**HỌC VIỆN NGÂN HÀNG**

**KHOA CÔNG NGHỆ THÔNG TIN VÀ KINH TẾ SỐ**

A logo with a map and a square

Description automatically generated

**TIỂU LUẬN NHÓM**

**Học Phần: Lập Trình DotNet**

***ĐỀ TÀI*: XÂY DỰNG ỨNG DỤNG QUẢN LÝ CỬA HÀNG MÁY TÍNH**

**Giáo viên hướng dẫn: ThS. Lê Cẩm Tú**

**Lớp học phần: 232IS25A03 – Nhóm 14**

**Danh sách thành viên:**

|  |  |
| --- | --- |
| **Họ và tên** | **Mã sinh viên** |
| 1. Nguyễn Đức Công | 24A4042427 |
| 1. Lê Hương Giang | 24A4042434 |
| 1. Nguyễn Cảnh Phong | 24A4042605 |
| 1. Ngô Mạnh Thắng | 24A4042611 |
| 1. Nguyễn Thị Thảo Trang | 24A4041688 |

**Hà Nội, 5/2024**

**HỌC VIỆN NGÂN HÀNG**

**KHOA CÔNG NGHỆ THÔNG TIN VÀ KINH TẾ SỐ**

A logo with a map and a square

Description automatically generated

**TIỂU LUẬN NHÓM**

**Học Phần: Lập Trình DotNet**

***ĐỀ TÀI*: XÂY DỰNG ỨNG DỤNG QUẢN LÝ CỬA HÀNG MÁY TÍNH**

**Giáo viên hướng dẫn: ThS. Lê Cẩm Tú**

**Lớp học phần: 232IS25A03 – Nhóm 14**

**Danh sách thành viên:**

|  |  |  |
| --- | --- | --- |
| **Họ và tên** | **Mã sinh viên** | **Phần trăm đóng góp** |
| 1. Nguyễn Đức Công (Nhóm trưởng) | 24A4042427 | 22% |
| 1. Lê Hương Giang | 24A4042434 | 19.5% |
| 1. Nguyễn Cảnh Phong | 24A4042605 | 20% |
| 1. Ngô Mạnh Thắng | 24A4042611 | 18.5% |
| 1. Nguyễn Thị Thảo Trang | 24A4041688 | 20% |

**Hà Nội, 5/2024**

**LỜI CAM ĐOAN**

Tôi xin cam đoan kết quả đạt được trong báo cáo là sản phẩm nghiên cứu, tìm hiểu của riêng nhóm chúng tôi. Trong toán bộ nội dung của báo cáo, những điều được trình bày hoặc là của cá nhân tôi hoặc là được tổng hợp từ nhiều nguồn tài liệu. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Tôi xin hoàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

**SINH VIÊN THỰC HIỆN**

Nguyễn Đức Công

Lê Thị Hương Giang

Nguyễn Cảnh Phong

Ngô Mạnh Thắng

Nguyễn Thị Thảo Trang

**LỜI CẢM ƠN**

Trong suốt thời gian từ khi bắt đầu thực hiện bài thực tập đến nay, chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ của Khoa và quý thầy cô. Với lòng biết ơn sâu sắc, chúng em xin gửi lời cảm ơn đầu tiên đến quý thầy cô của Khoa Công Nghệ Thông Tin Và Kinh Tế Số – Học viện Ngân Hàng đã tạo điều kiện cho chúng em được học tập bộ môn “Lập trình DotNet” trong điều kiện tốt nhất. Kiến thức từ môn học này đã giúp nhóm có cái nhìn tổng quát nhất về “Lập trình hướng đối tượng” cũng như “ngôn ngữ lập trình C#” và cách thức triển khai một ứng dụng Winform.

Chúng em cũng xin đặc biệt gửi lời cảm ơn đến cô Lê Cẩm Tú đã tận tâm hướng dẫn, giúp đỡ chúng em trong suốt quá trình thực hiện bài báo cáo, từ những ngày đầu tiên cho đến khi hoàn thành. Nếu không có sự chỉ bảo của cô thì bài thu hoạch này của chúng em rất khó có thể hoàn thiện được. Một lần nữa, chúng em xin chân thành cảm ơn cô! Trong quá trình giải quyết bài toán, do kiến thức, lý luận cũng như kinh nghiệm thực tiễn của chúng em còn hạn chế nên bài báo cáo không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được góp ý quý báu của quý thầy, cô để bài báo cáo của em được hoàn thiện. Từ đó chúng em có thể rút ra được nhiều kinh nghiệm để hoàn thành tốt hơn những bài báo cáo sau này!

Sau cùng, chúng em xin kính chúc quý thầy, cô trong Khoa Công Nghệ Thông Tin Và Kinh Tế Số thật dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh trồng người.

Chúng em xin chân thành cảm ơn!

**MỤC LỤC**

[DANH MỤC HÌNH ẢNH 1](#_Toc166130963)

[DANH MỤC BẢNG BIỂU 2](#_Toc166130964)

[LỜI MỞ ĐẦU 3](#_Toc166130965)

[CHƯƠNG 1: TỔNG QUAN VỀ DỰ ÁN 4](#_Toc166130966)

[**1.1. CÁC CÔNG CỤ SỬ DỤNG 4**](#_Toc166130967)

[CHƯƠNG 2: TRIỂN KHAI DỰ ÁN 6](#_Toc166130968)

[**2.1. XÂY DỰNG VÀ THIẾT KẾ DATABASE 6**](#_Toc166130969)

[**2.1.1. Mô tả các thực thể và thuộc tính 6**](#_Toc166130970)

[**2.1.2. Mối quan hệ 6**](#_Toc166130971)

[**2.1.3. Chuẩn hóa CSDL 7**](#_Toc166130972)

[**2.1.4. Biểu đồ Diagram 9**](#_Toc166130973)

**DANH MỤC HÌNH ẢNH**

DANH MỤC BẢNG BIỂU

# LỜI MỞ ĐẦU

1. **TỔNG QUAN VỀ DỰ ÁN**
   1. **CÁC CÔNG CỤ SỬ DỤNG**

Nhìn chung nhóm sẽ sử dụng các công cụ tương tự như cô giáo đề xuất. Đại khái gồm có các công cụ sau:

* IDE - môi trường tích hợp phát triển: Visual Studio 2022 Community – Miễn Phí[[1]](#footnote-1)
* Framework: .NetFramework, ADO.Net Entity Data Model[[2]](#footnote-2)
* Database: SQL server 2022 và công cụ quản lý SSMS20v1[[3]](#footnote-3).
* Công cụ lưu trữ và quản lý mã nguồn: Github và Git (vì đây là thành quả của nhóm nên mình sẽ không để public repository này!)
* Ngoài ra còn có một số Packet cài thêm từ bên ngoài
  1. **DESIGN PATTERN**

Sau khi tham khảo ở nhiều nguồn khác nhau về kiến trúc, cách tổ chức thư mục thì nhóm đã quyết định chọn mô hình MVP(Model – View – Presenter). Giới thiệu sơ lược về mô hình này thì mô hình MVP gồm 3 phân lớp đó là:

* Model: Phân lớp này sẽ chứa các class liên quan đến dữ liệu và các logic kinh doanh liên quan đến dữ liệu. Có thể nói phân lớp này sẽ nặng về mặt “hướng đối tượng”. Nhiệm vụ chính của phân lớp này là thực hiện các truy xuất đến cơ sở dữ liệu.
* View: Phân lớp này sẽ chứa các form giao diện người dùng của ứng dụng nó sẽ chứa các lớp cung cấp việc hiển thị dữ liệu và xác định những hành động của người dùng sau đó thông báo tới phân lớp Presenter để xử lý logic.
* Presenter: Phân lớp này sẽ chứa các lớp liên quan tới việc xử lý logic trong phân lớp View. Presenter sẽ lấy dữ liệu từ lớp Model và sau đó áp dụng các logic đã được thiết lập sẵn trong phân lớp này để hiển thị liên View.

Điểm đặc biệt của mô hình này chính là việc Model và View sẽ không biết về sự tồn tại của nhau. Qua đó, hệ thống sẽ được đảm bảo an toàn về vấn đề bảo mật. Hơn thế nữa, nếu làm việc theo mô hình này thì tốc độ triển khai dự án sẽ nhanh chóng hơn. Nếu chẳng may có lỗi xảy ra thì chỉ có lỗi cục bộ chứ không hỏng luôn chương trình, điều này rất quan trọng trong việc rút ngắn thời gian debug.

Một khả năng tiếp theo cũng đáng để nhắc tới đó chính là khả năng mở rộng ứng dụng cực kỳ linh hoạt của model này. Đơn giản là bạn chỉ cần thêm các Form và chức năng tương ứng mà không ảnh hưởng đến phần cũ.

Phần Model kết hợp với Entity Framework sẽ cực kỳ mạnh mẽ trong việc truy vấn cũng như kết hợp với phân lớp phía trên, chưa kể nếu như trong Database có gì cập nhật thì chỉ cần một vài thao tác cũng có thể update thành công cũng như là không phải tạo Connection thủ công như phương pháp cũ nữa.

Chưa kể đến việc khi ta sử dụng được EF thì sẽ khắc phục được nhược điểm chí tử của lỗ hổng bảo mật SQL Injection do quá trình dùng các câu truy vấn SQL bằng string. Cụ thể, EF sẽ sử dụng kết hợp với Linq qua đó đưa người dùng tìm đến đúng đích, đúng dữ liệu mà không có hơn. Một phần cũng hạn chế được sự tấn công của các đối tượng khả khi vào hệ thống.

1. **TRIỂN KHAI DỰ ÁN**
   1. **XÂY DỰNG VÀ THIẾT KẾ DATABASE**
      1. **Mô tả các thực thể và thuộc tính**

* Loa (Mã loa, tên loa)
* Loại Máy ( Mã loại máy, tên loại máy)
* Chip( Mã chip, tên chip)
* Ổ cứng (Mã Ổ cứng, tên ổ cứng)
* Dung lượng (Mã dung lượng, tên dung lượng)
* Tốc độ (Mã tốc độ, tên tốc độ)
* OCD (Mã ổ, tên ổ)
* RAM (Mã ram, tên ram)
* Màn hình (Mã màn hình, tên loại MH)
* Cỡ MH (Mã cỡ MH, tên cỡ MH)
* Chuột (Mã chuột, tên chuột)
* Bàn phím (Mã bàn phím, tên bàn phím)
* USB (Mã ổ USB, tên ổ usb, dung lượng)
* Hãng sản xuất (Mã hãng sx, tên hãng sx)
* Máy vi tính (Mã máy VT, tên máy VT, giá nhập, giá bán, thời gian BH, số lượng, ảnh, ghi chú)
* Nhà cung cấp (Mã NCC, tên NCC, địa chỉ, điện thoại)
* Nhân viên (Mã NV, tên NV, ngày sinh, giới tính, địa chỉ, điện thoại, mật khẩu)
* Khách hàng (Mã khách, tên khách, địa chỉ, điện thoại)

### Mối quan hệ

* Máy vi tính <có> Loa (1:M)
* Máy vi tính <có> Loại máy (1:M)
* Máy vi tính <có> Chip (1:M)
* Máy vi tính <có> Ổ cứng (1:M)
* Máy vi tính <có> Dung lượng (1:M)
* Máy vi tính <có> Tốc độ (1:M)
* Máy vi tính <có> Ổ CD (1:M)
* Máy vi tính <có> RAM (1:M)
* Máy vi tính <có>  Màn hình (1:1)
* Máy vi tính <có> Cỡ MH (1:M)
* Máy vi tính <có> Chuột (1:M)
* Máy vi tính <có> Bàn phím (1:M)
* Máy vi tính <có> USB (1:M)
* Máy vi tính <có> Hãng sản xuất (1:1)
* Nhân Viên <nhập> Máy vi tính(1:M): (Mã HDN, ngày nhập, số lượng, thành tiền, tổng tiền)
* Khách hàng <mua> Máy vi tính (1:M) (Mã HDB, Ngày bán, số lượng, thành tiền, tổng tiền)
* Máy vi tính <thuộc> Nhà cung cấp (1:M)

### Chuẩn hóa CSDL

* Máy vi tính (Mã máy VT, tên máy VT, giá nhập, giá bán, thời gian BH, số lượng, ảnh, ghi chú, Mã loa, Mã loại máy, Mã chip, Mã màn hình, Mã cỡ MH, Mã ổ cứng, Mã USB, Mã dung lượng, Mã tốc độ, Mã ổ CD, Mã bàn phím, Mã chuột, Mã RAM, Mã hãng sản xuất)

Ta có bảng Hóa đơn nhập được sinh ra từ hoạt động dựa trên mối quan hệ giữa nhân viên và Máy vi tính, Hóa đơn bán dựa trên mối quan hệ giữa Khách hàng và Máy vi tính:

* Hóa đơn nhập (Mã máy vi tính, Mã HDN, ngày nhập, số lượng, đơn giá, thành tiền, tổng tiền, Mã NV, Mã NCC)
* Hóa đơn bán (Mã máy vi tính, Mã HDB, ngày bán, số lượng, tổng tiền, thành tiền, Mã NV,  Mã khách)

Vì Thành tiền là thuộc tính dẫn xuất có thể tính được từ số lượng\*đơn giá nên ta cần loại bỏ nó để không bị vi phạm chuẩn 1NF. Hóa đơn nhập và Hóa đơn bán chưa đạt chuẩn dạng chuẩn 2NF cần đưa về dạng 2NF thì cần tách thành các bảng sau đây:

* Chi tiết HDN (Mã máy vi tính, Mã HĐN,  số lượng, đơn giá)
* Hóa đơn nhập( MaHĐN, ngày nhập, tổng tiền,  MãNV, MaNCC)
* Chi tiết HDB(Mã máy vi tính, Ma HĐB, số lượng, đơn giá)
* Hóa đơn bán (Ma HĐB, ngày bán, tổng tiền, Mã NV, Mã khách)

Để đảm bảo vấn đề bảo mật, không cho nhân viên khác xem được thông tin tài khoản và mật khẩu. Nhóm sẽ tách trường thuộc bảng nhân viên thành một bảng riêng biệt là LoginAccount.

* LoginAccount(Mã NV , mật khẩu)

Dữ liệu trên đã ở dạng chuẩn 2NF và cũng đã đạt chuẩn 3NF. Vậy ta có dữ liệu đã được chuẩn hóa là:

* Loa (Mã loa, tên loa)
* Loại Máy ( Mã loại máy, tên loại máy)
* Chip( Mã chip, tên chip)
* Ổ cứng (Mã Ổ cứng, tên ổ cứng)
* Dung lượng (Mã dung lượng, tên dung lượng)
* Tốc độ (Mã tốc độ, tên tốc độ)
* OCD (Mã ổ, tên ổ)
* RAM (Mã ram, tên ram)
* Màn hình (Mã màn hình, tên loại MH)
* Cỡ MH (Mã cỡ MH, tên cỡ MH)
* Chuột (Mã chuột, tên chuột)
* Bàn phím (Mã bàn phím, tên bàn phím)
* USB (Mã ổ USB, tên ổ usb, dung lượng)
* Hãng sản xuất (Mã hãng sx, tên hãng sx)
* Máy vi tính (Mã máy VT, tên máy VT, giá nhập, giá bán, thời gian BH, số lượng, ảnh, ghi chú, Mã loa, Mã loại máy, Mã chip, Mã màn hình, Mã cỡ MH, Mã ổ cứng, Mã USB, Mã dung lượng, Mã tốc độ, Mã ổ CD, Mã bàn phím, Mã chuột, Mã RAM, Mã hãng sản xuất)
* Nhà cung cấp (Mã NCC, tên NCC, địa chỉ, điện thoại)
* Nhân viên (Mã NV, tên NV, ngày sinh, giới tính, địa chỉ, điện thoại)
* LoginAccount(Mã NV, mật khẩu)
* Khách hàng (Mã khách, tên khách, địa chỉ, điện thoại)
* Hóa đơn nhập( Mã HĐN, ngày nhập, tổng tiền,  Mã NV, MaNCC)
* Chi tiết HDN (Mã máy vi tính, Mã HĐN,  số lượng, đơn giá)
* Hóa đơn bán (Ma HĐB, ngày bán, tổng tiền, Mã NV, Mã khách)
* Chi tiết HDB(Mã máy vi tính, Ma HĐB, số lượng, đơn giá)

### Biểu đồ Diagram

A screenshot of a computer screen

Description automatically generated

Hình 1: Lược đồ quan hệ của database

* 1. **XÂY DỰNG KHUNG CÂY THƯ MỤC CHO DỰ ÁN**
     1. **Xây dựng theo mô hình MVP (Model – Presenter - View)**
        1. Model

Ở trong thư mục Model sẽ là các class liên quan tới các logic nghiệp vụ, mối quan hệ, ràng buộc hay các cách thức để lấy được dữ liệu từ database lên.

Bình thường, chúng ta sẽ phải áp dụng lập trình hướng đối tượng để tạo ra các hiện thực hóa các đối tượng đó, rồi sau đó mới sử dụng tới các hàm và phương thức để “hứng” dữ liệu từ database đổ vào.

Nhưng tại sao chúng ta phải làm quá “basic” như thế trong khi Visual Studio đã hỗ trợ sẵn một thứ để làm công việc như vậy. Đó chính là EntityFrameWork, tuy ở trên lớp không được giới thiệu qua nhưng công cụ mạnh mẽ như vậy mà không sử dụng thì lại hơi uổng phí công sức của các nhà thiết kế đã tạo ra chúng.

Khi ta sử dụng công cụ này, đồng nghĩa với việc Visual Studio sẽ giúp chúng ta sẽ tạo ra tất cả các thực thể và các bảng có liên quan trong database, giống y hệt như cách chúng ta tạo ra các class để “hứng” data từ database vậy, qua đó làm giảm một khối lượng công việc khổng lồ trong bài báo cáo.

Giới thiệu đến EF[[4]](#footnote-4) cũng đã dài, sau đây nhóm sẽ hướng dẫn cách để cài đặt EF làm Model trong mô hình MVP:

* + - * 1. Cách cài đặt EF từ DB có sẵn.

Bước 1: Tạo thư mục Model trong project. Trong cửa sổ solution Explore, bấm phải chuột vào dự án. Chọn Add=> chọn New Folder => Đặt tên cho thư mục là Model.

A screenshot of a computer

Description automatically generated

Hình 2: Thêm thư mục Model

Bước 2: Nhấn phải chuột vào Folder=> chọn Add=> chọn New Item. Trong cửa sổ tìm kiếm phía trên cùng bên phải, các bạn hãy gõ Entity. Sau đó chọn ADO.NET Entity Data Model. Đặt tên cho nó rồi nhấn Enter.

A screenshot of a computer

Description automatically generated

Hình 3: Tạo Entity Data Model

Bước 3: Chọn EF Desginer from Database=> Next.

A screenshot of a computer

Description automatically generated

Hình 4: Chọn EF Designer From Database

Bước 4: Chọn New Connection. Ở bước này các bạn làm giống hệt như khi các cách bạn kết nối tới cơ sở dữ liệu mà mình đã học ở trên lớp.

Bước 5: Khi đã kết nối tới cơ sở dữ liệu xong thì các bạn hãy đặt tên cho EF này, ở đây mình sẽ đặt tên cho nó là Entity (lưu ý cái tên này sẽ được sử dụng để tương tác với dữ liệu và nó được lưu trữ trong App.config dưới dạng file XML). Sau đó nhấn Next.

A screenshot of a computer

Description automatically generated

Hình 5: Tạo Entity

Bước 6: Chọn EF 6x (mặc định). Nhấn Next, tích chọn vào Table rồi finish. Sau khi tạo xong thì… Bùm!!!!

A screenshot of a computer

Description automatically generated

Hình 6: Database Diagram trong Visual Studio.

Vừa rồi là các bước thực hiện quá trình tạo ra EF ở trong thư mục Model. Về cách sử dụng nó trong thư mục này mình sẽ đề cập đến trong các chương sau.

* + - 1. Presenter

Như đã giới thiệu ở phần bên trên, phân lớp Presenter sẽ chứa các logic hoạt động mà người dùng có thể thực hiện ở trong View. Nhiệm vụ của nó là lấy dữ liệu từ lớp Model và trả lại cho phân lớp view. Vậy trong thư mục này có gì?

Đầu tiên, các bạn hãy tạo một thư mục có tên Presenter. Trong thư mục này sẽ là các class sử dụng để tương tác, gọi đến lớp Model để lấy dữ liệu của nó lên. Đặc điểm của phân lớp này là các Item chứa trong nó hầu hết là các class C# và các phương thức của nó đều là các phương thức trả về dữ liệu. Thường thì đối với mỗi Form mà phân lớp View có sẽ tương ứng có một class để diễn tả sự trình bày của phân lớp View. Ví dụ như hình sau:

A screenshot of a computer

Description automatically generated

Hình 7: Các Item trong thư mục Presenter

Ngoài ra, các class này còn có một đặc điểm chung đó chính là **Đều phải có** dòng code:

**using [Tên Project].Model;**

Và bên trong Class tương ứng của nó là dòng code

**Entity db= new Entity();**

Công dụng của hai dòng này là:

* + Dòng thứ nhất: gọi thư mục mà chúng ta muốn sử dụng. Trong trường hợp này là thư mục Model nơi chứa EF của chúng ta.
  + Dòng thứ 2: khởi tạo một biến Entity để tương tác với EF. “Entity” ở đây chính là tên mà mình đã đặt ở bước tạo EF trong Model, nếu các bạn đặt tên khác thì phải gọi đúng cái tên đó để đặt tên biến.

A screenshot of a computer program

Description automatically generated

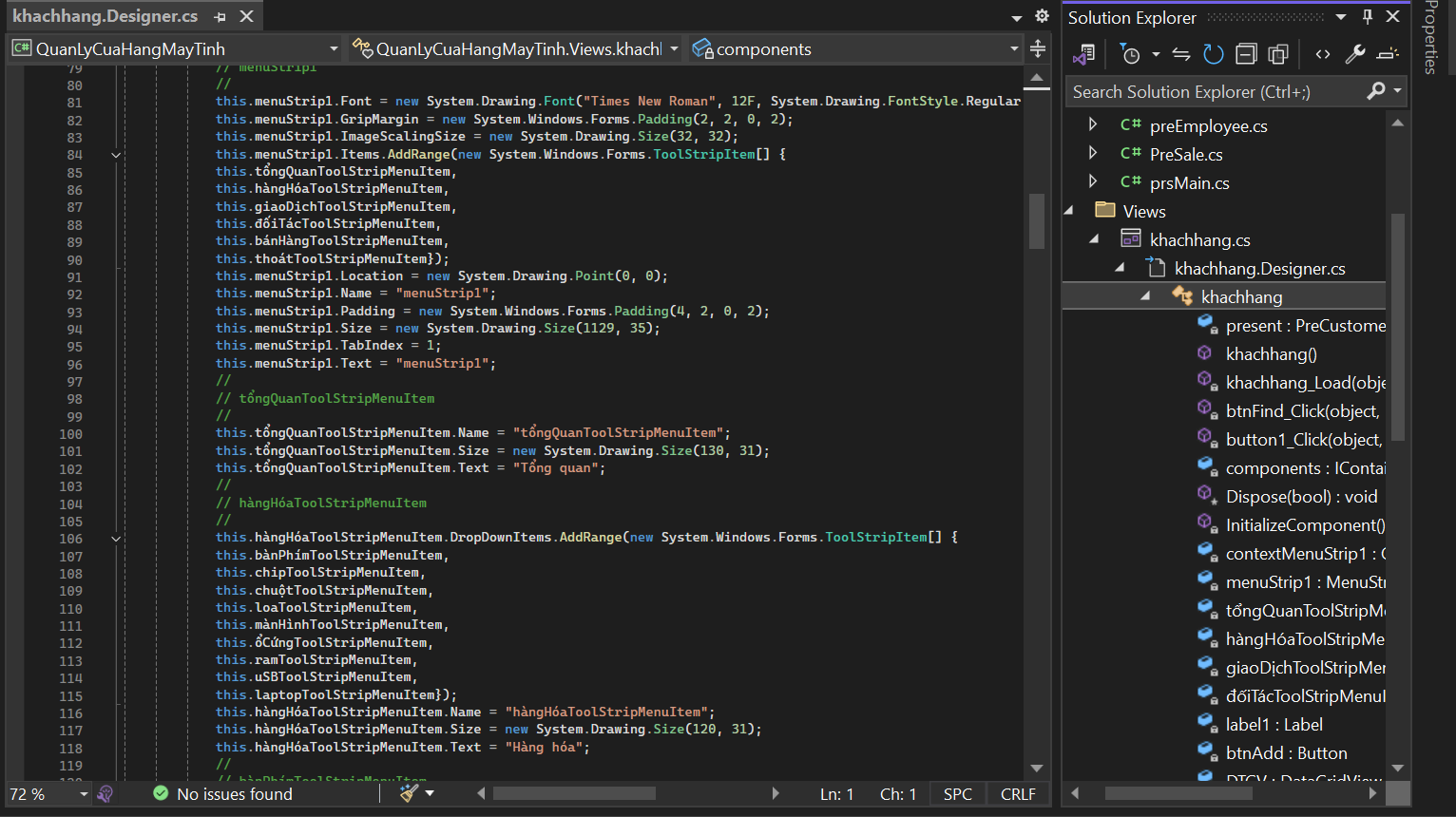
Hình 8: Dòng code mà ở class nào trong Presenter hầu như cũng có.

* + - 1. View

Phân lớp View thì có vẻ đơn giản hơn đối với các bạn khi mà nó chứa các Form giao diện như chúng ta đã học ở trên lớp.

Có thể các bạn chưa biết một Form giao diện sẽ chia thành 3 phân vùng gồm có các vùng sau:

* + Vùng giao diện: đó chính là phần giao diện mà chúng ta code.
  + Vùng code giao diện: đó chính là File code Design.cs, File code này chính là sự code hóa của vùng giao diện. Khi bạn làm bất cứ thứ gì liên quan tới vùng giao diện thì vùng code giao diện cũng sẽ thay đổi theo.
  + Vùng Code sự kiện: đó chính là thứ mà khi chúng ta doubleclick vào bất kì một component nào trong form khi đang code thì nó sẽ hiện ra. Các bạn có thể tìm thấy nó ở dưới đây.



Hình 9: Vùng code sự kiện

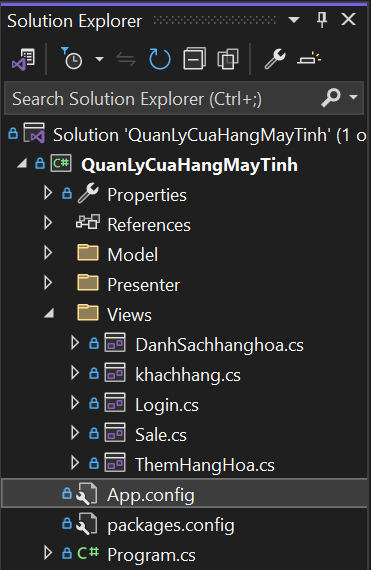
* + Khi bạn doubleclick vào các Item trong phần cửa sổ bên phải sổ xuống thì code của giao diện sẽ hiện ra. Vùng code này có mối liên hệ tương tác mật thiết với vùng code giao diện, vùng code giao diện sẽ sử dụng các method trong này để tiến hành thực thi sự kiện. Vì thế cho nên nhiều khi các bạn ấn doubleClick vào một Component mà lại xóa phương thức đó đi thì ngay lập tức bên vùng giao diện sẽ báo lỗi. Lỗi này là do vùng code giao diện đã lấy phương thức của vùng xử lý sự kiện để chạy nhưng lại không tìm thấy nó trong code sự kiện (vì bạn xóa đi rồi còn đâu mà tìm được). Để sửa lỗi này thì cực kỳ đơn giản, các bạn hãy tìm đến dòng code lỗi trong phần code giao diện rồi xóa nó đi là xong.
    1. **Cách chuyển phần code của mình cho người khác khi sử dụng EF**
       1. Chuyển cả Project

Để chuyển cả Project cho các thành viên trong nhóm, nhóm mình quyết định sử dụng các công cụ quản lý phiên bản là Git và kho lưu trữ mã nguồn Github. Cách sử dụng Github và Git thì có lẽ các bạn sẽ biết rồi nên mình sẽ không nhắc tới trong bài này. Nếu các bạn chưa biết thì các bạn có thể xem [tại đây!](https://www.youtube.com/watch?v=beuKZ9PW_rI&list=PLyxSzL3F7485Xgn7novgNxnou8QU6i485)

* + - 1. Chuyển đổi EntityFrameWork

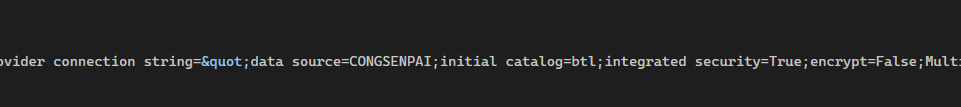
Nếu như bạn Clone Repository từ Github về thì chắc chắn các bạn sẽ không chạy được đâu, bởi vì đơn giản là Project này đang được thiết lập cho Database của các thành viên trong nhóm bạn chứ không phải là thiết lập cho máy của bạn, nếu bạn Clone lần đầu thì đây là giải pháp cho bạn để có thể chạy chương trình trên máy của mình một cách đơn giản nhất.

* + Bước 1: Cài đặt Database tại máy của mình. Đây là phần mà khi học trên lớp ai cũng phải biết mình sẽ không nhắc lại[[5]](#footnote-5).
  + Bước 2: mở Project trong VS[[6]](#footnote-6) trên máy local. Trong cửa sổ Solution Explore tìm đến file App.Config[[7]](#footnote-7). Sau đó mở nó ra.



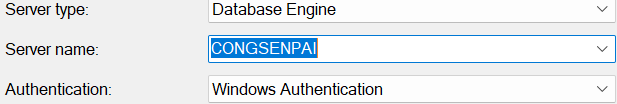
Hình 10: App.config trong Solution Explore

* + Bước 3: Nhìn xuống phía cuối File sẽ xuất hiện cặp thẻ <connectionString></connectionString>. Chúng ta sẽ kéo thanh cuộn của trang sang phía bên phải. Tìm đến đoạn data source =…



Hình 11:Đoạn code có chứa Data source

* + Bước 4: Bật SSMS lên trong cửa sổ SSMS[[8]](#footnote-8) tại dòng Server name. Các bạn hãy copy đoạn text này của mình rồi sau đó paste vào phần data source trong File App.config vừa tìm là đã hoàn thành.



Hình 12: Tên server trong SSMS

* + - 1. Cách gộp Form của các thành viên khác vào Project của mình.

Nếu các bạn gặp phải lỗi liên quan đến Github mà không biết cách khắc phục thì bạn có thể sử dụng phương pháp thủ công là gửi File zip Project trực tiếp cho nhau. Tuy nhiên cần phải chú ý, VS chỉ hỗ trợ copy theo kiểu chỉ định tức là nếu bạn copy Form (theo cách thông thường) thì sẽ không copy được file thiết kế. Để thực hiện quá trình Copy thì các bạn hãy làm theo các bước sau:

* + Bước 1: Mở Project trên máy của bạn. Trong cửa sổ Solution Explore, tìm đến thư mục View (vì mình xây dựng dự án theo MVP nên Form sẽ được lưu tại đây). Nhấn phải chuột chọn Add => Existing Item.

A screenshot of a computer

Description automatically generated

Hình 13 :Thêm Item đã tồn tại

* + Bước 2: Tìm đến thư mục View của Project mà thành viên khác trong nhóm gửi cho bạn. Sau đó chọn Form bạn muốn thêm. Ví dụ ở đây mình sẽ chọn Form login. Nhớ là hãy chọn file Login.cs để không xuất hiện hiệu quả ngoài ý muốn.

A screenshot of a computer

Description automatically generated

Hình 14: Chọn Form Login

* + Bước 3: Nếu chương trình của bạn chạy bị lỗi thì hãy nhấn vào Design của form vừa thêm. Nhấn F7 để vào phần code, theo dõi các dòng code xuất hiện trong bài, rồi sau đó kiểm tra xem các đoạn code đó đã có trong thư mục Presenter chưa. Nếu chưa có thì các bạn cũng làm tương tự các bước trên để copy các File Presenter tương ứng vào Project của mình hoặc chỉ Copy các đoạn code mà mình cần. Chúc các bạn thành công!

1. **XÂY DỰNG CÁC TÍNH NĂNG CHO PROJECT**
   1. **Chức năng đăng nhập**
   2. **Xây dựng MenuStrip**
   3. **Hàng hóa**
   4. **Giao dịch**
   5. **Đối tác**
   6. **Bán hàng**
   7. **Nhập hàng**
   8. **Tổng quan**

1. :https://visualstudio.microsoft.com/vs/community/ [↑](#footnote-ref-1)
2. Entity Framework [↑](#footnote-ref-2)
3. Phiên bản 20+ năm 2024 [↑](#footnote-ref-3)
4. Entity FrameWork [↑](#footnote-ref-4)
5. Có 3 cách để cung cấp Database cho người khác [↑](#footnote-ref-5)
6. Visual Studio [↑](#footnote-ref-6)
7. Đây là File chứa cấu hình cho ứng dụng của bạn, được viết dưới dạng XML [↑](#footnote-ref-7)
8. SQL Server Management Studio [↑](#footnote-ref-8)