)

[3.3. Cài đặt và kiểm thử 57](#_Toc128606270)

[3.3.1. Cài đặt chương trình 57](#_Toc128606271)

[3.3.2. Kiểm thử chương trình 62](#_Toc128606272)

[3.3.3. Đánh giá hệ thống 62](#_Toc128606273)

[**CHƯƠNG 4. KỸ NĂNG MỀM 63**](#_Toc128606274)

[4.1. Kỹ năng mềm là gì? 63](#_Toc128606275)

[4.2. Các kỹ năng mềm học được tại công ty 63](#_Toc128606276)

[**CHƯƠNG 5: KẾT LUẬN 65**](#_Toc128606277)

[TÀI LIỆU THAM KHẢO 67](#_Toc128606281)

# DANH MỤC HÌNH ẢNH

[Hình 1‑1:Mẫu điểm danh của phòng phòng CTSV](file:///F:\CNTT11-01_DNU\NGHIEN_CUU_KHOA_HOC\ThucTap\ATD\ATD_ad\BaoCaoThucTap3_2405.docx#_Toc76334207)

[Hình 2‑1: Quá trình chuyển đổi MSIL code thành Native code 17](file:///F:\CNTT11-01_DNU\NGHIEN_CUU_KHOA_HOC\ThucTap\ATD\ATD_ad\BaoCaoThucTap3_2405.docx#_Toc76334208)

[Hình 2‑2: Logo ASP.NET CORE 20](#_Toc76334209)

[Hình 2‑3: Mô hình MVC 22](#_Toc76334210)

[Hình 2‑4: Giao diện Visual Studio Community 2015 26](#_Toc76334211)

[Hình 2‑5: Ví dụ về cơ sở dữ liệu 29](file:///F:\CNTT11-01_DNU\NGHIEN_CUU_KHOA_HOC\ThucTap\ATD\ATD_ad\BaoCaoThucTap3_2405.docx#_Toc76334212)

[Hình 2‑6: Mô hình dữ liệu phân cấp 30](file:///F:\CNTT11-01_DNU\NGHIEN_CUU_KHOA_HOC\ThucTap\ATD\ATD_ad\BaoCaoThucTap3_2405.docx#_Toc76334213)

[Hình 2‑7: Mô hình dữ liệu mạng 31](file:///F:\CNTT11-01_DNU\NGHIEN_CUU_KHOA_HOC\ThucTap\ATD\ATD_ad\BaoCaoThucTap3_2405.docx#_Toc76334214)

[Hình 2‑8: Mô hình dữ liệu quan hệ 31](#_Toc76334215)

[Hình 2‑9: Mô hình dữ liệu hướng đối tượng 32](file:///F:\CNTT11-01_DNU\NGHIEN_CUU_KHOA_HOC\ThucTap\ATD\ATD_ad\BaoCaoThucTap3_2405.docx#_Toc76334216)

[Hình 2‑10: Các thành phần trong SQL Server 34](#_Toc76334217)

[Hình 2‑11: Đăng nhập SQL Server 38](#_Toc76334218)

[Hình 2‑12: Tạo database 38](#_Toc76334219)

[Hình 2‑13: Tạo table trong database 39](#_Toc76334220)

[Hình 2‑14: Thêm dữ liệu cho table 39](#_Toc76334221)

[Hình 2‑15: Kết nối SQL Server với Visual Studio - 01 40](#_Toc76334222)

[Hình 2‑16: Kết nối SQL Server với Visual Studio - 02 40](#_Toc76334223)

[Hình 2‑17: Kết nối SQL Server với Visual Studio - 03 41](#_Toc76334224)

[Hình 2‑18: Kết nối SQL Server với Visual Studio - 04 41](#_Toc76334225)

[Hình 2‑19: Kết nối SQL Server với Visual Studio - 05 42](#_Toc76334226)

# DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Ký hiệu / Viết tắt** | **Ý nghĩa** |
| 2 | QĐ | Quyết định |
| 3 | QĐ-TTg | Quyết định – Thủ tướng |
| 4 | ĐHĐN | Đại học Đại Nam |
| 5 | TGĐ | Tổng giám đốc |
| 6 | Bộ GD&ĐT | Bộ Giáo dục và Đào tạo |
| 7 | TS | Tiến Sỹ |
| 8 | ThS | Thạc Sỹ |
| 9 | VNĐ | Việt Nam đồng |
| 10 | UBND | Ủa ban nhân dân |
| 11 | ha | Hecta (đơn vị đo lường) |
| 12 | HĐQT | Hội đồng quản trị |
| 13 | & | Và |
| 14 | NCKH | Nghiên cứu khoa học |
| 15 | SĐH | Sau đại học |
| 16 | CNTT | Công nghệ thông tin |
| 17 | KĐCL | Kiểm định chất lượng |
| 18 | NCKH | Nghiên cứu khoa học |
| 19 | ĐH | Đại học |
| 20 | THPT | Trung học phổ thông |
| 21 | TCCN | Trung cấp chuyên nghiệp |
| 22 | SPA | Single-page Applicatión |
| 23 | MVC | Model View Controller |
| 24 | IoT | Internet of Things |
| 25 | WPF | Windows Presentation Foundation |
| 26 | CLR | Common Language Runtime |
| 27 | OOP | Object-oriented Programming |
| 28 | CSDL | Cơ sở dữ liệu |
| 29 | HTML | Hypertext Markup Language |
| 30 | JSP | Java Server Pages |
| 31 | BFD | Business Function Diagram |
| 32 | DFD | Data Flow Diagram |

# LỜI MỞ ĐẦU

Hiện nay, ngành Công nghệ thông tin là một trong những ngành học được chú trọng trong hệ thống đào tạo của trường Đại học Đại Nam cũng như các trường Đại học khác có đào tạo ngành học này. Nó được xem là ngành đào tạo mũi nhọn hướng đến sự phát triển của công nghệ và khoa học kỹ thuật trong thời đại số hóa ngày nay; Công nghệ thông tin cũng là chìa khóa của cuộc cách mạng công nghiệp 4.0 và công cuộc chuyển đổi số.

Theo báo cáo VietnamWorks, trang tuyển dụng nhân sự trung và cao cấp, trong 10 năm (từ năm 2010 đến nay) nhu cầu tuyển dụng của ngành công nghệ thông tin (CNTT) của Việt Nam đã tăng gấp 4 lần, trong đó, 7 nhóm ngành thuộc lĩnh vực CNTT có như cầu tuyển dụng phổ biến gồm: Phát triển phần mềm; Hỗ trợ kỹ thuật; Quản lý dự án/Sản phẩm; Thiết kế trải nghiệm người dùng (UX) và Giao diện (UI); Kỹ sư kiểm định chất lượng sản phẩm QA/QC; Khoa học dữ liệu.

Đối với nhóm ngành phát triển phần mềm luôn đạt mức tăng trưởng gần gấp đôi, chiếm hơn 50% nhu cầu tuyển dụng của toàn ngành CNTT. Đây cũng là nhóm ngành có nguồn nhân lực trí tuệ chất lượng cao, gồm tập hợp các kỹ sư về giải pháp phần mềm quản lý đa chức năng (Mobile, Web, ERP), kỹ năng lập trình ngôn ngữ (JAVA, PHP, NET).

Cũng theo thống kê từ TopDev, chuyên trang tuyển dụng về công nghệ phần mềm, CNTT luôn là một trong số những nghề phát triển nhất trong 10 năm tới. Tuy nhiên, nhu cầu nhân lực CNTT tăng nhưng thị trường lao động lĩnh vực này tại Việt Nam luôn trong tình trạng thiếu hụt về cả số lượng và chất lượng. Cụ thể, năm 2019, số lượng nhân lực CNTT cần có là 350.000 người, nhưng thiếu khoảng 90.000 người. Năm 2020, số nhân lực ngành CNTT cần có ước tính khoảng 400.000 người và ước tính thiếu hụt 100.000 nhân sự và năm 2021 cần 500.000 người và thiếu hụt 190.000 người.

Đại diện VietnamWorks chia sẻ: “Nhu cầu tuyển dụng nhân sự ngành CNTT đã tăng tới 3,8 lần, riêng ngành công nghệ phần mềm là 4,1 lần. Tuy nhiên, nhân lực ngành CNTT vẫn còn gặp nhiều rào cản trong việc tìm được công việc do thiếu những kỹ năng và công việc thực tế tại doanh nghiệp”.

Trong khi ngành CNTT thiếu về số lượng thì chất lượng cũng chưa đạt yêu cầu của doanh nghiệp. Theo Bộ Giáo dục và Đào tạo, tại Việt Nam hiện có khoảng 50 trường đào tạo ngành CNTT. Hàng năm có khoảng 50.000 sinh viên CNTT ra trường nhưng chỉ có khoảng 30% lao động là có thể đáp ứng yêu cầu, số còn lại cần phải được đào tạo bổ sung, đào tạo lại.

Nắm bắt được những được thông tin đó ngay từ sớm, ngày từ những ngày đầu thành lập (năm 2009) khoa Công nghệ thông tin trường Đại học Đại Nam đã xây dựng chương trình đào tạo với mục tiêu đào tạo kỹ sư CNTT có kiến thức chuyên môn toàn diện, có kiến thức thực tế vững chắc, có khả năng vận dụng kiến thức trong làm việc đáp ứng được yêu cầu ngày càng cao về nguồn nhân lực trong lĩnh vực công nghệ của Việt Nam và thế giới. Để đáp ứng yêu cầu của thực tiễn cũng như nhu cầu đa dạng của người học, từ năm 2009 đến nay, chương trình đào tạo kỹ sư CNTT đã được điều chỉnh nhiều lần.

Đặc biệt năm 2015, chương trình được điều chỉnh để đáp ứng mục tiêu đào tạo theo hướng ứng dụng; năm 2017, dựa trên sự tham khảo các chương trình đào tạo của các trường đại học lớn trong và ngoài nước như Trường Đại học Công nghệ Đại học Quốc gia Hà Nội, Đại học Bách Khoa, Đại học Mở Hà Nội, Đại học Công nghệ Giao thông Vận tải và Swinburne University of Technology, Melbourne, Australia cũng như tham khảo các ý kiến của các chuyên gia, các nhà tuyển dụng,… chương trình có rà soát bổ sung, thay thế một số môn nhằm thích nghi với nhu cầu ngày càng cao trong lĩnh vực CNTT như: Trí tuệ nhân tạo, công nghệ phần mềm,…

Song song với việc liên tục cập nhật chương trình đào đạo, phương pháp giảng dạy, nhà trường còn tạo điều kiện cho sinh viên CNTT môi trường ứng dụng thực tế kiến thức được học vào ứng dụng thực tế thông qua chương trình nghiên cứu khoa học – một trong những hoạt động thường niên của nhà trường. Qua hoạt động này, giúp cho sinh viên trường Đại học Đại Nam nói chung và sinh viên khoa CNTT nói riêng hiểu sâu hơn về kiến thức chuyên môn, có cái nhìn chi tiết hơn về công việc sau khi ra trường, ngoài ra sinh viên còn có thể rèn được khả năng tự nghiên cứu sâu, kỹ năng làm việc nhóm, kỹ năng quản lý dự án, kỹ năng thu thập thông tin, kỹ năng phân tích thiết kế hệ thống,… đây là một sự trải nghiệm vô cùng quý báu và cũng là nơi giúp sinh viên làm quen với quy trình làm việc trong doanh nghiệp.

Công nghệ thông tin trong giai đoạn hiện nay đang phát triển như vũ báo ở mọi lĩnh vực hoạt động khắp nơi trên thế giới. Điều xảy ra trong vòng vài năm qua ở Việt Nam là sự đầu tư ô ạt vào công nghệ. Tin học đã và đang là một trong các vấn đề không thể thiếu đối với bất kỳ công ty nào. Đặc biệt tin học ngày càng giữ vai trò quan trọng trong vấn đề quản lý tại các tổ chức nhất là tại các lĩnh vực thu thập thông tin. Tuy ngành công nghệ thông tin ở nước ta mới chỉ phát triển trong vài năm trở lại đây và đang từng bước phát triển nhưng những bước phát triển đó đã cho thấy tiềm năng lớn lao trong lĩnh vực công nghệ thông tin ở nước ta, trong đó phát triển hệ thống thông tin đang là thế mạnh của đất nước. Hệ thống thông tin giúp công việc quản lý dễ dàng hơn, nâng cao hiệu quả sản xuất, tiết kiệm được lượng lớn thời gian và công sức.

Qua thời gian tìm hiểu thực tế để xác định đề tài thực tập, em nhận thấy hệ thống thông tin quản lý công nhân là một trong những vấn đề rất quan trọng trong việc quản lý một dự án xây dựng đó.

# CHƯƠNG 1. GIỚI THIỆU VỀ CÔNG TY THỰC TẬP

#### 1.1. Tổng quan về Evotek

Evotek là một nhóm liên ngành tạo điều kiện cho quá trình chuyển đổi kỹ thuật số bằng cách cung cấp sản phẩm và dịch vụ phù hợp với từng khách hàng và đối tác. Evotek tập trung vào các giải pháp thông minh và các giải pháp mới nổi công nghệ, cung cấp một loạt các phần mềm chất lượng cao dịch vụ phát triển, phân phối và bảo trì. Hay nói tóm lại Evotek là công ty phát triển phần mềm nguồn chuyên môn và sự xuất sắc của công ty cho toàn cầu, và Evotek có trụ sở tại Hà Nội, Việt Nam.



**Evotek làm gì?**

Evotek giúp khách hàng tự động hóa và tích hợp các quy trình kinh doanh của họ vào một tổng thể mạch lạc. Evotek tạo ra linh hoạt, mạnh mẽ và thông minh các giải pháp được thiết kế đặc biệt cho từng một trong những nhu cầu của khách hàng của bạn. Evotek thiết kế, xây dựng và mở rộng quy mô di động, web và ứng dụng máy tính để bàn, sử dụng tiên tiến công nghệ và cách tiếp cận linh hoạt.

**Evotek làm như thế nào?**

Evotek cam kết cho sự thành công, khách hàng, giúp họ đạt được mục tiêu với hiệu quả, độ tin cậy của sản phẩm, và trong một thời gian nổi bật để đưa ra thị trường.

Tiêu chuẩn quốc tế: Quản lý dự án theo phương pháp quản lý quốc tế và thực hành, sử dụng phương pháp thử nghiệm nhanh và đa cấp

Giao tiếp có ý thức: Hiểu yêu cầu khách hàng của chúng tôi và quy trình kinh doanh của họ, có gần các phiên khái niệm hóa có ý thức, đưa ra và mong đợi phản hồi

Nghiên cứu & Phát triển: Evotek là chuyên gia về những gì Evotek làm, nhưng Evotek hỏi và Evotek nghiên cứu về những gì Evotek không làm. Evotek trong lĩnh vực công nghiệp của khách hàng và cần tạo ra một sản phẩm có ý nghĩa



#### 1.2. Lĩnh vực hoạt động

**Enterprise Application Development**: Evotek tạo ra những trải nghiệm phù hợp với nhu cầu của khách hàng của bạn, tập trung vào sự dễ dàng của sử dụng và thực tiễn.

**Mobile App Development**: Evotek tạo ra những trải nghiệm phù hợp với nhu cầu của khách hàng của bạn, tập trung vào sự dễ dàng của sử dụng và thực hiện.

**Custom Software Development**: Các giải pháp riêng biệt để nâng cao hiệu quả, năng suất, hiểu biết sâu sắc và lợi nhuận. Hệ thống tích hợp, phân tích dữ liệu và nâng cấp.

**Ui/Ux Design & Consulting**: Evotek cố gắng mang đến cho khách hàng những gì tốt nhất thiết kế và trải nghiệm người dùng họ có thể tưởng tượng. Chúng tôi cung cấp phù hợp nhất và thiết kế nổi bật tăng cường và tối đa hóa hành trình của người dùng.

**Search Engine Optimization**: Để thành công trong 4IR, bạn cần biến kỹ thuật số, Evotek hướng dẫn bạn về SEO các quy trình để đảm bảo doanh nghiệp của bạn trước các đối thủ cạnh tranh của bạn. Sự hiện diện trực tuyến sẽ được chú ý của những khách hàng chất lượng đang tìm kiếm những gì bạn và công ty của bạn phải cung cấp.

**Branding Strategy**: Thương hiệu của bạn cho phép bạn giao tiếp với đối tượng mục tiêu của bạn với một độc đáo nhân cách trên tất cả các kênh. Chúng tôi xây dựng thương hiệu có thể phát triển cùng bạn, làm nền tảng cho các giá trị cốt lõi và bản chất của sản phẩm hoặc dịch vụ của bạn.

**Rapid Prototyping Package**: Trước tiên, Evotek lập bản đồ các mục tiêu kinh doanh của bạn. Sau đó Evotek phát triển một nguyên mẫu trong vài ngày chứ không phải vài tuần bằng cách sắp xếp hợp lý các vòng phản hồi giữa kinh doanh và công nghệ thông qua nhiều các bước tạo mẫu. Từ đầu đến cuối được quản lý bởi đội ngũ thiết kế, quản lý sản phẩm của chúng tôi, kỹ sư và chuyên gia thương mại hóa

**Community & Education**: Evotek hỗ trợ và giáo dục cộng đồng xung quanh các công nghệ mới nổi. chúng tôi làm các sự kiện hàng tháng về FinTech và blockchain.



# CHƯƠNG 2: KIẾN THỨC CƠ BẢN

## 2.1. Giới thiệu về C# và ASP.NET CORE

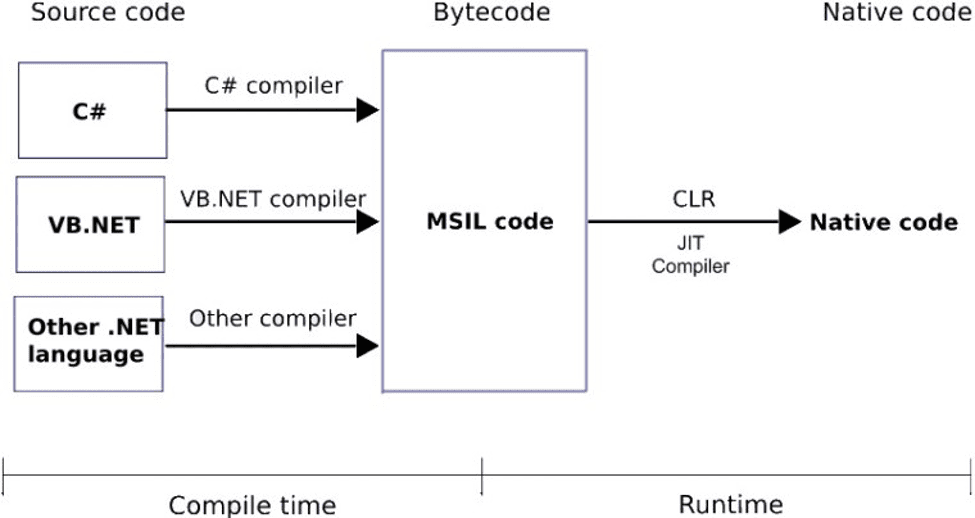
##### Giới thiệu về ngôn ngữ lập trình C#

###### **C# là gì?**

C# (hay C sharp) là một ngôn ngữ lập trình đơn giản, được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000. C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

Trong các ứng dụng Windows truyền thống, mã nguồn chương trình được biên dịch trực tiếp thành mã thực thi của hệ điều hành. Trong các ứng dụng sử dụng .NET Framework, mã nguồn chương trình (C#, VB.NET) được biên dịch thành mã ngôn ngữ trung gian MSIL (Microsoft intermediate language).

Sau đó mã này được biên dịch bởi Common Language Runtime (CLR) để trở thành mã thực thi của hệ điều hành *(Hình 2.1)*



Hình 1‑1: Quá trình chuyển đổi MSIL code thành Native code

C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng Windows Forms hay WPF (Windows Presentation Foundation), phát triển game, ứng dụng Web, ứng dụng Mobile trở nên rất dễ dàng.

###### **Đặc trưng của ngôn ngữ lập trình C#**

* C# là ngôn ngữ đơn giản:

C# loại bỏ một vài sự phức tạp và rối rắm của những ngôn ngữ như Java và c++, bao gồm việc loại bỏ những macro, những template, đa kế thừa, và lớp cơ sở ảo (virtual base class).

Ngôn ngữ C# đơn giản vì nó dựa trên nền tảng C và C++. Nếu chúng ta thân thiện với C và C++ hoặc thậm chí là Java, chúng ta sẽ thấy C# khá giống về diện mạo, cú pháp, biểu thức, toán tử và những chức năng khác được lấy trực tiếp từ ngôn ngữ C và C++, nhưng nó đã được cải tiến để làm cho ngôn ngữ đơn giản hơn.

* C# là ngôn ngữ hiện đại:

C# phù hợp cho việc phát triển trong thời đại 4.0, bao gồm việc phát triển web, mobile app, game, học máy và trí tuệ nhân tạo, phát triển đám mây, IoT, blockchain, microservices...

* C# là một ngôn ngữ lập trình thuần hướng đối tượng:

Lập trình hướng đối tượng (OOP: Object-oriented programming) là một phương pháp lập trình có 4 tính chất. Đó là tính trừu tượng (abstraction), tính đóng gói (encapsulation), tính đa hình (polymorphism) và tính kế thừa (inheritance). C# hỗ trợ cho chúng ta tất cả những đặc tính trên. Ngoài ra C# còn hỗ trợ các phương pháp lập trình chức năng thông qua các biểu thức lamba, khớp mẫu, functions, các thuộc tính bất biến.

* C# là một ngôn ngữ ít từ khóa:

C# là ngôn ngữ sử dụng giới hạn những từ khóa. Phần lớn các từ khóa được sử dụng để mô tả thông tin. Chúng ta có thể nghĩ rằng một ngôn ngữ có nhiều từ khóa thì sẽ mạnh hơn. Điều này không phải sự thật, ít nhất là trong trường hợp ngôn ngữ C#, chúng ta có thể tìm thấy rằng ngôn ngữ này có thể được sử dụng để làm bất cứ nhiệm vụ nào.

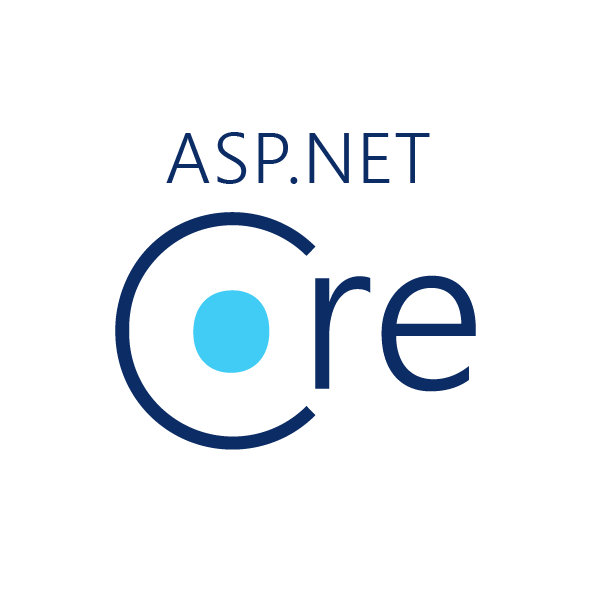
*\* Ngoài ra C# còn có những ưu điểm:*

* [C# là ngôn ngữ lập trình mã nguồn mở](https://github.com/dotnet/csharplang),vì vậy C# là miễn phí với tất cả mọi người, đồng thời mọi người đều có thể cùng tham gia phát triển, đề xuất thiết kế ngôn ngữ C#
* C# là ngôn ngữ đa nền tảng vì vậy có thể biên dịch trên nhiều nền tảng máy tính khác nhau (Windows, Linux, MacOS)
* C# có hiệu suất cao và tốc độ thực thi nhanh do sử dụng trình biên dich trung gian ([CLR](https://docs.microsoft.com/en-us/dotnet/standard/clr)), điểm cộng nữa là tốc độ phát triển phần mềm nhanh chóng so với đa số các ngôn ngữ hiện tại.
* C# có IDE Visual Studio cùng nhiều plug-in vô cùng mạnh mẽ. ngoài ra có thể viết C# bằng bất kỳ text editor nào khác như Visual Studio Code, Vim, Netbeam...
* C# có cấu trúc khá gần gũi với các ngôn ngữ lập trình truyền thống, song cũng được bổ sung các yếu tố mang tính hiện đại nên dễ dàng tiếp cận cho người mới học và học nhanh với C#.
* C# có cộng đồng nhà phát triển vô cùng lớn mạnh.
* C# được phát triển và cải tiến không ngừng với tần suất 1 phiên bản/ 1 năm, đáp ứng các mong muốn cải thiện, cải tiến cho phù hợp với nhu cầu công nghệ của các nhà phát triển.
* C# có [tài liệu tham khảo](https://docs.microsoft.com/en-us/dotnet/csharp/) và hướng dẫn vô cùng phong phú và chất lượng, đồng thời có các buổi hội thảo giới thiệu tính năng mới và định hướng phát triển ngôn ngữ trong tương lai.
* C# và.NET được đánh giá là có design tốt, vì vậy cú pháp và logic rất nhất quán, mã nguồn C# dễ đọc và mở rộng.
* C# được thiết kế và phát triển bởi Microsoft nên rất được Microsoft quan tâm và hỗ trợ.

##### Giới thiệu về ASP.NET CORE

###### **2.1.2.1. Tổng quan về ASP.NET CORE**

* ASP.NET Core là phiên bản mới của khung công tác web ASP.NET chủ yếu được nhắm mục tiêu để chạy trên nền tảng .NET Core.
* ASP.NET Core là một khuôn khổ miễn phí, mã nguồn mở và đa nền tảng để xây dựng các ứng dụng dựa trên đám mây, chẳng hạn như ứng dụng web, ứng dụng IoT và phụ trợ di động. Nó được thiết kế để chạy trên đám mây cũng như tại chỗ.
* Giống như .NET Core, nó được cấu trúc theo mô-đun với chi phí tối thiểu và sau đó các tính năng nâng cao khác có thể được thêm vào dưới dạng gói NuGet theo yêu cầu của ứng dụng. Điều này dẫn đến hiệu suất cao, yêu cầu ít bộ nhớ hơn, kích thước triển khai ít hơn và dễ bảo trì.
* ASP.NET Core là một framework mã nguồn mở được hỗ trợ bởi Microsoft và cộng đồng, vì vậy lập trình viên cũng có thể đóng góp hoặc tải xuống mã nguồn từ [ASP.NET Core Repository trên Github](https://github.com/dotnet/aspnetcore) .
* ASP.NET 3.x chỉ chạy trên .NET Core 3.x, trong khi ASP.NET Core 2.x chạy trên .NET Core 2.x cũng như .NET Framework.



Hình 1‑2: Logo ASP.NET CORE

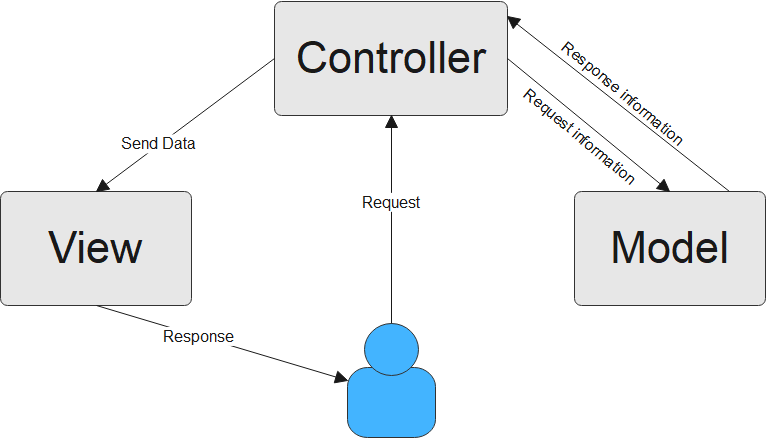
###### **2.1.2.2. Tại sao sử dụng ASP.NET CORE**

* **Hỗ trợ nhiều nền tảng:** Các ứng dụng ASP.NET Core có thể chạy trên Windows, Linux và Mac. Vì vậy không cần phải xây dựng các ứng dụng khác nhau cho các nền tảng khác nhau bằng cách sử dụng các framework khác nhau.
* **Nhanh chóng:** ASP.NET Core không còn phụ thuộc vào System.Web.dll để giao tiếp trình duyệt-máy chủ. ASP.NET Core cho phép bao gồm các gói mà lập trình viên cần cho ứng dụng của mình. Điều này làm giảm luồng yêu cầu và cải thiện hiệu suất và khả năng mở rộng.
* **IoC Container:** ASP.NET Core bao gồm IoC container tích hợp để tiêm phụ thuộc tự động giúp nó có thể bảo trì và kiểm tra được.
* **Tích hợp với các khung giao diện người dùng hiện đại:** Nó cho phép bạn sử dụng và quản lý các khung giao diện người dùng hiện đại như AngularJS, ReactJS, Umber, Bootstrap, v.v. bằng cách sử dụng Bower (một trình quản lý gói cho web).
* **Lưu trữ:** Ứng dụng web ASP.NET Core có thể được lưu trữ trên nhiều nền tảng với bất kỳ máy chủ web nào như IIS, Apache, v.v. Nó không chỉ phụ thuộc vào IIS như một .NET Framework tiêu chuẩn.
* **Chia sẻ mã:** ASP.NET cho phép xây dựng một thư viện lớp có thể được sử dụng với các khuôn khổ .NET khác như .NET Framework 4.x hoặc Mono. Do đó, một cơ sở mã duy nhất có thể được chia sẻ trên các framework.
* **Phiên bản ứng dụng song song:** ASP.NET Core chạy trên .NET Core, hỗ trợ chạy đồng thời nhiều phiên bản ứng dụng.
* **Deployment footprint nhỏ hơn**: Ứng dụng ASP.NET Core chạy trên .NET Core, nhỏ hơn .NET Framework đầy đủ. Vì vậy, ứng dụng chỉ sử dụng một phần .NET CoreFX sẽ có kích thước deployment nhỏ hơn. Điều này làm giảm deployment footprint.

##### Giới thiệu mô hình Model – View – Controller (MVC)

Mô hình MVC (viết tắt chữ cái đầu của 3 từ Model - View - Controller) là một kiến trúc phần mềm hay mô hình thiết kế được sử dụng trong kỹ thuật phần mềm (đặc biệt đối với phát triển ứng dụng web). Mô hình này giúp cho tổ chức ứng dụng (phân bố source code ứng dụng) thành 3 phần khác nhau là: *Model*, *View* và *Controller*. Mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác.

* **Model**: là các thành phần chứa tất cả các nghiệp vụ logic, phương thức xử lý, truy xuất CSDL, đối tượng mô tả dữ liệu như các Class, hàm xử lý... Model được giao nhiệm vụ cung cấp dữ liệu cho CSDL và lưu dữ liệu vào các kho chứa dữ liệu. Tất cả các nghiệp vụ logic được thực thi ở Model. Dữ liệu vào từ người dùng sẽ thông qua View để kiểm tra ở Model trước khi lưu vào cơ sở dữ liệu. Việc truy xuất, xác nhận và lưu dữ liệu là một phần của Model.
* **View**: View hiển thị các thông tin cho người dùng của ứng dụng và được giao nhiệm vụ cho việc nhận các dữ liệu vào từ người dùng, gửi đi các yêu cầu người dùng đến bộ điều khiển (Controller), sau đó là nhận lại các phản hồi từ bộ điều khiển và hiển thị kết quả cho người dùng. Các trang HTML, JSP,… là một phần của View.
* **Controller**: Controller là tầng trung gian giữa Model và View. Controller được giao nhiệm vụ nhận các yêu cầu từ người dùng (phía máy khách). Một yêu cầu được nhận từ máy khách được thực hiện bởi một chức năng logic thích hợp từ thành phần Model và sau đó sinh ra các kết quả cho người dùng và được thành phần View hiển thị. ActionServlet, Action, ActionForm, struts-config.xml là các thành phần của Controller.



Hình 1‑3: Mô hình MVC

Một ứng dụng web được phát triển theo mô hình MVC, có nguyên lý hoạt động xử lý yêu cầu của người dùng như sau:

* Khi có yêu cầu phát sinh từ người dùng (phía Client), yêu cầu này sẽ được gửi đến phía Server, tại đây Controller sẽ tiếp nhận yêu cầu để xử lý.
* Bên trong Controller chứa nhiều phương thức (action), mỗi action tương ứng với một yêu cầu cụ thể từ người dùng. Controller sẽ lựa chọn action phù hợp với yêu cầu để xử lý. Trong quá trình xử lý, Controller tương tác với Model để có được dữ liệu mà người dùng mong muốn.
* Sau khi Controller làm việc với model để có được dữ liệu theo yêu cầu, Controller sẽ gửi về cho View thành phần Data Model, và nhiệm vụ của View sẽ chuyển Data Model nhận được thành dữ liệu ứng dụng và gửi trả về phía Client để hiển thị kết quả yêu cầu.

##### Giới thiệu về jQuery Ajax

###### **2.1.4.1. AJAX là gì?**

AJAX là từ viết tắt của từ Asynchronous JavaScript And XML - Bất đồng bộ trong Javascript và XML.

* Asynchoronous (Bất đồng bộ): Bất đồng bộ có nghĩa là một chương trình có thể xử lý không theo tuần tự các hàm, không có quy trình, có thể nhảy đi bỏ qua bước nào đó. Ích lợi dễ thấy nhất của bất đồng bộ là chương trình có thể xử lý nhiều công việc một lúc.
* Javascript: Đây là một ngôn ngữ lập trình rất phổ biến hiện nay. Trong số rất nhiều chức năng của nó là khả năng quản lý nội dung động của website và hỗ trợ tương tác với người dùng.
* XML: Đây là một dạng ngôn ngữ gần giống với HTML, tên đầy đủ là eXtensible Markup Language. Nếu HTML được dùng để hiển thị dữ liệu, XML được thiết kế để chứa dữ liệu.

Với AJAX, một máy khách (tức là trình duyệt) có thể liên lạc với một máy chủ web và gửi yêu cầu để nhận được dữ liệu. Sau đó, nó sẽ xử lí phản hồi của máy chủ và tạo ra sự thay đổi trên trang web mà không cần phải tải lại hoàn toàn trang web đó.

###### **2.1.4.2. jQuery là gì?**

jQuery là thư viện được viết từ JavaScript, jQuery giúp xây dựng các chức năng bằng Javascript dễ dàng, nhanh và giàu tính năng hơn.

jQuery được tích hợp nhiều module khác nhau, từ module hiệu ứng cho đến module truy vấn selector. jQuery được sử dụng đến 99% trên tổng số website trên thế giới.

Các module phổ biến của jQuery bao gồm:

* Ajax – xử lý Ajax
* Atributes – Xử lý các thuộc tính của đối tượng HTML
* Effect – xử lý hiệu ứng
* Event – xử lý sự kiện
* Form – xử lý sự kiện liên quan tới form
* DOM – xử lý Data Object Model
* Selector – xử lý luồng lách giữa các đối tượng HTML

jQuery không phải là một ngôn ngữ lập trình riêng biệt mà hoạt động liên kết với JavaScript. jQuery cung cấp các API giúp việc duyệt tài liệu HTML, hoạt ảnh, xử lý sự kiện và thao tác AJAX đơn giản hơn. jQuery hoạt động tốt trên nhiều loại trình duyệt khác nhau.

###### **2.1.4.3. jQuery làm được gì cho AJAX?**

Jquery cung cấp một số phương thức để thực hiện các chức năng AJAX.

Với các phương thức jQuery AJAX, chúng ta có thể yêu cầu các text, HTML, XML và JSON từ server sử dụng cả giao thức HTTP GET và HTTP POST, chúng ta cũng có thể lấy dữ liệu từ bên ngoài trực tiếp vào trong phần tử được chọn.

Nếu không có jQuery, việc viết mã AJAX sẽ hơi phức tạp:  Viết mã AJAX thông thường có thể hơi phức tạp, vì các trình duyệt khác nhau có cú pháp khác nhau để triển khai AJAX. Điều này có nghĩa là chúng ta sẽ phải viết thêm mã để kiểm tra các trình duyệt khác nhau. Tuy nhiên, nhóm jQuery đã giải quyết việc này để chúng ta có thể viết chức năng AJAX chỉ với một dòng mã duy nhất.

##### Thư viện đọc, ghi dữ liệu từ bảng tính EPPlus

EPPlus là một thư viện .NET đọc và ghi các tệp Excel 2007/2010/2013 bằng cách sử dụng định dạng Open Office Xml (xlsx).

EPPlus hỗ trợ nhiều thuộc tính của bảng tính như phạm vi ô, kiểu ô, biểu đồ, hình ảnh, hình dạng, nhận xét, bảng, bảo vệ, mã hóa, bảng tổng hợp, xác thực dữ liệu, định dạng có điều kiện, tính toán công thức, v.v.

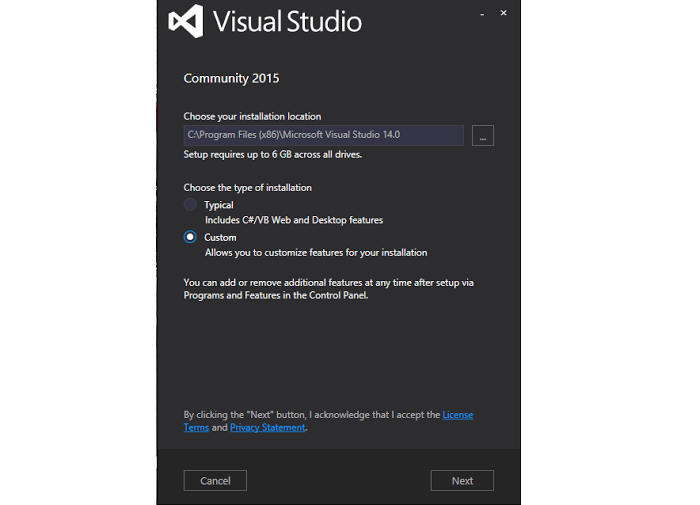
Phiên bản EPPlus đầu tiên được ra đời vào ngày 30/11/2009, được phát triển bởi lập trình viên người Thụy Điển Jan Källman. Vào thời điểm đó, tiêu chuẩn Office Open XML để biểu diễn bảng tính còn tương đối mới và không có nhiều trong không gian .NET. Sau một vài năm, anh được tham gia cùng với anh trai của mình là Mats Alm. Dự án đã tạo cơ hội cho hai anh em, cả hai đều là lập trình viên kể từ khi họ tranh giành Commodore 64 vào những năm 80, để cùng nhau xây dựng một thứ gì đó.

Nhờ sự phong phú về tính năng, hiệu suất và chất lượng, EPPlus đã sớm trở nên phổ biến trong giới phát triển .NET trên toàn thế giới. Tính đến năm 2021, nó là thư viện bảng tính được tải xuống nhiều nhất trên Nuget với hơn 30 triệu lượt tải xuống kể từ khi bắt đầu.

##### Giới thiệu Visual Studio 2019

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, chúng ta có thể viết code, sửa lỗi, chỉnh sửa thiết kế ứng dụng dễ dàng chỉ với 1 phần mềm Visual Studio. 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.



Hình 1‑4: Giao diện Visual Studio Community 2015

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](https://mona.software/). 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 mở, các tổ chức phi doanh nghiệp dưới 5 người dùng).

Tính đến nay, Visual Studio vẫn được coi là phần mềm lập trình hệ thống hàng đầu, chưa có phần mềm nào có thể thay thế được nó. Được đánh giá cao như vậy bởi Visual Studio sở hữu nhiều tính năng cực kỳ hấp dẫn. Cụ thể:

* 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 biên dịch 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 phát triển 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](https://vi.wikipedia.org/wiki/Visual_Basic#:~:text=Visual%20Basic%20(vi%E1%BA%BFt%20t%E1%BA%AFt%20VB,h%E1%BB%A3p%20(IDE)%20k%E1%BA%BFt%20b%C3%B3.), JavaScript,… Bởi vậy, Visual Studio có thể dễ dàng phát hiện và thông báo cho lập trình viên 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](https://mona.solutions/thiet-ke-website).
* 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, chúng ta có thể dễ dàng tải xuống các tiện ích mở rộng. Tính năng này được hoạt động như một phần chương trình độc lập nên không 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 người lập trình lưu trữ an toàn, dễ tìm, dễ sử dụng.
* Kho lưu trữ an toàn: Với Visual Studio, chúng ta 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 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](https://github.com/) 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: Tính năng này cho phép lập trình viên để lại nhận xét, giúp dễ dàng ghi nhớ công việc cần hoàn thành, và ghi nhớ nội dung đoạn code trước đó, giúp lập trình viên cùng các thành viên trong cùng một nhóm có thể hiểu được đoạn mã đó một cách dễ dàng. Hỗ trợ đắc lực trong việc phát triển ứng dụng.

Với các ưu điểm trên của Visual Studio, có thể khẳng định đây chính là phần mềm lý tưởng để xây dựng nên hệ thống hỗ trợ điểm danh sinh viên cho trường Đại học Đại Nam.

#### 2.2. Hệ quản trị cơ sở dữ liệu SQL Server

##### 2.2.1. Giới thiệu về cơ sở dữ liệu

###### **2.2.1.1. Dữ liệu là gì?**

Là các thông tin của đối tượng (người, vật, một khái niệm, sự việc…) được lưu trữ trên máy tính.

Dữ liệu được mô tả dưới nhiều dạng khác nhau (các ký tự, ký số, hình ảnh, ký hiệu, âm thanh…). Mỗi cách mô tả gắn với một ngữ nghĩa nào đó.

Dữ liệu về đối tượng có thể khác nhau, tùy thuộc vào ngữ cảnh. Ví dụ: dữ liệu về đối tượng sinh viên có thể khác nhau tùy vào mục đích quản lý:

Quản lý điểm: Tên, mã sinh viên, điểm môn 1, điểm môn 2, điểm môn 3. Trong khi đó quản lý nhân thân: Tên, địa chỉ, ngày sinh, quê quán.

###### **2.2.1.2. Cơ sở dữ liệu là gì?**

Cơ sở dữ liệu (CSDL) là tập hợp dữ liệu được tổ chức có cấu trúc liên quan với nhau và được lưu trữ trong máy tính.

CSDL được thiết kế, xây dựng cho phép người dùng lưu trữ dữ liệu, truy xuất thông tin hoặc cập nhật dữ liệu.

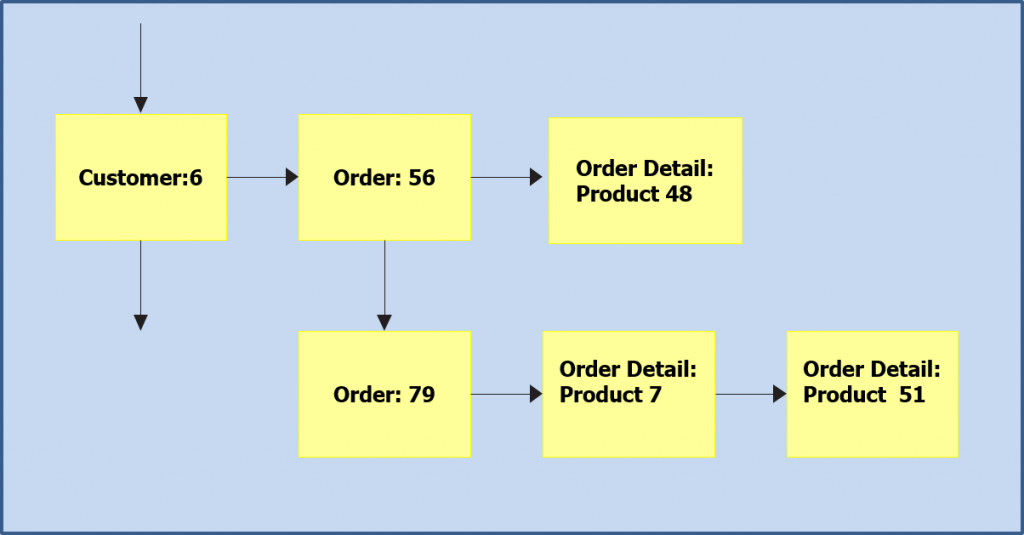
CSDL được tổ chức có cấu trúc: Các dữ liệu được lưu trữ có cấu trúc thành các bản ghi (record), các trường dữ liệu (field). Các dữ liệu lưu trữ có mối quan hệ (relation) với nhau.

CSDL được cấu trúc để dễ dàng truy cập, quản lý và cập nhật.



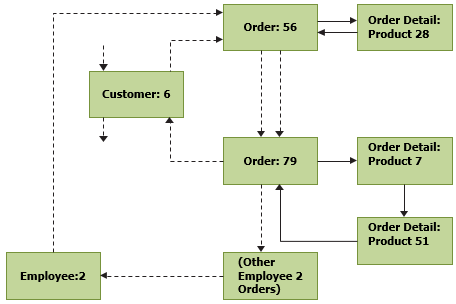
Hình 1‑5: Ví dụ về cơ sở dữ liệu

###### **2.2.1.3. Các mô hình cơ sở dữ liệu**

* Mô hình dữ liệu file: CSDL dạng file phẳng thường là file kiểu văn bản chứa dữ liệu dạng bảng. Ví dụ một file phẳng thể hiện thông tin về Customer (Khách hàng) dưới dạng bảng của công ty Northwind Traders.
* Mô hình dữ liệu phân cấp: Tổ chức theo hình cây, mỗi nút biểu diễn một thực thể dữ liệu. Liên hệ dữ liệu thể hiện trên liên hệ giữa nút cha và nút con. Mỗi nút cha có thể có một hoặc nhiều nút con, nhưng mỗi nút con chỉ có thể có một nút cha.

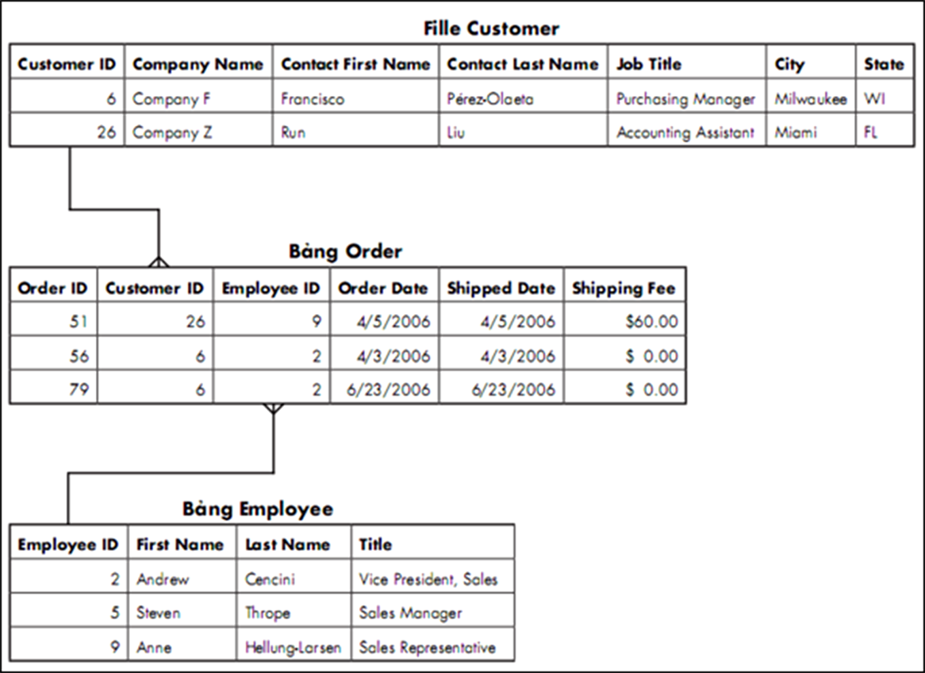
Hình 1‑6: Mô hình dữ liệu phân cấp

* Mô hình dữ liệu mạng: Các file riêng biệt trong hệ thống file phẳng được gọi là các *bản* *ghi* . Tập hợp bản ghi cùng kiểu tạo thành một kiểu thực thể dữ liệu. Các kiểu thực thể kết nối với nhau thông qua mối quan hệ cha-con. Mô hình dữ liệu mạng biểu diễn bởi một đồ thị có hướng, và các mũi tên chỉ từ kiểu thực thể cha sang kiểu thực thể con.



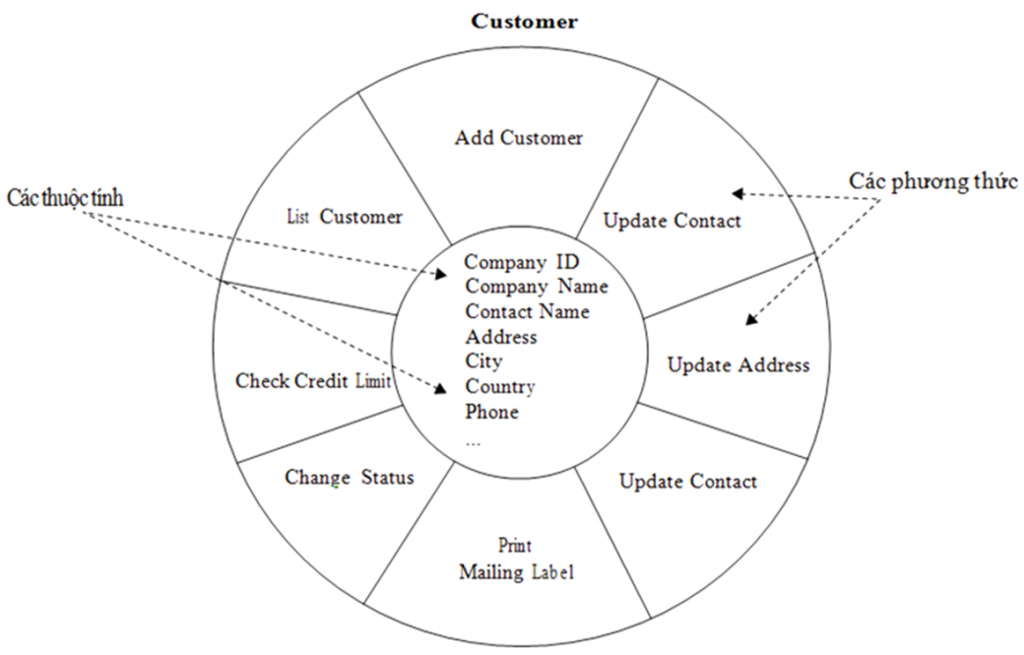
Hình 1‑7: Mô hình dữ liệu mạng

* Mô hình dữ liệu quan hệ: Trong mô hình dữ liệu quan hệ, không có các liên kết vật lý. Dữ liệu được biểu diễn dưới dạng bảng với các hàng và các cột: CSDL là tập hợp các bảng (còn gọi là quan hệ). Mỗi hàng là một bản ghi (record), còn được gọi là bộ (tuple). Mỗi cột là một thuộc tính, còn được gọi là trường (field)



Hình 1‑8: Mô hình dữ liệu quan hệ

* Mô hình dữ liệu hướng đối tượng: Mỗi đối tượng bao gồm các thuộc tính, phương thức (hành vi) của đối tượng. Các đối tượng trao đổi với nhau thông qua các phương thức. Một đối tượng có thể được sinh ra từ việc thừa kế từ đối tượng khác, nạp chồng (hay định nghĩa lại) phương thức của đối tượng khác…



Hình 1‑9: Mô hình dữ liệu hướng đối tượng

##### 2.2.2. Giới thiệu công cụ Microsoft SQL Server Management Studio 18

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System (RDBMS) ) sử dụng câu lệnh SQL (Transact-SQL**)**để trao đổi dữ liệu giữa máy Client và máy cài SQL Server. Một RD BMS bao gồm databases, database engine và các ứng dụng dùng để quản lý dữ liệu và các bộ phận khác nhau trong RDBMS.

SQL được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn (Very Large Database Environment), lên đến Tera-Byte và có thể phục vụ cùng lúc cho hàng ngàn user. SQL Server có thể kết hợp với các server khác như Microsoft Internet Information Server (IIS), E-Commerce Server, Proxy Server,…

Một vài ấn bản SQL Server:

* **Enterprise** : chứa tất cả cá đặc điểm nổi bật của SQL Server, bao gồm nhân bộ máy cơ sở dữ liệu và các dịch vụ đi kèm cùng với các công cụ cho tạo và quản lý phân cụm SQL Server. Nó có thể quản lý các CSDL lớn tới 524 [**petabytes**](https://en.wikipedia.org/wiki/Petabyte) và đánh địa chỉ 12 [**terabytes**](https://en.wikipedia.org/wiki/Terabyte)bộ nhớ và hỗ trợ tới 640 bộ vi xử lý(các core của cpu)
* **Standard** : Rất thích hợp cho các công ty vừa và nhỏ vì giá thành rẻ hơn nhiều so với Enterprise Edition, nhưng lại bị giới hạn một số chức năng cao cấp (advanced features) khác, edition này có thể chạy tốt trên hệ thống lên đến 4 CPU và 2 GB RAM.
* **Developer** : Có đầy đủ các tính năng của Enterprise Edition nhưng được chế tạo đặc biệt như giới hạn số lượng người kết nối vào Server cùng một lúc…. Ðây là phiên bản sử dụng cho phát triển và kiểm tra ứng dụng. Phiên bản này phù hợp cho các cá nhân, tổ chức xây dựng và kiểm tra ứng dụng
* **Workgroup**: ấn bản SQL Server Workgroup bao gồm chức năng lõi cơ sở dữ liệu nhưng không có các dịch vụ đi kèm. Chú ý phiên bản này không còn tồn tại ở SQL Server 2012.
* **Express**: SQL Server Express dễ sử dụng và quản trị cơ sở dữ liệu đơn giản. Được tích hợp với Microsoft Visual Studio, nên dễ dàng để phát triển các ứng dụng dữ liệu, an toàn trong lưu trữ, và nhanh chóng triển khai. SQL Server Express là phiên bản miễn phí,  không giới hạn về số cơ ở dữ liệu hoặc người sử dụng, nhưng nó chỉ dùng cho 1 bộ vi xử lý với 1 GB bộ nhớ và 10 GB file cơ sở dữ liệu. SQL Server Express là lựa chọn tốt cho những người dùng chỉ cần một phiên bản SQL Server 2005 nhỏ gọn, dùng trên máy chủ có cấu hình thấp, những nhà phát triển ứng dụng không chuyên hay những người yêu thích xây dựng các ứng dụng nhỏ.

###### **2.2.2.1. Lịch sử ra đời SQL Server và các phiên bản**

Phiên bản đầu tiên của Microsoft SQL Server ra đời đầu tiên vào năm 1989 cho các hệ điều hành chạy 16 bít với SQL Server phiên bản 1.0 và tiếp tục phát triển cho tới ngày nay.

SQL Server của Microsoft được thị trường chấp nhận rộng rãi kể từ version 6.5. Sau đó Microsoft đã cải tiến và hầu như viết lại một engine mới cho SQL Server 7.0. Cho nên có thể nói từ version 6.5 lên version 7.0 là một  bước nhảy vọt. Có một số đặc tính của SQL Server 7.0 không tương thích với version 6.5. Trong khi đó từ Version 7.0 lên version 8.0 (SQL Server 2000) thì những cải tiến chủ yếu là mở rộng các tính năng về web và làm cho SQL Server 2000 đáng tin cậy hơn.

Một điểm đặc biệt đáng lưu ý ở phiên bản 2000 là **Multiple-Instance**. Tức là bạn có thể cài dặt phiên bản 2000 chung với các phiên bản trước mà không cần phải gỡ chúng. Nghĩa là bạn có thể chạy song song version 6.5 hoặc 7.0 với phiên bản 2000 trên cùng một máy (điều này không thể xảy ra với các phiên bản trước đây). Khi đó phiên bản cũ trên máy bạn là **Default Instance** còn phiên bản 2000 mới vừa cài sẽ là **Named Instance**.

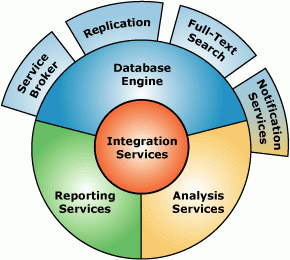
Từ tháng 10 năm 2016, các phiên bản sau được Microsoft hỗ trợ:

* SQL Server 2008 R2
* SQL Server 2012
* SQL Server 2014
* SQL Server 2016

Phiên bản hiện tại là Microsoft SQL Server 2016, xuất bản vào ngày 1/6/2016. SQL Server 2016 chỉ hỗ trợ cho các bộ vi xử lý 64 bit.

###### **2.2.2.2. Các thành phần cơ bản trong SQL Server**

SQL Server được cấu tạo bởi nhiều thành phần như Database Engine, Reporting Services, Notification Services, Integration Services, Full Text Search Service…. Các thành phần này khi phối hợp với nhau tạo thành một giải pháp hoàn chỉnh giúp cho việc lưu trữ và phân tích dữ liệu một cách dễ dàng.



Hình 1‑10: Các thành phần trong SQL Server

###### **2.2.2.3. Database Engine**

Cái lõi của SQL Server: Ðây là một engine có khả năng chứa data ở các quy mô khác nhau dưới dạng table và support tất cả các kiểu kết nối (data connection) thông dụng của Microsoft như ActiveX Data Objects (ADO), OLE DB, and Open Database Connectivity (ODBC).

Ngoài ra nó còn có khả năng tự điều chỉnh (Tune up) ví dụ như sử dụng thêm các tài nguyên (resource) của máy khi cần và trả lại tài nguyên cho hệ điều hành khi một User log off.

###### **2.2.2.4. Replication**

Cơ chế tạo bản sao (Replica): Giả sử ta có một database dùng để chứa dữ liệu được các ứng dụng thường xuyên cập nhật. Nếu ta muốn có một cái database giống hệt như thế trên một server khác để chạy báo cáo (report database, mà report server cũng cần phải được cập nhật thường xuyên để đảm bảo tính chính xác của các báo cáo. Nhưng chúng ta không thể dùng cơ chế back up and restore trong trường hợp này. Lúc đó cơ chế replication của SQL Server sẽ được sử dụng để bảo đảm cho dữ liệu ở 2 database được đồng bộ (synchronized).

###### **2.2.2.5. Integration Services (DTS)**

Integration Services là một tập hợp các công cụ đồ họa và các đối tượng lập trình cho việc di chuyển, sao chép và chuyển đổi dữ liệu.

Nếu làm việc trong một công ty lớn trong đó data được chứa trong nhiều nơi khác nhau và ở các dạng khác nhau cụ thể như chứa trong Oracle, DB2 (của IBM), SQL Server, Microsoft Access….Chắc chắn sẽ cần di chuyển data giữa các server này (migrate hay transfer) và còn muốn định dạng (format) nó trước khi lưu vào database khác, khi đó DTS giúp giải quyết công việc trên dễ dàng.

###### **2.2.2.6. Analysis Services**

Một dịch vụ phân tích dữ liệu của Microsoft: Microsoft cung cấp công cụ cho việc phân tích dữ liệu bằng cách dùng khái niệm hình khối nhiều chiều (multi-dimension cube) và kỹ thuật “khai phá dữ liệu” (data mining).

###### **2.2.2.7. Notification Services**

Dịch vụ thông báo Notification Services là nền tảng cho sự phát triển và triển khai các ứng dụng tạo và gửi thông báo. Notification Services có thể gửi thông báo theo địch thời đến hàng ngàn người đăng ký sử dụng nhiều loại thiết bị khác nhau.

###### **2.2.2.8. Reporting  Services**

Reporting Services bao gồm các thành phần server và client cho việc tạo, quản lý và triển khai các báo cáo. Reporting Services cũng là nền tảng cho việc phát triển và xây dựng các ứng dụng báo cáo.

###### **2.2.2.9. Full Text Search Service**

Dịch vụ SQL Server Full Text Search là một dịch vụ đặc biệt cho đánh chỉ mục và truy vấn cho dữ liệu văn bản không cấu trúc được lưu trữ trong các CSDL SQL Server. Đánh chỉ mục với Full Text Search có thể dduwowcj tạo trên bất kỳ cột dựa trên dữ liệu văn bản. Rất hiệu quả cho việc tìm các sử dụng toán tử LIKE trong SQL với trường hợp tìm văn bản.

###### **2.2.2.10. Service Broker**

Được sử dụng bên trong mỗi Instance, là môi trường lập trình cho việc các ứng dụng nhảy qua các Instance. Service Broker giao tiếp qua giao thức TCP/IP và cho phép các component khác nhau có thể được đồng bộ cùng nhau theo hướng trao đổi các message. Service Broker chạy như một phần của bộ máy cơ sở dữ liệu, cung cấp một nền tảng truyền message tin cậy và theo hàng đợi cho các ứng dụng SQL Server.

##### 2.2.3. Mục đích sử dụng cơ sở dữ liệu

Quản lý dữ liệu bằng CSDL giúp dữ liệu được lưu trữ một cách hiệu quả và có tổ chức, cho phép quản lý dữ liệu nhanh chóng và hiệu quả:

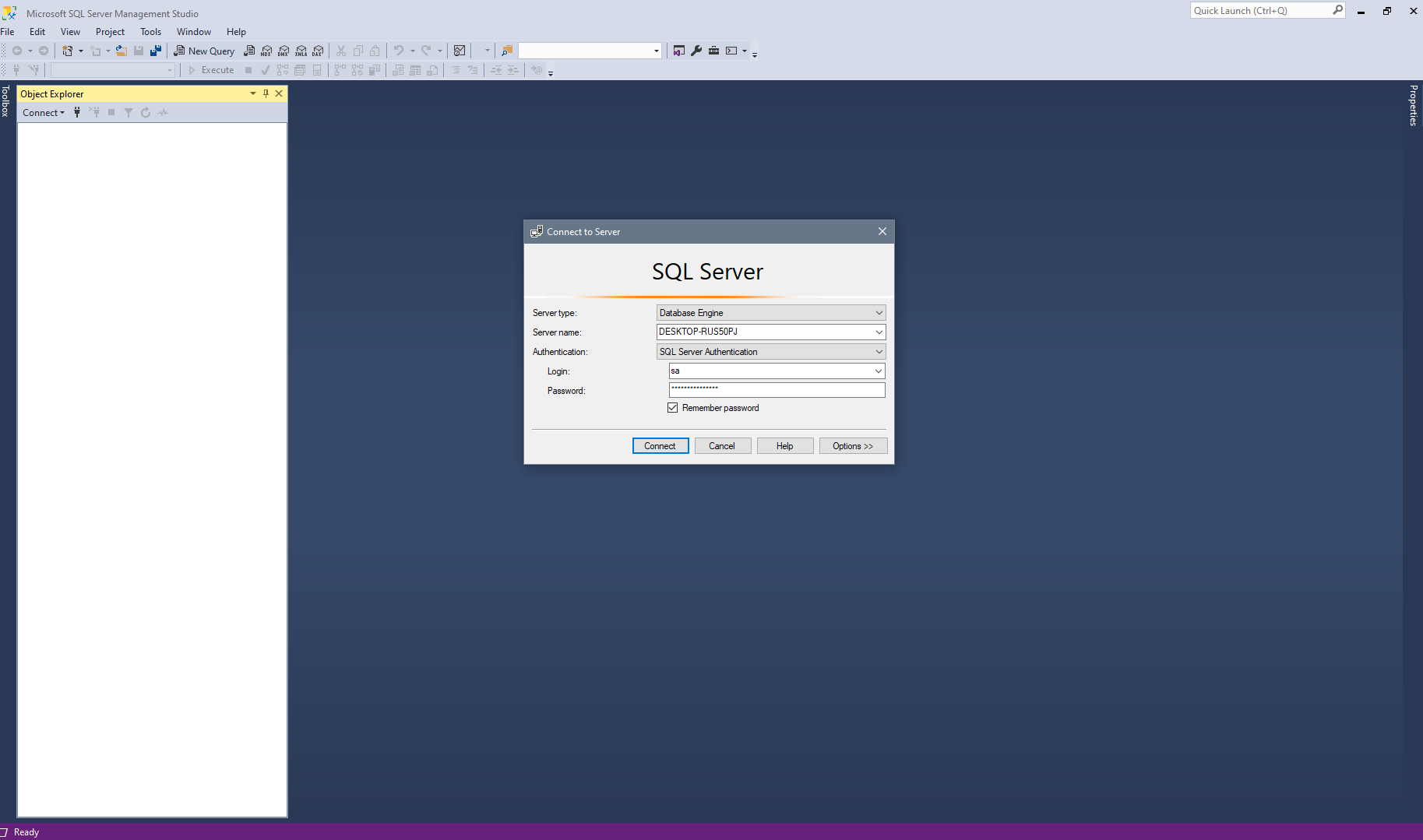
* Lưu trữ thông tin có hệ thống: Cấu trúc sắp xếp có tính hệ thống – đây là điều làm nên sự khác biệt lớn nhất giữa dữ liệu thông thường và cơ sở dữ liệu database. Dữ liệu sẽ được lưu trữ theo một cấu trúc nhất định, có tính nhất quán cao. Với đặc điểm này, database giúp người dùng thuận tiện trong việc tạo lập, lưu trữ, tìm kiếm và sử dụng dữ liệu một cách chính xác và nhanh chóng.
* Đảm bảo an toàn dữ liệu: Đảm bảo toàn vẹn và an toàn dữ liệu là điểm quan trọng hàng đầu trong công tác lưu trữ dữ liệu. Và database xứng đáng nhận “điểm 10” về an toàn dữ liệu.
* Đảm bảo khả năng truy xuất đồng thời của nhiều người dùng trên dữ liệu: Nhiều người dùng có thể sử dụng database cùng lúc mà không phải qua các khâu rườm rà phức tạp nhờ vào việc truy xuất từ các cách khác nhau. Do đó, sẽ thuận lợi hơn trong việc sử dụng, quản lý, truy cập dữ liệu,…
* Linh hoạt thay đổi theo nhu cầu người dùng: Có thể linh hoạt thay đổi kích cỡ và độ phức tạp của một database. Có những database chỉ vài trăm bản ghi (danh sách sinh viên của một lớp) và có những database có dung lượng rất lớn (như database quản lý hàng hóa của một hệ thống siêu thị). Song song đó, hình thức lưu trữ database cũng rất đa dạng. Database có thể được lưu trữ dưới nhiều dạng khác nhau như ổ cứng, USB, cloud,…
* Công tác quản lý dễ dàng hơn: Một database được thiết kế, hình thành, lưu trữ để dễ dàng trong việc tạo lập, cập nhập và khai thác thông tin. Dữ liệu sẽ được cập nhật thường xuyên và hoàn toàn không trùng lặp. Sử dụng database giúp tạo ra các sản phẩm chuyên nghiệp hơn, lưu trữ có hệ thống, dễ dàng trong công tác quản lý.

Với ưu điểm đó, database ngày càng phố biển trong lĩnh vực lập trình ứng dụng nói riêng và công nghệ thông tin nói chung.

#### Kiểm tra kết nối SQL Server và Visual Studio 2019

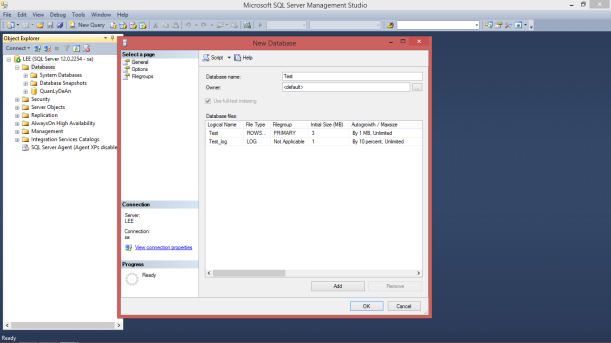
Sau khi cài đặt thành công SQL Server và Visual Studio 2019, tiến hành thực hiện kiểm tra kết nối Database trong SQL Server với Visual Studio 2019.

Mở SQL Server và tiến hành đăng nhập:



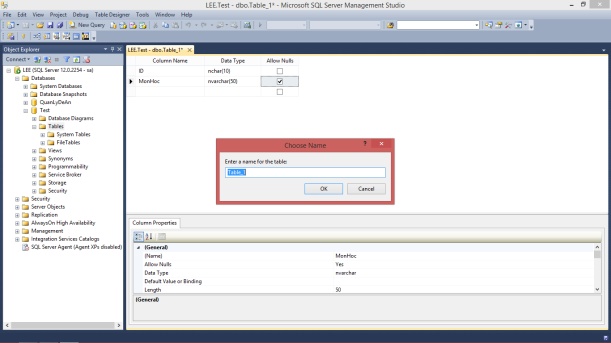
Hình 1‑11: Đăng nhập SQL Server

Tạo một Database (Nếu chưa có database nào):

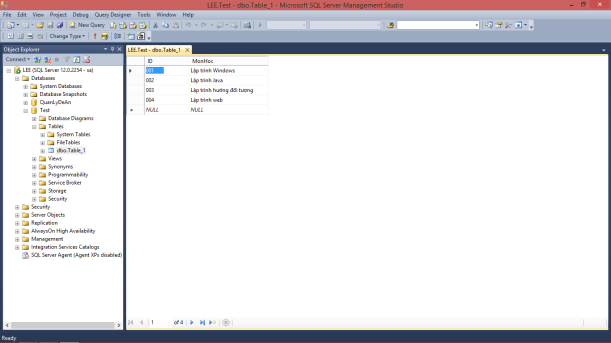


Hình 1‑12: Tạo database

Tạo một table bất kỳ với dữ liệu kèm theo:

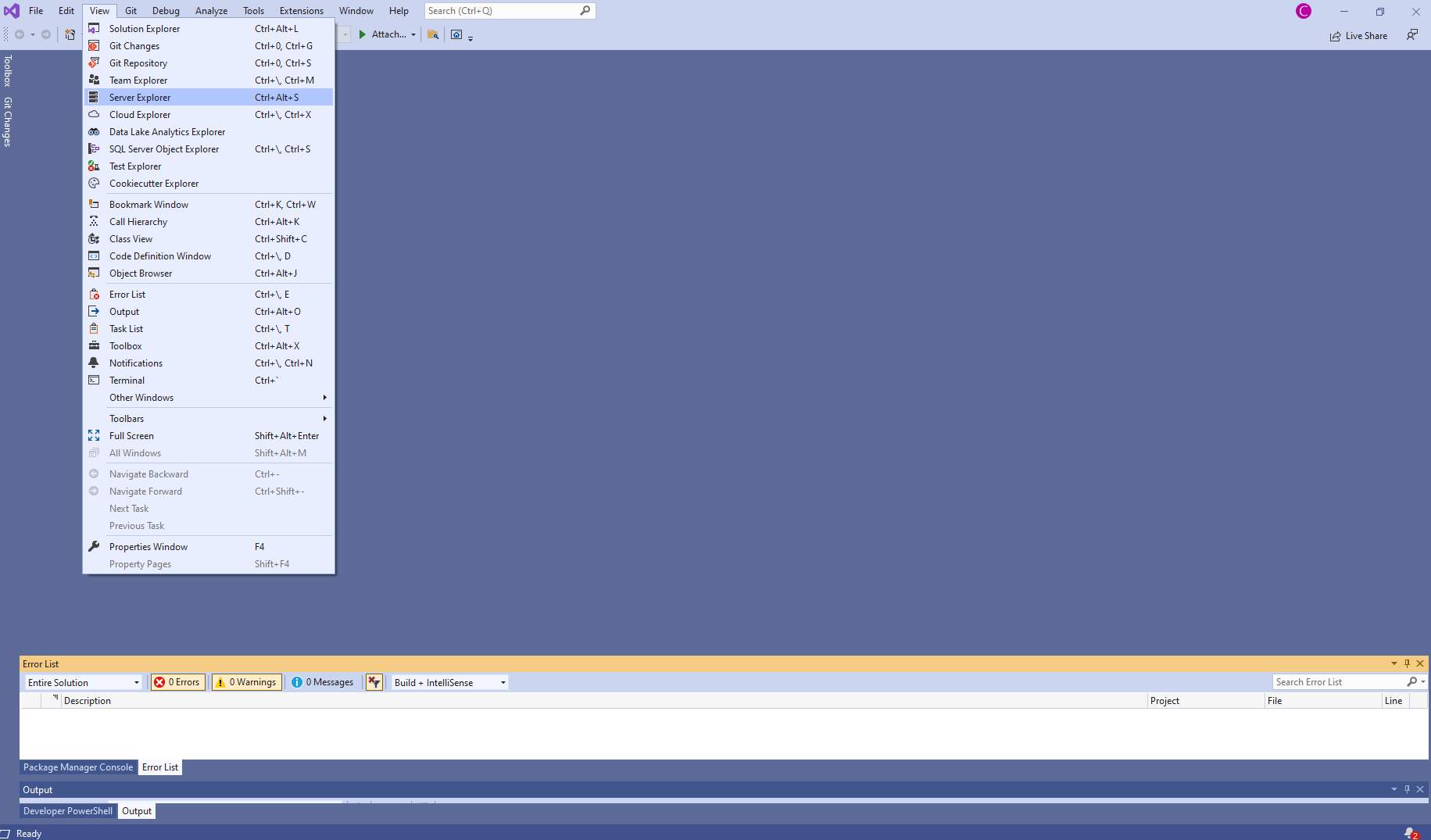


Hình 1‑13: Tạo table trong database



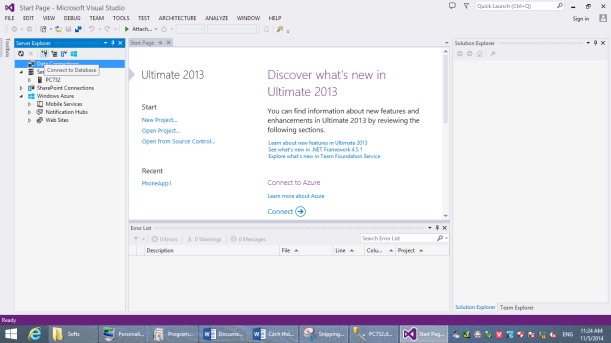
Hình 1‑14: Thêm dữ liệu cho table

Hoặc có thể sử dụng database có sẵn. Sau đó mở Visual Studio, chọn View mở cửa sổ Server Explorer.



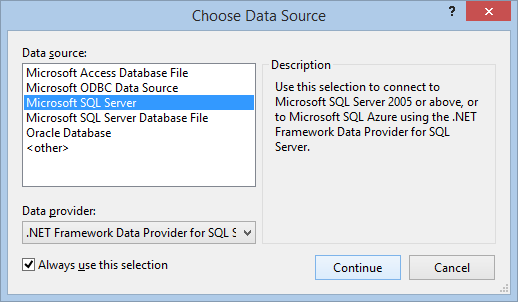
Hình 1‑15: Kết nối SQL Server với Visual Studio - 01

Click vào biểu tượng Connect



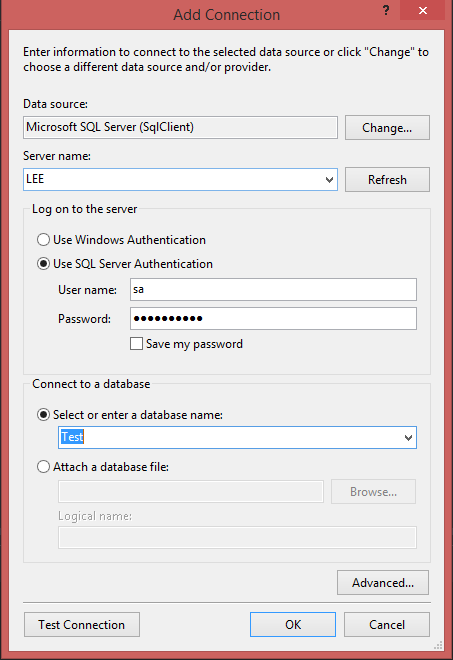
Hình 1‑16: Kết nối SQL Server với Visual Studio - 02

Chọn Microsoft SQL Server, nhấn Continue



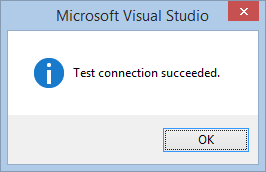
Hình 1‑17: Kết nối SQL Server với Visual Studio - 03

Nhập thông tin Account, chọn đúng Server và Database muốn kiểm tra



Hình 1‑18: Kết nối SQL Server với Visual Studio - 04

Click Test Connection, nếu hiện lên cửa sổ sau là kết nối thành công.



Hình 1‑19: Kết nối SQL Server với Visual Studio - 05

CHƯƠNG 3: NỘI DUNG THỰC HIỆN

3.1. Phân tích yêu cầu bài toán và thiết kế database

* + 1. **Các chức năng cơ bản của hệ thống**

Qua quá trính nghiên cứu nhóm đã xây dựng được một hệ thống CSDL quản lý người lao động cơ bản đáp ứng được một số yêu cầu của như sau:

* Quản lý nhân viên đầy đủ các thông tin cần thiết như: mã, họ, tên, ngày sinh, giới tính, địa chỉ thường trú và tạm trú
* Theo dõi, cập nhật thường xuyên thông tin nhân viên, phòng bạn, chức vụ,…
* Tra cứu tím kiếm thông tin nhân viên, chức vụ, phòng bạn,…
* Thêm, sửa, xóa thông tin nhân viên.
  + 1. **Mô hình phân cấp chức năng**

Diagram

Description automatically generated

* + 1. **Phát hiện thực thể**
* TAI\_KHOAN(MaTK, TenHienThi, TenDangNhap, MatKhau)
* PHAN\_CONG(MaNV, MaCT, ThoiGian)
* CONG\_TRINH(MaCT, TenCT, DiaDiem, NgayCP, NgayKC, NgayHTDK)
* NHAN\_VIEN(MaNV, MaPB, MaCV, HoTen, Ngaysinh, GioiTinh, DiaChiThuongTru, DiaChiTamTru)
* PHONG\_BAN(MaPB, TenPB)
* CHUC\_VU(MaCV, TenCV)
  + 1. **Mô hình ERD**

Diagram

Description automatically generated

* + 1. **Mô hình quan hệ**

**Diagram

Description automatically generated**

#### 3.2. Ý tưởng thiết kế

Dựa trên những phân tích yêu cầu bài toán ở trên, nhóm đưa ra ý tưởng thiết kế các chức năng như sau:

* Đầu tiên là phần đăng nhập, nếu người dùng nhập sai tài khoản hoặc mật khẩu thì thông báo lỗi và để người dùng nhập lại mật khẩu. Ngược lại, chuyển người dùng đến trang quản trị.

Diagram

Description automatically generated

* Sau khi được chuyển đến trang quản trị, tại đây người dùng có thể thực hiện quản lý thông tin nhân viên, chức vụ, phòng ban, phân công, công trình, tài khoản mật khẩu.

Diagram

Description automatically generated

* Khi người dùng chọn vào chức năng quản lý nhân viên thì sẽ hiển thị danh sách nhân viên và các chức năng thêm, sửa, xóa tương ứng.
* Khi muốn thêm mới thì người dùng phải nhập mã nhân viên không được trùng với mã đã có và tên nhân viên không được để trống, nếu không thì không được phép thêm.
* Khi muốn sửa thì người dùng phải chọn vào đối tượng cần sửa, khi đó sẽ hiển thị thông tin cần sửa, sau khi sửa người dùng có thể lưu thay đổi hoặc hủy bỏ thao tác. Ngược lại nếu không chọn đối tượng cần sửa thì không có thông tin để sửa.
* Khi muốn xóa thì người dùng phải chọn đối tượng cần xóa, khi đó sẽ hiển thị thông báo xác nhận xóa, người dùng có thể chọn xóa hoặc hủy thao tác. Ngược lại, nếu không chọn đối tượng thì không có thông tin để xóa.

A picture containing text, sky

Description automatically generated

* Khi người dùng chọn vào chức năng quản lý phòng ban thì sẽ hiển thị danh sách phòng ban và các chức năng thêm, sửa, xóa tương ứng.
* Khi muốn thêm mới thì người dùng phải nhập mã phòng ban không được trùng với mã đã có và tên phòng ban không được để trống, nếu không thì không được phép thêm.
* Khi muốn sửa thì người dùng phải chọn vào đối tượng cần sửa, khi đó sẽ hiển thị thông tin cần sửa, sau khi sửa người dùng có thể lưu thay đổi hoặc hủy bỏ thao tác. Ngược lại nếu không chọn đối tượng cần sửa thì không có thông tin để sửa.
* Khi muốn xóa thì người dùng phải chọn đối tượng cần xóa, khi đó sẽ hiển thị thông báo xác nhận xóa, người dùng có thể chọn xóa hoặc hủy thao tác. Ngược lại, nếu không chọn đối tượng thì không có thông tin để xóa.

A picture containing text, sky, several

Description automatically generated

* Khi người dùng chọn vào chức năng quản lý công trình thì sẽ hiển thị danh sách công trình và các chức năng thêm, sửa, xóa tương ứng.
* Khi muốn thêm mới thì người dùng phải nhập mã công trình không được trùng với mã đã có và tên công trình không được để trống, nếu không thì không được phép thêm.
* Khi muốn sửa thì người dùng phải chọn vào đối tượng cần sửa, khi đó sẽ hiển thị thông tin cần sửa, sau khi sửa người dùng có thể lưu thay đổi hoặc hủy bỏ thao tác. Ngược lại nếu không chọn đối tượng cần sửa thì không có thông tin để sửa.
* Khi muốn xóa thì người dùng phải chọn đối tượng cần xóa, khi đó sẽ hiển thị thông báo xác nhận xóa, người dùng có thể chọn xóa hoặc hủy thao tác. Ngược lại, nếu không chọn đối tượng thì không có thông tin để xóa.

Diagram, schematic

Description automatically generated

* Khi người dùng chọn vào chức năng phân công thì sẽ hiển thị danh sách phân công và các chức năng thêm, sửa, xóa tương ứng.
* Khi muốn thêm mới thì người dùng phải nhập mã nhân viên và mã công trình không được trùng với mã đã có và thời gian không được để trống, nếu không thì không được phép thêm.
* Khi muốn sửa thì người dùng phải chọn vào đối tượng cần sửa, khi đó sẽ hiển thị thông tin cần sửa, sau khi sửa người dùng có thể lưu thay đổi hoặc hủy bỏ thao tác. Ngược lại nếu không chọn đối tượng cần sửa thì không có thông tin để sửa.
* Khi muốn xóa thì người dùng phải chọn đối tượng cần xóa, khi đó sẽ hiển thị thông báo xác nhận xóa, người dùng có thể chọn xóa hoặc hủy thao tác. Ngược lại, nếu không chọn đối tượng thì không có thông tin để xóa.

Diagram, schematic

Description automatically generated

* Khi người dùng chọn vào chức năng quản lý tài khoản thì sẽ hiển thị danh sách phân công và các chức năng thêm, sửa, xóa tương ứng.
* Khi muốn thêm mới thì người dùng phải nhập mã tài khoản và mã tài khoản không được trùng với mã đã có và tên hiển thị, tên đăng nhập, mật khẩu không được để trống, nếu không thì không được phép thêm.
* Khi muốn sửa thì người dùng phải chọn vào đối tượng cần sửa, khi đó sẽ hiển thị thông tin cần sửa, sau khi sửa người dùng có thể lưu thay đổi hoặc hủy bỏ thao tác. Ngược lại nếu không chọn đối tượng cần sửa thì không có thông tin để sửa.
* Khi muốn xóa thì người dùng phải chọn đối tượng cần xóa, khi đó sẽ hiển thị thông báo xác nhận xóa, người dùng có thể chọn xóa hoặc hủy thao tác. Ngược lại, nếu không chọn đối tượng thì không có thông tin để xóa.

Diagram, schematic

Description automatically generated

* 1. **Phần mềm quản lý công nhân**
     1. **Giao diện**
     2. **Khu vực menu**

**Graphical user interface, application

Description automatically generated**

Đây là phần menu của ứng dụng, có các chức năng giúp người dùng đến các trang quản lý công trình, nhân viên, phòng ban, chức vụ, phân công và tài khoản đăng nhập. Ngoài ra, phần trên cùng là tên của phần mềm.

##### 3.2.3. Khu vực điều hướng

****

Đây là khu vực mà người dùng có thể thực hiện các thao tác ẩn hiện phần menu, tìm kiếm và có một vài nút trợ giúp để cho người xem thông báo, thông tin tài khoản. Sơ đồ như sau:

Diagram

Description automatically generated

## Giao diện trang chủ

Graphical user interface, application

Description automatically generated

Đây là giao diện khi người dùng đăng nhập thành công, người dùng sẽ được đưa đến giao diện như hình trên.

##### 3.2.4. Giao diện quản lý công trình

Graphical user interface, application

Description automatically generated

1. **Chức năng tìm kiếm**

Graphical user interface, application, Word

Description automatically generated

Chức năng tìm kiếm theo tên của công trình, khi người dùng nhập tên của công trình thì sẽ hiện thị tên công trình tương ứng với nội dung tìm kiếm.

**b. Chức năng thêm mới công trình**

Graphical user interface, application

Description automatically generated

Nếu muốn thêm mới, người dùng nhấn vào nút thêm mới có biểu tượng dấu “+”. Khi đó, một popupbox sẽ hiển thị lên để người dùng nhập thông tin thêm mới như trên hình. Để xác nhận thêm mới thì chọn vào nút thêm, ngược lại thì chọn hủy. Sau khi nhấn button thêm ta được kết quả như sau:

Graphical user interface, application, table

Description automatically generated with medium confidence

**c. Chức năng sửa thông tin**

Nếu muốn sửa, người dùng nhấn vào nút sửa. Khi đó, một popupbox sẽ hiển thị lên để người dùng sửa thông tin. Để xác nhận sửa thì chọn vào nút lưu, ngược lại thì chọn hủy.

Graphical user interface, table

Description automatically generated with medium confidence

**d. Chức năng xóa thông tin**

Nếu muốn xóa, người dùng nhấn vào nút xóa. Khi đó, một dialog sẽ hiển thị lên. Để xác nhận xóa thì chọn vào Accept, ngược lại thì chọn Cancel.

Graphical user interface, application

Description automatically generated

##### 3.2.5. Giao diện quản lý nhân viên

Table

Description automatically generated

Các chức năng thêm, sửa, xóa tương tự phần quản lý trên.

##### 3.2.6. Giao diện quản lý phòng ban

Graphical user interface, text

Description automatically generated

Các chức năng thêm, sửa, xóa tương tự phần quản lý trên.

##### 3.2.7. Giao diện quản lý chức vụ

Graphical user interface, text, application

Description automatically generated

Các chức năng thêm, sửa, xóa tương tự phần quản lý trên.

##### 3.2.8. Giao diện phân công

Graphical user interface, application

Description automatically generated

Các chức năng thêm, sửa, xóa tương tự phần quản lý trên.

##### 3.2.9. Giao diện quản lý tài khoản

Graphical user interface, application

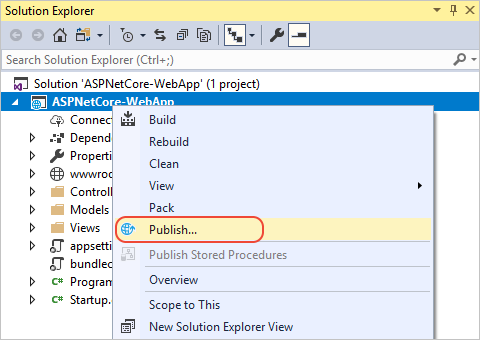
Description automatically generated

Các chức năng thêm, sửa, xóa tương tự phần quản lý trên.

#### 3.3. Cài đặt và kiểm thử

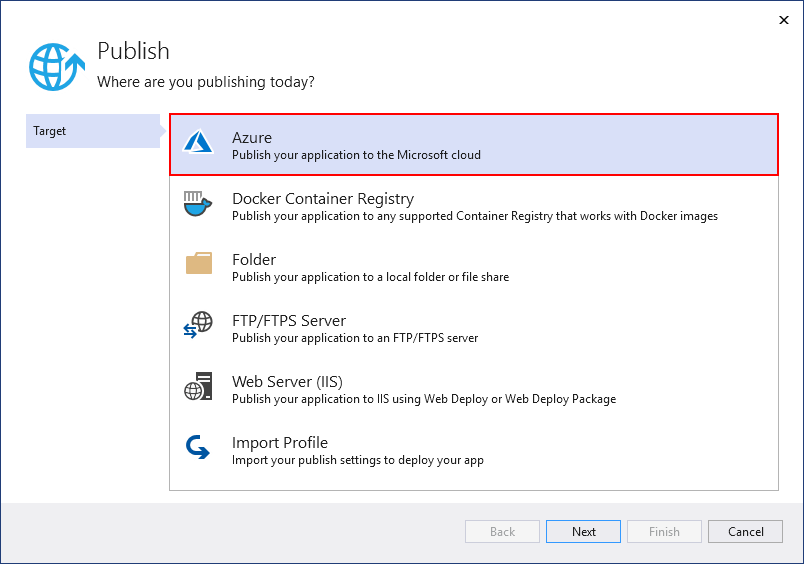
#### Cài đặt chương trình

* + Chuẩn bị:
* Tài khoản Azure.
* Hosting chạy apache hoặc nginx.
* Triển khai ứng dụng Backend lên server:
* Trong Solution Explorer, bấm chuột phải vào project và chọn **Publish** (hoặc chọn **Build** > **Publish** trên thanh menu).



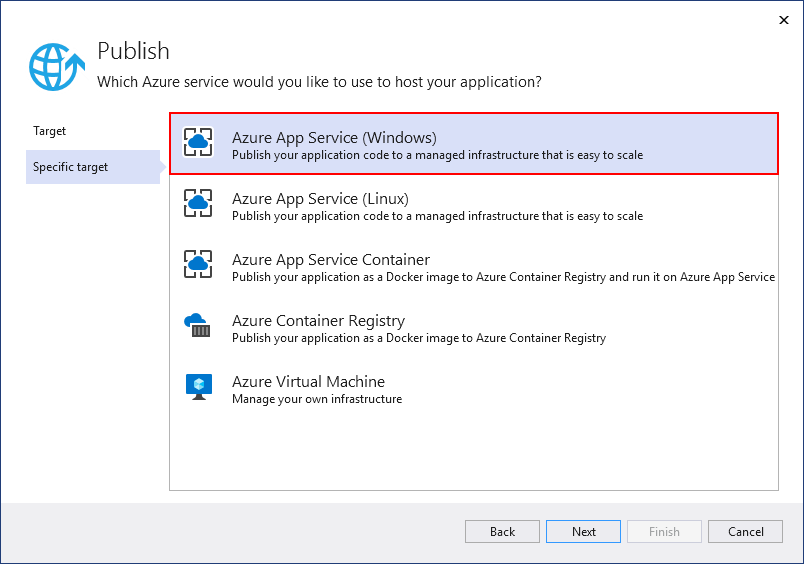
Hình 1‑1: Triển khai ứng dụng lên server - 1

* Nếu trước đó đã cài đặt bất kỳ cấu hình Publish nào, cửa sổ **Publish** sẽ xuất hiện. Chọn **New**.
* Trong cửa sổ **Publish**, chọn **Azure**.

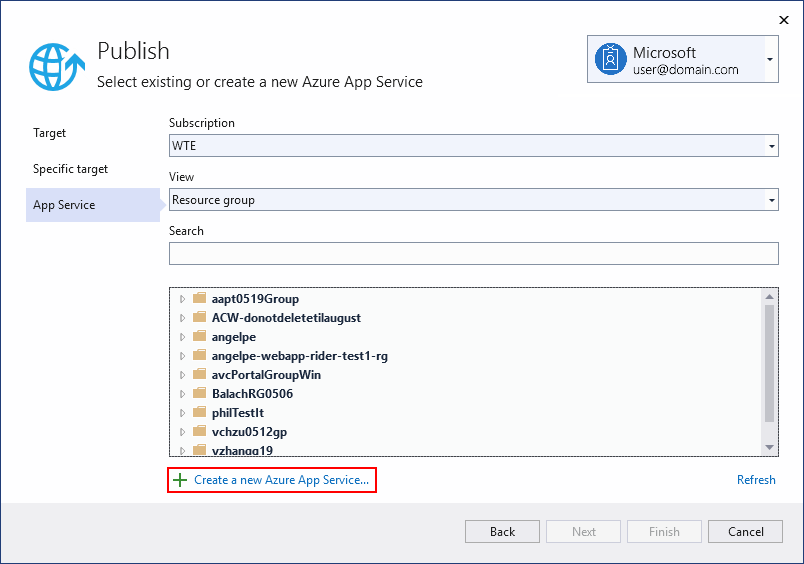


Hình 1‑2: Triển khai ứng dụng lên server - 2

* Chọn dịch vụ và ứng dụng Azure (Windows) và **Next.**



Hình 1‑3: Triển khai ứng dụng lên server - 3

* Đăng nhập bằng tài khoản Azure. Chọn **Create a new Azure App Service.**

Hình 1‑4: Triển khai ứng dụng lên server - 4

Graphical user interface, text, application, email

Description automatically generatedTrong hộp thoại **Create Azure App Service (Windows),** lựa chọn hoặc nhập các trường **App Name**, **Resource Group** và **App Service Plane**. Sau đó chọn **Create**.

Hình 1‑5: Triển khai ứng dụng lên server - 5

* Trong hộp thoại **Publish**, sau khi phiên bản mới đã được tạo tự động. Chọn **Finish**.

Graphical user interface, text, application, email

Description automatically generated

Hình 1‑6: Triển khai ứng dụng lên server - 6

* Chọn **Publish** để triển khai ứng dụng cho Azure App Service.

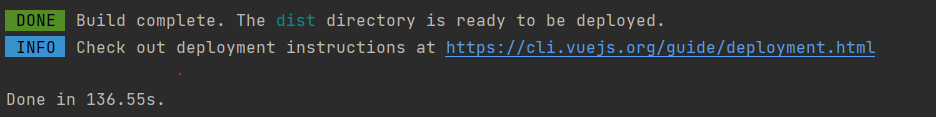
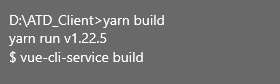
Graphical user interface, text, application, email

Description automatically generated

Hình 1‑7: Triển khai ứng dụng lên server - 7

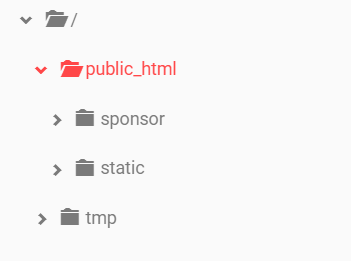
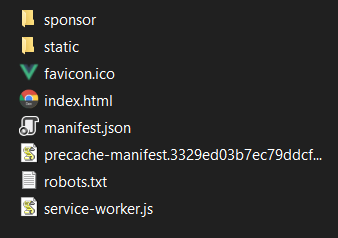
* Triển khai ứng dụng Frontend trên server:
* Trong thư mục project chạy lệnh yarn build để xuất bản thư mục dist.

Hình 1‑8: Triển khai ứng dụng lên server - 8



* Copy file trong thư mục dist vào thư mục public\_tml trên hosting

Hình 1‑9: Triển khai ứng dụng lên server - 9



#### Kiểm thử chương trình

* Kiểm thử chức năng thêm mới tài khoản user.
* Kiểm thử chức năng vô hiệu hóa, kích hoạt tài khoản.
* Kiểm thử chức năng điểm danh lớp học.
* Kiểm thử chức năng báo cáo, thống kê.
* Kiểm thử chức năng thêm, cập nhật, xóa dữ liệu.

#### Đánh giá hệ thống

* Hệ thống hoạt động tốt đối với chức năng thêm mới tài khoản user.
* Hệ thống có thể vô hiệu hóa, kích hoạt tài khoản.
* Chức năng thêm mới, cập nhật, xóa dữ liệu hoạt động ổn định.

**CHƯƠNG 4. KỸ NĂNG MỀM**

**4.1. Kỹ năng mềm là gì?**

Tùy thuộc vào lĩnh vực hoạt động, mỗi công ty có nhu cầu tuyển nhân viên có kỹ năng và kinh nghiệm khác nhau. Tuy nhiên, trong thời đại ngày nay, chỉ giỏi chuyên môn thôi là chưa đủ chúng ta cần phải có những kỹ năng mềm nhất định.

Kỹ năng mềm là một thuật ngữ xã hội học chỉ những kỹ năng có liên quan đến việc sử dụng ngôn ngữ, khả năng hòa nhập xã hội, thái độ và hành vi ứng xử áp dụng vào việc giao tiếp giữa người với người. kỹ năng mềm là những kỹ năng có liên quan đến việc hòa mình vào, sống với hay tương tác với xã hội, cộng đồng, tập thể hoặc tổ chức.

**4.2. Các kỹ năng mềm học được tại công ty**

Sau thời gian thực tập ở công ty em đã học được một số kỹ năng mềm:

- Tinh thần làm việc cao: Là người có động lực và dồn tâm sức để hoàn thành công việc, cho dù đó là việc khó. Luôn tận tâm và luôn muốn làm việc một cách tốt nhất.

- Thái độ tích cực: Luôn lạc quan và tích cực có thể tạo ra một nguồn năng lượng và ý chí dồi dào.

- Kỹ năng giao tiếp hiệu quả: Học được cách nói năng mạch lạc, khúc chiết và biết lắng nghe người khác.

- Khả năng quản lý thời gian: Biết cách sắp xếp các nhiệm vụ theo thứ tự ưu tiên và cùng lúc xử lý nhiều công việc khác nhau. Sử dụng thời gian làm việc một cách hiệu quả.

- Kỹ năng giải quyết vấn đề: Học được cách giải quyết các vấn đề phát sinh

- Có tinh thần đồng đội: Có khả năng làm việc theo nhóm. Tham gia các hoạt động ngoại khóa của công ty.

- Tự tin: Tin tưởng mình sẽ làm được việc. Tạo được cho những người xung quanh sự tin tưởng. Dũng cảm để đưa ra những câu hỏi cần thiết và thoải mái trình bày những ý tưởng của mình.

- Khả năng chấp nhận và học hỏi từ những lời phê bình: Biết cách xử lý những lời phê bình, chỉ trích.

- Linh hoạt và có khả năng thích nghi: Học được cách thích nghi với những tình huống mới và thách thức mới. Chấp nhận sự thay đổi và cởi mở với những ý tưởng mới.

- Khả năng làm việc độc lập: Tự tìm hiểu tài liệu và giải quyết các vấn đề phát sinh

- Xây dựng các mối quan hệ: Học được cách giao tiếp giữa các cá nhân với nhau tại nơi làm việc. Xây dựng tình bạn với các đồng nghiệp.

- Kỹ năng lắng nghe: Học được các lắng nghe người khác, không ngắt lời người khác để thể hiện sự tôn trọng của mình đối với người nói.

- Kỹ năng viết: Kỹ năng viết có nhiều thời gian cho mình chỉnh sửa và hoàn thiện trước khi gửi đi. Học được cách chỉnh sửa chính tả và cách sử dụng từ, thay đổi cấu trúc câu, viết xúc tích hơn thay vì dài dòng. Ghi nhận cách dung câu của những người khác để hỗ trợ cách viết của mình có chiều sâu, rõ ràng và mạch lạc hơn.

# CHƯƠNG 5: ****KẾT LUẬN****

Trong đời sống xã hội hiện nay, yêu cầu của con người đặt ra ngày càng cao, các công việc không những đòi hỏi phải được giải quyết nhanh chóng mà còn phải chính xác, việc “quản lý công nhân” mà em thực hiện đang nghiên cứu cũng vậy. Qua từng bước khảo sát hệ thống, xây dựng mô hình ERD, DFD, mô hình quan hệ thì hệ thống “Quản lí công nhân” đã ngày càng hoàn thiện hơn. Đã tin học hóa được việc quản lí công nhân thay vì phải quản lí thủ công bằng cách ghi chép sổ sách. Tạo sự thuận lợi, nhanh chóng cho việc nhập dữ liệu, tìm kiếm, tra cứu thông tin trở nên chính xác hơn với một giao diện trực quan thân thiện.

Sau 03 tháng thực hiện nghiên cứu đề tài, dưới sự hướng dẫn tận tình của thầy Lê Chí Luận, dựa vào việc phân tích những yêu cầu của bài toán, cũng như dựa trên những đặc tả của dữ liệu em đã hoàn thành thiết kế database cho bài toán, thiết kế giao diện cho phần mềm quản lí lao động. Các chức năng chính của hệ thống như yêu cầu đã đề ra như: các chức năng thêm, sửa xóa, tìm kiếm dữ liệu. Xem được chi tiết nội dung từng công nhân trên nền tảng website. Dễ dàng thêm mới, cập nhật các thông tin của công nhân, công trình, lịch làm việc, vật liệu,…

Thông qua quá trình thực hiện Đồ án tốt nghiệp với đề tài “Quản lý công nhân ”, bản thân em tự thấy mình đã thu được các kết quả như:

*Hiểu biết được phương pháp hướng cấu trúc, từ đó đã áp dụng các hiểu biết của mình để phân tích thiết kế bài toán.*

*Có được các kinh nghiệm thực tế khi được tham gia vào một dự án lớn cụ thể để có thế áp dụng được các kiến thức đã được học vào thực tiễn.*

*Tiến hành phân tích thiết kế hoàn thiện hệ thống bằng phương pháp hướng cấu trúc một cách đầy đủ.*

*Cài đặt được chương trình bằng ngôn ngữ Visual Basic 6.0 và hệ quản trị CSDL SQL Server 2000.*

Trong thời gian qua, em đã cố gắng hết sức để tìm hiểu thực hiện đề tài. Bên cạnh những gì đã làm được, phần nghiên cứu của nhóm vẫn tồn tại những điểm hạn chế như database mới dừng ở mức cơ bản của việc quản lí lao động, giao diện chưa được đẹp như kì vọng đơn giản và khó sử dụng. Các chức năng hiện tại chưa tối ưu, chưa cập nhật đầy đủ. Ngoài ra, do kiến thức về phân tích thiết kế hệ thống còn một số hạn chế, thời gian thực hiện ngắn nên hệ thống chưa được sửa chữa hoàn thiện. Nhóm rất mong được sự góp ý của thầy cô để hệ thống được hoàn thiện hơn. Chức năng hệ thống còn sơ sài.

Vì vậy trong thời gian tới nhóm mong rằng sẽ có những điều chỉnh để cho sản phẩm hoàn thiện hơn, đáp ứng được những yêu cầu cần phải có của một phần mềm quản lí, giao diện đẹp và thân thiện với người dùng hơn.

Đề tài này thật sự hữu ích đối với chuyên ngành công nghệ thông tin nhƣ một tài liệu tham khảo đề tài này có tính thực tế cao. Trong tương lai, nếu có điều kiện khách quan cho phép, em sẽ tiếp tục hoàn thiện và phát triển để triển khai hệ thống vào thực tế, giúp cho việc quản lý công nhân có thể diễn ra một cách dễ dàng.

Để hệ thống trong tương lai có thể đưa vào sử dụng tốt, có tính hiệu quả cao thì cần phải đạt được những mục tiêu như:

*Hoàn thiện toàn bộ chức năng của hệ thống.*

*Nâng cấp thêm chức năng (nếu có).*

*Tinh chỉnh giao diện cho thân thiện với người dùng, dễ sử dụng.*

*Tối ưu tốc độ load của trang web.*

*Cấu trúc lại chương trình để dễ dàng cho việc bảo trì, nâng cấp hệ thống.*

Trong quá trình hoàn thiện đồ án, do còn có nhiều sai sót mong được các thầy cô góp ý, giúp đỡ để chương trình của em ngày càng hoàn thiện hơn. Em xin chân thành cảm ơn!

### TÀI LIỆU THAM KHẢO

[1]. Bắt đầu với ASP.NET CORE (https://docs.microsoft.com/vi-vn/aspnet/core/)

[2]. AJAX và jQuery (https://www.w3schools.com/)

[3]. Bài giảng Phân tích thiết kế hệ thống đối tượng (Nguyễn Thanh Bình, Trường Đại học Bách Khoa, Đại học Đà Nẵng).

[4]. Lập trình C# cơ bản .NET CORE (https://tedu.com.vn/khoa-hoc/lam-du-an-voi-aspnet-core-31-34.html).