**MÔ TẢ KỸ THUẬT / CÔNG NGHỆ ĐỒ ÁN**

# .Net Framwork

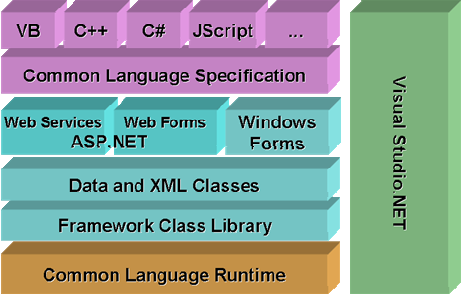
## Giới thiệu

.Net Framework là một nền tảng phát triển phần mềm được Microsoft tạo ra [1].

Chủ yếu chạy trên hệ điều hành Windows, được hình thành vào năm 2002.

.Net Framework có nhiều phiên bản từ lúc ra đời cho đến nay, khởi đầu là .Net Framework 1.0, và qua quá trình cập nhập, phát triển đến hiện tại là phiên bản .Net Framework 4.8 [2].

.Net Framework được sử dụng khá là rộng rãi trong việc tạo ra các ứng dụng, các website từ các nền tảng như Form, Web và các thư viện có sẵn để viết, tạo ra sản phẩm.



Hình 1.1. Giới thiệu .NET Framework

## Một số bộ thư viện trong .Net Framework

* WinForms: Là bộ thư viện dùng để viết ra các ứng dụng trên windows, sử dụng giao diện là các Form control và tiến hành xử lý các yêu cầu từ giao diện xuống database.
* ASP.Net: Là bộ thư viện được sử dụng để viết các ứng dụng web, chạy và hiển thị trên trình duyệt.
* ADO.Net: Là bộ thư viện giúp trao đổi qua lại với cơ sở dữ liệu.

## Một số ưu điểm của .Net Framework

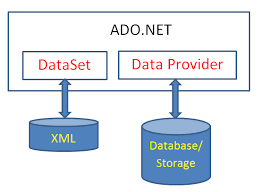
* Có khả năng trao đổi qua lại giữa các phiên bản, vẫn chạy được ứng dụng khi phiên bản cũ hơn chạy trên phiên bản mới hơn.
* Có sự linh hoạt cao, chạy được trên toàn bộ các nền tảng của Windows.
* Có tính bảo mật cao, có các cơ chế bảo mật để chứng thực trong ứng dụng, cấp quyền cho người dùng.
* Có khả năng quản lý bộ nhớ tối ưu, chỉ sử dụng các tài nguyên được gọi đến và giải phóng các tài nguyên không sử dụng nữa.
* Dễ sử dụng, triển khai ứng dụng, các sản phẩm được đóng gói, khi đến người dùng nó sẽ tự động cài đặt ứng dụng.

# ADO.Net

ADO.Net là các lớp và các phương thức trao đổi, kết nối với cơ sở dữ liệu, liên kết và truy xuất dữ liệu [3].

Các loại dữ liệu được lấy sẽ thông qua các loại thư viện khác nhau, gọi chung là Data Provider.

Dữ liệu được lưu trữ và chỉnh sửa nhờ DataSet.

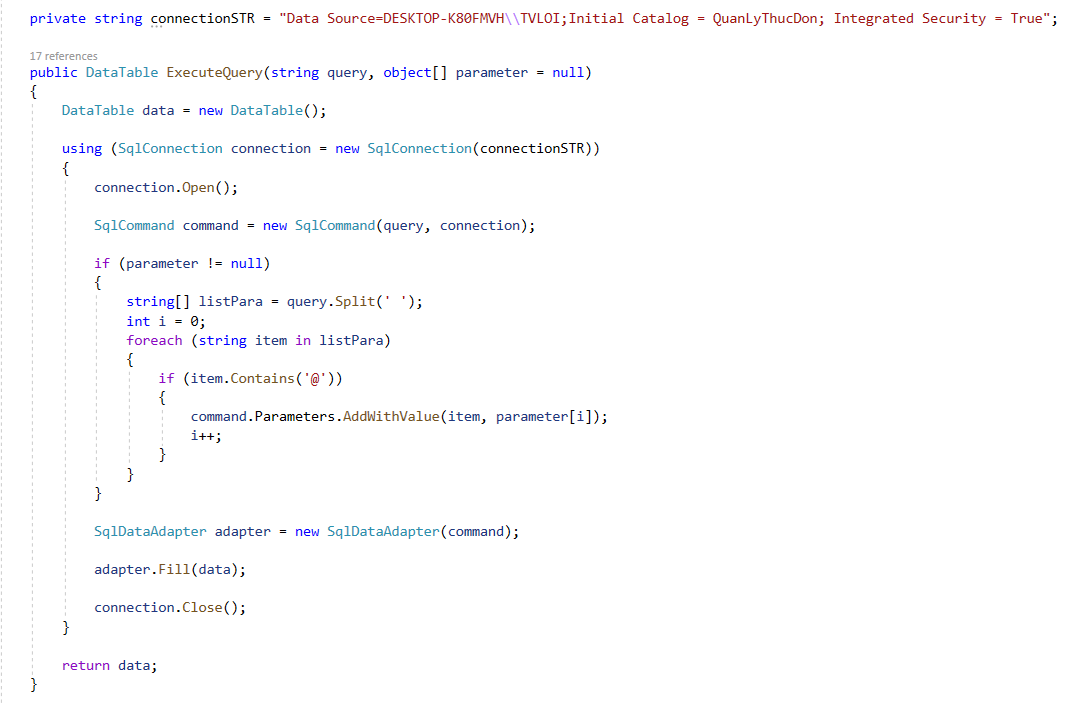


Hình 2.1. ADO.Net

Các đối tượng thường được sử dụng:

* SqlConnection: dùng để liên kết với SQL server.
* SqlCommand: dùng để tiến hành các câu lệnh query.
* SqlDataAdapter: dùng để kết nối data set với data source.
* SqlException: dùng để xử lý ngoại lệ.

…



Hình 2.2. Áp dụng ADO.Net

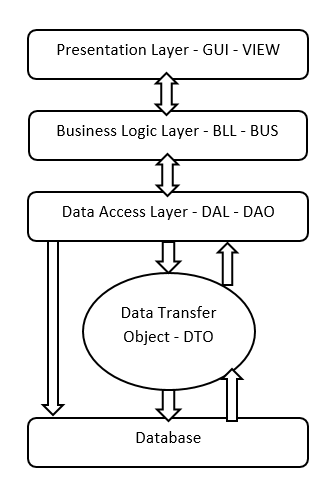
Đây là hàm ExecuteQuery có thực thi các câu lệnh của ADO.Net để truy vấn cơ sở dữ liệu trả về dữ liệu dạng bảng – DataTable.

Sử dụng SqlConnection để tạo kết nối với cơ sở dữ liệu nguồn, tạo mới đối tượng SqlCommand để thực thi câu truy vấn và sử dụng SqlDataAdapter để chuyển dữ liệu lấy được từ cơ sở dữ liệu nguồn thành dữ liệu lưu trên hệ thống.

# Mô hình 3 lớp

Mô hình 3 lớp (3-Layer) là mô hình phân nhóm các thành phần trong hệ thống, các thành phần có chức năng giống nhau sẽ được gộp lại để dễ quản lý [4].

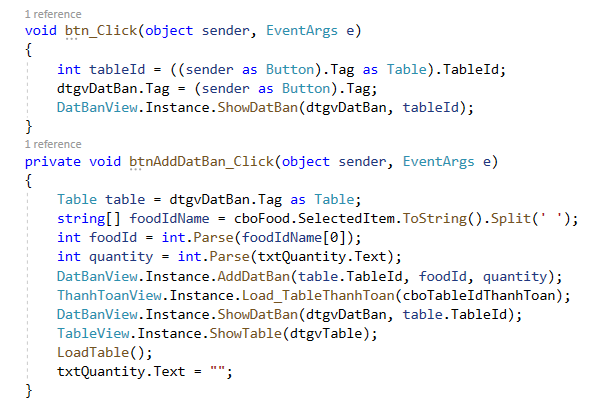
Mô hình giúp chúng ta quản lý và thực hiện dễ dàng hơn, phân loại các hành động, các truy xuất dữ liệu.



Hình 3.1. Mô hình 3 lớp

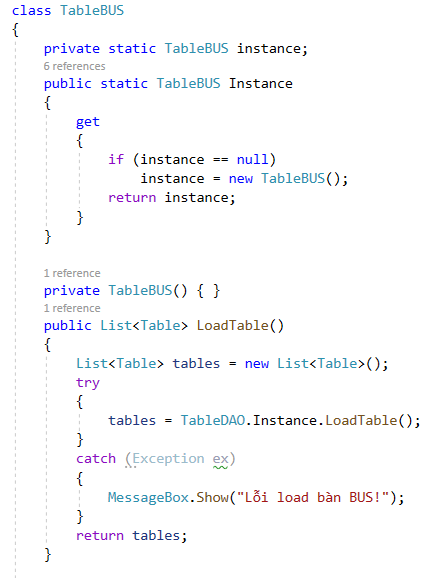
Mô hình 3 lớp gồm 3 thành phần chính:

* Presentation Layer (GUI): là lớp có nhiệm vụ trao đổi, giao tiếp với người dùng, xử lý trên giao diện (WinForm), thực thi một số thao tác nhập, xuất, kiểm tra dữ liệu để xử lý ở tầng sau.



Hình 3.2. Thao tác với lớp GUI

* Lớp GUI thao tác với các thành phần, control của giao diện winform để xuất dữ liệu người dùng cần, kiểm tra dữ liệu đầu vào của người dùng để thao tác với các lớp dưới.
* Business Logic Layer (BLL/BUS): là nơi thao tác, xử lý nguồn dữ liệu từ giao diện (GUI) trước khi làm việc với database hệ quản trị cơ sở dữ liệu, kiểm tra dữ liệu đầu vào, ra, xử lý các nghiệp vụ.



Hình 3.3. Tạo đối tượng Table trong BUS

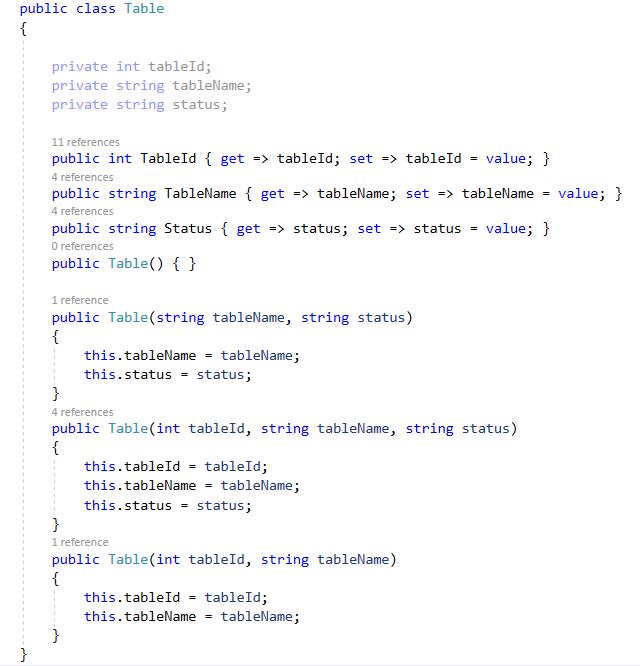
* Lớp TableBUS được tạo ra với mục đích kiểm tra dữ liệu trả về từ lớp DAO, nếu dữ liểu đúng thì sẽ được đưa đi sử dụng cho lớp GUI.
* Data Access Layer (DAL/DAO): là lớp truy vấn, lưu trữ, thực thi với cơ sở dữ liệu.



Hình 3.4. Tạo đối tượng Table trong DAO

* Lớp TableDAO với mục đích tiến hành thực thi các câu truy vấn liên quan đến các thông tin về bàn, lưu trữ dữ liệu thông qua đối tượng Table đã khởi tạo ở lớp DTO, xử lý các thao tác với cơ sở dữ liệu như đọc, thêm, sửa, xóa, lưu trữ dữ liệu đã được xử lý nếu có thay đổi về cơ sở dữ liệu nguồn.

Ngoài ra còn có lớp Data Transfer Object (DTO), là lớp khởi tạo đối tượng, lớp trung gian để lưu trữ dữ liệu thuận tiện khi gọi đến ở các tầng khác.



Hình 3.5. Tạo đối tượng Table trong DTO

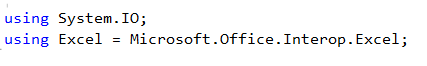
* Lớp Table là nơi lưu trữ các thông tin liên quan đến đối tượng bàn để thuận tiện cho các lớp khác cần sử dụng khi gọi đến.

# Microsoft.Office.Interop.Excel

Microsoft.Office.Interop.Excel là một thư viện với chức năng liên kết với trang tính Excel [5].

Sử dụng các phương thức có sẵn của thư viện, viết code để truyền dữ liệu cần hiển thị trên Excel, tương tác và lưu trữ trên Excel.

Một số phương thức, sự kiện có sử dụng trong thư viện: Application, Worksheet, Workbook, Add, Saveas, Close, Quit.



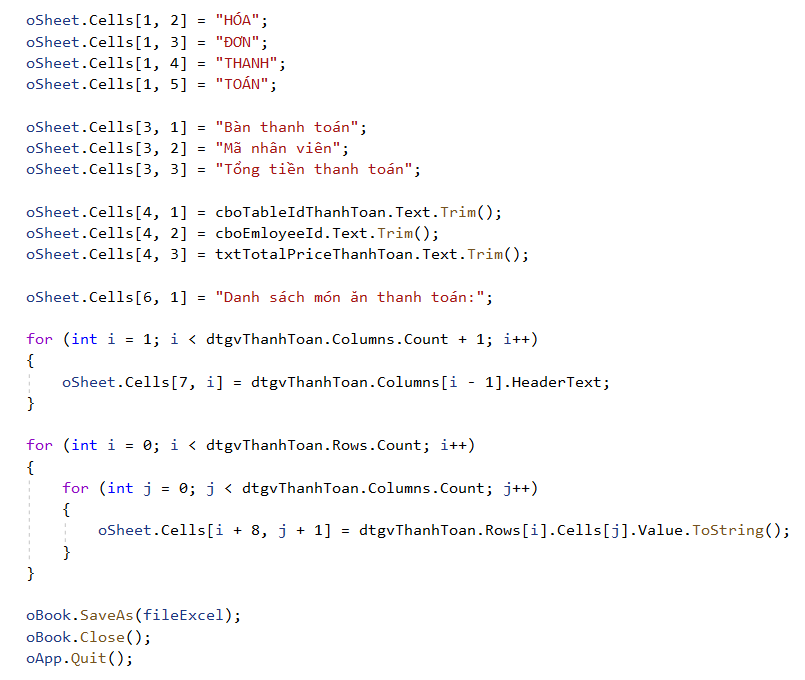
Hình 4.1. Sử dụng thư viện Microsoft.Office.Interop.Excel

Khai báo để có thể sử dụng thư viện.



Hình 4.2. Khởi tạo đường dẫn và trang tính

Đường dẫn để trỏ tới nơi lưu trữ các bản hóa đơn in bằng Excel, khởi tạo trang tính để có thể lưu trữ dữ liệu cần in ra trong đó.



Hình 4.3. Thêm dữ liệu, lưu và đóng trang tính

Thêm dữ liệu vào từng ô trong Excel, dữ liệu gồm các tiêu đề, tên các trường, các giá trị cần in ra. Sau khi thêm xong thì lưu file, đóng trang tính và thoát khỏi

TÀI LIỆU THAM KHẢO

(Theo chuẩn IEEE)

[1]. ‘NET framework là gì? Các khái niệm cơ bản về .NET framework’. https://bizflycloud.vn/tin-tuc/net-framework-la-gi-cac-khai-niem-co-ban-ve-net-framework-20180912090957839.htm (accessed Nov. 10, 2021).

[2]. ‘NET Framework’, *Wikipedia*. Nov. 12, 2021. Accessed: Nov. 14, 2021. [Online]. Available: <https://en.wikipedia.org/w/index.php?title=.NET_Framework&oldid=1054816535>

[3]. Lê Viết Tuấn, *“Lập trình cơ sở dữ liệu”*, lần thứ 1. Thành phố Hồ Chí Minh: Trường Đại học Mở Thành phố Hồ Chí Minh, 2017.

[4] ‘Tìm Hiểu Về Mô Hình 3 Lớp (3 Layer)’, *Viblo*, Jul. 26, 2020. https://viblo.asia/p/tim-hieu-ve-mo-hinh-3-lop-3-layer-Qbq5Qmyz5D8 (accessed Nov. 10, 2021).

[5]. BillWagner, ‘How to access Office interop objects - C# Programming Guide’. https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/interop/how-to-access-office-onterop-objects (accessed Nov. 10, 2021).