**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN (seminar)MÔN HỌC**

Kiến Trúc Phần Mềm

**ĐỀ TÀI:** Áp dụng kiến thức đã học cải thiện ứng dụng

**Giảng viên hướng dẫn:** thầy Trần Minh Triết

**Sinh viên thực hiện:** 19120652 – Nguyễn Trọng Thái

**TP HCM – 2022/2023**

Mục lục

[1. Giới thiệu 1](#_Toc124243924)

[2. Phân tích ứng dụng 1](#_Toc124243925)

[2.1 Kiến trúc ứng dụng 1](#_Toc124243926)

[2.2 Các kỹ thuật sử dụng 1](#_Toc124243927)

[3. Kỹ thuật Model Validations 1](#_Toc124243928)

[4. Sử dụng ủy nhiệm: deledate 2](#_Toc124243929)

[5. Migration 3](#_Toc124243930)

[6. Mẫu singleton 4](#_Toc124243931)

[Tài liệu tham khảo 4](#_Toc124243932)

# Giới thiệu

* Ứng dụng được phát triền từ yêu cầu: xây dựng mô hình quản lý tiêm chủng ở môn **Phân tích thiết kế hệ thống thông tin. Gồm 4 thành viên**

+ Nguyễn Phước Sâm

+ Nguyễn Mạnh Tiến

+ Ngô Mậu Trường

+ Nguyễn Trọng Thái

* Ứng dụng sẽ được tiếp tục phát triển trong môn **Kiến trúc phần mềm. Thành viên**

**+** Nguyễn Trọng Thái

* Chức năng sẽ được áp dụng: Các chức năng của Nhân viên tiếp tân

+ Thêm phiếu đăng ký

* Mục tiêu:

+ Hiểu và áp dụng vào ứng dụng các pattern và cách cách quản lý kiến trúc phần mềm.

+ Chỉ áp dụng vào một phần nhỏ chủ yếu sử dụng nhiều kỹ thuật hay giúp quản lý tốt ứng dụng

# Phân tích ứng dụng

- Để login nhập

+ Tài khoản: NV000001

+ Mật khẩu: thai2001

## 2.1 Kiến trúc ứng dụng

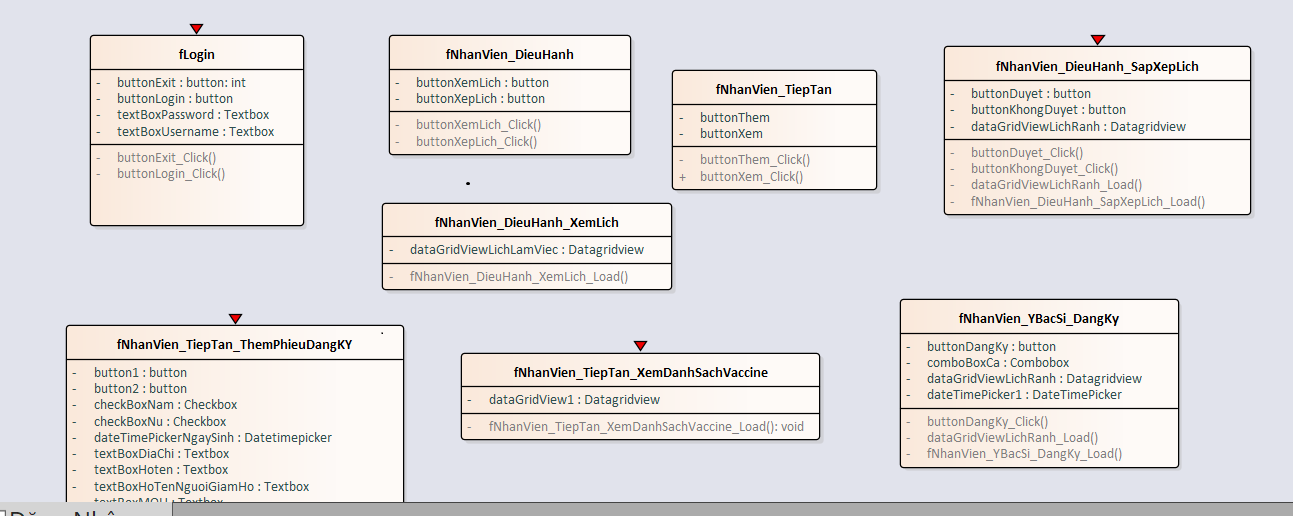
* Sử dụng mô hình 3-tier

+ Tầng view: để quản lý form và xử lý dữ liệu lên xuống form

+ Tầng BUS để tiếp nối và xử lý các yêu cầu giữa tầng view và tầng DAO(model)

+ Tầng DAO quản lý các lớp dữ liệu, lưu trữ ở database

- View



- Model

Diagram, engineering drawing

Description automatically generated

## 2.2 Các kỹ thuật sử dụng

* Model Validations
* Biến Delegate
* Migration Database
* Mẫu singleton

# Kỹ thuật Model Validations

* Model cần kiểm tra dữ liệu đầu vào xem có hợp lệ không. Quá trình kiểm tra này có thể được hoàn thành bởi client trước khi gửi lên server hoặc server kiểm tra sau khi nhận được từ client. Cơ chế validation phía client (client-side validation) rất quan trọng vì nó giúp tăng trải nghiệm người dùng khi kiểm tra dữ liệu mà không cần chờ đến server nhưng phía server lại nên đảm bảo một lần nữa để các dữ liệu không hợp lệ không thể đưa vào hệ thống.
* Sử dụng ở phần view để kiểm tra đầu vào xem có hợp lệ không. Các form đã sử dụng

+ Flogin

+ fNhanVien\_TiepTan\_ThemPhieuDangKy

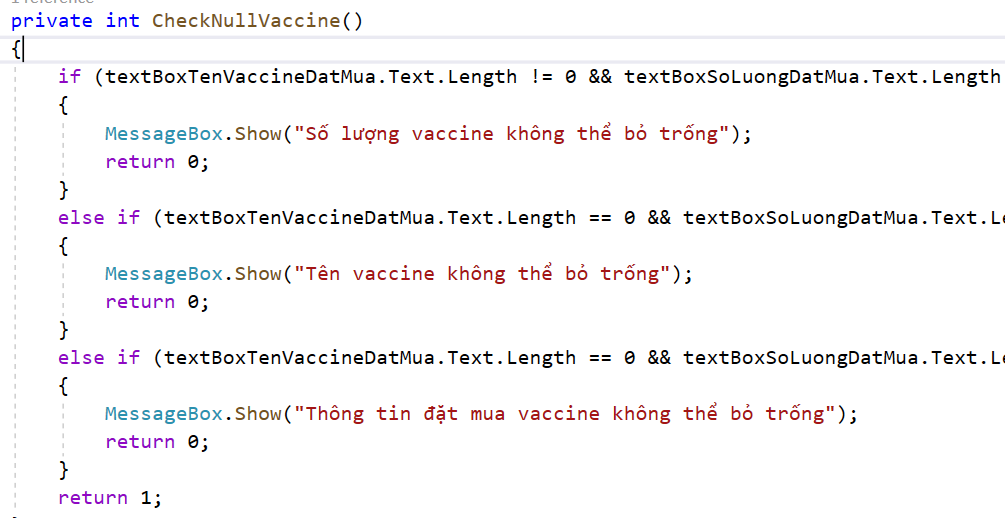
* Lợi ích:

+ Giúp đỡ tốn thời gian truy vấn. Ví dụ cơ sở dữ liệu lớn: nhiều công ty sẽ sử dụng cơ sở dữ liệu từ các tổ chức khác. Việc trả tiền sẽ phụ thuộc vào việc số dữ liệu mỗi lần truy vấn nên quá trình này giúp giảm chi phí và thời gian rất nhiều

+ Đỡ gánh nặng cho controller: việc chuyển đổi dữ liệu dữ view – BUS – DAO(view – controller - model) sẽ tốn thời gian và có thể gây quá tải cho controller

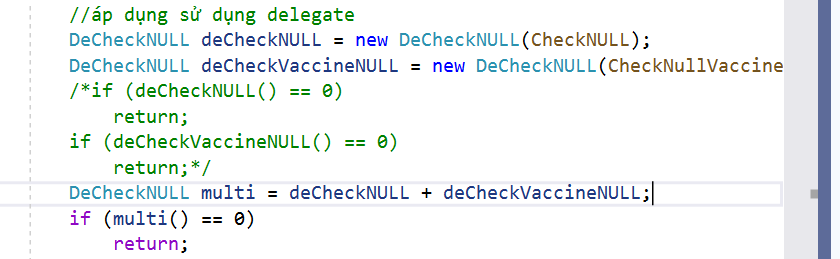
Graphical user interface, text, application

Description automatically generated



# Sử dụng ủy nhiệm: deledate

* Delegate là một biến tham chiếu đến Func
* Delegate cho phép một class uyển chuyển và linh động hơn trong việc sử dụng phương thức. Theo đó, nội dung cụ thể của một phương thức không được định nghĩa sẵn trong class mà sẽ do người dùng class đó tự định nghĩa trong quá trình khởi tạo object. Điều này giúp phân chia logic của một class ra các phân ra các phần khác nhau và do những người khác nhau xây dựng.
* Delegate được sử dụng để giúp một class object tương tác ngược trở lại với thực thể tạo ra và sử dụng class đó. Điều này giúp class không bị cô lập bên trong thực thể đó
* Với khả năng tạo ra tương tác ngược như vậy, delegate trở thành hạt nhân của lập trình hướng sự kiện.
* Sử dụng trong đồ án: dùng để gọi một loạt hàm 1 lần



* Sử dụng để gọi tuần tự không cần gọi từng hàm

# Migration

* **Migration** là **kỹ thuật** trong việc tương tác với cơ sở dữ liệu, theo đó việc thay đổi về cấu trúc CSDL ở code sẽ được cập nhật lên CSDL đảm bảo dữ liệu đang tồn tại không bị mất, lịch sử (phiên bản) cập nhật được lưu lại sau mỗi lần cập nhật.
* Cách sử dụng:

+Ở manager console gõ lệnh: enable-migrations để kết kết nối. Lúc này chúng ta sẽ có danh sách các bảng

+Tiếp theo gõ lệnh: add-migration <name> để thêm mới quản lý db. Lúc này sẽ xuất hiện bản cập nhật mới so với db

+ Gõ lệnh **Update-Database** để cập nhật database

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

# Mẫu singleton

* Trong quá trình hoạt động việc “**đăng ký và đặt mua vaccine”** sẽ xuất hiện rất nhiều nên mẫu singleton sẽ giúp giải quyết vấn đề sự cố về hiệu năng hệ thống. Cùng một thời điểm, hệ thống đang sử dụng một lúc nhiều đối tượng và chúng làm tiêu tốn quá nhiều tài nguyên của hệ thống. Đây là vấn đề mà hệ thống cần phải khắc phục, và Singleton pattern có thể giúp bạn thực hiện được điều đó.

Graphical user interface, text, application, email

Description automatically generated

# Tài liệu tham khảo

1. <https://phuxuan.edu.vn/dao-tao/cong-nghe-thong-tin-mo-hinh-3-tier-va-3-layer/>
2. <https://tuhocict.com/tham-dinh-du-lieu-input-validation-trong-aspnet-core/#:~:text=Th%E1%BA%A9m%20%C4%91%E1%BB%8Bnh%20d%E1%BB%AF%20li%E1%BB%87u%20(data,tri%E1%BB%83n%20%E1%BB%A9ng%20d%E1%BB%A5ng%20n%C3%B3i%20chung>.
3. <https://tuhocict.com/delegate-trong-csharp/#:~:text=Delegate%20l%C3%A0%20nh%E1%BB%AFng%20ki%E1%BB%83u%20d%E1%BB%AF,di%E1%BB%87n%20ho%E1%BA%B7c%20ki%E1%BB%83u%20%E1%BB%A7y%20nhi%E1%BB%87m>.
4. <https://xuanthulab.net/ef-core-tao-migration-trong-entityframework-voi-c-csharp.html>
5. <https://viblo.asia/p/design-patterns-singleton-pattern-maGK7zra5j2>